



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



ENERHOMGY

Gestión de la demanda eléctrica domiciliaria

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE LA REPÚBLICA POR

Juan Korenko

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS PARA LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN COMPUTACIÓN.

Diego Acuña y Lucía Korenko

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS PARA LA
OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRICISTA.

TUTORES

Ariel Sabiguero..... Instituto de Computación
Álvaro Giusto..... Instituto de Ingeniería Eléctrica

CO-TUTOR

Franco Robledo Instituto de Computación

TRIBUNAL

Daniel Meerhoff..... Instituto de Computación
Leonardo Steinfeld Instituto de Ingeniería Eléctrica
Mario Vignolo..... Instituto de Ingeniería Eléctrica

Montevideo
miércoles 22 junio, 2016

ENERHOMGY: *Gestión de la demanda eléctrica domiciliaria*, Diego Acuña, Juan Korenko,
Lucía Korenko.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 231 páginas.
Compilada el lunes 25 julio, 2016.

Agradecimientos

El presente trabajo no podría haber sido posible sin la colaboración de muchas personas que de una manera u otra aportaron a la realización del mismo.

En primer lugar queremos agradecer a nuestras familias y amigos por el apoyo incondicional y la comprensión brindada durante el proceso. A nuestros tutores por haber confiado en nosotros, orientado y asesorado durante el proyecto. A Gonzalo Belcredi, Pablo Modernell y Nicolás Sosa del proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente”, cuya plataforma sirvió como base para este trabajo, por su guía en la utilización de la misma.

También agradecemos a la Fundación Julio Ricaldoni de la Facultad de Ingeniería y a la ANII (Agencia Nacional de Investigación e Innovación), por la financiación del proyecto a través de la convocatoria CII (Centro de Innovación en Ingeniería), así como al directorio de UTE (Administración Nacional de Usinas y Trasmisiones Eléctricas) por el interés demostrado por el proyecto.

Adicionalmente, queremos mencionar a todos los que participaron en los seminarios brindados para el intercambio de conocimiento que ayudaron a enriquecer nuestro trabajo. A los institutos de Ingeniería Eléctrica e Instituto de Computación de la facultad por facilitarnos los materiales necesarios para la realización del proyecto. Queremos agradecer además al Plan de Obras de la Facultad de Ingeniería, especialmente a Adrian Santos, por la colaboración en la instalación física del laboratorio para llevar a cabo la prueba experimental.

Finalizamos nuestro agradecimiento recordando a todos aquellos compañeros de estudio y docentes que nos acompañaron hasta esta instancia de la carrera por haber contribuido a nuestra formación profesional y personal.

A todos MUCHAS GRACIAS!

Resumen

En los últimos años se han aplicado distintas políticas para modificar la matriz energética, promoviéndose esencialmente la incorporación de nuevas fuentes de energía renovable. Estas fuentes de energía presentan características distintas a las utilizadas históricamente, es decir la energía hidroeléctrica y la proveniente de combustibles fósiles. Esto junto con el constante crecimiento de la demanda de energía hacen que los sistemas eléctricos sean cada vez más difíciles de operar y manejar. En la actualidad existen múltiples líneas de investigación que buscan darle mayor inteligencia a la red eléctrica con el fin de que ésta sea más eficiente y más confiable.

Una estrategia innovadora, conocida como Respuesta a la Demanda o *Demand Response* (DR), propone obtener el equilibrio entre la generación y la demanda ajustando no sólo la generación, como se hace históricamente, sino también manipulando la demanda de energía. Se cree que las políticas de Respuesta a la Demanda permitirán reducir costos y mejorar la confiabilidad del sistema. En particular, este proyecto se enfoca en las políticas de Respuesta a la Demanda a nivel residencial en las cuales la empresa suministradora motiva a los clientes a moldear su consumo a través de una tarifa dinámica de precios. Se desarrolló un prototipo sistema de gestión a instalar en los hogares de los clientes a fin de que estos puedan adoptar fácilmente este tipo de tarifas. El sistema de gestión implementado gestiona los electrodomésticos del hogar de acuerdo al precio de la energía y los deseos del usuario.

Para lograr esto, en primer lugar, se desarrolló un algoritmo de gestión encargado de gestionar los distintos electrodomésticos de un hogar. Para construir el mismo se realizó una extensa revisión bibliográfica de algoritmos y se diseñó uno que se ajustara al problema a atacar. Adicionalmente, se implementó un sistema capaz de accionar los electrodomésticos en función de las decisiones tomadas por el algoritmo de gestión. Para este propósito se utilizó la plataforma desarrollada en el proyecto de grado “Controlador de energía domiciliario para una Red Eléctrica Inteligente” por Belcredi, Modernell y Sosa. Para que el cliente pueda configurar el sistema de gestión a su gusto, se desarrolló una interfaz web a través de la cual puede consultar las distintas variables del sistema y realizar ajustes según sus preferencias.

Finalmente, se realizó una evaluación integral del sistema desarrollado. Se comprobó el correcto funcionamiento del sistema en una prueba experimental con electrodomésticos reales. Además se realizó una simulación del funcionamiento del algoritmo de gestión en distintos hogares tipo a fin de evaluar su desempeño y qué tan conveniente es para el usuario. A través de la evaluación realizada se concluyó que el algoritmo de gestión es bueno, pero para que la instalación de este tipo de sistema sea atractiva para los clientes es necesario reducir los costos de la plataforma.

Tabla de contenidos

Agradecimientos	I
Resumen	III
1. Introducción	1
1.1. Motivación	1
1.2. Objetivo y alcance	2
1.3. Estructura del documento	3
2. Estado del arte y revisión bibliográfica de algoritmos de gestión	5
2.1. <i>Smart Grid</i>	5
2.2. Respuesta a la Demanda	8
2.3. Respuesta a la Demanda a nivel residencial	11
2.4. Algoritmo de gestión de cargas	13
3. Descripción del sistema HEMS	19
3.1. Descripción del escenario del problema y sus actores	19
3.2. Arquitectura del HEMS	20
3.3. Descripción del hardware utilizado	22
3.4. Nodos de Carga	23
3.5. Red HAN	24
4. Algoritmo de gestión	27
4.1. Descripción del algoritmo de gestión	27
4.2. Selección de electrodomésticos a controlar	30
4.3. Modelo del aire acondicionado	31
4.4. Modelo del calefón	35
4.5. Modelo de electrodomésticos de consumo discreto	40
4.6. Restricción del hogar	42
5. Implementación del Nodo Controlador	45
5.1. Descripción del Nodo Controlador	45
5.2. Arquitectura del Nodo Controlador	46
5.3. Clases de electrodomésticos	48
5.4. Módulo Configurador	51
5.5. Módulo Optimizador	54
5.6. Módulo Intérprete	57
5.7. Módulo Actuador	61

Tabla de contenidos

5.8. Módulo Registro Mediciones	64
5.9. Programa principal	65
6. Interfaz de usuario	69
6.1. Descripción de la interfaz de usuario	69
6.2. Herramientas utilizadas	69
6.3. Funcionalidades de la Interfaz Web	72
6.4. Seguridad en la Interfaz Web	75
7. Prueba Experimental	79
7.1. Objetivos	79
7.2. Laboratorio de pruebas	80
7.3. Nodos de Carga	82
7.4. Experimento realizado	84
7.5. Evaluación de la prueba experimental	89
8. Evaluación del Sistema	91
8.1. Simulación del funcionamiento del algoritmo	91
8.2. Análisis económico	97
8.3. Cualidades del sistema HEMS	102
8.4. Trabajo a Futuro	104
9. Conclusiones	109
9.1. Conclusiones generales	109
9.2. Dificultades y aprendizajes	110
Apéndices	111
A. Descripción de la ideas planteadas en [18]	111
B. Detalle del modelado de los electrodomésticos	117
C. Determinación de los parámetros de la función de utilidad	133
D. Atributos y funciones de las clases electrodomésticos	139
E. Caracterización del problema de optimización y librería CVXOPT	147
F. Concurrencia en Python	155
G. Módulo wsncommunication	159
H. Tablas de Base de Datos	163
I. Alternativas interfaz de usuario	169
J. Detalle Interfaz Web	173
K. Configuración de los hogares tipo	189
L. Detalle Nodos de Carga	193
M. Protocolos de las pruebas experimentales	209
N. Gestión del proyecto	213
Referencias	215
Glosario	221

Capítulo 1

Introducción

1.1. Motivación

En las últimas décadas la demanda de energía eléctrica ha presentado un crecimiento exponencial a nivel mundial. Las fuentes de generación tradicionales, hidráulica y combustibles fósiles, no son capaces de equiparar este crecimiento. Esto genera la necesidad de diversificar la matriz energética, incorporando nuevas fuentes de generación promoviéndose fundamentalmente las fuentes de energía renovable.

Las nuevas fuentes de energía incorporadas se encuentren geográficamente dispersas, conformando un sistema de generación desperdigado. Éstas también presentan características distintas a las fuentes de generación tradicionales. Por ejemplo, la energía eólica, que ha crecido mucho en Uruguay en los últimos años, se caracteriza por ser muy variable en cortos períodos de tiempo. Esto, junto con el constante crecimiento en la demanda, hacen que las redes eléctricas se tornen cada vez más complejas de analizar y operar. Con el fin de que la red eléctrica sea más eficiente y confiable nace la necesidad de otorgarle inteligencia a la red. Lo cual actualmente es posible gracias a los grandes avances obtenidos en el área de las telecomunicaciones y del procesamiento de datos. Estas estrategias de innovación en la red eléctrica se las engloba bajo el concepto de *Smart Grid*, campo que es foco actual de estudio en el mundo.

Existen muchas líneas de trabajo dentro de *Smart Grid*, entre las cuales se encuentra la Respuesta a la Demanda o *Demand Response*, en esta línea de trabajo se encuentra enmarcado el presente proyecto. Las políticas de Respuesta a la Demanda proponen que el equilibrio entre la generación y la demanda de energía, se logre no sólo ajustando la generación como se hace tradicionalmente, sino también manipulando la demanda de energía eléctrica. Estas políticas permiten reducir los costos de generación, transporte y distribución, otorgar mayor confiabilidad y reducir la contaminación.

Existen dos maneras de ajustar la demanda, que un agente externo controle directamente las cargas de los clientes o incentivando a los clientes a que ajusten su consumo voluntariamente. En el sector residencial una estrategia común es utilizar tarifas de precio en las cuales el precio de la energía varía a lo largo del día. Bajo la hipótesis de que el consumidor desea minimizar su factura eléctrica, éste buscará reducir su consumo cuando el precio es alto y trasladarlo hacia las horas de menor precio.

Las políticas de Respuesta a la Demanda son más efectivas cuanto mayor es la demanda que pueden gestionar. En el sector residencial es necesario que un alto porcentaje de clientes se adhieran a la tarifa dinámica. Se considera que ningún cliente estaría dispuesto a contratar este tipo de tarifas si esto requiere consultar periódicamente el precio de la energía y operar

Capítulo 1. Introducción

los electrodomésticos manualmente. Es por esto que es necesario que exista un sistema de gestión que opere automáticamente los electrodomésticos en función de las señales de precio. El objetivo de este proyecto es desarrollar y evaluar un prototipo de sistema que cumpla con esta función.

En el proyecto de grado “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1] realizado en el 2014, los estudiantes Belcredi, Modernell y Sosa diseñaron una plataforma abierta destinada a la implementación y prueba de sistemas de gestión de los electrodomésticos. La presente propuesta busca complementar el trabajo realizado por ellos a fin de aprovechar los conocimientos adquiridos y los progresos logrados en el mismo. Buscando dar mayor profundidad al algoritmo de gestión y a la interfaz de usuario, ampliando en este sentido el alcance del trabajo realizado.

1.2. Objetivo y alcance

El objetivo de este proyecto es diseñar e implementar un prototipo de sistema de gestión que opere un conjunto de electrodomésticos de un hogar. El mismo debe tomar las decisiones en función del precio de la energía de una tarifa dinámica, buscando minimizar el costo en energía eléctrica y cumplir con las consignas de confort del usuario.

Los aspectos a estudiar en este proyecto son los siguientes:

- Realizar un estudio del estado del arte referente a los conceptos de *Smart Grid* y de **Respuesta a la Demanda**.
- Realizar una **búsqueda bibliográfica** de las distintas propuestas de algoritmos de gestión enfocados a políticas de Respuesta a la Demanda a nivel residencial.
- Diseñar e implementar un **algoritmo** de gestión para uso residencial. El algoritmo debe satisfacer los criterios de confort del cliente, operar con una tarifa dinámica y obtener un cierto ahorro en el gasto por consumo de energía.
- **Simular** el desempeño del algoritmo en diferentes hogares tipo para evaluar su comportamiento y relevar los resultados obtenidos.
- **Evaluar** los beneficios obtenidos en cada uno de los hogares tipo para estudiar la conveniencia de adquisición de la plataforma. Se debe calcular el costo de la plataforma y compararlo con los ahorros estimados en la simulación.
- **Implementar en hardware** el algoritmo de gestión utilizando la plataforma desarrollada en el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente”. Se debe programar el nodo central de la plataforma para que ejecute el algoritmo de gestión diseñado y opere los electrodomésticos.
- Desarrollar una **interfaz de usuario** a través de la cual el usuario pueda interactuar con el sistema de gestión. La misma debe permitir al usuario configurar sus preferencias de uso y obtener información de interés.
- Realizar una **prueba experimental** del funcionamiento del sistema de gestión desarrollado en un laboratorio con electrodomésticos reales. Esto también incluye adquirir los electrodomésticos para realizar la prueba y las adaptaciones necesarias para que los nodos periféricos puedan accionar los electrodomésticos.

Los siguientes puntos quedaron fuera del alcance de este proyecto:

1.3. Estructura del documento

- Estudiar los métodos de resolución del problema de optimización planteado por el algoritmo de gestión. Se utilizará un biblioteca que implementa un método de resolución y no se estudiará en detalle cómo funciona la misma.
- Evaluar el beneficio que obtiene el resto de los actores del sistema eléctrico al aplicar políticas de Respuesta a la Demanda. Sólo se estimará el beneficio que obtendría el cliente al utilizar el sistema desarrollado.
- Diseñar una tarifa de precios dinámica adaptada a la realidad uruguaya. En las simulaciones del algoritmo se utilizará una curva de precios ficticia a fin de poder estimar el ahorro que obtendría el cliente al utilizar el sistema.
- Mejorar la plataforma desarrollada en el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente”. Se realizarán modificaciones menores a fin de poder probar el sistema de gestión en un laboratorio.
- Desarrollar nodos periféricos genéricos que puedan ser utilizados en una amplia gama de electrodoméstico.

1.3. Estructura del documento

El presente documento reporta el trabajo realizado en este proyecto. A continuación se indica cómo está organizado el mismo.

En el Capítulo 2 se realiza una revisión de los conceptos más importantes dentro de *Smart Grid* y Respuesta a la Demanda. Esto permite brindar un marco y conocer la motivación del presente trabajo. En este mismo capítulo, se muestran los resultado del estudio bibliográfico acerca de los algoritmos de gestión realizado, y se justifica la elección del algoritmo utilizado como base en este proyecto.

Luego se procede a enfocarse directamente en el trabajo desarrollado en este proyecto. En el Capítulo 3 se describe el sistema de gestión definiendo el escenario en el que opera el mismo, su arquitectura y sus componentes. A continuación, en el Capítulo 4 se detalla el algoritmo de gestión utilizado, mostrando la formulación matemática y cómo fueron modelados cada uno de los electrodomésticos controlados por el sistema. En el Capítulo 5 se explica el diseño y la implementación del nodo central del sistema de gestión. En el Capítulo 6 se describe la interfaz de usuario implementada y las funcionalidades esta ofrece al usuario.

Posteriormente, se describe cómo fue evaluado el sistema de gestión desarrollado y cuáles fueron los resultados obtenidos. En el Capítulo 7 se explica la prueba experimental realizada, en la cual se evaluó el desempeño del algoritmo y del sistema de gestión en la práctica. Finalmente en el Capítulo 8 se muestran los resultados de la simulación del algoritmo en los distintos hogares tipo y se evalúan el resto de las características del sistema. Durante la evaluación del sistema y en el transcurso del proyecto, surgieron múltiples modificaciones que mejoraron el desempeño del sistema. En el Capítulo 8 se explican las principales mejoras que podrían realizarse al sistema en futuros trabajos en la materia.

Finalmente, en el Capítulo 9 se presentan las conclusiones generales del proyecto. En este mismo capítulo se cuenta el aprendizaje y las principales dificultades que encontró el grupo durante la ejecución del proyecto.

Capítulo 2

Estado del arte y revisión bibliográfica de algoritmos de gestión

El objetivo de este capítulo es describir el escenario dentro del cual se enmarca este proyecto. Con este fin, se explican las principales ideas detrás del concepto *Smart Grid* y Respuesta a la Demanda, prestándole mayor atención a las políticas de Respuesta a la Demanda a nivel residencial.

En este capítulo además, se describe la búsqueda bibliográfica realizada para seleccionar el algoritmo de gestión de los electrodomésticos que se utiliza como base. Se exponen los principales resultados obtenidos y se justifica la elección del algoritmo utilizado como base en este proyecto.

2.1. *Smart Grid*

La demanda energética mundial registró una tasa de crecimiento exponencial en las últimas décadas. En la Figura 2.1 se muestra un gráfico con la demanda energética anual uruguaya discriminada por fuente de generación. En la misma se puede observar que Uruguay pasó de un consumo energético de 6244 GWh en el año 2004 a uno de 8342 GWh en el 2014, lo que totalizó un aumento del 33,6% en sólo 10 años.

La mayor parte de la energía eléctrica utilizada mundialmente proviene de combustibles fósiles y de represas hidroeléctricas. Uruguay presenta el mismo comportamiento, como se observa en la Figura 2.1. Los combustibles fósiles, es decir petróleo, carbón y gas natural, además de ser muy contaminantes son un recurso finito. De acuerdo a [4], se estima que si se continúa con la tendencia de crecimiento actual de la demanda, las reservas de carbón se agotarán en 125 años, las de gas natural en 55 años y las de petróleo en 37 años. Por otro lado, la energía de origen hidroeléctrico presenta la ventaja de ser renovable y por lo tanto es una fuente inagotable de energía. Éste recurso ya se encuentra ampliamente explotado y las perspectivas de crecimiento de generación son bajas. Considerando esto y el crecimiento en la demanda de energía, es inminente que en algún momento estas fuentes de generación no van ser suficientes para abastecer a la demanda.

La estrategia incorporada a nivel mundial para combatir el problema ha sido la diversificación de la matriz energética incorporando nuevas fuentes de energía. Se ha puesto especial énfasis en las fuentes de energía renovables como energía eólica, solar y geotérmica. Uruguay también ha apostado a la diversificación de su matriz energética. Las fuentes de generación

Capítulo 2. Estado del arte y revisión bibliográfica de algoritmos de gestión

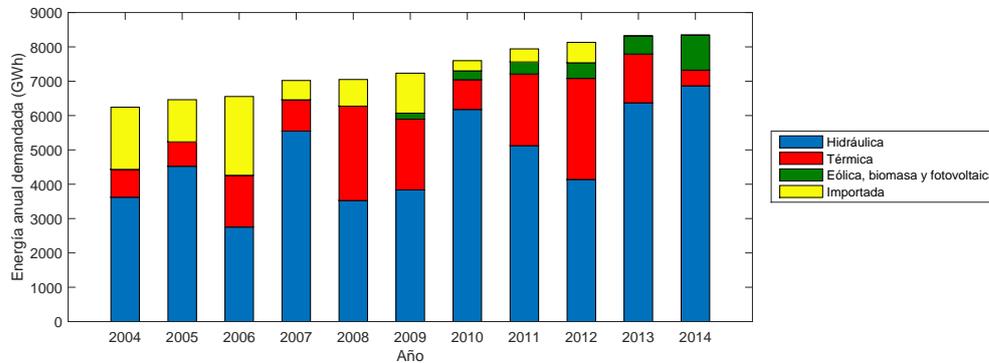


Figura 2.1: Abastecimiento anual de la demanda discriminada por fuente de generación. Fuentes [2] y [3].

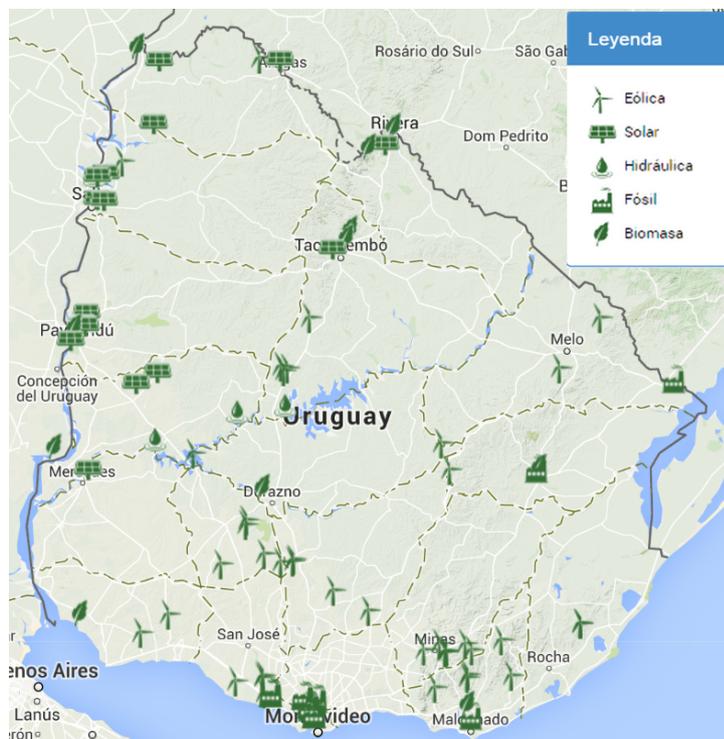


Figura 2.2: Mapa de generación actual de Uruguay. Fuente: [6]

eólica, biomasa y solar han tenido un importante crecimiento en Uruguay desde el 2009, ver Figura 2.1. En particular, se destaca el aumento de la generación eólica que a partir del año 2014 cuadruplicó su aporte[5].

Las nuevas fuentes de generación presentan el inconveniente que deben ser instaladas donde se encuentra el recurso a explotar. Históricamente, las fuentes de energía en un sistema de generación se encontraban concentradas en unos pocos puntos geográficos. La incorporación de nuevas fuentes de energía ha llevado a conformar sistemas de generación desperdigados. En la Figura 2.2 se encuentra el mapa de generación actual de Uruguay discriminado por fuente.

2.1. *Smart Grid*

Los sistemas de generación desperdigados, junto con el constante aumento de la demanda energética, hacen que la red eléctrica se torne cada vez más compleja de analizar y operar. Esto genera la necesidad de otorgarle inteligencia a la red para que funcione de manera más eficiente y más confiable. De este modo nace el concepto red inteligente o *Smart Grid*, el cual es un tema de interés mundial. El paradigma de *Smart Grid* es dotar a la red eléctrica de una mayor inteligencia para poder utilizar de forma óptima los recursos disponibles y brindar una mejor calidad de servicio a los usuarios. Esto es posible gracias a los avances adquiridos a partir de la década de 90 en el campo de la informática, de las telecomunicaciones y de la electrónica de potencia.

Conceptos de *Smart Grid*

En este apartado se encuentran algunos de los proyectos más relevantes del área que permiten migrar a una red más inteligente.

Generación distribuida

La generación distribuida refiere a los generadores que se conectan directamente en la red de distribución. Al estar ubicadas cerca de los puntos de consumo permiten una reducción de las pérdidas técnicas en el transporte de la energía. Las principales fuentes de energía distribuida son el viento, la radiación solar y la biomasa. El estándar IEEE 1547[7] establece los requisitos de operación, seguridad, ensayos y mantenimiento que se deben tener en cuenta para conectar la generación distribuida a la red.

Micro redes

Las redes eléctricas comenzaron siendo pequeños sistemas aislados y posteriormente fueron evolucionando a grandes sistemas centralizados. Los sistemas centralizados ganaron popularidad porque brindan una mejor calidad de la energía y permitían aprovechar los grandes cursos de agua como fuentes de generación. En la actualidad hay muchas centrales generadoras desperdigadas que podrían alimentar de forma aislada a un conjunto de cargas de la red. Lo que representa como principal ventaja que ante una eventual contingencia en la red principal, no se interrumpe el suministro de energía eléctrica a los clientes. Este régimen de operación se conoce como 'isla' y actualmente no está permitido por la mayoría de las empresas suministradoras porque durante la operación en 'isla', no pueden asegurar la calidad del servicio y por los efectos transitorios que se producen en la conexión y desconexión a la red principal¹.

Un avance en esta línea es el concepto de micro redes o *Micro Grid*. La idea es conformar pequeñas redes que pueden ser barrios o pueblos, que cuenten con generación propia y capacidad de almacenamiento de energía. Básicamente son un sistema eléctrico que puede operar de dos formas, integradas a la red centralizada o en 'isla'. Estas redes son un avance en la dirección de descentralización de la red[8].

Almacenamiento de energía en baterías

De la mano de las micro redes toma fuerza el uso de las baterías como forma de almacenar la energía. Un ejemplo clásico es la generación fotovoltaica, en la cual se genera mientras hay luz solar y la potencia que excede a la demanda se utiliza para cargar baterías. Las baterías

¹En necesario agregar infraestructura adicional en la red eléctrica para interconectar de forma segura dos sistemas aislados.

Capítulo 2. Estado del arte y revisión bibliográfica de algoritmos de gestión

aumentan la complejidad en la operación de la micro-red porque agregan un nuevo grado de libertad, ya que en cada momento se debe elegir si es más conveniente utilizar energía proveniente de la red o de las baterías. Para ello se debe agregar una lógica que determine el despacho óptimo, considerando no sólo cuál es la fuente de energía más barata sino que también cuide el desgaste de la batería con sus ciclos de carga y descarga.

Flujos de carga en tiempo real

Un flujo de carga es una herramienta de cálculo utilizada en sistemas eléctricos de potencia para determinar las corrientes y las tensiones que se presentan en la red bajo un determinado escenario de carga. Los flujos de carga son el pilar fundamental para determinar si la infraestructura de la red es la adecuada para transportar la energía demandada. Cada vez es más habitual que las redes tiendan a anillarse para mejorar la calidad de servicio, esto hace que haya más de un camino para transportar la energía a un determinado cliente. A dicho camino se lo llama explotación y mediante los flujos de carga se puede determinar cuál es la explotación más conveniente, es decir, la que presenta mejores tensiones, la que tiene menos pérdidas, la que exige menos a los conductores y a los transformadores.

Tradicionalmente se determina la explotación más conveniente de la red para un determinado escenario de carga. Actualmente es posible realizarlo en tiempo real, con medidas y proyecciones de consumo, lo que permite determinar la explotación óptima para cada instante considerando las variaciones en el consumo. Por lo cual los flujos de carga no sólo serían de utilidad para la planificación, sino también para la operación de la red eléctrica.

EnerNet

El proyecto más ambicioso dentro de *Smart Grid* es llegar a formar lo que se conoce como *EnerNet*, concepto desarrollado por el ingeniero estadounidense Bob Metcalfe[9]. La *EnerNet* es el equivalente para la red eléctrica, de lo que es Internet para las comunicaciones. Se aspira que la red del futuro sea una inmensa malla distribuida en la cuál miles de generadores y consumidores intercambien energía. Para esto todos deben estar interconectados, de forma de coordinar y optimizar los recursos disponibles en la red. Esta es una visión futurista y para alcanzarla se debe avanzar en varios puntos previos y más cercanos en el tiempo.

Demand Response

Otra área de estudio en *Smart Grid* es la llamada Respuesta a la Demanda o *Demand Response* (DR), que define la base en la cual está sustentado este proyecto de fin de carrera. En la siguiente sección se estudia en qué consiste esta política.

2.2. Respuesta a la Demanda

Un sistema eléctrico de potencia puede pensarse como una balanza, donde de un lado se encuentra la generación y del otro la demanda. Para que la báscula se mantenga equilibrada es necesario que a ambos lados el peso sea el mismo. Tradicionalmente lo que se hace es ajustar la generación para que se equipare a la demanda instantánea de forma de mantener el equilibrio del sistema. Este paradigma tradicional es modificado por los métodos de Respuesta a la Demanda. Se plantea que no sólo la generación se mueva para buscar el equilibrio sino que también lo haga la demanda. Los beneficios buscados al aplicar las políticas de Respuesta a la Demanda son[10, 49]:

2.2. Respuesta a la Demanda

1. Reducción de la potencia de generación total, lo que permite utilizar las fuentes de generación más baratas.
2. Reducción de los picos de potencia lo que reduce las pérdidas en la transmisión y distribución.
3. Reducción de los costos de planificación y aumento de la vida útil de la infraestructura existente.
4. Mejora de la calidad del servicio.
5. Reducción de las emisiones de CO₂.

Clasificación de la demanda

Típicamente se clasifica el consumo de energía eléctrica en tres sectores, residencial, comercial e industrial. Cada sector presenta un comportamiento distinto que debe ser tenido en consideración a la hora de diseñar políticas de Respuesta a la Demanda. Las principales características de cada sector son las siguientes[10]:

1. Sector Residencial:

En Uruguay el consumo residencial representa un gran porcentaje del consumo total, en el 2014 representó un 41 % del consumo total del país[2]. Este sector se caracteriza por presentar un pico de consumo al inicio y final de la jornada laboral, porque son las horas en que las personas están más activas en los hogares. Las políticas de Respuesta a la Demanda a nivel residencial buscan mover estos picos de consumo, tratando de equidistribuir la demanda a lo largo del día. El mayor desafío de estas políticas es que los usuarios no tienen un patrón de consumo definido y que es difícil modificar su consumo sin afectar el confort.

2. Sector Comercial:

El consumo denominado comercial es el de los locales de pública concurrencia destinados a fines comerciales, como lo son los almacenes, los cines, los supermercados y los centros comerciales. Las principales componentes de su consumo son el acondicionamiento térmico del local y la iluminación. Los patrones de consumo dependen principalmente del comportamiento operacional del comercio y en un segundo lugar de las condiciones climáticas y del diseño del local.

3. Sector Industrial:

Se denomina consumo de tipo industrial al que está destinado para ser utilizado para confeccionar, elaborar u obtener objetos. Este sector también se lo discrimina por la potencia que solicita a la red eléctrica, en el caso de Uruguay el pliego tarifario los cataloga como medianos (entre 3 kW a 250 kW) o grandes consumidores (mayores que 250 kW). A diferencia del sector residencial, los clientes de este sector presentan un consumo predecible. Los equipos automatizados en las líneas de producción hacen que el consumo sea controlable, siendo posible adelantar o retrasar procesos en coordinación con la empresa suministradora. En el caso de los grandes consumidores, la empresa suministradora además puede diseñar planes de respuesta personalizados para cada cliente de manera que ambas partes se beneficien al máximo. Esto último es muy difícil de llevar a la práctica en el sector residencial, dada la gran cantidad de clientes y la aleatoriedad de sus hábitos de consumo.

Tipos de Respuesta a la Demanda

Cada política de DR se puede clasificar según su mecanismo de control, según el tipo de motivación ofrecida al cliente y según las variables que se controlan. A continuación se describe la clasificación realizada por Vardakas, Zorba y Verikoukis en [10], ver Figura 2.3.

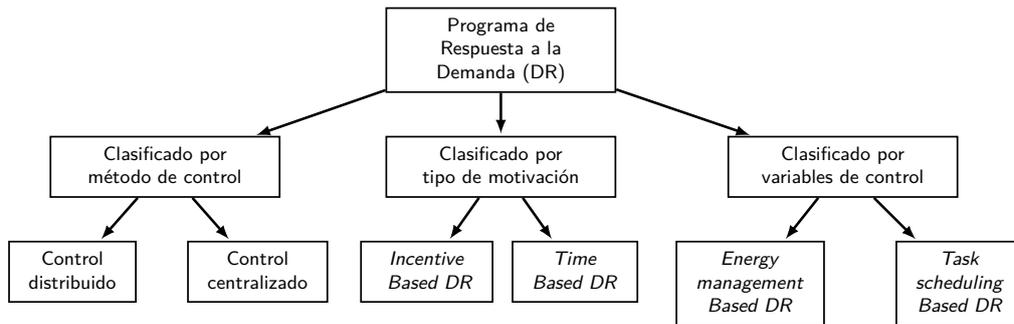


Figura 2.3: Clasificaciones de los programas de Respuesta a la Demanda

Clasificación por mecanismo de control

En primer lugar, es posible clasificar las políticas de Respuesta a la Demanda según quién es el encargado de tomar las decisiones y gestiona el consumo de los clientes.

En una política con **control centralizado** la empresa suministradora, o algún otro agente central, se encarga de tomar las decisiones en función de las necesidades de los clientes. En este tipo de políticas, las redes grandes presentan una gran dificultad por la cantidad de cargas que se deben controlar, lo que requiere una gran infraestructura de comunicación y capacidad de procesamiento de información. El control centralizado es adecuado para ser utilizado en micro redes, ya que es un sistema pequeño y se pueden tomar las decisiones que maximicen en beneficio global.

En una política de **control distribuido** la empresa suministradora brinda una señal a los clientes, generalmente de señales de precios. Los clientes son libres de tomar acciones sobre su consumo en respuesta a la señal recibida. Este método de control descentralizado preserva la privacidad de los usuarios y a su vez reduce la dimensión del problema porque cada usuario es el encargado de tomar decisiones. La desventaja es que la empresa suministradora no tiene control exacto sobre el consumo, por lo cual la eficacia del método depende de que cómo se fijan las señales de precio y qué tan rápido responden los clientes.

Clasificación por tipo de motivación

En la segunda categoría se clasifica considerando el tipo de motivación que ofrece la empresa para que los consumidores se adhieran a la política de Respuesta a la Demanda y/o modifiquen su consumo. Los dos tipos de motivación existentes son *Time-Based DR*, recomendado para clientes residenciales y los *Incentive-Based DR*, el cual es más adecuado para el sector industrial.

Por un lado, en políticas tipo **Time-Based DR** la empresa suministradora brinda una curva de precios variable. Se espera que cada cliente muestre sensibilidad ante las variaciones de precios moviendo su consumo hacia las horas de menor precio. En el global, se consigue moldear el consumo total de todos los clientes.

2.3. Respuesta a la Demanda a nivel residencial

Por otro lado, en *Incentive-Based DR* la empresa suministradora ofrece incentivos monetarios a los usuarios que reduzcan su consumo en períodos de pico o de exigencia de la red eléctrica. La participación de los consumidores en esta política es voluntaria y se realizan penalizaciones a los clientes que declaren una cierta reducción de consumo y posteriormente no la cumplan.

Clasificación por variable de control

La tercera y última clasificación planteada es según la variable sobre la cual se actúa para obtener cambios de la demanda. En esta clasificación también existen dos categorías:

En *Task scheduling based DR* la variable de control es el momento en el cuál se deben encender las cargas. Los dispositivos se dividen en dos tipos, los que obligatoriamente deben ejecutar en determinado instante y los gestionables que se pueden pausar/reanudar.

En *Energy management based DR* se controla la energía que consume cada dispositivo en un intervalo de tiempo. La principal dificultad de este método es que a la mayoría de los dispositivos no se les puede indicar cuánta energía deben consumir, sino que por lo general se los puede encender, pausar o apagar.

2.3. Respuesta a la Demanda a nivel residencial

Actualmente no se dispone de la infraestructura necesaria para involucrar a un gran número de clientes en las políticas de Respuesta a la Demanda. Por esa razón casi todas las políticas están dirigidas a grandes consumidores de energía, pertenecientes a los sectores Industrial y Comercial[11]. No obstante, esta situación está cambiando y en el futuro será posible aplicar políticas de Respuesta a la Demanda masivas.

Control centralizado vs. distribuido

Como se describió en la sección anterior, las políticas de Respuesta a la Demanda pueden ser clasificadas según su mecanismo de control. Por un lado se encuentran las política centralizadas, en las cuales un agente externo toma las decisiones para un conjunto de hogares. Por otro lado, están los mecanismos distribuidos en los cuales las decisiones de cómo operar cada electrodoméstico son tomadas en cada hogar en función del precio de la energía o algún incentivo que se ofrece.

El primero precisa conocer los electrodomésticos presentes en cada hogar y las preferencias de consumo de todos los clientes. Esto tiene como ventaja que el agente externo dispone de más información, y por lo tanto posiblemente llegue a una decisión más óptima. Sin embargo, es posible que los clientes residenciales no deseen proveer esta información por cuestiones de privacidad. Además el manejo de los electrodomésticos del hogar por un agente externo puede generar cierta desconfianza en los clientes. Por lo tanto, en general las políticas de Respuesta a la Demanda a nivel residencial son de tipo distribuidas.

Tarifas de precios

En una política de Respuesta a la Demanda de tipo distribuida es necesario que la empresa suministradora envíe señales a los clientes para que estos moldeen su consumo. En el sector residencial es común utilizar señales de precio que indican cuál es el costo de consumir energía en cada instante de tiempo (*Time-Based DR*). Existen varios tipos de tarifas. A continuación se describen los principales tipos de tarifas utilizados a nivel residencial.[10, 12]

Capítulo 2. Estado del arte y revisión bibliográfica de algoritmos de gestión

En una tarifa **Flat Pricing (FP)** el precio de la energía es constante a lo largo del día. En esta tarifa se le cobra a los clientes según el costo de abastecimiento promedio, el cual que no refleja el costo real que conlleva suministrar energía en cada instante. Esta es la tarifa utilizada históricamente a nivel mundial y se caracteriza por ser sencilla y completamente predecible. Sin embargo, la misma no incentiva a los clientes a modificar sus hábitos de consumo.

En una tarifa **Time Of Use (TOU)** el día se divide en franjas horarias con distintos valores de precios. En este tipo de tarifas, la estrategia usual es establecer precios altos en las horas pico de consumo. De esta manera se incentiva a los clientes a mover su consumo hacia las horas más baratas, reduciendo así el pico en la demanda. Un ejemplo de tarifa TOU es la tarifa inteligente que actualmente UTE (Administración Nacional de Usinas y Trasmisiones Eléctricas) ofrece a sus clientes. Las tarifas TOU se caracterizan por que las franjas horarias y los precios varían con poca asiduidad, por ejemplo, una vez al año. Se trata de una tarifa fácil de interpretar para el cliente lo que facilita su inserción en el sector residencial. Sin embargo, la misma no es flexible a corto plazo, lo que no permite cumplir con todos los objetivos planteados para las políticas de Respuesta a la Demanda.

En una tarifa **Critical Peak Pricing (CPP)** se establece una tarifa base, que puede ser una tarifa plana o TOU, y un precio más alto llamado precio de pico crítico. La empresa suministradora anuncia, con un día de anticipación o dentro del mismo día, la existencia de horarios críticos en los cuales rige el precio de pico crítico. En general, existe un pre-acuerdo entre la empresa y los clientes que restringe el total de horas que pueden ser declaradas como críticas en un año. Esta tarifa combinada con TOU presenta una mayor flexibilidad a corto plazo y por ello es más adecuada para utilizarse en los métodos de Respuesta a la Demanda.

La tarifa **Peak Day Rebates (PDR)** es similar a la tarifa CPP, se establece una tarifa base y la empresa suministradora puede anunciar horarios críticos en el corto plazo. Durante los horarios críticos el cliente decide voluntariamente si mantener su tarifa habitual o que se le pague a cambio de reducir su consumo. El principio de esta tarifa y CPP son equivalentes. No obstante, la tarifa Peak Day Rebates resulta más atractiva al cliente ya que la participación es voluntaria y se ofrece una recompensa a quien participe. Por otra parte, esta tarifa requiere contar con una predicción del consumo del cliente para calcular cuál es la reducción en las horas críticas. Para esto se requieren recursos adicionales en comparación con CPP y además existe la posibilidad de que los clientes jueguen con su consumo habitual de forma de que su reducción en las horas críticas parezca mayor.

Las tarifas más dinámicas son las tarifas horarias, en las cuales el precio de la energía varía hora a hora. La empresa suministradora puede publicar los precios con un día de anticipación en una tarifa **Day-Ahead Pricing (DAP)** o con una hora de anticipación en **Real-Time Pricing (RTP)**. Los precios se calculan en función de la estimación de potencia demandada y la generación disponible en cada hora. Las tarifas horarias son las que dan mayor poder a las políticas de Respuesta a la Demanda. Sin embargo, son las menos atractivas para los clientes por la incertidumbre de precios y su complejidad[13].

Desafíos

Existen dos grandes desafíos para la inserción de las políticas de Respuesta a la Demanda en el sector residencial.

Infraestructura necesaria

El primer desafío es la falta de infraestructura necesaria para implementar políticas de Respuesta a la Demanda. En particular en Uruguay, la mayoría de los hogares cuenta con

2.4. Algoritmo de gestión de cargas

la infraestructura necesaria para una tarifa plana. Esto es un medidor de energía que lleva registro del acumulado histórico de consumo del hogar. Mensualmente se releva el registro histórico, se calcula la diferencia con el mes anterior y esto es lo que se le cobra al cliente.

Bajo otro tipo de tarifa sería necesario contar con un medidor inteligente que mida el consumo de energía por hora, o por franja horaria. Adicionalmente, es necesario implementar canales de comunicación entre la empresa suministradora y los clientes a través de los cuales la empresa obtiene los valores registrados por el medidor inteligente, y publica los cambios en la tarifa de precios.

El costo de desplegar toda esta infraestructura es alto. Históricamente, el beneficio obtenido por utilizar tarifas dinámicas no justificaba la inversión inicial necesaria. En los últimos años, el costo de los medidores inteligentes ha bajado y ofrecen funcionalidades adicionales que hacen que sea más atractiva su instalación por parte de las empresas suministradoras [12].

Atracción de clientes

El éxito del uso de tarifas dinámicas para moldear la demanda en el sector residencial depende fuertemente del número de clientes que adopten la tarifa y cómo responden a las variaciones en la curva de precios.

Los participantes de planes piloto con tarifas dinámicas han mostrado gran satisfacción con este tipo de tarifas. Sin embargo, los estudios demuestran que generalmente los clientes residenciales prefieren las tarifas planas (*FP*), por su sencillez y porque no presentan riesgo económico[12]. Actualmente existe una porción de clientes dispuestos a asumir los riesgos de una tarifa dinámica. El número de clientes interesados puede incrementarse con mayor educación, mejor marketing, difusión de las opiniones positivas de los primeros participantes y tarifas más atractivas.

Sistema HEMS

En el caso de las tarifas horarias, podría pensarse en una primera instancia que el cliente puede gestionar su consumo consultando el precio de cada hora. No obstante, esto es muy poco práctico y ningún cliente estaría dispuesto a consultar periódicamente los precios y operar los electrodomésticos manualmente. Para facilitar la inserción de las tarifas dinámicas y maximizar la respuesta de los clientes, es necesario que exista un sistema automatizado que gestione las cargas del hogar en función del precio de la energía y las preferencias del usuario [12, 14, 15, 16]. Este sistema recibe múltiples nombres en la bibliografía: “Home Energy Management System”(HEMS), “Residential Energy Management System”(REMS), “Energy Management System”(EMS) y “Load Manager Household”, entre otros[16]. En el presente trabajo se optó por utilizar el término HEMS para referirse al mismo.

2.4. Algoritmo de gestión de cargas

Como se describió en el Capítulo 1 el objetivo de este proyecto es implementar un prototipo de sistema HEMS. El sistema HEMS debe presentar un algoritmo de gestión de cargas que decide cómo se debe agendar el consumo de los electrodomésticos, en función de la curva de precios y los requerimientos del usuario[17]. Este algoritmo es un aspecto muy importante dentro del proyecto, dado que el éxito del sistema diseñado depende en gran medida de qué tan efectivo es el algoritmo de gestión.

Con el fin de seleccionar correctamente el algoritmo se realizó una búsqueda y estudio de la bibliografía existente. El objetivo de esta sección es describir la búsqueda bibliográfica

Capítulo 2. Estado del arte y revisión bibliográfica de algoritmos de gestión

realizada, detallándose los parámetros de búsqueda utilizados y los resultados más interesantes, para finalmente justificar la elección del algoritmo base seleccionado.

Parámetros de búsqueda

La bibliografía existente que trata sobre algoritmos de gestión es muy vasta. Fue preciso acordar un conjunto de características deseables que debe presentar el algoritmo de gestión, a fin de acotar el tiempo necesario para la búsqueda y centrar el estudio en el área de mayor interés.

Las características deseables son las siguientes:

- enmarcarse en una política de Respuesta a la Demanda con un mecanismo de control distribuido y señales de precios;
- responder a una tarifa horaria de tipo: DAP, RTP u otra de similares características;
- determinar las acciones a tomar sobre un conjunto de electrodomésticos;
- ser aplicable a nivel residencial;
- tener como objetivo minimizar el costo en energía eléctrica sin descuidar el confort de los usuarios;
- cuidar que se cumplan las restricciones eléctricas del hogar;
- ser implementable en una placa controladora.

Se buscaron y estudiaron algoritmos en la bibliografía existente, buscando la compatibilidad con las características mencionadas. En el trabajo “A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms”[10] se realizó una revisión muy completa de distintos programas de Respuesta a la Demanda y de algoritmos de gestión. Se utilizó este trabajo como punto de partida para la búsqueda bibliográfica, sin restringirse sólo a los trabajos citados en el mismo.

Resultados del estudio bibliográfico

El diseño de un algoritmo de gestión de cargas de un hogar bajo una tarifa de precios dinámica es un problema ampliamente estudiado. A continuación se describen los resultados más relevantes encontrados durante la búsqueda bibliográfica. En la Tabla 2.1 se realiza una comparación de los algoritmos propuestos en la bibliografía consultada.

Referencia	Tipo cargas	Función objetivo	Tarifa	Incertidumbre precios
[18]	Genérico	Minimizar costo y maximizar utilidad	DAP	No
[14]	Genérico	Minimizar costo y tiempo de espera	RTP	No
[19]	Genérico	Minimizar costo y maximizar utilidad	RTP	Si
[13]	Genérico	Minimizar costo y maximizar utilidad	DAP	No
[17]	Genérico	Minimizar costo y maximizar beneficio	TOU	No
[20]	Genérico	Minimizar costo	RTP	Si
[21]	Cargas discretas	Minimizar costo	RTP	Si
[22]	Calefón	Minimizar costo	RTP	No
[23]	HVAC	Minimizar costo	DAP	No

Tabla 2.1: Comparación algoritmos distribuidos

2.4. Algoritmo de gestión de cargas

Dentro de los algoritmos de gestión que cumplen con las características deseadas se distinguen dos clases de algoritmos. Primero se encuentran los algoritmos genéricos, aquellos que contemplan todos los electrodomésticos del hogar en su conjunto. Posteriormente se encuentran los algoritmos por dispositivo, aquellos que toman decisiones para un único electrodoméstico. Estos últimos presentan la ventaja de estar completamente adaptados al electrodoméstico en cuestión y por lo tanto es posible que tomen mejores decisiones. Sin embargo, los mismos no tienen en consideración el resto de los electrodomésticos por lo cual es necesario contar con una lógica central que coordine las decisiones tomadas por cada uno de los algoritmos y cuide de las restricciones del hogar. En otras palabras, si cada electrodoméstico es optimizado de forma individual, es altamente probable que todos decidan encender en las horas más baratas, lo cual puede provocar que se viole la restricción de potencia contratada del hogar.

Algoritmo por dispositivo:

Se entiende por algoritmos por dispositivo aquellos que están enfocados a gestionar un electrodoméstico del hogar. Tal como se describió anteriormente estos presentan la ventaja de ser estar completamente adaptados, pero requieren de una lógica centralizada que coordine las decisiones tomadas por los mismos.

El algoritmo propuesto en [21] está diseñado para electrodomésticos de consumo discreto. Es decir, aquellos que deben funcionar por una determinada cantidad de tiempo para cumplir su función, al usuario le interesa que la tarea finalice antes de una hora determinada, pero no le afecta el momento específico en el cual se activa el electrodoméstico. Dentro de este tipo se encuentran: lavarropas, lavavajillas, sistema de riego, bombeo de agua de una piscina.

En [21] se plantea un algoritmo para una tarifa RTP que toma la decisión, sobre el encendido o apagado del electrodoméstico en cada uno de los intervalos de tiempo, en base a un diagrama de Markov. El algoritmo busca tomar las decisiones que minimicen el costo futuro de la energía consumida, en función del valor esperado de los precios. En el mismo se plantea separar los electrodomésticos a controlar en dos categorías:

- No interrumpibles: una vez que se toma la decisión de iniciar la tarea la misma no se puede detener hasta que finalice.
- Interrumpibles: en cada paso de tiempo se puede tomar una nueva decisión.

Este algoritmo es adecuado para tratar cargas de consumo discreto, no así cargas del tipo de acondicionamiento térmico. Las cargas de acondicionamiento térmico tienen como objetivo conseguir una temperatura en determinado intervalo de interés, lo que no es tenido en cuenta por el algoritmo. De acuerdo a lo desarrollado anteriormente, se puede utilizar el algoritmo propuesto combinado con otros. Con lo cual se podrían incorporar algoritmos específicos para las cargas de acondicionamiento térmico. Por ejemplo, en [22] se propone un algoritmo específico para un calefón², y en [23] otro específico para aire acondicionado.

Algoritmos por hogar:

Se entiende por algoritmos por hogar aquellos en los cuales la toma de decisión se realiza nivel de cada hogar contemplando todos los electrodomésticos conjuntamente.

Existen diversos estudios en los que se plantea el diseño de este tipo de algoritmos como un problema de optimización en el cual se cuenta con una función objetivo que se desea maximizar

²Un calefón, o termotanque, es un dispositivo eléctrico que permite elevar la temperatura del agua caliente para su uso doméstico.

Capítulo 2. Estado del arte y revisión bibliográfica de algoritmos de gestión

y un conjunto de restricciones que se deben cumplir. Los estudios difieren principalmente en cuál es la función objetivo utilizada y cómo se plantean las restricciones.

Medición del confort del usuario:

En [14] se propone que cada hogar gestione su consumo energético de manera de minimizar el costo y a la vez minimizar el tiempo de espera. Se plantea que el usuario va obtener una mayor satisfacción cuanto antes se ejecute la tarea programada, por lo que se utiliza el tiempo de espera para medir la disconformidad del usuario. Por otro lado en [18] se propone minimizar el costo en energía eléctrica y maximizar la utilidad de los electrodomésticos. La función de utilidad cuantifica la satisfacción que obtiene el usuario de acuerdo a cuánta energía se le suministra en cada instante a la carga. El planteo de minimizar el tiempo de espera parece ser apropiado sólo para un conjunto acotado de cargas, mientras que la función de utilidad propuesta en [18] es más flexible y por lo tanto parece ser aplicable de forma genérica.

Clasificación de electrodomésticos:

En [19] y [13] se plantean algoritmos que utilizan la función de utilidad al igual que en [18]. Si bien estos trabajos son similares entre sí, difieren en cómo se clasifican las cargas y cómo se modela la utilidad de las mismas.

En [19] se busca maximizar la utilidad del hogar en su conjunto y no resuelve cómo gestionar las cargas por separado.

Por otro lado, en [18] se propone separar las cargas en cuatro tipos en función de cómo modelan la utilidad de las mismas y se dan ejemplos de funciones de utilidad y restricciones a considerar para los electrodomésticos más típicos de un hogar.

Finalmente, en [13] se propone clasificar los electrodomésticos en tres clases distintas. En la primer clase, se encuentran los electrodomésticos “elásticos”, los mismos tienen una función de utilidad asociada en función de cuánta energía se le suministra en cada instante. La segunda clase, son los electrodomésticos “semi-elásticos” que deben consumir una determinada cantidad de energía para cumplir con su tarea. En este caso, el algoritmo debe decidir en qué instantes debe encender los electrodomésticos dentro un intervalo de preferencia. Por último, se encuentran los electrodomésticos “inelásticos” que son aquellos cuyo consumo es inflexible y por lo tanto no son gestionadas por el algoritmo.

Tarifa de precios utilizada

Un aspecto importante para comparar los distintos algoritmos de gestión, es para qué tipo de tarifa están diseñados. Como se describió en la sección anterior existen distintas tarifas a nivel residencial. Dentro de las tarifas horarias, se destacan *DAP* en la cual los precios se publican con un día de antelación y *RTP* en la cual se publican con una hora de anticipación.

En [18] y [13] se considera un sistema tarifario *DAP*. Se plantea que al inicio de cada jornada la empresa suministradora intercambia información con los usuarios de manera de fijar el precio para las siguientes 24 horas. Además se considera que tampoco hay incertidumbre en la demanda del usuario, por lo que el problema de optimización se resuelve una única vez al inicio del día, y luego en cada hora se ejecutan las decisiones obtenidas.

En el caso de utilizarse una tarifa *RTP* no se conoce los precios futuros sino que en general se cuenta con una predicción de los mismos. Estas predicciones pueden ser proporcionadas por la empresa suministradora o puede ser calculadas por el sistema de gestión del hogar. En [14] se propone un método sencillo de predicción de precios futuros.

En [19] y en [17] se propone utilizar un concepto de ventanas deslizantes para una tarifa *RTP*. En este el problema de optimización se resuelve utilizando la información del precio y la

2.4. Algoritmo de gestión de cargas

demanda actual, los cuales se conocen con certeza, y una estimación de precios y demanda para una ventana de tiempo hacia adelante. El algoritmo de optimización se ejecuta periódicamente para poder actualizar la información. El resultado del proceso de optimización determina cuanta energía debe suministrarse a cada carga en cada intervalo de tiempo perteneciente a la ventana. Pero sólo se llevan a cabo los resultados de los intervalos de tiempo anteriores a cuando se recalcula el problema de optimización.

El hecho de que no se conozca con precisión los precios y la demanda futura lleva a que los resultados obtenidos por los algoritmos de optimización no sean óptimos. Esto fue considerado en [19] y [20], en los que se plantean algoritmos que son conscientes de esta incertidumbre. En [19] se propone un problema de optimización robusto en el cual se utiliza una predicción y un intervalo de confianza de los precios futuros, los cuales son calculados utilizando datos históricos. En cambio en [20] se propone un problema de optimización robusto basado en la simulación de múltiples escenarios simulados utilizando el método de Monte Carlo.

En [21] se plantea un algoritmo para una tarifa *RTP* que toma decisiones inspiradas en un diagrama de Markov. El algoritmo propuesto toma las decisiones que minimicen el costo futuro de la energía consumida, en función del valor esperado de los precios. Para esto se asume que se conoce el precio de la energía para el primer intervalo de tiempo y la función de distribución para los precios futuros.

Algoritmo Seleccionado

En el proceso de búsqueda y estudio de bibliografía se encontraron múltiples algoritmos de interés que ameritan ser estudiados con mayor profundidad. Sin embargo, se decidió implementar en este proyecto un solo algoritmo, para analizarlo con profundidad y evaluar los resultados que se obtienen con el mismo. Se seleccionó como base para el algoritmo de este proyecto el algoritmo planteado por Li, Chen y Low, en el trabajo "*Optimal demand response based on utility maximization in power networks*" [18].

La elección de este trabajo como base se debió a las siguientes razones. En primer lugar el algoritmo propuesto cumple con todas las características deseadas planteadas inicialmente. Además, si bien es un algoritmo genérico, permite utilizar funciones de utilidad y restricciones adaptadas a cada electrodoméstico. Otra de las razones para seleccionar este trabajo, es que el algoritmo es explicado de manera clara y detallada y se dan ejemplos de cómo deben ser las restricciones y funciones de utilidad para algunos electrodomésticos. Esto facilita la entendimiento del algoritmo y el trabajo con el mismo.

Posibles modificaciones

Tal como se explicó anteriormente el algoritmo seleccionado presenta muy buenas características, pero se entiende que el mismo puede ser mejorado integrando ideas de otros trabajos que fueron relevados durante la búsqueda bibliográfica.

Algunas modificaciones al algoritmo seleccionado que se creen de interés analizar son:

- **Cargas no interrumpibles.** Existen electrodomésticos, como lavarropas o lavavajillas, que una vez encendidos no es deseable que interrumpan su funcionamiento. Esto no fue tenido en consideración en el algoritmo seleccionado. En el trabajo [14] se proponen agregar restricciones al problema de optimización de manera de tener esto en consideración.
- **Consumo discreto de las cargas.** El algoritmo propuesto decide cuánta energía se debe suministrar a cada carga en cada intervalo de tiempo. Esta energía puede tomar cualquier valor entre un mínimo y un máximo de energía. Esto puede ser difícil de llevar a la práctica ya que existen muchos dispositivos en los que la única acción que se puede

Capítulo 2. Estado del arte y revisión bibliográfica de algoritmos de gestión

tomar sobre los mismo es encender o apagar, pero no es posible regular la potencia consumida. En el caso de cargas interrumpibles, esto se puede conseguir encendiendo el dispositivo sólo en una porción del intervalo de tiempo. Sin embargo, esta técnica no es conveniente en cargas no interrumpibles, por lo cual es necesario incluir en el planteo del problema que la energía consumida por estas cargas sólo puede tomar valores discretos. En [14] se propone una manera de considerar esto. Cabe destacar que esta modificación implica resolver un problema de optimización más complejo, dado que se pasa de tener una variable a optimizar en \mathcal{R}^N a tener una variable mixta con componentes en \mathcal{R} y en \mathcal{Z} .

- **Consumo no controlable** En el algoritmo seleccionado se considera que todas las cargas son controladas por el sistema de gestión. Existen diversas razones para que esto no sea así en la práctica. En primer lugar están las limitaciones económicas, dado que es muy costoso y complejo de instalar en un sistema de gestión que sea capaz de operar la totalidad de las cargas de un hogar. Adicionalmente, existen cargas que pueden comprometer severamente el confort del usuario si no se adaptan completamente a sus deseos, por ejemplo cargas de iluminación o de entretenimiento. Por consiguiente, se entiende que es conveniente que éstas sean operadas únicamente por el usuario. Por esta razón, se considera que existirá un ‘consumo de fondo’, que no va a poder ser controlado por el algoritmo de gestión. En [13] se considera un tipo de cargas inelásticas cuyo consumo es inflexible, las mismas son tenidas en cuenta en el proceso de optimización para computar el costo de la energía y para poder cumplir con la restricción de potencia máxima que puede consumir el hogar.
- **Flexibilidad para el usuario** El algoritmo planteado considera que al inicio de la jornada se conoce cómo es la demanda de las próximas 24 horas. Esto implica que el ‘consumo de fondo’ está determinado y que el usuario no puede cambiar sus preferencias a lo largo del día. Para corregir esto podría utilizarse el concepto de ventanas deslizantes propuesto en [19] y [17] y que el proceso de optimización se actualice periódicamente o cada vez que el usuario cambia sus preferencias de consumo.
- **Adaptación a tarifa RTP** El algoritmo propuesto está pensado para una tarifa ‘*day-ahead*’. Podría adaptarse a una tarifa RTP utilizando la idea propuesta de ventanas deslizantes y considerando que se cuenta con una predicción de los precios futuros. Adicionalmente, se podría considerar la manera de hacer ‘robusto’ al algoritmo frente a la incertidumbre de precios, o sea hacer que el mismo sea consciente de que la predicción de precios futuros no es exacta, y que tome las decisiones en función del valor esperado y la desviación del mismo.

No fue posible contemplar dentro del alcance de este proyecto todas las modificaciones propuestas en esta sección. El estudio detallado e implementación de las mismas queda para trabajos posteriores en el tema.

Capítulo 3

Descripción del sistema HEMS

En los capítulos anteriores se repasaron los conceptos más importantes de *Smart Grid* y Respuesta a la Demanda. Se destacó la importancia de que exista un sistema HEMS para que las políticas de Respuesta a la Demanda puedan ser adoptadas por un mayor número de clientes residenciales. Adicionalmente, se repasaron los posibles algoritmos de gestión a utilizar en el HEMS y se justificó la elección del algoritmo base planteado en [18].

El objetivo de este capítulo es describir cómo es el sistema HEMS desarrollado en este proyecto. En una primera instancia, se describe el escenario en el cual se enmarca el HEMS y sus actores. Luego se describe la arquitectura del HEMS, sus elementos y la plataforma utilizada. Finalmente se recuerda cuáles son los elementos del sistema HEMS en los que se trabajó en este proyecto y cuáles fueron tomados del proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1].

3.1. Descripción del escenario del problema y sus actores

Como se mencionó en Capítulo 1 y Capítulo 2 este proyecto se encuentra enmarcado dentro de una política de Respuesta a la Demanda a nivel residencial, donde la empresa suministradora motiva a los clientes a moldear su consumo energética utilizando un tarifa dinámica tipo *DAP*. En la Figura 3.1 se muestra un esquema del escenario en que se enmarca el proyecto y sus principales actores.

En primer lugar se encuentra la empresa suministradora que es la encargada de proveer y comercializar el servicio de energía eléctrica. Por ejemplo, en Uruguay esta empresa es UTE.

Por otra parte, se encuentran los cliente residenciales quienes compran la energía a la empresa suministradora. Cada cliente presenta un sistema de gestión (HEMS) que asiste al mismo a moldear su demanda de energía en función del precio y sus preferencias de consumo. En el presente proyecto se consideró que cada cliente es una célula independiente que se comunica únicamente con la empresa suministradora. Es decir, los clientes no interactúan entre sí.

El precio de la energía está regido por una tarifa de precios tipo *DAP* de acuerdo al algoritmo de gestión seleccionado (ver Sección 2.4). Para este tipo de tarifa el precio de la energía varía de hora en hora y antes del inicio de cada jornada se publican los precios para las 24 horas del día. En este proyecto se asume que los precios son fijados por un agente externo, por ejemplo, la empresa suministradora, y los clientes residenciales no influyen en la toma de decisión. Actualmente no se dispone de datos de precios reales adaptados a la realidad uruguaya ya que este tipo de tarifa aún no se encuentra en funcionamiento en Uruguay.

Capítulo 3. Descripción del sistema HEMS

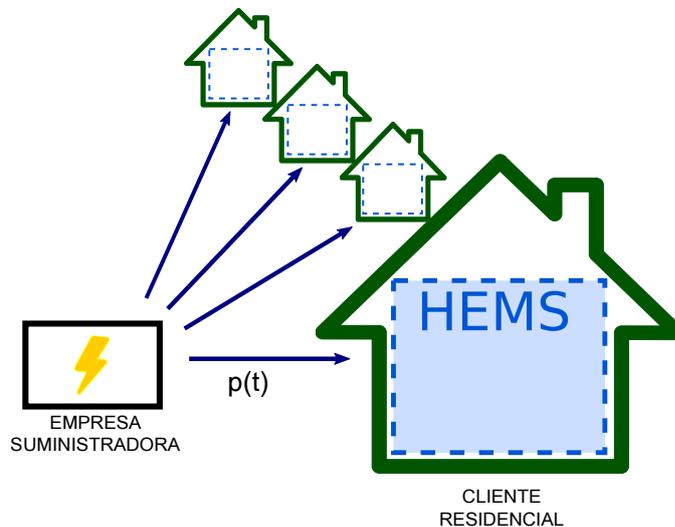


Figura 3.1: Esquema de la política de Respuesta a la Demanda en la que se enmarca el proyecto y sus actores

Existe un canal de comunicación entre el cliente y la empresa suministradora, a través del cual se comunica la curva de precios. En Uruguay, dado el alto nivel de acceso a internet, y considerando que este acceso es proporcionado por un ente estatal, resulta razonable considerar que la información sobre la tarifa será brindada al usuario a través de Internet.

El objetivo de este proyecto es diseñar e implementar un prototipo del sistema HEMS que asista al cliente residencial a moldear su demanda. Este debe controlar un conjunto de electrodomésticos del hogar y tomar las decisiones en función del precio de la energía y las consignas de confort determinadas por el usuario.

Proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente”

En el proyecto de fin de carrera “Controlador de energía domiciliario para una Red Eléctrica Inteligente”, Belcredi, Modernell y Sosa fue desarrollada una plataforma abierta para la implementación de un HEMS. Se trata de una plataforma abierta y versátil, capaz de facilitar la implementación de algoritmos de gestión. Asimismo, fue generada una API (Application Programming Interface), que permite la implementación y pruebas de algoritmos de optimización de consumo desarrollados por terceros[1]. El presente trabajo hace uso de dicha plataforma para la implementación del HEMS poniendo especial énfasis en el nodo central de la plataforma a modo de complementar el trabajo realizado por Belcredi, Modernell y Sosa.

3.2. Arquitectura del HEMS

Dado que se utilizó la plataforma del proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” se optó por usar la misma arquitectura de tipo centralizada planteada en ese trabajo. En la Figura 3.2 se presenta un diagrama de esta arquitectura y sus principales componentes.

El módulo central, llamado **Nodo Controlador**, es donde radica la inteligencia del sistema. Éste es el encargado de reunir los datos de los electrodomésticos, las preferencias del usuario y

3.2. Arquitectura del HEMS

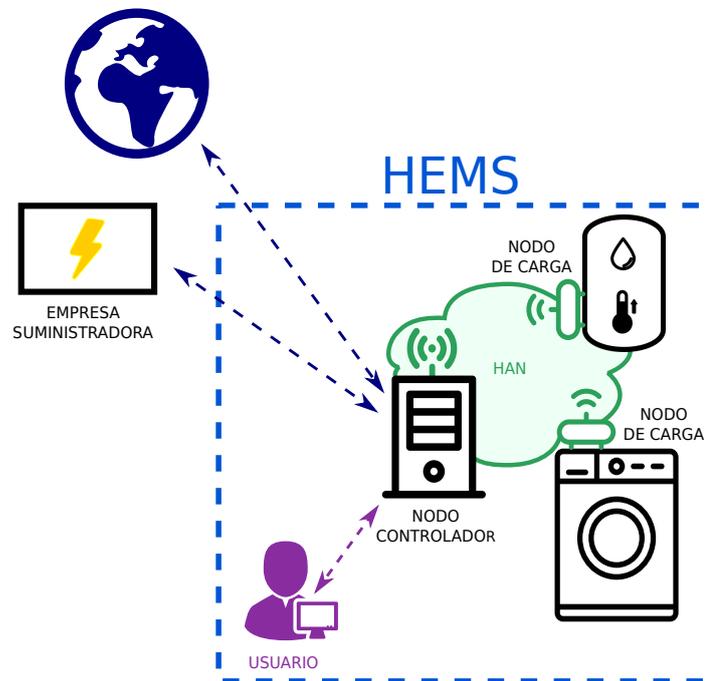


Figura 3.2: Esquema de sistema HEMS y sus componentes

la curva de precios. Con esa información decide cómo deben gestionarse los electrodomésticos y envía las órdenes correspondientes.

El sistema también cuenta con un conjunto de nodos periféricos que permiten medir magnitudes de interés y comandar los electrodomésticos, a estos nodos se les llamó **Nodos de Carga**. Adicionalmente, existe una **red HAN** (Home Area Network) que permite la comunicación entre el Nodo Controlador y los Nodos de Carga.

El Nodo Controlador posee una canal de comunicación con la empresa suministradora para obtener la curva de precios de cada día y acceso a Internet que le permite obtener información útil para gestionar los electrodomésticos, por ejemplo la predicción de la temperatura para el resto del día.

Finalmente, existe una **Interfaz de Usuario** en el Nodo Controlador que permite la comunicación con el usuario del sistema. A través de la misma el usuario puede configurar el sistema para que opere los electrodomésticos según su preferencia y obtener también información de interés.

El HEMS está compuesto por varios elementos y, como se describió en el Capítulo 1, en este proyecto no se trabajó en todos ellos. El alcance de este trabajo incluye la implementación del Nodo Controlador y la Interfaz de Usuario. El diseño y desarrollo de estos elementos será cubierto en capítulos posteriores de este documento. Por otro lado, los Nodos de Carga y la red HAN quedaron por fuera del alcance. Para estos se tomó lo realizado en “Controlador de energía domiciliario para una Red Eléctrica Inteligente” y se realizaron modificaciones menores para que pudieran funcionar correctamente con el Nodo Controlador desarrollado. En secciones posteriores en este capítulo se describe el funcionamiento de los Nodos de Carga y la red HAN.

3.3. Descripción del hardware utilizado

El hardware utilizado para el sistema HEMS en general coincide con el empleado por [1] en su plataforma, ver Figura 3.3. En esta sección se describen los principales componentes de hardware utilizados y sus características más relevantes.

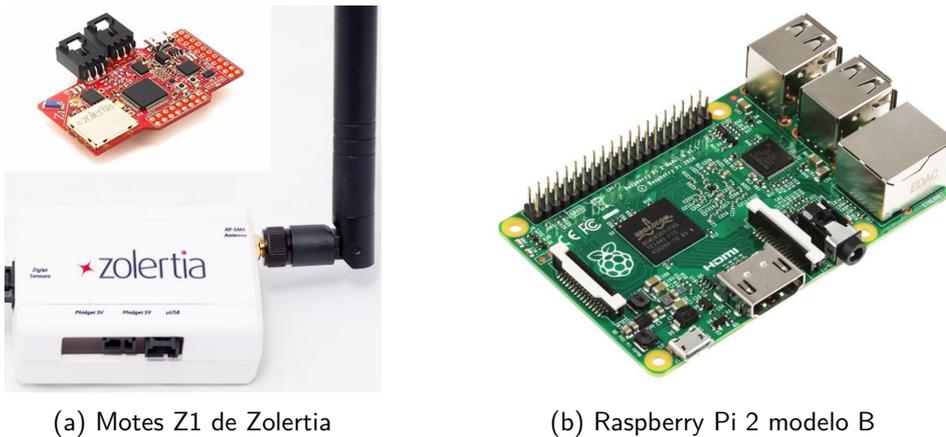


Figura 3.3: Componentes de hardware utilizados en el sistema HEMS

Nodos de Carga

En la plataforma del proyecto [1] los Nodos de Carga están compuestos por un mote Z1 de Zolertia y, de forma opcional, por sensores y/o actuadores. Para la programación de los motes Z1 se hizo uso de la plataforma de software *Contiki OS*. En el presente trabajo se utilizaron los mismo modelos de motes y la misma plataforma de software.

