

# **CORD**

## **Control Operativo Remoto de Dispositivos**

### **Documentación Técnica**

Federico Amchite  
Fernando Rodríguez  
Daniel Ruiz

Proyecto de Grado – Ingeniería Eléctrica  
Facultad de Ingeniería - UdelaR  
Tutor : Rafael Sotelo

Marzo 2006

## INDICE DE CONTENIDO

1 - INTRODUCCION.....	3
1.1 - ¿QUE ES LA DOMOTICA ?.....	3
1.2 - CARACTERISTICAS DE LAS VIVIENDAS INTELIGENTES.....	4
1.3 - GESTION DE LA DOMOTICA.....	4
1.4 - CARACTERISTICAS DE LOS SISTEMAS DOMOTICOS.....	5
2 - OBJETIVO.....	6
2.1 - C.O.R.D. (Control Operativo Remoto de Dispositivos).....	6
3 - DESCRIPCION DEL SISTEMA.....	8
3.1 - DESCRIPCION GENERAL.....	8
3.1.1 - DESCRIPCION DE FUNCIONAMIENTO.....	8
3.1.2 - DESCRIPCION DE COMPONENTES.....	9
3.2 - HARDWARE.....	10
3.2.1 - ESQUEMA GENERAL.....	10
3.2.2 - FUENTE DE ALIMENTACION.....	11
3.2.3 - TINI.....	12
3.2.4 - DS1820 ( Termómetro digital 1 – Wire ) .....	13
3.2.4.1 - COMUNICACION TINI - 1-WIRE .....	14
3.2.5 - PIC.....	15
3.2.5.1 - COMUNICACION TINI - PIC.....	15
3.2.6 - CIRCUITOS AUXILIARES.....	16
3.2.7 - CONEXIONADO INTERNO DEL SISTEMA.....	17
3.3 - SOFTWARE.....	20
3.3.1 - ESQUEMA Y DIAGRAMA DE FLUJOS.....	20
3.3.2 - HERRAMIENTAS Y DESARROLLO DE SOFTWARE.....	21
3.3.2.1 - Java .....	21
3.3.2.2 - TINI SDK.....	21
3.3.2.3 - MPLAB.....	22
3.3.2.4 - NOKIA INTERNET KIT.....	23
3.3.2.5 - NAVEGADORES WAP – EMULADORES DE TELEFONOS CELULARES .....	23
3.3.3 - TINI.....	24
3.3.3.1 - DEMONIOS.....	24
3.3.3.2 - CLASES AUXILIARES.....	25
3.3.4 - APPLET.....	27
3.3.4.1 - CLASES AUXILIARES.....	28
3.3.5 - COMUNICACION TINI - APPLET.....	30
3.3.6 - CODIGO WML PARA WAP.....	33
3.3.7 - PROGRAMA PIC.....	35
3.3.8 - ARCHIVOS DE CONFIGURACION DEL SISTEMA.....	36
4 - INSTALACION.....	37
4.1 - INSTALACION FISICA.....	37
4.2 - INSTALACION DE SOFTWARE.....	39
4.2.1 - REQUERIMIENTOS - COMPUTADOR PERSONAL.....	39
4.2.2 - REQUERIMIENTOS - TELEFONO CELULAR.....	40
5 - MODO DE OPERACION.....	41
5.1 - INTERFAZ GRAFICA - WEB.....	41

5.2 - INTERFAZ GRAFICA – WAP.....	51
6 - OBSERVACIONES.....	53
6.1 - MEJORAS EN HARDWARE.....	53
6.2 - MEJORAS EN SOFTWARE.....	53
7 - CONCLUSIONES.....	54
8 - REFERENCIAS Y BIBLIOGRAFIA.....	55
9 - APENDICES.....	57
9.1 - APENDICE A: ESQUEMATICOS.....	57
9.2 - APENDICE B: PERFIL – MANUAL DE AYUDA.....	59
9.3 - APENDICE C: HOJAS DE DATOS.....	62

# 1 - INTRODUCCION

## 1.1 - ¿QUE ES LA DOMOTICA ?

Domótica es el término "científico" que se utiliza para denominar la parte de la tecnología (eléctrica e informática), que integra el control y supervisión de los elementos existentes en una vivienda o en un edificio de viviendas.