Las principales características del mote Z1 son las siguientes¹:

- Microprocesador MSP430F2617 con 92KB de ROM y 8KB de RAM.
- Radio CC2420 (IEEE 802.15.4 2.4 GHz).
- Puerto MicroUSB. Interfaz estándar de comunicación con PC para programación y debug.
- Dos posibles fuentes de alimentación:
 - Alimentación $3V_{DC}$ con fuente $3V_{DC}$ o 2 baterías AA.
 - Alimentación $5V_{DC}$ por puerto MicroUSB.

La principal ventaja de los motes Z1 es que los mismos soportan el sistema operativo *Contiki OS* y son compatibles con una gran variedad de sensores y actuadores. Estas dos características facilitaron la implementación de los Nodos de Carga.

¹Por más información acerca de este equipo consultar, la documentación del proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1], la hoja de datos del mote [24] y/o la guía de usuario de la familia del microprocesador MSP430F2617 [25].

Nodo Controlador

En la plataforma del proyecto [1] el Nodo Controlador está compuesto por una placa *Raspberry Pi* y un mote Z1 de *Zolertia*. La placa *Raspberry Pi* es el elemento más importante del Nodo Controlador y en ella radica toda su lógica. En este proyecto se utilizó una placa modelo *Raspberry Pi 2 Modelo B*[26] con sistema operativo *Raspbian*[27]. Las principales características de la placa *Raspberry Pi 2 Modelo B* son las siguientes:

- Chip Broadcom BCM2836 SoC (procesador Quad-core ARM Cortex-A7 de 900 MHz).
- 1 GB de memoria RAM .
- Alimentación $5V_{DC}$ por puerto MicroUSB.
- Puerto Ethernet 10/100 BaseT.
- Cuatro puertos USB 2.0.
- Tarjeta de memoria Micro SDIO.

El diseño de la plataforma y el software permiten que la *Raspberry Pi* pueda ser sustituida por una computadora personal convencional sin necesidad de hacer modificaciones.

Adicionalmente, el Nodo Controlador presenta un mote Z1 que oficia de *border router*. Este es un *gateway* a través del cual la placa *Raspberry Pi* puede acceder a la red *HAN* de los Nodos de Carga. Permite el ruteo desde una red estándar *TCP/IP* a la red con *HAN 6LowPAN*. Por más información acerca de por qué es necesario utilizar un *border router* y cómo funciona el mismo consultar [1].

3.4. Nodos de Carga

Los Nodos de Carga son los intermediarios entre el Nodo Controlador y los electrodomésticos. Su función es ejecutar órdenes y reportar mediciones al Nodo Controlador. Cada electrodoméstico controlado por el HEMS debe tener asociado al menos un Nodo de Carga capaz de comandarlo. En esta sección se describen las características más relevantes de los Nodos de Carga.

Si bien el desarrollo de los Nodos de Carga no se encontraba dentro del alcance de este proyecto, fue necesario implementar Nodos de Carga básicos para poder llevar a cabo la prueba experimental (ver Capítulo 7). Para la implementación de los mismos se tomó como base los nodos periféricos implementados en el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” y se los adaptó para los electrodomésticos de la prueba. Por más detalle acerca de cómo se implementaron los mismos consultar el Apéndice L.

Módulos de un Nodo de Carga

Un Nodo de Carga puede presentar distintos módulos en función de los electrodomésticos a los que esté asociado. Los módulos posibles son los siguientes:

- **Módulo de comunicación:** permite la comunicación del Nodo de Carga con el Nodo Controlador.
- **Módulo de comando:** permite comandar un electrodoméstico, ya sea encender/apagar, pausar/reanudar un proceso o cambiar el modo de operación del mismo.

Capítulo 3. Descripción del sistema HEMS

- **Módulo de medición:** permite medir magnitudes de interés, como lo son la temperatura de una habitación, la temperatura del agua dentro de un calefón o la potencia consumida por los electrodomésticos.



Figura 3.4: Módulos de un Nodo de Carga

Un Nodo de Carga debe presentar necesariamente un módulo de comunicación y al menos un módulo de comando o medición. Todos los electrodomésticos deben tener asociado al menos un Nodo de Carga con módulo de comando. Esto no implica que deba existir una relación uno a uno de los Nodos de Carga con los electrodomésticos. Un mismo electrodoméstico puede tener más de un Nodo de Carga asociado, por ejemplo uno de comando y otro de medición. A su vez un mismo Nodo de Carga puede comandar varios electrodomésticos.

3.5. Red HAN

La red HAN es la plataforma de comunicación entre el Nodo Controlador y los Nodos de Carga dentro del hogar. La misma debe ofrecer un mecanismo de comunicación bidireccional entre cada uno de los Nodos de Carga y el Nodo Controlador. En este trabajo se utilizó la implementación de la red HAN del proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1].

Stack de Protocolos

En los últimos años se han desarrollado múltiples protocolos y tecnologías de comunicación inalámbrica, que pueden ser utilizadas en aplicaciones de *Smart Grid*. Algunos ejemplos de estas tecnologías son: *Zigbee*, *6LoWPAN*, *Z-Wave*, *INSTEON* y *Wavenis*. En [1] se seleccionó *IPv6/6LoWPAN* sobre IEEE 802.15.4 como base del stack de comunicación, y *CoAP* como protocolo de aplicación, ver Figura 3.5.

Este stack de protocolos se encuentra implementado y es soportado por la plataforma de software *Contiki OS*. A continuación se describe el protocolo de aplicación *CoAP*. No se pretende en esta documentación describir el resto los protocolos presentes del stack ni justificar la elección de los mismos. No obstante, en el caso de querer conocer estos detalles es posible consultar la documentación del proyecto [1].

Protocolo CoAP (Constrained Application Protocol)

Constrained Application Protocol (CoAP) es un protocolo de capa de aplicación especialmente desarrollado para dispositivos y redes de recursos limitados [28]. Se trata de un protocolo similar a *HTTP* que agrega funcionalidades que hacen posible utilizarlo en sistemas de recursos limitados.

El protocolo *CoAP* utiliza un modelo cliente/servidor y se basa en la arquitectura *REST (Representational State Transfer)*. En esta arquitectura los servidores publican recursos, que

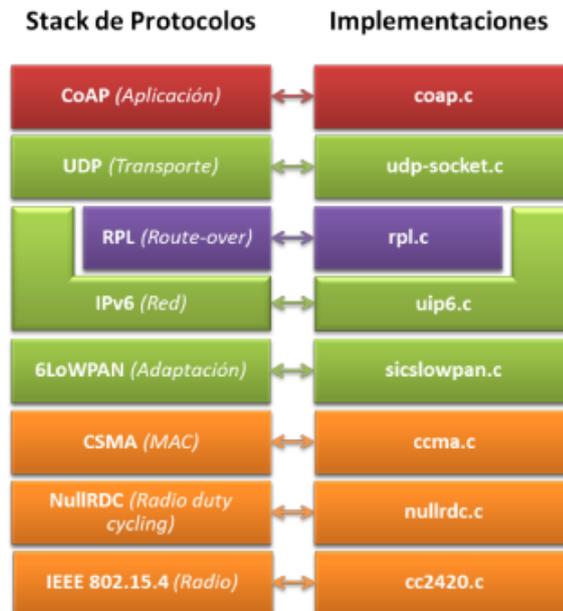


Figura 3.5: Stack de protocolo de red HAN, fuente [1].

son accedidos mediante un URI. Los clientes acceden a estos recursos a través de cuatro métodos: *PUT*, *POST*, *GET* y *DELETE*. En el sistema HEMS, un Nodo de Carga es un servidor *CoAP* y cada módulo de comando y medición está asociado a un recurso en el servidor. El Nodo Controlador es un cliente *CoAP* que hace uso de estos recursos para comandar los electrodomésticos y obtener medidas.

CoAP es lo suficientemente sencillo como para ser implementado desde cero para aplicaciones simples. Pero también existen implementaciones genéricas para múltiples plataformas. Algunos ejemplos son: *Erbium* en *Contiki OS*, *Californium* y *nCoAP* en *Java*, *txThings* y *aiocoap* en *Python*.

Implementación *Erbium* para *Contiki OS*

Los Nodos de Carga utilizan la plataforma de software *Contiki OS* y en [1] se seleccionó la implementación *Erbium* (*ER*) de *CoAP* para esta plataforma. Esta implementación viene incluida con *Contiki OS*. Un beneficio importante que presenta esta implementación es que pone a disposición una serie de códigos de ejemplo con las aplicaciones más típicas. En los Nodos de Carga se utilizó como base el ejemplo `er-example-server.c`, el cual implementa un servidor *CoAP* y contiene ejemplos de recursos. En el *Border Router* se utilizó el ejemplo `rpl-border-router.c` sin realizarle modificaciones.

Implementación *aiocoap* para *Python*

El Nodo Controlador se encuentra programado en *Python*. En [1] se optó por la implementación *aiocoap* de *CoAP* para este lenguaje. La misma utiliza la librería *asyncio* para el

Capítulo 3. Descripción del sistema HEMS

manejo de operaciones concurrentes².

En [1] se implementó una librería, denominada *wsncommunication*, para la comunicación del Nodo Controlador con los Nodos de Carga. Esta usa la librería *aiocoap* y ofrece las funciones básicas necesarias para obtener las mediciones desde los nodos y enviar comandos hacia ellos. Adicionalmente, ofrece funciones que permiten conocer cuáles son los nodos accesibles en la red HAN, observar su estado y definir sus atributos. En el Apéndice G se explica con mayor detalle las funciones presentes en el módulo *wsncommunication* y cómo fueron utilizadas en este trabajo.

²Por más información acerca de cómo se maneja la concurrencia en Python y de la librería *asyncio* consultar el Apéndice F

Capítulo 4

Algoritmo de gestión

En el capítulo anterior se describió el sistema HEMS y cuáles son los elementos sobre los que se trabajó en este proyecto. El algoritmo de gestión de cargas presente en el Nodo Controlado, gobierna el comportamiento del sistema HEMS y es el encargado de decidir cómo se deben comandar los electrodomésticos. Este algoritmo es un aspecto muy importante en este trabajo, dado que el éxito del sistema diseñado depende en gran medida de qué tan efectivo es el algoritmo de gestión.

En el presente capítulo se describe en detalle el algoritmo gestión implementado, incluyendo la formulación matemática del mismo, cómo se modelaron los electrodomésticos contemplados por el algoritmo y una primera evaluación de su funcionamiento.

4.1. Descripción del algoritmo de gestión

Tal como fue descrito en la Sección 2.4, el algoritmo de gestión que se utilizó en el Nodo Controlador del sistema está basado en la solución propuesta por Li, Chen y Low, en el trabajo “Optimal demand response based on utility maximization in power networks” [18].

Este proyecto se limitó a abordar un problema a nivel de hogar, en el cual cada cliente decide su consumo en función de la curva de precios de la energía. En [18] los autores plantean un problema más general al abordado en este proyecto, en el cual la empresa suministradora y los clientes negocian con el fin de maximizar el beneficio global del sistema. Si bien esta es una idea interesante, el estudio de la misma se encuentra por fuera del alcance de este proyecto. En el Apéndice A se describe la solución que plantean los autores a este problema, así como otras ideas que quedaron por fuera del alcance del proyecto.

Introducción al problema y notación utilizada

El algoritmos de gestión implementado considera el escenario que se describe a continuación. El escenario está compuesto por un hogar con múltiples electrodomésticos y una empresa suministradora. El hogar, o cliente de energía, debe pagar en función de cuánta energía consume y cuándo consume dicha energía, de acuerdo a una tarifa de precios variable tipo DAP. Se asume que existe un canal de comunicación con la empresa suministradora a través del cual el cliente conoce el precio de la energía, ver Figura 4.1.

Se propone dividir el tiempo en *timeslots*, cada intervalo temporal se denota con la letra t con $t \in [0, N - 1]$ siendo N el horizonte de tiempo de optimización. Para fijar ideas se puede

Capítulo 4. Algoritmo de gestión

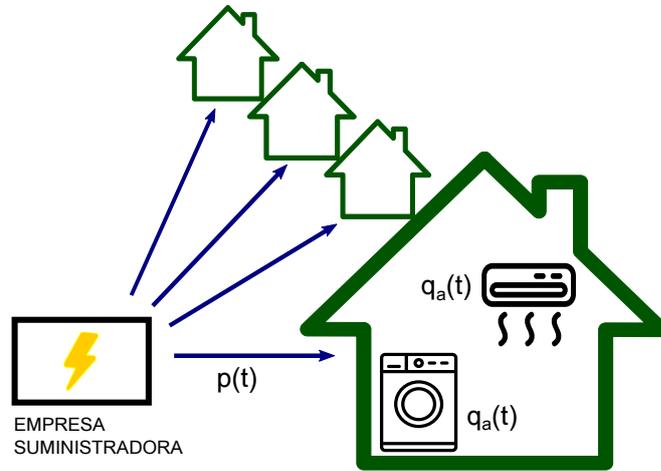


Figura 4.1: Diagrama del escenario del problema

considerar que cada *timeslot* t representa una hora y que el horizonte de tiempo es un día, es decir $N = 24$.

Al inicio de cada jornada la empresa suministradora determina el vector de precios \mathbf{p} del día. El mismo establece cuál es el precio de la energía en cada *timeslot* (i.e. $\mathbf{p} = [p[1], \dots, p[N - 1]]$). En este trabajo se consideró que el vector de precios presenta *timeslots* de una hora y que el horizonte temporal es de un día.

Sea \mathcal{A} el conjunto de electrodomésticos presentes en el hogar que son controlados por el sistema de gestión HEMS. De aquí en más se utilizará el sub-índice a para denotar el a -ésimo electrodoméstico del hogar ($a \in \mathcal{A}$). Cada electrodoméstico tiene un vector \mathbf{q}_a de largo N asociado, que representa cuánta energía se le suministra al electrodoméstico en cada *timeslot* t .

El objetivo del algoritmo de gestión es determinar cuánta energía se le debe entregar a cada electrodoméstico en cada *timeslot* de forma de maximizar el beneficio del cliente. Es decir, debe hallar:

$$\begin{aligned} q_a[t] \quad \forall t \in [0, N - 1] \\ \forall a \in \mathcal{A} \end{aligned}$$

Medición de beneficio

Un aspecto importante del algoritmo de gestión es cómo se mide el beneficio del cliente. En este trabajo se adoptó el criterio utilizado en [18], en el cual se propone que el usuario desea minimizar sus costos de energía y a la vez maximizar la satisfacción que le dan los electrodomésticos. Para cuantificar la satisfacción, se debe asociar a cada electrodoméstico una función de utilidad $U_a(\mathbf{q}_a)$.

La función de utilidad de cada electrodoméstico debe cuantificar cuánta satisfacción recibe el usuario en función del vector de energía \mathbf{q}_a que se le suministra al electrodoméstico. Es preciso destacar que depende del patrón de energía suministrado, \mathbf{q}_a . Es decir, no depende sólo de cuánta energía se le suministra al electrodoméstico sino que también de cuándo se le da esa energía. A modo de ejemplo, de nada le sirve al usuario que el calentador tenga el agua

4.1. Descripción del algoritmo de gestión

caliente al mediodía si este deseaba bañarse a las 8 AM. Otro aspecto a considerar es que la utilidad sólo modela el confort del usuario y es independiente del precio de la energía.

Considerando lo anterior se propone que el beneficio recibido por el usuario de cada electrodoméstico esta dado por la Ecuación 4.1. Y el beneficio total del cliente es la suma del beneficio de todos los electrodomésticos.

$$B(\mathbf{q}_a, \mathbf{p}) = U_a(\mathbf{q}_a) - \sum_{t=0}^{N-1} p[t] q_a[t] \quad (4.1)$$

Problema de optimización

El problema que debe resolver el algoritmo de gestión puede plantearse formalmente como un problema de optimización, en el cual se desea hallar los vectores de energía \mathbf{q}_a que maximizan el beneficio del cliente.

$$\max_{\mathbf{q}_a} \sum_{a \in \mathcal{A}} \left(U_a(\mathbf{q}_a) - \sum_{t=0}^{N-1} p[t] q_a[t] \right) \quad (4.2)$$

Restricto a:

$$\begin{aligned} M^a \mathbf{q}_a &\leq \eta_a && \forall a \in \mathcal{A} && \text{(Restricciones aparatos)} \\ 0 \leq \sum_{a \in \mathcal{A}} q_a[t] &\leq Q^{max}[t] && \forall t \in [0, N-1] && \text{(Energía máxima del hogar)} \end{aligned}$$

El conjunto de soluciones posibles se encuentra restringido en primer lugar por un conjunto de restricciones $M^a \mathbf{q}_a \leq \eta_a$ asociadas a los electrodomésticos. En el caso de un calefón estas restricciones podrían ser la energía máxima que puede consumir en un intervalo t y el rango de temperaturas admisibles del agua dentro del tanque.

Por otra parte, la última restricción limita la energía que puede consumir la casa en su conjunto a fin de cuidar la restricción de potencia contratada del hogar. El funcionamiento de esta restricción será explicado con mayor detalle en la Sección 4.6 del presente capítulo.

Algoritmo de gestión

El algoritmo de gestión consiste en hallar antes del inicio de cada jornada el vector \mathbf{q}_a para cada uno de los electrodomésticos, en función de la curva de precios del día y los deseos del usuario. El algoritmo debe buscar a través de un método iterativo la solución del problema de optimización planteado en la Ecuación 4.2. El método iterativo debe detenerse cuando se alcanza un vector \mathbf{q}_a los suficientemente cerca de la solución óptima o luego de una cantidad límite de iteraciones.

Convexidad del problema

Se dice que un problema es de optimización convexa si se cumple que la función objetivo es una función convexa y el espacio de soluciones posibles delimitado por las restricciones es un conjunto convexo¹[36].

¹Por más información acerca de esta definición y de las características de los problemas de optimización convexa consultar el Apéndice E.

Capítulo 4. Algoritmo de gestión

Los problemas de optimización convexa presentan diversas características que facilitan su resolución. La primera es que si el espacio de soluciones posibles es no nulo entonces la solución existe. Otra característica importante de estos problemas es que todo mínimo local de la función objetivo es necesariamente un mínimo global. Esto permite asegurar la convergencia de los métodos iterativos, evitándose el inconveniente de la convergencia a un mínimo local. Otra ventaja que presentan los problemas convexos es que son una rama largamente estudiada. Si bien esta es una rama aún activa y no hay consenso sobre cuáles son los mejores algoritmos de resolución, ya existen métodos que son muy efectivos[36]. Adicionalmente, estos algoritmos se encuentran implementados en diversos lenguajes de programación y resulta sencillo acceder a bibliotecas que los incluyan. Por ejemplo, *CVX* en *MatLab* o *CVXOPT* en *Python*.

Con el fin de simplificar la resolución del problema y asegurar la convergencia del método iterativo se optó por forzar a que el problema de optimización sea convexo. Para ello se limitó a utilizar funciones de utilidad cóncavas y restricciones lineales en \mathbf{q}_a , es decir de la forma $M^a \mathbf{q}_a \leq \eta_a$. El estudio de los métodos de resolución de problema de optimización convexa no se encuentra dentro del alcance del proyecto. Por esta razón se utilizó la biblioteca *CVXOPT* en *Python* para la resolución del problema, el uso de esta biblioteca se describe con mayor detalle en la Sección 5.5.

Una vez planteado formalmente el problema de optimización que debe resolver el algoritmo de gestión. El problema se reduce a seleccionar qué electrodomésticos del hogar son controlados por el sistema y en hallar funciones de utilidad $U_a(\mathbf{q}_a)$ y restricciones $M^a \mathbf{q}_a \leq \eta_a$ adecuadas para cada uno de dichos electrodomésticos. Este es un problema no menor, que será abordado en las siguientes secciones de este capítulo.

4.2. Selección de electrodomésticos a controlar

En [18] los autores consideran que todos los electrodomésticos del hogar son controlados por el sistema de gestión. Sin embargo, se entiende que esto es muy difícil de llevar a la práctica. Por esta razón, se decidió separar los electrodomésticos en dos categorías: controlables y no controlables.

Tal como ya se mencionó en la Sección 2.4, existen diversas razones por las cuales se cree que no es conveniente controlar todos los electrodomésticos del hogar. En primer lugar, los electrodomésticos a controlar deben ser ‘inteligentes’ o bien contar con adaptaciones que permitan ser accionados por el Nodo Controlador. En consecuencia, la infraestructura necesaria para poder controlar todos los electrodomésticos sería muy grande y además muy costosa. Adicionalmente, existen determinados electrodomésticos que pueden comprometer severamente el confort del usuario si no son controlados correctamente. A modo de ejemplo, al usuario le resultaría muy molesto no poder acceder a una habitación porque la luz no está encendida. Tampoco es deseable que las cargas de tipo entretenimiento (TV, radio, etc.) sean programadas por el sistema de gestión. Por ejemplo si hay un partido de fútbol a las 20 hs, el usuario va a preferir mirarlo en directo más allá de que sea más económico mirarlo en diferido dos horas después.

Considerando estas razones, se adoptó la idea planteada en [13] donde se propone separar los electrodomésticos en controlables y no controlables. Por un lado, las cargas no controlables son aquellas operadas únicamente por el usuario. El sistema de gestión no controla cuando se encienden o apagan, pero si se debe tener en consideración su consumo de forma de asegurar no romper la restricción de energía máxima del hogar. En la Sección 4.6 se describe cómo se tomó esto en consideración.

Por el contrario, las cargas controlables son aquellas gestionadas por el sistema de gestión.

4.3. Modelo del aire acondicionado

Dentro de los electrodomésticos controlables se destacan dos tipos: los electrodomésticos de acondicionamiento térmico y los de consumo discreto.

Electrodomésticos de acondicionamiento térmico:

Dentro de esta categoría se encuentran el calefón, el aire acondicionado, la loza radiante, entre otros. Estos electrodomésticos son de gran interés porque representan un alto porcentaje del consumo residencial. Por ejemplo, se estima que en Uruguay un 37% de la factura eléctrica de una familia está destinada al calentamiento de agua[29].

Asimismo, estos dispositivos pueden almacenar energía gracias a la inercia térmica que presentan. Esta característica permite mover su consumo de energía a horas de menor precio, lo cual hace que estos electrodomésticos presenten un gran potencial en las técnicas de Respuesta a la Demanda. De acuerdo a [16] los mismos son los electrodomésticos más frecuentes en las propuestas de Respuesta a la Demanda a nivel residencial.

Electrodomésticos de consumo discreto:

Dentro de esta categoría se encuentran el lavarropas, el lavavajillas y el sistema de carga de la batería de un auto. Estos electrodomésticos ejecutan una tarea y deben funcionar por un determinado lapso de tiempo para cumplir la misma. En estos electrodomésticos, al usuario le interesa que la tarea finalice antes de una hora determinada, pero no le afecta el momento específico en el cual se activa el dispositivo. Ello permite programarlos para que funcionen en las horas de menor precio.

Electrodomésticos seleccionados

En la evaluación del algoritmo de gestión, ver Capítulo 8, se realizó una simulación del desempeño del algoritmo en distintos hogares tipo. Teniendo en cuenta cuáles son los electrodomésticos más comunes en los hogares uruguayos se seleccionaron los electrodomésticos comentados a continuación. Dentro del grupo dispositivos de acondicionamiento térmico se eligieron el calefón y el aire acondicionado y dentro de los electrodomésticos de consumo discreto el lavarropas, el lavavajillas, el lavasecarropas y el sistema de riego².

En las secciones posteriores se detallan las funciones de utilidad y restricciones de cada uno de estos electrodomésticos. Adicionalmente, el sistema de gestión requiere contar con modelos eléctricos y térmicos de los electrodomésticos. En este capítulo se muestran los modelos utilizados pero no se describen en profundidad. Por más información acerca de estos modelos y los parámetros utilizados consultar Apéndice B.

4.3. Modelo del aire acondicionado

Los sistemas de aire acondicionado (o HVAC) son uno de los dispositivos de acondicionamiento térmico más utilizados a nivel mundial. En particular en Uruguay, más de un 25% de los hogares cuenta con un aire acondicionado de acuerdo a los últimos datos relevados por el Instituto Nacional de Estadística[30].

² Se observa que este dispositivo no es común de encontrar en los hogares uruguayos, sin embargo fue seleccionado por ser un electrodoméstico de consumo discreto con características distintas a la de un lavarropas o lavavajillas

Capítulo 4. Algoritmo de gestión

Modelo térmico

El modelo térmico de un aire acondicionado permite describir cómo evoluciona la temperatura en la habitación a lo largo del día, en función de cuánta energía entrega el aire acondicionado en cada *timeslot* ($q_a[t]$).

Se realizó una búsqueda bibliográfica de los posibles modelos térmicos para este dispositivo. Se encontró que el modelo dado por la Ecuación 4.3 era adecuado dadas las necesidades del sistema de gestión.

$$T_{hab}[t] = T_{hab}[t - 1] + \alpha (T_{ext}[t] - T_{hab}[t - 1]) + \beta q_{HVAC}[t] \quad (4.3)$$

donde:

$T_{hab}[t]$ es la temperatura de la habitación en el *timeslot* t .

$T_{ext}[t]$ es la temperatura exterior del hogar en el *timeslot* t .

$q_{HVAC}[t]$ es la energía entregada al aire acondicionado en el *timeslot* t .

α es un parámetro asociado a la disipación térmica con el exterior.

β es un parámetro asociado a la capacidad térmica de la habitación.

Se observa que $q_{HVAC}[t]$ puede tomar valores positivos o negativos dependiendo del modo de funcionamiento del aire acondicionado. Si el equipo se encuentra en modo calefacción $q_{HVAC}[t]$ es positivo, y es negativo en modo refrigeración.

En el Sección B.1 se describe con mayor detalle la búsqueda realizada, el modelo seleccionado y cómo se obtuvieron valores adecuados para los parámetros α y β .

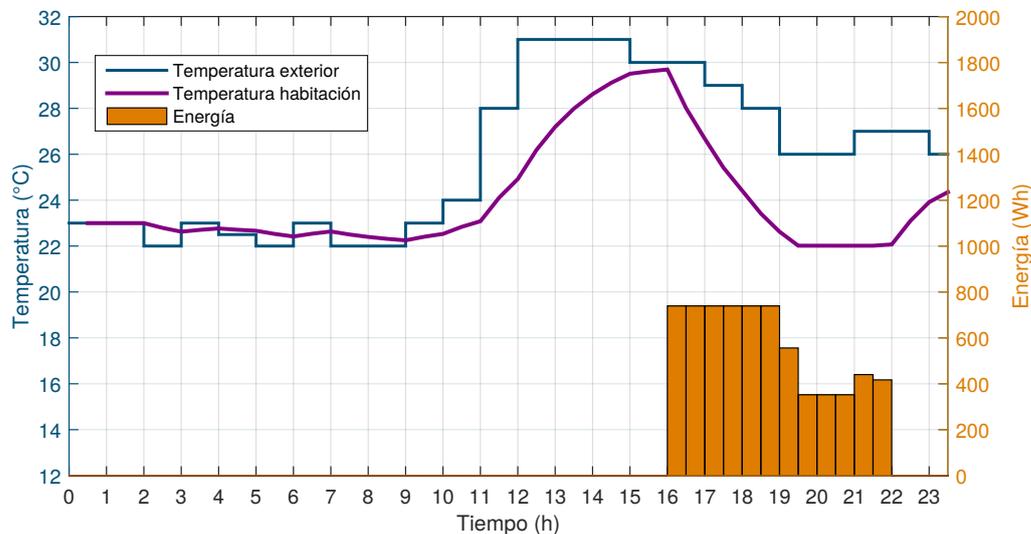


Figura 4.2: Ejemplo: evolución de la temperatura en un día de verano

En la Figura 4.2 se puede ver el modelo térmico en funcionamiento. En la misma, se observa como la temperatura en la habitación, en violeta, sigue a la temperatura exterior. Salvo cuando se entrega energía al aire acondicionado, representada con barras anaranjadas, donde ambas curvas se separan.

Observación:

La temperatura de una habitación puede presentar grandes variaciones dentro de una hora,

4.3. Modelo del aire acondicionado

esto se debe a que la inercia térmica es pequeña. Para poder modelar más correctamente la evolución de la temperatura y no perder estas variaciones, se optó por utilizar *timeslots* más pequeños en este electrodoméstico (20 o 30 minutos). En la Sección 5.3 se describe cómo se implementó la posibilidad de que cada electrodoméstico presente una resolución de *timeslot* distinta.

Función de utilidad

La variable de interés en este dispositivo es la temperatura de la habitación en la que se encuentra instalado el equipo. El usuario desea que esta temperatura sea lo más cercana posible a su temperatura de confort³.

Considerando esto se propuso utilizar la función de utilidad de la Ecuación 4.4, la cual mide que tan cerca está la temperatura en la habitación de la temperatura de confort.

$$U(\mathbf{q}_a) = b - a \sum_{t \in \mathcal{T}_a} \left(T_{hab}[t] - T_{hab}^{conf}[t] \right)^2 \quad (4.4)$$

donde:

$T_{hab}[t]$ es la temperatura de la habitación en el *timeslot* t .

$T_{hab}^{conf}[t]$ es la temperatura de confort en el *timeslot* t .

\mathcal{T}_a es el período de interés del usuario, es decir los *timeslots* en los cuales el usuario desea aclimatar la habitación.

a, b son parámetros de ajuste de la función de utilidad.

Tanto la temperatura de confort como el período de interés son configurables por el usuario a través de la interfaz web, por más información acerca de este punto se puede consultar el Capítulo 6.

La Figura 4.3 muestra una gráfica de la función de utilidad en función de $T_{hab}[t]$. Se observa que la misma presenta las siguientes características:

- Es una función cóncava.
- La utilidad es máxima cuando $T_{hab}[t] = T_{hab}^{conf}[t] \quad \forall t \in \mathcal{T}_a$.
- La utilidad decrece a medida que $T_{hab}[t]$ se aleja de $T_{hab}^{conf}[t]$ por ambos lados. Lo cual es consistente con que al usuario le molesta tanto que la habitación esté demasiado fría como que esté demasiado caliente.

En la Sección 4.1 se estableció que la utilidad debe ser una función del vector de energía \mathbf{q}_a . Para ello es necesario combinar función de utilidad (Ecuación 4.4) con el modelo térmico (Ecuación 4.3). Se observa que la función resultante es también cóncava en \mathbf{q}_a .

Restricciones

Las restricciones que se fijaron para este electrodoméstico son las siguientes:

³Los equipos de aire acondicionado son utilizados para climatizar ambientes. Esto quiere decir que no sólo controlan la temperatura de la habitación sino que también pueden controlar la humedad y ventilación de la misma. Por simplicidad, en este trabajo sólo se consideraron las funciones de control de temperatura, es decir refrigeración y calefacción.

Capítulo 4. Algoritmo de gestión

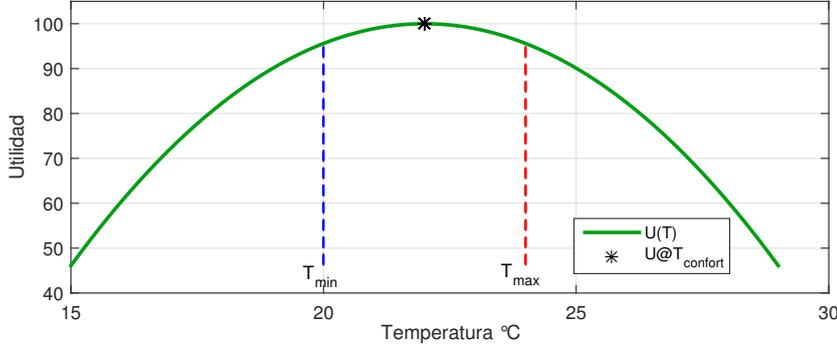


Figura 4.3: Función de utilidad del aire acondicionado

- El usuario puede establecer una temperatura límite de confort. En el caso de que el aire acondicionado se encuentre en modo calefacción puede fijar un temperatura mínima de confort, o temperatura máxima en caso contrario.

$$\text{Modo calefacción:} \quad T_{min}^{conf}[t] \leq T_{hab}[t] \quad \forall t \in \mathcal{T}_A$$

$$\text{Modo refrigeración:} \quad T_{hab}[t] \leq T_{max}^{conf}[t] \quad \forall t \in \mathcal{T}_A$$

- El aire acondicionado no puede consumir más de su potencia nominal.

$$q_a[t] \leq q_{max} = P_N \Delta t \quad \forall t \in [0, N - 1]$$

Por simplicidad se asumió que el consumo eléctrico del equipo de aire acondicionado cuando está encendido es constante e igual a su potencia nominal la cual está dada por la siguiente ecuación:⁴

$$P[t] \simeq P_N = \frac{\text{capacidad calefacción/refrigeración}}{\text{eficiencia}(\eta)}$$

- El consumo del aire acondicionado es siempre positivo, es decir este no genera energía eléctrica.

$$q_a[t] \geq q_{min} = 0 \quad \forall t \in [0, N - 1]$$

Ejemplo de resultado del algoritmo

En la Figura 4.4 se muestra un ejemplo del resultado del algoritmo para el aire acondicionado en un día de verano⁵. En este caso el usuario desea que su casa esté aclimatada desde las 18hs hasta las 21hs. Se configuró una $T_{confort} = 22^\circ\text{C}$ y $T_{max}^{conf} = 24^\circ\text{C}$ para todo el período de interés.

El algoritmo decide encender el aire acondicionado un par de horas antes del período de interés indicado por el usuario, para poder cumplir con la restricción de temperatura máxima.

⁴Por más información acerca de cómo se modelo el consumo eléctrico del aire acondicionado consultar el Sección B.1.

⁵En este ejemplo se utilizó un *timeslot* de 30 minutos para el aire acondicionado, es decir $\text{resolucionTS}=2$ (ver Sección 5.3)

4.4. Modelo del calefón

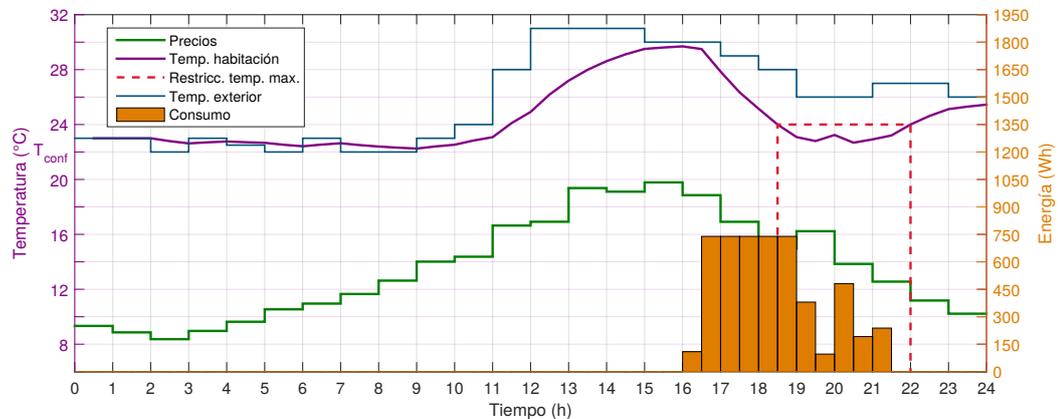


Figura 4.4: Ejemplo del resultado del algoritmo para el aire acondicionado para un día de verano

Luego continúa enfriando para mantener la temperatura dentro del rango deseado. A su vez, dentro del intervalo de interés, el algoritmo decide brindar menos energía entre las 19:00 y las 20:00. Esto se debe a que el precio de la energía es mayor en dicho intervalo, por lo cual se sacrifica confort para reducir los costos.

Se observa que en este caso el algoritmo no puede aprovechar las horas más baratas del día. Lo mismo ocurre ya que las horas más baratas se dan de madrugada y se encuentran muy lejos de las horas de interés para el usuario. Esto se debe a que la inercia térmica de la habitación no es lo suficientemente grande para poder sacar provecho de las horas baratas.

4.4. Modelo del calefón

El calefón es el dispositivo más utilizado en los hogares para el calentamiento de agua. Según encuestas realizadas en el 2013, cerca de una 82% de los hogares uruguayos cuentan con al menos un calefón[30]. De acuerdo a [29], se estima que en Uruguay un 37% del consumo de energía eléctrica a nivel residencial está destinada al calentamiento de agua.

La variable de interés en este dispositivo es la temperatura del agua dentro del tanque. El usuario desea contar con agua caliente en el momento que decide bañarse.

Modelo térmico

El modelo térmico para un calefón permite predecir la temperatura de agua dentro del tanque a lo largo del día, en función de la energía suministrada y la demanda de agua caliente del usuario.

Al igual que en el caso del aire acondicionado, se realizó una búsqueda bibliográfica acerca de las distintas propuestas existentes para modelar un calefón. A continuación se describe el modelo seleccionado para utilizar en el algoritmo de gestión. En la Sección B.2 se describe con mayor detalle este modelo y el resultado de la búsqueda.

El modelo utilizado para predecir la evolución de la temperatura del agua dentro del tanque está dado por la siguiente ecuación:

Capítulo 4. Algoritmo de gestión

$$\begin{aligned} T_W^{\delta_V=0}[t] &= T_W[t-1] + \alpha (T_{ext} - T_W[t-1]) + \beta q[t] && \text{(sin extracción)} \\ T_W[t] &= T_W^{\delta_V=0}[t] (1 - \delta_V[t]) + T_{inlet} \delta_V[t] && \text{(con extracción)} \end{aligned} \quad (4.5)$$

Donde:

- $T_W[t]$ es la temperatura del agua caliente en el *timeslot* t .
- T_{ext} es la temperatura de la habitación en la cual se encuentra el calefón.
- T_{inlet} es la temperatura del agua fría que ingresa al calefón.
- $\delta_V[t]$ es el volumen de agua caliente que extrae el usuario en el *timeslot* t .
- $q[t]$ es la energía que se le entrega al calefón en el *timeslot* t .
- α es un parámetro del modelo asociado a la disipación térmica con el ambiente.
- β es un parámetro del modelo asociado a la capacidad térmica del equipo.

La primera igualdad representa a la evolución de la temperatura del agua cuando no se realiza extracción de agua caliente. Se observa que esta ecuación presenta la misma forma que el modelo del aire acondicionado.

En el caso que exista extracción de agua caliente, el modelo asume que la temperatura del agua caliente evoluciona según la primer igualdad a lo largo del *timeslot*. Luego, al finalizar el mismo se quita instantáneamente el volumen de agua caliente consumida. De esta manera, la temperatura al finalizar el *timeslot* puede ser obtenida combinando ambas igualdades.

A continuación se encuentra el planteo con el cual se determina el modelado del calefón para el caso sin y con extracción de agua.

Modelo sin consumo de agua caliente:

En una primera instancia se consideró un modelo simplificado en el cual no se extrae agua del tanque. Para ello es usual utilizar un modelo de parámetros térmicos equivalentes (ETP - 'Equivalent Thermal Parameters')[31, 22].

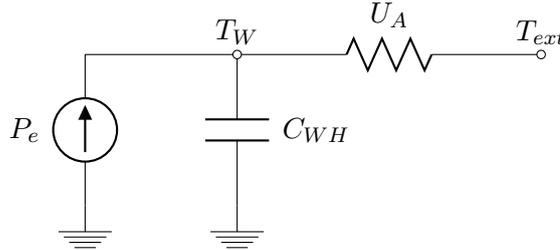


Figura 4.5: Circuito eléctrico equivalente

El circuito equivalente es idéntico al utilizado para el aire acondicionado, ver Figura 4.5. La única diferencia en este caso es que P_e es siempre mayor o igual a cero. Mediante un razonamiento análogo al realizado para el aire acondicionado es posible obtener la expresión de la Ecuación 4.6.

$$\begin{aligned} T_W[t+1] &= T_W[t] + \alpha (T_{ext}[t] - T_W[t]) + \beta q_e[t] \\ \text{donde :} \quad \alpha &= \frac{U_A \Delta t}{C_{WH}} \\ \beta &= \frac{1}{C_{WH}} \end{aligned} \quad (4.6)$$

4.4. Modelo del calefón

Modelo con consumo de agua caliente

El modelo de la Ecuación 4.6 no es completo dado que no considera la demanda de agua caliente del usuario. Por eso, se realizó una revisión bibliográfica acerca de cómo modelar la evolución de la temperatura del agua cuando existen extracciones.

Los autores en [22] proponen utilizar la ecuación de mezcla de dos líquidos para determinar cuál es la temperatura del agua resultante luego de una ducha. En cambio, en [31] se consideran dos modelos distintos, uno en el que se toma el calefón como una masa de agua homogénea cuya temperatura desciende a medida que se extrae agua del tanque. Y un segundo modelo que considera que existen dos masas de agua dentro del tanque una caliente y otra fría. Cuando el usuario extrae agua del tanque el volumen de la masa de agua fría crece.

El algoritmo de gestión debe asegurar que el usuario pueda bañarse confortablemente. Bajo la hipótesis de que el calefón se encuentra correctamente dimensionado, se puede asumir que si la temperatura del agua es adecuada en el instante inicial del baño, luego el usuario va a poder bañarse sin inconveniente. Por esta razón, no es de interés modelar con precisión cómo evoluciona la temperatura del agua mientras la persona se baña. Pero si se debe conocer cuál es la temperatura resultante luego del baño. Considerando lo anterior se optó por utilizar el modelo de propuesto en [22] dado que es el más sencillo de los tres modelos y permite calcular la correctamente la temperatura en régimen, que es cuando la temperatura se homogeneiza luego de la extracción.

Para que el modelo funcione correctamente es importante identificar adecuadamente los valores de parámetros α , β , $\delta_V[t]$ y T_{inlet} del electrodoméstico en el ambiente que va a ser utilizado. En el Sección B.2 se explica cómo se determinaron estos parámetros para el calefón utilizado en la prueba experimental.

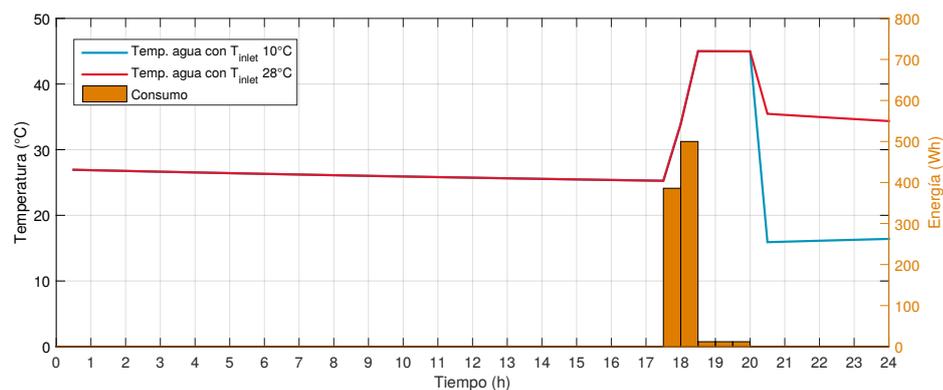


Figura 4.6: Ejemplo de la evolución de la temperatura del agua en el calefón para invierno y verano

En la Figura 4.6 se observa un ejemplo de la evolución de la temperatura del agua dentro del calefón de acuerdo al modelo seleccionado. Se puede apreciar que la temperatura luego de que el usuario se bañe, es decir a las 20hs, depende fuertemente de la temperatura del agua fría que ingresa al calefón. En la figura se puede ver el resultado para $T_{inlet} = 28^{\circ}\text{C}$ (verano) y 10°C (invierno).

Capítulo 4. Algoritmo de gestión

Función de utilidad

La consigna que debe cumplir un calefón para satisfacer al usuario, es que el agua en el interior del tanque esté caliente a la hora que el mismo desea bañarse. El usuario todos los días especifica cuáles son los rangos horarios dentro de los cuales desea que el agua esté lista para un baño, al cuál se le denomina \mathcal{T}_a .

La función de utilidad se usó para este dispositivo es la siguiente:

$$U(T_W[t], T_{conf}) = b - a \sum_{t \in \mathcal{T}_a} e^{-\frac{\gamma}{T_{conf}} T_W[t]}$$

Donde:

T_{conf} es la temperatura de confort del usuario.

\mathcal{T}_a es el período de interés del usuario, es decir los *timeslots* en los cuales el usuario desea bañarse.

a, b, γ son parámetros de ajuste de la función de utilidad.

Esta función presenta las siguientes características (ver Figura 4.7):

- La función es cóncava.
- Si $T_W[t] = T_{conf}$, el usuario se puede bañar correctamente por lo que la utilidad es grande.
- Si $T_W[t] < T_{conf}$, el usuario no se puede bañar cómodamente por lo que la utilidad decae rápidamente.
- Si $T_W[t] > T_{conf}$, el usuario se puede bañar correctamente pero no obtiene mayor satisfacción que en el caso $T_W[t] = T_{conf}$ por lo que la utilidad es similar.

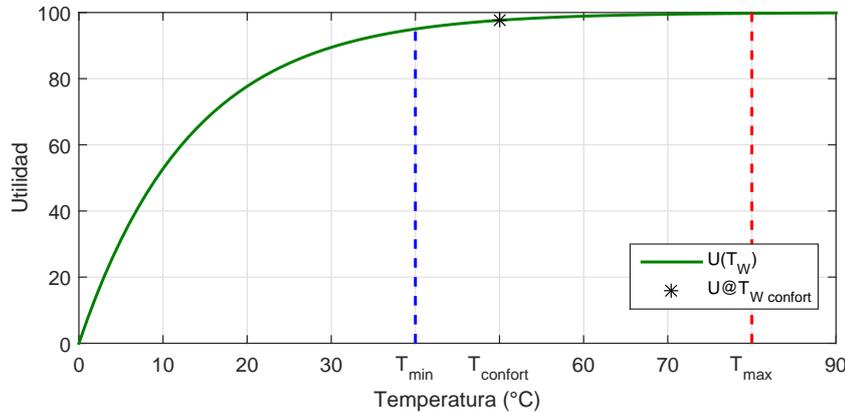


Figura 4.7: Función de utilidad y restricciones propuestas para el calefón

La utilidad debe ser expresada de forma explícita en función del vector de energía $q_a[t]$. Por tanto, es necesario combinar la función de utilidad propuesta con el modelo térmico detallado anteriormente.

Restricciones

Las restricciones que debe cumplir este electrodoméstico son las siguientes:

4.4. Modelo del calefón

- La temperatura del agua no debe superar en ningún momento la temperatura máxima a la cual es seguro que opere este electrodoméstico.

$$T_W(t) \leq T_{max} \quad \forall t \in [0, N - 1]$$

- La temperatura del agua dentro del intervalo de interés no puede ser inferior a una temperatura mínima de confort. La misma está asociada a la temperatura mínima con la cual el usuario encuentra 'aceptable' bañarse.

$$T_W(t) \geq T_{min}^{conf} \quad \forall t \in \mathcal{T}_a$$

- El calefón no puede consumir más energía en un *timeslot*, que la correspondiente a un funcionamiento a potencia nominal durante todo el *timeslot*.

$$q_a[t] \leq q_{max} = P_N \Delta t \quad \forall t \in [0, N - 1]$$

- El consumo del calefón es siempre positivo, es decir este no genera energía eléctrica.

$$q_a[t] \geq q_{min} = 0 \quad \forall t \in [0, N - 1]$$

Ejemplo de resultado del algoritmo

En la Figura 4.8 se muestra un ejemplo del resultado del algoritmo para el calefón⁶. El usuario desea bañarse entre las 18:30hs y las 20:00hs y configuró como temperatura de confort $T_{conf} = 55^\circ\text{C}$ y como temperatura mínima admisible $T_{min} = 40^\circ\text{C}$.

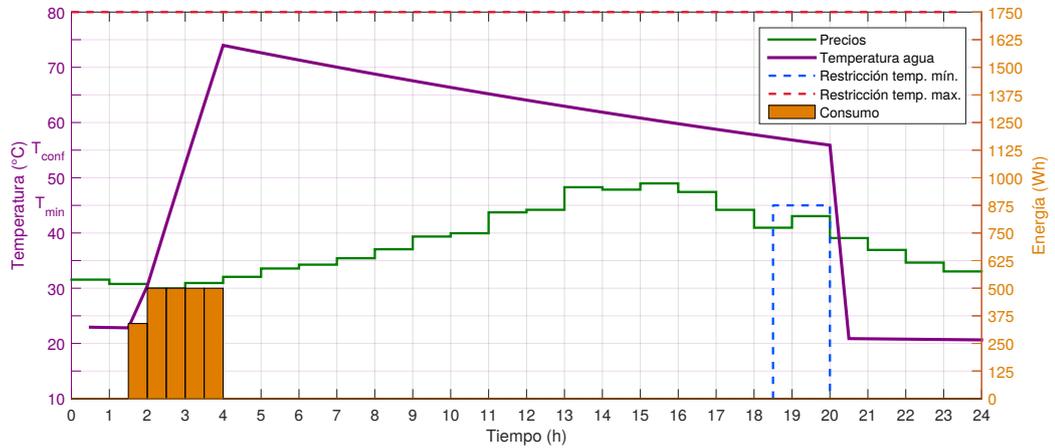


Figura 4.8: Ejemplo del resultado del algoritmo para el calefón

El algoritmo decide entregar energía al calefón entre las 1:30hs y 3:00hs, debido a que son las horas más baratas del día. En estas horas se eleva la temperatura del agua hasta 75°C para que luego en el período de interés la temperatura esté en el entorno de la temperatura de confort. A diferencia del aire acondicionado en este caso el sistema de gestión puede aprovechar las horas más baratas debido a que la inercia térmica del calefón es mucho mayor.

⁶En este ejemplo se utilizó un *timeslot* de 30 minutos para el calefón (ver Sección 5.3)

Capítulo 4. Algoritmo de gestión

Se destaca que con esta solución se ‘desperdicia’ parte de la energía que se entrega al calefón. Encender el calefón en las horas más cercanas al período de interés precisa sólo dos tercios de la energía que el algoritmo decide dar. Sin embargo, estas son horas que duplican en precio las horas de la madrugada. En conclusión, la solución obtenida por el algoritmo es la que maximiza el beneficio del usuario.

Observación:

El algoritmo desconoce el momento exacto en el que se baña el usuario, por lo que asume que este se baña en el final del período de interés (\mathcal{T}_a).

4.5. Modelo de electrodomésticos de consumo discreto

Los electrodomésticos de consumo discreto son aquellos que ejecutan una tarea específica, la cual el usuario desea que finalice antes de una hora determinada, pero no le afecta el momento concreto en el cual se activa el dispositivo.

Cada tarea que ejecuta el electrodomésticos presenta un tiempo de funcionamiento o una cantidad de energía asociados. Dentro de esta clase de electrodomésticos se distinguen dos tipos:

- **Electrodomésticos con tarea de duración fija.**

El tiempo de duración es fijo y por lo tanto la energía necesaria para realizar la tarea también lo es. El sistema de gestión puede decidir en qué horas suministrarle energía a estos electrodomésticos. Dentro de esta categoría se encuentran el lavarropas y el lavavajillas.

- **Electrodomésticos con tarea de duración flexible.**

Estos electrodomésticos pueden operar durante una cantidad de tiempo variable. El sistemas de gestión puede decidir el total de energía a suministrar y en qué horas hacerlo. El usuario puede fijar los valores mínimos y máximos de energía. Dentro de esta categoría se encuentra el bombeo de agua para regar un jardín, sistema de bombeo de agua para una piscina o la carga de la batería de un auto eléctrico.

Función de utilidad

En este tipo de electrodomésticos el usuario especifica diariamente si desea que el dispositivo opere o no. En caso afirmativo debe indicar el rango horario dentro del cual el dispositivo puede operar (\mathcal{T}_a).

La función de utilidad seleccionada para este tipo de dispositivos se encuentra en la Ecuación 4.7, ésta es recomendada para los electrodomésticos de consumo discreto en [18].

$$U(\mathbf{q}_a) = b + a \sum_{t \in \mathcal{T}_a} q_a[t] \quad (4.7)$$

La función de utilidad propuesta establece que la utilidad crece linealmente con el total de energía suministrada. Esto lleva a que si los precios en el día son bajos, el sistema de gestión va a decidir suministrar al electrodoméstico toda la energía posible. En cambio, si los precios del día son altos se va a tender a suministrar el mínimo de energía necesario. Se observa que esta función de utilidad sólo tiene sentido en los electrodomésticos con tarea de duración flexible, ya que en los de duración fija presenta un valor constante.

En el caso de los electrodomésticos con tarea de duración fija, el usuario desea simplemente que la tarea finalice a tiempo. La satisfacción del usuario no aumenta si se le entrega al electrodomésticos más energía que la asociada a la tarea. Por ejemplo, el usuario no reciba más

4.5. Modelo de electrodomésticos de consumo discreto

satisfacción si el algoritmo decide suministrar a un lavarropas la energía asociada a un lavado y medio. Es por esto que se tomó $a = 0$ en este tipo de electrodomésticos. En el Apéndice C se describe cómo se seleccionaron las constantes a y b para la función de utilidad de cada electrodoméstico.

Restricciones

Las restricciones que debe cumplir este tipo de electrodomésticos son las siguientes:

- El total de energía suministrada al dispositivo a lo largo del día debe estar dentro un rango configurado por el usuario.

$$\text{Tarea de duración variable:} \quad Q^{min} \leq \sum_{t \in \mathcal{T}_a} q_a[t] \leq Q^{max}$$

$$\text{Tarea de duración fija:} \quad Q_{tarea} \leq \sum_{t \in \mathcal{T}_a} q_a[t]$$

En el caso de los electrodomésticos con tarea de largo fijo, sólo se establece el valor mínimo de energía. Como la función de utilidad no crece con la cantidad de energía, porque $a = 0$, el algoritmo siempre decide suministrar el mínimo de energía.

- El dispositivo no puede consumir más energía en un *timeslot*, que la correspondiente a un funcionamiento a potencia promedio durante todo el *timeslot*.

$$q_a[t] \leq q_{max} = P_{prom} \Delta t \quad \forall t \in [0, N - 1]$$

Se optó por limitar la energía que se le pueda dar al electrodoméstico en cada *timeslot* de acuerdo a la potencia promedio que el mismo consume. De esta manera se asegura que el algoritmo brinde la cantidad de energía necesaria para completar la tarea.

Se observa que no sería correcto considerar el máximo de la potencia que consume el dispositivo, es decir $q_{max} = P_{max} \Delta t$. Dado que de esta manera el algoritmo podría agendar toda la energía en un *timeslot*, cuando el electrodoméstico requiere de más de un *timeslot* para realizar la tarea.

- El dispositivo sólo puede consumir energía.

$$q_a[t] \geq q_{min} = 0 \quad \forall t \in [1, T]$$

Ejemplo de resultado del algoritmo

En la Figura 4.9 se muestra un ejemplo del resultado del algoritmo para un lavavajillas. Las características más relevantes del mismo se encuentran en la Tabla 4.1. El usuario cargó el lavavajillas el día anterior y desea que los platos estén limpios antes de las 17hs del día siguiente.

Tal como es de esperar, el algoritmo decide darle energía al lavavajillas en las horas más baratas del período de interés. Se puede apreciar que la limitación de energía máxima que se le puede dar al electrodoméstico por *timeslot*, no permite entregarle toda la energía en la hora más barata. Esto es consistente con que la duración de la tarea es de 102 minutos y por lo tanto es necesario entregar energía en dos horas distintas.

Capítulo 4. Algoritmo de gestión

Electrodomésticos	Tipo	Duración [min.]	Energía [Wh]	P_{prom} [W]	P_{max} [W]
Lavarropas ^a	T. fijo	35	254	444	794
Lavavajillas ^b	T. fijo	102	733	434	1240
Lavasecarropas ^c	T. fijo	147	1949	795	2500
Bomba Riego	T. variable	[15 60]	[325 1300]	1300	1300

^aLavarropas General Electric modelo “WSM 2420D3 WW “

^bLavavajillas Kenmore modelo “ 665.13242K900 “

^cDatos correspondientes al lavasecarropas utilizado en la prueba experimental. En el Apéndice B se describe cómo se hallaron estos valores.

Tabla 4.1: Valores de ejemplo de las variables para algunos electrodomésticos de consumo discreto.

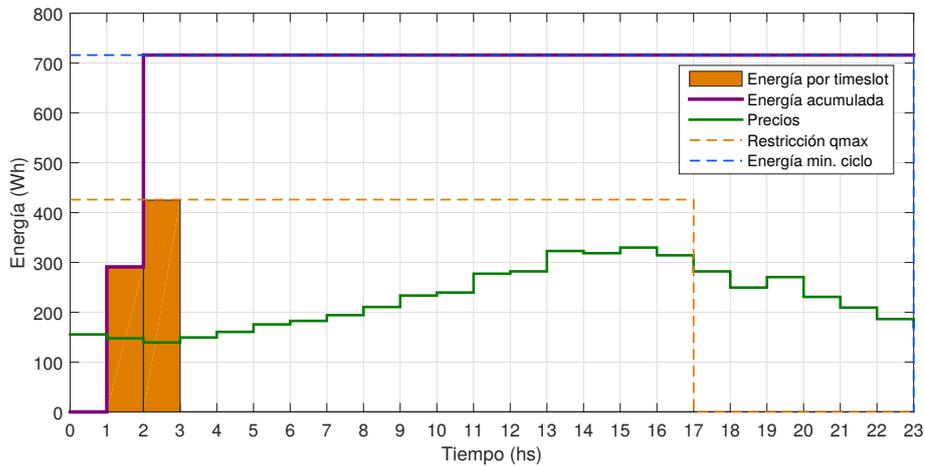


Figura 4.9: Ejemplo del resultado del algoritmo para un lavavajillas

4.6. Restricción del hogar

En los ejemplos mostrados en las secciones anteriores se observó que el algoritmo de gestión tiende a agendar los electrodomésticos en las horas más baratas del día. Hecho que ocurre especialmente con los electrodomésticos más flexibles como el calefón y los electrodomésticos de consumo discreto. Si bien esto permite reducir el costo en energía, se debe tener cuidado de no romper la restricción de potencia contratada del hogar. A continuación se describe cómo este límite es tenido en consideración en el algoritmo de gestión.

Potencia contratada

Generalmente los contratos que ofrecen las empresas suministradoras a los clientes residenciales establecen un límite de la potencia, conocido como potencia contratada. Este límite establece la potencia máxima que el cliente puede consumir, en caso que se supere la empresa corta el suministro de energía del hogar. Este límite comúnmente se controla a través de un

4.6. Restricción del hogar

Interruptor de Control de Potencia colocado a la entrada del hogar⁷. Si dicho límite es superado se acciona el interruptor y se corta el suministro eléctrico. Luego, el usuario debe accionar manualmente el interruptor para restaurar el servicio.

Si el sistema de gestión no tiene en cuenta la restricción de potencia contratada del hogar es probable que esta se rompa, especialmente en las horas más baratas del día. Esto tiene un impacto muy negativo en la calidad del servicio brindado, dado que el usuario deberá restaurar el servicio manualmente. En consecuencia, es importante que el Nodo Controlador asegure que sus decisiones respeten las restricciones del hogar.

Restricción de energía máxima

El problema de optimización que debe resolver el sistema de gestión (Ecuación 4.2) presenta la siguiente restricción asociada al hogar.

$$0 \leq \sum_{a \in \mathcal{A}} q_a[t] \leq Q^{max}[t] \quad \forall t \in [0, N - 1] \quad (4.8)$$

Esta restricción establece que el conjunto de electrodomésticos controlados por el sistema de gestión no puede superar el límite establecido por $Q^{max}[t]$ en cada *timeslot*. La elección de los valores $Q^{max}[t]$ debe tener en consideración cuál es la potencia contratada del hogar y la potencia que consumen los electrodomésticos que no son controlados por el sistema de gestión.

Es importante destacar que esta es una restricción sobre la energía que pueden consumir los electrodomésticos en un *timeslot* y no sobre la potencia instantánea. El algoritmo de gestión determina cuánta energía darle a cada electrodoméstico en cada *timeslot*. Sin embargo, este no determina cómo se distribuye esta energía dentro del *timeslot* esta es la responsabilidad del módulo Intérprete, el cual se describirá más adelante en la Sección 5.6.

Consumo de fondo

Como ya se mencionó existen electrodomésticos que no es conveniente sean controlados por el sistema de gestión, sino que deben ser controlados directamente por el usuario. Esto puede ser porque pueden comprometer severamente el confort del usuario si no son controlados correctamente o porque el beneficio económico que ofrece gestionarlo no justifica la inversión en un nodo de carga que lo opere.

Los electrodomésticos no controlables están por fuera de sistema de gestión, y este no controla cuándo se encienden o apagan, pero sí debe tener en consideración su consumo de forma de garantizar que no se viola la restricción de potencia contratada del hogar. No es necesario conocer en detalle cuánto consume cada electrodoméstico no controlable, sino que basta con disponer de una predicción de cómo es el consumo agregado de todos esos electrodomésticos a lo largo del día. A este consumo agregado se le llamó consumo de fondo del hogar.

En un hogar existen electrodomésticos que operan durante todas las horas del día, como lo son la heladera, un radio reloj o los electrodomésticos en ‘stand-by’. El consumo de estos electrodomésticos es relativamente estable y es independiente de si hay personas en el hogar o no. Por otra parte, existen electrodomésticos cuyo consumo depende fuertemente de la presencia de personas en el hogar y de sus hábitos. Este es el caso de una jarra eléctrica que el usuario utiliza para calentar el agua del mate.

Para obtener una predicción del consumo de fondo del hogar, el sistema de gestión podría medir el consumo total del hogar y restarle la potencia que consumen los electrodomésticos

⁷En Uruguay UTE utiliza Interruptores Termomagnéticos con curva C para este fin.

Capítulo 4. Algoritmo de gestión

que él controla. De esta manera mediría el consumo del fondo del hogar en tiempo real. En base a dichas mediciones, el sistema podría aprender los hábitos de consumo del usuario y armar predicciones del consumo de fondo del hogar para cada día. Si bien esta es una solución interesante, la implementación de la misma es compleja. Por consiguiente, quedó fuera del alcance de este proyecto y se optó por implementar una solución más sencilla.

La solución implementada en este proyecto requiere que el usuario especifique todos los días las horas en la cuales no hay consumo activo en el hogar. Es decir, las horas durante las que todos los integrantes de la familia están durmiendo o se encuentran fuera.

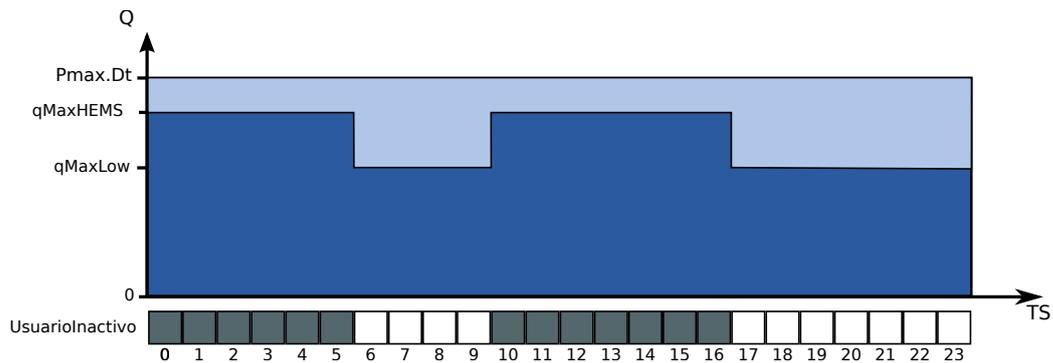


Figura 4.10: Diagrama de la restricción de energía del hogar.

Durante las horas que el usuario se encuentra inactivo se le permite al sistema de gestión utilizar casi la totalidad de la energía del hogar, reservando sólo una porción para los 'electrodomésticos en stand-by'. En cambio, en las horas en las que hay usuarios activos en el hogar, se reserva una porción de energía mayor para que estos puedan hacer uso de los electrodomésticos a su gusto (Figura 4.10).

El vector $Q^{max}[t]$ a utilizar en la Ecuación 4.8 resultante es el siguiente:

$$Q^{max}[t] = \begin{cases} qMaxHEMS & \text{si UsuarioInactivo}[t] = 1 \\ qMaxLow & \text{si UsuarioInactivo}[t] = 0 \end{cases}$$

No es un problema trivial determinar valores adecuados para los parámetros qMaxHEMS y qMaxLow. Los mismos dependen del consumo de los electrodomésticos presentes en el hogar y del uso que le da el usuario a los mismos. En este proyecto se utilizaron valores que estimaron correctos, pero no se realizó un estudio exhaustivo sobre este punto.

Capítulo 5

Implementación del Nodo Controlador

Como se describió en el Capítulo 3 el Nodo Controlador es el cerebro del sistema HEMS. Éste es el encargado de ejecutar el algoritmo de gestión, enviar comandos a los Nodos de Carga, comunicarse con la empresa suministradora y con el usuario.

En el presente capítulo se describe cómo se implementó el Nodo Controlador del sistema HEMS. En primer lugar se detallan las funciones que cumple el Nodo Controlador y las decisiones de diseño tomadas. Luego se describe la arquitectura del Nodo Controlador y los módulos que lo componen. En las secciones posteriores se describe cómo se implementó cada uno de los módulos y cómo se coordinó el funcionamiento de los mismos.

5.1. Descripción del Nodo Controlador

En el Nodo Controlador reside la inteligencia del sistema HEMS. El funcionamiento del mismo es complejo ya que éste debe cumplir diversas funciones, las cuales se enumeran a continuación.

Funciones del Nodo Controlador:

1. Ejecución del algoritmo de gestión.

El principal cometido del Nodo Controlador es ejecutar el algoritmo de gestión descrito en el Capítulo 4.

2. Comunicación con la empresa suministradora.

Al inicio de cada jornada la empresa suministradora publica la curva de precios del día. Esta curva es una de las entradas fundamentales del algoritmo de gestión. Es responsabilidad del Nodo Controlador actualizar diariamente esta curva para utilizarla en el proceso de optimización.

3. Comunicación con el usuario.

Otra entrada fundamental del algoritmo de gestión son los parámetros de configuración de uso de los electrodomésticos. Éstos son determinados por el usuario, y reflejan qué tareas desea el mismo que los electrodomésticos del sistema ejecuten cada día.

4. Envío de comandos a los Nodos de Carga.

El Nodo Controlador es el encargado de que se cumplan las decisiones tomadas por el algoritmo de gestión. Para ello debe traducir las decisiones a órdenes que sean interpretables por los Nodos de Carga y enviar las órdenes correspondientes.

Capítulo 5. Implementación del Nodo Controlador

5. Registro de las mediciones tomadas por los sensores del HEMS.

El Nodo de Carga es el responsable de consultar periódicamente las mediciones registradas por los nodos sensores del sistema y de almacenarla en la base de datos.

Hardware utilizado:

El Nodo Controlador está compuesto por una placa Raspberry Pi 2 y un módulo inalámbrico Z1 de Zollertia. La inteligencia del Nodo Controlador radica en la placa Raspberry Pi 2, mientras que el módulo inalámbrico es utilizado sólo para la comunicación con los Nodos de Carga del sistema.

Lenguaje de programación utilizado:

En el Nodo Controlador se utilizó *Python* [32] como lenguaje de programación. *Python* es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. *Python* es un lenguaje fácil de aprender y potente. Tiene estructuras de datos de alto nivel eficientes y un enfoque simple pero efectivo de programación orientada a objetos. Se trata de un lenguaje de programación multiparadigma, ya que soporta además de orientación a objetos, programación imperativa y, en menor medida, programación funcional. Su sintaxis y tipado dinámico, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para el desarrollo de aplicaciones en muchas áreas en la mayoría de las plataformas.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation que es compatible con la Licencia GNU GPL (GNU General Public License) a partir de la versión 2.1.1[33].

La primera motivación para elegir este lenguaje fue que la librería *wsmcommunication* está implementada en *Python*. Esta librería fue implementada en “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1] por Belcredi, Modernell y Sosa y permite la comunicación entre el Nodo Controlador y los Nodos de Carga. En el Apéndice G se encuentra una descripción de esta librería y de las funcionalidades que ofrece.

Por otro lado, también existen múltiples bibliotecas desarrolladas en *Python* que facilitaron la implementación del Nodo Controlador. Por ejemplo, se utilizó la librería *CVXOPT* para la resolución del problema de optimización y *asyncio* y *threading* para el manejo de la concurrencia. En el Apéndice F se describen las principales librerías de *Python* utilizadas.

5.2. Arquitectura del Nodo Controlador

Tal como fue descrito en las secciones anteriores, el Nodo Controlador presenta múltiples funcionalidades. Para simplificar el diseño del mismo, éste fue dividido en módulos que cumplen con funciones más sencillas. En la Figura 5.1 se puede apreciar un diagrama de la arquitectura del Nodo Controlador detallando con los distintos módulos que lo componen.

La **clase Electrodoméstico** y sus subclases encapsulan toda la información de relevancia de los electrodomésticos a controlar. Por tal motivo, el Nodo Controlador cuenta con una instancia de estas subclases por cada dispositivo a controlar.

El **módulo Configurator** es el encargado de cargar y mantener actualizada la información dentro del Nodo Controlador para que pueda ser utilizada por los otros módulos. Esto incluye crear los electrodomésticos, actualizar diariamente la curva de precios $p(t)$, los requerimientos del usuario y el resto de la información de relevancia.

5.2. Arquitectura del Nodo Controlador

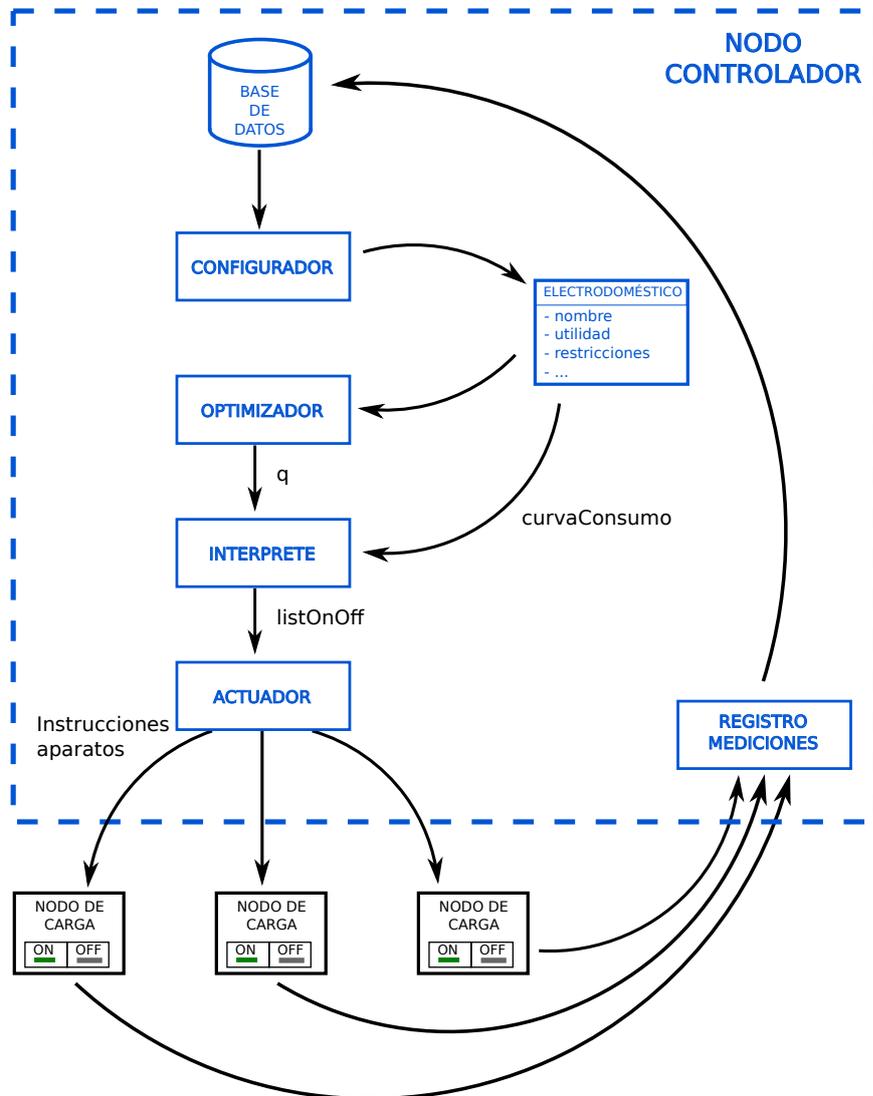


Figura 5.1: Módulos del Nodo Controlador

El **módulo Optimizador** es responsable de determinar cuánta energía se le debe suministrar a cada electrodoméstico en cada *timeslot* (q_a). Es decir, debe ejecutar el algoritmo de optimización descrito en el Capítulo 4.

La salida del módulo Optimizador q_a tiene que ser procesada y traducida a órdenes que puedan ser ejecutadas por los electrodomésticos. Esta es la función del **módulo Intérprete**, el cual utiliza la información acerca de cómo es el perfil de consumo del electrodoméstico para traducir los vectores de energía a una lista de vectores binarios `listOnOff`. Los vectores en la lista indican en qué estado, encendido o apagado, debe estar el electrodoméstico en cada minuto.

El **módulo Actuador** es el encargado de asegurar que los electrodomésticos estén en todo momento en el estado indicado por el vector en `listOnOff`. Para cumplir con tal objetivo el

Capítulo 5. Implementación del Nodo Controlador

Actuador debe enviar comandos de encendido o apagado a los Nodos de Carga en los instantes correspondientes.

Por último, el **módulo Registro de Mediciones** es el encargado de consultar las medidas obtenidas de los distintos Nodos de Carga, procesarlas y almacenarlas en la base de datos.

5.3. Clases de electrodomésticos

En el Capítulo 4 se describieron los modelos utilizados para los distintos electrodomésticos del hogar. Estos modelos no son simples y presentan múltiples parámetros, parte de los cuales varían diariamente de acuerdo a lo configurado por el usuario. Para que el manejo de esta información sea más sencilla, se crearon distintas clases de electrodomésticos que encapsulan toda la información referente a las particularidades de los mismos.

Los módulos que componen el Nodo Controlador requieren conocer cierta información acerca de los electrodomésticos, a la cual acceden a través de atributos y funciones presentes en las clases de electrodomésticos.

En primer lugar se definió la clase Electrodoméstico. Esta es una clase abstracta que actúa como interfaz para sus subclasses. Su función es definir todos los atributos y funciones que son utilizados por los módulos, con fin de asegurar que sean implementados por todas las subclasses de electrodomésticos.

Toda clase que implemente algún tipo de electrodoméstico, debe heredar de la clase Electrodoméstico, y por lo tanto debe implementar todas sus funciones y atributos. En este proyecto se crearon tres clases instanciables que heredan de Electrodoméstico: Calefón, HVAC y CargaDiscreta. En la Figura 5.2 se puede ver un diagrama de estas clases.

Ancho de *timeslot* de los electrodomésticos

El Nodo Controlador está implementado de manera de que cada electrodoméstico pueda tener un tamaño de *timeslot* distinto. Los largos de *timeslot* posibles son: 60, 30, 20 o 15 minutos. Un *timeslot* es el paso temporal del proceso de optimización. La duración de un *timeslot* se determina con el atributo `resolucionTS`, el cual indica en cuánto se particiona un *timeslot* de una hora de un electrodoméstico.

Los *timeslot* más pequeños dan mayor flexibilidad al usuario, y le permiten programar los electrodomésticos a su gusto. Sin embargo los *timeslot* pequeños, hacen que la variable de optimización `q` aumente su dimensión, lo cual hace que el Optimizador demore más en converger.

Clase abstracta Electrodoméstico

La clase Electrodoméstico es una clase abstracta que actúa como interfaz para las subclasses de electrodomésticos. La misma define todos los atributos y funciones que necesariamente debe ser implementados por las subclasses de electrodomésticos. Los cuales se describen a continuación.

Atributos

- `idElectrodomestico`: *entero* que identifica al electrodoméstico.
- `nombreElectrodomestico`: *string* que identifica al electrodoméstico.

5.3. Clases de electrodomésticos

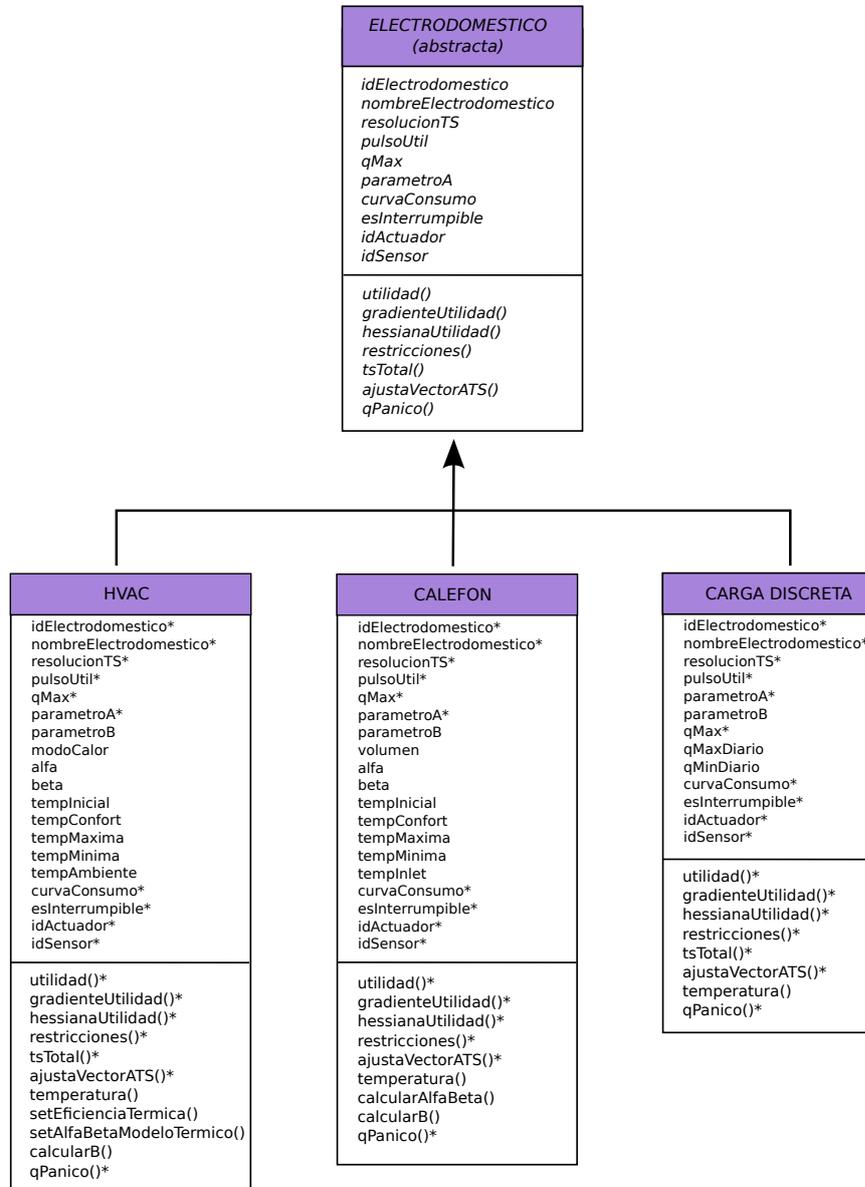


Figura 5.2: Diagrama de clases de Nodo Controlador

- **resolucionTS**: *entero* que especifica en cuántos *timeslot* está dividido el intervalo temporal de una hora.
- **pulsoUtil**: *vector binario* que especifica cuáles son los *timeslot* que se encuentran dentro del intervalo de interés (\mathcal{T}_a) del usuario.

Como se describió en el Capítulo 4, el significado del intervalo de interés (\mathcal{T}_a) depende de cada electrodoméstico. En los electrodomésticos de consumo discreto define los *timeslot* en los cuales se puede ejecutar la tarea. Mientras que en los dispositivos de acondicionamiento térmico, indica los *timeslot* en los cuales la temperatura debe estar acondicionada.

Capítulo 5. Implementación del Nodo Controlador

- **qMax**: *entero* que indica la energía máxima que el Optimizador puede suministrarle al electrodoméstico en una hora.
- **parametroA**: *entero* que representa el coeficiente de proporcionalidad de la función de utilidad.
Este parámetro es utilizado por el Optimizador en el plan de contingencia ante no convergencia, ver Sección 5.5. A través de este parámetro es posible ajustar la amplitud de las variaciones de la utilidad.
- **curvaConsumo**: *vector entero* que representa la variable que describe el consumo eléctrico del electrodoméstico minuto a minuto en Wh. Se previeron dos posibilidades:
 - Se especifica la curva de consumo al crear la instancia del electrodoméstico.
 - No se especifica la curva. En este caso se asume que el electrodoméstico presenta un consumo constante igual a: $qMax/60$.
- **esInterrumpible**: *booleano* que indica si el electrodoméstico es interrumpible o no. Esta información es utilizada por el Intérprete como se describe en la Sección 5.6.
- **idActuador**: *entero* que identifica al Nodo de Carga que opera el electrodoméstico.
- **idSensor**: *entero* que identifica al sensor asociado.
Si **idSensor** = **null** el electrodoméstico no tiene ningún sensor asociado.

Funciones

- **utilidad(q_a)**: devuelve la utilidad del electrodoméstico evaluada en el punto q_a .
- **gradienteUtilidad(q_a)**: devuelve el gradiente de la función de utilidad del electrodoméstico evaluado en q_a .
- **hessianaUtilidad(q_a)**: devuelve la matriz Hessiana de la función de utilidad del electrodoméstico evaluada en q_a .
- **restricciones()**: devuelve las restricciones G y h del electrodoméstico configurados para el día.
- **tsTotal()**: función auxiliar que devuelve el largo del vector de optimización q_a . Es decir devuelve la cantidad de *timeslot* en un día.

$$\text{Time slot en un día} = \text{resolucionTS} * 24$$

- **ajustaVecHorarioATS(v)**: función auxiliar que permite ajustar un vector v de largo 24 a uno de largo **tsTotal**. Para esto se repite cada entrada de v tantas veces como **resolucionTS**.
- **qPanico()** función que devuelve un vector q_a razonable para el electrodomésticos. Este vector es utilizado por el Optimizador cuando entra en estado de pánico, lo que se explica más adelante en la Sección 5.5 del presente capítulo.

A continuación se enumeran las subclases de electrodomésticos implementadas en este proyecto. En el Apéndice D se describen estas clases en detalle y cómo fueron implementadas.

Clase HVAC

La clase HVAC representa a un aire acondicionado y encapsula todo la información relevante del mismo. Esta clase implementa todas las funciones y atributos declaradas en la clase Electrodoméstico utilizando los modelos especificados en la Sección 4.3 y de acuerdo a lo que haya sido configurado por el usuario.

Clase Calefón

La clase Calefón representa un calefón eléctrico doméstico y utiliza los modelos descritos en la Sección 4.4.

Clase CargaDiscreta

La clase CargaDiscreta permite modelar el funcionamiento de los electrodomésticos de consumo discreto, de acuerdo a lo definido en la Sección 4.5.

Es posible que en aplicaciones futuras se desee agregar otro tipo de electrodomésticos. En ese caso basta con crear una nueva clase que herede de Electrodoméstico y editar el módulo Configurador de modo que pueda actualizar la información dicho electrodoméstico.

5.4. Módulo Configurador

El **Módulo Configurador** es el encargado de cargar y mantener actualizada la información dentro del Nodo Controlador para que pueda ser utilizada por los otros módulos. Sus funcionalidades incluyen obtener los datos de los electrodomésticos, actualizar diariamente la curva de precios $p(t)$, obtener diariamente la predicción de temperatura exterior y actualizar los requerimientos del usuario. El módulo Configurador tiene como cometido obtener la información antes mencionada desde la base de datos, procesarla y almacenarla.

Base de datos

Se cuenta con una base de datos local dentro del Nodo Controlador cuyo propósito es almacenar tanto los parámetros de los electrodomésticos del hogar, como las variaciones en los precios y temperaturas, y los requerimientos configurables por el usuario. Allí también se guardan las decisiones tomadas por el algoritmo y las mediciones de los distintos sensores del sistema. Para este prototipo en particular se decidió utilizar la Base de Datos MySQL.

MySQL

MySQL[34] es una de las bases de datos de código abiertos más populares respaldada por una amplia comunidad de desarrolladores, entusiastas y empresas como Facebook, Twitter y Youtube. En esta oportunidad se decidió utilizar su edición comunitaria la cuál puede ser obtenida gratuitamente y está disponible bajo la licencia GPL. MySQL está disponible para las mayores plataformas, incluyendo el sistema operativo Raspbian utilizado en la Raspberry Pi[27].

Las razones más importantes para elegir MySQL como motor de base de datos fueron su alto rendimiento, escalabilidad y flexibilidad. Además, MySQL es soportado en todas las versiones de Linux, UNIX y Windows. La naturaleza de código abierto de MySQL permite una personalización completa para aquellos que quieran añadir requisitos únicos para el servidor de base de datos.

Por otra parte MySQL presenta una aplicación integral al desarrollo. Es decir, que proporciona soporte completo para todas las necesidades de desarrollo de aplicaciones. Dentro de la base de datos, se puede encontrar soporte para procedimientos almacenados, triggers, funciones, vistas, cursores, y más. MySQL también proporciona conectores y manejadores (ODBC, JDBC, etc.) que permiten a todo tipo de aplicaciones hacer uso de MySQL como un servidor de gestión de datos preferido.

Por último, es relevante destacar su capacidad de iniciar rápidamente, su simple proceso de instalación, y su completa suite de herramientas de mantenimiento y gestión.

Capítulo 5. Implementación del Nodo Controlador

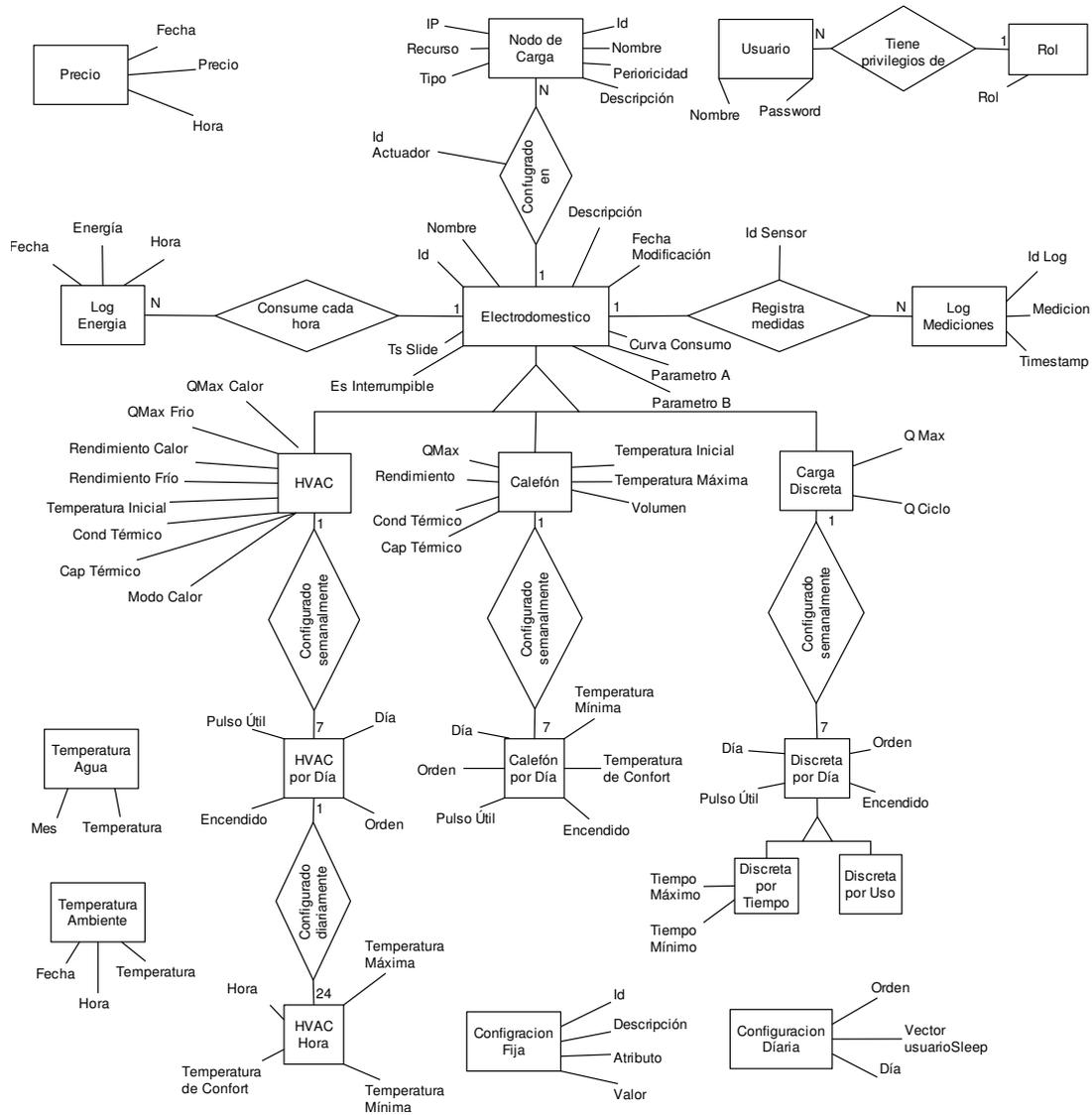


Figura 5.3: Modelo Entidad Relación

Modelo Entidad Relación

La Figura 5.3 representa el modelo entidad relación del Configurador. Por más detalle acerca de las entidades y atributos definidos ver el Apéndice H.

Implementación

La función más importante de este módulo es **configurador** la cual devuelve toda la información necesaria para llevar a cabo luego el proceso de optimización.

Función: `p listaElectros qCasa vecUsuarioInactivo = configurador([date])`

Esta función se encarga de recolectar cada uno de los datos necesarios para ejecutar la optimización para el día siguiente. Es decir, levanta los datos contenidos en la base de datos que más tarde servirán como parámetros de entrada para el algoritmo de optimización.

La misma recibe como parámetro opcional la fecha para la cual se desea obtener la configuración disponible en la base de datos. En caso de no recibir ese parámetro se asume como fecha por defecto el día siguiente y se obtienen las configuraciones necesarias para ese día. Este parámetro opcional no es utilizado en el funcionamiento normal del sistema HEMS, sino que fue utilizado durante la evaluación del funcionamiento del sistema en el Capítulo 8.

La función retorna: `p` la curva de precios del día, `listaElectros` lista de electrodomésticos a gestionar en ese día, `qCasa` la energía máxima que puede ser consumida en una hora por el hogar y `vectorUsuarioInactivo` vector en el que se indica las horas en las cuales no hay usuarios activos en el hogar.

El resto de las funciones de este módulo son auxiliares. Su funcionamiento no será detallado en este documento, para más información se puede consultar el código anexo.

Realimentación de variables

Para los electrodomésticos del tipo de acondicionamiento térmico se utiliza un valor de temperatura inicial para predecir la evolución de la temperatura a lo largo del día. Este dato debe ser actualizado diariamente por el módulo Configurador. Antes de que se inicie el proceso de optimización del día, el módulo Configurador obtiene la última medida disponible de la tabla `logMedidas` de la base de datos y la incluye para utilizarla proceso de optimización. Si no hay medidas disponibles, el módulo Configurador utiliza el último valor de la predicción de temperatura de la optimización del día anterior, el cual se encuentra guardado en la tabla del electrodoméstico de la base de datos.

Es importante que el valor de temperatura inicial sea obtenido a partir de una medida y no que sea estimado, ya que permite corregir los errores presentes en la estimación de temperaturas del día anterior. Estos errores no son sólo originados por el modelo matemático sino porque el usuario haya realizado un uso del electrodoméstico diferente al planificado. Al utilizar el valor de temperatura medido se mitiga esta fuente de error obteniéndose una mejor predicción de la temperatura.

Datos Exteriores al Sistema

La información presente en la base de datos proviene de distintas fuentes. En algunos casos se trata de parámetros fijos que son configurados cuando se instala el sistema HEMS, como lo son los parámetros físicos de los electrodomésticos o de la casa. Por otro lado, existen parámetros configurables por el usuario que son configurados a través de la interfaz de usuario, ver Capítulo 6. Finalmente, existe información que debe ser obtenida desde fuentes externas al sistema, como los son la curva de precios y las predicciones de temperatura. A continuación de detalla cómo se obtiene esta información en el sistema diseñado.

Temperatura Ambiente

El modelo térmico utilizado para el aire acondicionado precisa una predicción de los valores de temperatura ambiente a lo largo del día. Esta información se obtiene del sitio web *WUunderground*. El mismo brinda una API (Application Programming Interface) que permite consultar cada cierto tiempo la temperatura en determinada zona. Para poder utilizar esta API

Capítulo 5. Implementación del Nodo Controlador

se creó un usuario para este proyecto con fines educativos que permite utilizar este recurso con una periodicidad menor a 500 veces por hora, cumpliendo con todos los términos y condiciones del servicio[35].

A través de la aplicación web, donde también está contenida la Interfaz de Usuario, se desarrolló un servicio que consulta esta API cada cierto tiempo para obtener una predicción de temperatura para los próximos 10 días para la ciudad de Montevideo. Los datos obtenidos son almacenados en la base de datos para luego ser consumidos. Esta información se utiliza tanto para mostrar al usuario una predicción de temperatura desde la interfaz web, tal como se detalla en el Apéndice J, así como por el Optimizador para el modelo térmico del aire acondicionado.

Adicionalmente, se aprovechó este recurso para obtener un historial de temperaturas de un año entero el cual fue utilizado para evaluar el desempeño del algoritmo de gestión, lo cual se describe en el Capítulo 8.

Precio

En la actualidad no existe una tarifa de precios tipo DAP en Uruguay, y por lo tanto no hay ningún agente en la red que publique la curva de precios. Para este proyecto como no existe este recurso, se creó una curva de precios ficticia basada en la curva histórica de precios de una empresa estadounidense. Por más información acerca de cómo se creó esta curva de precios consultar la Sección 8.1.

Para mayor consistencia se guardaron estas tarifas en la base de datos, en caso de que se implemente este recurso en el futuro solo habría que crear un servicio que actualice estos valores de precios en la base de datos.

Con el fin de realizar pruebas del funcionamiento del algoritmo, diariamente el módulo Configurador obtiene la curva de precios correspondiente de la base de datos y la prepara para que sea utilizada por el módulo Optimizador. Esta curva de precios también se muestra a través de la interfaz web para que el usuario puede conocer los precios.

5.5. Módulo Optimizador

El módulo Optimizador es el encargado de determinar diariamente el vector \mathbf{q}_a para cada uno de los electrodomésticos. En otras palabras, debe decidir cuánta energía se le debe suministrar en cada *timeslot* a cada electrodoméstico.

El objetivo del módulo Optimizador es resolver el problema de optimización descrito en el Capítulo 4 para obtener \mathbf{q}_a . Si bien se tomaron ciertos cuidados con el fin de facilitar la resolución del problema y asegurar la convergencia de los métodos iterativos, existe la posibilidad de que el Optimizador no pueda hallar la solución del problema planteado. Ante este inconveniente, el Optimizador igualmente debe devolver alguna solución razonable que cumpla mínimamente con los deseos del usuario.

Resolución del problema de optimización

El problema que debe resolver el Optimizador es el planteado en la Ecuación 4.2. En el mismo se busca minimizar el costo en energía y a la vez maximizar la utilidad de los electrodomésticos.

Dada las funciones de utilidad y las restricciones seleccionadas este es un problema de optimización convexa con restricciones lineales. Por más información acerca de la caracterización del problema ver el Apéndice E.

Una de las ventajas que presentan los problemas de optimización convexos es que son un campo largamente estudiado y se han desarrollado métodos de resolución muy efectivos[36]. Como consecuencia, estos algoritmos ya se encuentran implementados en diversos lenguajes de programación y es fácil acceder a librerías que los incluyan.

En este proyecto se utilizó la librería *CVXOPT* de *Python*. Esta es una librería de software libre de optimización convexa desarrollado por M. Andersen, J. Dahl, y L. Vandenberghe. La misma presenta una serie de funciones que permiten resolver problemas de optimización de distintas características. En este trabajo en particular, se utilizó la función `cvxopt.solvers.cp` que permite resolver problemas convexos no lineales. No se encuentra dentro del alcance del proyecto el estudio de los métodos de resolución del problema de optimización. Por esta razón no se estudio en detalle el funcionamiento de la `cvxopt.solvers.cp`, sino que sólo se buscó aprender cómo utilizarla adecuadamente. Por más información acerca de esta función y de la librería *CVXOPT* consultar el Apéndice E.

Plan de contingencia ante no convergencia

Si bien se tomaron ciertos cuidados para facilitar la resolución del problema existe la posibilidad de que la función `cvxopt.solvers.cp` devuelva una solución marcada con la bandera `status = unknown`. Esto indica que el método iterativo no alcanzó a una solución cercana a la óptima. Existen dos razones por las cuales puede suceder esto. La primera es que el algoritmo encuentre un problema numérico en el proceso iterativo. La segunda es que el método llegue al máximo de pasos iterativos admitido ¹. El Optimizador presenta un plan de contingencia ante este inconveniente de modo de devolver una solución razonable que cumpla mínimamente con los deseos del usuario.

Acondicionamiento del problema

En primer lugar, se acondicionó el problema de modo de facilitar su resolución de acuerdo a indicaciones encontradas en la documentación de la librería[37] y foros de consulta de la misma. Este acondicionamiento incluyó: eliminar de la lista de electrodomésticos a optimizar aquellos electrodomésticos no agendados porque su solución es trivial ($q_a[t] = 0 \forall t$), eliminar restricciones redundantes y normalizar las restricciones.

Medidas correctivas

Una vez acondicionado el problema se observó que, si bien la probabilidad de que el algoritmo no converja se redujo, aún existían algunos casos de no convergencia. Estos casos se debían principalmente a problemas numéricos y podían ser solucionados mediante pequeños cambios en la formulación del problema.

Mediante pruebas se encontró que variando el parámetro a de la función del electrodoméstico se podía eludir el problema numérico y lograr que el algoritmo alcance una solución óptima. Por esta razón, se decidió agregar esto como una medida correctiva ante la no convergencia. Cada vez que el algoritmo no converge se re intenta resolver problema variando levemente los valores de a .

El plan de correcciones implementado fue el siguiente: en cada paso correctivo se multiplica el parámetro a de cada electrodoméstico por un factor correctivo. A su vez, existe un máximo

¹Para realizar la simulación del funcionamiento del algoritmo, ver Sección 8.1, se utilizó un máximo de 120 pasos iterativos

Capítulo 5. Implementación del Nodo Controlador

de pasos correctivos configurable ².

Observación:

Al modificar el parámetro a de la función de utilidad, se reduce el peso que se le da al confort del usuario en la función de beneficio. Y por lo tanto el Optimizador va a tender a obtener soluciones menos confortables para el usuario, pero que igualmente cumplen con las restricciones originales. Por más información acerca del significado del parámetro a y de cómo se obtuvieron los valores para cada electrodomésticos consultar el Apéndice C.

Solución “de pánico”

En caso que el Optimizador no alcance una solución óptima luego de aplicar todos los pasos correctivos, se dice que el Optimizador entra en estado de pánico. En dicho estado el módulo devuelve una solución “de pánico”, la cual busca cumplir mínimamente con lo que desea el usuario sin tener en consideración la curva de precios.

Tal como se describió en la Sección 5.3, cada electrodoméstico presenta un función `qPanico()` la cual devuelve un vector \mathbf{q} que cumple con los requerimientos mínimos para ese electrodoméstico. En el Apéndice D se describe cómo se implementó esta función para cada uno de los electrodomésticos.

El Optimizador cuando entra en estado de pánico: consulta la solución de pánico de cada uno de los electrodomésticos de la lista y arma una solución concatenando la solución de cada electrodoméstico. Luego verifica que se cumpla la restricción de energía máxima del hogar. En el caso que se rompa esta restricción, se ‘des-agenda’ el último electrodoméstico de la lista asignándole $q_a[t] = 0 \forall t \in [0, N - 1]$.

Implementación

Las funcionalidades del módulo Optimizador pueden ser accedidas por otros módulos a través de la función `optimizador`.

Función: `optimizador(p listaElectros qCasa vectorUsuarioInactivo)`

Esta función ejecuta el proceso de optimización y devuelve el resultado de la optimización \mathbf{q} , que es la concatenación de los vectores \mathbf{q}_a de cada electrodoméstico.

Se debe pasar como parámetro: p la curva de precios del día, `listaElectros` lista de electrodomésticos a gestionar en ese día, `qCasa` la energía máxima que puede ser consumida en una hora por el hogar y `vectorUsuarioInactivo`.

El pseudo-código de la función es el siguiente:

```
def optimizador(p listaElectros qCasa vecUsuarioInactivo):
    filtrarElectrodomesticosNoAgendados()
    [G h]= armarRestricciones(listaElectros qCasa vecUsuarioInactivo)
    for nroCorrecciones in range(0, MAX_CORRECCIONES+1):
        sol = cvxopt.solvers.cp(F G h)
    if status is optimal:
        break
```

²Para realizar la simulación del funcionamiento del algoritmo, ver Sección 8.1, se optó por utilizar un factor igual a 0,8 y un máximo de ocho pasos correctivos. Se utilizó este par de valores porque los mismos se comportaban correctamente en la práctica, pero no se realizó un estudio detallado para determinar los mismos.

```

else:
    % Paso correctivo
    for electro in listaElectros:
        electro.parametroA *= factorCorrectivo
if status is optimal:
    qSolucion = sol['x']
elif status is unknown:
    % Optimizador en estado de panico
    qSolucion = armarQPanico(listaElectros)
agregar_electrodomesticos_no_agendados()
return qSolucion

```

Función: `concatenarRestricciones(listaElectros qCasa vecUsuarioInactivo)`

Esta función genera las matrices G y h que especifican las restricciones del problema de optimización, ver Ecuación 4.2. En primer lugar, obtiene las restricciones de cada uno de los electrodomésticos pertenecientes a la lista `listaElectros`. Para esto utiliza la función `restricciones()` definida en cada una de las clases de electrodomésticos. Luego calcula la restricción del hogar, utilizando los parámetros `vectorUsuarioInactivo` y `qCasa` de acuerdo a las características indicadas en la Sección 4.6. Todas estas restricciones se concatenan convenientemente para formar las matrices G y h .

5.6. Módulo Intérprete

El módulo Intérprete es el encargado de procesar la salida del Optimizador y traducirla a vectores de estado que indican en qué estado debe estar el electrodoméstico en cada minuto del día. Este módulo además debe buscar minimizar la simultaneidad entre cargas a fin de cuidar que no se supere la potencia contratada del hogar. En esta sección se describe el funcionamiento de este módulo y cómo fue implementado el mismo.

Descripción

En la Figura 5.4 se puede ver un diagrama de bloques del funcionamiento del Intérprete, con sus entradas y salidas.

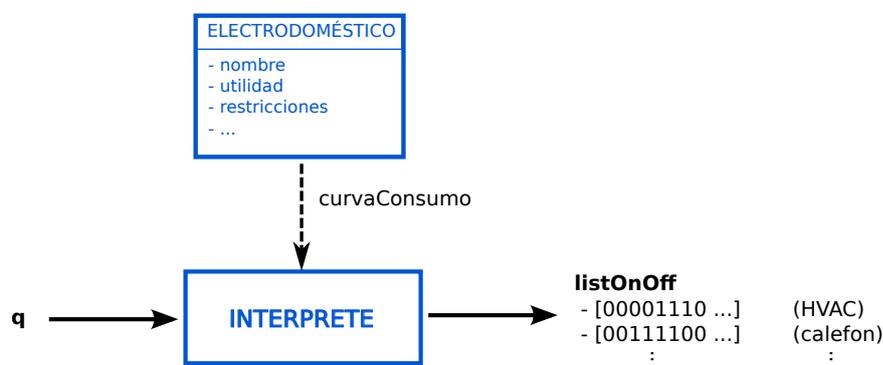


Figura 5.4: Diagrama de bloques del módulo Intérprete

Capítulo 5. Implementación del Nodo Controlador

La entrada del Intérprete, es un vector \mathbf{q} el cual es la concatenación de los vectores \mathbf{q}_a de cada electrodoméstico. Estos \mathbf{q}_a indican cuánta energía se debe suministrar al dispositivo en cada *timeslot*. El módulo Intérprete debe procesar esta información y traducirla lo más fielmente posible a instrucciones de encendido y apagado.

La salida del Intérprete es una lista de vectores de estados. La lista contiene un vector para cada electrodoméstico. Y cada vector indica en qué estado debe estar el electrodoméstico en cada minuto del día. En el marco de este proyecto se consideraron dos estados posibles:

- **Encendido (ON)**. En este estado se acciona el electrodoméstico para que consuma energía.
- **Apagado o en pausa (OFF)**. Se corta el suministro de energía al electrodoméstico. Existen algunos equipos, como un lavarropas, que si se les corta el suministro de energía luego no se re-establecen correctamente. En estos casos el Nodo de Carga debe pausar el proceso en vez de cortar el suministro.

Objetivos del Intérprete

A la hora de decidir en qué estado debe estar cada electrodoméstico en cada minuto del día, el Intérprete debe tener en cuenta los siguientes aspectos.

1. Cumplir con el objetivo de energía $q_a[t]$.

No es posible regular cuánta potencia consumen los electrodomésticos convencionales. Por lo que la manera de conseguir que un electrodoméstico consuma una cantidad de energía $q_a[t]$ en un *timeslot* es mantener encendido el electrodoméstico el tiempo suficiente para que consuma $q_a[t]$ y luego apagarlo. El Intérprete conoce cómo es la curva de consumo de cada electrodoméstico y puede, por lo tanto, determinar el tiempo de encendido necesario para que el dispositivo consuma $q_a[t]$.

El Intérprete no siempre puede conseguir que el consumo del electrodoméstico en el *timeslot* t sea exactamente igual a $q_a[t]$. Las principales razones para esto son las siguientes:

- Las decisiones de encendido o apagado tiene una resolución de un minuto. Por lo que el Intérprete debe redondear este tiempo hacia una cantidad entera de minutos. La energía que se consume de más (o de menos) en un *timeslot* es restada (o sumada) al próximo *timeslot* que cumpla $q_a[t] \neq 0$.
- El tiempo necesario para que el dispositivo consuma $q_a[t]$ es superior a un *timeslot*. Esto puede suceder en los electrodomésticos que presentan una curva de consumo con gran variabilidad y se debe a que el Optimizador no dispone de información detallada de cómo es su curva de consumo.

El Optimizador, sólo conoce la energía promedio que consume el dispositivo en cada *timeslot* (q_{max}^*). De acuerdo a las restricciones del Optimizador el valor de $q_a[t]$ debe pertenecer al intervalo $[0, q_{max}^*]$. En los electrodomésticos que presentan una curva de consumo variable, es posible que existan períodos en los cuales el consumo real del electrodoméstico es muy bajo. A modo de ejemplo, el ciclo de llenado de agua en un lavarropas. Si el Optimizador decide dar una cantidad de energía al electrodoméstico cercana a q_{max} , cuando el mismo se encuentra en un período de bajo consumo, no va a ser posible suministrar esa cantidad de energía en un sólo *timeslot*. En este caso el Intérprete decide encender el dispositivo durante todo el *timeslot* y pasar la energía faltante al próximo *timeslot* con $q_a[t] \neq 0$.

2. Minimizar las interrupciones en los electrodomésticos no interrumpibles.

En la Sección 2.4 se explicó que existen electrodomésticos que una vez encendidos no es conveniente interrumpir su proceso. Esta característica no es considerada por el algoritmo propuesto en [18]. En la búsqueda bibliográfica se encontraron propuestas acerca de cómo modificar el algoritmo de forma de contemplar estos electrodomésticos. Sin embargo, estas modificaciones suponen un cambio importante en el planteo del problema de optimización y el mismo no puede ser resuelto con los métodos de resolución utilizados. Por esta razón se decidió dejar fuera del alcance del proyecto la implementación de estas modificaciones.

El algoritmo no contempla la existencia de electrodomésticos no interrumpibles, por lo que es posible que decida suministrar energía a estos electrodomésticos en *timeslots* disjuntos. En este caso el Intérprete no puede tomar ninguna acción para que el electrodoméstico no se interrumpa. Sin embargo, debe tomar los siguientes cuidados:

- no interrumpir el funcionamiento del dispositivo dentro de un *timeslot*. En otras palabras, todos los minutos de encendido dentro de un *timeslot* deben ser contiguos.
- procurar juntar los minutos de encendido entre dos *timeslots* adyacentes.

3. Minimizar la simultaneidad entre las cargas.

El Intérprete debe coordinar el consumo de los distintos electrodomésticos con el fin de que el consumo agregado de los electrodomésticos sea lo más uniforme posible a lo largo del *timeslot*.

Implementación

La implementación del módulo Intérprete se encuentra en el archivo “Interprete.py”. Las funciones más relevantes de este módulo son `getPerfilConsumo()` y `distribuirCargas()`. A continuación se describe cómo fueron implementadas las mismas.

Función: `getPerfilConsumo(electro q)`

Función que devuelve el perfil de consumo del electrodoméstico `electro` en cada *timeslot* de acuerdo al vector de energía `q`. Esta es una función auxiliar utilizada por la función `distribuirCargas()`.

La función `getPerfilConsumo()` especifica cuántos minutos debe estar encendido el electrodoméstico en cada *timeslot* para cumplir con la consigna del vector `q`. También especifica cuánta energía, en Wh, consume el electrodoméstico en cada uno de los minutos que está encendido.

Esta información es devuelta en una lista de largo igual a la cantidad de *timeslots* en el día. Sus elementos son:

- Un elemento vacío si $q_a[t] = 0$.
- Vector con el perfil de consumo si $q_a[t] \neq 0$. Este vector contiene el consumo minuto a minuto de energía.

Esta función utiliza el atributo `curvaConsumo` de la clases de electrodomésticos para conocer en detalle el consumo de energía del mismo. Este atributo puede presentar dos tipos de datos:

- Vector conteniendo el consumo minuto a minuto del electrodoméstico en Wh.
- Número flotante. En este caso, se asume que el electrodoméstico presenta una curva consumo constante igual al valor de `curvaConsumo`.

Capítulo 5. Implementación del Nodo Controlador

Función: `distribuirCargas(q listaDispositivos)`

Esta es la función principal de módulo Intérprete y determina en qué estado debe estar cada electrodoméstico en cada minuto del día.

La misma recibe como parámetros `listaDispositivos`, una lista de electrodomésticos a gestionar en el día, y el vector `q` conteniendo cuánta energía se debe suministrar en cada *timeslot* a cada uno de los electrodomésticos. Y devuelve una lista cuyo largo es la cantidad de electrodomésticos en la `listaDispositivos`. Los elementos de esta lista son vectores booleanos de $24 * 60$ entradas que indican el estado del electrodoméstico en cada minuto del día. Los estados posibles son:

- 1 - encendido (ON).
- 0 - apagado o en pausa (OFF).

Esta función utiliza `getPerfilConsumo(electro q)` para conocer cómo es la curva de consumo de cada electrodoméstico en cada *timeslot*. Luego con esta información decide cómo distribuir el consumo de cada electrodoméstico dentro de cada *timeslot*. A continuación se detalla el algoritmo utilizado para distribuir los consumos.

En primer lugar se procesan los electrodomésticos no interrumpibles. El procedimiento es el siguiente:

1. Se obtiene la curva de consumo del electrodoméstico utilizando `getPerfilConsumo()`.
2. Se procesan los *timeslot* del electrodoméstico en orden creciente, con el siguiente criterio:
 - a) Si el elemento devuelto por `getPerfilConsumo()` para el *timeslot* t es nulo, el electrodoméstico no debe consumir energía en este *timeslot*. En este caso el estado del electrodoméstico es '0' en todos los minutos del *timeslot*.
 - b) Si `getPerfilConsumo()` devuelve un vector de consumo para el *timeslot* t :
 - 1) Si el electrodoméstico se encontraba encendido al finalizar el *timeslot* anterior, se coloca el consumo del electrodoméstico en los minutos iniciales del *timeslot*. De esta manera no se interrumpe el proceso del electrodoméstico.
 - 2) Si el electrodoméstico no se encontraba encendido al finalizar el *timeslot* anterior, se coloca el consumo del electrodoméstico en los minutos finales del *timeslot*. De manera que en el próximo *timeslot* se pueda continuar con el proceso de forma ininterrumpida.
3. Se suma el consumo del electrodoméstico al vector `consumoTotal` en los minutos correspondientes. Este vector lleva registro del consumo acumulado de los electrodomésticos que ya fueron procesados. Se continúa procesando el siguiente electrodoméstico no interrumpible.

Luego de que todos los electrodomésticos no interrumpibles fueron procesados, se procede a procesar los electrodomésticos interrumpibles. Los mismos son procesados de acuerdo a su `resolucionTS` en orden decreciente. El procedimiento utilizado para estos electrodomésticos es el que sigue:

1. Se obtiene la curva de consumo del electrodoméstico utilizando `getPerfilConsumo()`.
2. Se procesan los *timeslot* del electrodoméstico en orden creciente, con el siguiente criterio:
 - a) Si el elemento devuelto por `getPerfilConsumo()` es nulo, el estado del electrodoméstico debe ser '0' en todos los minutos.

- b) En caso contrario, el largo del vector devuelto por `getPerfilConsumo()` indica cuántos minutos debe estar encendido el electrodoméstico. Se identifican los minutos de menor consumo dentro del *timeslot* consultado el vector `consumoTotal`. Luego, se distribuye el consumo del electrodoméstico en los minutos de menor consumo. De esta manera se cumple con el objetivo del punto 3 de la Sección 5.6.
3. Se suma el consumo del electrodoméstico al vector `consumoTotal` en los minutos correspondientes.

5.7. Módulo Actuador

El Actuador es el encargado de enviar comandos a los electrodomésticos de acuerdo al resultado obtenido en el proceso de optimización. Es decir, de acuerdo a los vectores en la lista `listOnOff` que devuelve el Intérprete. Dichos vectores contienen el estado en el cual debe estar el electrodoméstico en cada minuto del día.

Los estados posibles son:

- 1 - encendido (ON).
- 0 - apagado o en pausa (OFF).

Funcionalidades utilizadas de `wsncommunication`

Con el propósito de lograr la comunicación del Nodo Controlador con los Nodos de Carga de la red se utilizó el módulo *wsncommunication*, implementada en “Controlador de energía domiciliario para una Red Eléctrica Inteligente”, por Belcredi, Modernell y Sosa[1]. Este módulo brinda las funciones básicas necesarias para descubrir los nodos presentes en la red, obtener mediciones desde los Nodos de Carga y enviar comandos a los mismos.

El módulo *wsncommunication* utiliza la librería *asyncio*, la cual fue descrita en el Apéndice F. Algunas de las funcionalidades que ofrece este módulo son corrutinas de esta librería y deben por lo tanto ser invocadas como tales. Esta fue la razón que motivó el uso de la librería *asyncio* en la implementación del Actuador.

A continuación se describen, sin entrar en detalle, cuáles son las funcionalidades utilizadas en el módulo Actuador³.

- **Clase `SgNode`.**
Clase que encapsula toda la información relacionada con un Nodo de Carga, incluyendo nombre y dirección IPv6 de los mismos.
- **Clase `SgNList`.**
Esta clase representa una lista de todos los nodos presentes en la red HAN. Ofrece las funcionalidades: `update()`, la cual actualiza la lista de nodos presentes en la red, y `findNode()` que busca un nodo dentro de la lista por su dirección IPv6 y lo devuelve.
- **Corrutina `wsncommunication.putCommand(nodo comm [recurso])`.**
Esta corrutina permite enviar comandos a los Nodos de Carga. Recibe como parámetro: `nodo` un objeto del tipo `SgNode` representando al nodo que se quiere comandar, `comm` string con el comando a enviar y `recurso` string que indica a qué recurso dentro del nodo se debe acceder.

³Por más información acerca del módulo *wsncommunication* consultar el Apéndice G

Capítulo 5. Implementación del Nodo Controlador

Implementación

El Actuador debe enviar comandos a los electrodomésticos de acuerdo al resultado obtenido en el proceso de optimización. Es decir, de acuerdo a los vectores de estados presentes en la lista `listOnOff` que devuelve el Intérprete. Estos comandos deben ser enviados desde el comienzo hasta el fin de la jornada, o sea desde las 00:00 hs. hasta las 23.59 hs. Como consecuencia, es necesario que la información de `listOnOff` se actualice antes de las 00:00 hs del día.

En este módulo se implementaron dos corrutinas. Por un lado la corrutina `controlElectro()` encargada de recorrer el vector de estados y enviar órdenes a un electrodoméstico. El Actuador presenta una instancia de esta corrutina por cada electrodoméstico controlable. Por otro lado, se implementó la corrutina `actualizadorVectoresOnOff()`, la cual se encarga de consultar diariamente el resultado del proceso de optimización y de enviarle esta información a la corrutina `controlElectro()`. La comunicación entre las corrutinas `controlElectro()` y `actualizadorVectoresOnOff()` se implementó mediante el uso colas de la librería *asyncio*. Por más información acerca del concepto de corrutina y la librería *asyncio* consultar el Apéndice F.

Corrutina: controlElectro(nodo recurso colaElectro)

Esta corrutina es la encargada de enviar comandos al Nodo de Carga de un electrodoméstico de acuerdo a las instrucciones recibidas en el vector de estados `vectorOnOff`.

Recibe como parámetro: `nodo`, instancia de la clase `SgNode` que representa el nodo que comanda el electrodoméstico, `recurso` un string que indica el recurso del nodo al cual se debe acceder y `colaElectro` la cola a través de la cual se obtiene el vector `vectorOnOff` de cada día.

El pseudo-código de la corrutina es el siguiente:

```
def controlElectro(nodo, recurso, colaElectro):
    while True:
        % Comienza un nuevo dia
        vectorOnOff = colaElectro.get()
        for instruccion in vectorOnOff:
            if instruccion = 0
                putCommand(nodo, OFF, recurso)
            if instruccion = 1
                putCommand(nodo, ON, recurso)
            sleep(~60 segundos)          % La corrutina se bloquea
```

En primer lugar, se observa que el largo del vector `vectorOnOff` recibido, debe ser igual a la cantidad de minutos en un día. De esta forma, si se ejecuta una instrucción por minuto, la última es enviada a las 23:59. Se asume que el vector de estados va a estar disponible en la cola antes de las 00:00 del día, lo cual es responsabilidad de la corrutina `actualizadorVectoresOnOff()`.

Por otro lado, es importante destacar que el tiempo que debe dormir la corrutina no son 60 segundos exactamente. Sino que se debe restar el tiempo consumido desde que se despertó la corrutina hasta que se terminó de enviar el comando. De esta forma se consigue que la corrutina envíe comandos cada 60 segundos exactamente.

Se observa que en todos los minutos se envía al electrodoméstico el estado en el que debe estar. Esto se debe a que los comandos enviados no sólo son utilizados para cambiar el estado del electrodoméstico, sino que también permiten la verificación de que el canal de comunicación está funcionando correctamente. Dicha verificación es necesaria debido a un inconveniente encontrado en los Nodos de Carga, ver Apéndice L.

Corrutina: `actualizadorVectoresOnOff(colaOptim listaColasElectro)`

Esta corrutina tiene como cometido obtener el resultado del proceso de optimización, procesarlo y pasarlo los vectores de estado a cada una de las corrutinas `controlElectro()`. Recibe como parámetro `colaOptim`, cola a través de la cual recibe el resultado de la optimización y `listaColasElectro` una lista de colas donde debe colocar los vectores obtenidos. El pseudo-código de esta corrutina es el siguiente:

```
def actualizadorVectoresOnOff(colaOptimizador listaColasElectro):
    while True:
        sleep(segundos hasta horaOptimizacion)
        while colaOptimizador esta vacia:
            sleep(periodo)                % se chequea el estado de
                                         la cola periodicamente
        listOnOff = colaOptimizador.get()
        for i in range(0, cantidadElectrodomesticos):
            vectorOnOff = listOnOff[i]
            listaColasElectro[i].put(vectorOnOff)
```

El proceso de optimización comienza a ejecutarse a partir de una hora acordada. A partir de la misma se puede recibir la lista `listOnOff` a través de `colaOptim`. Dado que se desconoce cuánto demora el proceso de optimización se consulta periódicamente el estado de la cola hasta detectar que hay un elemento.

Una vez recibida la lista `listOnOff` se separan los distintos vectores de estado y se encola cada vector en la cola del electrodoméstico correspondiente. Se asume que `listOnOff` y `listaColasElectro` presentan el mismo orden que la lista de electrodomésticos.

Función: `actuador(listaElectro colaOptimizador loop)`

Esta es la función principal del módulo Actuador. Su función es crear las distintas instancias de corrutinas y agendarlas en el `eventLoop`.

Recibe como parámetro: `listaElectrodomesticos` la lista de electrodomésticos del sistema, `colaOptimizador` la cola a través de la cual se obtienen los resultados del proceso de optimización y `loop` el *event loop* del *thread* en el cual se ejecuta la función `actuador`.

```
def actuador(listaElectrodomesticos, colaOptimizador, loop)
    asyncio.set_event_loop(loop)
    tasks = []
    listaColasElectro = []
    % Se crean las instancias de la corrutina controlElectro
    for electro in listaElectrodomesticos:
        nodoElectro = listaNodos.findNode(direccionIP)
        nuevaCola = asyncio.Queue()
        nuevaCorrutina = controlElectro(nodoElectro, recurso, nuevaCola)
        tasks.append(nuevaCorrutina)
        listaColasElectro.append(nuevaCola)

    % Se crea una instancia de la corrutina actualizadorVectoresOnOff
    tasks.append(actualizadorVectoresOnOff(colaOptimizador,
                                           listaColasElectro))

    % Se agendan todas las tareas en el event loop
```

Capítulo 5. Implementación del Nodo Controlador

```
loop.run_until_complete(asyncio.wait(tasks))
loop.close()
```

En primer lugar, se crea una instancia de corrutina `controlElectro()` por cada electrodoméstico en `listaElectrodomesticos`. Se utiliza la funcionalidad `findNode()` del módulo `wsncommunication` para obtener el nodo actuador asociado a cada electrodoméstico. Posteriormente, se crea una instancia de la corrutina `actualizadorVectoresOnOff()`. Y por último se agendan todas las corrutinas en el `eventLoop`.

5.8. Módulo Registro Mediciones

El sistema de gestión puede presentar distintos sensores que reportan mediciones de interés para el usuario. A modo de ejemplo estas magnitudes pueden ser: consumo de potencia eléctrica, temperatura y humedad. La función del módulo Registro Mediciones es consultar de forma periódica las medidas obtenidas por los sensores, procesarlas y almacenarlas en la base de datos.

Funcionalidades utilizadas de `wsncommunication`

Se hizo uso del módulo `wsncommunication` para obtener las medidas de los sensores y almacenarlas. Además de las funcionalidades descritas en la sección anterior, se empleó la corrutina `getMeasure` del `wsncommunication`.

- **Corrutina `wsncommunication.getMeasure(node, recurso)`.**
Esta corrutina permite adquirir las medidas de los Nodos de Carga. Recibe como parámetro, `node` un objeto del tipo `SgNode` el cual representa al nodo del sensor y `recurso` string que indica a qué recurso se debe acceder para obtener la medida⁴.

Implementación

El módulo Registro Mediciones es el responsable de consultar de forma periódica las medidas obtenidas por los sensores de la red, procesarlas y almacenarlas en la base de datos.

La información acerca de cuáles son los nodos a controlar y sus características se encuentra almacenada también en la base de datos del sistema. Por tanto, lo primero que debe hacer el módulo de Registro Mediciones es obtener la lista de sensores a monitorear. Con tal fin se implementó la clase `EhgSensor` la cual encapsula toda la información relacionada con un sensor y la función `getListaSensores()` encargada de recabar la lista de sensores.

Una vez se conoce la lista de sensores a monitorear, el módulo Registro Mediciones debe enviar de forma periódica solicitudes a cada uno de los sensores. Para ello, se desarrolló la corrutina `logMedidas()`. La misma se encarga de enviar solicitudes a un sensor con la periodicidad correspondiente.

Clase: `EhgSensor`

Esta clase encapsula toda la información referente a un sensor. Extiende la clase `SgNode` agregando atributos relevantes para los sensores, como el recurso que se debe consultar y la frecuencia con la que es preciso enviar las consultas. Los atributos de la clase son los siguientes:

- **id:** *entero* que identifica al sensor dentro de la lista de sensores y actuadores en la base de datos.

⁴Por más información acerca de los Nodos de Cargas y sus recursos consultar Apéndice L

5.9. Programa principal

- **name:** *string* que identifica al sensor.
- **node:** *SgNode* instancia de la clase *SgNode* que encapsula información del Nodo de Carga al cual se le debe enviar las solicitudes.
- **recurso:** *string* que identifica el recurso al que se debe acceder para obtener la medida.
- **periodicidad:** *entero* que indica el período en segundos con el que se deben enviar las solicitudes de medidas.

Función: `getListaSensores()`

Esta función es la encargada de crear la lista de sensores a monitorear. Para esto obtiene la información acerca de los sensores desde la base de datos y crea una instancia de la clase *EhgSensor* por cada sensor. Devuelve una lista conteniendo todos los sensores.

Además, esta función se encarga de verificar que todos los sensores de la red se encuentran accesibles. Para cumplir con tal objetivo se vale de la función `findNode()` del módulo *wsn-communication*. En el caso que alguno de los sensores no sea accesible, este es removido de la lista.

Corrutina: `logMedidas(sensor)`

Esta corrutina es la encargada de obtener las medidas de un sensor con la frecuencia indicada en el atributo `periodicidad` del sensor y almacenarla en la base de datos. Recibe como parámetro el *EhgSensor* al cual se le debe enviar las solicitudes.

Función: `registroMedidas(loop)`

Esta es la función principal del módulo Registro Mediciones. Su función es crear las instancias de corrutinas y agendarlas en el event loop.

El pseudo-código de esta función es el siguiente:

```
def registroMedidas(loop):
    asyncio.set_event_loop(loop)
    tasks = []
    listaSensores = getListaSensores()
    for sensor in listaSensores:
        tasks.append(asyncio.async(logMedidas(sensor)))
    loop.run_until_complete(asyncio.wait(tasks))
    loop.close()
```

5.9. Programa principal

En el presente capítulo se describió el funcionamiento de todos los módulos que componen el Nodo Controlador, es decir, de los módulos Configurator, Optimizador, Intérprete, Actuador y Registro de Mediciones. El propósito de esta sección es describir cómo se inician y se coordinan el funcionamiento de los mismos en el programa principal.

Coordinación de los módulos

Por un lado, los módulos Actuador y Registro de Mediciones deben enviar comando o solicitudes a los Nodos de Carga en forma periódica y ambos deben operar durante las 24 horas del día. Es sencillo concluir que resulta necesario que los módulo Actuador y Registro de Mediciones operen en forma concurrente.

Capítulo 5. Implementación del Nodo Controlador

Por otro lado, los módulos Configurador, Optimizador e Intérprete son los encargados de ejecutar el algoritmo de gestión y obtener la lista `listOnOff` de las próximas 24 horas. Estos módulos debe ejecutarse en forma secuencial, uno detrás del otro, una vez al día.

A priori podría pensarse que los módulos Configurador, Optimizador e Intérprete podrían funcionar en forma secuencial con el módulo Actuador. Sin embargo, el Actuador envía la última orden de día a las 23:59 horas y necesita que la lista `listOnOff` esté pronta antes de las 00:00 hs del día siguiente. Si los módulos funcionaran de manera secuencial, se contaría con menos de un minuto para que los módulos Configurador, Optimizador e Intérprete cumplan con su tarea. En la Sección 8.1 se observó que el proceso de optimización puede demorar varios minutos. Por consiguiente, no es posible que el Optimizador y el Actuador funcionen de forma secuencial.

Se optó por que el Nodo Controlador presente tres tareas distintas que se ejecutan de forma concurrente. La primera ejecuta los módulos Configurador, Optimizador e Intérprete. La segunda ejecuta el Actuador y la tercera el Registro de Mediciones.

Scheduling

Los módulos Actuador y Registro de Mediciones fueron implementados utilizando la librería `asyncio` y sus corrutinas fueron programadas de forma que detienen su ejecución periódicamente. No obstante, los módulos Configurador, Optimizador e Intérprete no detienen su ejecución hasta que finalizan.

Para que las tres tareas se puedan ejecutar de forma concurrente es necesario que el Configurador, Optimizador e Intérprete se detengan durante su ejecución. Esto puede realizarse de dos maneras distintas. Una opción es re-programar los módulos y utilizar la librería `asyncio` para agregar puntos de salida en donde la tarea detenga su ejecución voluntariamente. La otra es utilizar la librería `threading` que otorga la capacidad de implementar *scheduling apropiativo* en el cual el sistema operativo es el encargado de detener la ejecución de los módulos. Se optó por la última opción con el como forma de implementar la concurrencia.

Por más información sobre las librerías de Python antes mencionadas y concurrencia en Python en general favor dirigirse al Apéndice F.

Descripción de los hilos de ejecución

Tal como fue descrito el Nodo Controlador presenta tres tareas que debe ejecutarse en ‘simultáneo’. Se decidió que cada una de ellas sea ejecutada en un *thread*, o hilo de ejecución, distinto. En la Figura 5.5 se aprecia el diagrama de los hilos de ejecución implementados.

Thread: Optimización

Este *thread* es el encargado de ejecutar una vez al día las funcionalidades de los módulos Configurador, Optimización e Intérprete.

Antes de que el proceso de optimización pueda comenzar resulta imprescindible que todas las variables del problema se encuentren disponibles. Es decir, la empresa suministradora debe haber publicado el vector de precios para el día siguiente, y el usuario debe haber programado las tareas de los electrodomésticos según sus preferencias. Se supone que existe un acuerdo con la empresa suministradora y el usuario referente a que dicha información debe estar disponible a partir de una determinada hora, a la cual se le denomina hora de optimización. En el diagrama de la Figura 5.5 se asumió que la hora de optimización es las 23:00hs.

El *thread* Optimización aguarda a que sea la hora de optimización para ejecutar luego las funcionalidades del Configurador, Optimizador e Intérprete en forma encadenada. Habiendo

5.9. Programa principal

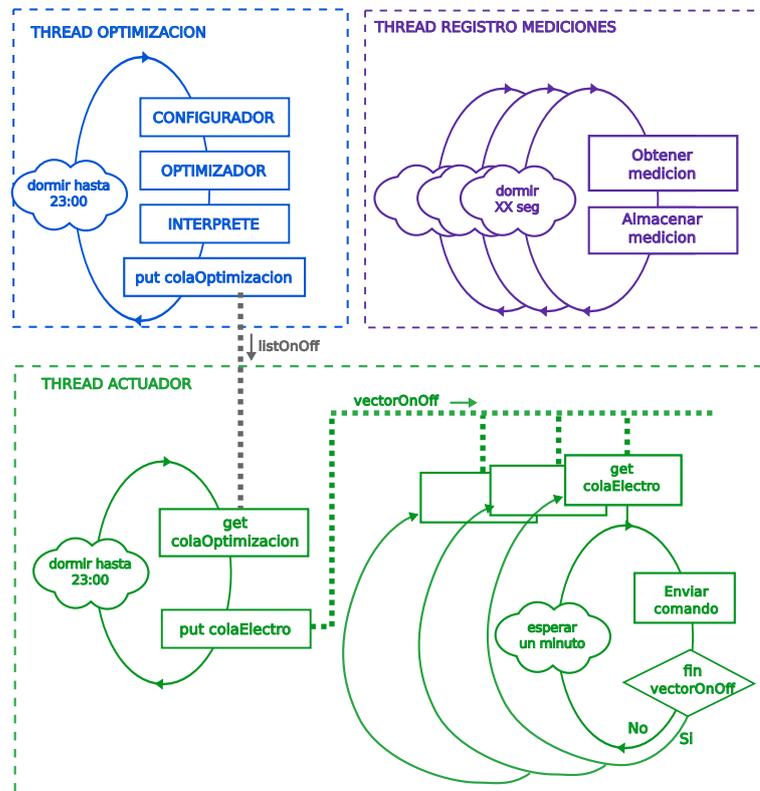


Figura 5.5: Diagrama de flujo de los distintos hilos de ejecución.

el Intérprete devuelto la lista `listOnOff`, se procede a pasar el resultado al `threadActuador`. Para la comunicación entre *threads* se utilizó una instancia de la clase `queue.queue`, la cual implementa una cola que permite la comunicación segura entre *threads*. La clase `queue.queue` se describe con mayor detalle en el Apéndice F.

Para construir el *thread* Optimizacion, en primer lugar se implementó la función `optimizacionLoop`. Esta ejecuta el proceso descrito en el párrafo anterior, a continuación se muestra un pseudo-código de la misma.

```
def optimizacionLoop(colaOptimizador):
    while True:
        time.sleep(hasta horaOptimizacion)
        ejecutar_configurador()
        q = ejecutar_optimizador()
        listOnOff = ejecutar_interprete(q)
        colaOptimizador.put(listOnOff)
```

Para crear un hilo de ejecución que invoque esta función, se debe crear una instancia de la clase `Thread` de la siguiente manera:

```
colaOptimizador = queue.Queue()
threadOptimizacion = threading.Thread(target=optimizacionLoop,
```

Capítulo 5. Implementación del Nodo Controlador

```
threadOptimizacion.start()           args=(colaOptimizador))
```

Thread: Actuador

Este *thread* es el responsable de ejecutar las funcionalidades del módulo Actuador. La función `actuador`, descrita en la Sección 5.7 de este capítulo, ya ejecuta todas las funcionalidades del módulo. Por tanto, para implementar este *thread* basta con crear una instancia de la clase `Thread` que invoque dicha función.

Thread: Registro Mediciones

Este *thread* ejecuta las funcionalidades del módulo Registro Mediciones. Al igual que en el Actuador, el módulo Registro de Mediciones presenta un función, llamada `registroMedidas`, que ejecuta todas las funcionalidades del módulo. Por ello, para implementar este *thread* basta con crear una instancia de la clase `Thread` que invoque dicha función.

Capítulo 6

Interfaz de usuario

El sistema HEMS, en especial el algoritmo de gestión, presentan un gran cantidad de parámetros que deben ser ajustado por el usuario de forma que el sistema se adapte a sus preferencias y hábitos de consumo. Con el fin de que el usuario pueda ajustar estos parámetros se diseñó e implementó una interfaz de usuario. El objetivo de este capítulo es describir la interfaz de usuario implementada en este proyecto. Se describen las herramientas utilizadas en el desarrollo de la interfaz, las funcionalidades que esta ofrece y los mecanismos de seguridad utilizados.

6.1. Descripción de la interfaz de usuario

Uno de los puntos relevantes del proyecto es el de la interacción que debe existir entre el usuario final y el Nodo Controlador. Es imprescindible permitir que el usuario pueda modificar los distintos parámetros del sistema con el propósito de adaptarlo a sus necesidades. Esto incluye cuáles son los horarios en los que puede operar un electrodoméstico, modificar la temperatura deseada, etc. Asimismo, se debe brindar información de interés para el usuario, como los son la configuración actual del sistema, cómo están agendados los electrodomésticos del día y la tarifa de precios.

En primer lugar, se evaluaron las posibles alternativas para la implementación de la interfaz de usuario, el detalle de este estudio se encuentra en Apéndice I. Se optó por desarrollar una interfaz web pensada para que el usuario pueda acceder desde cualquier explorador web, inclusive desde un teléfono móvil. En una primera instancia, este acceso será posible sólo desde una red hogareña. Pudiendo ser extendido en una instancia posterior para permitir el acceso a través de Internet punto que se trata más detalladamente en la Sección 8.4.

La interfaz web está basada en un módulo central instalado en el Nodo Controlador. Esta accede a la base de datos, también local al dispositivo, para actualizar las preferencias del usuario. Esta base de datos es la misma que utiliza el módulo Configurador del Nodo Controlador. Por mas detalle de la base de datos consultar la Sección 5.4

6.2. Herramientas utilizadas

En general, la interfaz se basó en el uso de herramientas *open source* gratuitas. La utilización de este tipo de herramientas es muy amplia a nivel mundial, por lo que cuentan con soporte, actualización continua y una comunidad que las respalda. Esto contribuye a facilitar el desarrollo

Capítulo 6. Interfaz de usuario

y a su vez evita costos extra. A a continuación se detallan las herramientas empleadas en el desarrollo de la interfaz web.

Implementación Interfaz de usuario basada en JSP y Servlets

Java, Java Server Pages (JSP) y Servlets

El lenguaje de programación *Java* es un lenguaje de alto nivel que se puede caracterizar por los siguientes atributos¹:

- sencillo,
- orientado a objetos,
- multiproceso,
- dinámico,
- portátil,
- de alto rendimiento,
- robusto,
- seguro.

Si bien *Java* cuenta con los múltiples beneficios mencionados, una de las principales características por las que esta tecnología fue elegida fue el hecho de ser multi-plataforma, lo cual permite ejecutar en cualquier sistema operativo. A las ventajas ya mencionadas, se suma una amplia colección de bibliotecas que son popularmente usadas por grandes compañías en producción. Otro punto fundamental, es que parte del equipo ya contaba con experiencia trabajando en esta tecnología.

La tecnología *Java Server Pages (JSP)*[39] permite desarrollar rápidamente y mantener fácilmente páginas Web dinámicas. Como parte de la familia de la tecnología *Java*, la tecnología *JSP* permite un rápido desarrollo de aplicaciones basadas en Web que son independientes de la plataforma. La tecnología *JSP* separa la interfaz de usuario de la generación de contenidos, permitiendo a los diseñadores cambiar el diseño general de la página sin alterar el contenido dinámico subyacente. Entre sus beneficios se destaca su facilidad de aprendizaje, su extensión y su fácil mantenimiento.

La tecnología *JSP* utiliza etiquetas similares a las de *XML* para encapsular la lógica que genera el contenido de la página. La lógica de la aplicación puede residir del lado del servidor. Al separar la lógica de la página de su diseño y visualización, y al contar con un diseño basado en componentes reutilizables, la tecnología *JSP* hace que sea más rápido y más fácil que nunca construir aplicaciones basadas en Web[40].

JSP es una extensión de la tecnología *Java Servlet*. Los *servlets* son módulos independientes de la plataforma del lado del servidor, que se ajustan a la perfección en un marco de servidor web y se pueden utilizar para ampliar las capacidades del mismo con un costo, mantenimiento y soporte mínimos. A diferencia de otros lenguajes de *script*, los *servlets* implican ninguna modificación específica de la plataforma. Son componentes de la aplicación que se descargan, a demanda, a la parte del sistema que los necesita. En conjunto, la tecnología *JSP* y *servlets* proporcionan una alternativa atractiva a otros tipos de programación dinámica, ofreciendo: independencia de la plataforma, rendimiento mejorado, separación de la lógica de la interfaz, facilidad de administración, extensibilidad y, lo más importante, la facilidad de uso.

¹Cada uno de estos atributos es explicado detalladamente por Gosling y McGilton en *The Java Language Environment*[38].

Apache Tomcat

Para la interfaz se decidió utilizar el servidor web *Apache Tomcat*[41]. Dado que *Tomcat* fue escrito en *Java*, funciona en cualquier sistema operativo que disponga de la máquina virtual *Java*.

Las principales ventajas de utilizar el servidor web *Tomcat* son las siguientes:

- Es Software Libre: como se mencionó anteriormente este es uno de los requisitos buscado en las herramientas a utilizar.
- Es liviano: *Tomcat* sólo ofrece la funcionalidades más básicas necesarias para ejecutar un servidor, lo que significa que los tiempos de carga y deploy son relativamente rápidos en comparación con otros servidores más complejos. Esta naturaleza liviana también le permite contar con un ciclo de desarrollo mucho más rápido. En este caso no se precisaba contar con un servidor con características avanzadas, sino que por el contrario se buscaba un medio rápido y fácil, por esta razón se optó por utilizar *Tomcat*.

Aplicación *Standalone Java* usando *Tomcat* embebido

La manera tradicional de empaquetar aplicaciones *Java* que están destinados a funcionar dentro de un contenedor es agruparlos como un archivo *WAR*. Este es un archivo comprimido con una estructura de directorios estándar, que contiene todas las bibliotecas y dependencias a nivel de aplicación necesarias en tiempo de ejecución.