El gran avance de la electrónica y de las comunicaciones en los últimos tiempos orientado al concepto de vivienda ha generado nuevas aplicaciones y tendencias basadas en la capacidad de procesar información y en la integración y comunicación entre equipos e instalaciones. Así concebida, una vivienda inteligente puede ofrecer una amplia gama de aplicaciones en áreas tales como:

- x Automatización de tareas domésticas
- x Gestión de la energía
- x Monitorización de salud
- x Comunicación con servicios externos
- x Seguridad
- x Operación y mantenimiento de las instalaciones, etc.

La definición de vivienda domótica o inteligente presenta múltiples versiones y matices. También aquí son diversos los términos utilizados en distintas lenguas: "casa inteligente" (smart house), automatización de viviendas (home automation), domótica (domotique), sistemas domésticos (home systems), etc.

De una manera general, un sistema domótico se compone de una red de comunicación que permite la interconexión de una serie de equipos que brindan información sobre el entorno del hogar y, basándose en ésta, realizan determinadas acciones sobre dicho entorno.

Ciertos dispositivos como detectores, sensores, y otros, transmitirán señales a un sistema central inteligente que procesará la información recibida. En función de dicha información y de una determinada programación, el sistema central actuará sobre determinados circuitos relacionados con las señales recogidas por los dispositivos.

En este sentido, una vivienda domótica se puede definir como: "aquella vivienda en la que existen agrupaciones automatizadas de equipos, que disponen de la capacidad de comunicarse interactivamente entre sí gracias a un sistema que las integra".

## 1.2 - CARACTERISTICAS DE LAS VIVIENDAS INTELIGENTES

Se pueden determinar rasgos generales propios y comunes a los distintos sistemas de una vivienda inteligente que son los que la caracterizan como tal.

Estas características generales, son las siguientes:

- x Control remoto desde fuera de la vivienda: presupone un cambio en los horarios en los que se realizan las tareas domésticas (por ejemplo: la posibilidad de que el usuario pueda activar la cocina desde el exterior de su vivienda, implica que previamente ha de preparar los alimentos) y como consecuencia permite al usuario un mejor aprovechamiento de su tiempo.
- x Programabilidad: el hecho de que los sistemas de la vivienda se pueden programar ya sea para que realicen ciertas funciones con sólo tocar un botón o que las lleven a cabo en función de otras condiciones del entorno (hora, temperatura interior o exterior, etc.) produce un aumento del confort y un ahorro de tiempo.
- x Control remoto desde dentro de la vivienda: a través de un esquema de comunicación con los distintos equipos (mando a distancia, bus de comunicación, etc.). Reduce la necesidad de moverse dentro de la vivienda, este hecho puede ser particularmente importante en el caso de personas de la tercera edad o discapacitadas.
- x Acceso a servicios externos: servicios de acceso a Internet, telecompra, etc. Para ciertos colectivos estos servicios pueden ser de gran utilidad (por ejemplo, unidades familiares donde ambos cónyuges trabajan) ya que producen un ahorro de tiempo.

## 1.3 - GESTION DE LA DOMOTICA

La domótica se encarga de gestionar principalmente los siguientes cuatro aspectos del hogar:

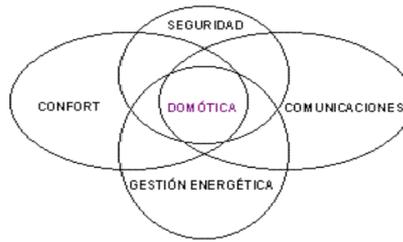
*Energía eléctrica:* La domótica puede gestionar el consumo de energía, mediante temporizadores,, termostatos, etc. También puede optimizar los costos ejecutando acciones durante los horarios de tarifa económica.

*Confort:* La domótica proporciona una serie de comodidades, como pueden ser el control automático de servicios como calefacción, iluminación y la gestión de elementos como portones, ventanas, riego automático, etc.