Sin embargo, existe otra opción que implica incorporar el contenedor embebido en la aplicación misma. Es decir, en vez de desplegar la aplicación dentro de un contenedor, el contenedor está incorporado y es desplegado dentro de la propia aplicación. Este tipo de aplicación se empaqueta para su ejecución en línea de comandos como cualquier aplicación normal. De esta forma, la aplicación puede ser distribuida como un único ejecutable.

Se evaluó la posibilidad de usar un contenedor de *Servlets* preinstalado en la placa *Raspberry Pi 2*. Sin embargo, también se tenía la opción de embeber un servidor web dentro de la aplicación misma. La gran ventaja de esta última alternativa es que no hay necesidad de tener preinstalado ningún servidor lo cual puede resultar tedioso para desplegar una única aplicación. Esto también ahorra costos de instalación y mantenimiento. Tomando en cuenta lo antes expuesto es que se decidió utilizar la opción embebida.

Maven

Maven[42] es una herramienta de software para la gestión y construcción de proyectos *Java* de *Apache Software Foundation*. Esta tiene un modelo de configuración de construcción simple, basado en un formato XML.

Maven utiliza un *Project Object Model (POM)* para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Hibernate

Hibernate[43] es una herramienta de *Mapeo Objeto-Relacional (ORM)* para la plataforma *Java* que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones las entidades que permiten establecer estas relaciones. *Hibernate* es software libre distribuido bajo

Capítulo 6. Interfaz de usuario

los términos de la licencia *GNU LGPL (GNU Lesser General Public License)*, que permite el uso en aplicaciones propietarias.

JFreeChart

Fueron utilizadas las funcionalidades proporcionadas por *JFreeChart*[44] para mostrar diversos datos que pueden ser de interés para el usuario como, por ejemplo, información de precios y consumo. *JFreeChart* es una biblioteca *Java* gratuita que sirve para mostrar gráficas de distintos tipos. Se decidió utilizar esta biblioteca debido a las múltiples ventajas que esta presenta:

- es una biblioteca liviana con pocas dependencias;
- es software libre y se distribuye bajo los términos de la licencia *GNU LGPL*;
- presenta una API consistente y bien documentada;
- soporta muchos tipos de salidas incluyendo: componentes *Swing* y *JavaFX*, archivos de imagen (como PNG y JPEG) y archivos de gráficos vectoriales (como PDF, EPS y SVG);
- sus gráficos pueden ser incorporados fácilmente en servlets u otras aplicaciones.

6.3. Funcionalidades de la Interfaz Web

La Interfaz Web permite que el usuario pueda controlar el funcionamiento del sistema HEMS de forma sencilla a través de una página web. La interfaz cuenta con múltiples menús que permiten al usuario acceder a las distintas funcionalidades que ésta presenta. A continuación se enumeran estas funcionalidades. Por más información acerca de la interfaz de usuario y sus prestaciones favor dirigirse al Apéndice J.

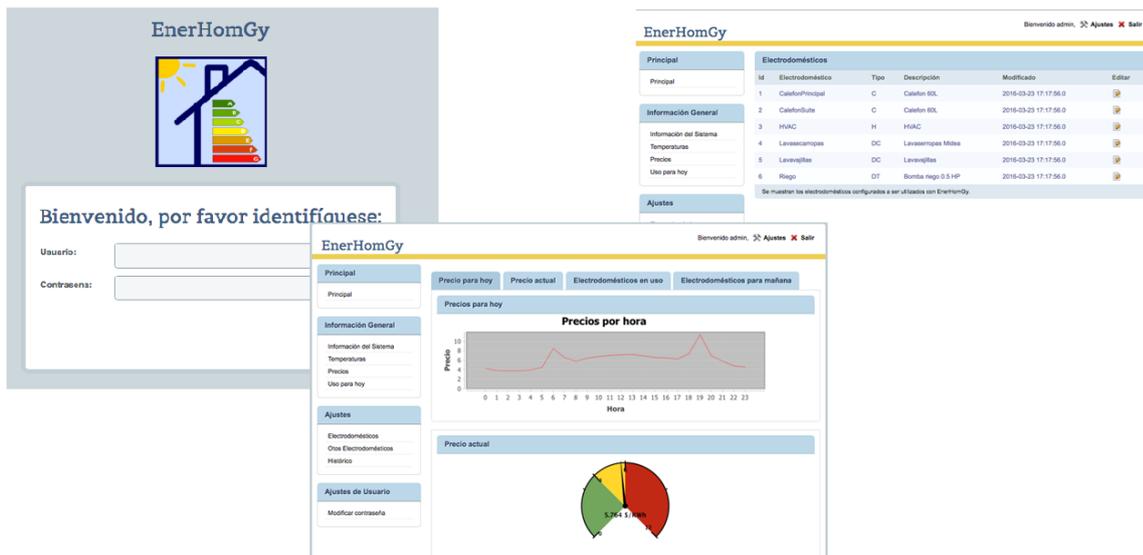


Figura 6.1: Captura de pantallas de Interfaz Web

Autenticación de usuario

El usuario debe autenticarse para poder acceder a algunas de las funcionalidades provistas por el sistema. El mecanismo de autenticación empleado es descrito más adelante en este capítulo en la Sección 6.4.

Modificar contraseña

Se provee una pantalla para cambiar la contraseña en caso de que usuario lo desee.

Consulta de tarifas y temperaturas

Bajo la premisa de que cuanto más informado esté el usuario más eficiente será el uso del sistema, éste puede verificar los datos que se detallan a seguir.

Ver tarifas actuales

Permite consultar las distintas tarifas que ofrece la empresa suministradora ya sea en ese instante o en las futuras horas. La información se despliega en formato de gráfico indicando precio máximo y mínimo para ese día.

Ver temperaturas

Muestra en una gráfica por horas la temperatura ambiente para el día actual así como también para el día siguiente. Además se cuenta con la posibilidad de consultar un promedio mensual estimado de la temperatura del agua provista por el suministrador de agua potable.

Electrodomésticos

Para que el usuario pueda modificar los distintos parámetros de uso de los electrodomésticos según sus preferencias se proveen varias funcionalidades de consulta y modificación.

Ver estado de los electrodomésticos

Permite consultar en qué horas el algoritmo decidió encender cada uno de los electrodomésticos para el día actual.

Modificar parámetros

Los electrodomésticos controlador por el sistema presentan una serie de parámetros que el usuario puede modificar. En todos los electrodomésticos se permite que el configure qué días de la semana desea que el electrodoméstico se active. Adicionalmente, cada uno de los tipos de electrodoméstico, definidos en el Capítulo 4, presenta parámetros específicos que el usuario puede ajustar.

Aire acondicionado:

En el aire acondicionado el usuario puede configurar para cada día de la semana el período de interés, es decir las horas en que desea la habitación esté aclimatada. Asimismo, el usuario tiene la posibilidad de ajustar la temperatura de confort así como las temperaturas máxima y mínima que admite para cada hora del día.

Calefón:

En el caso del calefón es posible configurar el período de interés, es decir el rango horario en el cual el usuario desea contar con el agua caliente para poder hacer uso de la misma. Por otra parte, también es posible configurar para cada día de la semana la temperatura de confort y la temperatura mínima del agua.

Electrodomésticos de consumo discreto con tareas de duración fija:

Para este tipo de electrodomésticos basta con que el usuario configure el rango horario en el

Capítulo 6. Interfaz de usuario

cual permite que se ejecute la tarea del electrodoméstico. El usuario también debe configurar los días en lo que desea que electrodoméstico se active, como se describió anteriormente.

Electrodomésticos de consumo discreto con tareas de duración flexible:

En estos electrodomésticos el usuario puede elegir para cada día el rango horario en que permite que este se encienda. Además es posible ajustar un tiempo mínimo y máximo de funcionamiento del electrodoméstico durante el día.

Uso de otros electrodomésticos

Con el fin de que el sistema realice una mejor optimización del uso de la energía eléctrica, existe una sección para configurar durante qué horas se hace uso de los demás electrodomésticos del hogar que no son controladas por el sistema. Esta información es utilizada para calcular la restricción de energía máxima que puede agenda el sistema HEMS en cada hora, ver Sección 4.6.

Histórico e indicadores

Se proveen una serie de tablas y gráficos, implementados con *JFreeChart*, que permiten al usuario evaluar el uso de los electrodomésticos y el funcionamiento del sistema. Se muestra el consumo de los electrodomésticos controlados para diferentes periodos de tiempo. Asimismo, también se realiza una comparación con lo que sería el gasto de los electrodomésticos en caso de encontrarse con tarifa fija *FP*, como la de la Sección 8.2, en lugar de una *DAP*. Esta información permite que el usuario sea consciente de cómo está funcionando el sistema y pueda programar el comportamiento del HEMS de forma más eficiente en el futuro.

6.4. Seguridad en la Interfaz Web



Figura 6.2: Histórico de Gastos

6.4. Seguridad en la Interfaz Web

En esta sección se describen los mecanismos de seguridad utilizados en la Interfaz Web desarrollada y se justifica la elección de los mismos.

Mecanismos de Autenticación

Los mecanismos de autenticación aseguran que el usuario se haya autenticado antes de concederle el acceso a cualquier recurso que esté limitado por alguna restricción de seguridad. Puede haber varias restricciones de seguridad que se aplican a múltiples recursos, pero el mismo método de autenticación se aplica a todos los recursos limitados en una aplicación.

Un mecanismo de autenticación de usuario específica:

- la forma en que un usuario obtiene acceso a los contenidos;
- el ámbito en el que será autenticado el usuario;
- atributos adicionales.

Capítulo 6. Interfaz de usuario

Niveles

Las *JSP* y los *Servlets* proveen varios mecanismos que permiten proporcionar seguridad a una aplicación. Existen varios niveles de seguridad que pueden implementarse, que van desde la autenticación básica usando identificadores y contraseñas, hasta métodos sofisticados que utilizan certificados de autenticación.

Autenticación básica

En este método, el servidor utiliza el nombre de usuario y la contraseña proporcionada por el cliente para autenticar al usuario. El servidor permite varios intentos erróneos (por ejemplo hasta 3 intentos) después de los cuales se despliega una página de error (normalmente un error HTTP 401 No autorizado).

Autenticación con digest

Es similar a la autenticación básica, pero en este las contraseñas se cifran utilizando la fórmula *Hash*.

Autenticación con certificados

Se requiere que cada cliente que accede al recurso tenga que enviar un certificado para autenticarse a sí mismo. Esto requiere del protocolo SSL.

Autenticación basada en formularios

Para este método se debe proporcionar una página de *login* para desplegar al usuario y que este pueda ingresar su nombre de usuario y contraseña.

Selección

Se debe reconocer que la autenticación básica *HTTP* y la autenticación basada en formularios no son mecanismos de autenticación extremadamente seguros. La autenticación básica envía los nombres de usuario y contraseñas a través de Internet como texto codificado en *Base64* mientras que la autenticación basada en formularios envía estos datos como texto sin formato. En ambos casos, el servidor de destino no está autenticado.

Si bien se entiende que esto implica una vulnerabilidad ya que si alguien puede interceptar la transmisión, el nombre de usuario y contraseña pueden ser decodificados, para este tipo de aplicación en particular no se considera tan importante ya que nadie podría obtener un beneficio económico de la obtención de esos datos. Además, en primera instancia la interfaz de usuario está pensada para ser utilizada dentro de una red LAN hogareña, sin acceso remoto a la misma a través de Internet. En conclusión, dado que para este caso no se requiere de un nivel de seguridad elevado fueron descartados los métodos más complejos optando por utilizar una autenticación básica basada en formularios.

Implementación

Uno de los objetivos buscados durante el diseño de la interfaz de usuarios es que esta sea práctica y simple. Por esta razón, se restringió el acceso sólo a los recursos de la Interfaz Web en los que es posible modificar el funcionamiento del sistema. En las páginas informativas no aplica el requerimiento de autenticación de usuario, estas son las páginas en las cuales se brinda información acerca del estado del sistema, históricos de gastos o predicciones para el futuro.

Como se mencionó anteriormente se decidió utilizar autenticación basada en formularios la cuál requiere proporcionar un formulario de login que se despliegue para solicitar usuario y contraseña. Dicho formulario debe contar con algunas características particulares:

6.4. Seguridad en la Interfaz Web

- Contener un elemento `j_username` donde se ingresa el nombre de usuario.
- Contener un elemento `j_password` donde se ingresa la contraseña.
- La acción en el tag `<form>` debe ser `j_security_check`.

Cuando el usuario quiere acceder a algún recurso restringido se despliega el formulario de login solicitando que se ingresen el nombre de usuario y contraseña. Cuando el contenedor detecta que la acción es `j_security_check`, el mismo utiliza algún mecanismo interno para efectuar la autenticación.

Para validar el usuario el contenedor, Tomcat en este caso, debe determinar cuál es el *Security Realm* configurado. Un *Security Realm* se compone de un conjunto de proveedores de seguridad configurados, usuarios, grupos, roles y políticas de seguridad. Para este proyecto utilizamos como *Realm* la clase *DataSourceRealm* de *Apache Foundation*[45]. La misma es una implementación de *Realm* que funciona con cualquier fuente de datos *JDBC JNDI*.

Una vez determinado el *Security Realm* se utiliza el mismo en la autenticación. El nombre de usuario y contraseña proporcionados se usarán como parámetros para invocar al método `public java.security.Principal authenticate(String username, String credentials)` de la clase *DataSourceRealm*.

Si el usuario es autenticado, el contenedor recupera los roles de seguridad para ese usuario y valida si este pertenece a alguno de los roles habilitados para el recurso al que se desea acceder. En caso contrario, se muestra un mensaje de error indicando que el usuario no está habilitado para acceder al recurso.

Capítulo 7

Prueba Experimental

En el Capítulo 4 se describió el algoritmo de gestión utilizado y cómo se modelaron los electrodomésticos del hogar. Luego en el Capítulo 5 se explicó cómo se implementó el Nodo Controlador, el cual ejecuta el algoritmo de gestión y envía las órdenes correspondientes a los electrodomésticos. Una vez implementado todo esto, se procedió a llevar a cabo una prueba experimental para verificar el correcto funcionamiento de lo antedicho.

Para la prueba experimental se instaló el sistema HEMS en un ‘hogar laboratorio’ compuesto por un calefón y un lavasecarropas. El experimento realizado consistió en dejar funcionando el sistema durante dos días completos de forma autónoma, y luego verificar el cumplimiento de una serie de requisitos establecidos.

En este capítulo se describe la prueba experimental realizada y el trabajo previo necesario para realizar dicha prueba. Primero se describe la instalación del laboratorio detallando los electrodomésticos utilizados y las modificaciones necesarias. Luego se describen los Nodos de Carga implementados para operar los electrodomésticos. Finalmente se detalla el experimento realizado y los resultados obtenidos.

7.1. Objetivos

La prueba experimental realizada permitió verificar el correcto funcionamiento del Nodo Controlador implementado en este proyecto. Los objetivos específicos que se buscaron en la misma fueron los siguientes:

- Observar el funcionamiento del Nodo Controlador en régimen de uso continuo, dejando el sistema en funcionamiento ininterrumpido durante más de un día.
- Evaluar la robustez del sistema HEMS implementado.
- Verificar el correcto accionamiento de los electrodomésticos de acuerdo a lo determinado por los módulos Optimizador e Intérprete del Nodo Controlador.
- Validar los modelos de electrodomésticos utilizados. En particular, comparar el modelado de evolución de temperatura con la evolución real.
- Verificar el cumplimiento de los criterios de confort configurados por el usuario.

7.2. Laboratorio de pruebas

Uno de los mayores desafíos que presentó la prueba experimental fue la instalación del ‘hogar laboratorio’ para realizar las pruebas. Para ello fue necesario contar con un local físico donde efectuar la prueba, adecuar la instalación eléctrica del local y adquirir e instalar los electrodomésticos a utilizar.

Electrodomésticos utilizados

Para realizar las prueba experimental se seleccionaron dos electrodomésticos representativos de un hogar: un calefón y un lavasecarropas¹. En la Tabla 7.1 se muestran las principales características de los electrodomésticos.

Calefón		Lavasecarropas	
Marca	Orion	Marca	Midea
Volumen	45 lts.	Capacidad	7 kg.
Potencia nominal	1000 W	Potencia pico	2500 W
Temperatura máxima	63 °C		

Tabla 7.1: Características de los electrodomésticos utilizados en la prueba experimental

Instalación eléctrica

El Instituto de Ingeniería Eléctrica permitió hacer uso de su Laboratorio de Proyectos de Fin de Carrera como sitio para ejecutar el experimento. Para realizar la prueba fue necesario modificar la instalación eléctrica del local con el fin de poder energizar todos los elementos del sistema HEMS, incluyendo los electrodomésticos.

Los elementos de la instalación eléctrica que fue necesario agregar son los siguientes:

- Para la conexión del calefón:
 - Tomacorriente 10 A para la alimentación del electrodoméstico.
 - Tomacorriente 10 A para la alimentación del Nodo Carga.
- Para la conexión del lavasecarropas:
 - Tomacorriente 16 A para la alimentación del electrodoméstico.
 - Tomacorriente 10 A para la alimentación del Nodo Carga.
- Para la conexión del Aire acondicionado²:
 - Tomacorriente 16 A para la alimentación del electrodoméstico.
 - Tomacorriente 10 A para la alimentación del Nodo de Carga de medición.
 - Tomacorriente 10 A para la alimentación del Nodo de Carga de comando.
- Protecciones para todos los elementos agregados.

¹Inicialmente se pensaba instalar un lavarropas ya que son más comunes de encontrar en un hogar uruguayo. Sin embargo, se optó por un lavasecarropas por ser más práctico de usar en un laboratorio.

²Se había previsto utilizar un aire acondicionado en la prueba experimental. Luego en revisiones del avance del proyecto se decidió dejar el aire acondicionado por fuera de la prueba experimental.

7.2. Laboratorio de pruebas

Con el fin de no afectar la instalación eléctrica existente, se decidió agregar un nuevo tablero (TD_EHG) derivado del tablero general del laboratorio y agregar todos los nuevos elementos debajo de ese tablero. En la Figura 7.1 se puede apreciar un diagrama unifilar del nuevo tablero.

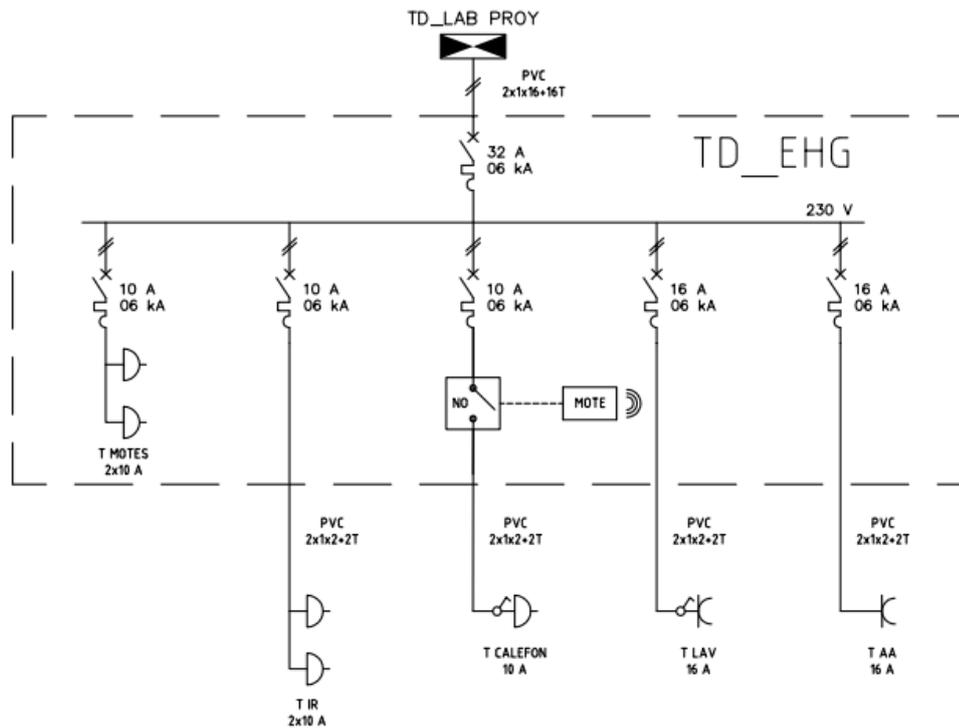


Figura 7.1: Unifilar del nuevo tablero instalado (TD_EHG)

Instalación de los equipos

El sector de Plan de Obras de la Facultad de Ingeniería ejecutó los trabajos de montaje de los electrodomésticos, instalación de las canalizaciones de agua e instalación eléctrica de acuerdo a los planos elaborados por el grupo de proyecto. En la Figura 7.2 se encuentra una foto del 'hogar laboratorio'.

Capítulo 7. Prueba Experimental



Figura 7.2: Equipamiento e instalación eléctrica del laboratorio de pruebas

7.3. Nodos de Carga

Los Nodos de Carga son los intermediarios entre el Nodo Controlador y los electrodomésticos. Su función es ejecutar órdenes y reportar mediciones al Nodo Controlador. Este proyecto no está enfocado en la implementación de Nodos de Carga. Sin embargo, para ejecutar la prueba experimental fue necesario implementar Nodos de Cargas básicos para comandar los electrodomésticos y medir las magnitudes de interés. En esta sección se describen brevemente los Nodos de Carga implementados. En el Apéndice L se realiza una descripción más detallada de los mismos.

Es importante destacar que los Nodos de Carga implementados en este proyecto fueron diseñados con el fin de poder ejecutar la prueba experimental. En consecuencia, no se buscaron soluciones genéricas ni eficientes, sino soluciones prácticas que funcionaran con los

electrodomésticos de dicha prueba.

Nodos de Carga utilizados

Cada electrodoméstico tiene su forma particular de ser comandado. Por tanto, fue necesario diseñar Nodos de Carga específicos para comandar cada uno de electrodomésticos de la prueba experimental. Es decir, se implementó un Nodo de Carga para comandar el calefón y otro para comandar el lavasecarropas. Adicionalmente, se deseaba medir la temperatura del agua del calefón durante el experimento por lo que se implementó un Nodo de Carga de medición para ese electrodoméstico.

Se recuerda que, tal como se describió en la Sección 3.4, los Nodos de Carga están compuestos por un mote Z1 de Zolertia que operan un actuador y/o un sensor.

Nodo de comando para el calefón

El Nodo Controlador puede enviar a los nodos de comando órdenes de ‘ON’ y ‘OFF’. Para comandar el calefón se diseñó un Nodo de Carga con un relé que controla la alimentación del calefón. Cuando el Nodo de Carga recibe una señal de ‘ON’ se acciona el relé para que cierre el circuito y se le da energía al calefón. En cambio, cuando recibe una señal del ‘OFF’ el relé abre el circuito de potencia del calefón.

Inicialmente se utilizó un relé electromecánico para comandar el calefón. Sin embargo, se descubrió que el ruido electromagnético que produce el relé al conmutar genera una falla en el sistema de comunicación del mote Z1. Esta falla hacía que se perdiera la comunicación con el Nodo Controlador, siendo necesario resetear el microprocesador para re-establecer la comunicación. Esta misma falla ya había sido reportada por el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1].

Para mitigar el problema se tomaron dos medidas. En primer lugar, se eliminó la fuente de ruido cambiando el relé por uno de estado sólido. Se utilizó un relé de estado sólido tipo *Zero-Crossing trigger* modelo *SSR-40 DA*, ver Figura 7.3. En segundo lugar, se implementó un algoritmo que detecta la falla y resetea el mote. Por más información acerca de la falla y del algoritmo generado revisar el Apéndice L.



Figura 7.3: Relé de estado sólido utilizado

Nodo de comando para lavasecarropas

A diferencia del calefón el lavasecarropas no puede ser comandado mediante su alimentación. Esto se debe a que el equipo no se repone correctamente luego de un corte de energía. En caso que se corte la alimentación del equipo, el usuario debe reiniciar manualmente el proceso que se encontraba en ejecución.

Capítulo 7. Prueba Experimental

El lavasecarropas utilizado presenta un botón de inicio/pausa, el cual permite pausar y reanudar el lavado en cualquiera de sus etapas. Se diseñó un Nodo de Carga capaz de emular el accionamiento del botón. Para esto se colocó el contacto seco de un relé en paralelo con el botón de inicio/pausa. El Nodo de Carga acciona el relé de forma de general pulso similares a los que genera el usuario cuando presiona el botón.

Nodo de medición calefón

El nodo de medición del calefón es el encargado de medir la temperatura del agua y reportarla al Nodo Controlador. Las mediciones de temperatura permiten validar el modelo térmico propuesto para el calefón³ y verificar que la temperatura del agua se mantiene dentro del rango configurado por el usuario.

Para medir la temperatura del agua dentro del tanque se colocó un sensor de temperatura en la pared exterior del tanque de cobre. Se utilizó módulo *MOD-1018* del fabricante *Embedded Adventures*, el cual cuenta con un sensor de temperatura y humedad *SHT25* de *Sensiron*, ver Figura 7.4.

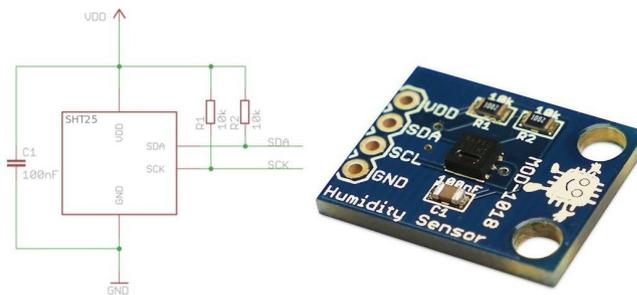


Figura 7.4: Módulo *MOD-1018* utilizado para medir la temperatura del agua del calefón

7.4. Experimento realizado

Una vez instalado el ‘hogar laboratorio’ se procedió a llevar a cabo la prueba experimental. En esta sección se describe el experimento realizado y los resultados obtenidos.

Descripción del experimento

El objetivo del experimento es evaluar en el ‘hogar laboratorio’ el desempeño del sistema HEMS implementado. El experimento consistió en dejar corriendo el sistema HEMS durante dos días consecutivos. Esto permitió corroborar que el sistema realiza correctamente la transición entre dos días, actualizando todas las variables y re-ejecutando el algoritmo correctamente.

El sistema HEMS fue configurado de la siguiente manera:⁴

- **Día 1:** (20 de Marzo de 2016)
 - **Calefón:** Se configuró que el usuario desea darse dos baños en el día, el primero entre las 06:30hs y las 08:30hs y el segundo entre las 16:00hs y las 19:30hs. Dentro

³Ver Apéndice B: Detalle del modelado de los electrodomésticos

⁴Se utilizaron timeslots de treinta minutos tanto para el calefón como para el lavasecarropas

7.4. Experimento realizado

de esos horarios se configuró que la temperatura del agua caliente debe ser superior a los 40 °C con una temperatura de confort de 50 °C.

- Lavasecarropas: Se configura para realizar un lavado en el día entre las 07:00 hs y las 20:00 hs.
- **Día 2:** (21 de Marzo de 2016)
- Calefón: Se configuró para realizar un baño en el día entre las 16:30 hs y las 20:00 hs. Se usó la misma configuración de temperatura.
 - Lavasecarropas: Se configura para realizar un lavado en el día entre las 07:00 hs y las 20:00 hs.

Se utilizaron dos curvas de precios publicadas por la empresa suministradora estadounidense *ComEd* ajustando su precio a la realidad uruguaya⁵. En la Figura 7.5 se aprecian las curvas de precios utilizadas, junto con el precio de la tarifa plana actual⁶.

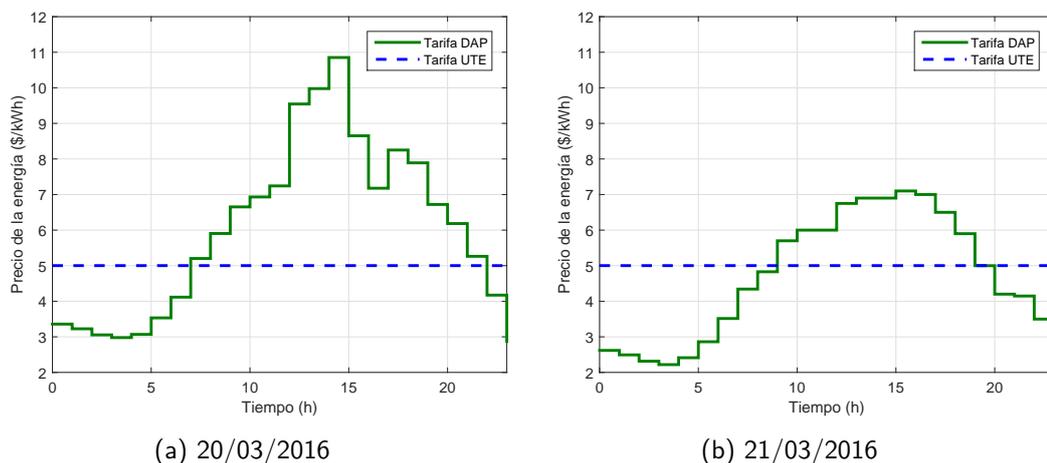


Figura 7.5: Precios de la energía utilizados en la prueba experimental

Fue diseñado un protocolo de ensayos para registrar los resultados más relevantes del experimento, los cuales se encuentran en el Apéndice M. Para la inspección de la prueba experimental se concurrió al laboratorio en las horas en las cuales se debían accionar los electrodomésticos, por ejemplo a extraer agua del calefón simulando una ducha. Adicionalmente, se tomaron fotos cada 30 segundos durante ambos días para observar el funcionamiento en todo momento. Se anexa al documento el vídeo elaborado con estas imágenes para facilitar su visualización.

Primer día del experimento

A continuación se encuentran los resultados del proceso de optimización para el primer día de las pruebas junto con un análisis del desempeño del sistema.

⁵Por más información acerca de esta curva de precios, consultar la Sección 8.1

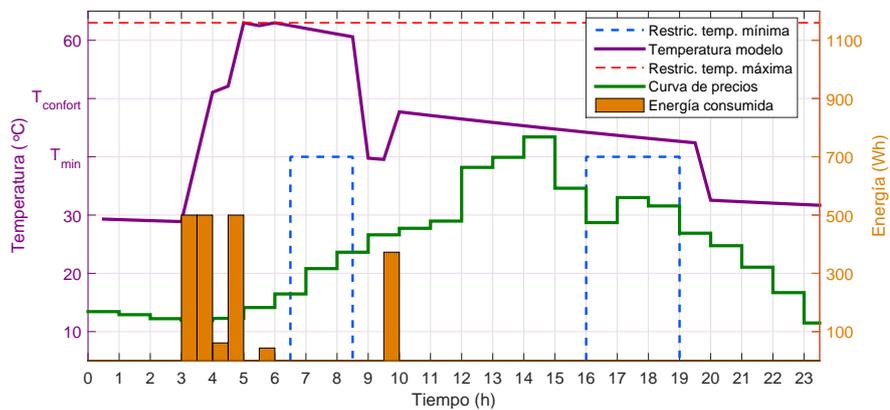
⁶Más precisamente, el valor del precio del escalón entre 101 kWh y 600 kWh de la tarifa de un consumidor residencial simple en pliego tarifario de UTE vigente en el año 2015.

Capítulo 7. Prueba Experimental

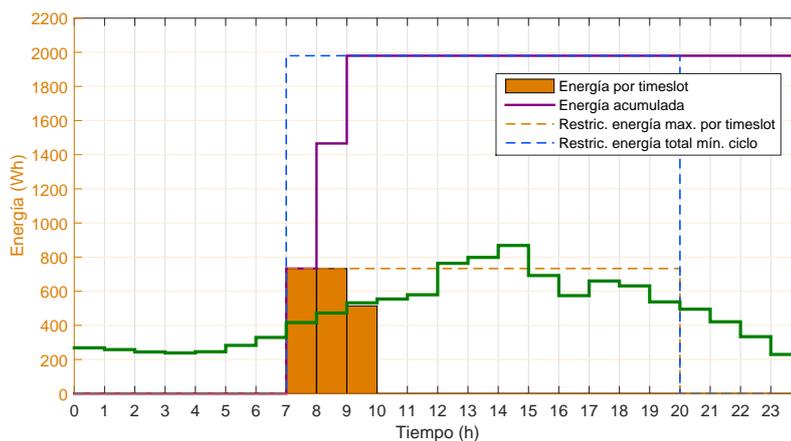
Resultado optimización

El algoritmo de gestión del primer día se ejecutó a las 23:40 horas del día previo (19/03/2016). El proceso de optimización convergió a una solución en 198 segundos. Los resultados de la optimización de ambos electrodomésticos se presentan en forma gráfica en la Figura 7.6. En primer lugar, se observa que el lavado del lavasecarropas se agenda lo más temprano posible dentro del período de interés, de forma de aprovechar las horas más baratas de consumo.

En el caso del calefón, el algoritmo determinó que lo más conveniente es aprovechar al máximo las dos horas más baratas del día (3am y 4am), llevando el agua del calefón al máximo de temperatura admitido. De esta manera, se minimiza la energía que se debe entregar en el período entre los dos baños. La temperatura resultante para el primer baño es superior a la temperatura de confort y la temperatura del segundo baño se encuentra entre la mínima y la de confort.



(a) Calefón



(b) Lavasecarropas

Figura 7.6: Resultados de la optimización del día 20 de Marzo

7.4. Experimento realizado

Ejecución en tiempo real

La prueba se ejecutó sin problemas de comunicación, con una correcta ejecución de todos los comandos enviados por el módulo **Actuador**. El protocolo de este ensayo del día se puede consultar en el Apéndice M.

El lavasecarropas ejecutó su tarea sin interrupciones, teniendo la ropa lavada y seca a partir de las 9:51. El Nodo Controlador predijo que el lavado finalizará a las 10:00. Esta diferencia de tiempo se debe a que estableció un margen de seguridad en la curva de consumo del electrodoméstico para asegurar que el lavado finalice.

Respecto al calefón, el mismo también se comportó de acuerdo a lo esperado. Fue efectuada una primera extracción de 25 litros de agua caliente a las 7:40. Dicho volumen de agua simula la cantidad de agua caliente que el usuario utilizaría en una ducha de 10 minutos, considerando las temperaturas que tenían en ese momento el agua fría de la línea de OSE y el agua caliente del calefón.

En la Figura 7.7 se encuentra la comparación entre la temperatura real y la predicción del modelo térmico. Se puede apreciar que la predicción es muy buena cuando el calefón funciona libremente, es decir, cuando está apagado y no se extrae agua del tanque. Asimismo, el modelo térmico también predice correctamente la evolución de la temperatura cuando se le entrega energía al calefón. Sin embargo, el modelo térmico falla en las extracciones de agua.

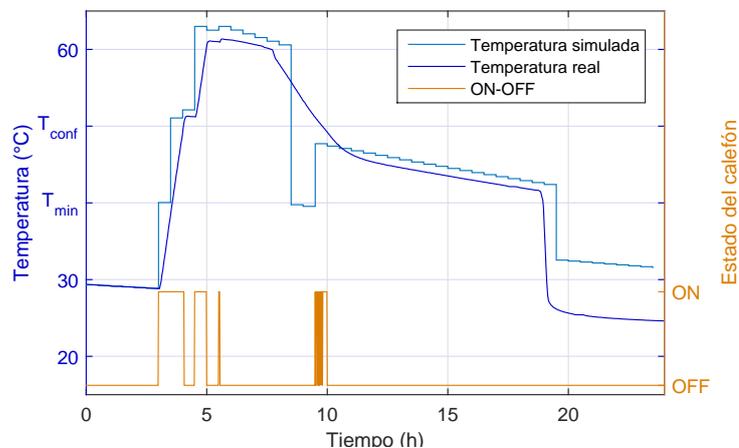


Figura 7.7: Comparación de temperatura real y simulada

Existen dos razones por las cuales el modelo no se ajustó correctamente a la realidad luego de las extracciones de agua. En primer lugar, se recuerda que el modelo térmico asume que la extracción se realiza instantáneamente al final del pulso útil. Esto lleva a que la predicción no sea correcta en los periodos transitorios que se producen durante la ducha. En segundo lugar, el modelo térmico no conoce con exactitud el volumen de agua extraído durante una ducha. Esta es la razón por la cual la predicción del modelo y la temperatura real presenta una diferencia constante durante las últimas horas del día. El volumen de agua utilizado durante una ducha depende en gran medida del usuario por lo que es muy difícil predecirlo correctamente. Para corregir esta fuente de error se realimenta la última medición de temperatura como se explica en la Sección 5.4.

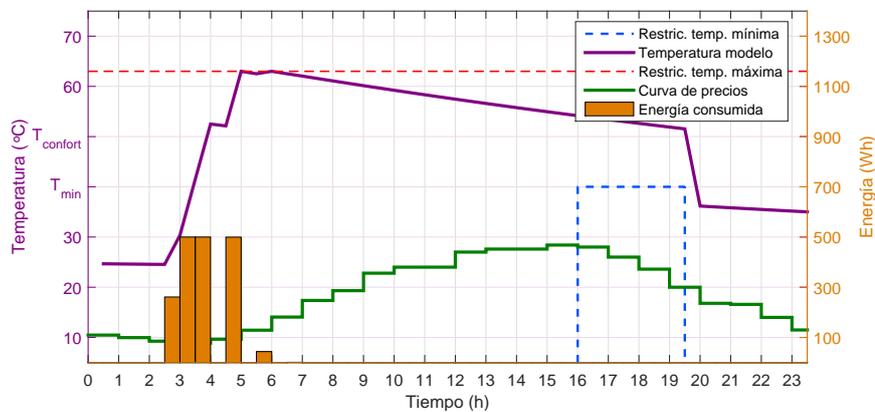
Capítulo 7. Prueba Experimental

Segundo día del experimento

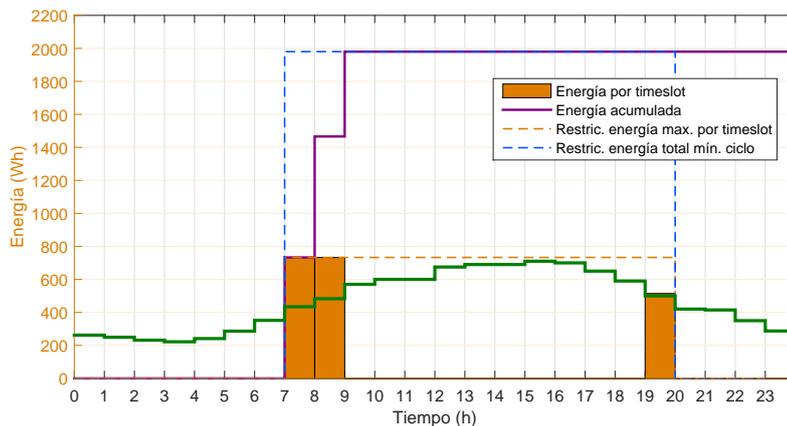
En esta sección se muestran los resultados del proceso de optimización del segundo día del experimento y un análisis del desempeño del sistema de gestión.

Resultado optimización

El proceso de optimización del segundo día se ejecutó a las 23:40 horas del primer día y convergió a la solución en 178 segundos. El último registro de temperatura del calefón, previo a la ejecución de la optimización, fue a las 23:38 con un valor de 26,65 °C. Se corroboró que efectivamente este fue el valor de temperatura utilizado como condición inicial en el algoritmo de optimización. En la Figura 7.8 se muestra el resultado del proceso de optimización.



(a) Calefón



(b) Lavasecarropas

Figura 7.8: Resultados de la optimización del día 21 de Marzo

El algoritmo de gestión decidió nuevamente calentar el agua en la mañana, que son las horas de energía más baratas. Este día tuvo un precio medio menor que el anterior, por esta razón el algoritmo permite tener una temperatura más alta al final del día. El lavasecarropas

7.5. Evaluación de la prueba experimental

tuvo dos períodos de funcionamiento, se decide iniciar el lavado al comienzo del pulso útil de 7:00 a 9:00 y culminarlo a las 19:00. La explicación para ello, es que el precio a las 19:00 es inferior que a las 9:00, por tanto es más económico consumir en la última hora del pulso útil.

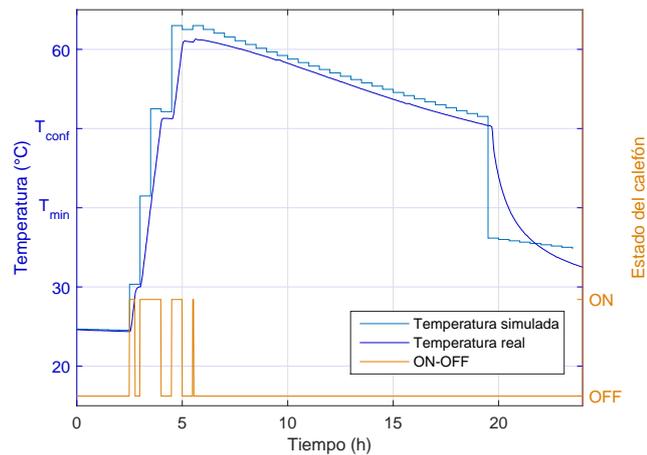


Figura 7.9: Comparación de temperatura real y simulada en la prueba experimental del día 2

Ejecución en tiempo real

La prueba se ejecutó con normalidad, sin pérdidas de comunicación ni mal funcionamiento de los Nodos de Carga. En cuanto al calefón, este se comportó nuevamente como estaba previsto, ver Figura 7.9.

Por otra parte, el lavasecarropas finalizó su programa de lavado a las 09:51 cuando estaba planificado que se pausara a las 9:00hs y continuara el proceso en la tarde. Habiendo revisado el historial de comandos que envió el nodo, se verificó que el Nodo Controlador efectivamente envió la orden de pausado y que la misma fuera recibida por el Nodo de Carga. En pruebas posteriores se constató que el lavasecarropas no es interrumpible durante el ciclo de secado, e ignora el botón de inicio/pausa durante este ciclo. En conclusión, el Nodo Controlador no pudo pausar el lavasecarropas a las 09:00 porque el mismo se encontraba en su ciclo de secado. Se entiende que esto no fue una falla del sistema de gestión sino que el lavasecarropas no se comportó de acuerdo lo esperado.

7.5. Evaluación de la prueba experimental

La prueba experimental permitió evaluar el funcionamiento en tiempo real del sistema de gestión diseñado, y esta arrojó resultados muy positivos.

En primer lugar, se observa que las decisiones del algoritmo de gestión fueron correctas y se cumplieron los criterios de confort del usuario. El lavasecarropas culminó el lavado dentro del intervalo definido, a excepción del caso en que el proceso no pudo ser interrumpido por el Nodo de Carga debido a restricciones propias del dispositivo. Respecto al calefón, la temperatura se mantuvo siempre por encima de la mínima admisible para cada uno de los baños programados.

Adicionalmente, se concluyó que el modelado de los parámetros físicos del calefón es correcto. La predicción de temperatura es correcta cuando la temperatura del agua evoluciona

Capítulo 7. Prueba Experimental

libremente y cuando el calefón está encendido. Tal como era de esperar, la predicción no es buena cuando se realizan extracciones de agua. Sin embargo, fue posible comprobar que estos errores son mitigados por la realimentación de la temperatura inicial que se hace al inicio de cada día.

Finalmente, se observó que los Nodos de Carga accionaron los electrodomésticos de acuerdo a las ordenes enviadas por el Nodo Controlador. No hubieron problemas de pérdidas de comunicación por lo tanto la estrategia de mitigación de los efectos del ruido ambiente, detallada en el Apéndice L fue eficaz.

Capítulo 8

Evaluación del Sistema

El sistema HEMS diseñado debe ser capaz de moldear la demanda energética de un hogar de acuerdo a los precios de un tarifa *DAP* y el confort del usuario. A su vez el mismo debe brindar un servicio estable y amigable para el usuario. En el Capítulo 7 se comprobó que el Nodo Controlador diseñado funciona correctamente en un ‘hogar laboratorio’. Se verificó que el algoritmo se ejecuta correctamente y que los electrodomésticos se accionan en función del resultado devuelto. En el presente capítulo se evalúa cómo sería el desempeño del sistema diseñado en un hogar real a partir de simulaciones.

En primer lugar, se definieron dos hogares tipo y se simuló el funcionamiento del sistema para estos hogares durante un año. A partir de los resultados de las simulaciones se evalúa el desempeño del algoritmo y el beneficio económico que puede obtener el cliente por usar el sistema HEMS. A continuación, se realiza un breve análisis de la conveniencia económica del sistema desarrollado. Finalmente, se evalúan otras características relevantes del sistema, como su robustez, portabilidad y seguridad.

La evaluación del sistema HEMS en su conjunto permitió ver que existen múltiples mejoras y modificaciones que podrían realizarse al sistema para hacerlo más eficiente y más atractivo para el usuario. El desarrollo de estas mejoras quedó fuera del alcance de este proyecto. En este capítulo se enumeran las ideas más interesantes que surgieron a lo largo de la implementación y evaluación del sistema.

8.1. Simulación del funcionamiento del algoritmo

En primer lugar se evaluó el desempeño del algoritmo de gestión. Como se explicó en el Capítulo 4, este algoritmo es un aspecto muy importante del sistema ya que el éxito del mismo depende en gran medida de qué tan efectivo es el algoritmo de gestión. En la prueba experimental se pudo verificar el correcto funcionamiento del mismo para los días del experimento. Con el fin de tener una evaluación más extensiva, se realizó una simulación del funcionamiento del algoritmo para dos hogares durante un año. A continuación se describe cómo se armó la simulación y cuales fueron los resultados obtenidos.

Descripción de la simulación

Para poder realizar una simulación del algoritmo durante un año es necesario especificar todas las entradas del sistema para todos los días del año. Se debe especificar la curva de precios, los electrodomésticos presentes en el hogar, las configuraciones del usuario y la

Capítulo 8. Evaluación del Sistema

información necesaria para los modelos térmicos de los electrodomésticos (temperatura ambiente y temperatura de agua fría).

Curva de precios

La curva de precios es una de las entradas más importantes del sistema de gestión. Día a día el algoritmo realiza un balance entre costo y confort para decidir cuánta energía suministrar a cada electrodoméstico.

El algoritmo fue diseñado para una curva de precios del tipo *DAP*. Actualmente en Uruguay no existe una tarifa de precios de este estilo, por lo que fue necesario crear una curva de precios ficticia para realizar la simulación. Para esto se utilizaron los datos históricos de la tarifa *DAP* de la empresa estadounidense *ComEd*¹ y se realizaron ajustes para obtener una curva más representativa de la realidad uruguaya. A continuación se describe los ajustes realizados:

1. Corrimiento de seis meses de la curva de precios.

El perfil de consumo de energía eléctrica del sector residencial depende fuertemente del clima. Los usuarios tienen distintos patrones de consumo en las distintas estaciones del año. Por lo tanto, es razonable pensar que el patrón de la curva de precios también varíe con las estaciones del año. Se decidió realizar un corrimiento de seis meses en la curva de precios de *ComEd* de forma que las estaciones en la curva de precios coincidan con las estaciones del año uruguayas.

2. Ajuste de la media a la realidad uruguaya

La curva de precios original presenta valores que son ajenos a la realidad uruguaya, producto de la diferencia entre las divisas y los pliegos tarifarios de ambos países. Es de interés que la curva de precios a utilizar en la simulación sea comparable con la tarifa plana uruguaya. Esto permite obtener una estimación del ahorro que obtiene el cliente al utilizar el sistema HEMS. Con este fin, se decidió tomar la hipótesis de que la media de precios anual de la tarifa *DAP* es igual a la media de una tarifa plana (5.517 \$/kWh)².

En la Figura 8.1 se muestra la curva de precios del año ajustada a la realidad uruguaya. Se observa que los precios presentan una gran variabilidad alcanzándose los 20 \$/kWh. Por otro lado en el gráfico de la Figura 8.2 muestra los percentiles de precios por hora. Este gráfico permite tener un mejor entendimiento de cómo varía la curva hora a hora, dado que muestra la probabilidad ocurrencia de un precio para cada una de las horas del día.

En el gráfico se destacan además tres curvas de precio **Barato**, **Promedio** y **Caro**, los cuales corresponden a los percentiles 15 %, 50 % y 85 % respectivamente³. Para fijar ideas, la curva de precio **Barato** en cada hora contiene el precio que deja por debajo el 15 % de todos los precios de esa hora. Se observa que en promedio las horas más caras se dan al inicio de la jornada laboral y en el retorno a los hogares, de forma de desincentivar el consumo en los momentos de mayor estrés para la red eléctrica. En cambio, las horas con precios más baratos se dan en la madrugada las cuales corresponde a las de menor demanda acumulada.

Observación: las hipótesis asumidas para ajustar la curva de precios son suposiciones simples que se realizaron en función de los datos disponibles. Se podría haber confeccionado

¹ *Commonwealth Edison Company (ComEd)* es una empresa suministradora que brinda el servicio de distribución y comercialización de energía para clientes residenciales y comerciales en Estados Unidos. <https://hourlypricing.comed.com/>.

² El valor 5.517 \$/kWh corresponde al precio de la energía en el segundo escalón (101 - 600 kWh) del pliego tarifario vigente en 2016[46].

³ Estas tres curvas de precio se utilizaron para ajustar los parámetros de las funciones de utilidad en la simulación, ver Apéndice C.

8.1. Simulación del funcionamiento del algoritmo

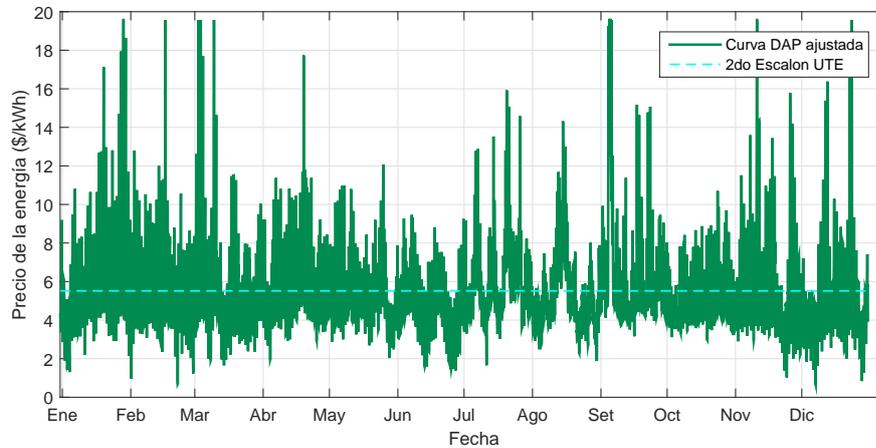


Figura 8.1: Curva de precios DAP utilizada en la simulación

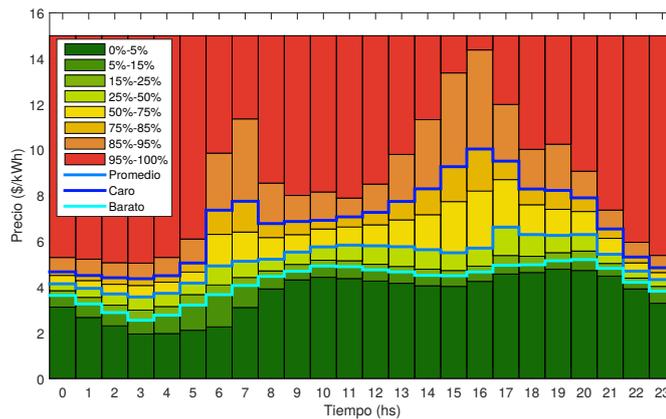


Figura 8.2: Percentiles de precio por hora de la curva utilizada

otro tipos de curvas haciendo ajustes similares con otras fuentes de información, por citar un ejemplo el precio SPOT de la energía eléctrica. Por simplicidad se optó por la primera opción, ya que no se encontraba dentro del alcance de este proyecto generar una curva de precios adaptada a la realidad uruguaya.

Hogares tipo

Se definieron dos hogares tipo para realizar las simulaciones y evaluar el algoritmo. Para cada hogar tipo se debe definir la cantidad de personas presentes en el hogar, el conjunto de electrodomésticos controlables y las configuraciones que realiza el usuario en cada uno de ellos. No se diseñó una configuración de usuario distinta para cada día del año, sino que se diseñaron cuatro semanas tipo, una para cada estación del año. A continuación se describen las principales características de los dos hogares tipo. En el Apéndice K se describe estos hogares y sus configuraciones con mayor detalle.

Capítulo 8. Evaluación del Sistema

Hogar Tipo 1

El Hogar Tipo 1 se diseñó tomando como referencia un apartamento habitado por dos personas adultas, que cumplen con una jornada laboral típica de 9:00 a 18:00. En este hogar se contemplaron tres electrodomésticos controlables un calefón, un aire acondicionado y un lavarropas. Se tomó que la potencia contratada del apartamento era de 4,6 kW.

Hogar Tipo 2

El Hogar Tipo 2 se construyó tomando como referencia una casa habitada por dos personas adultas y dos niños. Los adultos trabajan de 9:00 a 18:00, mientras que los niños concurren a la escuela en el turno matutino. Este hogar cuenta con seis electrodomésticos controlables: un calefón familiar, un calefón utilizado sólo por los adultos, un aire acondicionado, un lavasecarropas, un lavavajillas y una bomba de riego. Se tomó como potencia contrata 9,2 kW.

Se tomó como hipótesis que este hogar presenta un mayor poder adquisitivo que el hogar anterior. Por lo tanto las configuración de usuario priorizan más el confort de los habitantes del hogar.

Temperatura de ambiente

El modelo del aire acondicionado, descrito en el Apéndice B, precisa una estimación de la evolución de la temperatura a lo largo del día. Para la simulación fue necesario contar con una curva de temperatura ambiente por hora para todos días del año. Esta información fue obtenida del sitio web *WUnderground*, como se describió en la Sección 5.4.

Temperatura de agua fría

El modelo térmico del calefón es sensible a la temperatura del agua fría que ingresa al tanque. En la simulación se utilizaron los datos proporcionados por OSE en [47], ver Tabla B.6.

Evaluación de los resultados

Con los datos descritos, se procedió a ejecutar la simulación y se analizaron los resultados obtenidos. En primer lugar, se revisaron los resultados devueltos por el algoritmo y se encontró que los mismos eran razonables y cumplían con las restricciones impuestas⁴.

Convergencia del algoritmo

Luego se analizó el desempeño del algoritmo en cuanto a la convergencia del proceso de optimización. De un total de 732 días simulados (366 días del Hogar Tipo 1 y 366 días del Hogar Tipo 2) se obtuvo:

- En el 79,2% de los días se llegó a la *solución óptima*. Es decir el algoritmo convergió en el primer intento.
- En el 20.5% de los días se encontraron *soluciones con correcciones*. En estos casos el algoritmo no encontró la solución óptima en el primer intento, por lo que se hicieron pequeñas modificaciones de los parámetros del problema hasta obtener una solución óptima. Este proceso correctivo fue descrito en la Sección 5.5.

⁴Se cumplió con las restricciones todos los días, a excepción de los días en que fue necesario recurrir a la solución de pánico.

8.1. Simulación del funcionamiento del algoritmo

- En el 0,3 % de los días no se llegó a una solución óptima luego de realizar el máximo de correcciones posibles. Por lo tanto fue necesario recurrir a la *solución de pánico*, ver Sección 5.5.

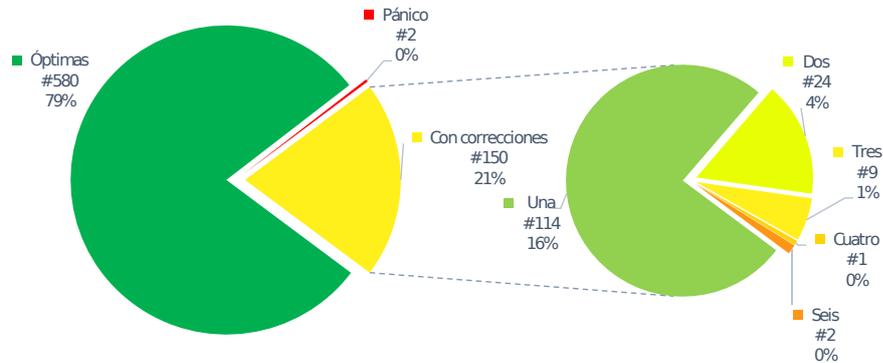


Figura 8.3: Resultados del proceso de optimización en las simulaciones de hogares tipo

En la Figura 8.3 se muestra gráficamente con los resultados de la convergencia del algoritmo. En el gráfico de la izquierda se muestran las proporciones de soluciones *óptimas*, *con correcciones* y *pánico*. En el gráfico de la derecha se indican la cantidad de correcciones necesarias para el conjunto de soluciones *con correcciones*. En base a los resultados obtenidos se concluyó que la convergencia del algoritmo fue muy satisfactoria.

Tiempo de ejecución

Otro punto interesante de analizar es el tiempo de ejecución del algoritmo. El algoritmo utilizado es tipo *day-ahead*, es decir se ejecuta una vez al día al inicio de la jornada. Por esta razón no se requiere de una gran velocidad para resolver el problema. Igualmente es necesario corroborar que estos tiempos no sean excesivos.

Hogar Tipo	Tiempo promedio (mm:ss)	Tiempo máximo (mm:ss)
1	00:49	07:07
2	02:10	36:13

Tabla 8.1: Tiempos de ejecución del algoritmo durante la simulación

En la Tabla 8.1 se muestran los tiempos de ejecución promedio y máximo para cada uno de los hogares. En el Hogar Tipo 1 el tiempo de ejecución fue inferior a 10 minutos en todos los casos. En el Hogar Tipo 2 el tiempo promedio de ejecución un 164 % mayor, esto se debe a la mayor cantidad de electrodomésticos. El tiempo de ejecución máximo del segundo hogar es mayor a treinta minutos. Sin embargo, este tiempo corresponde a un día en el que fue necesario recurrir a una solución de pánico. Si se eliminan los días con soluciones de pánico el tiempo máximo de ejecución es de 8:11 minutos.

El módulo Actuador del Nodo Controlador asume que el resultado del proceso de optimización está disponible a las 00:00hs del día. En caso que esta información no esté disponible el módulo falla. Es necesario que el tiempo de ejecución del algoritmo no supere el tiempo disponible antes del inicio del próximo día. Para esto es posible reducir la cantidad de correcciones

Capítulo 8. Evaluación del Sistema

máximas que puede hacer el algoritmo. Incluso sería conveniente agregar un procedimiento que asegure que el resultado de la optimización está disponible antes de la medianoche, el cual recurre a la solución de pánico si el algoritmo no convergió a una determinada hora límite. Este procedimiento quedó por fuera del alcance de este proyecto.

Evaluación del ahorro obtenido

Finalmente se estimó cuál es el ahorro que obtiene cada uno de los hogares al utilizar el sistema HEMS. Para esto se comparó el gasto de cada hogar tipo sin utilizar el sistema y bajo una tarifa plana, con el gasto en energía del mismo hogar con sistema HEMS y bajo una tarifa DAP.

Para estimar el gasto de cada hogar sin HEMS, se calculó el consumo energético que tendría cada hogar tipo con los mismos requerimientos de confort y operando los electrodomésticos manualmente. Para calcular el gasto se tomó un precio fijo de 5.517 \$/kWh. A modo de ejemplo se analizan los resultados obtenidos para el Hogar Tipo 1. Más adelante en este mismo capítulo se muestran los resultados del Hogar Tipo 2.

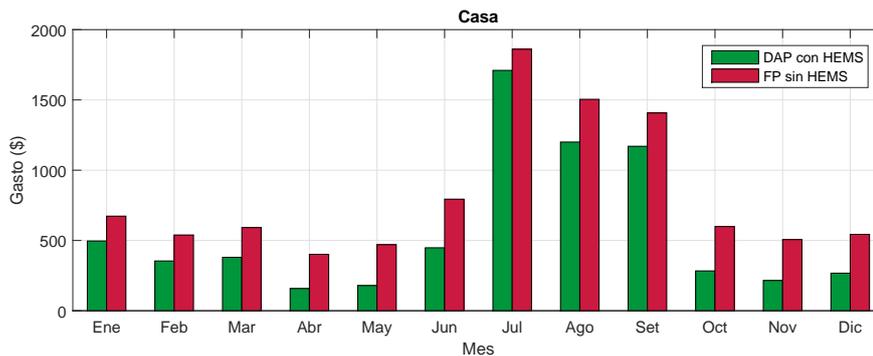


Figura 8.4: Gasto mensual del Hogar Tipo 1 para escenarios con HEMS bajo DAP y sin HEMS bajo FP

En la Figura 8.4 se muestran los gastos por mes en ambos escenarios, DAP con HEMS y FP sin HEMS. Se observa que al utilizar el sistema permite se reducen los costos en todos los meses del año, totalizando un ahorro anual de \$3.027 que representa un 30,6 % del gasto total sin utilizar el HEMS.

En la Figura 8.5 se muestra el ahorro por electrodoméstico. En primer lugar se observa que el calefón es el electrodoméstico que otorga mayor beneficio, esto se debe a las siguientes dos razones. La primera es que el calefón presenta una gran inercia térmica lo que permite aprovechar las horas más baratas del día para calentar el agua. La segunda razón es que el HEMS hace un uso más eficiente del calefón ya que mantiene el agua caliente sólo en el período de interés del usuario. Para este electrodoméstico, en el caso sin HEMS, se consideró un usuario que no toma medidas para reducir el costo de uso del electrodoméstico, sino que sólo se preocupa por su confort. Bajo esta suposición se asumió que el calefón se encuentra conectado durante todo el día, con su termostato regulando la temperatura a la de temperatura de confort del usuario. Se consideró que este es el caso más representativo de la realidad uruguaya y fue el utilizado en la simulación.

Por otro lado, se observa que el aire acondicionado presenta ahorro negativo. En otras palabras el gasto del aire acondicionado en el escenario con HEMS es mayor al escenario sin

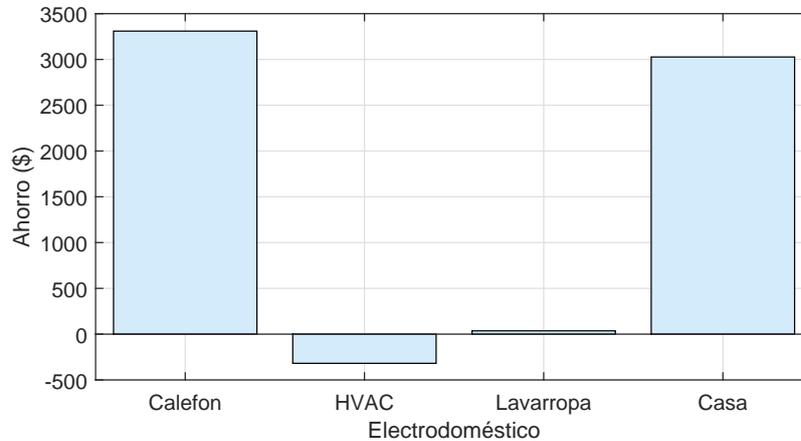


Figura 8.5: Ahorro por electrodoméstico en Hogar Tipo 1

HEMS. En el escenario con HEMS, como la inercia térmica de los hogares no es buena, el algoritmo de gestión decide encender el aire acondicionado durante las horas de interés del usuario. Por lo tanto, el patrón de energía que se le da al aire acondicionado muy similar en ambos escenarios⁵. La diferencia en el gasto se debe a que se utilizan tarifas de precios distintas en ambos escenarios. En ambos hogares se consideró que el usuario desea que la habitación esté aclimatada cuando retornan a sus hogares, las cuales se corresponden en general a las horas más caras del día de la tarifa *DAP* (ver Figura 8.2). El precio de estas horas en general son más altos que el de la *FP* considera y por esta razón el costo en el escenario con HEMS es mayor al costo en el escenario sin HEMS.

Finalmente, el sistema HEMS decidió agendar el lavarropas en las horas más baratas del día y por lo tanto también se observa un ahorro para este electrodoméstico. Sin embargo, el consumo energético de este electrodoméstico es muy bajo y por lo tanto no presenta un peso de relevancia en el ahorro total.

En base a lo anterior, se verifica que el calefón es el electrodoméstico del hogar que presenta un mayor potencial en las políticas de Respuesta a la Demanda. El lavarropas también ofrece beneficio, pero no presenta un potencial tan alto por su bajo consumo energético. Se concluyó también que no es conveniente utilizar un aire acondicionado para aclimatar los hogares bajo una tarifa *DAP*. Es recomendable que un hogar que cuenta con una tarifa horaria debe utilizar algún otro mecanismo para aclimatar el hogar, por ejemplo un acumulador de calor. En caso de utilizar un aire acondicionado se debe procurar que el hogar presente buena aislación térmica con el fin de maximizar su inercia térmica.

8.2. Análisis económico

En esta sección se evalúa la conveniencia económica de adoptar la plataforma desarrollada en el marco uruguayo. En una primera instancia se evalúa cuál es el beneficio económico de utilizar la plataforma para cada uno de los hogares tipos. Posteriormente se calcula el costo de la misma y, por último, se realiza un estudio de costo/beneficio.

⁵Se observa que en el escenario sin HEMS se consideró que el equipo de aire acondicionado se enciende una hora antes del inicio del período de interés del usuario.

Capítulo 8. Evaluación del Sistema

Es importante destacar que este análisis económico se realizó con fines académicos para evaluar la viabilidad de la solución propuesta. El sistema diseñado es un prototipo y presenta muchos aspectos que deben ser mejorados para aumentar la eficiencia y reducir los costos. Tampoco se dispone de información certera de cómo sería una tarifa de precios *DAP* en Uruguay. Por estas razones se entiende que el análisis económico no es riguroso pero sí da una idea de qué tan conveniente puede llegar a ser el sistema diseñado.

Beneficio económico de utilizar el sistema HEMS

El beneficio económico que puede obtener el cliente al utilizar el sistema HEMS depende directamente de las características de la tarifa utilizada. En la actualidad no existe una tarifa de precios tipo *DAP* en Uruguay, y por lo tanto no es posible estimar con precisión el beneficio económico que obtiene el cliente. Con el fin de obtener una estimación del beneficio se simuló el funcionamiento del sistema HEMS para distintas tarifas.

El algoritmo de gestión se diseñó para una tarifa de precios tipo *DAP*. Sin embargo, el mismo puede ser utilizado bajo cualquier tarifa en la que se conozcan los precios con 24 horas de anticipación. Considerando esto se decidió simular los dos hogares tipo definidos bajo las siguientes tarifas de precios:

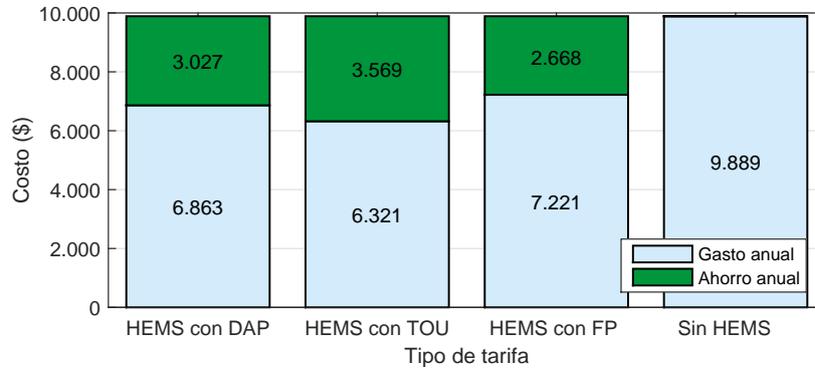
- **Tarifa *DAP***: tarifa *DAP* ficticia descrita anteriormente. Se recuerda que para elaborar la misma se tomó la hipótesis de que la media anual de la tarifa *DAP* coincide con la media de la tarifa plana *FP*.
- **Tarifa *FP***: tarifa plana, constante e igual a 5.517\$/kWh. Este precio se corresponde al segundo escalón de la tarifa residencial simple del pliego tarifario de UTE [46].
- **Tarifa *TOU***: tarifa doble horario que ofrece la empresa UTE en Uruguay, también llamada *Tarifa Inteligente*. Se caracteriza por tomar dos valores en dos períodos horarios definidos, ver Tabla 8.2.

Denominación	Rango horario (hh:mm)		Precio (\$/kWh)
	Inicio	Fin	
Punta	17:00	23:00	7.352
Fuera de punta	0:00	17:00	2.944
	23:00	00:00	

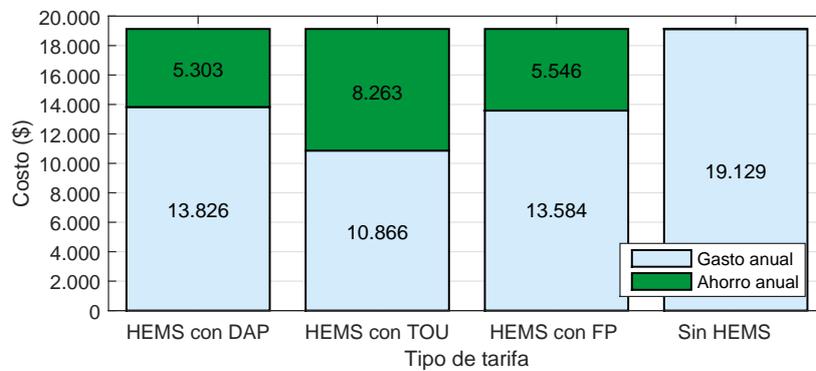
Tabla 8.2: Tarifa doble horario residencial (*TOU*). Fuente [46]

Es importante destacar que el algoritmo de gestión fue diseñado para una tarifa tipo *DAP*. Para ser utilizado con tarifas como la *FP* y *TOU* se deben hacer algunas modificaciones al mismo. Actualmente el algoritmo tiende a equidistribuir la energía que entrega a los electrodomésticos de consumo discreto en las horas más baratas dentro del intervalo de validez \mathcal{T}_a . Esto se debe a que el algoritmo no tiene información sobre cómo priorizar un *timeslot* sobre el otro en los electrodomésticos de consumo discreto. En las tarifas *FP* y *TOU* existen múltiples horas de igual precio, lo que lleva a que el algoritmo decida encender los electrodomésticos una fracción de tiempo en cada hora. Esto puede ser problemático en algunos electrodomésticos, por lo cual sería conveniente modificar el algoritmo de gestión. Una posible solución a este problema es modificar la función de utilidad utilizadas para estos electrodomésticos (Ecuación 4.7) de forma que pondere los *timeslot*, ver Sección 8.4.

8.2. Análisis económico



(a) Hogar Tipo 1



(b) Hogar Tipo 2

Figura 8.6: Comparación de gastos y ahorros con HEMS y distintas tarifas

En la Figura 8.6 se muestran los resultados obtenidos para cada tarifa y cada hogar tipo. También se muestra el ahorro anual respecto a un escenario sin HEMS y con tarifa *FP*. Se observa que en todas las tarifas se obtiene beneficio económico al utilizar el sistema HEMS.

A pesar de que el algoritmo de gestión fue diseñado para una tarifa tipo *DAP*, se destaca que en ambos hogares el mayor beneficio se obtiene bajo la tarifa doble horario (*TOU*). Esto se debe a que la tarifa doble horario de UTE tiene un valor medio de 4.046 \$/kWh que es inferior a la media de la tarifa *DAP* (5.517 \$/kWh).

Considerando lo anterior parece razonable considerar que la empresa suministradora fomente el uso de una nueva tarifa *DAP* de igual manera que la actual tarifa doble horario. Por lo tanto se creó una nueva tarifa ficticia, a la que llamamos *DAP2*, en la cual se cambia la hipótesis de la media anual fijándola en 4.046 \$/kWh. En la Tabla 8.3 se muestra el beneficio económico que obtiene los hogares tipo bajo esta tarifa. Se decidió tomar estos valores para realizar el análisis costo/beneficio de la plataforma.

Capítulo 8. Evaluación del Sistema

Hogar Tipo	Gasto (\$) sin HEMS y <i>FP</i>	Gasto (\$) con HEMS y <i>DAP2</i>	Ahorro (\$ (%))	
1	9.889	5.033	4.856	49
2	19.129	10.140	8.989	47

Tabla 8.3: Resultados de ahorro total por Hogar Tipo

Costo de la plataforma por Hogar Tipo

El costo de la plataforma depende de la cantidad y del tipo de electrodoméstico que debe controlar el sistema. En la figura Tabla 8.4 se muestran la composición y el costo de la plataforma para cada uno de los hogares tipo. Con el fin de comparar el costo de la plataforma con el ahorro mensual, se convirtió la inversión de dólares americanos a pesos uruguayos. Para esto se tomó un tipo de cambio de 31,58 pesos uruguayos por cada dólar, valor correspondiente al promedio desde el 1 de enero al 28 de abril de 2016[48]. En la Tabla 8.5 se muestra la inversión inicial necesaria en cada hogar tipo.

Función	Equipo	Precio un.	Hogar Tipo 1		Hogar Tipo 2	
			Cant.	Subtotal	Cant.	Subtotal
Lógica central	Raspberry PI 2	41,0	1	41,0	1	41,0
Comunicación inalámbrica	Zolertia Z1	135,6	4	542,6	7	949,5
Sensor de temperatura	SHT 25	34,2	1	34,2	3	68,4
Conmutación	Relé SSR	14,7	2	29,3	5	73,3
Comando HVAC	Módulo IR	18,0	1	18,0	1	18,0
Total				665,1	Total	1150,1

Tabla 8.4: Costo de la plataforma en dólares americanos

Hogar Tipo	Inversión (\$)
1	21.005
2	36.325

Tabla 8.5: Inversión en la plataforma por hogar tipo

Observación:

Se recuerda que la plataforma diseñada en este proyecto es un prototipo, en el cual no se buscaron que los componentes fueran de bajo costo.

Análisis costo/beneficio por Hogar Tipo

Con la información de costos y beneficios se realizó un análisis económico simple para estudiar cuales son los beneficios reales de utilizar la plataforma diseñada. Dicho análisis consistió en determinar cuanto tiempo se debe utilizar la plataforma para recuperar la inversión inicial. Los resultados se encuentran en la Tabla 8.6.

La inversión inicial en el Hogar Tipo 2 fue un 72 % superior a la del Hogar Tipo 1, sin embargo el período de repago resultante es menor. Esto se debe a que los beneficios anuales

8.2. Análisis económico

Hogar Tipo	Inversión (\$)	Beneficio anual (\$)	Período de repago (meses)
1	21.005	4.856	52
2	36.325	8.989	48

Tabla 8.6: Análisis de período de repago de la plataforma

obtenidos fueron aproximadamente duplican a los del Hogar Tipo 1. La razón por la cual el ahorro en el Hogar Tipo 2 es mayor, es que el mismo posee más electrodomésticos controlables, en particular tiene dos calefones que son los que brindan mayor beneficio.

En el mejor de los casos, el período de repago es de 48 meses. Bajo estas condiciones, no resulta seductor para los usuarios introducir el HEMS en sus hogares, ya que otorga beneficios económicos recién a partir del cuarto año posterior a su instalación. Para mejorar el atractivo de la plataforma se pueden aplicar varias estrategias, incentivos por parte de la empresa suministradora que faciliten la adquisición, aumentar las funcionalidades que ofrece el sistema HEMS y la reducción del costo de la plataforma.

La empresa suministradora obtiene múltiples beneficios cuando se aplican políticas de Respuesta a la Demanda, por ejemplo se reducen los costos de generación, las pérdidas técnicas y los costos de planificación [10, 49]. La empresa suministradora podría aplicar políticas de subvención o préstamos de dinero a los clientes que deseen adquirir la plataforma. De esta manera se reduce la inversión inicial de los consumidores y la empresa suministradora obtendría una inserción más rápida de la tarifa *DAP*. En Uruguay existen antecedentes de este tipo de subvención como por ejemplo el *Plan Solar* y el *Plan Recambio Termotanque*.

El agregado de funcionalidades adicionales a la plataforma permitiría aumentar el atractivo de instalar sistema HEMS. En la Sección 8.4 se describen algunas de estas funcionalidades. Los usuarios adquirirían la plataforma no sólo porque brinda un beneficio económico sino porque también aumenta el confort del hogar.

Por último, se debe reducir el costo de la plataforma realizando un diseño a medida de la misma. A continuación se realiza un análisis de los costos de la misma.

Reducción del costo de la plataforma

Con el fin de analizar cómo es posible reducir los costos de la plataforma, se calculó cuál es el peso relativo de cada uno de los elementos que la componen. En la Tabla 8.7 se encuentra la proporción de cada dispositivo en la inversión total.

Función	Equipo	Proporción de la inversión (%)	
		Hogar Tipo 1	Hogar Tipo 2
Lógica central	Raspberry PI 2	6,2	3,6
Comunicación inalámbrica	Zolertia Z1	81,6	82,6
Sensor de temperatura	SHT 25	5,1	5,9
Conmutación	Relé SSR	4,4	6,4
Comando HVAC	Módulo IR	2,7	1,6

Tabla 8.7: Análisis de proporción de costo por elemento de la plataforma

Se observa que más del 80% del costo de la plataforma proviene de los dispositivos de comunicación inalámbrica (motes Z1). Como se describió en el Capítulo 3 el sistema HEMS

Capítulo 8. Evaluación del Sistema

requiere un mote para cada Nodo de Carga que operan los electrodomésticos y además uno que oficie de *Border Router* para el Nodo Controlado. Era esperable que su peso en la inversión total fuese grande por la cantidad de motes necesarios. Por lo tanto para producir una importante reducción de la inversión total se debe trabajar sobre este dispositivo.

Los motes Z1 utilizados en el prototipo presentan un costo muy elevado y son el componente con mayor costo unitario. Estos dispositivos genéricos son diseñados para el uso de investigadores y desarrolladores, no para fines comerciales. Además, los motes están sobredimensionados para el uso que se le da en el sistema HEMS, presentando funcionalidades que no son utilizadas. Es por esto que se cree que un nuevo mote, diseñado a medida para cumplir con su función en el sistema HEMS podría tener un costo mucho menor.

8.3. Cualidades del sistema HEMS

A continuación se analizan otras cualidades del Nodo Controlador implementado y del sistema HEMS en general.

Performance

El Nodo Controlador diseñado es un primer prototipo, en consecuencia algunas de las decisiones de diseño tomadas no fueron óptimas desde el punto de vista del desempeño. Podrían haberse utilizado otras herramientas computacionales que realicen un uso más eficiente de los recursos, y así reducir los requerimientos de hardware y en consecuencia los costos de la plataforma.

Adicionalmente, se podrían mejorar los tiempos de respuesta del sistema, por ejemplo los tiempos de respuesta de la interfaz usuario y de ejecución del algoritmo de optimización. Sin embargo, el algoritmo no presenta grandes exigencias en cuanto a la velocidad de respuesta por ser del tipo *day-ahead*. La interfaz de usuario actual cumple con tiempos normales de respuesta que no dificultan el uso por parte del usuario. Por último, el problema de optimización se resuelve a tiempo para enviar las órdenes a los nodos del sistema.

Robustez

Ante cortes en el suministro de energía eléctrica el Nodo Controlador deberá ser reiniciado. Al reiniciar el mismo se seguirán obteniendo las mediciones de los distintos nodos y registrando los cambios que realiza el usuario a través de la interfaz. Sin embargo, sí se pierden todas las decisiones tomadas por el algoritmo de optimización para ese día. La implementación de una mejora en el restablecimiento del sistema quedó fuera del alcance de este proyecto.

Por otro lado, si hubiera problemas de conectividad a Internet sólo provocaría la no actualización de las predicciones de temperatura y precios. En caso de que no se actualicen los valores a tiempo, existen valores por defecto configurados tanto para el precio como para la temperatura ambiente, que se utilizarán para realizar la optimización. Por lo que la pérdida de conectividad, no representa un problema si es que se pierde la conexión por un período corto de tiempo.

A nivel de la Interfaz Usuario, la misma está diseñada para desplegar un mensaje de error en caso de que el usuario incurra en errores involuntarios al configurar el sistema, como por ejemplo, el ingreso de parámetros inválidos o fuera del rango permitido. Los mensajes retornados cuentan con una breve descripción del problema para que el usuario pueda corregirlo fácilmente y así hacer las modificaciones que desea.

Seguridad

Los usuarios pueden modificar el funcionamiento del sistema de gestión HEMS a través de la interfaz web de usuario, tal como se explicó en el Capítulo 6. Esta interfaz inicialmente estuvo pensada para ser utilizada dentro de una LAN hogareña, sin acceso a la misma a través de Internet. Dentro de la interfaz hay funcionalidades e información que puede ser accedidas por cualquier usuario. Pero se restringe el acceso a las operaciones más importantes del sistema, que sólo pueden ser realizadas por usuarios con los permisos necesarios. Para ello, el usuario debe registrarse en el sistema con un usuario y contraseña tal como es detallado en la Sección 6.4.

La interfaz web sólo puede ser accedida dentro de una red local del hogar. No es posible operar directamente los electrodomésticos desde de la Interfaz Usuario, sino que sólo se puede afectar las variables del problema de optimización que se ejecuta una vez al día. De esta manera no es posible que un usuario mal intencionado pueda dañar los electrodomésticos mediante la operación reiterativa de los mismos por esta vía. Se entiende que un usuario mal intencionado no podría obtener ningún beneficio ni tampoco dañar los electrodomésticos y por lo tanto no existe motivación por intentar violar la seguridad del sistema. Por estas razones no se cree necesario agregar otro tipo de seguridad para el sistema.

Portabilidad

Desde el comienzo se buscó que el sistema pueda ser instalado rápidamente en los hogares. Todo el sistema se encuentra alojado en la placa central, inclusive la información referente a la parametrización correspondiente a cada usuario. Simplemente habría que hacer un relevamiento de las características de los electrodomésticos que se quiere incluir en el sistema y las características del hogar donde se instala.

Cabe destacar que las configuraciones varían mucho no sólo en función de las características de los electrodomésticos sino que también del lugar donde se instalarán los mismos. Por lo que la instalación del sistema diseñado requiere que un técnico releve las características específicas de cada hogar. Una alternativa muy interesante sería que el sistema HEMS contara con una etapa de aprendizaje, de modo los parámetros físicos de los electrodomésticos y del hogar fueran calculados a través de medidas obtenidas de los nodos sensores, ver Sección 8.4.

Evolucionabilidad

Al tratarse de un prototipo en esta versión no fue considerado el caso de que hayan modificaciones en el proceso de optimización o en la Interfaz Usuario con el paso del tiempo. A modo de ejemplo, se podría agregar una página en la Interfaz Web desde la cual el usuario podría descargar e instalar nuevas actualizaciones que incluyan cambios en el sistema ya sea en la apariencia de la interfaz de usuario, nuevas funcionalidades o alteraciones en el funcionamiento del nodo controlador.

Por otro lado, el agregado de un nuevo electrodoméstico al sistema de gestión supone realizar dos configuraciones que deben ser realizadas por un técnico.

1. Instalación de un nuevo Nodo de Carga.

Se debe instalar un nuevo Nodo de Carga en el hogar para que el Nodo Controlador pueda comandar el electrodomésticos y obtener mediciones en caso de que sea necesario. Si bien se implementaron prototipos de Nodos de Carga en este proyecto, su diseño depende del electrodoméstico a controlar. No está dentro del alcance de este proyecto el diseño de Nodos de Carga universales que puedan ser adquiridos e instalados directamente por el usuario.

Capítulo 8. Evaluación del Sistema

2. Actualización de la base de datos.

Se debe agregar a la base de datos la información del electrodoméstico y de su Nodo de Carga. Actualmente esto se puede realizar sólo configurando directamente la base de datos en la placa central, no siendo posible de realizarlo desde la Interfaz Web. No se encuentra dentro del alcance del proyecto agregar esta facilidad de instalación de nuevos electrodomésticos. La misma consistiría en adicionar a la Interfaz Usuario la posibilidad de ingresar con un rol especial, que permita realizar la configuración de los electrodomésticos, y así evitar que usuarios no expertos puedan hacer estos cambios sin el conocimiento adecuado.

Mantenibilidad

Si hubiera problemas con el hardware, tanto en la placa central así como en cualquiera de los otros nodos periféricos, se podría desconectar de forma simple el sistema de gestión dejando que el usuario comande los electrodomésticos normalmente.

En los Nodos de Carga implementados en este proyecto es posible comandar los electrodomésticos manualmente una vez desconectado el nodo de carga. Sin embargo, esto depende fuertemente de las características de los Nodos de Carga utilizado.

Flexibilidad para el usuario

Como se describió en el Capítulo 6, el usuario puede a través de la interfaz web consultar cómo será el despacho eléctrico de los distintos electrodomésticos para el día actual. En caso que se desee cambiar la configuración actual y encender alguno de los electrodomésticos fuera del período de que el algoritmo decidió agendarlo, el usuario tiene la posibilidad de accionar manualmente los electrodomésticos. Nuevamente se destaca que esta facilidad depende de las características de los Nodos de Carga utilizados.

Es interés agregar la posibilidad del usuario pueda operar los electrodomésticos por fuera de las decisiones del algoritmo a través de la Interfaz de Usuario. Esta funcionalidad quedó fuera de alcance de este proyecto, quedando la misma para trabajos futuros.

8.4. Trabajo a Futuro

Si bien lo implementado en este proyecto cumple en gran medida con los requisitos y objetivos propuestos inicialmente, es preciso señalar que existen muchas características del sistema HEMS que aún no han sido del todo explotadas y que es posible realizar mejoras en varios frentes. En el transcurso del proyecto y la evaluación de los resultados obtenidos se encontraron múltiples mejoras y modificaciones que podrían realizarse a fin de que el sistema sea más eficiente y atractivo para los usuarios. A continuación se describen algunas de estas mejoras.

Manejo de cargas no interrumpibles

Una de las principales falencias que presenta el algoritmo de gestión implementado, es que no considera la existencia de cargas no interrumpibles. Esto lleva a que el sistema HEMS no sea del todo adecuado para el manejo de electrodomésticos como un lavarropas o lavavajillas.

En la solución propuesta el módulo Intérprete del Nodo Controlado (ver Sección 5.6), tiene en consideración cuáles son los electrodomésticos que no se desea interrumpir y busca interrumpirlos lo menos posible. Sin embargo, no existe ninguna restricción en el problema de

optimización que impida que el algoritmo decida dar energía en *timeslot* disjuntos. Tal como se explicó en la Sección 2.4, en la búsqueda bibliográfica se encontraron propuestas acerca de cómo considerar esto en el problema de optimización. A futuro podría agregarse este tipo de consideraciones, pero para ello habría que hacer cambios en el planteo del problema y sus restricciones.

Función de utilidad

La función de utilidad de los electrodomésticos ($U(\mathbf{q}_a)$) permite cuantificar la satisfacción que recibe el usuario de acuerdo a la vector de energía que se le suministra (\mathbf{q}_a). En este proyecto se propusieron funciones de utilidad para los electrodomésticos más representativos del hogar y estas demostraron funcionar adecuadamente. Sin embargo se entiende que aún existe mejoras que se pueden hacer a las mismas, de forma que reflejen de mejor manera la satisfacción del usuario.

En particular para los electrodomésticos de consumo discreto se propuso que la satisfacción del usuario crece linealmente con el total de energía suministrada. En los electrodomésticos que se debe suministrar una cantidad fija de energía durante el día, la función de utilidad carece de sentido, ver Sección 4.5. Se entiende que esta función de utilidad puede ser modificada a fin de que el usuario pueda realizar una configuración más personalizada. Una modificación posible es implementar una función de utilidad que varíe según en qué *timeslot* se suministra la energía, de forma que el usuario pueda ajustar cuándo prefiere que se ejecute la tarea: al inicio, al final o en una determinada hora dentro del periodo de interés (\mathcal{T}_a). Por ejemplo, si el usuario desea que la tarea se ejecute cerca del final del periodo la función de utilidad debería crecer con t . De esta manera el algoritmo de gestión tendería a ejecutar la tarea lo más tarde posible, buscando un equilibrio entre utilidad y costo.

La modificación de las funciones de utilidad de los electrodomésticos se puede realizar de forma sencilla en el sistema implementado. Para esto basta con modificar las funciones `utilidad`, `gradienteUtilidad` y `hessianaUtilidad` de alguna de las clases de electrodomésticos implementadas o creando una nueva clase de Electrodoméstico, como se describió en la Sección 5.3.

Mayor flexibilidad para el usuario

Por tratarse de un sistema pensado para tarifas DAP, se decidió que el algoritmo se ejecute una única vez antes del inicio de cada día. Esto presenta la desventaja de que el usuario debe configurar cómo desea que se agenden los electrodomésticos con 24 horas de anticipación. Una vez que se ejecutó el algoritmo de gestión, el usuario ya no puede hacer modificaciones. En caso que ocurra un imprevisto y el usuario precise que re-configurar algún electrodoméstico, esto debe hacerse por fuera del algoritmo de gestión operando los electrodomésticos manualmente. Se entiende que se deben realizar las modificaciones anteriores para otorgarle mayor flexibilidad al usuario.

Una solución posible sería modificar el funcionamiento del Nodo Controlador, para que el algoritmo se ejecute periódicamente verificando si el usuario realizó alguna modificación en la configuración. O que cada vez que el usuario hace cambios para el día actual se ejecute nuevamente el algoritmo de optimización.

Otra mejora interesante, sería que el usuario pueda operar otros electrodomésticos no considerados en el algoritmo de gestión a través de la interfaz de usuario, por ejemplo las luces del hogar.

Capítulo 8. Evaluación del Sistema

Las mejoras descritas implican modificaciones tanto en la programación del Nodo Controlador como en la interfaz de usuario. Estas además, presentan como contrapartida que los usuarios con malas intenciones, si llegan a acceder a la interfaz web, pueden encender y apagar los electrodomésticos repetidamente y así conseguir dañar los mismos. Por esta razón también sería necesario agregar controles adicionales en la seguridad del sistema. Se entiende que la implementación de estas mejoras agregan mucho valor al sistema HEMS y motivarían a una mayor cantidad de usuarios a adquirir el mismo.

Cambio de tarifa utilizada

El algoritmo de gestión implementado fue diseñado para una tarifa tipo *DAP*. Existe la posibilidad de adaptar la solución propuesta para utilizar otros tipo de tarifas.

En el caso de tarifas tipo *FP* y *TOU* las modificaciones a realizar son menores. Como se describió en la Sección 8.2 de este capítulo, el algoritmo tiende a equidistribuir la energía entre las horas de igual precio en los electrodomésticos de consumo discreto. La posible solución a este inconveniente es implementar una función de utilidad que varíe con el *timeslot*, tal como se describió en el punto '*Función de utilidad*' de esta sección.

En una tarifa *RTP* el precio de la energía varía hora a hora y la empresa suministradora lo anuncia con una hora de anticipación. En general, el usuario también dispone de una predicción de los precios futuros. La migración a una tarifa *RTP* implica cambios más grandes que para las tarifas *FP* y *TOU*. En primer lugar, el algoritmo debería re-ejecutarse como se describió en el punto '*Mayor flexibilidad para el usuario*'. En este caso, la re-ejecución se dispararía ante cambios del usuarios en la configuración y ante cambios en la curva de precios. Por otro lado, como se mencionó en la descripción de los resultados de la búsqueda bibliográfica (ver Sección 2.4) para manejar tarifas del tipo RTP es conveniente que el algoritmo de optimización sea robusto y considera la incertidumbres que presentan las predicciones de precios.

Obtención Dinámica de Precios

Como ya fue mencionado, no existe hoy en día una tarifa de precios tipo *DAP* en Uruguay. Sin embargo, una vez que se encuentre disponible este servicio sería muy fácil incorporarlo al modelo actual. Simplemente habría que crear un proceso que obtenga los precios para el día siguiente una vez publicados y los almacene en la base de datos. Actualmente, el módulo Configurator ya levanta la curva de precios desde la base de datos por lo que no habría que hacer ninguna otra adaptación en el sistema.

Aprendizaje de parámetros

Una mejora adicional que se podría agregar a lo ya implementado sería que el sistema HEMS aprendiera y ajustara automáticamente sus parámetros, a partir de mediciones de los sensores y la realimentación del usuario.

En primer lugar, el sistema HEMS podría ajustar los parámetros físicos de los modelos de los electrodomésticos a partir de las medidas obtenidas por los sensores del sistema. Para llevar a cabo esta modificación es necesario agregar un módulo que sea capaz de obtener los valores de las variables de interés y que a su vez sea capaz de tomar decisiones inteligentes acerca de cómo hacer los ajustes correspondientes en las variables del sistema. Además para evitar la complejidad de configurar los valores iniciales de estos parámetros físicos, se podrían establecer un periodo inicial donde el sistema no tome decisiones, y solo se dedique a tomar medidas y realizar los ajustes en los parámetros de los distintos electrodomésticos del hogar.

8.4. Trabajo a Futuro

Adicionalmente, se podría brindar la posibilidad al usuario de manifestar el grado de satisfacción que recibe de cada electrodoméstico así como también el ahorro mensual deseado. Con estos ajustes se podría adecuar el algoritmo más específicamente a cada usuario, personalizando la relación que existe entre el confort y el ahorro. Para incorporar esta prestación habría que agregar a la interfaz web una página donde el usuario tenga la posibilidad de manifestar su satisfacción para cada uno de los electrodomésticos y con el ahorro obtenido. Esa información debería ser almacenada en la base de datos y en función de estos valores se puedan hacer pequeños ajustes correctivos a los parámetros de las funciones de utilidad de los electrodomésticos. Por más información, acerca de cómo fueron ajustados estos parámetros en este proyecto ver Apéndice C.

Interfaz de Usuario

La Interfaz de Usuario fue diseñada para contemplar lo que se consideró más relevante para este prototipo. El diseño estuvo orientado a hacer la misma lo más práctica posible, pensando en que el usuario no presente impedimentos para adaptarse a su uso. Evidentemente existen innumerables mejoras para realizar en pos de hacer la interfaz más atractiva visualmente y maximizar la experiencia del usuario, teniendo en cuenta que ninguno de los integrantes del grupo es experto en la materia. Asimismo, se entiende que sería posible brindar nuevas posibilidades de configuración o mostrar más información del funcionamiento del sistema.

En la interfaz web implementada se cuenta con un grupo de tablas y gráficas que enseñan los gastos que ha tenido el usuario desde la instalación del nuevo sistema. Se entiende que sería de interés flexibilizar más aún la forma en que se muestra la información de modo de permitir al usuario, por ejemplo, comparar gastos entre diferentes días y electrodomésticos. Una opción sería brindar la posibilidad de elegir los rangos de fechas y electrodomésticos de los cuales quiere obtener la información de gastos. Esto permitirá que el usuario pueda tomar decisiones más eficientes referentes a la configuración del sistema.

Otro aspecto que puede ayudar a las decisiones tomadas por el usuario sería contar con las medidas obtenidas por los sensores del sistema. De esta manera, además de mostrarle la temperatura ambiente al usuario, el mismo podría conocer a qué temperatura está la habitación en determinado momento o saber si dispone de agua caliente para hacer uso de la misma.

También sería útil contar con la funcionalidad de crear y almacenar diferentes perfiles de uso. A modo de ejemplo, podrían guardarse diferentes configuraciones como: 'Vacaciones', 'Invierno', 'Licencia Médica', etc. De esta forma se evita tener que cambiar uno a uno los parámetros para adaptarse a las distintas situaciones lo cual puede resultar engorroso.

Finalmente, otro punto importante que agregaría mucho valor al usuario es dar la posibilidad de controlar el sistema HEMS no sólo dentro de la red local sino desde cualquier ubicación a través de Internet. Esto permitirá al usuario hacer los cambios remotamente cuando más le conviene y tener información de cómo se encuentra su hogar sin necesidad de encontrarse físicamente en este último. Como contrapartida, agregar esta funcionalidad requiere agregar controles de seguridad más robustos, ya que habilitar el acceso a través de Internet el sistema presentaría una mayor vulnerabilidad. Una alternativa para esto es utilizar direcciones IP fijas para cada uno de los hogares donde se instale el sistema. Además de ser necesarios mayores controles de seguridad, el hecho de tener que contar con direcciones IP fijas implicaría un costo adicional.

Capítulo 8. Evaluación del Sistema

Reducción de los costos de la plataforma

Al tratarse de un primer prototipo, no se tuvo en consideración el costo de la plataforma a la hora de elegir los componentes de hardware del sistema. En general, se optó por utilizar los componentes seleccionados en “*Controlador de energía domiciliario para una Red Eléctrica Inteligente*”[1]. Como se concluyó en el análisis económico, ver Sección 8.2, es necesario reducir los costos de la plataforma para hacer que el producto sea más conveniente para el usuario. Los motes Z1, utilizados en los Nodos de Carga y como *border router*, son los elementos más caros del sistema y dónde se cree que se debe poner mayor énfasis para reducir los costos.

Diseño de Nodos de Carga

El diseño y la implementación de Nodos de Carga no se encontraba dentro del alcance de este proyecto. Con el fin de poder ejecutar la prueba experimental fue necesario implementar Nodos de Carga básico para poder operar los electrodomésticos del laboratorio, ver Capítulo 7 y Apéndice L. Un trabajo futuro de gran interés es la implementación de Nodos de Carga capaces de operar una amplia gama de electrodomésticos y que presenten un costo razonable.

Comunicación HAN

El diseño de la red HAN tampoco se encontraba dentro del alcance de este proyecto y se utilizó la implementación realizada en “*Controlador de energía domiciliario para una Red Eléctrica Inteligente*”[1]. Queda para trabajos futuros la optimización de esta red. Así como el estudio de la posibilidad de cambiar el canal de comunicación inalámbrico por otro, como por ejemplo *PLC*.

Arquitectura Distribuida

Si bien consiste una gran reestructura de todo lo que se implementó en este proyecto, es preciso mencionar la posibilidad que existe de utilizar una arquitectura distribuida en el sistema HEMS.

Esto implicaría agregar en cada uno de los Nodos de Cargas el algoritmo de optimización de los electrodomésticos. Además se debería agregar algún procedimiento que asegure el cumplimiento de la restricción de potencia máxima que puede consumir el hogar. En caso que las decisiones individuales de los nodos no cumpla con la restricción se debe establecer algún tipo de negociación con fin de modificar estas decisiones.

Capítulo 9

Conclusiones

9.1. Conclusiones generales

En primer lugar se realizó un estudio de un gran número de algoritmos de gestión embebidos en el marco de las políticas Respuesta a la Demanda. Se realizó una comparación de los mismos, analizando qué problema atacaba cada uno, cuál era su enfoque para resolverlo y cómo podría adaptarse para ser utilizados en este proyecto. A raíz de dicho análisis, se seleccionó un algoritmo, que se consideró que era el que mejor se ajustaba a los objetivos del proyecto y se lo utilizó como base para el algoritmo implementado en el sistema HEMS.

Posteriormente, se escogió la tecnología a utilizar y se decidió cuál sería la arquitectura del sistema, realizando la implementación del algoritmo de gestión en los módulos de hardware correspondientes. Se logró obtener una arquitectura sólida en funcionamiento que, en función de los parámetros y variables obtenidas, ejecuta el algoritmo y transmite las órdenes de encendido o apagado a cada uno de los electrodoméstico en cada minuto.

Por otra parte, se realizó el diseño de una interfaz web para que el usuario pueda hacer uso del sistema de gestión. La misma brinda al usuario la posibilidad de configurar sus electrodomésticos a través de una aplicación web ya sea en una PC o un dispositivo móvil.

Se realizó una prueba experimental con electrodomésticos reales en el *'Laboratorio de Proyectos de Fin de Carrera'* del Instituto de Ingeniería Eléctrica de la Facultad. Para esto se adquirió un calefón y un lavasecarropa, se los instaló en el laboratorio y se realizaron algunas modificaciones a los mismos para que pudieran ser accionado remotamente. La prueba experimental realizada fue un éxito y si bien hubo que hacer adaptaciones en los electrodomésticos para que puedan ser operados por el sistema de gestión, se comprobó que distintos electrodomésticos pueden ser integrados al sistema sin mayores inconvenientes. La prueba experimental permitió verificar el correcto funcionamiento de todo el sistema en una situación próxima a la realidad de un hogar.

Se observa que tanto los nodos como los electrodomésticos quedaron instalados en el *'Laboratorio de Proyectos de Fin de Carrera'*. Por lo que se dispone de todo lo necesario para que el laboratorio continúe funcionando. También quedó pronta la instalación eléctrica para agregar un aire acondicionado al laboratorio. La intención es que este laboratorio sea utilizado en el futuro por otros proyectos, a fin de continuar con la investigación en la políticas de Respuesta a la Demanda en el sector residencial.

Adicionalmente, se realizó una simulación del funcionamiento del algoritmo de gestión en distintos hogares tipo. Esto permitió evaluar el desempeño y estimar el beneficio económico que obtiene el cliente al utilizar el HEMS. Se determinó que con la plataforma actual los clientes

Capítulo 9. Conclusiones

pueden recuperar la inversión inicial luego de cuatro años de instalado el sistema. Se entiende que este es un período de tiempo muy grande y es necesario reducirlo a fin de que el producto sea más atractivo para el cliente. La elección de los componentes de hardware del sistema se basó en facilidad de uso y no en el costo de los mismo. Por lo tanto se entiende que es posible reducir en gran medida los costos de la plataforma y así conseguir un producto más atractivo.

Finalmente, luego de una evaluación integral del sistema se concluyó que el desempeño del mismo es más que satisfactorio y cumple con los requerimientos planteados en el proyecto. Sin embargo, se considera que queda mucho por explotar en el tema y existen múltiples mejoras que se pueden abordar en un futuro, algunas de las cuales se describieron en la Sección 8.4.

9.2. Dificultades y aprendizajes

Mas allá de los objetivos específicos del proyecto, el mismo constituyó un gran aprendizaje en varios ámbitos. Entre ellos cabe destacar la experiencia de trabajar con un equipo interdisciplinario, en el cual existió un diálogo muy fluido entre sus integrantes, cooperando cada uno dentro de su área de conocimiento con el propósito de llegar a un producto final.

Por otra parte, este proyecto presentaba el desafío de complementar el trabajo realizado en el proyecto de fin de carrera “Controlador de energía domiciliario para una Red Eléctrica Inteligente”. En particular se deseaba utilizar la plataforma desarrollada en mencionado proyecto, incorporando a la misma del algoritmo diseñado. Finalmente, se consiguió acoplar la mayor parte de la plataforma, lo que evitó la necesidad de implementar una nueva, área con la cual el equipo no está familiarizado.

Otras de las dificultades enfrentadas fue el trabajo realizado sobre el hardware, entendiendo como hardware no sólo al propio sistema controlador sino también a los electrodomésticos. Fue necesario dedicar muchas horas del proyecto en este punto ya que el equipo de proyecto no contaba con formación en esta área.

Como última reflexión queremos valorar la importancia del trabajo en equipo que es el pilar fundamental para que un proyecto de estas características pueda ser llevado a cabo con éxito.

Apéndice A

Descripción de la ideas planteadas en [18]

El algoritmo de gestión utilizado en el sistema en el Nodo Controlador del sistema HEMS para gestionar los electrodomésticos está basado en la solución propuesta por Li, Chen y Low, en el trabajo “Optimal demand response based on utility maximization in power networks” [18].

En [18] los autores plantean un problema más general al abordado en este proyecto, en el cual la empresa suministradora y los clientes negocian con el fin de maximizar el beneficio global del sistema. Si bien esta es una idea interesante, el estudio de la misma quedó fuera del alcance de este proyecto. En la Sección A.1 se describe en detalle la solución propuesta por los autores, incluyendo la descripción de los algoritmos centralizados y distribuidos.

Adicionalmente, en el trabajo se planteó controlar un conjunto más amplio de electrodomésticos que el contemplado en este proyecto. En la Sección A.2 se describen los electrodomésticos contemplados y las funciones de utilidad propuestas.

A.1. Descripción de la solución propuesta

Escenario del problema

Para poder comprender el funcionamiento del algoritmo propuesto en [18] es necesario conocer primero en detalle el escenario propuesto. En la Figura A.1 se puede ver un diagrama del escenario del problema, los agentes principales y las variables más importantes.

Se propone que existe una empresa suministradora de energía, un conjunto de clientes y una comunicación bi-direccional entre la empresa y los clientes. Se utiliza el sub-índice i para referirse al i -ésimo cliente y N_i representa el número total de clientes.

El tiempo es dividido en *timeslots*, cada intervalo se denota con la letra t siendo $t \in [1, T]$ siendo N el horizonte de tiempo de optimización. Para fijar ideas se puede considerar que cada *timeslot* t representa una hora y que el horizonte de tiempo es un día, es decir $N = 24$.

El i -ésimo cliente posee un conjunto de electrodomésticos \mathcal{A}_i , se denota con el sub-índice a al electrodoméstico a -ésimo del conjunto \mathcal{A}_i . Cada electrodoméstico tiene un vector \mathbf{q}_{ia} asociado de largo T , que representa cuánta energía se le suministra al electrodoméstico en cada *timeslot* t . Una función de utilidad $U_{ia}(\mathbf{q}_{ia})$ que cuantifica cuánta satisfacción recibe el cliente del electrodoméstico a si al mismo se le suministra una energía \mathbf{q}_{ia} . Y un conjunto de restricciones $M^{i,a} \mathbf{q}_{i,a} \leq \eta_{i,a}$. En el caso de un calefón estas restricciones podrían ser, por ejemplo, la energía máxima que se le puede dar al mismo en un intervalo t y el rango de temperaturas admisibles del agua dentro del tanque.

Apéndice A. Descripción de la ideas planteadas en [18]

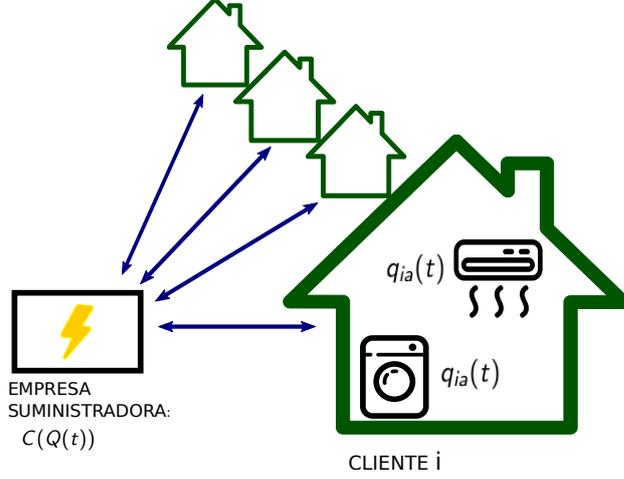


Figura A.1: Diagrama del escenario del problema

Además, se plantea la posibilidad de que el cliente cuente con una batería para almacenar energía. Se denota B_i como la capacidad de la batería, $b_i[t]$ el nivel de carga en el *timeslot* t y $r_i[t]$ la energía que se le entrega a la batería en t . Cabe destacar que si $r_i[t] > 0$ la batería se está cargando y si $r_i[t] < 0$ la batería está entregando energía. Las restricciones para una batería son: $0 \leq b_i[t] \leq B_i$ y $r_i^{min} \leq r_i[t] \leq r_i^{max}$. También se plantea que existe un costo $D(r_i)$ asociado al desgaste de la batería de acuerdo al patrón de uso de la misma.

$$b_i[t] = \sum_{\tau=0}^t r_i[\tau] + b[0]$$

Los electrodomésticos del cliente i , consumen un total de energía $Q_i[t]$ en cada *timeslot*, esta debe cumplir $0 \leq Q_i[t] \leq Q_i^{max}$. Donde Q_i^{max} está asociado con la potencia contratada del hogar.

$$Q_i[t] = \sum_{a \in \mathcal{A}_i} q_{ia}[t] + r_i[t]$$

Sea $Q[t]$ la energía total consumida por los N clientes, la empresa suministradora tiene un costo $C(Q[t])$ asociado a darle esta energía a los clientes.

$$Q[t] = \sum_{i=1}^{N_i} Q_i[t]$$

Algoritmo centralizado

Si se considera que la empresa suministradora no tiene fines de lucro, la misma no desea maximizar su beneficio sino que desea maximizar el beneficio global. En [18] se plantea que maximizar el beneficio global es equivalente a maximizar la utilidad de todos los electrodomésticos de los clientes y a la vez minimizar el costo en energía.

A.1. Descripción de la solución propuesta

$$\max_{\mathbf{q}_{ia}, \mathbf{r}_i} \sum_i \left(\sum_{a \in \mathcal{A}_i} U_{i,a}(\mathbf{q}_{i,a}) - D_i(\mathbf{r}_i) \right) - \sum_t C(Q[t]) \quad (\text{A.1})$$

Restricto a:

$$\begin{aligned} M^{ia} \mathbf{q}_{ia} &\leq \eta_{ia} && \forall a, i && \text{(Restricciones aparatos)} \\ 0 \leq Q_i[t] &\leq Q_i^{max} && \forall t, i && \text{(Potencia máxima del hogar)} \\ r_i^{min} &\leq r_i[t] \leq r_i^{max} && \forall i && \text{(Batería)} \end{aligned}$$

Bajo las siguientes hipótesis la solución del problema es única y puede ser hallada:

- Las funciones de utilidad $U_{ia}(\cdot)$ son funciones cóncavas.
- La función de costo de las baterías $D(\cdot)$ es convexa.
- El costo $C(\cdot)$ es una función convexa.
- El espacio donde viven las posibles soluciones del problema, dado por las restricciones del mismo, es un espacio convexo.

Si el problema se plantea de esta manera, la solución sólo puede ser hallada por la empresa suministradora, u otro agente externo, dado que es necesario conocer las funciones de utilidad y las restricciones de los todos aparatos de los clientes. Un algoritmo que resuelva este problema sería un algoritmo centralizado. Tal como fue mencionado con anterioridad, los algoritmos centralizados presentan la desventaja de que es posible que muchos clientes no estén dispuestos a divulgar la información necesaria para resolver el problema.

Algoritmo distribuido

Tal como fue mencionado el problema anterior sólo puede ser resuelto con un algoritmo centralizado lo cual presenta inconvenientes del punto de vista práctico. Sin embargo, gracias al teorema fundamental de la teoría económica del bienestar, y bajo las hipótesis descritas anteriormente, es posible dividir el problema centralizado en varios sub-problemas que pueden ser resueltos a nivel residencial.

Para esto es necesario agregar una variable $p[t]$ al problema que representa el precio que debe pagar el cliente por unidad de energía que consume en cada hora.

Es sencillo suponer que cada cliente desea maximizar de manera egoísta su propio bienestar. Es decir, desean maximizar la utilidad de sus electrodomésticos y a la vez minimizar el costo de la energía que consumen. Esto equivale a resolver el problema de optimización de la Ecuación A.2.

$$\max_{q_{ia}, r_i} \left(\sum_{a \in \mathcal{A}_i} U_{i,a}(q_{i,a}) - D_i(r_i) \right) - \sum_t p[t] Q_i[t] \quad (\text{A.2})$$

Restricto a:

$$\begin{aligned} M^{ia} q_{ia} &\leq \eta_{ia} && \forall a, i && \text{(Restricciones aparatos)} \\ 0 \leq Q_i[t] &\leq Q_i^{max} && \forall t, i && \text{(Potencia máxima del hogar)} \\ r_i &\in R_i && \forall i && \text{(Batería)} \end{aligned}$$

De acuerdo a la teoría económica del bienestar, se puede demostrar que existe $p^*[t]$ óptimo con el cual si todos los clientes actúan de manera egoísta maximizando su propio bienestar se llega a la solución de máximo bienestar global. Es decir, existe $p^*[t]$ que cumple que el conjunto de soluciones q_{ia} y r_i de la Ecuación A.2, también es solución de la Ecuación A.1.

Apéndice A. Descripción de la ideas planteadas en [18]

Negociación

En [18] se propone que el $p^* [t]$ óptimo puede ser hallado de manera iterativa mediante una negociación entre la empresa suministradora y los clientes.

Al comienzo de cada jornada la empresa suministradora predice cuál va a ser el consumo $Q_i^0 [t]$ de los clientes a lo largo del día. Con esta información calcula cuál es el costo de suministrar esta energía y fija $p^1 [t] = C (Q_i^0 [t])$. El precio $p^1 [t]$ es comunicado a los clientes, quienes responden con cuál sería su consumo $Q_i^1 [t]$ si ese fuera el precio de la energía. Con la nueva información, $Q_i^1 [t]$, la empresa obtiene una nueva curva de precios $p^2 [t] = C (Q_i^1 [t])$. Este proceso se repite iterativamente. Los autores en [18] afirman que es posible demostrar no sólo que este procedimiento converge, sino que converge a la curva de precios $p^* [t]$ óptima.

A.2. Funciones de utilidad

La función de utilidad es una herramienta de micro-economía, la cual es propuesta por los autores para representar el confort de los usuarios con el consumo de energía eléctrica. La misma modela la satisfacción que le produce a un consumidor racional el perfil de consumo de un electrodoméstico a lo largo del día. Es importante el concepto de perfil de consumo ya que depende del instante en el cuál el electrodoméstico consume energía si le es útil o no al usuario.

Las propiedades de estas funciones son las siguientes:

1. *Propiedad 1: no decreciente* $\frac{\partial U_{i,a}(q_{i,a})}{\partial q_{i,a}} \geq 0$

Implica que al usuario le da más satisfacción el electrodoméstico, cuanto más energía consume.

2. *Propiedad 2: saturación* $\frac{\partial^2 U_{i,a}(q_{i,a})}{\partial^2 q_{i,a}} \leq 0$

Es una limitante de la propiedad anterior, la cual significa que si bien el confort es creciente con la energía consumida tiende a una asíntota. Es una restricción del crecimiento de la utilidad.

3. *Propiedad 3: sin gasto no hay utilidad* $U_{i,a}(0) = 0$

Sin gastos de energía no hay satisfacción para el usuario

Un concepto que se debe tener en consideración es que la utilidad sólo modela el confort del usuario, no depende del precio de la energía.

Ejemplos funciones de utilidad

Mediante la función de utilidad se busca cuantificar la satisfacción que le produce un electrodoméstico a cada usuario. Los autores proponen una serie de funciones para distintos electrodomésticos. A continuación se describen estas funciones propuestas.

Tipo 1: Electrodomésticos de acondicionamiento térmico

En los electrodomésticos de tipo acondicionamiento térmico, como lo son el aire acondicionado y el calefón, la variable de interés para el usuario es la temperatura. En el caso de un aire acondicionado la temperatura de interés es la una habitación, en el caso de un calefón es la

A.2. Funciones de utilidad

temperatura del agua en el interior del tanque. Por esta razón los autores proponen que la función de utilidad de estos electrodomésticos sea de la forma de la Ecuación A.3.

$$U_{i,a}(q_{i,a}) = \sum_{t \in \mathcal{T}_{i,a}} U_{i,a} \left(T_{i,a} [t], T_{i,a}^{conf} [t] \right) \quad (\text{A.3})$$

Donde:

$T_{i,a} [t]$ es la temperatura de interés en el instante t .

$T_{i,a}^{conf} [t]$ es la temperatura deseada por el usuario.

$\mathcal{T}_{i,a}$ es el intervalo de tiempo de interés.

La función de utilidad de un electrodoméstico, tal como se mencionó anteriormente, es una función de q_{ia} , es decir $U_{ia} = f(q_{ia})$. La forma propuesta en la Ecuación A.3 a priori no parece cumplir con este requisito. Para llevarlo a esta forma es necesario establecer un vínculo entre $T_{i,a}$ y $q_{ia} [t]$, para esto los autores proponen utilizar la Ecuación A.4.

$$T_{ia} [t] = T_{ia} [t - 1] + \alpha (T_{ia}^{out} - T_{ia} [t - 1]) + \beta q_{ia} [t] \quad (\text{A.4})$$

Donde:

T_{ia}^{out} es la temperatura externa en el instante t .

α caracteriza cómo varía la temperatura con el intercambio de energía con el ambiente.

β caracteriza cómo varía la temperatura de acuerdo a la energía suministrada.

Los autores dan un ejemplo de esta función para un aire acondicionado, ver Ecuación A.5. En la Figura A.2 se encuentra una gráfica de esta función para $T_{i,a}^{conf} [t] = 25^\circ\text{C}$. Se observa como esta función penaliza que la temperatura de interés esté por encima de la temperatura de confort tanto como por debajo.

$$U_{i,a}(q_{i,a}) = \sum_{t \in \mathcal{T}_{i,a}} \left(c_{i,a} - b_{i,a} \left(T_{i,a} [t] - T_{i,a}^{conf} \right)^2 \right) \quad (\text{A.5})$$

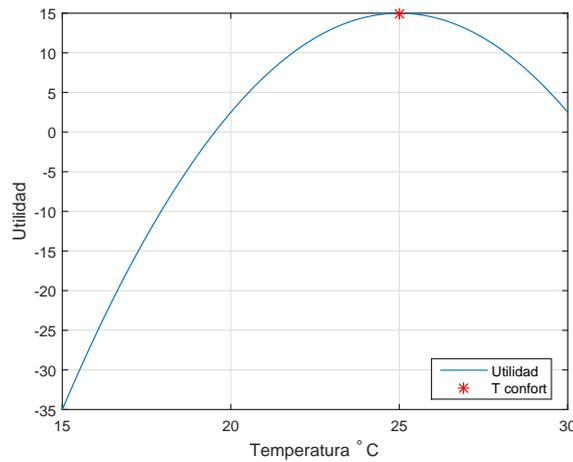


Figura A.2: Función de utilidad para aire acondicionado

Apéndice A. Descripción de la ideas planteadas en [18]

Tipo 2: Electrodomésticos de consumo discreto

Se le denomina cargas de consumo discreto a aquellas que deben finalizar su tarea antes de un determinado tiempo fijado por el usuario. La función de utilidad propuesta para este tipo de electrodomésticos esta dada por la Ecuación A.6.

$$U_{i,a}(q_{i,a}) = U_{i,a}(Q) = U_{i,a}\left(\sum_{t \in \mathcal{T}_{i,a}} q_{i,a}[t]\right) \quad (\text{A.6})$$

Más específicamente se propone utilizar la función de utilidad de la Ecuación A.7. En la Figura A.3 se encuentra una gráfica de esta función. Se observa que la misma no necesariamente pasa por cero.

$$U_{i,a}(Q) = c_{i,a} + b_{i,a}Q \quad (\text{A.7})$$

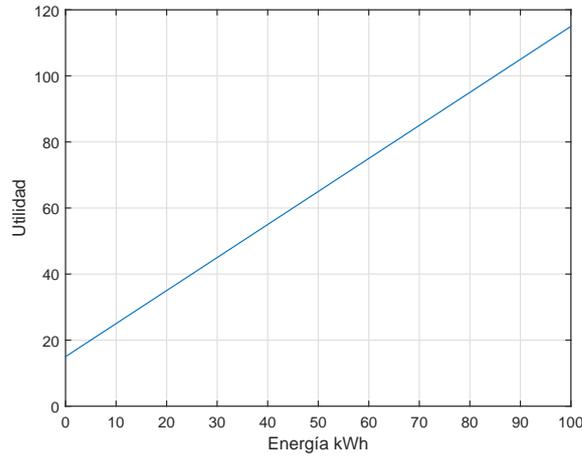


Figura A.3: Función de utilidad para electrodomésticos de consumo discreto

Tipo 3: Electrodomésticos de tipo Iluminación

Los autores proponen utilizar para este tipo de electrodomésticos una función de utilidad de la forma de la Ecuación A.8.

$$U_{i,a}(q_{i,a}) = \sum_t U_{i,a}(q_{i,a}[t], t) \underset{\text{porejemplo}}{=} c_{i,a} - \frac{1}{\left(b_{i,a} - \frac{q_{i,a}[t]}{q}\right)^{3/2}} \quad (\text{A.8})$$

Tipo 4: Electrodomésticos de tipo Entretenimiento

Los autores proponen utilizar la misma función de utilidad que para los electrodomésticos tipo 3.

Apéndice B

Detalle del modelado de los electrodomésticos

El sistema HEMS es el encargado de gestionar un conjunto de los electrodomésticos del hogar. Para poder agendar su consumo y comandarlos correctamente precisa conocer en detalle cómo es el funcionamiento de cada uno de los dispositivos.

En el Capítulo 4 se describió cuáles son los electrodomésticos contemplados dentro del sistema de gestión diseñado. Así como la función de utilidad y las restricciones utilizadas en cada uno de ellos. En el caso de los electrodomésticos acondicionamiento térmico, se encontró que es necesario disponer además de un modelo térmico que permita predecir la evolución de la temperatura.

Por otro lado, el módulo Intérprete precisa conocer el perfil de consumo de los electrodomésticos para traducir las decisiones del algoritmo a órdenes de encendido y apagado. Es decir, necesita contar con un modelo eléctrico para cada dispositivo.

El objetivo de este apéndice es describir los modelos térmicos y eléctricos utilizados para cada uno de los electrodomésticos. A fin de complementar lo mencionado en Capítulo 4 y Capítulo 5.

B.1. Modelos del aire acondicionado

Los sistemas de aire acondicionado (o HVAC) son unos de los dispositivos de acondicionamiento térmico más utilizados a nivel mundial. En particular en Uruguay, más de un 25 % de los hogares cuenta con un aire acondicionado de acuerdo a los últimos datos relevados por el Instituto Nacional de Estadística[30] .

La variable de interés en este dispositivo es la temperatura de la habitación en la que se encuentra instalado el equipo. Se decidió utilizar una función de utilidad y restricciones que dependen de la temperatura de la habitación. Esto llevó a la necesidad de disponer de un modelo térmico para poder expresar la utilidad y las restricciones en función de q_a .

A continuación se detallan los modelos, eléctrico y térmico, utilizados para el aire acondicionado. En este trabajo se buscaron modelos sencillos que describieran el funcionamiento con un grado de detalle acorde a las necesidades del sistema de gestión.

Apéndice B. Detalle del modelado de los electrodomésticos

Modelo eléctrico

El consumo eléctrico de los sistemas de aire acondicionado es complejo y difícil de modelar. Este depende de diversos factores, como el modo de funcionamiento y la tecnología del equipo, entre otros. Un modelo que contemple todos los factores no sólo sería complejo sino que además requeriría de datos que el usuario usualmente no dispone.

En general, el usuario selecciona el modelo de aire acondicionado que más se adecua a sus necesidades en función de la capacidad de térmica nominal y la eficiencia energética del equipo. La capacidad de calefacción/refrigeración indica cuánto calor el equipo entrega/extrae del ambiente. La misma se expresa usualmente en BTU/h y sus valores más típicos para uso residencial son: 9000, 12000 y 18000 BTU/h. La eficiencia energética, también llamada EER (Energy Efficiency Ratio), indica cuánto calor entrega/extrae el equipo del ambiente por unidad de potencia eléctrica consumida.

En este proyecto se modeló el consumo de un aire acondicionado como constante e igual a la potencia nominal del equipo, la cual puede ser calculada de acuerdo a la siguiente fórmula.

$$P[t] \simeq P_N = \frac{\text{capacidad calefacción/refrigeración}}{\text{eficiencia}(\eta)} \quad (\text{B.1})$$

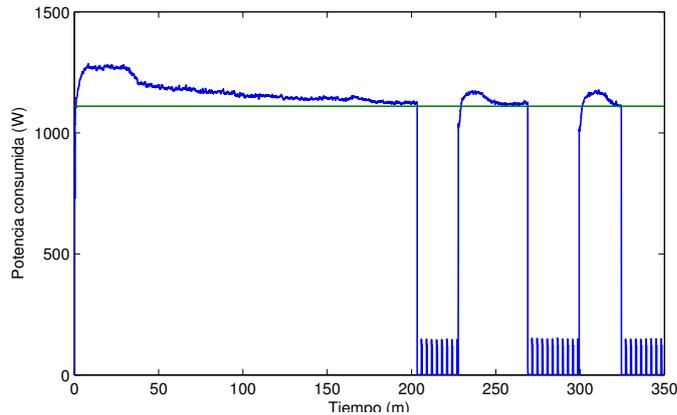


Figura B.1: Curva de consumo del aire acondicionado marca LG modelo LW1212-ER[50]

En Figura B.1 se encuentra un ejemplo de cómo es la curva de consumo de un aire acondicionado¹. En azul se encuentra la curva de consumo real relevada minuto a minuto y en verde la potencia nominal del producto declarada por el fabricante. Se observa que, si bien la curva azul presenta cierta variabilidad, la potencia nominal es una correcta aproximación cuando el equipo está encendido.

Modelo térmico

El modelo térmico de un aire acondicionado debe describir cómo evoluciona la temperatura en la habitación a lo largo del día, en función de cuánta energía entrega el aire acondicionado en

¹Esta curva corresponde a un aire acondicionado marca LG modelo: “LW1212-ER”. La misma fue extraída del trabajo [50], en el que se relevó el consumo de los electrodomésticos más comunes de un hogar. Esta información se encuentra disponible en www.ari.vt.edu/research-data/.

B.1. Modelos del aire acondicionado

cada *timeslot*. Se realizó una búsqueda bibliográfica y se encontró que existe una gran variedad de modelos propuestos.

La temperatura de una habitación depende de múltiples factores como por ejemplo, la energía que entrega el aire acondicionado, la temperatura exterior, la energía solar que ingresa por el techo y ventanas, filtraciones de aire, fuentes de calor internas como personas, luminarias u equipos electrónicos[51, 52, 53, 54]. Algunos de los modelos encontrados en la búsqueda bibliográfica contemplan todos o la mayoría de estos factores[52, 53]. Si bien estos modelos son completos, los mismos presentan la desventaja de ser complejos y requerir la determinación de una gran cantidad de parámetros.

Otros trabajos proponen modelos más sencillos, en los que se simplifica el problema considerando sólo la energía entregada por el aire acondicionado y la temperatura exterior[18, 51, 55, 56, 54]. Dentro de este conjunto, se destacan los modelos de parámetros térmicos equivalentes (ETP) en los cuales se modela el problema a través un circuito eléctrico[51, 54, 56].

Descripción del modelo seleccionado

El sistema de gestión no requiere predecir con gran precisión la evolución de la temperatura en la habitación. Tampoco dispone de información acerca de todos los factores que pueden afectar al sistema. Por estas dos razones se seleccionó el modelo propuesto en [54]. Donde se plantea el problema como un sencillo sistema lineal e invariante en el tiempo, cuya entrada es la potencia entregada por el aire acondicionado y la temperatura exterior.

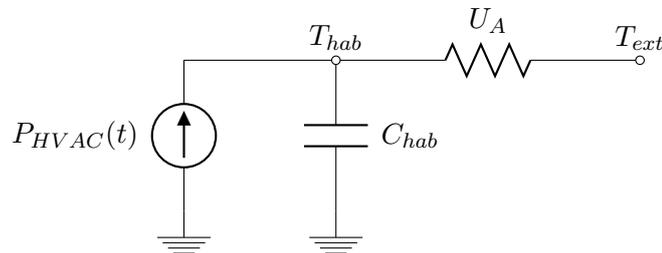


Figura B.2: Circuito eléctrico equivalente

El modelo seleccionado es un modelo de tipo de parámetros térmico equivalentes o ETP. En este tipo de modelos se establece una relación de equivalencia entre las variables termodinámicas del problema con variables eléctricas de un circuito equivalente, ver Figura B.2.

La temperatura de la habitación (T_{hab}) es representada como un voltaje en un nodo del circuito. La carga almacenada en la capacitancia C_{hab} modela la energía almacenada en la habitación. La fuente de corriente de valor P_{HVAC} representa la energía entregada al aire acondicionado. Se observa que P_{HVAC} puede tomar valores positivos o negativos dependiendo del modo de funcionamiento del aire acondicionado. Si el mismo calefacciona la habitación P_{HVAC} es positiva, y negativa en caso contrario. Por otra parte, la corriente que atraviesa la conductancia U_A modela las pérdidas de calor con el exterior.

Para que sea correcto modelar el problema con el circuito de la figura es necesario que se cumplan las siguientes hipótesis[54]:

- Se considera un único ambiente a acondicionar y que la temperatura es uniforme. Esto puede ser una habitación dentro del hogar que intercambia energía con el exterior a través de una pared y que se encuentra en equilibrio térmico con el resto de las habitaciones

Apéndice B. Detalle del modelado de los electrodomésticos

acondicionadas del hogar. O, puede considerarse todo el edificio como un ambiente en el cual no se consideran los efectos de circulación.

- Se asume que existe un único aire acondicionado en el ambiente.
- No se considera el control de la humedad del aire en la habitación.
- Se desprecian las fuentes de calor en el interior de la habitación.

Utilizando las leyes de Kirchoff en el circuito de la figura, es posible deducir la ecuación diferencial que rige cómo es la evolución de la temperatura de habitación.

$$\begin{cases} \dot{T}_{hab} = \frac{U_A}{C_{hab}}(T_{ext} - T_{hab}) + \frac{1}{C_{hab}}P_{HVAC}(t) \\ T(\tau_0) = T_{inicial} \end{cases}$$

En el sistema de gestión, el tiempo se encuentra discretizado en *timeslots*, por lo que es necesario pasar a una ecuación en diferencias. Utilizando la aproximación de derivada hacia adelante y se considerando que P_{HVAC} es constante en todo el *timeslot*, se obtiene el siguiente resultado.

$$\begin{cases} \frac{T_{hab}[t+1] - T_{hab}[t]}{\Delta t} = \frac{U_A}{C_{hab}}(T_{ext}[t] - T_{hab}[t]) + \frac{1}{C_{hab}}P_{HVAC}[t] \\ T_{hab}[-1] = T_{inicial} \end{cases}$$

Trabajando sobre esta ecuación se llega a la Ecuación 4.3. Se observa que esta expresión coincide con el modelo planteado en [18] para las cargas de acondicionamiento térmico. Se destaca que $q_{HVAC}[t]$ al igual que P_{HVAC} puede tomar valores positivos o negativos dependiendo del modo de funcionamiento del aire acondicionado.

$$T_{hab}[t+1] = T_{hab}[t] + \alpha(T_{ext}[t] - T_{hab}[t]) + \beta q_{HVAC}[t] \text{ donde: } \alpha = \frac{U_A \Delta t}{C_{hab}} \quad \beta = \frac{1}{C_{hab}}$$

En la Ecuación 4.3 la temperatura de la habitación está expresada en función del valor de la temperatura en el instante anterior. Es posible re-escribir esta expresión de manera que la temperatura en el *timeslot* t quede expresada en función de la temperatura en el instante inicial $T_{inicial}$ y los valores $q[t]$ de los *timeslots* anteriores.

$$T_{hab}[t] = (1 - \alpha)^{t+1} T_{inicial} + \sum_{k=0}^j (1 - \alpha)^{t-k} \alpha T_{ext}[k] + \sum_{k=0}^t (1 - \alpha)^{t-k} \beta q_a[k]$$

Identificación de los parámetros

Para que el modelo funcione correctamente es muy importante identificar adecuadamente los valores de parámetros U_A y C_{hab} del sistema.

Dichos parámetros pueden ser hallados en forma teórica o mediante un experimento. La determinación teórica de los mismos suele ser trabajosa y requiere conocer en detalle cómo es el mobiliario presente en la habitación, los materiales de construcción utilizados, entre otras cosas. Por consiguiente, los parámetros suelen obtenerse de forma experimental. A continuación se describe el experimento realizado para obtener los parámetros para una habitación de un hogar.

Descripción del experimento

La habitación para la cual se hallaron los parámetros U_A y C_{hab} fue el dormitorio principal de una casa. El mismo tiene una dimensiones de $4 \times 4,8 \times 3,2$ metros. Sólo una de las paredes

B.1. Modelos del aire acondicionado

del dormitorio da al exterior y la misma cuenta con un ventanal de $2,5 \times 1,5$ metros. En la misma se encuentra instalado un aire acondicionado tipo split de 9000 BTU marca Fujitsu. En la tabla B.1 se describen las principales características del equipo.

Características	Modo de operación	
	Frío	Calor
Cap. térmica [BTU/h]	8500	10900
Cap. térmica [kW]	2.49	3.19
Eficiencia [kW/kW]	3.97	4.27
P_N [kW]	0.627	0.748

Tabla B.1: Especificaciones del aire acondicionado Fujitsu modelo ASYA09LG.

Se utilizaron dos Nodos de Carga para medir la temperatura ambiente en dos puntos de la habitación². El primer nodo fue colocado sobre una mesa cercana al aire acondicionado y el otro en un armario alejado del aire acondicionado. Se configuró el Nodo Controlador para que consulte las mediciones de ambos nodos cada 30 segundos y almacene las mismas.

El experimento se realizó una mañana de verano con una temperatura exterior $T_{ext} = 27^\circ\text{C}$. El procedimiento fue el siguiente:

1. Se registró la temperatura de la habitación con el equipo de aire acondicionado apagado durante 8 minutos.
2. Se encendió el aire en modo refrigeración y se registraron las medidas durante 75 minutos aproximadamente. La temperatura objetivo del aire se configuró en 17°C de modo de asegurar que el dispositivo no detuviese su funcionamiento durante esta fase.
3. Se apagó el aire acondicionado y se registró como evolucionaba la temperatura de la habitación durante 100 minutos.

Resultados del experimento

En la Figura B.3 se puede observar cuál fue la temperatura registrada por ambos nodos durante el experimento, así como la energía eléctrica consumida por el aire acondicionado de acuerdo al modelo eléctrico propuesto.

Los valores registrados por ambos nodos coinciden cuando el sistema se encuentra en régimen, es decir al inicio y al final del experimento. Sin embargo, se observa que las curvas presentan comportamientos distintos durante los periodos transitorios. La temperatura registrada por el nodo sobre la mesa presenta un descenso o ascenso mucho más rápido cuando se enciende o apaga el aire acondicionado. Se entiende que esta diferencia se debe a la ubicación de los nodos. El nodo sobre la mesa estaba expuesto a la corriente de aire frío que genera el aire acondicionado y es por esto que su temperatura desciende muy rápidamente. El nodo en el armario no estaba expuesto a esta corriente y es por esto que su temperatura varía más lentamente.

Como era de esperar la temperatura dentro de la habitación no es uniforme. Es deseable que el electrodoméstico controle la temperatura media de la habitación, que es dónde se ubica el usuario. Si se controla la temperatura en el nodo de la mesa, el impacto del flujo de aire hace que los efectos transitorios sean muy grandes lo que provoca la pérdida de validez del modelo matemático utilizado. Al utilizar la temperatura del nodo sobre el armario, no se evidencia el

²La descripción de cómo fueron implementados estos nodos se encuentra en el Apéndice L.

Apéndice B. Detalle del modelado de los electrodomésticos

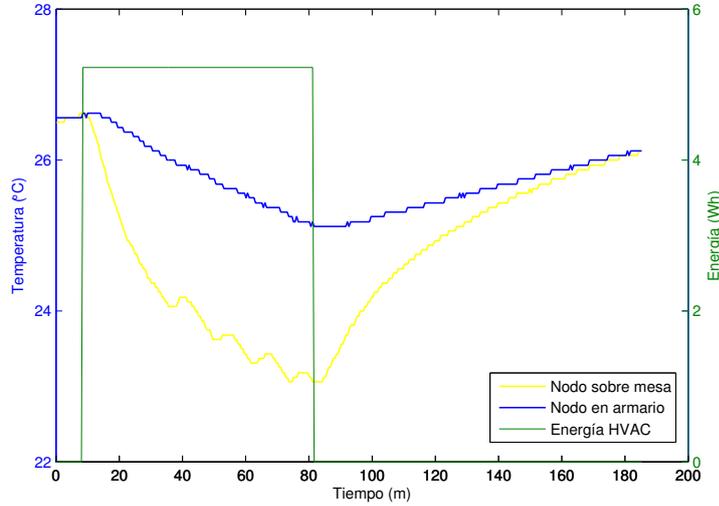


Figura B.3: Valores de temperatura registrados por los nodos durante el experimento

efecto del flujo de temperatura, pero en contrapartida se controla una temperatura que no es la que experimenta el usuario. Por estas razones, se utilizó el promedio de los valores registrados por los nodos para extraer los datos del modelo. La curva de temperatura promedio también se puede apreciar en la Figura B.3. Sin embargo, debe tenerse en cuenta que el modelo obtenido no es representativo en todos los puntos de la habitación.

Para identificar los parámetros U_A y C_{hab} del modelo, se buscó el par de valores que minimizaba la suma de los cuadrados de las distancias entre las mediciones obtenidas y las predicciones del modelo. En la Tabla B.2 se encuentran los valores obtenidos.

Parámetro	Valor
U_A	476 W/°C
C_{hab}	939 Wh/°C

Tabla B.2: Valores de U_A y C_{hab} obtenidos para el aire acondicionado

En la Figura B.4 se puede ver una comparación entre el promedio de los registros reales de temperatura y los valores obtenidos utilizando el modelo con estos parámetros. Se observa que los valores obtenidos son adecuados dado que la curva de registros es muy próxima a los valores puntuales del modelo.

Para identificar los valores de U_A y C_{hab} se tomó un *timeslot* de 30 minutos. Estos dos parámetros son característicos del sistema, mientras que α y β dependen del *timeslot* seleccionado. En la Tabla B.3 se muestran los valores de α y β obtenidos para el *timeslot* correspondiente.

Temperatura de confort

El confort térmico no sólo depende de la temperatura del aire dentro de la habitación sino que se ve afectado por otros factores como la humedad y la velocidad del aire. Además,

B.2. Modelo del calefón

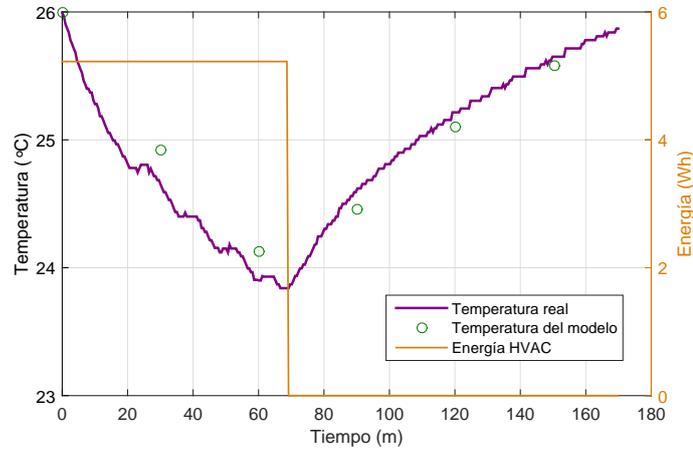


Figura B.4: Comparación del modelo obtenido con los valores reales de temperatura

<i>Timeslot</i>	30 min
α [adim]	0.2534
β [$^{\circ}\text{C}/\text{Wh}$]	$1,1 \times 10^{-3}$

Tabla B.3: Valores de α y β obtenidos.

este también depende de factores que varían según la persona como la vestimenta, nivel de actividad, el metabolismo e incluso factores psicológicos. Por esta razón es difícil establecer un valor de temperatura de confort para todas las personas.

Existen recomendaciones acerca de cuál debe ser la temperatura en el interior de un edificio para alcanzar el confort térmico de sus ocupantes. Estas se realizan tomando rangos de temperatura dentro de los cuales un determinado porcentaje de personas se encuentran a gusto. En particular el estándar “ASHRAE Standard 55 - *Thermal Environmental Conditions for Human Occupancy*” establece cuál debe ser la combinación de factores necesarios para obtener confort térmico en interiores. De acuerdo a [57], este estándar establece que el rango de confort en invierno es de 20°C a 24°C y de 22°C a 26°C en verano. Estos valores están basados en un ambiente con 60 % de humedad, sin corriente de aire, nivel de actividad de 1.2 met y una vestimenta acorde a la estación del año.

Dado que la temperatura de confort depende de cada persona, el sistema de gestión permite que el usuarios configure la misma a su gusto. Sin embargo, es conveniente que el usuario siga las recomendaciones mencionadas en el párrafo anterior.

B.2. Modelo del calefón

El calefón es el dispositivo más utilizado en los hogares para el calentamiento de agua. Según encuestas realizadas en el 2013, cerca de una 82 % de los hogares uruguayos cuentan con al menos un calefón[30]. Al igual que en el aire acondicionado, en el caso de un calefón es necesario contar con un modelo eléctrico y un modelo térmico.