*Seguridad:* La seguridad que puede llegar a proporcionar un sistema domótico puede ser incluir:

Seguridad de los bienes: Control de acceso y control de presencia, así como la simulación de presencia o alarmas. Seguridad de las personas: Especialmente, personas mayores y/o personas enfermas. Incidentes y averías: Mediante sensores, se pueden detectar incendios, pérdidas de gas y agua, etc.

*Comunicaciones:* Este aspecto es imprescindible para acceder a multitud de servicios ofrecidos por los operadores de telecomunicaciones. La domótica tiene una característica fundamental, la integración de sistemas, por eso hay nodos que interconectan la red domótica con diferentes otras redes, como Internet, la red telefónica, etc.



*Figura 0. DOMOTICA INTEGRACION DE SISTEMAS*

## 1.4 - CARACTERISTICAS DE LOS SISTEMAS DOMOTICOS

*Tipo de arquitectura:* La arquitectura de un sistema domótico, especifica el modo en que los diferentes elementos del sistema se van a ubicar. Existen dos arquitecturas básicas: la arquitectura centralizada y la distribuida.

*Arquitectura centralizada:* Es aquella en la que los elementos a controlar y supervisar (sensores, luces, etc.) han de cablearse hasta el sistema de control de la vivienda (PC o similar).

*Arquitectura distribuida:* Es aquella en la que el elemento de control se sitúa próximo al elemento a controlar.

Cada elemento del sistema tiene su propia capacidad de proceso y puede ser ubicado en cualquier parte de la vivienda. Esta característica proporciona al instalador domótico una libertad de diseño que le posibilita adaptarse a las características físicas de cada vivienda en particular.

*Medio de transmisión:* En todo sistema domótico, los diferentes elementos de control deben intercambiar información unos con otros a través de un soporte físico (par trenzado, línea de potencia o red eléctrica, radio, infrarrojos, etc.).

*Protocolo de comunicaciones:* Una vez establecido el soporte físico, un sistema domótico se caracteriza por el o los protocolos de comunicaciones que utiliza, que no es otra cosa que el idioma o formato de los mensajes que los diferentes elementos de control del sistema deben utilizar para entenderse unos con otros y que puedan intercambiar su información de una manera coherente. Alguno de los protocolos y buses existentes son X-10, 1-Wire, IEEE Bus, etc.

*Preinstalación domótica:* La preinstalación domótica es la posibilidad de dejar preparada una vivienda para que, con el menor número de actuaciones, se le pueda instalar el sistema domótico en el momento en que el usuario lo demande. Para que un sistema pueda ofrecer una verdadera preinstalación domótica en una vivienda, ha de ser compatible con la instalación eléctrica actual, de tal manera que el usuario pueda, en la fase de construcción, elegir la preinstalación domótica y la instalación eléctrica convencional y con posterioridad, realizar cualquier tipo de automatización de su vivienda.

*Descripción del tipo de nodos:* Aplicado sobre todo a una red domótica de arquitectura distribuida

Nodos de control estándar, nodos de supervisión, nodos exteriores, nodos de comunicaciones.

En una red de arquitectura centralizada casi todos estos nodos pueden concurrir en un solo equipo.