Apéndice B. Detalle del modelado de los electrodomésticos

Modelo eléctrico

Se propuso modelar el calefón como una simple resistencia. Cada vez que se enciende el calefón el mismo consume una potencia constante igual a la potencia nominal del calefón. En Uruguay un valor típico de potencia nominal es $1000W$.

$$P(t) = P_N$$

Por otro lado, se observa que se consideró que la eficiencia del calefón es $\eta = 1$. Lo cual implica que toda la energía eléctrica que se entrega al calefón es transferida al agua en forma de calor.

Modelo térmico

El modelo térmico para un calefón debe permitir predecir la temperatura de agua dentro del tanque en función de la energía suministrada y la demanda de agua caliente del usuario.

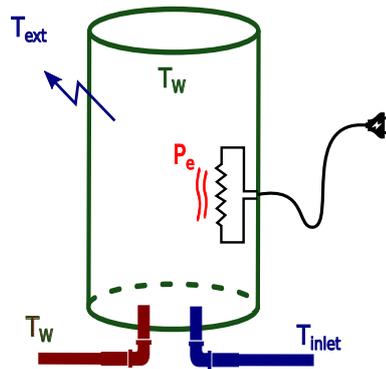


Figura B.5: Modelado del calefón

En la Figura B.5 se muestra un esquema de las magnitudes de relevancia del sistema. Las mismas representan:

- T_w - Temperatura del agua en el interior del tanque.
- P_e - Potencia de origen eléctrico entregada al agua.
- T_{ext} - Temperatura exterior al tanque.
- T_{inlet} - Temperatura del agua entrante al tanque.

Modelo sin consumo de agua caliente:

En primer lugar, se consideró un modelo simplificado en el cual no se extrae agua del tanque. Para esto es usual utilizar un modelo de parámetros térmicos equivalentes (ETP - 'Equivalent Thermal Parameters')[31][22].

El circuito equivalente es idéntico al utilizado para el aire acondicionado, ver Figura B.6. La única diferencia en este caso es que P_e es siempre mayor o igual a cero. Mediante un razonamiento análogo al realizado para el aire acondicionado es posible obtener la expresión de

la Ecuación B.2.

$$T_W[t+1] = T_W[t] + \alpha (T_{ext}[t] - T_W[t]) + \beta q_e[t]$$

donde :

$$\alpha = \frac{U_A \Delta t}{C_{WH}} \quad \text{(B.2)}$$

$$\beta = \frac{1}{C_{WH}}$$

Modelo con consumo de agua caliente

El modelo de la Ecuación 4.6 no es completo dado que no considera la demanda de agua caliente del usuario. Se realizó una revisión bibliográfica acerca de cómo modelar la evolución de la temperatura del agua cuando existen extracciones.

Los autores en [22] proponen utilizar la ecuación de mezcla de dos líquidos para determinar cuál es la temperatura del agua resultante luego de una ducha. Por otro lado en [31] se consideran dos modelos distintos, uno en el que se toma el calefón como una masa de agua homogénea cuya temperatura descende a medida que se extrae agua del tanque. Y un segundo modelo que considera que existen dos masas de agua dentro del tanque una caliente y otra fría. Cuando el usuario extrae agua del tanque el volumen de la masa de agua fría crece.

El algoritmo de gestión debe asegurar que el usuario pueda bañarse confortablemente. Bajo la hipótesis de que el calefón se encuentra correctamente dimensionado, se puede asumir que si la temperatura del agua es adecuada en el instante inicial del baño, luego el usuario va a poder bañarse sin inconveniente. Por esta razón, no es de interés modelar con precisión cómo evoluciona la temperatura del agua mientras la persona se baña. Pero si se debe conocer cuál es la temperatura resultante luego del baño. Considerando lo anterior se optó por utilizar el modelo de propuesto en [22] dado que es el más sencillo de los tres modelos.

Modelo seleccionado:

El modelo utilizado para predecir la evolución de la temperatura del agua dentro del tanque es el siguiente. Se asume que la temperatura del agua caliente evoluciona según la Ecuación 4.6 a lo largo del *timeslot*. Y luego al finalizar el mismo se quita instantáneamente el volumen de agua caliente consumida. De esta manera, la temperatura al finalizar el *timeslot* puede ser

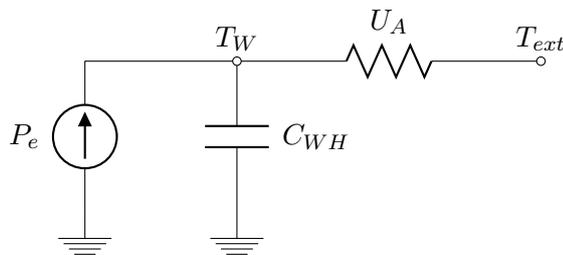


Figura B.6: Circuito eléctrico equivalente

Apéndice B. Detalle del modelado de los electrodomésticos

obtenida utilizando la siguiente ecuación:

$$\begin{aligned}
 T_W[t] &= \left(T_W[t-1] + \alpha (T_{ext} - T_W[t-1]) + \beta q[t] \right) \left(\frac{V_{total} - \Delta V[t]}{V_{total}} \right) \\
 &\quad + T_{inlet} \frac{\Delta V[t]}{V_{total}} \\
 &= \left(T_W[t] + \alpha (T_{ext} - T_W[t]) + \beta q[t] \right) (1 - \delta_V[t]) + T_{inlet} \delta_V[t]
 \end{aligned} \tag{B.3}$$

En la Ecuación B.3 la temperatura del agua está expresada en función del valor de la temperatura en el instante anterior. Es posible re-escribir esta expresión de manera que la temperatura en el *timeslot* t quede expresada en función de la temperatura en el instante inicial T_{W0} y los valores de $q[t]$ de los *timeslot* anteriores.

$$\begin{aligned}
 T_W[t] &= (1 - \alpha)^{t+1} \prod_{i=0}^t (1 - \delta_V[i]) T_{W0} \\
 &\quad + \sum_{i=0}^t \left[(1 - \alpha)^{t-i} \prod_{j=i}^t (1 - \delta_V[j]) (\alpha T_{ext} + \beta \eta q[i]) \right] \\
 &\quad + \sum_{i=0}^t \left[(1 - \alpha)^{t-i} \prod_{j=i+1}^t (1 - \delta_V[j]) \delta_V[i] T_{inlet} \right]
 \end{aligned}$$

Identificación de los parámetros del modelo

El modelo seleccionado requiere identificar los valores de los parámetros C_{WH} y U_A , conocer el volumen de agua que se extrae en cada *timeslot* ($\delta_V[t]$) y conocer cuál es la temperatura del agua fría que ingresa al calefón (T_{inlet}). A seguir se explica cómo se determinaron estos parámetros para el calefón utilizado en la prueba experimental.

Identificación de las constantes C_{WH} y U_A

Los valores de las constantes C_{WH} y U_A fueron obtenidos experimentalmente mediante un ensayo. A continuación se describe el experimento realizado para obtener estas constantes para el calefón utilizado en la prueba experimental del sistema y los resultados obtenidos.

Descripción del experimento

El calefón utilizado en la prueba experimental presenta las siguientes características:

Características del calefón	
Marca	Orion
Volumen	45 lts.
Potencia nominal	1000 W

Se utilizó un Nodo de Carga para medir la temperatura del agua del calefón³. Se configuró el Nodo Controlador para que consultara las medidas obtenidas cada treinta segundos y las almacenara.

³En el Sección L.4 se describe cómo fue implementado este nodo.

B.2. Modelo del calefón

La temperatura ambiente durante el experimento era $26,4\text{ }^\circ\text{C}$. El procedimiento realizado fue el siguiente:

1. Inicialmente el agua dentro del calefón se encontraba a $30\text{ }^\circ\text{C}$ aproximadamente.
2. Se mantuvo encendido el calefón hasta que el termostato cortó la energía⁴.
3. Se mantuvo apagado el calefón y se registró la temperatura de las siguientes doce horas.

Se observa que durante el experimento no se extrajo agua del calefón por lo que es válida la siguiente ecuación:

$$T_W[t + 1] = T_W[t] + \alpha (T_{ext}[t] - T_W[t]) + \beta q_e[t]$$

donde:

$$\alpha = \frac{U_A \Delta t}{C_{WH}} \quad \beta = \frac{1}{C_{WH}}$$

Resultados obtenidos

Para determinar los valores de C_{WH} y U_A se procedió de forma análoga al HVAC, buscando minimizar la suma cuadrática de las diferencias entre los datos experimentales y la curva paramétrica. Los valores obtenidos se muestran en la Tabla B.4, se incluyen los valores por unidad correspondientes a la conductancia térmica por unidad de área y la capacitancia térmica por unidad de volumen.

Parámetro	Valor	Valor pu
U_A	$1.08\text{ W}/^\circ\text{C}$	$1.2744\text{ W}/(\text{m}^2\text{ }^\circ\text{C})$
C_{WH}	$44.5\text{ Wh}/^\circ\text{C}$	$0.9883\text{ Wh}/(\text{l }^\circ\text{C})$

Tabla B.4: Valores de U_A y C_{WH} obtenidos para el calefón

En la Figura B.7 se encuentra la gráfica de modelo matemático predictivo de la temperatura con los parámetros obtenidos, contrastados con la medida real de temperatura. En la misma se aprecia que el modelo y los parámetros obtenidos describen correctamente cómo evoluciona la temperatura del agua dentro del calefón.

Uso de agua caliente ($\delta_V[t]$)

La variable $\delta_V[t]$ representa el porcentaje de volumen de agua caliente que se extrae del calefón en cada *timeslot*. Esta variable permite determinar cuánta energía pierde el calefón cuando se extrae agua del tanque. En la Tabla B.5 se muestran el volumen de agua caliente utilizado para los usos más típicos⁵.

$$\delta_V[t] = \frac{\Delta V[t]}{V_{total}} \quad (\text{B.4})$$

⁴Se observó que la temperatura máxima soportada por el calefón utilizado es $63\text{ }^\circ\text{C}$, bastante inferior a la que indica el termostato ($75\text{ }^\circ\text{C}$)

⁵El volumen de la ducha fue calculado tomando: temperatura agua fría $15\text{ }^\circ\text{C}$, temperatura agua caliente $55\text{ }^\circ\text{C}$, temperatura del agua utilizada $40\text{ }^\circ\text{C}$. El resto de los valores fueron extraídos de [58].

Apéndice B. Detalle del modelado de los electrodomésticos

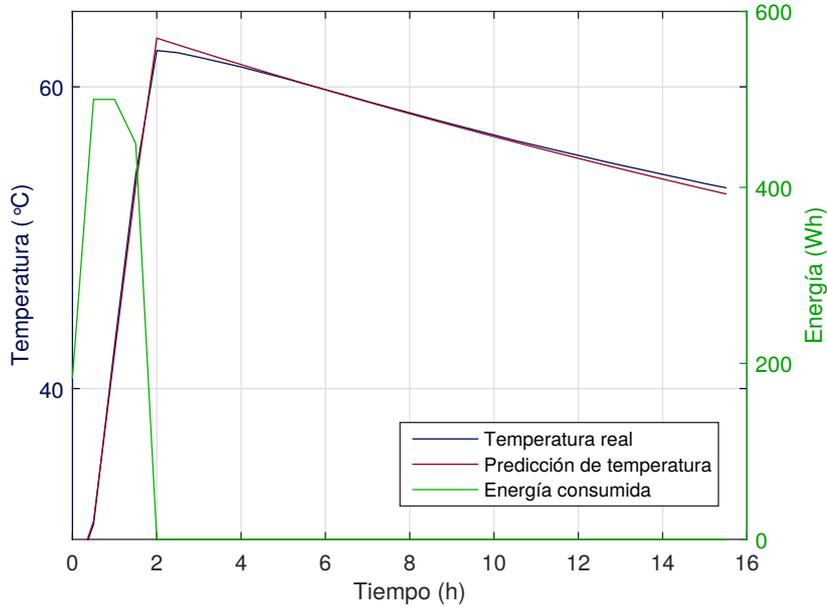


Figura B.7: Resultados experimentales del ajuste del calentador

Actividad	δ_V (l)
Ducha (8min)	35
Lavar manos	0.67
Fregar platos a mano	20

Tabla B.5: Uso de agua caliente según actividad

Temperatura agua fría (T_{inlet})

Es de público conocimiento que el gasto energético del calentador es menor en verano que en invierno. Esto se debe a tres razones, el usuario está dispuesto a bañarse con agua a menor temperatura, las disipaciones del tanque son menores porque T_{ext} es más grande y porque la temperatura de agua fría aumenta. Siendo esta última la razón más relevante.

En la Tabla B.6 se puede observar cómo varía la temperatura del agua suministrada por OSE a lo largo del año en el departamento de Montevideo[47]. Se observa que el agua fría presenta una gran variación a lo largo del año, el mínimo se da en Agosto con un promedio de temperatura de 9,2°C mientras que el máximo es 28,5°C y ocurre en el mes de Febrero.

Mes	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Set	Oct	Nov	Dic
Temp. [°C]	28.4	28.5	26.1	21.7	16.6	12.0	9.3	9.2	11.7	16.1	21.3	25.8

Tabla B.6: Temperatura de agua fría para Montevideo. (Fuente: OSE [47]).

En la Figura B.8 se observa un ejemplo de la evolución de la temperatura del agua dentro del calentador de acuerdo al modelo seleccionado. Se observa que la temperatura luego de que

B.3. Modelo de electrodomésticos de consumo discreto

el usuario se bañe, es decir a las 20hs, depende fuertemente de la temperatura del agua fría que ingresa al calefón. En la figura se puede observar el resultado para $T_{inlet} = 28^{\circ}\text{C}$ (verano) y 10°C (invierno).

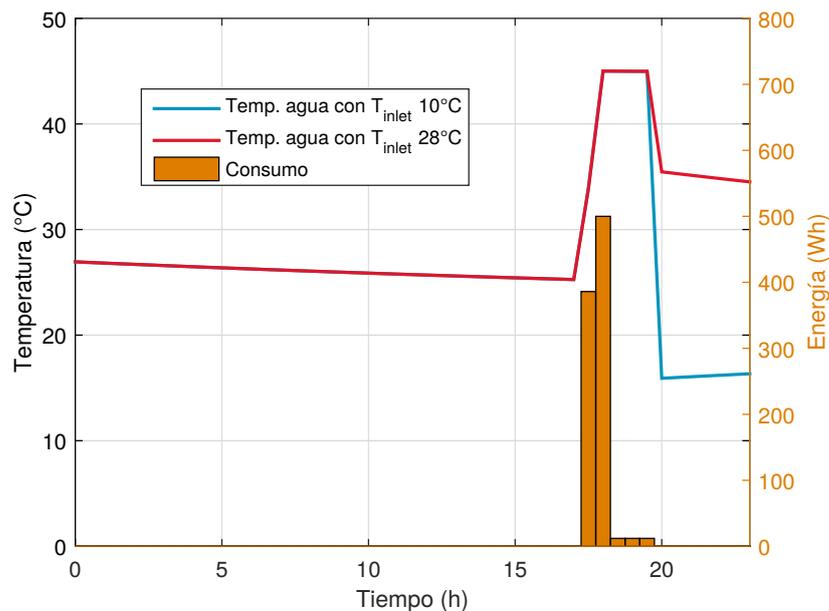


Figura B.8: Ejemplo de la evolución de la temperatura del agua en el calefón para invierno y verano

B.3. Modelo de electrodomésticos de consumo discreto

Los electrodomésticos de consumo discreto son aquellos que ejecutan una tarea específica. En los cuales el usuario desea que la tarea finalice antes de una hora determinada, pero no le afecta el momento específico en el cual se activa el dispositivo.

Cada tarea que ejecuta el electrodomésticos presenta un tiempo de funcionamiento asociado o una cantidad de energía asociada. Dentro de esta clase electrodomésticos se distinguen dos tipos:

- **Electrodomésticos con tarea de duración fija.**

El tiempo de duración es fijo y por lo tanto la energía necesaria para realizar la tarea también lo es. El sistema de gestión puede decidir en qué horas suministrarle energía a estos electrodomésticos. Dentro de esta categoría se encuentran el lavarropas y el lavavajillas.

- **Electrodomésticos con tarea de duración flexible.**

Estos electrodomésticos pueden operar durante una cantidad de tiempo variable. El sistemas de gestión puede decidir el total de energía a suministrar y en qué horas hacerlo. El usuario puede fijar los valores mínimos y máximos de energía. Dentro de esta categoría se encuentra el bombeo de agua para regar un jardín, sistema de bombeo de agua para una piscina o la carga de la batería de un auto eléctrico.

Apéndice B. Detalle del modelado de los electrodomésticos

Modelo eléctrico

Dada la gran variedad de electrodomésticos presentes en esta categoría no es posible caracterizar de forma genérica el consumo de los mismos. Pero se observó que en general existen dos comportamientos distintos.

En primer lugar, existen electrodomésticos que presentan una curva de consumo plana. Este es el caso la bomba de riego si se desprecian los transitorios de arranque.

$$P[t] = P_N$$

Por otro lado, existen electrodomésticos con curvas de consumo más compleja. Es común que la tarea a ejecutar presente varios procesos que se ejecutan secuencialmente y cada uno de estos procesos presentan un consumo distinto. Este es el caso de un lavarropas, el cual puede presentar ciclos de lavado, enjuague y centrifugado.

En Figura B.9 se encuentra la curva de consumo de un lavarropas durante un programa de lavado normal⁶. En la misma se puede ver que se presentan varios niveles de consumo diferentes, los cuales están asociados a distintas etapas del proceso de lavado.

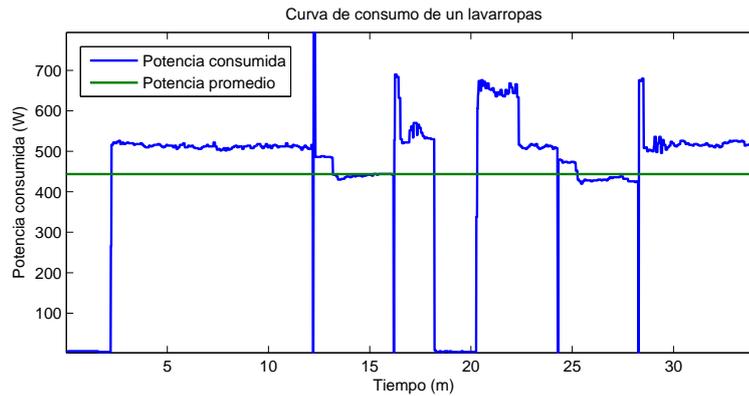


Figura B.9: Curva de consumo del lavarropas marca General Electrics modelo WSM2420D3WW [50]

En el marco del algoritmo de gestión, el consumo de estos electrodomésticos fue aproximado por la potencia promedio que consume.

$$P[t] = P_{prom}$$

El Intérprete requiere tener un mayor conocimiento del perfil de consumo del electrodoméstico. Dado que no es posible caracterizar el consumo de estos electrodomésticos de forma genérica, es necesario relevar experimentalmente el perfil de consumo. A continuación se describe el experimento realizado para caracterizar el consumo del lavasecarropas y los resultados obtenidos.

B.4. Perfil de consumo lavasecarropas

Se utilizó un lavasecarropas en la prueba experimental del sistema de gestión. No se disponía información detallada acerca del perfil de consumo del dispositivo. Por lo cual, fue necesario

⁶Esta curva fue extraída del trabajo [50], en el que se relevó el consumo de los electrodomésticos más comunes de un hogar. Esta información se encuentra disponible en www.ari.vt.edu/research-data/.

B.4. Perfil de consumo lavasecarropas

obtener sus características y perfil de consumo experimentalmente.

El lavasecarropas utilizado presenta múltiples programas de lavado. Cada programa presenta múltiples procesos que se ejecutan secuencialmente. Y cada uno de los procesos tiene su propia demanda de energía eléctrica. Dependiendo del programa de lavado se determina cuáles son los procesos que se ejecutan y la duración de los mismos. Es decir, cada programa de lavado del lavasecarropas presenta distintas características de consumo. En el marco de este proyecto se relevó el perfil de consumo de un programa de lavado completo, el cual utiliza agua caliente y seca la ropa.

Para relevar las características de consumo se programó el lavasecarropas para que realice el programa de lavado seleccionado. Utilizando un pinza amperimétrica, se registró la potencia consumida por electrodoméstico cada 30 segundos. En la Tabla B.7 se encuentra un resumen de los resultados obtenidos y en la Figura B.10 se muestra el perfil de consumo relevado .

Características del lavasecarropas	
Duración de la tarea	147 min.
Energía total	1949 Wh
Energía promedio por hora	795 Wh

Tabla B.7: Características del perfil de consumo del lavasecarropas en un ciclo de lavado completo

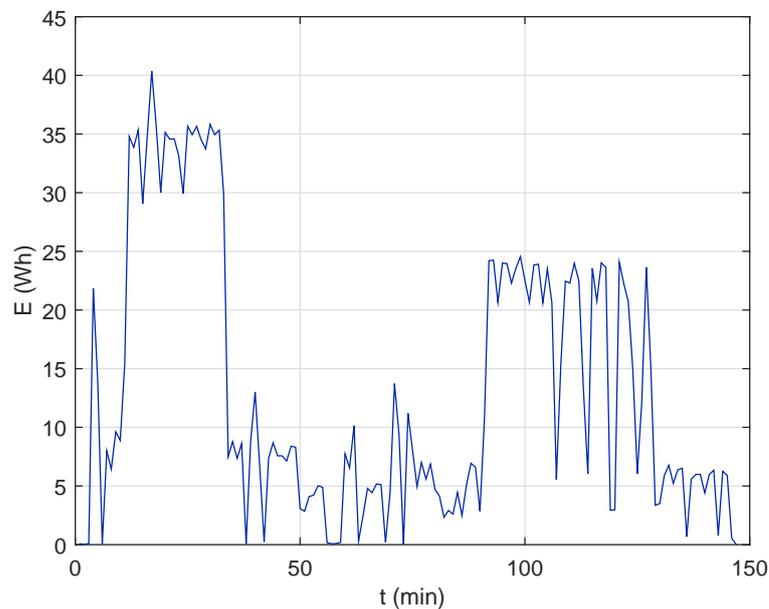


Figura B.10: Perfil energético del lavasecarropas en un ciclo de lavado completo

Apéndice C

Determinación de los parámetros de la función de utilidad

En el Capítulo 4 se describieron las funciones de utilidad utilizadas en los electrodomésticos. Las mismas presentan parámetros cuyos valores deben ser ajustados adecuadamente de modo que el algoritmo funcione correctamente.

Significado de los parámetros

Todas las funciones de utilidad seleccionadas presentan la siguiente forma:

$$U(\mathbf{q}_a) = b + af(\mathbf{q}_a)$$

con: $a \geq 0$

El término independiente b permite ajustar el nivel de utilidad de la función de utilidad. El mismo se utiliza para que las funciones de utilidad cumplan con las propiedades que éstas presentan¹. La cual establece que sin gasto la función de utilidad debe ser nula: $U(\mathbf{q}_a = \vec{0}) = 0$.

Por otro lado, el parámetro a permite ajustar la amplitud de las variaciones de la utilidad, y determina el peso que toma la utilidad frente al costo en el balance que realiza el algoritmo de gestión. Cuanto mayor es a más peso tiene la utilidad en el balance.

Si el valor de a es muy grande, una pequeña variación en \mathbf{q}_a representa una gran variación de la utilidad lo que lleva a que la variación en el costo sea despreciable. Esto hace que el algoritmo tome las decisiones basándose únicamente en la utilidad. En caso contrario, si a es muy pequeño, el algoritmo sólo busca minimizar los costos sin importar el confort del usuario.

Es importante destacar que no existe un ajuste óptimo de a que sirva para todos los usuarios. Algunos usuarios prefieren priorizar el confort en el balance y otros priorizar el ahorro, lo cual puede ser logrado con pequeñas variaciones del valor de a de cada electrodoméstico. La preferencia del usuario incluso puede variar entre un electrodoméstico y otro, o en función de la época del año. Una posibilidad es que los valores de a de cada electrodoméstico se ajusten automáticamente de acuerdo a la realimentación del usuario. El cual podría indicar diariamente su grado de satisfacción con cada electrodoméstico o el objetivo de ahorro mensual deseado. Esta es una solución interesante pero compleja de llevar a cabo que queda para ser futuros trabajos en el tema.

¹Más específicamente para que cumpla con la propiedad número tres, descrita en el Sección A.2.

Apéndice C. Determinación de los parámetros de la función de utilidad

A la hora de evaluar el comportamiento del algoritmo se utilizaron valores de a que se entendieron adecuados para un usuario promedio. A continuación se detallan los valores de a y b utilizados y cómo se llegó a los mismos.

Cargas discretas

Se recuerda que la función de utilidad utilizada en este tipo de cargas es la siguiente:

$$U(\mathbf{q}_a) = b + a \sum_{t \in \mathcal{T}_a} q_a[t]$$

Parámetro b :

Se tomó $b = 0$ de forma que $U(\mathbf{q}_a = \vec{0}) = 0$.

Parámetro a :

Se distinguen dos casos de acuerdo al tipo de electrodoméstico:

- **Tarea de duración fija.**

En este tipo de electrodomésticos requiere de una determinada cantidad de energía para cumplir con su tarea. La satisfacción del usuario no aumenta si se le entrega más energía que la necesaria para realizar la tarea. Por ejemplo, el usuario no recibe mayor satisfacción si el algoritmo decide suministrar a un lavarropas la energía asociada a un lavado y medio. Por esto se tomó $a = 0$ en este tipo de electrodomésticos.

Se observa que con $a = 0$ y $b = 0$ la utilidad de los electrodomésticos es nula. Por lo que el algoritmo de optimización sólo busca minimizar los costos y cumplir con las restricciones impuestas.

- **Tarea de duración variable.**

En este caso, el total de energía a suministrar puede variar dentro de un rango. Se observa que la función de beneficio de estos electrodomésticos está dado por la siguiente ecuación:

$$\begin{aligned} B(\mathbf{q}_a, \mathbf{p}) &= \sum_t a q_a[t] - \sum_t p[t] q_a[t] \\ &= \sum_t (a - p[t]) q_a[t] \end{aligned}$$

Entonces:

$$\frac{\partial B(\mathbf{q}_a, \mathbf{p})}{\partial q_a[t]} = a - p[t]$$

La pendiente de la derivada parcial depende de la relación entre a y $p[t]$. En las horas en las que $a \geq p[t]$ el algoritmo tiende a dar el máximo de energía posible. Mientras que las horas en donde $a \leq p[t]$ el algoritmo tiende a dar el mínimo de energía posible.

De esta manera, el valor de a puede ser interpretado como el umbral de precios por debajo del cual desea dar el máximo de energía.

Por ejemplo, en el caso del sistema de riego dada las características de la curva de precios utilizada se decidió utilizar un umbral de 3,5 \$/kWh. Es decir, se tomó $a = 0,0035$.

Aire acondicionado

La función utilizada para el aire acondicionado es la siguiente:

$$U(\mathbf{q}_a) = b - a \sum_{t \in \mathcal{T}_a} \left(T_{hab}[t] - T_{hab}^{conf}[t] \right)^2$$

Parámetro b :

El parámetro b es calculado diariamente para que se cumpla $U(\mathbf{q}_a = \vec{0}) = 0$. Es decir:

$$b = a \sum_{t \in \mathcal{T}_a} \left(T_{libre}[t] - T_{hab}^{conf}[t] \right)^2$$

Donde $T_{libre}[t]$ es la evolución de la temperatura de la habitación si $\mathbf{q}_a = \vec{0}$

Parámetro a :

Para ajustar el valor de a para el aire acondicionado se utilizaron tres curvas de precios:

- **precio_Promedio:** curva con la mediana de los precios de cada hora, ver figura C.1.
- **precio_Barato:** curva el percentil 15 % de los precios de cada hora.
- **precio_Caro:** curva que con el percentil 85 % de los precios de cada hora.

El valor de a del aire acondicionado fue calculado de forma iterativa hasta que el comportamiento del algoritmo sea satisfactorio bajo las tres curvas de precios. Buscando que se cumpliera:

- Bajo una curva de precios **precio_Barato** entonces el algoritmo debe satisfacer los deseos del usuario, es decir: $T_{hab}[t] \simeq T_{conf}[t]$.
- Bajo una curva de precios **precio_Caro** el algoritmo debe decidir sacrificar parte de la satisfacción del usuario con el fin de reducir los costos. Es decir la temperatura resultante debe tender a la restricción máxima/mínima.
- Bajo una curva de precios **precio_Promedio** la temperatura resultante debe quedar cerca de la temperatura de confort.

Este proceso iterativo se realizó para los dos hogares tipo. En la Tabla C.1 se muestran los valores obtenidos.

Calefón

La función utilizada para el calefón es la siguiente:

$$U(T_W[t], T_{conf}) = b - a \sum_{t \in \mathcal{T}_a} e^{-\frac{\gamma}{T_{conf}} T_W[t]}$$

Se observa que la utilidad de este electrodoméstico presenta un parámetro adicional γ el cual permite ajustar la tasa de crecimiento de la exponencial.

Apéndice C. Determinación de los parámetros de la función de utilidad

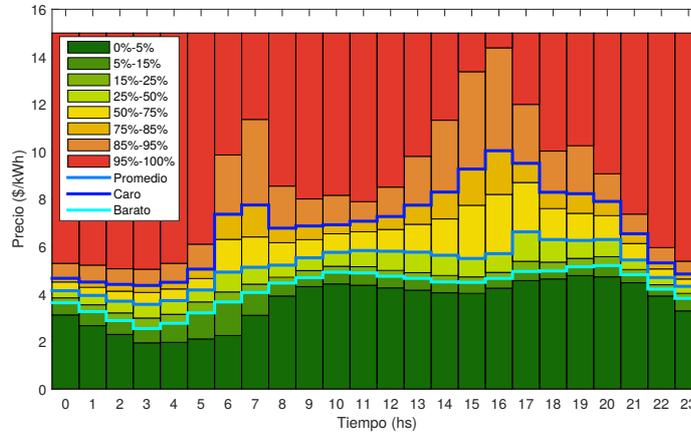


Figura C.1: Percentiles de la tarifa de precios utilizada

Parámetro b :

Al igual que en el caso del aire acondicionado, el valor de b es calculado diariamente de forma que se cumpla $U(\mathbf{q}_a = \vec{0}) = 0$.

Parámetro a y γ :

Para determinar los valores de a y γ para el calefón, se utilizó el mismo procedimiento que en el aire acondicionado.

En este caso se buscó que el algoritmo cumpliera:

- Bajo una curva de precios **precio_Promedio** la temperatura resultante al final del período de interés debe estar cerca de la temperatura de confort.
- Bajo una curva de precios **precio_Caro** el algoritmo debe decidir sacrificar parte de la satisfacción del usuario con el fin de reducir los costos. Es decir la temperatura resultante en el final de la temperatura de interés debe ser cercana a la restricción de temperatura mínima.
- Bajo una curva de precios **precio_Barato** la utilidad debe saturar y por lo tanto la temperatura no debe alejarse mucho de la temperatura de confort.

Ambos hogares tipo presentaban calefones de iguales características por lo que se utilizaron los mismos valores de parámetros en ambos hogares. Para facilitar el proceso de búsqueda se limitó que γ sólo tomara valores enteros.

Valores de parámetros utilizados en los hogares tipos

A continuación se detallan los valores de parámetros utilizados en los electrodomésticos de los hogares tipo:

Electrodoméstico	Hogar	Parámetros		
		a	b	γ
Lavarropas	1	0	0	-
Lavavajillas	1,2	0	0	-
Lavasecarropas	2	0	0	-
Riego	2	0.0035	0	-
Aire acond.	1	0.10	0	-
Aire acond.	2	0.13	0	-
Calefón	1	32	0	4
Calefón ppal. y suite	2	36	0	4

Tabla C.1: Valores de constantes utilizados en el Hogares Tipo 1 y 2

Apéndice D

Atributos y funciones de las clases electrodomésticos

La función de las clases de electrodomésticos es encapsular toda la información referente a los electrodomésticos e implementar los atributos y funciones que precisan los módulos del Nodo Controlador.

En este proyecto se crearon tres clases instanciables de electrodomésticos: Calefón, HVAC y CargaDiscreta. Todas ellas heredan de la clase abstracta Electrodoméstico, ver Figura D.1. El objetivo de esta clase abstracta es definir todos los atributos y funciones que necesariamente deben ser implementados en sus subclases. El objetivo de este apéndice es describir en detalle cómo se implementaron las clases de electrodomésticos.

Clase HVAC

La clase HVAC permite modelar y gestionar un aire acondicionado utilizado para el acondicionamiento térmico de una habitación. Esta clase implementa todos los atributos y funciones declarados por la clase abstracta Electrodoméstico.

La función de utilidad de este electrodoméstico depende de la temperatura de la habitación¹. Por lo que fue necesario implementar atributos y funciones para modelar la evolución de la temperatura a lo largo del día. Se agregan también, atributos que permiten al usuario configurar sus preferencias respecto a la temperatura de la habitación.

A continuación se detalla cómo fue implementada esta clase, describiendo los atributos y las funciones de la misma².

Atributos

- **idElectrodomestico***: *entero* que identifica al electrodoméstico.
- **nombreElectrodomestico***: *string* que identifica al electrodoméstico.
- **resolucionTS***: *entero* que especifica en cuántos *timeslot* está dividido el intervalo temporal de una hora.

¹ver Sección 4.3.

²Los atributos o función marcados con un asterisco son aquellos que necesariamente deben ser implementados porque son declarados en la clase abstracta electrodoméstico.

Apéndice D. Atributos y funciones de las clases electrodomésticos

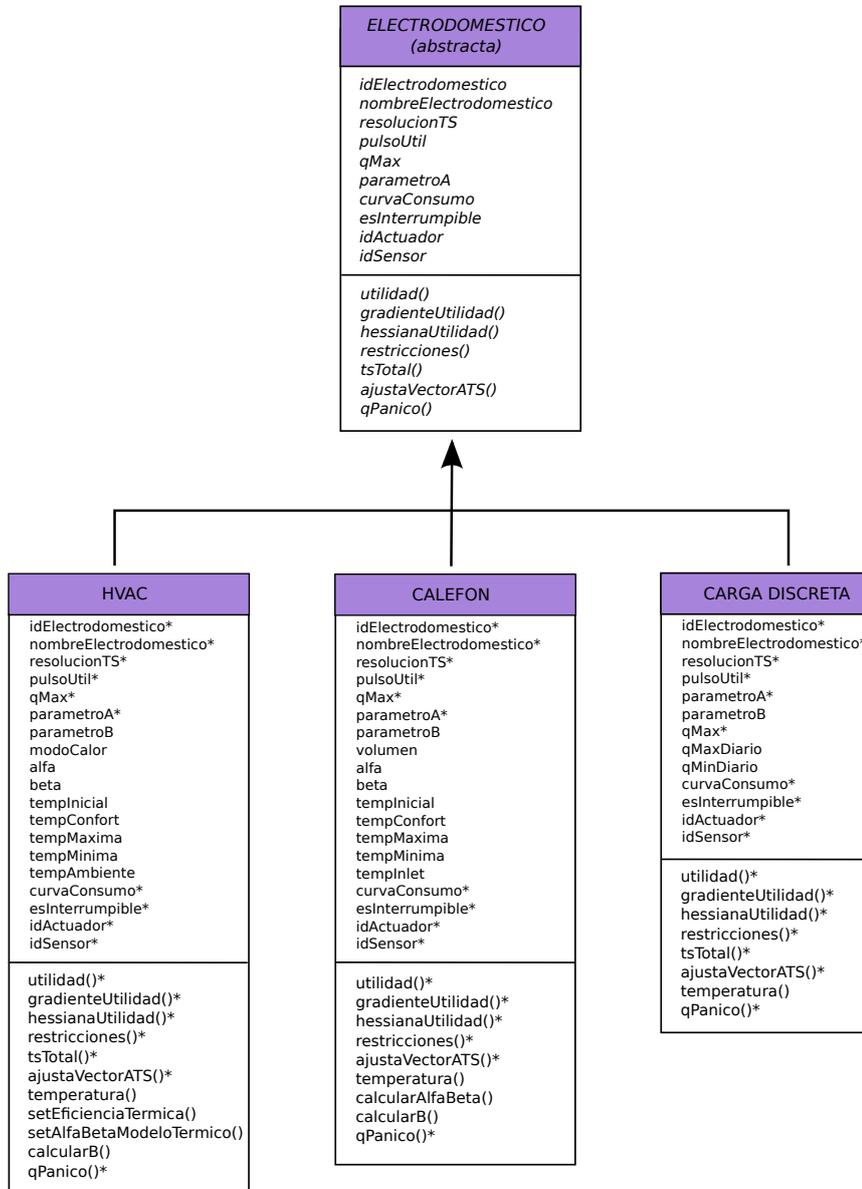


Figura D.1: Diagrama de clases de Nodo Controlador

- **pulsoUtil***: *vector binario* que especifica cuáles son los *timeslots* que se encuentran dentro del intervalo de interés (\mathcal{T}_a) del usuario.
 En el caso de un aire acondicionado se cumple que en los *timeslots* de interés ($\text{pulsoUtil}[i] = 1$):
 - La temperatura dentro de la habitación debe cumplir con la configuración de temperatura mínima, o máxima dependiendo del modo de funcionamiento del aire acondicionado.
 - El usuario recibe mayor satisfacción cuanto más cerca esta la temperatura en la

habitación $T[t]$ de la temperatura de confort $T_{confort}$.

Los *timeslot* que no son de interés para el usuario ($pulsoUtil[i] = 0$) no aportan a la utilidad del usuarios y se permite que la temperatura tome cualquier valor.

- **qMax***: *entero* que representa la energía máxima que el Optimizador puede suministrarle en una hora. Equivalente a la energía consumida por el HVAC operando a potencia nominal durante una hora de forma ininterrumpida. Esta energía depende del modo de operación del HVAC (`modoCalor`).
- **parametroA***: *entero* que representa el parámetro a de la función de utilidad.
- **parametroB**: *entero* que representa el parámetro b de la función de utilidad.
- **modoCalor**: *booleano* que indica el modo de operación del aire acondicionado. El modelo del HVAC prevé dos modo de operación:
 - Si `modoCalor=True`, el dispositivo calefacciona la habitación. Modo utilizado durante el invierno.
 - Si `modoCalor=False`, se refrigera la habitación. Modo utilizado durante el verano.
- **alfa**: *entero* que representa el parámetro α del modelo térmico del aire acondicionado.
- **beta**: *entero* que representa el parámetro β de la modelo térmico del aire acondicionado en $^{\circ}\text{C}/\text{Wh}$.
- **tempInicial**: *entero* que representa la temperatura en la habitación al inicio del día.
- **tempConfort**: *vector de enteros* que indica la temperatura deseada por el usuario en cada *timeslot*. Si bien se puede configurar valores en todos los *timeslot* del día, estos sólo son tenidos en consideración en los *timeslot* de interés.
- **tempMaxima**: *vector de enteros* que indica la temperatura máxima aceptada por el usuario en cada *timeslot*. Si bien se puede configurar valores en todos los *timeslot* del día, estos sólo son tenidos en consideración en los *timeslot* de interés. Esta restricción sólo es tenida en consideración si `modoCalor = False`
- **tempMinima**: *vector de enteros* que indica la temperatura mínima aceptada por el usuario en cada *timeslot*. Si bien se puede configurar valores en todos los *timeslot* del día, estos sólo son tenidos en consideración en los *timeslot* de interés. Esta restricción sólo es tenida en consideración si `modoCalor = True`
- **tempAmbiente**: *vector de decimales* que indica la temperatura ambiente del exterior en cada hora. Esta información es obtenida del sitio web de *WUnderground* por el módulo Configurator del Nodo Controlador[35].
- **curvaConsumo***: *vector de enteros* que representa la variable que describe el consumo eléctrico del electrodoméstico minuto a minuto en Wh. En el caso de aire acondicionado se asume que su curva constante e igual a: $qMax/60$.
- **esInterrumpible***: *booleano* que indica si el electrodoméstico es interrumpible o no. En el aire acondicionado se configuró que por defecto se tome `esInterrumpible =True`.
- **idActuador***: *entero* que identifica al Nodo de Carga que opera el electrodoméstico.
- **idSensor***: *entero* que identifica al sensor asociado. Es recomendable que este electrodoméstico presente un sensor de temperatura ambiente.

Funciones

- **utilidad(q_a)***: devuelve la utilidad del electrodoméstico evaluada q_a .
- **gradienteUtilidad(q_a)***: devuelve el gradiente de la función de utilidad evaluado en q_a .

Apéndice D. Atributos y funciones de las clases electrodomésticos

- **hessianaUtilidad(\mathbf{q}_a)***: devuelve la matriz Hessiana de la función de utilidad evaluada en \mathbf{q}_a .
- **restricciones()***: devuelve las restricciones G y h del electrodoméstico del día.
- **tsTotal()***: función auxiliar que devuelve el largo del vector de optimización \mathbf{q}_a .
- **ajustaVecHorarioATS(\mathbf{v})***: función auxiliar que permite ajustar un vector \mathbf{v} de largo 24 a uno de largo **tsTotal**.
- **temperatura(\mathbf{q}_a)**: devuelve vector con la evolución de la temperatura en la habitación si se le suministra un vector de energía \mathbf{q}_a .
- **setEficienciaTermica()**: función que ajusta la variable **qMax** y el modelo térmico del aire de acuerdo al valor de **modoCalor**. Esta función es invocada cada vez que se actualiza la variable **modoCalor**.
- **setAlfaBetaModeloTermico()**: función que calcula los valores de los atributos **alfa** y **beta** en función de los parámetros **condTermica** y **capTermica** que se pasan al constructor de la clase. Donde **condTermica** es el parámetro U_A del modelo térmico del aire acondicionado en $W/^\circ C$, y **capTermica** es parámetro C_{hab} de la modelo térmico del aire acondicionado en $Wh/^\circ C$.
- **calcularB()**: función utilizada para actualizar diariamente el valor del atributo **parametroB** de forma que la utilidad cumpla $U_a(\mathbf{q}_a = 0) = 0^3$.
- **qPanico()***: función que devuelve un vector \mathbf{q}_a que cumple con los requerimientos mínimos del usuario.
La solución de pánico implementada para el aire acondicionado es en cada *timeslot*, con **pulsoUtil**= 1, se entrega la energía necesaria para que se cumpla la restricción máxima o mínima de temperatura. En el caso que la energía necesaria supere la energía máxima que puede darse en un *timeslot* se entrega la energía máxima (**qMax/resolucionTS**).

Clase Calefón

La clase Calefón permite modelar el funcionamiento de un calefón eléctrico doméstico. De manera similar que el aire acondicionado, la utilidad de este electrodoméstico depende la temperatura del agua dentro del tanque. Por lo que en esta clase se encuentran implementados los atributos y funciones que permiten predecir la evolución de la temperatura a lo largo del día.

A continuación se detalla cómo fue implementada esta clase, describiendo los atributos y las funciones de la misma⁴.

Atributos

- **idElectrodomestico***: *entero* que identifica al electrodoméstico.
- **nombreElectrodomestico***: *string* que identifica al electrodoméstico.
- **resolucionTS***: *entero* que especifica en cuántos *timeslot* está dividido el intervalo temporal de una hora.
- **pulsoUtil***: *vector binario* que especifica cuáles son los *timeslots* que se encuentran dentro del intervalo de interés (\mathcal{T}_a) del usuario.

En el caso de Calefón se cumple que en los *timeslots* de interés:

³ver Apéndice C.

⁴Los atributos o función marcados con un asterisco son aquellos que necesariamente deben ser implementados porque son declarados en la clase abstracta electrodoméstico.

- La temperatura dentro de la habitación debe estar dentro del rango:

$$T_{\text{minima}} \leq T[t] \leq T_{\text{maxima}}$$

- El usuario recibe mayor satisfacción cuanto mayor es la temperatura del calefón.

Los *timeslot* que no son de interés para el usuario no aportan a la utilidad y se permite que la temperatura tome cualquier valor por debajo de la temperatura máxima del calefón.

- **qMax***: *entero* que representa la energía máxima que el Optimizador puede suministrarle al calefón en una hora.
- **parametroA***: *entero* que representa el parámetro a de la función de utilidad.
- **parametroB**: *entero* que representa el parámetro b de la función de utilidad .
- **volumen**: *entero* que representa el volumen del calefón en litros.
- **alfa**: *entero* que representa el parámetro α de la modelo térmico del calefón.
- **beta**: *entero* que representa el parámetro β de la modelo térmico del calefón en $^{\circ}\text{C}/\text{Wh}$.
- **tempInicial**: *entero* que representa la temperatura del agua dentro del tanque al inicio del día.
- **tempConfort**: *entero* que representa la temperatura del agua deseada por el usuario.
- **tempMaxima**: *entero* que indica la temperatura máxima del agua dentro del calefón, depende de las limitantes de seguridad del equipo.
- **tempMinima**: *entero* que indica la temperatura mínima aceptada por el usuario para poder bañarse.
- **tempInlet**: *entero* que indica la temperatura del agua fría entrante al tanque de agua.
- **curvaConsumo***: *vector de enteros* que representa la variable que describe el consumo eléctrico del calefón. En el caso de calefón por defecto se asume que su curva constante e igual a: $\text{qMax}/60$.
- **esInterrumpible***: *booleano* que indica si el electrodoméstico es interrumpible o no⁵.
- **idActuador***: *entero* que identifica al Nodo de Carga que opera el electrodoméstico.
- **idSensor***: *entero* que identifica al sensor asociado. Es recomendable que este electrodoméstico presente un sensor que mida la temperatura del agua.

Funciones

- **utilidad(\mathbf{q}_a)***: devuelve la utilidad del electrodoméstico evaluada en \mathbf{q}_a .
- **gradienteUtilidad(\mathbf{q}_a)***: devuelve el gradiente de la función de utilidad evaluado en \mathbf{q}_a .
- **hessianaUtilidad(\mathbf{q}_a)***: devuelve la matriz Hessiana de la función de utilidad evaluada en \mathbf{q}_a .
- **restricciones()***: devuelve las restricciones G y h del electrodoméstico del día.
- **tsTotal()***: función auxiliar que devuelve el largo del vector de optimización \mathbf{q}_a .
- **ajustaVecHorarioATS(\mathbf{v})***: función auxiliar que permite ajustar un vector \mathbf{v} de largo 24 a uno de largo **tsTotal**.
- **temperatura(\mathbf{q}_a)**: devuelve vector con la evolución de la temperatura del agua dentro del tanque si se le suministra un vector de energía \mathbf{q}_a .

⁵Esta información es utilizada por el módulo Intérprete, ver Sección 5.6.

Apéndice D. Atributos y funciones de las clases electrodomésticos

- **calcularAlfaBeta()**: función que calcula los parámetros α y β del modelo térmico de acuerdo a las dimensiones del calefón, es decir en función de **volumen** y **altura** y sus características térmica, es decir de los parámetros **condTermica** y **capTermica** que se pasan al constructor de la clase. Donde **condTermica** es el parámetro U_A del modelo térmico del calefón en $W/(m^2 \text{ } ^\circ\text{C})$, y **capTermica** es parámetro C_{WH} en $Wh/(l \text{ } ^\circ\text{C})$.
- **calcularB()**: función utilizada para actualizar diariamente el valor del atributo **parametroB** de forma que la utilidad cumpla $U_a(\mathbf{q}_a = 0) = 0^6$.
- **qPanico()***: función que devuelve un vector \mathbf{q}_a que cumple con los requerimientos mínimos del usuario.
La solución de pánico implementada para el calefón es dar la energía suficiente para que se cumpla la restricción de temperatura mínima, configurada por el usuario, en los *timeslots* i -ésimos dónde $pulsoUtil[i] = 1$.

Clase CargaDiscreta

La clase CargaDiscreta permite modelar el funcionamiento de las cargas de tipo discretas, por ejemplo: lavarropas, lavavajillas, sistema de riego.

A continuación se describen los atributos y funciones implementados en esta clase.⁷

Atributos

- **idElectrodomestico***: *entero* que identifica al electrodoméstico.
- **nombreElectrodomestico***: *string* que identifica al electrodoméstico.
- **resolucionTS***: *entero* que especifica en cuántos *timeslot* está dividido el intervalo temporal de una hora.
- **pulsoUtil***: *vector binario* que especifica cuáles son los *timeslots* que se encuentran dentro del intervalo de interés (\mathcal{T}_a) del usuario.
En el caso de las cargas discretas se denomina *timeslots* de interés, a aquellos en los cuales está permitido encender el electrodoméstico. En los *timeslot* que no son de interés ($pulsoUtil[i] = 0$) el consumo de las cargas discretas debe ser nulo.
- **parametroA***: *entero* que representa el parámetro a de la función de utilidad.
- **parametroB**: *entero* que representa el parámetro b de la función de utilidad.
- **qMax***: *entero* que representa la energía máxima que el Optimizador puede suministrarle al electrodoméstico en una hora. En el caso de las cargas discretas equivale a la energía que consume la carga si se le entrega la una potencia P_{prom} durante una hora. Donde P_{prom} es la potencia promedio que consume la carga durante un ciclo completo de operación.
- **qMaxDiario**: *entero* que representa la energía máxima que se le puede suministrar al electrodoméstico en el día.
- **qMinDiario**: *entero* que representa la energía mínima que se le puede suministrar al electrodoméstico en el día.
- **curvaConsumo***: *vector de enteros* que representa la variable que describe el consumo eléctrico del electrodoméstico minuto a minuto en Wh.
- **esInterrumpible***: *booleano* que indica si el electrodoméstico es interrumpible o no.

⁶ver apéndice C.

⁷Los atributos o función marcados con un asterisco son aquellos que necesariamente deben ser implementados porque son declarados en la clase abstracta electrodoméstico.

- `idActuador*`: *entero* que identifica al Nodo de Carga que opera el electrodoméstico.
- `idSensor*`: *entero* que identifica al sensor asociado.

Funciones

- `utilidad(\mathbf{q}_a)*`: devuelve la utilidad del electrodoméstico evaluada en \mathbf{q}_a .
- `gradienteUtilidad(\mathbf{q}_a)*`: devuelve el gradiente de la función de utilidad evaluado en \mathbf{q}_a .
- `hessianaUtilidad(\mathbf{q}_a)*`: devuelve la matriz Hessiana de la función de utilidad evaluada en \mathbf{q}_a .
- `restricciones()*`: devuelve las restricciones G y h del electrodoméstico del día.
- `tsTotal()*`: función auxiliar que devuelve el largo del vector de optimización \mathbf{q}_a .
- `ajustaVecHorarioATS(\mathbf{v})*`: función auxiliar que permite ajustar un vector \mathbf{v} de largo 24 a uno de largo `tsTotal`.
- `qPanico()*`: función que devuelve un vector \mathbf{q}_a que cumple con los requerimientos mínimos del usuario.

La solución de pánico implementada para las cargas discretas consistió en entregar la energía mínima `qMinDiario` al inicio del `pulsoUtil`.

Apéndice E

Caracterización del problema de optimización y librería CVXOPT

En el Capítulo 4 se describió el funcionamiento del algoritmo de gestión implementado en el Nodo Controlador. El mismo debe resolver un problema de optimización convexa. El objetivo de este apéndice es describir cuales son las principales características de estos problemas. Se describe también la herramienta utilizada para resolverlo.

E.1. Optimización convexa

El estudio detallado de la teoría de optimización convexa está fuera del alcance de este proyecto, por lo tanto en este documento no se pretende cubrir este tema. A modo de introducción, a continuación se definen los conceptos más importantes y se enumeran las ventajas de tratar con este tipo de problemas. En caso de querer profundizar sobre este tema se puede consultar el libro[36].

Se entiende como problema de optimización a uno de la forma:

$$\begin{aligned} \text{mín} & \quad f_0(x) \\ \text{restringido a:} & \quad f_i(x) \leq b_i \quad i = 1, \dots, m \end{aligned} \tag{E.1}$$

Donde $\mathbf{x} = [x_1, \dots, x_n]$ es la variable de optimización, $f_0 : \mathcal{R}^n \rightarrow \mathcal{R}$ es la función objetivo. El espacio de soluciones posibles \mathcal{D} , también llamado región factible, está determinado por las funciones restrictivas $f_i : \mathcal{R}^n \rightarrow \mathcal{R}$ y los límites de las restricciones b_i .

Se desea hallar la solución \mathbf{x}^* que cumple que:

$$\mathbf{x}^* \in \mathcal{D} \quad \text{y} \quad f_0(\mathbf{x}^*) \leq f_0(\mathbf{z}) \quad \forall \mathbf{z} \in \mathcal{D}.$$

Conjunto convexo:

Se dice que un conjunto \mathcal{D} es convexo, si la línea que une dos puntos en \mathcal{D} pertenece a \mathcal{D} . Es decir:

$$\mathcal{D} \text{ es convexo} \iff \theta x_1 + (1 - \theta)x_2 \in \mathcal{D} \quad \forall x_1, x_2 \in \mathcal{D} \\ \forall \theta \in [0, 1]$$

Apéndice E. Caracterización del problema de optimización y librería CVXOPT

Función convexa:

Se dice que una función $f : \mathcal{R}^n \rightarrow \mathcal{R}$ es convexa, si en un dominio \mathcal{D}_f convexo, se cumple:

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad \forall x, y \in \mathcal{D}_f \\ \forall \theta \in [0, 1]$$

Gráficamente esto significa que el segmento de línea que une a $f(x)$ y $f(y)$ está siempre por encima del valor que toma f para los valores intermedios entre x e y .

Optimización convexa:

Se dice que un problema es de optimización convexa si se cumple que la función objetivo $f_0(\mathbf{x})$ es una función convexa y la región factible \mathcal{D} es un conjunto convexo.

Los problemas de optimización convexos presentan las siguientes propiedades:

- Si existe un mínimo local de la función $f_0(\mathbf{x})$ este es un mínimo absoluto.
- Si $f_0(\mathbf{x})$ es estrictamente convexa y esta presenta un mínimo, este mínimo es único.

No existe una fórmula analítica genérica para la resolución de problemas convexos, sino que es necesario utilizar métodos iterativos. La propiedad que presentan los problemas de optimización convexa, que establece que todo mínimo local de la función $f_0(\mathbf{x})$ es necesariamente el mínimo global, permite asegurar la convergencia de los métodos iterativos. Evitándose así el inconveniente de la convergencia a un mínimo local.

Otra ventaja que presentan los problemas convexos es que son una rama largamente estudiada. Si bien esta es una rama aún activa y no hay consenso sobre cuáles son los mejores algoritmos de resolución, ya existen métodos que son muy efectivos[36]. Adicionalmente, estos algoritmos se encuentran implementados en diversos lenguajes de programación y resulta sencillo acceder a bibliotecas que los incluyan. Por ejemplo, *CVX* en *MatLab* o *CVXOPT* en *Python*.

Caracterización del problema a resolver

El algoritmo de gestión es el encargado de determinar cada día el vector \mathbf{q}_a para cada uno de los electrodomésticos. Es decir, debe decidir cuánta energía se le ha de suministrar en cada *timeslot* a cada dispositivo, de manera de minimizar el costo en energía eléctrica y no descuidar el confort del usuario.

El algoritmo utilizado en este proyecto, plantea el problema de maximización de la Ecuación E.2. En el mismo se busca minimizar el costo en energía y a la vez maximizar la utilidad de los electrodoméstico. Esta función de utilidad cuantifica la satisfacción que recibe el usuario de acuerdo al patrón de energía que se le suministra al electrodoméstico¹.

$$\max_{\mathbf{q}_a} \sum_{a \in \mathcal{A}} \left(U_a(\mathbf{q}_a) - \sum_{t=0}^{N-1} p[t] q_a[t] \right) \quad (\text{E.2})$$

Restricto a:

$$M^a \mathbf{q}_a \leq \eta_a \quad \forall a \in \mathcal{A} \quad (\text{Restricciones aparatos}) \\ 0 \leq \sum_{a \in \mathcal{A}} q_a[t] \leq Q^{max}[t] \quad \forall t \in [0, N - 1] \quad (\text{Energía máxima del hogar})$$

¹Por más detalle acerca del algoritmo consultar Sección 4.1.

Tomando el opuesto de la función de optimización $f_0(\cdot)$, es posible reescribirlo como un problema de minimización de la forma de la Ecuación E.1.

$$\min_{\mathbf{q}_a} \sum_{a \in \mathcal{A}_i} \left(-U_a(\mathbf{q}_a) + \sum_{t=0}^{N-1} p[t] q_a[t] \right)$$

Restricto a:

$$\begin{aligned} M^a \mathbf{q}_a &\leq \eta_a & \forall a \in \mathcal{A} & \quad (\text{Restricciones aparatos}) \\ 0 \leq \sum_{a \in \mathcal{A}} q_a[t] &\leq Q^{max} & \forall t \in [0, N-1] & \quad (\text{Energía máxima del hogar}) \end{aligned}$$

En primer lugar, se observa que las restricciones son una combinación lineal del vector \mathbf{q}_a y por lo tanto determinan un conjunto convexo. Por otra parte, se observa que la función de costo de la energía ($p[t] q_a[t]$) es lineal y convexa. Por consiguiente, si se tiene el cuidado de elegir funciones cóncavas para modelar la utilidad de los electrodomésticos, la función objetivo ($f_0(\mathbf{x})$) resultante es convexa². Por lo tanto el problema resultante es uno de optimización convexa.

Al limitar que las funciones de utilidad de los electrodomésticos fueran funciones convexas, se limitaron las posibles funciones a utilizar. Igualmente, se decidió restringirse a este tipo de funciones con el propósito de que el problema fuese convexo y así poder explotar las ventajas que presentan los mismos.

E.2. Paquete CVXOPT

Para la implementación de módulo Optimizador se utilizó el paquete CVXOPT de *Python* (ver Capítulo 5). Este es un paquete de software libre para la resolución de problemas de optimización convexa, desarrollado por M. Andersen, J. Dahl, y L. Vandenberghe[37].

Este paquete contiene un biblioteca de funciones que permite resolver de forma sencilla problemas de optimización convexa tanto lineales como no lineales. De acuerdo a la documentación del mismo utiliza métodos de punto interior para la resolución de dichos problemas.

Función: `sol = cvxopt.solvers.cp(F [G h dims A b kkt solver])`

Dentro de las biblioteca de funciones, la función `cvxopt.solvers.cp` permite resolver problemas convexas no lineales que puedan ser expresados de la siguiente manera:

$$\begin{aligned} \max_x \quad & f_0(x) \\ \text{restricto a:} \quad & f_k(x) \leq 0 \quad k = 1, \dots, m \\ & Gx \leq h \\ & Ax = b \end{aligned}$$

El parámetro **F** es una función que debe presentar una estructura específica. A través de esta se especifica las funciones $f_k(\cdot)$, el gradiente $\nabla f_k(\cdot)$ y la matriz Hessiana $H(f_k(\cdot))$. El

²En este razonamiento se utilizó que el opuesto de una función cóncava es convexa. Y que la combinación lineal, con coeficientes positivos, de funciones convexas es convexa

Apéndice E. Caracterización del problema de optimización y librería CVXOPT

detalle de cómo fueron hallados el gradiente y la matriz Hessiana de $f_0(\cdot)$ se encuentra más adelante en este apéndice.

Los parámetros \mathbf{G} , \mathbf{h} , \mathbf{dims} , \mathbf{A} , \mathbf{b} y `kktsolver` son opcionales. En este trabajo, sólo se utilizaron los parámetros \mathbf{G} y \mathbf{h} para especificar las restricciones del problema. Los parámetros \mathbf{A} y \mathbf{b} no se utilizaron dado que no se hizo uso de restricciones de este tipo.

Con el parámetro `kktsolver` se puede dar información adicional de manera de explotar la estructura del problema y así obtener un mejor desempeño. Mediante pruebas se determinó que el desempeño de la función `cvxopt.solvers.cp()` sin especificar `kktsolver` era correcto para un algoritmo tipo ‘day-ahead’, en el cual el problema se resuelve una vez al día. Por ello, se decidió no utilizar esta funcionalidad. En aplicaciones en las que el problema deba resolverse en forma horaria, o con una periodicidad menor, sería conveniente evaluar la conveniencia de explotar la estructura del problema y el uso del parámetro `kktsolver`.

Esta función devuelve un diccionario `sol` que contiene la solución e información acerca de las características de la solución. Sus elementos más importantes son:

- `x`: solución del problema obtenida.
- `status`: estado de la solución:
 - `optimal`: el algoritmo de resolución convergió y la solución \mathbf{x} es óptima.
 - `unknown`: el algoritmo de resolución terminó antes de encontrar una solución óptima, debido a problemas numéricos o porque se alcanzó el máximo de iteraciones. En este caso \mathbf{x} es el punto en donde el algoritmo se detuvo.

E.3. Detalle del cálculo del gradiente y Hessiana

Sea \mathbf{q} el vector resultante de la concatenación de los vectores \mathbf{q}_a de cada electrodoméstico.

$$\mathbf{q} = [q_{a_1}[0], \dots, q_{a_1}[N-1], q_{a_2}[0], \dots, q_{a_2}[N-1], \dots, q_{a_A}[0], \dots, q_{a_A}[N-1]]$$

La función objetivo a maximizar puede re-escribirse de la siguiente forma:

$$\begin{aligned} f_0(\mathbf{q}) &= \sum_{a \in \mathcal{A}_i} \left(U_a(\mathbf{q}_a) - \sum_t p[t] q_a[t] \right) \\ &= \sum_{a \in \mathcal{A}_i} f_{0a}(\mathbf{q}_a) \end{aligned} \quad (\text{E.3})$$

Gradiente

En primer lugar se desea hallar el gradiente de $f_0(\mathbf{q})$.

$$\nabla U(\mathbf{q}_a) = \left[\frac{\partial U}{\partial q_0}, \frac{\partial U}{\partial q_1}, \dots, \frac{\partial U}{\partial q_{N-1}} \right]$$

Como la función de utilidad de un electrodoméstico no depende de la energía que se le suministra al resto de los electrodomésticos, el gradiente $\nabla f_0(\mathbf{q})$ es igual a la concatenación de los gradientes $\nabla f_{0a}(\mathbf{q}_a)$.

E.3. Detalle del cálculo del gradiente y Hessiana

$$\begin{aligned}
 \nabla f_o(\mathbf{q}) &= \nabla \left(\sum_{a \in \mathcal{A}_i} \left(U_a(\mathbf{q}_a) - \sum_t p[t] q_a[t] \right) \right) \\
 &= \left[\nabla f_{0a_1}(\mathbf{q}_{a_1}), \nabla f_{0a_2}(\mathbf{q}_{a_2}), \dots, \nabla f_{0a_A}(\mathbf{q}_{a_A}) \right] \\
 &= \left[[\nabla U_{a_1}(\mathbf{q}_{a_1}) - \mathbf{p}], [\nabla U_{a_2}(\mathbf{q}_{a_2}) - \mathbf{p}], \dots, [\nabla U_{a_A}(\mathbf{q}_{a_A}) - \mathbf{p}] \right]
 \end{aligned}$$

Matriz Hessiana

$$H(f_0(\mathbf{q})) = \begin{bmatrix} \frac{\partial^2 U}{\partial q_1^2} & \frac{\partial^2 U}{\partial q_1 \partial q_2} & \dots & \frac{\partial^2 U}{\partial q_1 \partial q_{N-1}} \\ \frac{\partial^2 U}{\partial q_2 \partial q_1} & \frac{\partial^2 U}{\partial q_2^2} & \dots & \frac{\partial^2 U}{\partial q_2 \partial q_{N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 U}{\partial q_{N-1} \partial q_1} & \frac{\partial^2 U}{\partial q_{N-1} \partial q_2} & \dots & \frac{\partial^2 U}{\partial q_{N-1}^2} \end{bmatrix} \quad (\text{E.4})$$

Con un razonamiento similar al utilizado para el gradiente, se puede mostrar que la matriz Hessiana de $f_0(\mathbf{q})$ es de la forma:

$$H(f_0(\mathbf{q})) = \begin{bmatrix} \left[H(f_{0a_1}(\mathbf{q}_{a_1})) \right] & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \left[H(f_{0a_2}(\mathbf{q}_{a_2})) \right] & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \left[H(f_{0a_A}(\mathbf{q}_{a_A})) \right] \end{bmatrix}$$

Por consiguiente, para determinar el gradiente y la matriz Hessiana de la función objetivo, basta con hallar el gradiente y la Hessiana de cada una de las funciones de utilidad.

A continuación se detalla el cálculo del gradiente y matriz Hessiana para cada una de las funciones de utilidad seleccionadas³.

Aire acondicionado

Función de utilidad:

³La función de utilidad seleccionada para cada electrodoméstico se describe en el Capítulo 4.

Apéndice E. Caracterización del problema de optimización y librería CVXOPT

$$U(\mathbf{q}_a) = b - a \sum_{t=0}^{N-1} \pi[j] (T[t] - T_{conf})^2 \quad (\text{E.5})$$

donde:

$$T[t] = T_H[t] + \sum_{k=0}^t (1 - \alpha)^{t-k} \beta \eta q_a[k]$$

$$T_H[t] = (1 - \alpha)^{t+1} T[-1] + \sum_{k=0}^j (1 - \alpha)^{t-k} \alpha T_{amb}[k]$$

$$\pi[j] = \begin{cases} 1 & \text{si } j \in \mathcal{T}_a \\ 0 & \text{si } j \notin \mathcal{T}_a \end{cases}$$

Gradiente:

La derivada parcial de $U(\mathbf{q}_a)$ respecto a una de las entradas del vector \mathbf{q}_a es:

$$\frac{\partial U}{\partial q_k} = -2a \sum_{j=0}^{N-1} \pi[j] (T[j] - T_{conf}) \frac{\partial T[j]}{\partial q_k}$$

Derivando el segundo término de la Ecuación E.5 se obtiene:

$$\frac{\partial T[j]}{\partial q_k} = \begin{cases} 0 & \text{si } k > j \\ (1 - \alpha)^{j-k} \beta \eta & \text{si } k \leq j \end{cases}$$

Se observa que esta derivada es nula si $k > j$. Lo cual implica que la temperatura del aire de la habitación en el instante j no depende de cuánta energía se le entrega al aire acondicionado en los instantes futuros. Esto era previsible dado que el sistema es causal.

Combinando los resultados anteriores se obtiene:

$$\frac{\partial U}{\partial q_k} = -2a\beta\eta \sum_{j=k}^{N-1} \pi[j] (T[j] - T_{conf}) (1 - \alpha)^{j-k}$$

Matriz Hessiana:

Las derivadas segundas de $U(\mathbf{q}_a)$ respecto a dos entradas cualesquiera del vector \mathbf{q}_a son de la forma:

$$\frac{\partial^2 U}{\partial q_k \partial q_s} = -2a\beta\eta \sum_{j=k}^{N-1} \pi[j] (1 - \alpha)^{j-k} \frac{\partial T[j]}{\partial q_s}$$

Utilizando la derivada de la temperatura respecto a q_i hallada, la derivada segunda de la función de utilidad resulta:

$$\frac{\partial^2 U}{\partial q_k \partial q_s} = -2a\beta^2\eta^2 \sum_{j=\max\{k,s\}}^{N-1} \pi[j] (1 - \alpha)^{j-k} (1 - \alpha)^{j-s}$$

E.3. Detalle del cálculo del gradiente y Hessiana

Calefón

Función de utilidad:

$$U(\mathbf{q}_a) = b - a \sum_{t=0}^{N-1} \pi(t) e^{\frac{-\gamma T[t]}{T_{conf}}}$$

donde:

$$T[t] = T_H[t] + \sum_{k=0}^t (1-\alpha)^{t-k} \prod_{j=k}^t (1-\delta_V[j]) \beta \eta q[k]$$

$$T_H(t) = (1-\alpha)^{t+1} \prod_{k=0}^t (1-\delta_V[k]) T[-1]$$

$$+ \sum_{k=0}^t (1-\alpha)^{t-k} \prod_{j=k}^t (1-\delta_V[j]) \alpha T_{ext}$$

$$+ \sum_{k=0}^t (1-\alpha)^{t-k} \prod_{j=k+1}^t (1-\delta_V[j]) \delta_V[k] T_{inlet}$$

Gradiente:

La derivada de la temperatura respecto a $q[k]$ es:

$$\frac{\partial T[j]}{\partial q_k} = \begin{cases} 0 & \text{si } k > j \\ \beta \eta (1-\alpha)^{j-k} \prod_{i=k}^j (1-\delta_V[i]) & \text{si } k \leq j \end{cases}$$

Por lo que las entradas del gradiente resultan:

$$\frac{\partial U}{\partial q_k} = \frac{\gamma a \beta \eta}{T_{conf}} \sum_{j=k}^{N-1} \pi[j] e^{\frac{-\gamma T[j]}{T_{conf}}} (1-\alpha)^{j-k} \prod_{i=k}^j (1-\delta_V[i])$$

Matriz Hessiana:

Razonando análogamente las entradas de la matriz Hessiana resultan:

$$\frac{\partial^2 U}{\partial q_k \partial q_s} = \frac{-\gamma^2 a \beta^2 \eta^2}{T_{conf}^2} \sum_{j=\max\{k,s\}}^{N-1} \pi[j] e^{\frac{-\gamma T[j]}{T_{conf}}} (1-\alpha)^{j-k} (1-\alpha)^{j-s} \prod_{i=k}^j (1-\delta_V[i]) \prod_{i=s}^j (1-\delta_V[i])$$

Cargas de consumo discreto

Función de utilidad:

$$U(\mathbf{q}_a) = b + a \sum_{t=0}^{N-1} \pi[t] q[t]$$

Gradiente:

La derivada de la utilidad respecto a $q[k]$ es:

$$\frac{\partial U}{\partial q_k} = a \pi[k]$$

Función de utilidad:

La matriz Hessiana es nula, por tanto las derivadas segundas son cero.

$$\frac{\partial^2 U}{\partial q_k \partial q_s} = 0$$

Apéndice F

Concurrencia en Python

La concurrencia es una propiedad de los sistemas que permite ejecutar varias tareas en forma 'simultánea' y que estas interactúen entre si. Este es un tema complejo y existen varios enfoques para lograr el objetivo antes mencionado. No se pretende en esta documentación cubrir este tema, sino que se da una breve introducción al concepto y a las librerías de *Python* utilizadas.