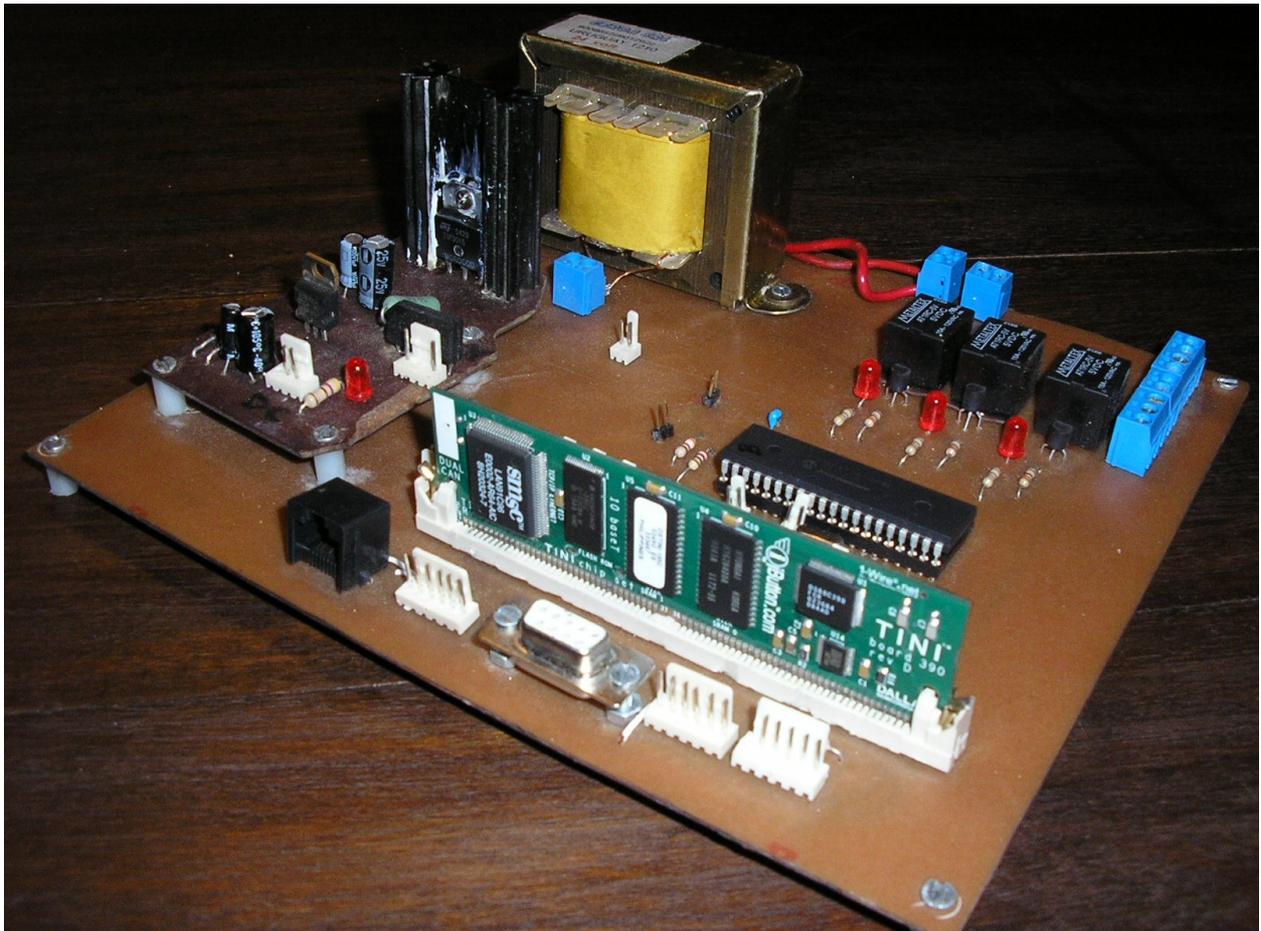
*Unidad de alimentación:* La unidad de alimentación es la encargada de suministrar energía (220V/50 Hz) a los diferentes elementos activos de la red domótica (sensores, nodos, etc.). En muchas ocasiones la unidad de alimentación incorpora una batería u otra unidad de alimentación redundante para prever ausencia de suministro eléctrico.

## 2 - OBJETIVO

### 2.1 - C.O.R.D. (Control Operativo Remoto de Dispositivos)

Es un modelo de sistema domótico de arquitectura centralizada que permite implementar las principales funciones de un sistema de este tipo.

Para una vivienda convencional el modelo puede gestionar y controlar a distancia un grupo de dispositivos ubicados en el entorno de la misma.



El sistema tiene las siguientes características y funcionalidades:

- x Acceso desde Internet sobre una red TCP/IP a través de una interfaz web.
- x Acceso desde la red celular a través de una interfaz wap sencilla.  
(No todas las funcionalidades están implementadas para la interfaz wap)
- x Control de usuarios para acceso al sistema.
- x Control de dispositivos de acciones ON/OFF.
- x Monitorización de sensores basados en el protocolo 1-Wire.  
(Por ejemplo: Sensores de temperatura)
- x Monitorización del consumo de energía eléctrica del hogar.
- x Ejecución de tareas programadas para el sistema.
- x Registro de sucesos del sistema.
- x Notificación de eventos al usuario vía correo electrónico.

### 3 - DESCRIPCION DEL SISTEMA

#### 3.1 - DESCRIPCION GENERAL

##### 3.1.1 - DESCRIPCION DE FUNCIONAMIENTO

C.O.R.D. permite la conexión a través de Internet con la vivienda del usuario, utilizando una interfaz desplegable en un Web Browser o en un teléfono celular con WAP. Permite monitorear y controlar dispositivos previamente definidos en el sistema (toma corrientes, interruptores, sensores de temperatura, etc.), además de entregar una medida de la potencia eléctrica consumida que esta teniendo el hogar (asumiendo un factor de potencia prefijado, 0.92).

Se pueden crear y luego seleccionar “perfiles” de funcionamiento de los dispositivos de la casa, y armar una agenda de funcionamiento del sistema para los distintos días del mes, la cual se mantiene actualizada por medio del programa.

C.O.R.D. mantiene un archivo de registro de cada suceso producido por los diferentes procesos que se están ejecutando y además puede notificar al usuario de algunos de estos eventos a una casilla de correo electrónico o a un teléfono celular (previa configuración).

Para poder comunicarse con el sistema, el usuario requiere en su hogar una conexión dedicada de banda ancha a Internet, ya que la comunicación está implementada sobre ethernet y para redes TCP/IP. El usuario podrá tener acceso a algunos parámetros de red del sistema y modificarlos.

También podrá acceder a un versión reducida del software del sistema a través de un teléfono celular, con capacidad para procesar páginas WAP.

Además será necesario hacer una adaptación a la instalación eléctrica de la vivienda (en caso que no se cuente con una pre-instalación domótica) para agregar los elementos necesarios para el funcionamiento del sistema (relés o llaves electrónicas, medidor de consumo y otros).

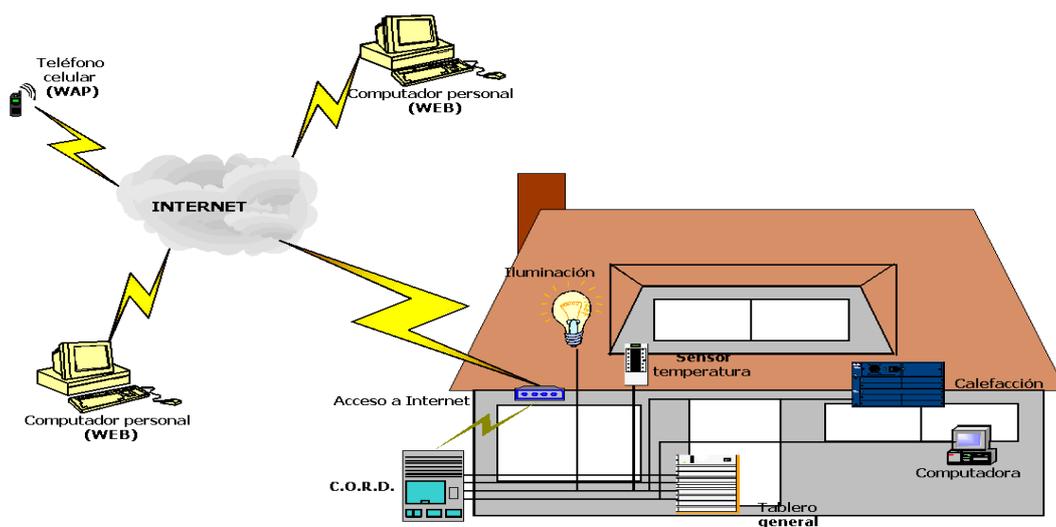


Figura 1. ESQUEMA EXPLICATIVO DEL SISTEMA DOMOTICO

### 3.1.2 - DESCRIPCION DE COMPONENTES

A nivel de hardware, C.O.R.D. se basa en un micro controlador de Dallas Semiconductor TINI (Tiny InterNet Interface) que tiene la inteligencia de todo el sistema (control de usuarios, manejo de dispositivos y buses, etc.) e implementa la interfaz de red. También tiene entre otras virtudes un bus 1-wire (utilizado principalmente para control de sistemas domóticos) y que permite trabajar a tales fines con todas las familias de dispositivos 1-wire.