La computación concurrente es opuesta a la computación secuencial. En la computación secuencial las tareas son ejecutadas secuencialmente, es decir una tarea se inicia luego de que finaliza la anterior. En cambio, en la computación concurrente en un instante dado pueden existir varias tareas en curso. No obstante, que un sistema sea concurrente no implica que las tareas se ejecuten en forma realmente simultánea sino que las tareas son interrumpidas durante su ejecución para que el procesador ejecute otras tareas, ver Figura F.1.

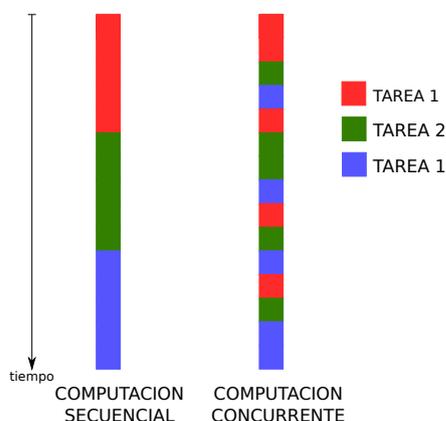


Figura F.1: Computación concurrente vs. secuencial

La computación concurrente permite optimizar el uso de los recursos del sistema y es especialmente útil cuando las tareas interactúan con dispositivos de entrada y salida. En este trabajo, los módulos Actuator y RegistroMediciones del Nodo Controlador deben comunicarse con los Nodos de Carga. Ello implica enviar un mensaje al nodo y esperar por su respuesta. En un programa secuencial el proceso se bloquearía en espera de la respuesta. Sin embargo, la comunicación con los Nodos de Carga se realiza muy frecuentemente, enviándose varios mensajes por minuto, por lo que el programa presentaría una gran cantidad de tiempo de

Apéndice F. Concurrencia en Python

espera. En un programa concurrente, en cambio, este tiempo de espera puede ser utilizado por otras tareas lo que permite hacer un uso más eficiente de los recursos.

F.1. *Scheduling*

Un aspecto importante de la computación concurrente refiere a cuándo se interrumpe una tarea y cómo se selecciona la siguiente tarea a ejecutar, lo cual se conoce como *scheduling*. Existen múltiples estrategias de 'scheduling' que pueden clasificarse en dos grandes tipos:

- *Scheduling cooperativo*: las tareas detienen voluntariamente su ejecución y ceden los recursos a las tareas en espera.
- *Scheduling apropiativo*: el sistema operativo es el encargado de interrumpir las tareas y administrar los recursos.

En este proyecto se utilizaron dos librerías de *Python* distintas de programación concurrente. En primer lugar, para la implementación de los módulos Actuador y RegistroMediciones se usó la librería `asyncio`, que permite desarrollar códigos con *scheduling cooperativo*. En segundo lugar, se utilizó la librería `threading` para ejecutar y coordinar las funciones de los diferentes módulos del Nodo Controlador.

Librería `threading`

Un thread, o hilo de ejecución, es la unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo. Cada procesador puede en cada instante ejecutar un único thread. En una máquina multi-procesador cada procesador puede ejecutar un thread, lo que permite la ejecución de tareas en simultáneo. Por otra parte, en las máquinas mono-procesador es posible alternar el procesador entre los distintos threads, aparentando que la ejecución es simultánea.

En Python la ejecución de los threads es controlada por el GIL (Global Interpreter Lock), el cual permite que sólo un thread pueda ejecutarse a la vez, independientemente del número de procesadores. Cada cierto número de instrucciones la máquina virtual detiene la ejecución del thread y selecciona otro de entre los que encontraban en espera. Por defecto el cambio de thread se realiza cada 10 instrucciones de bytecode, aunque esto es configurable[59]. La librería `thread` implementa las herramientas necesarias para escribir códigos con múltiples hilos de ejecución. Esta es una librería de bajo nivel, por lo que se eligió utilizar la librería `threading` que proporciona una interfaz de más alto nivel para trabajar con `thread`[60].

Estas librerías presentan un *scheduling* de tipo apropiativo, por consiguiente es responsabilidad del sistema operativo efectuar la coordinación entre los threads. Si bien existen mecanismos para bloquear un thread y hacer que un conjunto de instrucciones se ejecuten sin interrupción, en general no es posible saber precisamente en qué momento se alterna de un thread a otro.

■ **Crear un nuevo thread:**

Existen dos maneras de crear un nuevo thread. La primera es crear una subclase que extienda de `Thread` y sobrescribir su método `run()`. La segunda, implica crear directamente una instancia de la clase `Thread`. En este último caso se debe pasar como parámetro la función que debe ejecutar el hilo y sus parámetros, de la siguiente manera: `nuevo_thread=threading.Thread(target=funcion_a_ejecutar args=(arg1 arg2))`

■ **Iniciar la actividad de un thread:**

Una vez creado el thread este no inicia su actividad automáticamente sino que es necesario ejecutar el comando: `nuevo_thread.start()`.

Librería `asyncio`

La librería `asyncio` de Python 3.4 provee la infraestructura necesaria para escribir códigos concurrentes dentro de un mismo hilo de ejecución[61]. Esta librería implementa un tipo de *scheduling cooperativo* utilizando corrutinas. Una corrutina es una subrutina que presenta puntos de salida en los cuales la ejecución se detiene en espera de un resultado. Cuando el resultado está listo, la ejecución se reanuda, retornando al punto en que había quedado suspendida.

Se destacan los siguientes elementos de la librería `asyncio`:

- **Corrutinas.** Son generadores de Python que pueden suspender su ejecución en espera de un resultado futuro. Una vez que el resultado está listo se retorna al mismo punto dónde se había detenido. Una corrutina puede utilizar los siguientes comandos:
 - `resultado = yield from future(o task)`: la corrutina suspende su ejecución hasta que se complete el `future(o task)` y retorna su resultado.
 - `resultado = yield from coroutine`: la corrutina suspende su ejecución hasta que la corrutina produzca un resultado y lo devuelva.
 - `return expression`: produce un resultado y lo pasa a la corrutina que está esperando por ella (i.e. la corrutina que ejecutó el comando `resultado = yield from coroutine`).
 - `raise exception`: levanta un excepción y la pasa a la corrutina que está esperando por ella.

Para definir una corrutina basta utilizar el decorador `@asyncio.coroutine` en la función que debe ejecutar la corrutina.

- **Event Loop.** Es el encargado del *scheduling* de las corrutinas. Lleva registro de qué corrutina se está ejecutando y cuáles se encuentran en espera.
- **Future.** Es una abstracción del resultado de una operación que todavía no ha concluido. Cuando una corrutina suspende su ejecución lo hace en espera de un objeto de la clase `asyncio.Future`.
- **Task.** Un `Task` es una corrutina encapsulada en un objeto de la clase `Future`, lo que permite agendar la corrutina en el event loop. La clase `asyncio.Task` es una subclase de `asyncio.Future`.

Comunicación entre tareas

En los sistemas concurrentes se desconoce el orden exacto en el que se ejecutan las tareas. En consecuencia, uno de los mayores desafíos en este tipo de sistemas es la comunicación entre tareas y la coordinación del acceso a recursos compartidos.

El método más sencillo para la comunicación entre tareas es el uso de variables comunes. Sin embargo, esta forma de comunicación puede llevar a errores cuando dos tareas quieren escribir simultáneamente o cuando una tarea lee mientras otra escribe en la variable. Estos errores pueden evitarse mediante la implementación de secciones críticas, el uso de locks, semáforos, condiciones y eventos[59].

Otro método común es el uso de colas para compartir información entre tareas ‘productoras’ y ‘consumidoras’. En particular, las librerías `threading` y `asyncio` presentan las clases `Queue` y `asyncio.Queue` respectivamente, que implementan colas que permiten compartir información

Apéndice F. Concurrencia en Python

entre tareas de forma segura. Las ventaja de estas colas es que su uso es sencillo y el usuario no tiene que recurrir al uso locks o semáforos lo cual puede llevar a errores.

Clase `Queue`:

La clase `Queue` implementa las herramientas necesarias para la creación y uso de colas con múltiples productores y consumidores. Está especialmente diseñada para intercambiar información de forma segura entre distintos hilos de ejecución[62].

Se destacan los siguientes métodos de la clase:

- `empty()`: devuelve 'True' si la cola esta vacía o 'False' en caso contrario.
- `full()`: devuelve 'True' si la cola está llena o 'False' en caso contrario. Cuando se crea la cola es posible especificar el tamaño máximo de la cola. Por defecto el tamaño de la cola es infinito y esta nunca se llena.
- `get([block])`: función que saca un elemento de la cola y lo devuelve como resultado. En el caso que la cola esté vacía y el parámetro opcional `block` sea 'True' el thread se bloquea hasta que haya un elemento disponible.
- `put(item, [block])` función que agrega el elemento `item` en la cola. En el caso que la cola esté llena y `block` sea 'True' el thread se bloquea hasta que haya espacio disponible en la cola.

Clase `asyncio.queue`:

Esta clase implementa un cola que permite coordinar corrutinas 'productoras' y 'consumidoras' de información[61]. Los métodos más importantes de esta clase son los siguientes:

- `empty()`: devuelve 'True' si la cola esta vacía o 'False' en caso contrario.
- `full()`: devuelve 'True' si la cola está llena o 'False' en caso contrario.
- Corrutina `get()`: corrutina que saca un elemento de la cola y lo devuelve como resultado. En el caso que la cola esté vacía la corrutina se bloquea hasta que haya un elemento disponible.
- Corrutina `put(item)` corrutina que agrega el elemento `item` en la cola. En caso que la cola esté llena la corrutina se bloquea hasta que haya espacio disponible.

Apéndice G

Módulo *wsncommunication*

En el proyecto de fin de carrera “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1], se implementó el módulo *wsncommunication* el cual permite la comunicación del Nodo Controlador con la red HAN. Este módulo brinda las funciones básicas necesarias para obtener los nodos presentes en la red, adquirir mediciones desde los Nodos de Carga y enviar comandos a los mismos. En la Figura G.1 se encuentran las funciones y las clases definidas por este módulo.

El módulo *wsncommunication* utiliza la implementación de COAP presente en la librería *aiocoap* de Python. Esta librería a su vez hace uso de la librería *asyncio*¹.

A continuación se describen las funciones del módulo utilizadas en el presente proyecto y las modificaciones realizadas al mismo. En caso que el lector desee obtener más información acerca de este módulo consultar la documentación del proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1].

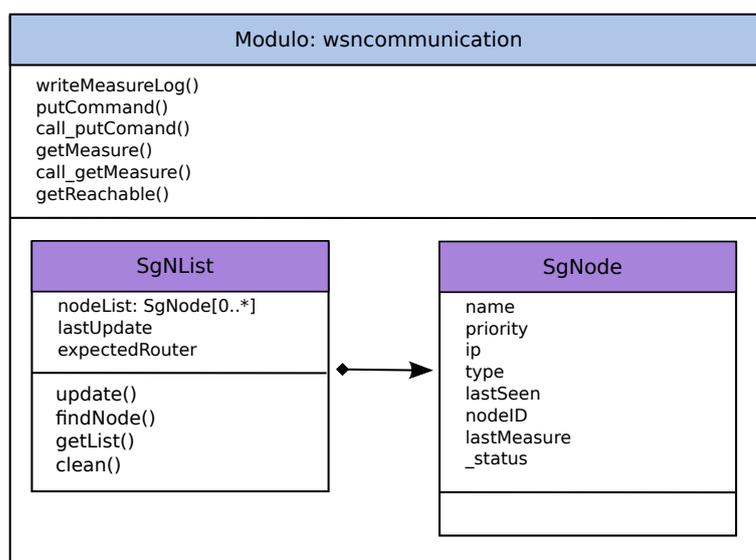


Figura G.1: Diagrama de módulo *wsncommunication*

¹Por más información acerca de la librería *asyncio* consultar el Apéndice F.

G.1. Funciones globales del módulo

Función: `getReachable()`

Esta función realiza una verificación de los nodos de la red, devolviendo una lista con las direcciones IPv6 accesibles (instancias de la clase `ipaddress.IPv6Address`). Se asume que el prefijo de la subred IPv6 es: “aaaa::c30c:0:0:0/64”. La búsqueda de los nodos se realiza enviando comandos `ping6` a cada uno de los nodos de la subred. Si el nodo responde el mensaje se agrega su dirección IPv6 a la lista.

En [1] se aclaró que está estrategia de búsqueda no es adecuada en redes de tamaño medio o grande. Se propusieron posibles mejoras a la misma, pero la implementación de estas modificaciones no se encuentra dentro del alcance de este proyecto.

Corrutina: `putCommand(node, comm, [recurso])`

Esta corrutina permite enviar comandos a los nodos. Recibe como parámetro un objeto del tipo `SgNode` representando al nodo que se quiere comandar y un string `comm` con el comando a enviar a través de CoAP.

Modificaciones realizadas:

En el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1] se consideró una única manera de comandar los electrodomésticos, la cual se efectuaba mediante la conmutación de un relé. Por esta razón se consideraba únicamente la existencia de un sólo tipo recursos actuadores llamado `relay` y este es el recurso utilizado por defecto en la implementación original de la corrutina `putCommand`.

En este proyecto se adicionaron otras maneras de comandar los electrodomésticos y se crearon nuevos recursos². Por ello, se modificó la implementación de la corrutina `putCommand` de manera que el recurso del nodo a utilizar sea pasado como parámetro cuando se invoca la corrutina.

Además de brindar la posibilidad de tener más de un tipo de recurso, esta modificación brinda la posibilidad de que un mismo Nodo de Carga controle más de un electrodoméstico. En este caso, el Nodo Controlador envía las órdenes al mismo Nodo de Carga, es decir a la misma dirección IPv6, pero a distintos recursos dentro del nodo.

Corrutina: `getMeasure(node, [recurso])`

Esta corrutina implementa una de las principales funcionalidades del módulo, que es la obtención de las mediciones de un nodo. Recibe como parámetro un objeto de la clase `SgNode`, el cual representa el nodo al que se le debe enviar la consulta. La corrutina se encarga de encaminar la consulta al nodo, procesar la respuesta y devolver el resultado obtenido.

Modificaciones realizadas:

En el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1] se consideró que los Nodos de Carga sólo miden el consumo de los electrodomésticos. En consecuencia, se consideró la existencia de un sólo recurso de medición, denominado `Mediciones`, el cual es utilizado por defecto en la implementación original de la corrutina `getMeasure`.

En este trabajo se agregó la posibilidad de que los Nodos de Carga midan otras magnitudes y se crearon nuevos recursos. Por esta razón, se modificó la implementación de la corrutina `getMeasure` de manera que el recurso del nodo a medir sea pasado como parámetro cuando se invoca la misma.

²ver Apéndice L

G.2. Representación de nodo y listas de nodos

Por otra parte, la implementación de la corrutina original invocaba a la función `writeMeasureLog()` luego de cada medición. Esto se eliminó en este trabajo, dado que el registro de mediciones fue desarrollado de forma distinta.

Función: `writeMeasureLog(data, nodeID)`

Esta función escribe entradas en el log de medidas, agregando una marca de tiempo correspondiente al momento en que se registran las mismas. Esta función no fue utilizada en este trabajo.

G.2. Representación de nodo y listas de nodos

Clase: `SgNode`

Esta clase sintetiza toda la información relacionada a un Nodo de Carga. Se almacena la siguiente información:

- **name:** *string* que representa al nodo.
- **priority:** *entero* que indica la prioridad del nodo. En el presente trabajo no se hizo uso de este atributo, por lo que todos los nodos presentaban `priority=1` (máxima prioridad).
- **type:** *string* que identifica las capacidades del nodo.
Posibles valores: `Measure`, `Command`, `Measure&Command`, `BorderRouter`, `Dummy`.
- **ip:** instancia de la clase `ipaddress.IPv6Address` que almacena la dirección IPv6 del nodo.
- **lastSeen:** *float* utilizado como etiqueta temporal, que indica la última vez que fue visto el nodo.
- **lastMeasure:** *entero* con la última medida reportada por el nodo.
- **_status** *entero* que indica el estado del nodo. Este atributo no será utilizado en este proyecto.

Clase: `SgNList`

El diseño del módulo contempla la necesidad de obtener un fácil acceso al listado de nodos conectados al sistema. Para ello se diseñó la clase `SgNList` que representa una lista de todos los nodos del sistema. Ofrece también funcionalidades para actualizar la lista, buscar nodos para fijar sus propiedades y operar sobre ellos.

Esta clase cuenta con los siguientes atributos, que representan el estado de la lista:

- **_n1:** lista de instancia de `SgNode` accesibles.
- **lastUpdated:** *float* que indica la hora de la última actualización.
- **expectedRouter:** *string* con la dirección del nodo que actúa como border-router.

Además, las listas de nodos cuentan con los siguientes métodos que permiten realizar actualizaciones y buscar nodos:

- **getList():** devuelve la lista `_n1`.
- **update():** método que permite actualizar la lista de nodos, buscando nuevos nodos que pueden haberse incorporado a la red recientemente. Dentro de este método, se usa de forma auxiliar la función global `getReachable()` para consultar qué nodos son accesibles.
- **clean():** método que realiza una limpieza de la lista, eliminando los nodos de los que no se registra actividad reciente.
- **findNode(name, dirip, type='Dummy'):** función que busca a un nodo con determinada dirección IP (incluida como parámetro en la llamada) dentro de la lista. En el caso de

Apéndice G. Módulo wsncommunication

que se encuentra el nodo se modifica su nombre y tipo con los parámetros **name** y **type** respectivamente. La función devuelve una referencia al nodo encontrado.

Apéndice H

Tablas de Base de Datos

En este apéndice se detalla cada una de las tablas que fueron creadas en la base de datos para representar los parámetros de los electrodomésticos, las configuraciones del usuario y el resto de las variables necesarias para la optimización del sistema.

A continuación se describen los nombres de las tablas, y para cada tabla, los campos de las mismas, los tipos de variables y una breve descripción de que valores se guardaran en cada uno de estos campos¹.

Temperatura Agua:

Esta tabla contiene la temperatura del agua para cada mes.

Campo	Tipo	Observaciones
Mes	Integer	Entero que representa el mes del año.
Temperatura	Float	Temperatura del agua para ese mes.

Temperatura Ambiente:

Esta tabla guarda para cada hora la temperatura ambiente así como también las predicciones para los próximos días.

Campo	Tipo	Observaciones
Fecha	Date	
Hora	Integer	
Temperatura	Float	Temperatura ambiente para la hora determinada.

Precio:

Esta tabla almacena el precio de energía para cada hora del día.

Campo	Tipo	Observaciones
Fecha	Date	
Hora	Integer	
Precio	Float	Precio en \$/kWh para esa hora.

¹Para mas detalle sobre los parámetros de los electrodomésticos consultar: Apéndice D y Apéndice B.

Apéndice H. Tablas de Base de Datos

Electrodomésticos:

Esta tabla contiene la lista de todos los electrodomésticos controlados por el sistema HEMS.

Campo	Tipo	Observaciones
Id	Integer	Identificador del electrodoméstico
Nombre	Varchar	Nombre corto con el que se identifica el electrodoméstico.
Tipo	Char(2)	Caracteres que identifican qué tipo de electrodoméstico es: HVAC, Calefón, etc.
IdActuador	Integer	Identifica el actuador del Nodo de Carga asociado al electrodoméstico.
Fecha Modificación	Date	Fecha de última modificación por el usuario. Es visible en la interfaz Web para que el usuario pueda saber cuándo fueron los últimos cambios.
Descripción	Text	Descripción detallada para que el usuario pueda identificar claramente el electrodoméstico en la Interfaz Web.
IdSensor	Integer	Identifica el sensor de Medidas asociado al electrodoméstico si corresponde.

Calefón Fijo:

Esta tabla guarda los parámetros específicos para cada uno de los electrodomésticos tipo Calefón del HEMS.

Campo	Tipo	Observaciones
Id	Integer	Identificador del electrodoméstico.
TsSlide	Integer	Número de <i>timeslots</i> en los que se divide una hora.
QMax	Integer	Energía máxima que se le puede suministrar al electrodoméstico en una hora.
Capacidad Térmica	Float	Capacidad térmica por unidad de volumen [Wh/Km^3].
Conductancia Térmica	Float	Conductancia térmica [W/K].
Volumen	Integer	Volumen del calefón en litros.
Rendimiento	Float	Rendimiento del calefón.
Temperatura Inicial	Float	Temperatura al inicio del día del agua en el calefón. Se actualiza diariamente con los datos obtenidos del sensor temperatura del calefón.
Temperatura Máxima	Float	Temperatura máxima que puede alcanzar el agua en el calefón.
Es Interrumpible	Boolean	Indica si es un electrodoméstico interrumpible.
Curva Consumo	Text	Array de energía que indica por minuto cómo es el consumo del electrodoméstico.
Parámetro A	Float	Parámetro <i>a</i> de la función de utilidad.
Parámetro B	Float	Parámetro <i>b</i> de la función de utilidad.

Calefón Día:

Esta tabla almacena la configuración de cada calefón para cada día de la semana.

Campo	Tipo	Observaciones
Id	Integer	Identificador del Electrodoméstico.
Orden	Integer	Identificador del Día.
Día	Varchar	Nombre del día de la semana.
Temperatura Mínima	Float	Temperatura mínima que admite el usuario.
Temperatura Confort	Float	Temperatura que desea el usuario.
Pulso Útil	Text	Vector que indica cuándo el usuario desea que el agua cumpla con los criterios de confort.
Encendido	Boolean	Indica si el electrodoméstico debe encender ese día.

HVAC Fijo:

Esta tabla guarda los parámetros específicos de cada uno de los aires acondicionados del hogar.

Campo	Tipo	Observaciones
Id	Integer	Identificador del electrodoméstico.
TsSlide	Integer	Número de <i>timeslots</i> en los que se divide una hora.
QMaxCalor	Integer	Energía máxima que se le puede suministrar al HVAC cuando está en modo calefacción.
QMaxFrio	Integer	Energía máxima que se le puede suministrar al HVAC cuando está en modo refrigeración.
Capacidad Térmica	Float	Capacidad térmica por unidad de volumen [Wh/Km^3].
Conductancia Térmica	Float	Conductancia térmica [W/K].
Rendimiento Calor	Float	EER en modo calefacción.
Rendimiento Frío	Float	EER en modo calefacción.
Temperatura Inicial	Float	Temperatura al inicio del día de la habitación.
Es Interrumpible	Boolean	Indica si es un electrodoméstico interrumpible
Curva Consumo	Text	Array de Energía por minuto como se comporta la carga.
Modo Calor	Boolean	Indica el modo en que está configurado el HVAC.
Parámetro A	Float	Parámetro <i>a</i> de la función de utilidad.
Parámetro B	Float	Parámetro <i>b</i> de la función de utilidad.

HVAC Día:

Esta tabla almacena la configuración de cada aire acondicionado para cada día de la semana.

Campo	Tipo	Observaciones
Id	Integer	Identificador del Electrodoméstico.
Orden	Integer	Identificador del Día de la semana.
Día	Varchar	Nombre del día de la semana.
Pulso Útil	Text	Vector que indica cuándo el usuario desea que la temperatura de la habitación cumpla con el criterio de confort y las restricciones.
Encendido	Boolean	Indica si el electrodoméstico debe encender ese día.

Apéndice H. Tablas de Base de Datos

HVAC Hora:

Esta tabla contiene la configuración de temperatura de cada aire acondicionado para cada hora.

Campo	Tipo	Observaciones
Id	Integer	Identificador del Electrodoméstico.
Orden	Integer	Identificador del Día de la semana.
Día	Varchar	Nombre del día.
Hora	Integer	Hora del día de la semana.
Temperatura Mínima	Float	Temperatura mínima que admite el usuario cuando el aire acondicionado está en modo calefacción.
Temperatura Confort	Float	Temperatura que desea el usuario en la habitación
Temperatura Máxima	Float	Temperatura máxima que admite el usuario cuando el aire acondicionado está en modo refrigeración.

Carga Discreta Fijo:

Esta tabla guarda los parámetros específicos de cada una de las cargas discretas del hogar.

Campo	Tipo	Observaciones
Id	Integer	Identificador del electrodoméstico.
TsSlide	Integer	Número de <i>timeslots</i> en los que se divide una hora.
QMax	Integer	Energía máxima que se le puede suministrar a la carga en un hora.
QCiclo	Integer	Energía total que se gasta en un ciclo completo.
EsInterrumpible	Boolean	Indica si es un electrodoméstico interrumpible.
CurvaConsumo	Text	Array de energía que indica por minuto cuánto consume el electrodoméstico.
Parámetro A	Float	Parámetro <i>a</i> de la función de utilidad.
Parámetro B	Float	Parámetro <i>b</i> de la función de utilidad.

Carga Discreta Día:

Esta tabla almacena la configuración de cada carga discreta para cada día de la semana.

Campo	Tipo	Observaciones
Id	Integer	Identificador del Electrodoméstico.
Orden	Integer	Identificador del Día.
Día	Varchar	Nombre del día de la semana.
Tiempo Mínimo	Float	Tiempo mínima que admite el usuario se use la carga en el día (si corresponde).
Tiempo Máximo	Float	Tiempo máximo que admite el usuario se use la carga en el día (si corresponde).
Pulso Útil	Text	Vector que indica cuándo el usuario admite que se encienda el electrodoméstico.
Encendido	Boolean	Indica si el electrodoméstico debe encender ese día.

Config:

Esta tabla contiene configuraciones globales del sistema.

Campo	Tipo	Observaciones
Id	Integer	
Atributo	Varchar	Nombre del valor de configuración del sistema.
Descripción	Text	Descripción del atributo.
Valor	Varchar	Valor que tiene configurado el atributo.

Config Día:

Esta tabla guarda para cada día la semana las horas que el usuario no utilizará otras cargas en el hogar.

Campo	Tipo	Observaciones
Orden	Integer	Identificador del Día.
Día	Varchar	Nombre del día de la semana.
Usuario Sleep	Text	Vector que indica en que horas el usuario utiliza las cargas del hogar que no son administradas por este sistema.

Usuario:

Esta tabla almacena la información de cada uno de los usuarios del sistema.

Campo	Tipo	Observaciones
Nombre	Varchar	Nombre del Usuario.
Password	Varchar	Contraseña del Usuario.

Rol Usuario:

Esta tabla contiene los roles asignados a cada usuario del sistema.

Campo	Tipo	Observaciones
Nombre	Varchar	Nombre del Usuario.
Rol	Varchar	Rol del Usuario.

Nodo de Carga:

Esta tabla guarda la información de los distintos Nodos de Carga del sistema.

Campo	Tipo	Observaciones
Id Nodo	Integer	Identificador del Nodo de Carga.
Nombre	Varchar	Nombre del Nodo de Carga.
IP	Varchar	Dirección IPv6 del Nodo de Carga
Recurso	Varchar	Recurso del Nodo de Carga al que se debe acceder.
Tipo	Char	Tipo de Nodo (Actuador, Sensor).
Periodicidad	Integer	Periodicidad en que se toma valor de medida (si corresponde).
Descripción	Varchar	Descripción más detalla del nodo .

Log Energía:

Esta tabla almacena las decisiones de consumo del algoritmo para cada uno de los electrodomésticos.

Apéndice H. Tablas de Base de Datos

Campo	Tipo	Observaciones
Id	Integer	Identificador del Electrodoméstico
Fecha	Date	
Hora	Integer	
Energía	Float	Consumo de energía para la hora determinada

Mediciones:

Esta tabla almacena las mediciones de cada uno de los sensores del sistema.

Campo	Tipo	Observaciones
Id Log	Integer	Identificador de la medida
Id Fila	Integer	
TimeStamp	Timestamp	Fecha de la medida
Id Sensor	Integer	Identificador del sensor de la medida
Medición	Integer	Valor de la medición

Apéndice I

Alternativas interfaz de usuario

A continuación se presenta el estudio comparativo que evalúa diferentes alternativas para la implementación de la interfaz de usuario. Se evalúan las ventajas y desventajas de cada una, haciendo foco en aquellos factores que son más relevantes para el proyecto en cuestión.

I.1. Alternativas consideradas

Por definición, una aplicación de escritorio o *desktop* es cualquier software que se puede instalar en un solo ordenador y se utiliza para realizar tareas específicas. En los inicios de la computación sólo se utilizaban aplicaciones de escritorio. Sin embargo, con la expansión del Internet y el comercio electrónico, el desarrollo de aplicaciones web ha cobrado mayor importancia.

Una aplicación web es aquella que se encuentra almacenada en un servidor remoto y puede ser accedida a través Internet por medio de un navegador. Con el crecimiento de la telefonía celular y tablets se incrementó el uso de aplicaciones móviles, tanto web como nativas.

Una aplicación móvil web es la que necesita de un navegador web para ejecutarse. Aplicación y datos pueden residir remotamente en un servidor u obtenerse del mismo dispositivo móvil.

Por otra parte, una aplicación nativa es aquella que se instala en el propio dispositivo y se desarrolla utilizando un lenguaje de programación compatible con el sistema operativo del dispositivo o de un *framework* de desarrollo.

I.2. Comparación

Generalmente escribir código para una aplicación de escritorio es más fácil, hay más controles disponibles y el programador tiene mayor control sobre cómo funciona la aplicación.

En cambio, al desarrollar aplicaciones web hay ciertas limitantes en los navegadores que deben ser tenidas en cuenta y además hay menos mecanismos disponibles para el desarrollador.

Por otra parte, las aplicaciones móviles nativas requieren un mayor esfuerzo de desarrollo, tanto en horas como en especialización del equipo. El dispositivo y los lenguajes utilizados son más limitados y complejos que en un entorno de servidor o de escritorio. Por lo antes dispuesto la selección de los dispositivos donde se ejecuta la aplicación nativa debe ser una decisión inicial. Normalmente esta decisión está condicionada por dos aspectos: la población objetivo y los requisitos técnicos.

Apéndice I. Alternativas interfaz de usuario

En el caso de la aplicación web no se cuenta con este problema ya que funciona en todas las plataformas, sólo se debe tener en cuenta la compatibilidad con el navegador. Es decir, una aplicación web funcionará en (casi) todos los smartphones mientras que las aplicaciones nativas requieren un desarrollo para cada uno de los sistemas operativos, e incluso para versiones diferentes de éstos.

La anterior es quizás la mayor ventaja de una aplicación web móvil sobre una aplicación nativa. La primera requiere una menor inversión inicial dado que no se precisa repetir la mayor parte del desarrollo para cada sistema operativo. Cabe destacar que el desarrollo no sólo implica la codificación sino también el testing y mantenimiento de la aplicación.

Instalación y Mantenimiento

Las aplicaciones de escritorio han estado tradicionalmente limitadas por el hardware en el que se ejecutan. Deben ser desarrollados para ser instaladas en un sistema operativo en particular, y pueden tener requisitos de hardware estrictos que deben cumplirse para garantizar su correcto funcionamiento. Las actualizaciones de las mismas deben ser aplicadas por el usuario directamente, lo cual agrega la dificultad de que el usuario debe tener conocimiento requerido para realizarla.

Las aplicaciones web, incluidas las móviles, implican un proceso de instalación y actualización más sencilla ya que sólo es necesario actualizar el servidor. Esto por consiguiente hace que la reparación de errores e incorporación de nuevas funcionalidades sea también más simple. Cabe resaltar que este proceso de actualización es transparente para el usuario.

Finalmente, la distribución de las aplicaciones móviles nativas de los sistemas operativos predominantes, iOS y Android, se realiza por medio de las tiendas oficiales o *stores*. Las aplicaciones móviles web no están admitidas en estas tiendas lo cual ocasiona que su descubrimiento y visibilidad sea menor que en el caso de las aplicaciones nativas.

Usabilidad

Cuando se programa utilizando una interfaz de usuario de escritorio, se obtiene un conjunto mucho más rico de controles de interfaz de usuario que cuando se usa HTML.

Si bien es cierto que la arquitectura cliente servidor de la web carece de la riqueza gráfica de las aplicaciones de escritorio que cuentan con controles inteligentes que dan mayor fluidez al trabajo del usuario. Esto ha sido resuelto con varias estrategias o tecnologías que le permiten a la web ofrecer todas sus ventajas pero ofreciendo a la vez controles visuales más amigables al trabajo del usuario.

Además actualmente la mayoría de los usuarios potenciales están familiarizados con la navegación gracias a Internet, por lo tanto también lo están con la interfaz gráfica de las páginas web.

Portabilidad

Una desventaja importante de las aplicaciones de escritorio es la escasa portabilidad dado que si son implementadas, por ejemplo, para un entorno Windows, no podrán ser usadas en una Tablet o un teléfono celular.

Lo mismo ocurre para las aplicaciones móviles nativas. Sólo pueden usarse en los dispositivos, tablets o celulares, para los que fueron desarrolladas.

Las aplicaciones web, por el contrario, están disponibles a través de varios tipos de dispositivos que cuenten con una navegador web: ordenadores, teléfonos móviles, tablets, etc.

I.3. Alternativa seleccionada

A modo de resumen en la tabla I.1 se encuentra las diferencias encontradas para cada una de las alternativas.

	Escritorio	Web	Móvil Nativa
Mantenimiento	Instalación y actualización debe realizarse en cada computadora.	Instalación y actualización centralizada.	Distribución y actualización mediante tienda oficial.
Portabilidad	Sólo puede usarse en PCs portátiles o de escritorio.	Puede accederse desde PCs, tablets y celulares.	Sólo puede accederse desde tablets y celulares.
Usabilidad	Mayor riqueza gráfica.	Menor riqueza gráfica.	Mayor riqueza gráfica.
Desarrollo	Hay más controles y herramientas disponibles.	Limitantes en los controles disponibles.	Alto costo de desarrollo, testing y mantenimiento para cada sistema operativo.

Tabla I.1: Tabla comparativa de alternativas de interfaz usuario

El dispositivo a desarrollar es este proyecto es innovador en la realidad uruguaya. Se desea que el mismo llegue a un gran espectro de usuarios, muchos de los cuales no tienen gran conocimiento de las nuevas tecnologías. Debido a esto se debe brindar una interfaz amigable y fácil de usar. Es importante la simplicidad para conseguir que el usuario no sienta barreras a la hora de enfrentarse con la misma y comience a utilizarla rápidamente sin inconvenientes.

Hoy en día la mayor parte de los usuarios potenciales están familiarizados con el uso de un navegador. Es por ello que se considera que la mejor opción es desarrollar una aplicación web. Asimismo, dado que lo que se busca es una aplicación sencilla que tenga sólo las funcionalidades necesarias para la comunicación con el Nodo Controlador, se entiende que las herramientas disponibles para el desarrollo web son más que suficientes.

Teniendo en cuenta que la aplicación debe estar disponible para la mayor cantidad de usuarios posibles si se optase por usar una aplicación móvil nativa la mejor solución, aunque más costosa, consiste en desarrollar la aplicación para todos los sistemas operativos posibles. Dado que esto requiere un gran esfuerzo de desarrollo y especialización del equipo esta opción fue descartada.

Apéndice J

Detalle Interfaz Web

El objetivo de este apéndice es el de describir las funcionalidades implementadas en la interfaz web de un modo detallado.

J.1. Información Sistema

Página Principal

La **Página Principal** del sistema muestra un resumen de la información básica que resulta de interés para el usuario. En la Figura J.1 se encuentra una imagen de la página principal. Esta página, así como todas las páginas de la interfaz, incluyen un encabezado que muestra el nombre de usuario si es que el mismo está autenticado. En el menú de ajustes de la izquierda de la página, se puede acceder a las funcionalidades de consulta datos y configuración del sistema. En la marco central de la página principal se muestran los precios de la energía eléctrica para el día actual. Para facilitar la interpretación se colocó un gráfico en forma de dial que marca el precio actual, indicando si el mismo se encuentra en un rango barato o en uno caro. Esta información facilita al usuario a comprender las decisiones tomadas por el HEMS y adicionalmente le da la información necesaria para que el mismo haga su propia gestión de los electrodomésticos no controlables por el sistema.

También en la **Página Principal** hay información sobre los electrodomésticos, tanto para el día actual como para el día próximo. En la Figura J.2 se encuentra una captura de esta página. En el panel superior se muestran los electrodomésticos que se van a encender en el corriente día y entre qué horas. Con esta información el usuario tiene un acceso visual, fácil de interpretar y resumido de la configuración del día en curso. En el panel inferior se muestra la configuración para el día siguiente. En él se indican cuáles son los electrodomésticos que están agendados para la optimización del día próximo, brindándose la posibilidad de activar o desactivar su agendado rápidamente. De esta forma se evita tener que editar las horas de uso para quitar un electrodoméstico del proceso de optimización. Para entrar en la página de edición de los electrodomésticos se cuenta con un acceso rápido en la columna **Editar**.

Apéndice J. Detalle Interfaz Web

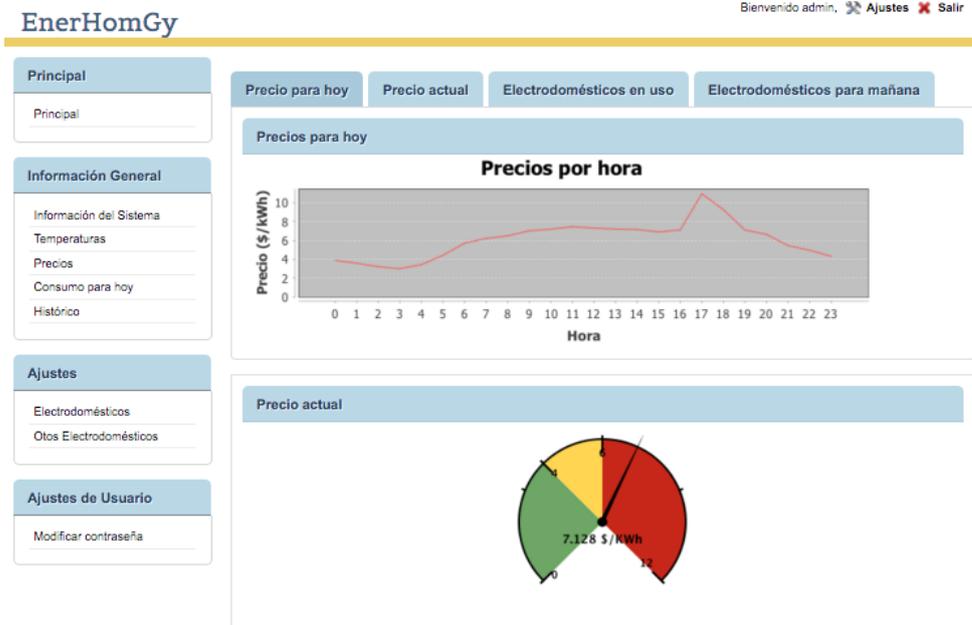


Figura J.1: Página Principal

Temperaturas

Seleccionando la opción **Temperaturas** del menú, se accede a los datos de temperaturas relevantes para el sistema.

En la Figura J.3 se puede ver el valor del pronóstico de la temperatura ambiente para el día actual y para el día siguiente. Esta información es útil para que el usuario sepa cuál es la temperatura exterior y de esta forma pueda decidir si quiere corregir la configuración del aire acondicionado del próximo día.

En la sección de temperaturas también se encuentra accesible por el usuario, la tabla de temperatura de agua de la línea de OSE Figura J.4 que es utilizada en el proceso de optimización del calefón.

J.1. Información Sistema

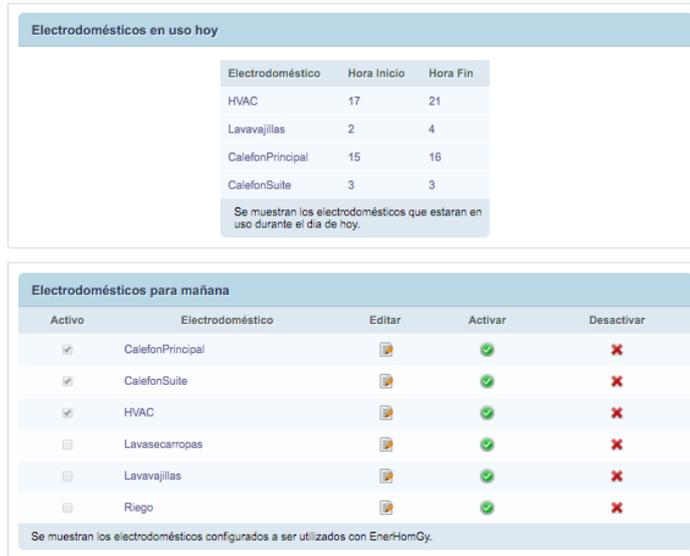


Figura J.2: Página Principal

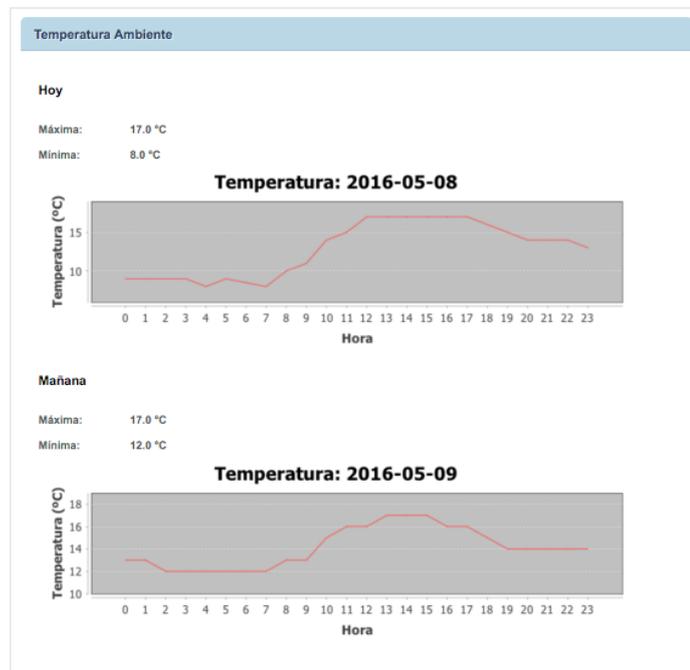


Figura J.3: Pronóstico de temperatura ambiente del día en curso y del próximo

Apéndice J. Detalle Interfaz Web



Figura J.4: Temperatura del agua de la línea de OSE

Uso de electrodomésticos

En la pestaña **Consumo para Hoy** se presenta un gráfico para cada uno de los electrodomésticos del hogar, con la distribución de consumo de energía para el día corriente. Esto le permite al usuario saber cuál es la simultaneidad de funcionamiento de los electrodomésticos controlados por el sistema. Con dicho gráfico, se facilita la visualización de gasto energético real de cada electrodoméstico, de forma de que el usuario tenga presente cuáles son los que originan mayor gasto. Esta página puede observarse en la Figura J.5.

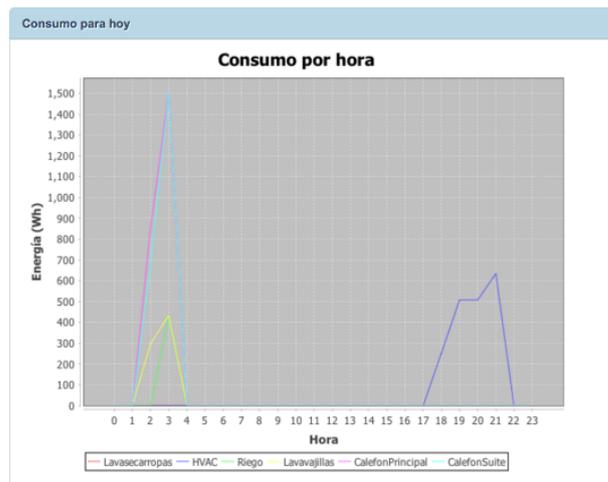


Figura J.5: Consumo de los electrodomésticos

Histórico

Desde el menú también es posible acceder a la página de **Histórico**. Desde allí se puede consultar el historial de gasto de los electrodomésticos controlables con la tarifa *DAP* o variable, y comparar el mismo con los gastos que hubiera tenido con un sistema de tarifa fija. Con este resultado el usuario puede evaluar si le es beneficioso contar con la tarifa variable. Esta consulta se puede realizar con distintos grados de disgregación como se muestra en la Figura J.6. Esto último se realiza con el fin de que el usuario pueda analizar su gasto. Los niveles de disgregación disponibles son los siguientes:

- **Gasto Mensual:** En la página de la Figura J.6a el usuario puede consultar el gasto mensual. Este formato de gasto habitual utilizado por UTE y por lo tanto es el que le resulta más natural al usuario.
- **Gasto Diario:** En la página de la Figura J.6b el usuario puede consultar el gasto de los últimos días. Esta información le sirve al usuario para saber si tiene un consumo muy elevado en algún día en particular, con lo que puede ajustar sus decisiones para días futuros. Además al compararlo con al tarifa fija se puede determinar fácilmente si ese gasto excesivo se debió a que la tarifa variable tomó valores elevados de precio.
- **Gasto por electrodoméstico:** En la página de la Figura J.6c el usuario puede consultar el gasto por electrodoméstico en los últimos días. Con esta información el usuario puede identificar cuales fueron sus mayores fuentes de consumo energético.

Apéndice J. Detalle Interfaz Web



(a) Gasto mensual

Gasto últimos días			
Día	Gasto DAP (\$)	Gasto FP (\$)	Ahorro (\$)
04-29	13.29	17.44	4.15
04-30	34.80	44.27	9.68
05-01	20.30	23.61	3.31
05-02	25.31	29.04	3.73
05-03	40.73	50.69	9.96
05-04	54.10	62.17	8.07
05-05	33.66	37.15	3.49
05-06	51.29	46.22	-5.07
05-07	66.32	66.82	0.51
05-08	30.28	41.95	11.66

(b) Gasto diario

Gasto por Electrodoméstico						
Electrodoméstico	Hoy DAP (\$)	Hoy FP (\$)	Hoy Ahorro (\$)	Ayer DAP (\$)	Ayer FP (\$)	Ayer Ahorro (\$)
Lavasecarrapas	0.00	0.00	0.00	9.26	10.92	1.66
HVAC	12.78	10.48	-2.30	40.36	29.24	-11.12
Riego	1.30	2.39	1.09	0.00	0.00	0.00
Lavavajillas	2.26	4.04	1.78	2.55	4.04	1.50
CalefonPrincipal	7.26	13.01	5.75	6.67	10.69	4.02
CalefonSuite	6.68	12.02	5.34	7.48	11.93	4.45

DAP: Day-Ahead Prices
FP: Flat Prices

(c) Gasto diario por Electrodoméstico

Figura J.6: Resultados de gastos con diferentes niveles de desagregación

J.2. Configuración Electrodomésticos

Electrodomésticos

The screenshot shows the 'Electrodomésticos' configuration page in the EnerHomGy system. The page includes a sidebar with navigation options and a main table listing configured appliances.

Id	Electrodoméstico	Tipo	Descripción	Modificado	Editar
1	CalefonPrincipal	C	Calefon 60L	2016-03-23 17:17:56.0	
2	CalefonSuite	C	Calefon 60L	2016-03-23 17:17:56.0	
3	HVAC	H	HVAC	2016-03-23 17:17:56.0	
4	Lavasecarropas	DC	Lavasecarropas Midea	2016-03-23 17:17:56.0	
5	Lavavajillas	DC	Lavavajillas	2016-03-23 17:17:56.0	
6	Riego	DT	Bomba riego 0.5 HP	2016-03-23 17:17:56.0	

Se muestran los electrodomésticos configurados a ser utilizados con EnerHomGy.

Figura J.7: Página de configuración de electrodomésticos

En la opción **Electrodomésticos** el usuario puede apreciar la información correspondiente a los distintos electrodomésticos que forman parte del sistema. En la Figura J.7 se encuentra la página correspondiente a la configuración por electrodoméstico. Entre otras cosas se puede ver el tipo de electrodoméstico, una breve descripción del mismo, así como también la fecha de última modificación realizada en su configuración.

Calefón

Al editar cualquier carga del tipo C (Calefón) se despliega una página como la que se ve en la Figura J.8b. La misma contiene información más detallada del electrodoméstico seleccionado. Por otra parte, incluye una tabla donde se indican los días de la semana en los que se desea que esta carga sea encendida y los valores de temperatura configurados para cada uno de estos días. Para cada uno de los días se da la posibilidad de activar o desactivar rápidamente el calefón y también editar su uso. La página que se despliega es la que se ve en la Figura J.8a. En ella el usuario puede modificar tanto la temperatura mínima y de confort para este día así como modificar el rango horario en el cual pretende que el calefón tenga el agua caliente disponible para su uso.

Apéndice J. Detalle Interfaz Web

CalefonPrincipal

Lunes Martes Miércoles Jueves Viernes Sábado Domingo

Ingrese los datos a continuación

Temperatura Mínima: Temperatura Confort:

Modificar

Horas de uso

Modificar Horas

Modifica la temperatura mínima y de confort para el día lunes

(a) Edición

CalefonPrincipal

Información General

Volumen: 60 l
 Temperatura Máxima: 80.0 °C
 Interrumpible: SI

Habilitado	Día	Temperatura Mínima	Temperatura Confort	Editar	Activar	Desactivar
<input checked="" type="checkbox"/>	lunes	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	martes	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	miércoles	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	jueves	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	viernes	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	sábado	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	domingo	45.0	55.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

(b) Configuración semanal

Figura J.8: Páginas de electrodoméstico tipo C (calefón)

HVAC

Para editar el electrodoméstico la primera página desplegada es la de la Figura J.9, que muestra el modo de funcionamiento, la energía máxima y el agendado semanal del mismo. Se ofrece la posibilidad de activar o desactivar rápidamente el HVAC para cada uno de los días, así como también editar su configuración.

HVAC				
Información General				
Modo:	Frio			
Consumo Máximo:	1470 Wh			
Interrumpible:	SI			
Habilitado	Día	Editar	Activar	Desactivar
<input checked="" type="checkbox"/>	lunes			
<input checked="" type="checkbox"/>	martes			
<input checked="" type="checkbox"/>	miercoles			
<input checked="" type="checkbox"/>	jueves			
<input checked="" type="checkbox"/>	viernes			
<input checked="" type="checkbox"/>	sabado			
<input checked="" type="checkbox"/>	domingo			

Figura J.9: Páginas de configuración inicial del electrodoméstico tipo H (aire acondicionado)

Las páginas de modificación de la configuración del electrodoméstico son las de la Figura J.10. Al presionar el botón de edición se abre una pagina dónde se muestra el rango de horas en las que el sistema controlará la temperatura de la habitación. La consigna de temperatura se muestra en conjunto en la **Tabla de temperaturas**, que aparece en la Figura J.10b. Para modificar la consigna de temperatura se despliega la página de la Figura J.10c. En esta última página, el usuario debe elegir entre que rango horario quiere hacer la modificación y seleccionar las temperaturas mínima, máxima y de confort que desea.

Apéndice J. Detalle Interfaz Web

The screenshot shows the 'EnerHomGy' web interface. At the top right, it says 'Bienvenido admin.' and has links for 'Ajustes' and 'Salir'. The left sidebar contains a 'Principal' section with a search bar, an 'Información General' section with links for 'Información del Sistema', 'Temperaturas', 'Precios', and 'Uso para hoy', and an 'Ajustes' section with links for 'Electrodomésticos', 'Otros Electrodomésticos', and 'Histórico'. The main content area is titled 'HVAC' and has tabs for 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sabado', and 'Domingo'. Below the tabs is a 'Horas de uso' bar chart showing usage hours from 0 to 24. A 'Modificar Horas' button is located below the chart. At the bottom of the screenshot, the 'Tabla de Temperaturas' section is partially visible.

(a) Horas de uso

Tabla de Temperaturas

Habilitado	Hora	Temperatura Mínima	Temperatura Confort	Temperatura Máxima
<input type="checkbox"/>	0	20	21	25
<input type="checkbox"/>	1	20	21	25
<input type="checkbox"/>	2	20	21	25
<input type="checkbox"/>	3	20	21	25
<input type="checkbox"/>	4	20	21	25
<input type="checkbox"/>	5	20	21	25
<input type="checkbox"/>	6	20	21	25
<input type="checkbox"/>	7	20	21	25
<input type="checkbox"/>	8	20	21	25
<input type="checkbox"/>	9	20	21	25
<input type="checkbox"/>	10	20	21	25
<input type="checkbox"/>	11	20	21	25
<input type="checkbox"/>	12	20	21	25
<input type="checkbox"/>	13	20	21	25
<input type="checkbox"/>	14	20	21	25
<input type="checkbox"/>	15	20	21	25
<input checked="" type="checkbox"/>	16	20	21	25
<input checked="" type="checkbox"/>	17	20	21	25
<input checked="" type="checkbox"/>	18	20	21	25
<input checked="" type="checkbox"/>	19	20	21	25
<input checked="" type="checkbox"/>	20	20	21	25
<input checked="" type="checkbox"/>	21	20	21	25
<input type="checkbox"/>	22	20	21	25
<input type="checkbox"/>	23	20	21	25

[Modificar](#)

(b) Tablas de temperatura por hora

HVAC Lunes

Ingrese los datos a continuación

Desde	Hasta	Temperatura Mínima	Temperatura Confort	Temperatura Máxima
<input type="text"/>				

[Modificar](#)

(c) Edición de temperatura y rango horario

Figura J.10: Páginas de configuración del aire acondicionado

Sistema de Riego

Las páginas correspondientes a la configuración de los electrodomésticos del tipo DT (consumo discreto de duración variable) son las de la Figura J.11. Para la edición se despliega la página de la Figura J.11a que contiene algunos datos adicionales acerca de este tipo de electrodoméstico.

Riego						
Información General						
Consumo Máximo:	650 Wh					
Interrumpible:	SI					
Habilitado	Día	Tiempo Mínimo	Tiempo Máximo	Editar	Activar	Desactivar
<input type="checkbox"/>	lunes	10	40			
<input type="checkbox"/>	martes	10	40			
<input type="checkbox"/>	miercoles	10	40			
<input checked="" type="checkbox"/>	jueves	20	40			
<input type="checkbox"/>	viernes	10	40			
<input type="checkbox"/>	sabado	10	40			
<input checked="" type="checkbox"/>	domingo	20	40			

(a) Configuración semanal

Riego						
Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
Ingrese los datos a continuación						
Tiempo Mínimo:		<input type="text"/>	Tiempo Máximo:		<input type="text"/>	
						<input type="button" value="Modificar"/>
Horas de uso						
						<input type="button" value="Modificar Horas"/>

(b) Edición del rango horario

Figura J.11: Páginas de electrodoméstico tipo DT (consumo discreto de duración variable)

El usuario puede ver para qué días está programado el uso del sistema de riego, así como también los tiempos de regado para cada uno de los días, así como también editar su configuración. Para editar la configuración se despliega la página de la Figura J.11b. En esta página

Apéndice J. Detalle Interfaz Web

el usuario puede, para cada día individualmente, ingresar el horario en que admite que se encienda el sistema de riego e indicar el tiempo mínimo y máximo que pretende que se riegue en el correr del día.

Otras Cargas Discretas

Existen otro tipos de cargas discretas de tipo DC (consumo discreto de duración fija), como pueden ser un lavarropas o un lavavajillas, que se caracterizan porque su uso se limita a un ciclo completo. Las páginas correspondientes a la configuración de los electrodomésticos del tipo DC son las de la Figura J.12.

Lavasecarropas

Información General

Consumo Máximo: 733 Wh
Interrumpible: NO

Habilitado	Día	Editar	Activar	Desactivar
<input type="checkbox"/>	lunes			
<input type="checkbox"/>	martes			
<input checked="" type="checkbox"/>	miercoles			
<input type="checkbox"/>	jueves			
<input type="checkbox"/>	viernes			
<input checked="" type="checkbox"/>	sabado			
<input type="checkbox"/>	domingo			

(a) Configuración semanal

Lavasecarropas

Lunes | Martes | Miercoles | Jueves | Viernes | Sabado | Domingo

Ingrese los datos a continuación

Horas de uso

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Activa el uso del electrodoméstico para el día lunes indicando las horas habilitadas para su uso

(b) Edición del rango horario

Figura J.12: Páginas de electrodoméstico tipo DC (consumo discreto de duración fija)

La página de configuración es como la que se puede ver en la Figura J.12a. Allí se muestran

J.2. Configuración Electrodomésticos

algunos datos más del electrodoméstico, incluyendo los días para los cuales el mismo está activo. Se ofrece la posibilidad de activar o desactivar rápidamente el electrodoméstico para cada uno de los días, así como también editar su configuración. La página de edición, apreciable en la figura J.12b, contiene una gráfica de barra que indica las horas del día en que se permite que el electrodoméstico sea encendido. Asimismo, aquí se puede acceder a modificar dicho rango horario, ya sea para incrementar el mismo, decrementarlo o moverlo dentro del día.

Otros Electrodomésticos

Existen otros electrodomésticos en el hogar que no son gestionados por el sistema. Si bien no se ejerce ningún tipo de control sobre los mismas, es deseable que el usuario informe en qué horas es más propenso a utilizar estos electrodomésticos que se encuentran fuera al sistema.

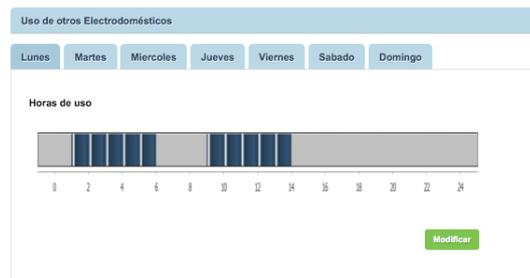


Figura J.13: Uso de otros Electrodomésticos

En la Figura J.13 se observa en forma de gráfico de barras las horas en las que el usuario manifiesta no va a estar utilizando otros electrodomésticos. Esto permite considerar la holgura disponible en el consumo energético total, en función de cuando el usuario se encuentra o no se encuentra en el hogar.

Modificar Horas de Uso



El formulario tiene un título 'Lavasecarropas - Lunes'. Debajo del título, hay un texto 'Ingrese los datos a continuación'. Hay dos campos de entrada: 'Desde:' con un valor de '00' y 'Hasta:' con un valor de '00'. Hay dos botones: 'Quitar horas' (rojo) y 'Agregar horas' (verde).

Figura J.14: Modificar horas de la barra horaria

Al querer modificar el rango horario de uso de cada electrodoméstico gestionado por el sistema HEMS así como también al indicar el horario en que el usuario no hace uso de los electrodomésticos no controlados se despliega la página de la Figura J.14. Luego de seleccionado un rango horario, y en función de si se quiere agregar o quitar horas, el usuario debe confirmar oprimiendo el botón correspondiente.

J.3. Usuarios del sistema

Login



The screenshot shows a login interface for 'EnerHomGy'. At the top center is the logo, which consists of a blue square containing a yellow sun, a blue line graph, and a bar chart with five bars of increasing height from left to right, colored in a gradient from blue to red. Below the logo, the text 'Bienvenido, por favor identifíquese:' is written in a dark blue font. Underneath this text are two white input fields with light gray borders. The first field is labeled 'Usuario:' and the second is labeled 'Contraseña:'. To the right of the second field is a green button with the white text 'Ingresar'.

Figura J.15: Login de Usuario

Para realizar algunas funcionalidades de configuración en el sistema es necesario que el usuario se autentique mediante una página de **Login**, como la de la Figura J.15. En esta página se le pide el usuario y contraseña para hacer efectiva la autenticación. Para acceder a la página de **Login** se selecciona el botón que se encuentra en la parte superior derecha de cada una de las páginas. Adicionalmente, el usuario será automáticamente redirigido a la página de **Login** cada vez que requiera hacer alguna modificación en la configuración si no ha sido debidamente autenticado previamente.

Modificar Contraseña

En caso que el usuario desee alterar su contraseña por cuestiones de seguridad se dispone, en el menú, de una opción para cambiar la contraseña del mismo como se observa en la Figura J.16



The screenshot shows a 'Modificar Contraseña' (Change Password) form. At the top left, there is a blue tab with the text 'Modificar Contraseña'. Below this, the text 'Ingrese su nueva contraseña' is displayed in a dark blue font. Underneath this text is a single white input field with a light gray border, labeled 'Nueva contraseña:'. To the right of the input field is a green button with the white text 'Modificar'.

Figura J.16: Modificar Contraseña

J.4. Manejo de Errores

El sistema esta preparado para manejar errores de parte del usuario. Se muestra una pantalla como se ve en la Figura J.17. En la misma se brinda una breve descripción del error encontrado para que el usuario no vuelva a cometerlo, así como la posibilidad de volver atrás para hacer nuevamente las modificaciones requeridas.

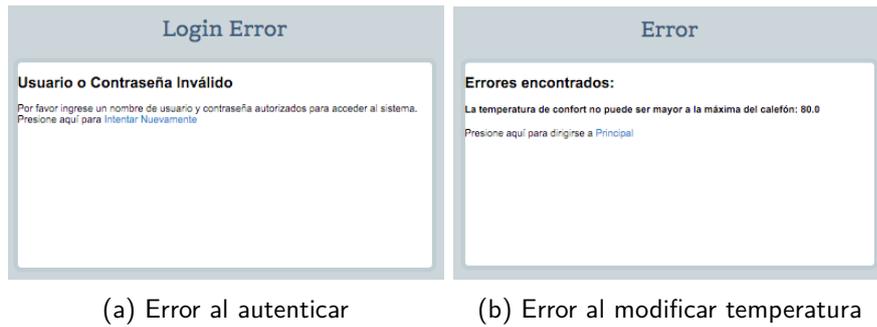


Figura J.17: Manejo de errores

Apéndice K

Configuración de los hogares tipo

En la Sección 8.1 se esbozó la configuración de electrodomésticos presentes en cada uno de los hogares tipo utilizados para la evaluación del sistema de gestión. En este apartado se encuentra en detalle la configuración y los parámetros de cada uno de los hogares.

K.1. Configuración y parámetros del Hogar Tipo 1

El Hogar Tipo 1 se compone por tres electrodomésticos, un calefón, un aire acondicionado y un lavarropas. Las configuraciones y características de cada electrodoméstico se detallan a continuación.

Variables fijas de cada electrodoméstico en el Hogar Tipo 1

En la siguiente tabla se encuentran las variables fijas utilizadas en el proceso de simulación para cada uno de los electrodomésticos.

Variable	Valor
Calefón:	
volumen	60 l
qMax	1500 W
tempMaxima	80 °C
tempAmbiente	22 °C
Aire acondicionado:	
Modo frío:	
qMax	1470 Wh
rendimiento	3.5 kW/kW
Modo calor:	
qMax	1680 Wh
rendimiento	3.75 kW/kW
Lavarropas:	
qCiclo	255 Wh
qMax	444 Wh

Apéndice K. Configuración de los hogares tipo

VARIABLES ESTACIONALES DE CADA ELECTRODOMÉSTICO DEL HOGAR TIPO 1

Las variables estacionales del hogar tipo 1, para cada uno de los electrodomésticos, se resumen en la siguiente tabla:

Variable	Verano		Otoño		Invierno		Primavera	
Calefón:								
pulsoUtil	L a V	06:30 a 08:00 19:00 a 21:00	L a V	06:30 a 08:00	L a V	06:30 a 08:00	L, Mi, V Ma, J	06:30 a 08:00 19:00 a 21:00 06:30 a 08:00
	S a D	08:00 a 10:00 20:00 a 22:00	S a D	20:00 a 22:00	S a D	20:00 a 22:00	S a D	20:00 a 22:00
TempMinima	40°C		45°C		50°C		45°C	
TempConfort	50°C		55°C		55°C		55°C	
Aire acondicionado:								
pulsoUtil	L a V	18:30 a 22:00	-	L a V	18:30 a 22:00	-	-	-
	S a D	11:00 a 22:00	-	S a D	11:00 a 22:00	-	-	-
TempMaxima	27°C		-		-		-	
TempConfort	21°C		-		24°C		-	
TempMinima	-		-		18°C		-	
Lavarropas:								
pulsoUtil	Mi	03:00 a 08:00	D	03:00 a 12:00	D	03:00 a 12:00	D	03:00 a 12:00
	D	03:00 a 12:00						
Usuario inactivo:								
vectorUsuarioInactivo	L a V	01:00 a 06:00 09:00 a 18:00	L a V	01:00 a 06:00 09:00 a 18:00	L a V	00:00 a 06:00 09:00 a 18:00	L a V	01:00 a 06:00 09:00 a 18:00
	S a D	02:00 a 07:00	S a D	02:00 a 07:00	S a D	01:00 a 07:00	S a D	02:00 a 07:00

K.2. Configuración y parámetros del Hogar Tipo 2

El hogar tipo 2 se compone por seis electrodomésticos, dos calefones, un aire acondicionado, un lavasecarropas, un lavavajillas y una bomba de riego. Las configuraciones y características de cada electrodoméstico se encuentran a continuación.

Variables fijas de cada electrodoméstico en el Hogar Tipo 2

En la siguiente tabla se encuentran las variables fijas utilizadas en el proceso de simulación para cada uno de los electrodomésticos.

Variable	Valor
Calefones principal:	
volumen	60 l
qMax	1500 W
tempMaxima	80 °C
tempAmbiente	22 °C
Calefón suite:	
volumen	60 l
qMax	1500 W
tempMaxima	80 °C
tempAmbiente	22 °C
Aire acondicionado:	
Modo frío:	
qMax	1470 Wh
rendimiento	3.5 kW/kW
Modo calor:	
qMax	1680 Wh
rendimiento	3.75 kW/kW
Lavasecarropas:	
qCiclo	1980 Wh
qMax	733 Wh
Lavavajillas:	
qCiclo	733 Wh
qMax	434 Wh
Riego:	
qMax	650 Wh
tMax	60 min
tMin	20 min

Variables estacionales de cada electrodoméstico del Hogar Tipo 2

Las variables estacionales del hogar tipo 2, para cada uno de los electrodomésticos, se resumen en la siguiente tabla:

Apéndice K. Configuración de los hogares tipo

Variable	Verano		Otoño		Invierno		Primavera	
Calefón principal:								
pulsoUtil	D a J V a S	17:00 a 20:00 19:00 a 21:00	D a J S	17:00 a 20:00 19:00 a 21:00	D, Ma, Ju S	17:00 a 20:00 19:00 a 21:00	D a J S	17:00 a 20:00 19:00 a 21:00
TempMinima		45°C		45°C		50°C		45°C
TempConfort		50°C		55°C		60°C		55°C
Calefón suite:								
pulsoUtil	L a V S a D	06:30 a 08:00 19:00 a 21:00 08:00 a 10:00 20:00 a 22:00	L a V S a D	06:30 a 08:00 20:00 a 22:00	L a V S a D	06:30 a 08:00 20:00 a 22:00	L, Mi, V Ma, J S a D	06:30 a 08:00 19:00 a 21:00 06:30 a 08:00 20:00 a 22:00
TempMinima		40°C		45°C		50°C		45°C
TempConfort		50°C		55°C		55°C		55°C
Aire acondicionado:								
pulsoUtil	L a V S a D	16:00 a 21:30 10:00 a 22:00	L a V S a D	18:00 a 22:00 10:00 a 22:00	L a V S a D	16:30 a 22:00 13:00 a 22:00	L a V S a D	18:00 a 22:00 10:00 a 22:00
TempMaxima		27°C		-		-		27°C
TempConfort		21°C		24 °C		24°C		21°C
TempMinima		-		18 °C		18°C		-
Lavasecarropas:								
pulsoUtil	Mi S	05:30 a 08:00 06:00 a 12:00	Mi S	05:30 a 08:00 06:00 a 12:00	Mi S	05:30 a 08:00 06:00 a 12:00	Mi S	05:30 a 08:00 06:00 a 12:00
Lavavajillas:								
pulsoUtil	L a V S a D	00:00 a 07:00 02:00 a 09:00	L a V S a D	00:00 a 07:00 02:00 a 09:00	L a V S a D	00:00 a 07:00 02:00 a 09:00	L a V S a D	00:00 a 07:00 02:00 a 09:00
Riego:								
pulsoUtil	M, J, S, D	00:00 a 08:00 22:00 a 23:00	J, D	00:00 a 09:00 21:00 a 23:00	D	00:00 a 09:00 21:00 a 23:00	M, J, D	00:00 a 08:00 22:00 a 23:00
Usuario Inactivo:								
vectorUsuarioInactivo	L a V S a D	01:00 a 06:00 02:00 a 07:00	L a V S a D	01:00 a 06:00 02:00 a 07:00	L a V S a D	01:00 a 06:00 02:00 a 08:00	L a V S a D	01:00 a 06:00 02:00 a 07:00

Apéndice L

Detalle Nodos de Carga

En el Capítulo 3 se describió la arquitectura del sistema HEMS implementado. A grandes rangos, el mismo está compuesto por un Nodo Controlador central y múltiples Nodos de Carga. El Nodo Controlador es el cerebro del sistema mientras que los Nodos de Carga son los intermediarios entre el Nodo Controlador y los electrodomésticos.

El presente trabajo se centró en la implementación del Nodo Controlador. Adicionalmente, se realizó una prueba experimental con electrodomésticos reales para comprobar su funcionamiento. Para ejecutar dicha prueba fue necesario implementar Nodos de Cargas básicos que comandaran los electrodomésticos y midieran las magnitudes de interés. El objetivo de este apéndice es describir en detalle cómo se implementaron los Nodos de Carga utilizados en la prueba experimental.

Cabe mencionar que los Nodos de Carga implementados en este proyecto fueron diseñados con el fin de poder ejecutar la prueba experimental. En consecuencia, no se buscaron soluciones genéricas ni eficientes, sino soluciones prácticas que funcionaran con los electrodomésticos empleados en la prueba.

L.1. Descripción de los Nodos de Carga

Los Nodos de Carga son los intermediarios entre el Nodo Controlador y los electrodomésticos. Cada electrodoméstico controlado por el sistema de gestión debe tener asociado al menos un Nodo de Carga capaz de comandarlo. Su función es ejecutar órdenes y reportar mediciones al Nodo Controlador.

Como se describió en el Capítulo 3, los Nodos de Carga están compuestos por un mote Z1 de Zolertia que opera sensores y/o actuadores. Se utilizó el sistema operativo *Contiki OS* en los motes Z1, el cual es compatible con una gran variedad de sensores y actuadores.

Módulos de un Nodo de Carga

Un Nodo de Carga puede presentar distintos módulos en función de los electrodomésticos a los que este asociado. Los módulos posibles son los siguientes:

- **Módulo de comunicación:** permite la comunicación del Nodo de Carga con el Nodo Controlador.
- **Módulo de comando:** permite comandar un electrodoméstico, ya sea encender/apagar, pausar/reanudar un proceso o cambiar el modo de operación del mismo.

Apéndice L. Detalle Nodos de Carga

- **Módulo de medición:** permite medir magnitudes de interés, como lo son la temperatura de una habitación, la temperatura de agua dentro de un calefón o la potencia consumida por los electrodomésticos.



Figura L.1: Módulos de un Nodo de Carga

Un Nodo de Carga debe presentar un módulo de comunicación y al menos un módulo de comando o medición. Aunque podría presentar múltiples módulos de comando y/o medición.

Todos los electrodomésticos deben tener asociado al menos un Nodo de Carga con módulo de comando. Esto no implica que deba existir una relación uno a uno de los Nodos de Carga con los electrodomésticos. Un mismo electrodoméstico puede presentar más de un Nodo de Carga asociado, uno de comando y otro de medición por ejemplo. Asimismo, un mismo Nodo de Carga puede comandar varios electrodomésticos.

En las siguientes secciones de este apéndice se describe el funcionamiento del módulo de comunicación, cómo se implementaron los distintos módulos de comando y medición utilizados en la prueba experimental.

L.2. Módulo de comunicación

El módulo de comunicación es el encargado permitir la comunicación del Nodo de Carga con el Nodo Controlador. Es gracias al módulo de comunicación que el Nodo Controlador puede utilizar los distintos módulos de comando y de medición presentes en el Nodo de Carga.

El módulo de comunicación fue implementado por Belcredi, Modernell y Sosa en “Controlador de energía domiciliario para una Red Eléctrica Inteligente”[1]. A continuación se describen las principales características del mismo.

Protocolo *CoAP*

En la Sección 8.4 se describió el stack de protocolos utilizado en la red HAN del sistema HEMS. Dentro de este stack, se utilizó el protocolo *CoAP* en la capa de aplicación. Se trata de un protocolo similar a HTTP que agrega funcionalidades que hacen posible utilizarlo en sistemas de recursos limitados.

El protocolo *CoAP* utiliza un modelo cliente/servidor y se basa en la arquitectura REST (Representational State Transfer). En esta arquitectura los servidores publican recursos, que son accedidos mediante un URI. Los clientes acceden a estos recursos a través de cuatro métodos: *PUT*, *POST*, *GET* y *DELETE*. En el sistema HEMS, un Nodo de Carga es un servidor *CoAP* y cada módulo de comando y medición está asociado a un recurso en el servidor. El Nodo Controlador es un cliente *CoAP* que hace uso de estos recursos para comandar los electrodomésticos y obtener medidas.

Erbium (ER)

Un beneficio importante de utilizar la plataforma *Contiki OS* en los motes Z1, es que cuenta con implementaciones del stack protocolos testeadas. En particular, se usó el motor REST *Erbium (ER)* que implementa el stack y brinda una serie de códigos de ejemplo muy útiles con las aplicaciones más usuales. En los Nodos de Carga se tomó como base el ejemplo `'er-example-server.c'`, el cual implementa un servidor *CoAP* con múltiples recursos. Puntualmente, se utilizaron dos de estos recursos como punto de partida:

- **Recurso 'leds'**. Se trata de un ejemplo típico para el uso de un actuador a través del método PUT. En este recurso se encienden/apagan los leds del mote de acuerdo al mensaje de consigna enviado desde el cliente. Este recurso fue empleado como base para los recursos de los módulos de comando.
- **Recurso 'battery'**. Se trata de un ejemplo típico para el uso de un sensor a través del método GET. En particular este recurso reporta la medida del sensor de batería del mote. El mismo fue utilizado como base para los recursos de los módulos de medición.

Como se mencionó anteriormente, el Nodo Controlador es un cliente *CoAP* y los Nodos de Carga son servidores *CoAP*. Por otra parte, Los módulos de comando y medición son recursos de los Nodos de Carga a los cuales el Nodo Controlador puede acceder mediante una URI. Por tanto, se debe implementar un recurso *CoAP* para cada uno de los módulos de comando y medición.

L.3. Módulos de comando

Los módulos de comando son los encargados de operar los electrodomésticos. Estos deben ser capaces de cambiar el estado del electrodoméstico de acuerdo a las órdenes recibidas desde el Nodo Controlador.

El sistema de gestión HEMS diseñado, requiere que todos los módulos de comando sean capaces procesar señales de 'ON' y 'OFF'. Ante una señal de 'ON' el módulo debe accionar el electrodoméstico para inicie o retome un proceso detenido. Y ante una señal de 'OFF' debe operar el electrodoméstico para que detenga su proceso y que pase a consumir la menor cantidad de potencia posible.

En la prueba experimental se utilizó un calefón y un lavasecarropas. Cada uno de ellos tiene su forma de ser comando y se implementó un módulo de comando distinto para cada uno de ellos. Por un lado, el calefón puede ser comandado mediante la conmutación de su fuente de alimentación. Se diseñó para el calefón un módulo que conmuta un relé. Cuando se recibe una señal de 'ON' se cierra el relé, y cuando se recibe un 'OFF' se abre el relé cortándose la alimentación del calefón.

Por otro lado, el lavasecarropas no se repone correctamente luego de un corte en la alimentación de energía. Si se corta la alimentación del dispositivo durante el proceso de lavado, este no retoma automáticamente el proceso cuando se repone la alimentación. El lavasecarropas cuenta con un botón 'inicio y pausa', que permite pausar y reanudar el proceso de lavado. Se diseñó un módulo de comando que acciona este botón.

Los módulos implementados fueron específicamente diseñados para los electrodomésticos de la prueba experimental. Sin embargo, estos son adaptables a otros electrodomésticos. Los dos principios utilizados, conmutación de la alimentación y emulación del accionamiento de un botón, permiten comandar una amplia gama de electrodomésticos.

Apéndice L. Detalle Nodos de Carga

Módulo de conmutación

Este módulo fue diseñado para el calefón y conmuta un relé para abrir y cerrar el circuito de potencia. Si bien este módulo fue diseñado para conmutar un calefón, el mismo puede ser utilizado para cualquier otra carga que cumpla con las especificaciones del relé seleccionado.

Relé electromecánico

En una primera instancia, se utilizó un relé de la empresa *Phidget* modelo 3051. Este fue el relé utilizado en el proyecto de fin de carrera “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1]. Las características más relevantes de este relé son las siguientes¹:

- Dos relés electromecánicos modelo HF3FF.
- Tensión máxima de carga (AC): 240V.
- Corriente máxima de carga (AC): 10A.
- Alimentación del circuito de control: [3,3V_{DC}, 12V_{DC}].

Para alimentar el circuito de control del relé se utilizó el puerto phidgets 5V_{DC} pre-soldado en el mote. Este puerto utiliza los pines 13 (+5V) y 11 (GND) del puerto JP1A. Para el comando del relé se utilizó el pin 12 (I/O) del mismo puerto.

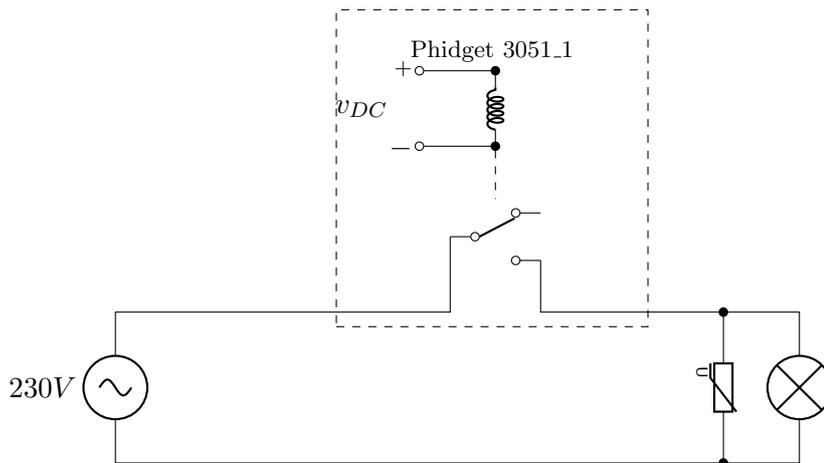


Figura L.2: Circuito de prueba de conmutación con lámpara

En un primera instancia, se probó el relé conmutando una lámpara de bajo consumo de 18W. En la Figura L.2 se puede ver el esquema de conexión utilizado. El relé consiguió conmutar la carga con éxito, pero se observó que en algunas ocasiones se perdía la comunicación con el Nodo Controlador luego de la conmutación.

Luego de múltiples pruebas se sacaron las siguientes conclusiones. La falla se debe a ruido electromagnético generado por el relé durante la conmutación. La misma parece afectar únicamente el sistema de comunicación dado que se realizaron pruebas con otros elementos del mote (timers, leds, botones, etc.) y estos continuaban funcionando correctamente. El único método encontrado para resolver la falla y que el mote volviese a su correcto funcionamiento

¹Phidgets. 3051_1 - Dual Relay Board. http://www.phidgets.com/products.php?product_id=3051. Especificaciones del producto.

fue el reseteo del microprocesador. Esta misma falla ya había sido reportada por el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” [1].

La tasa a la que se produce esta falla es muy alta, cerca de una 45% de las veces. Esto es inaceptable en el sistema de gestión por lo que fue necesario buscar una solución a este problema. Se tomaron las siguientes medidas con el fin de solucionar el problema:

1. **Utilizar una carga puramente resistiva.** Se cambió la lámpara de bajo consumo por resistencias. De esta manera se asegura que no existen transitorios que puedan producir ruido en el encendido/apagado de la carga.
2. **Aumentar la capacidad en el puerto de alimentación del mote.** Se agregaron capacitores a la entrada de la alimentación del mote para combatir una eventual caída de tensión producida por la energización/desenergización de la bobina.
3. **Varistor en bornes de la carga.** Se conectó un varistor en los bornes de la carga para filtrar sobretensiones al conmutar la misma.
4. **Apantallamiento con placa de cobre.** Se colocó el mote sobre una placa de cobre con el fin de desviar las ondas de campo magnético que podrían generar el problema.
5. **Filtro de línea.** Se agregó un filtro de línea entre la red y el mote con el fin de filtrar el ruido proveniente de la red.
6. **Transformador de aislación.** Se agregó un transformador de aislación entre la red y el mote, con el fin de mitigar las capacidades parásitas que aparecen entre los bobinados.
7. **Resetear del chip CC2420.** Se probó si apagar y volver a encender el chip de radio CC2420 recomponía al mote luego de la falla. Esto no fue así, la única manera encontrada de recomponerlo fue el reseteo del microprocesador.

No se observaron mejoras con ninguna de las modificaciones realizadas, la tasa de falla se mantuvo incluso combinando todas las modificaciones. Se procedió a cambiar el relé electro-mecánico por uno de estado sólido

Relé de estado sólido

Los relés de estado sólido utilizan la física de los semiconductores para realizar la conmutación del circuito. Las ventajas de este tipo de relés son que al conmutar no hay movimiento de partes mecánicas logrando reducir los transitorios producidos por la apertura del circuito y presentan una mayor velocidad de actuación.

Existen relés de estado sólido de distinto tipo de conmutación. Según [64] y [65] el tipo recomendado para las carga resistiva, como el calefón, son los *Zero-Crossing trigger*. En este tipo de relé la conmutación no se realiza en forma inmediata luego de recibir la señal de apertura, sino que se espera a que la sinusoidal de tensión pase por cero para abrir el circuito de potencia.

Se seleccionó un relé modelo SSR-40 DA para conmutar el calefón, en la Figura L.3 se encuentra una foto del mismo. Las características más importantes de este dispositivo son:²

- Relé de estado sólido *Zero-Crossing trigger*.
- Tensión máxima de carga (AC): 380V.
- Corriente máxima de carga (AC): 40A.

²Fotek. *SSR series*. Hoja de datos. URL: <https://cdn.sparkfun.com/datasheets/Components/General/SSR40DA.pdf>.

Apéndice L. Detalle Nodos de Carga

- Tensión de comando: $[3V_{DC}, 32V_{DC}]$.
 - $V_{DC} \geq 2,4V \rightarrow ON$.
 - $V_{DC} \leq 1,0V \rightarrow OFF$.



Figura L.3: Relé de estado sólido utilizado

La tensión en los pines I/O del mote no es adecuada para el comando del relé. Por lo que fue necesario diseñar un circuito *driver de encendido* para operar el mismo. En la figura L.4 se encuentra el circuito diseñado junto con los valores de componentes utilizados. Se utilizó un transistor NPN modelo BC547 el cual opera en las zonas de corte y saturación.

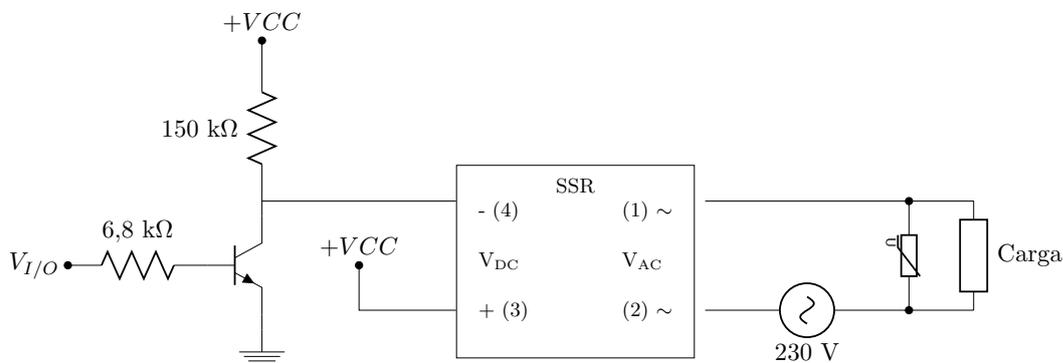


Figura L.4: Circuito de *driver* del SSR

Se observa que, si la salida del mote está en '0' ($V_{IO} \simeq 0$) el transistor está en corte. No circula corriente por el colector, por lo que la caída en $R_C = 150kV$ es nula. Esto hace que no exista diferencia de potencial entre los bornes de comando del relé. Bajo estas condiciones el circuito de alimentación del electrodoméstico se encuentra abierto.

Mientras que si la salida del mote está en '1' ($V_{IO} \simeq 2,4V$), el transistor está en zona de saturación. En este caso el borne (4) del SSR queda a $V_{CE\ sat} = 0,25V$. Por lo que la tensión en los bornes de comando de relé resulta $V_{DC} = 4,75 V$. Con esta diferencia de potencial el SSR cierra el circuito de alimentación de la carga.

Se realizaron pruebas con este relé utilizando el circuito de la Figura L.4 y una lámpara de bajo consumo de 18W. Con este relé se pudo conmutar la carga y no se volvió a observar la falla observada con el relé electromagnético. Luego se procedió a probar el circuito para conmutar un calefón y el módulo funcionó correctamente.

Si bien se logró solucionar la falla sustituyendo el relé, se observa el mote Z1 es muy sensible a ruidos del entorno. En un hogar real el dispositivo va a estar expuesto a múltiples fuentes de ruido por lo cual es deseable fortalecer el dispositivo en este aspecto. Para esto se implementó un procedimiento en el sistema de gestión que permite a los nodos recomponerse ante este tipo de fallas, el cual se describe más adelante en este apéndice.

Recurso: relay-commutation

Se implementó el recurso `relay-commutation` para que nodo pueda accionar el relé. Para la implementación del mismo se utilizó como base el recurso `leds` del ejemplo `'er-example-server.c'`. El recurso implementado permite el uso de los métodos PUT y POST y admite los siguientes comandos:

- `mode = on`: encender el electrodoméstico.
- `mode = off`: apagar el electrodoméstico.
- `mode = toggle`: cambiar el estado del electrodoméstico. Este comando fue implementado para agilizar las pruebas con los relés, pero no es utilizado por el Nodo Controlador.

En función de cuál es el comando recibido y cuál es el estado del relé el nodo determina si debe operar el relé o no. Para la conmutación del relé se utilizó un driver de relés diseñado para la plataforma Z1, el cual viene incluido dentro de la plataforma *Contiki*. Este driver cuenta con las funciones necesarias para poder operar el relé de forma sencilla.

- `relay_enable(pin)`: inicializa el driver y permite especificar el pin de control a utilizar.
- `relay_on()`: acciona el relé configurando el pin de control en '1'. En un relé normal abierto, se cierra el circuito de potencia.
- `relay_off()`: acciona el relé configurando el pin de control en '0'. En un relé normal abierto, se abre el circuito de potencia.
- `relay_toggle()`: intercambia el estado del relé, niega el estado anterior del pin de control.

Módulo comando Inicio - Pausa

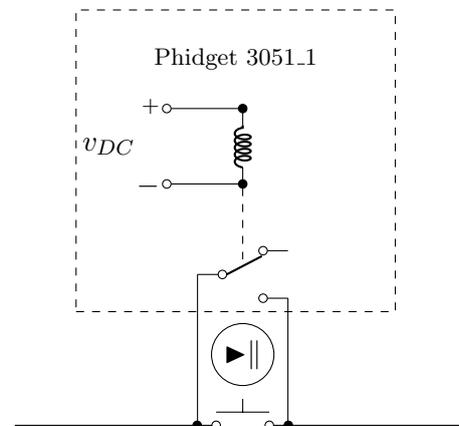
El lavasecarropas utilizado en la prueba experimental no puede ser operado mediante la conmutación de su alimentación. Luego de un corte de energía, el equipo requiere que el usuario resetee el proceso que se encontraba en ejecución y re programe el equipo para que inicie otro ciclo de lavado desde el principio. El lavasecarropas presenta un botón de inicio/pausa, el cual permite pausar y reanudar el lavado en cualquiera de sus etapas (ver Figura L.5a). Se encontró que la mejor forma de operar este electrodoméstico es mediante el accionamiento de este botón.

El botón de inicio/pausa presente en el lavasecarropas es un botón normal abierto. Cada vez que el usuario presiona el botón cierra el circuito momentáneamente enviando un pulso a la lógica del electrodoméstico. Se colocó un relé electromagnético *Phidget 3051_1* en paralelo con el botón de pausa, como se muestra en esquemático de la Figura L.5b. Accionando el relé se pueden emular los pulsos que genera el botón cuando es accionado por el usuario. En el manual del lavasecarropas, se especifica que si el botón es presionado por más de tres segundos el lavasecarropas reinicia el ciclo de lavado en ejecución. Por consiguiente, el pulso generado por el módulo de comando debe ser menor a tres segundos.

Apéndice L. Detalle Nodos de Carga



(a) Lavasecarropas utilizado en el proyecto



(b) Circuito de comando del lavasecarropas

Recurso: `relay_startANDpause`

Para que el Nodo Controlador pueda comandar el lavasecarropas se implementó un recurso llamado `relay_startANDpause`. Este recurso presenta un principio de funcionamiento similar al recurso `relay` y se utilizó el mismo driver de relé. La diferencia entre estos dos recursos se encuentra en que el `relay_startANDpause` no debe abrir o cerrar el circuito de potencia, sino que debe enviar pequeños pulsos.

Se observa que los pulsos de inicio/reinicio y pausa son idénticos. Esto provoca que el módulo de comando debe verificar cuál es el estado del electrodoméstico antes de ejecutar la orden recibida. Si se recibe una señal de 'ON' y el dispositivo ya se encuentra en estado 'ON', el módulo de comando no debe enviar ningún pulso dado que en el caso de que lo envíe el equipo interrumpirá su proceso. En este proyecto, el Nodo de Carga conoce cuál es el estado del electrodoméstico consultando una variable `state` dentro del programa y actualizando la misma cada vez que se envía un pulso.

El recurso implementado permite el uso de los métodos PUT y POST y admite los siguientes comandos:

- `mode = on`: Se verifica si el electrodoméstico está apagado. En caso afirmativo, envía un pulso para iniciar/reanudar el lavado y actualiza la variable `state`. En caso negativo, devuelve un mensaje de error.
- `mode = off`: análogo al caso anterior.

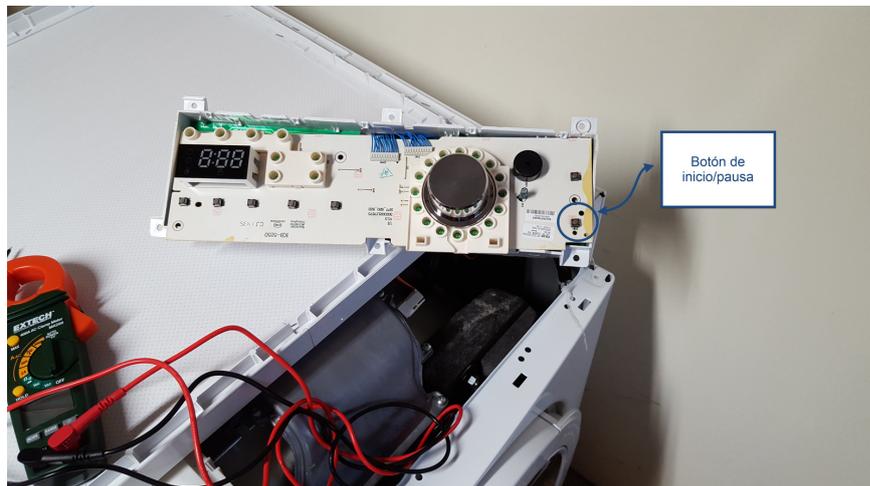
Los Nodos de Carga presentan un procedimiento que permite detectar problemas en el canal de comunicación con el Nodo Controlador. En caso de que se detecte un problema el mote se resetea. Cuando esto sucede se pierde la información guardada en la variable `state`. El módulo de comando debe cuidar de que la variable `state` no se quede desfasada con el verdadero estado del electrodoméstico. Una posible solución es medir la potencia que consume el dispositivo luego de reset para conocer en qué estado está el electrodoméstico. Esto no fue llevado a la práctica en este proyecto, dado que no se contaba con módulos de medición de potencia. Por tanto, se optó por una solución alternativa que consiste en forzar a que la variable `state` comience en 'OFF' luego de un reset y asegurar que el electrodoméstico pasa al estado 'OFF' antes del reset. Se puede afirmar que esta no es una solución óptima ya que pueden existir períodos de tiempo en los que el electrodoméstico no está en el estado deseado. Sin embargo, estos períodos son de corta duración, dado que el Nodo Controlador repite las

L.3. Módulos de comando

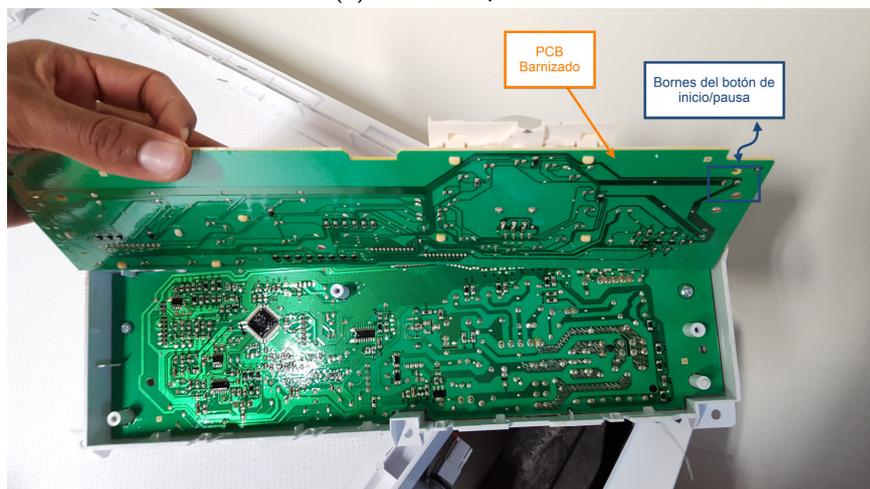
órdenes indicando el estado deseado cada un minuto.

Implementación en HW del módulo inicio-pausa

Se accedió al circuito interno del lavasecarropas para determinar cómo es el funcionamiento del botón de inicio/pausa. En la Figura L.6, se muestran las modificaciones realizadas al circuito con el fin de poder conectar el relé en paralelo con el botón de inicio/pausa. En Figura L.6a se muestra el panel de control del electrodoméstico sin la careta frontal observándose el botón de inicio/pausa. En Figura L.6b se muestra la placa de PCB, que presenta un barnizado de protección por lo que se decidió realizar la conexión del relé en los bornes del botón, como es posible apreciar en Figura L.6c.

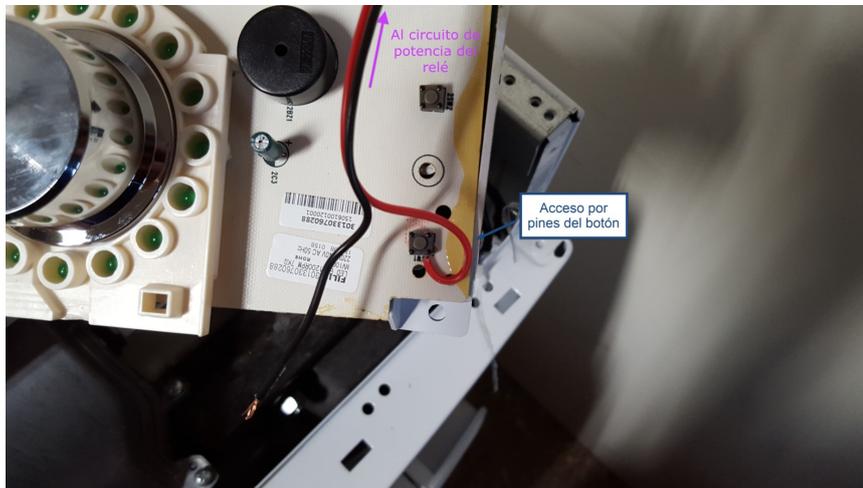


(a) Botón de pausa



(b) PCB barnizado

Apéndice L. Detalle Nodos de Carga



(c) Bornes del botón

Figura L.6: Modificaciones al circuito del lavasecarropas

L.4. Módulos de medición

En la sección anterior se describieron los módulos de comandos implementados para operar los electrodomésticos. La presente sección está enfocada en los módulos de medición implementados.

Los módulos de medición son los encargados de medir magnitudes de interés y reportarlas al Nodo Controlador. Dependiendo de los electrodomésticos controlados por el sistema de gestión pueden existir múltiples magnitudes de interés, por ejemplo, potencia eléctrica, temperatura, luminosidad, humedad del suelo, entre otras.

En el proyecto “Controlador de energía domiciliario para una Red Eléctrica Inteligente” se diseñó un módulo para la medición de corriente, voltaje y potencia consumida por las cargas[1]. Para conocer el detalle de funcionamiento y cómo fue implementado el mismo favor consultar la documentación de dicho proyecto. En este trabajo se implementaron dos módulos de medición adicionales. En primer lugar, un módulo que mide la temperatura ambiente en una habitación. En segundo lugar, un módulo para medir la temperatura de agua dentro de un calefón. Estos módulos permiten verificar el correcto funcionamiento del algoritmo y validar los modelos térmicos utilizados para el HVAC y el calefón.

Medición de temperatura ambiente

Este módulo es el encargado de medir la temperatura ambiente en una habitación. Dicho módulo se utilizó para validar el modelo térmico propuesto para el aire acondicionado. Sin embargo, este módulo no fue utilizado para la prueba experimental.

Sensor TMP102:

Para medir la temperatura ambiente se seleccionó el sensor de temperatura TMP102 del fabricante *Texas Instruments*, el cual viene incluido en el mote Z1. Las características más importantes del mismo son³:

³Texas Instruments. *TMP102 Low-Power Digital Temperature Sensor With SMBus and Two-Wire*

- Sensor digital de temperatura.
- Rango de medición: -25°C a 85°C .
- Precisión: $\pm 0,5^{\circ}\text{C}$.
- Interfaz: I2C y SMBus.

En la hoja de datos de los motes Z1⁴ se recomienda alimentar el mote Z1 con $3V_{DC}$ para utilizar el sensor TMP102. Esto se debe a que si el mote es alimentado con $5V_{DC}$ por el puerto microUSB, se utiliza un regulador de tensión y las disipaciones térmicas de este regulador pueden causar desviaciones en las medidas del sensor TMP102 de hasta $+2^{\circ}\text{C}$.

Recurso: `temperature_TMP102`

Para que el Nodo Controlador pueda acceder a las mediciones de temperatura se implementó un recurso llamado `temperature_TMP102`. En la implementación del mismo se usó como base el recurso `battery`. El recurso implementado permite el uso de los métodos GET y devuelve en el mensaje de la respuesta el valor de la temperatura medida multiplicada por 100.

El sistema operativo Contiki cuenta con un driver del sensor TMP102 para la plataforma Z1 que cuenta con las funciones necesarias para poder utilizar el sensor de forma sencilla. La funciones del driver utilizadas en este módulo fueron las siguientes:

- `tmp102_init()`: inicializa el sensor de temperatura. Es decir, configura el puertos y registros necesarios para la comunicación I2C con el sensor.
- `tmp102_read_temp_x100()`: devuelve la temperatura medida por el sensor multiplicada por cien.

Medición temperatura del agua en un calefón

Este módulo es el encargado de medir la temperatura del agua dentro del calefón. El mismo fue utilizado para validar el modelo térmico propuesto para el calefón⁵ y comprobar el correcto funcionamiento del algoritmo, verificando que la temperatura del agua se mantiene dentro del rango configurado por el usuario.

Se estudió cómo funcionan los termostatos en un calefón, más específicamente, se estudió cómo es que miden la temperatura del agua. Se encontró que en algunos termostatos simplemente se mide la temperatura de la pared del taque de cobre y se asume que la misma se encuentra en equilibrio térmico con el agua. Si bien existen otros termostatos más complejos, y probablemente más precisos, este método es sencillo y poco invasivo. Por ello, se decidió utilizar este método para medir la temperatura del agua.

Para medir la temperatura de la pared del tanque del calefón se seleccionó el sensor de temperatura y humedad SHT25 de *Sensiron*. Si bien este sensor también permite medir humedad, sólo se utilizó el sensor para medir temperatura. Las principales características del SHT25 como sensor de temperatura son las siguientes:

- Rango de medición: -40°C a 125°C .
- Precisión: $\pm 0,2^{\circ}\text{C}$.

Serial Interface in SOT563. <http://www.ti.com/lit/ds/symlink/tmp102.pdf>. Hoja de datos. Ago. de 2007.

⁴Zolertia. *Z1 Datasheet*. http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf. Hoja de datos. Mar. de 2010.

⁵Ver Apéndice B: Detalle del modelado de los electrodomésticos

Apéndice L. Detalle Nodos de Carga

- Interfaz: I2C.
- Alimentación $3V_{DC}$.

En este proyecto se utilizó el módulo MOD-1018 del fabricante *Embedded Adventures*. Tal como se puede ver en la Figura L.7, este módulo presenta un sensor SHT25 listo para usar, incluyendo las resistencias de *pull-up* necesarias para la comunicación I2C. Para obtener mayor detalle respecto al sensor SHT25 y el módulo MOD-1018, consultar sus respectivas hojas de datos⁶⁷.

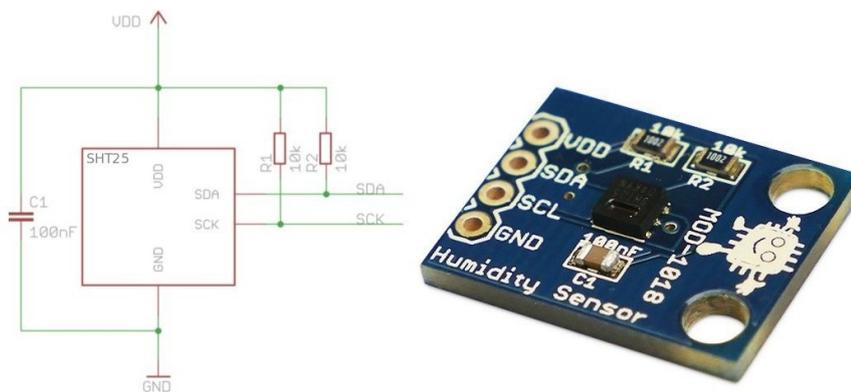


Figura L.7: Relé de estado sólido utilizado

Recurso: `temperature_SHT25`

Para que el Nodo Controlador pueda acceder a las mediciones de temperatura se implementó un recurso REST llamado `temperature_SHT25`.

El sistema operativo Contiki también cuenta con un driver del sensor SHT25. Su implementación se encuentra en: `'contiki/platform/z1/dev/sht25.c`. Las funciones del driver utilizadas en el módulo de medición fueron las siguientes:

- `SENSORS_ACTIVATE(sht25)`: inicializa el sensor de temperatura, configura los pines y registros necesarios para la comunicación I2C.
- `sht25.value(magnitud)` devuelve la medida del sensor multiplicada por cien. Dependiendo del parámetro `magnitud` cuál es la medida reportada:
 - `magnitud = SHT25_VAL_TEMP`: devuelve la medida de temperatura.
 - `magnitud = SHT25_VAL_HUM`: devuelve la medida de humedad.

Implementación en hardware:

Conexión con el mote:

El módulo MOD-1018 cuenta con 4 pines de conexión, ver Figura L.7 y Tabla L.1. Estos son los pines típicos de cualquier dispositivo con interfaz I2C. Los mismos fueron conectados a

⁶Sension. *Datasheet SHT25 - Humidity and Temperature Sensor*. http://www.soselectronic.com/a_info/resource/c/SHT25.pdf. Hoja de datos. Dic. de 2010.

⁷Embedded Adventures. *MOD - 1018 - datasheet*. http://www.embedded-adventures.com/datasheets/MOD-1018_hw_v1_doc_v1.pdf. Hoja de datos. 3 de abr. de 2013.

L.4. Módulos de medición

los pines de comunicación I2C del mote Z1. O sea, a los pines 22, 24, 26 y 28 del puerto JP1B del mote.

VDD	Alimentación (2.1 V a 3.6V)
SDA	I2C Datos
SCL	I2C Reloj
GND	Tierra

Tabla L.1: Pines del módulo MOD-1018

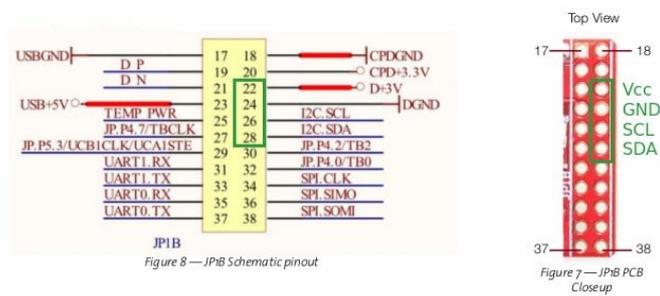


Figura L.8: Pines para comunicación I2C en Z1

Instalación y verificación del correcto funcionamiento

Para medir la temperatura del agua del calefón se colocó el sensor SHT25 en contacto con el tanque de cobre y se usó goma térmica, modelo ‘ARTIC Thermal Pad’, para asegurar un buen contacto térmico con el mismo.

Se identificaron dos ubicaciones posibles para colocar el sensor, la primera junto al panel del termostato y la segunda en la parte superior del tanque, ver Figura L.9. Se realizaron pruebas experimentales con el fin de determinar la mejor ubicación.

La prueba experimental consistió en comparar los valores medidos por el sensor y las medidas de un termómetro de laboratorio. Se evaluó el desempeño del sensor en un amplio rango de temperaturas y en los dos estados posibles del calefón, encendido y apagado. Para ello se comenzó el experimento con agua fría dentro del tanque, luego se encendió el calefón por un período superior a una hora y finalmente se apagó el mismo y se dejó enfriar el agua del tanque. Durante el experimento, se registraron los valores medidos por el sensor de temperatura cada treinta segundo y cada 15 minutos se extrajo agua del tanque y se midió con el termómetro de laboratorio.

Los resultados de la prueba experimental se muestran gráficamente en la Figura L.10, en Figura L.10a se encuentran los resultados con el sensor ubicado en el recinto del termostato y en Figura L.10b con el ubicado en la parte superior del tanque. Se observó que las medidas con el sensor en la parte superior del tanque se ajustan de mejor manera a los valores medidos por el termómetro. En este caso la diferencia máxima entre medidas fue de 1,5 °C, valor que se encuentra dentro de la incertidumbre de los instrumentos de medida. La curva de medidas en la tapa superior del tanque (Figura L.10b) es más suave que la del recinto del termostato (Figura L.10a), esto se debe a que en la parte superior los transitorios de extracción de agua y del calentamiento de la resistencia se filtran por la inercia térmica del agua.

Apéndice L. Detalle Nodos de Carga

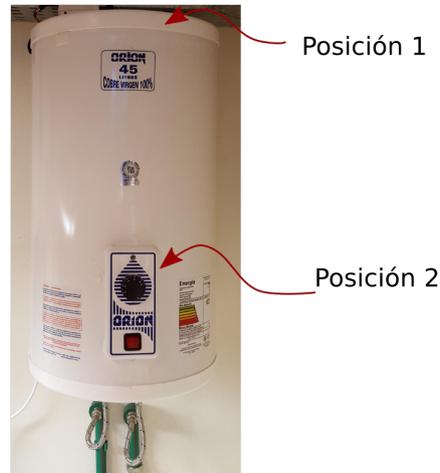


Figura L.9: Posibles ubicaciones para colocar el sensor

El usuario no requiere de precisión en la medida de temperatura ya que la temperatura del baño se ajusta mezclando agua fría con caliente. Además, este está acostumbrado a utilizar la perilla del termostato para estimar la temperatura del agua, la cual según lo observado durante el experimento puede presentar un error superior a 10 °C. Por esta razón no se requiere contar con una medida de precisión, sino que se considera que un error de ± 2 o 3 °C es aceptable. De acuerdo a los resultados obtenidos con los experimentos prueba, se considera que la precisión del sensor es adecuada para el uso en este proyecto.

Programa principal

A continuación se describen los aspectos más importantes de la implementación del programa principal de los Nodos de Carga, incluyendo la inicialización del programa y el procedimiento de recuperación ante las fallas de comunicación.

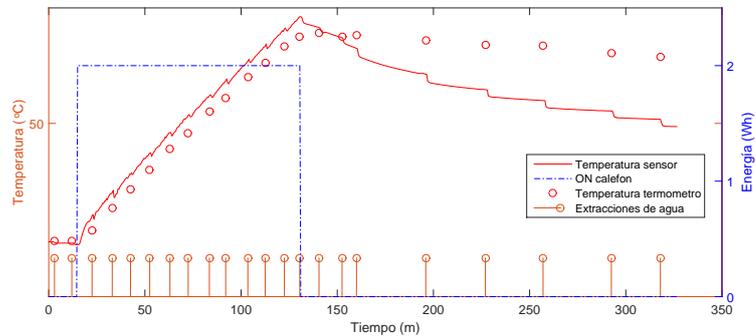
Inicialización del programa

Lo primero que debe hacer el programa del Nodo de Carga es inicializar el servidor REST y sus recursos, así como los sensores y/o actuadores que utilice. A modo de ejemplo, se muestra a continuación cómo es la inicialización de un Nodo de Carga que presenta un módulo de sensor de temperatura ambiente.

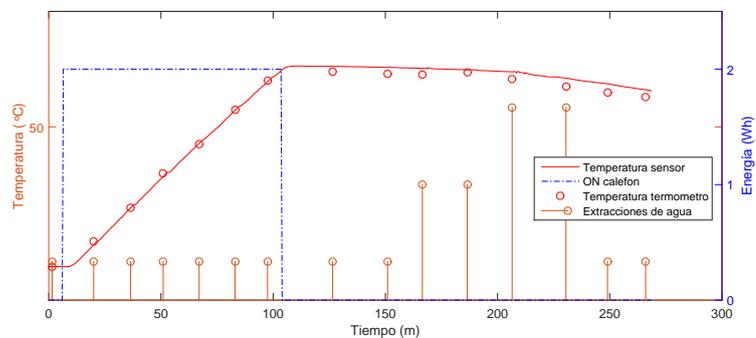
```
/* Inicialización del REST engine.*/  
rest_init_engine();  
  
/* Inicialización del sensor y el recurso de temperatura */  
tmp102_init();  
rest_activate_resource(&resource_temperature_TMP102)
```

Para activar un recurso este debe haber sido declarado previamente. Luego de estas instrucciones, el recurso queda activo y disponibles para cualquier cliente en la HAN.

L.4. Módulos de medición



(a) Panel del termostato



(b) Pared superior del tanque

Figura L.10: Prueba del sensor de temperatura del calefón en distintas posiciones

Fallas en la comunicación

En las pruebas realizadas con el relé electromagnético se observó que el mote Z1 es sensible a ruidos del entorno. Si bien al cambiar el tipo de relé utilizado para conmutar las cargas se eliminó la principal fuente de ruido, es muy probable que existan otras fuentes en un hogar que hagan que falle la comunicación entre el Nodo Controlador y los Nodos de Carga. Por esta razón fue necesario implementar un procedimiento que permita a los nodos recomponerse ante estas fallas.

El procedimiento implementado consiste en verificar de forma periódica que el canal de comunicación funciona correctamente. Para esto el Nodo Controlador debe enviar mensajes periódicamente a cada uno de los Nodos de Carga. Si el Nodo de Carga no recibe un mensaje desde el Nodo Controlador por un determinado tiempo, esto indica que perdió la comunicación con el Nodo Controlador y en consecuencia el mote Z1 debe resetearse. Esto se implementó en los Nodos de Carga mediante un timer tipo *watchdog*. Este timer se inicia cuando inicializa el mote y resetea su cuenta cada vez que recibe un mensaje desde el Nodo Controlador. Si el timer expira se procede a resetear el microprocesador del mote.

Apéndice M

Protocolos de las pruebas experimentales



Protocolo de prueba experimental
del nodo controlador
EnerHomGy

PROTOCOLO DE LA PRUEBA EXPERIMENTAL

FECHA: 20/03/2016
SITIO: Facultad de Ingeniería de la UdelaR
SECTOR: Instituto de Ingeniería Eléctrica
SALÓN: Lab. de proyectos de fin de carrera

Datos iniciales		
Magnitud	Hora	Valor
Temperatura ambiente	7:16	21.60 °C
Temperatura del agua fría	7:22	22.17 °C
Temperatura del agua en el calefón	7:22	60.49 °C (registro s/extracción)
Programación lavasecarropa	7:10	Restan 2:32 para la finalización del lavado

Calefón		
Magnitud	Hora	Valor
Primer pulso útil (6:30 – 8:30)		
Estado	7:30	OFF
Temperatura del agua fría	7:22	Esperado: 26.1°C Real: 22.17°C
Temperatura del agua en el interior del calefón	7:26	Esperado: 61.5°C Real: 60.49°C
Extracción de agua (Tbaño = 39.5 °C)	7:40	Se extraen 25.3 litros de agua Tinicial 59.2 °C –Tfinal 58.1 °C
Temperatura del agua en el interior del calefón	7:50	Interior: 58.8°C Extracción: 58.2°C
¿Temperatura aceptable al final de la ducha?	7:50	SI

Apéndice M. Protocolos de las pruebas experimentales



Protocolo de prueba experimental
del nodo controlador
EnerHomGy

Calefón			
Segundo pulso útil (16:00 – 19:30)			
Estado	18:40	OFF	
Temperatura del agua fría	18:42	Esperado: 26.1°C	Real: 23.0°C
Temperatura del agua en el interior del calefón	18:42	Esperado: 42.4°C	Real: 41.65°C
Extracción de agua (Tbaño = 39.5 °C)	19:00	Se extraen 45 litros de agua. Tinicial 41°C – Tfinal 32.4 °C	
Temperatura del agua en el interior del calefón	19:13	Interior: 26.88°C	Extracción: 26.0°C
¿Temperatura aceptable al final de la ducha?	19:13	SI	

Lavasecarropa		
Único pulso útil (7:00 – 20:00)		
Estado	7:00	OFF
Comienzo del ciclo	7:16	Inicio normal, en el tiempo esperado.
INTERRUPCIÓN 1	NA	No hubo interrupciones durante las 2:32 horas de funcionamiento.
Ciclo completado	19:14	Lavado completado de acuerdo a lo esperado.

Protocolo de prueba experimental – Hoja N° 2



PROTOCOLO DE LA PRUEBA EXPERIMENTAL

FECHA: 21/03/2016
SITIO: Facultad de Ingeniería de la UdelaR
SECTOR: Instituto de Ingeniería Eléctrica
SALÓN: Lab. de proyectos de fin de carrera

Datos iniciales		
Magnitud	Hora	Valor
Temperatura ambiente	7:42	22.1°C
Temperatura del agua fría	7:42	22.6°C
Temperatura del agua en el calefón	7:42	60.08 °C (registro s/extracción)
Programación lavasecarropa	7:10	Restan 2:32 para la finalización del lavado

Calefón			
Magnitud	Hora	Valor	
Único pulso útil (16-19:30)			
Estado	7:45	OFF	
Temperatura del agua fría	7:45	Esperado: 26.1°C	Real: 22.6 °C
Temperatura del agua en el interior del calefón	7:45	Esperado: 61.1°C	Real: 60.1 °C
Temperatura del agua en el interior del calefón	19:39	Esperado: 51.5°C	Real: 50.3 °C
Extracción de agua (Tbaño = 39.5 °C)	19:44	Se extraen 35 litros de agua. Tinicial 49.4°C –Tfinal 46.7°C	
Temperatura del agua en el interior del calefón	19:44	Interior: 48.0°C	Extracción: 49.4°C
¿Temperatura aceptable al final de la ducha?	19:44	SI	

Apéndice M. Protocolos de las pruebas experimentales



Protocolo de prueba experimental
del nodo controlador
EnerHomGy

Lavasecarropa		
Único pulso útil (7:00 – 20:00)		
Estado	7:15	OFF
Comienzo del ciclo	7:20	Lavado iniciado
INTERRUPCIÓN 1	9:00	Falla la interrupción. Se constata que el lavasecarropa no es interrumpible en su ciclo de secado.
Ciclo completado	SI	Ropa lavada correctamente, pero antes de lo esperado.

Apéndice N

Gestión del proyecto

En este apéndice se describe algunos aspectos de la gestión de proyecto. En particular se muestra cómo fue la dedicación horaria del grupo a lo largo del período de trabajo y el peso de cada una de las etapas.

Dedicación total

- Dedicación grupal real: 3124hs.
- Dedicación grupal planificada: 1800hs.

Se observa que el grupo tuvo una dedicación horaria mucho mayor a la planificada inicialmente. Se entiende que esto se debió a falta de experiencia en planificación de proyectos, lo cual llevó a una importante sub-estimación de las horas hombre necesarias para ejecutar cada una de las tareas. A raíz de esta sub-estimación y algunos imprevistos fue necesario solicitar un mes de prórroga para la entrega del proyecto.

Dedicación semanal

En la Figura N.1 se encuentra la evolución de la dedicación semanal del grupo a lo largo del proyecto. Se destaca que en los últimos meses fue necesario incrementar la dedicación para poder cumplir con los plazos.

Dedicación por tarea

Se incluyen los tiempos dedicados para cada una de las tareas que se llevó a cabo en el proyecto. En la Figura N.2a se muestra la dedicación horas dedicadas por tarea mientras que en la Figura N.2b se muestra el peso relativo de cada tarea.

Apéndice N. Gestión del proyecto

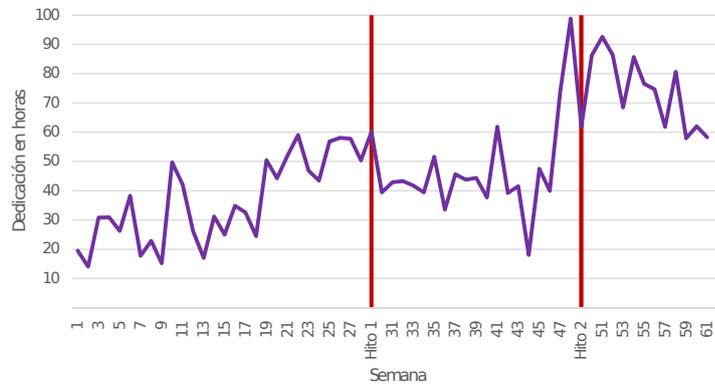
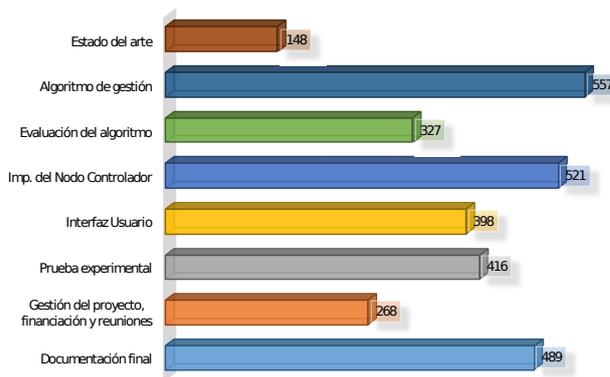
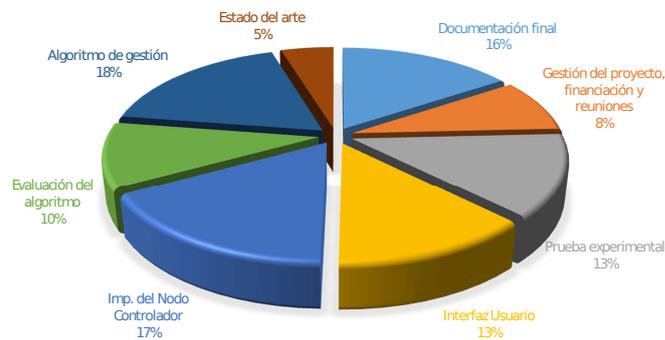


Figura N.1: Dedicación semanal del grupo



(a) Horas por tarea



(b) Peso relativo de cada tarea

Figura N.2: Dedicación de horas hombre discriminada por tema

Bibliografía

- [1] G. Belcredi, Pablo Modernell y Nicolas Sosa. “Controlador de energía domiciliario para una Red Eléctrica Inteligente”. Proyecto Fin de Carrera de Ingeniería Eléctrica. Uruguay: Facultad de Ingeniería, Universidad de la República, 2015.
- [2] *UTE en cifras*. Ver. 2014. Administración Nacional de Usinas y Trasmisiones Eléctricas. URL: <http://portal.ute.com.uy/sites/default/files/documents/files/Ute%20en%20cifras%202014.pdf> (visitado 25-12-2015).
- [3] *Energía Vendida por categoría tarifaria*. Ver. 2. Administración Nacional de Usinas y Trasmisiones Eléctricas. URL: http://portal.ute.com.uy/sites/default/files/documents/files/ENERG%C3%8DA%20VENDIDA%20POR%20CATEGOR%C3%8DA%20TARIFARIA_2.ods (visitado 25-12-2015).
- [4] D. Abbott. “Keeping the Energy Debate Clean: How Do We Supply the World’s Energy Needs?” En: *Proceedings of the IEEE* 98.1 (ene. de 2010), págs. 42-66. ISSN: 0018-9219. DOI: 10.1109/JPROC.2009.2035162.
- [5] *Balance energético nacional 2014*. Ministerio de industria, energía y minería, 2014. URL: <http://www.miem.gub.uy/documents/15386/6815916/1.1INFORMEGENERALBEN2014.pdf> (visitado 25-12-2015).
- [6] *Consulta geográfica de fuentes de generación*. URL: <http://portal.ute.com.uy/consultageograficadefuentesdegeneracion> (visitado 25-11-2015).
- [7] “IEEE Application Guide for IEEE Std 1547(TM), IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems”. En: *IEEE Std 1547.2-2008* (abr. de 2009), págs. 1-217. DOI: 10.1109/IEEESTD.2008.4816078.
- [8] Renato Céspedes y Guillermo Jiménez Estévez. “Implementación de Smart-Grids: retos y oportunidades”. Tutorial Introductorio a la Conferencia de Smart Grids ISGT LA 2015. Palacio de la Luz, Montevideo, Uruguay, 2 de oct. de 2015.
- [9] S. E. Collier. “The Emerging Enernet: Convergence of the Smart Grid with the Internet of Things”. En: *Rural Electric Power Conference (REPC), 2015 IEEE*. Abr. de 2015, págs. 65-68.
- [10] J. S. Vardakas, N. Zorba y C. V. Verikoukis. “A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms”. En: *IEEE Communications Surveys Tutorials* 17.1 (2015), págs. 152-178. ISSN: 1553-877X. DOI: 10.1109/COMST.2014.2341586.

Bibliografía

- [11] Lijun Chen y col. “Control and Optimization Methods for Electric Smart Grids”. En: ed. por Aranya Chakraborty y D. Marija Ilić. New York, NY: Springer New York, 2012. Cap. Optimal Demand Response: Problem Formulation and Deterministic Case, págs. 63-85. ISBN: 978-1-4614-1605-0. DOI: 10.1007/978-1-4614-1605-0_3. URL: http://dx.doi.org/10.1007/978-1-4614-1605-0_3.
- [12] Steven Braithwait, Dan Hansen y Michael O’Sheasy. *Retail Electricity Pricing And Rate Design In Evolving Markets*. Edison Electric Institute, jul. de 2007.
- [13] L. P. Qian y col. “Demand Response Management via Real-Time Electricity Price Control in Smart Grids”. En: *IEEE Journal on Selected Areas in Communications* 31.7 (jul. de 2013), págs. 1268-1280. ISSN: 0733-8716. DOI: 10.1109/JSAC.2013.130710.
- [14] A. H. Mohsenian-Rad y A. Leon-Garcia. “Optimal Residential Load Control With Price Prediction in Real-Time Electricity Pricing Environments”. En: *IEEE Transactions on Smart Grid* 1.2 (sep. de 2010), págs. 120-133. ISSN: 1949-3053. DOI: 10.1109/TSG.2010.2055903.
- [15] Ahmad Faruqui y Stephen George. “Quantifying Customer Response to Dynamic Pricing”. En: *The Electricity Journal* 18.4 (2005), págs. 53 -63. DOI: <http://dx.doi.org/10.1016/j.tej.2005.04.005>.
- [16] Marc Beaudin y Hamidreza Zareipour. “Home energy management systems: A review of modelling and complexity”. En: *Renewable and Sustainable Energy Reviews* 45 (2015), págs. 318 -335. ISSN: 1364-0321. DOI: 10.1016/j.rser.2015.01.046.
- [17] M. Beaudin y col. “Residential Energy Management Using a Two-Horizon Algorithm”. En: *IEEE Transactions on Smart Grid* 5.4 (jul. de 2014), págs. 1712-1723. ISSN: 1949-3053. DOI: 10.1109/TSG.2014.2310395.
- [18] N. Li, L. Chen y S. H. Low. “Optimal demand response based on utility maximization in power networks”. En: *Power and Energy Society General Meeting, 2011 IEEE*. Jul. de 2011, págs. 1-8. DOI: 10.1109/PES.2011.6039082.
- [19] A. J. Conejo, J. M. Morales y L. Baringo. “Real-Time Demand Response Model”. En: *IEEE Transactions on Smart Grid* 1.3 (dic. de 2010), págs. 236-242. ISSN: 1949-3053. DOI: 10.1109/TSG.2010.2078843.
- [20] Z. Chen, L. Wu e Y. Fu. “Real-Time Price-Based Demand Response Management for Residential Appliances via Stochastic Optimization and Robust Optimization”. En: *IEEE Transactions on Smart Grid* 3.4 (dic. de 2012), págs. 1822-1831. ISSN: 1949-3053. DOI: 10.1109/TSG.2012.2212729.
- [21] T. T. Kim y H. V. Poor. “Scheduling Power Consumption With Price Uncertainty”. En: *IEEE Transactions on Smart Grid* 2.3 (sep. de 2011), págs. 519-527. ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2159279.

- [22] P. Du y N. Lu. “Appliance Commitment for Household Load Scheduling”. En: *IEEE Transactions on Smart Grid* 2.2 (jun. de 2011), págs. 411-419. ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2140344.
- [23] S. Li y col. “Integrating Home Energy Simulation and Dynamic Electricity Price for Demand Response Study”. En: *IEEE Transactions on Smart Grid* 5.2 (mar. de 2014), págs. 779-788. ISSN: 1949-3053. DOI: 10.1109/TSG.2013.2279110.
- [24] Zolertia. *Z1 Datasheet*. http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf. Hoja de datos. Mar. de 2010.
- [25] Texas Instruments. *MSP430x2xx Family User’s Guide*. <http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=slau144&fileType=pdf>. Manual de usuario. Jul. de 2013.
- [26] Raspberry Pi. *Raspberry Pi 2, Model B*. <https://www.adafruit.com/pdfs/raspberrypi2modelb.pdf>. Hoja de datos. 3 de abr. de 2013.
- [27] *Welcome To Raspbian*. URL: <https://www.raspbian.org/> (visitado 09-04-2016).
- [28] Z. Shelby, K. Hartke y C. Bormann. *The Constrained Application Protocol (CoAP)*. RFC 7252. RFC Editor, jul. de 2014. URL: <http://www.rfc-editor.org/rfc/rfc7252.txt>.
- [29] *El Plan Solar de Uruguay*. Ministerio de industria, energía y minería, abr. de 2012. URL: <http://www.dne.gub.uy/documents/112315/133193/Impo%2004%202012%20-%20El%20Plan%20Solar%20de%20Uruguay.pdf> (visitado 25-12-2015).
- [30] *Uruguay en Cifras 2014*. Uruguay: Instituto Nacional de Estadísticas, oct. de 2014.
- [31] R. Diao y col. “Electric water heater modeling and control strategies for demand response”. En: *Power and Energy Society General Meeting, 2012 IEEE*. Jul. de 2012, págs. 1-8. DOI: 10.1109/PESGM.2012.6345632.
- [32] Python Software Foundation. *Python*. URL: <https://www.python.org/> (visitado 15-04-2016).
- [33] Python Software Foundation. *History and License*. URL: <https://docs.python.org/2/license.html> (visitado 15-04-2016).
- [34] *MySQL Reference Manual*. URL: <http://dev.mysql.com/doc/refman/5.7/en/> (visitado 09-04-2016).
- [35] *A weather API designed for developers*. URL: <https://www.wunderground.com/weather/api> (visitado 01-05-2016).
- [36] Stephen Boyd y Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN: 0521833787.
- [37] Martin Andersen, Joachim Dahl y Lieven Vandenberghe. *CVXOPT User’s Guide*. 22 de sep. de 2015. URL: <http://cvxopt.org/userguide/index.html>.

Bibliografía

- [38] James Gosling y Henry McGilton. *The Java Language Environment A Withe Paper*. URL: <http://www.oracle.com/technetwork/java/langenv-140151.html> (visitado 09-04-2016).
- [39] *JavaServer Pages Technology*. URL: <http://www.oracle.com/technetwork/java/javasee/jsp/index.html> (visitado 09-04-2016).
- [40] *JavaServer Pages Overview*. URL: <http://www.oracle.com/technetwork/java/overview-138580.html> (visitado 09-04-2016).
- [41] *Apache Tomcat*. URL: <http://tomcat.apache.org/> (visitado 16-05-2016).
- [42] *Apache Maven Proyect*. URL: <https://maven.apache.org/> (visitado 16-05-2016).
- [43] *Hibernate ORM*. URL: <http://hibernate.org/orm/> (visitado 16-05-2016).
- [44] *Welcome To JFreeChart!* URL: <http://www.jfree.org/jfreechart/> (visitado 09-04-2016).
- [45] *Class DataSourceRealm*. URL: <https://tomcat.apache.org/tomcat-7.0-doc/api/org/apache/catalina/realm/DataSourceRealm.html> (visitado 11-05-2016).
- [46] *Pliego tarifario vigente desde el 13/01/2016*. Ver. 1. Administración Nacional de Usinas y Trasmisiones Eléctricas. URL: http://www.ute.com.uy/sispubnoticias/pubadjuntos/21637_Pliego%20Tarifario%20enero%202016.pdf (visitado 01-03-2016).
- [47] *Instalaciones de Energía Solar Térmica Pliego de Condiciones Técnicas (PCT)*. Uruguay: Ministerio de Industria, Energía y Minería, dic. de 2010. URL: http://www.correo.com.uy/otrosdocumentos/pdf/comprasyproveedores/ETUS_2014.pdf (visitado 28-03-2016).
- [48] *Cotización histórica del dólar*. Ver. 1. URL: <http://www.bcu.gub.uy/Estadisticas-e-Indicadores/Paginas/Cotizaciones.aspx> (visitado 28-04-2016).
- [49] D. Acuna, M. Darré y H. Mourglia. *Tarifa en tiempo real para incentivar el uso eficiente de la red eléctrica uruguaya*. Inf. téc. Facultad de Ingeniería de la Universidad de la República, jun. de 2015.
- [50] M. Pipattanasomporn y col. "Load Profiles of Selected Major Household Appliances and Their Demand Response Opportunities". En: *IEEE Transactions on Smart Grid* 5.2 (mar. de 2014), págs. 742-750. ISSN: 1949-3053. DOI: 10.1109/TSG.2013.2268664.
- [51] A. Pahwa y C. W. Brice. "Modeling and System Identification of Residential Air Conditioning Load". En: *IEEE Transactions on Power Apparatus and Systems* PAS-104.6 (jun. de 1985), págs. 1418-1425. ISSN: 0018-9510. DOI: 10.1109/TPAS.1985.319247.
- [52] Refrigerating American Society of Heating y Air-Conditioning Engineers. *2001 ASHRAE Handbook: Fundamentals*. ASHRAE HANDBOOK FUNDAMENTALS SYSTEMS-INTERNATIONAL METRIC SYSTEM. ASHRAE, 2001. ISBN: 9781883413880.

- [53] M. Muratori y col. “Energy consumption of residential HVAC systems: A simple physically-based model”. En: *Power and Energy Society General Meeting, 2012 IEEE*. Jul. de 2012, págs. 1-8. DOI: 10.1109/PESGM.2012.6344950.
- [54] P. Constantopoulos, F.C. Schweppe y R.C. Larson. “Estia: A real-time consumer control scheme for space conditioning usage under spot electricity pricing”. En: *Computers & Operations Research* 18.8 (1991), págs. 751 -765. ISSN: 0305-0548. DOI: 10.1016/0305-0548(91)90013-H.
- [55] Y. Wen y W. Burke. “Real-Time Dynamic House Thermal Model Identification for Predicting HVAC Energy Consumption”. En: *Green Technologies Conference, 2013 IEEE*. Abr. de 2013, págs. 367-372. DOI: 10.1109/GreenTech.2013.63.
- [56] M. H. Nehrir y col. “Development and validation of a physically-based computer model for predicting winter electric heating loads”. En: *IEEE Transactions on Power Systems* 10.1 (feb. de 1995), págs. 266-272. ISSN: 0885-8950. DOI: 10.1109/59.373949.
- [57] V. Bradshaw. *The Building Environment: Active and Passive Control Systems*. Wiley, 2006. ISBN: 9780471689652.
- [58] Joakim Widén y col. “Constructing load profiles for household electricity and hot water from time-use data—Modelling approach and validation”. En: *Energy and Buildings* 41.7 (2009), págs. 753 -768. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2009.02.013.
- [59] R. González. *Python para todos*. 2011. URL: http://www.ceibal.edu.uy/contenidos/areas_conocimiento/aportes/python_para_todos.pdf.
- [60] *Threading — Thread-based parallelism — Python 3.4.4 documentation*. Python Software Foundation. 2015. URL: <https://docs.python.org/3.4/library/threading.html> (visitado 22-09-2015).
- [61] *asyncio – Asynchronous I/O, event loop, coroutines and tasks — Python 3.4.4 documentation*. Python Software Foundation. 2015. URL: <https://docs.python.org/3.4/library/asyncio.html> (visitado 02-02-2016).
- [62] *Queue — A synchronized queue class — Python 3.4.4 documentation*. Python Software Foundation. 2015. URL: <https://docs.python.org/3.4/library/queue.html> (visitado 04-02-2016).
- [63] Phidgets. *3051_1 - Dual Relay Board*. http://www.phidgets.com/products.php?product_id=3051. Especificaciones del producto.
- [64] Crydom. *Solid Statements*. Inf. téc. Documento técnico. Ago. de 2011.
- [65] Panasonic. *SSRs Technical Information*. Inf. téc. Documento técnico. Ago. de 2011. URL: https://www.panasonic-electric-works.com/pew/fr/downloads/ds_x61_en_ssr_technical_information.pdf.
- [66] Fotek. *SSR series*. Hoja de datos. URL: <https://cdn.sparkfun.com/datasheets/Components/General/SSR40DA.pdf>.

Bibliografía

- [67] Texas Instruments. *TMP102 Low-Power Digital Temperature Sensor With SMBus and Two-Wire Serial Interface in SOT563*. <http://www.ti.com/lit/ds/symlink/tmp102.pdf>. Hoja de datos. Ago. de 2007.
- [68] Sension. *Datasheet SHT25 - Humidity and Temperature Sensor*. http://www.soselectronic.com/a_info/resource/c/SHT25.pdf. Hoja de datos. Dic. de 2010.
- [69] Embedded Adventures. *MOD - 1018 - datasheet*. http://www.embedded-adventures.com/datasheets/MOD-1018_hw_v1_doc_v1.pdf. Hoja de datos. 3 de abr. de 2013.

Glosario

API (Application Programming Interface)

Conjunto de subrutinas, funciones y procedimientos que brinda cierta biblioteca para ser utilizado por otro software.

Calefón

El calefón, o termotanque, es un dispositivo eléctrico que eleva la temperatura del agua corriente para su uso doméstico. Posee un tanque donde se acumula el agua, la cual se calienta hasta alcanzar una temperatura controlada por un termostato.

CoAP (Constrained Application Protocol)

Protocolo de capa de aplicación similar a *HTTP* desarrollado para dispositivos y redes de recursos limitados[28].

DAP (Day-Ahead Pricing)

Sistema de tarifario de energía en el cual el precio de la energía varía hora a hora. Los precios de cada hora se fijan con un día de anticipación.

DR (Demand Response)

Respuesta a la Demanda, o en inglés *Demand Response*, refiere a las políticas que buscan modificar la demanda de energía eléctrica con el objetivo de realizar un uso más eficiente de los recursos disponibles.

Empresa suministradora

Es la empresa encargada de proveer y comercializar el servicio de energía eléctrica. En Uruguay esta empresa es la Administración Nacional de Usinas y Trasmisiones Eléctricas (UTE). En el marco de este proyecto también se asume que esta cumple el rol de fijar la curva de precios de la energía.

GNU GPL (GNU General Public License)

Licencia de software ampliamente usada que garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

GNU LGPL (GNU Lesser General Public License)

Licencia de software creada por la Free Software Foundation que pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios. A diferencia de GPL, permite que los desarrolladores utilicen programas bajo estas licencias sin estar obligados a someter el programa final a las mismas.

HAN (Home Area Network)

Es un tipo de red de área local (LAN) que se desarrolla a partir de la necesidad de facilitar la

Apéndice N. Glosario

comunicación y la interoperabilidad entre los dispositivos digitales presentes en el interior de una casa.

HEMS (Home Energy Management System)

Sistema capaz de cambiar y moldear la demanda energética de un hogar de acuerdo al precio de la energía y el confort de los usuarios. Este presenta un canal de comunicación a través del cual recibe información externa (e.g. precio de la energía y predicción del clima) y es capaz de comandar a los electrodomésticos del hogar. En el presente trabajo se considera que el sistema HEMS engloba al Nodo Controlador y los Nodos de Cargas que operan los electrodomésticos.

En otros textos también se lo puede encontrar como: “Residential Energy Management System” (REMS), “Energy Management System” (EMS) y “Load manager household”, entre otros [16].

HTTP (Hypertext Transfer Protocol)

Protocolo de comunicación que permite las transferencias de información en la Web.

HVAC (Heating, ventilating, and air conditioning)

Sistema de refrigeración del aire que se utiliza de modo doméstico para enfriar, ventilar o calentar el ambiente. Las tres funciones centrales de calefacción, ventilación y aire acondicionado están relacionados entre sí, especialmente con la necesidad de proporcionar confort térmico y la calidad del aire interior aceptable.

JDBC (Java Database Connectivity)

API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede.

JNDI (Java Naming and Directory Interface)

API de Java para servicios de directorio que permite a los clientes buscar objetos y datos a través de un nombre.

JSP (JavaServer Pages)

Tecnología para crear páginas web dinámicas basadas en HTML, XML.

LAN (Local Area Network)

Una red de área local o LAN es una red de computadoras que abarca un área reducida a una casa, un departamento o un edificio.

Precio SPOT

El precio SPOT es el costo marginal de generación horario que calcula diariamente la Administración del Mercado Eléctrico, para determinar cuál es el despacho económico de las centrales generadoras. Este precio idealmente representa el costo del agente generador más caro utilizado para abastecer la demanda del momento.

REST (REpresentational State Transfer)

Estilo de arquitectura para desarrollar servicios que define un conjunto de principios arquitectónicos haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren. El protocolo más usado que cumple esta definición, es el protocolo HTTP.

RTP (Real-Time Pricing)

Sistema tarifario de energía en el cual el precio de la energía varía hora a hora (o intervalos más pequeños). Al inicio de cada hora se fija el precio en función de la demanda proyectada y de la generación.

Security Realm

Conjunto de mecanismos que sirven para proteger los recursos que se encuentran dentro del servidor de aplicaciones Web.

Servlet

Clase Java, utilizada para ampliar las capacidades de un servidor.

SLS (Secure Sockets Layer)

Protocolo de seguridad diseñado para transmitir datos de manera íntegra y segura.

Smart Grid

Conjunto de políticas que buscan dotar de inteligencia la red eléctrica, coordinando el comportamiento y las acciones de todos los actores e integrando los desarrollos en ingeniería eléctrica con los avances en tecnología de la información y comunicación.

URI (Uniform Resource Identifier)

Cadena de caracteres que identifica los recursos de una red de forma unívoca.

WAR (Web Application Archive)

Archivo utilizado para distribuir una colección de JSP, Servlets, clases Java, archivos XML, librerías de tags y páginas web estáticas (HTML y archivos relacionados) que juntos constituyen una aplicación web.

XML (eXtensible Markup Language)

Metalinguaje que permite la organización y el etiquetado de documentos. XML no es un lenguaje en sí mismo, sino un sistema que permite definir lenguajes de acuerdo a las necesidades.