También se usa en conjunto con el TINI un PIC 16F87XA de Microchip. Con esto se pretende centralizar el control referido a puertos digitales y analógicos del sistema. Ambos se comunican por un bus I<sup>2</sup>C, y como se mencionó anteriormente desde las salidas digitales del PIC se puede actuar sobre los dispositivos a controlar (básicamente dispositivos ON/OFF como interruptores y similares), y además es posible desde los puertos analógicos del PIC la lectura de señales como por ejemplo el consumo eléctrico.

Para la actuación sobre los dispositivos de 220VAC se utiliza una bornera de relés controlados por lo puertos digitales del PIC.

El Software consta de una interfaz gráfica para acceder desde un computador que ejecuta un applet desarrollado enteramente en Java permitiendo el monitoreo y control el sistema domótico. Para el acceso desde un teléfono celular se ejecuta una aplicación WAP desarrollada en WML permitiendo el monitoreo del sistema dadas las limitaciones visuales que tienen los teléfonos celulares.

En el TINI corren varios procesos en simultáneo que en su conjunto logran implementar todas las tareas del sistema.

Se implementa un servidor web, ftp y telnet. También tenemos una "agenda general" del sistema para controlar los eventos y un registro que permite guardar los eventos significativos del sistema y realizar las notificaciones correspondientes.

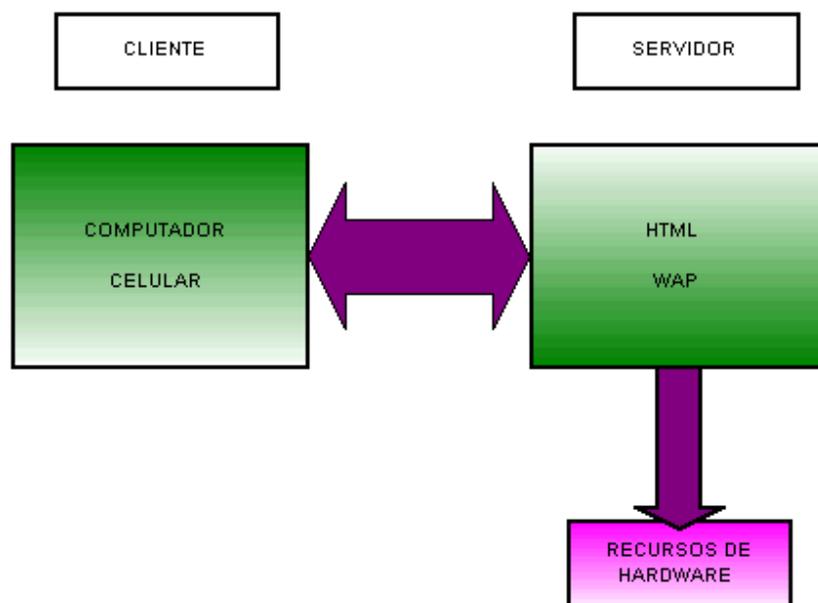


Figura 2. ESQUEMA DE COMPONENTES E INTERACCION

## 3.2 - HARDWARE

### 3.2.1 - ESQUEMA GENERAL

C.O.R.D dispone de 2 microcontroladores que controlan y procesan los dispositivos de entrada y salida al sistema.

La alimentación del sistema se realiza a través de la placa de fuente.

La entrada de medición se acondiciona a través del circuito de medida de consumo

La salidas a dispositivos digitales se realiza a través de una etapa de relés comandados por el PIC transistores y resistencias.

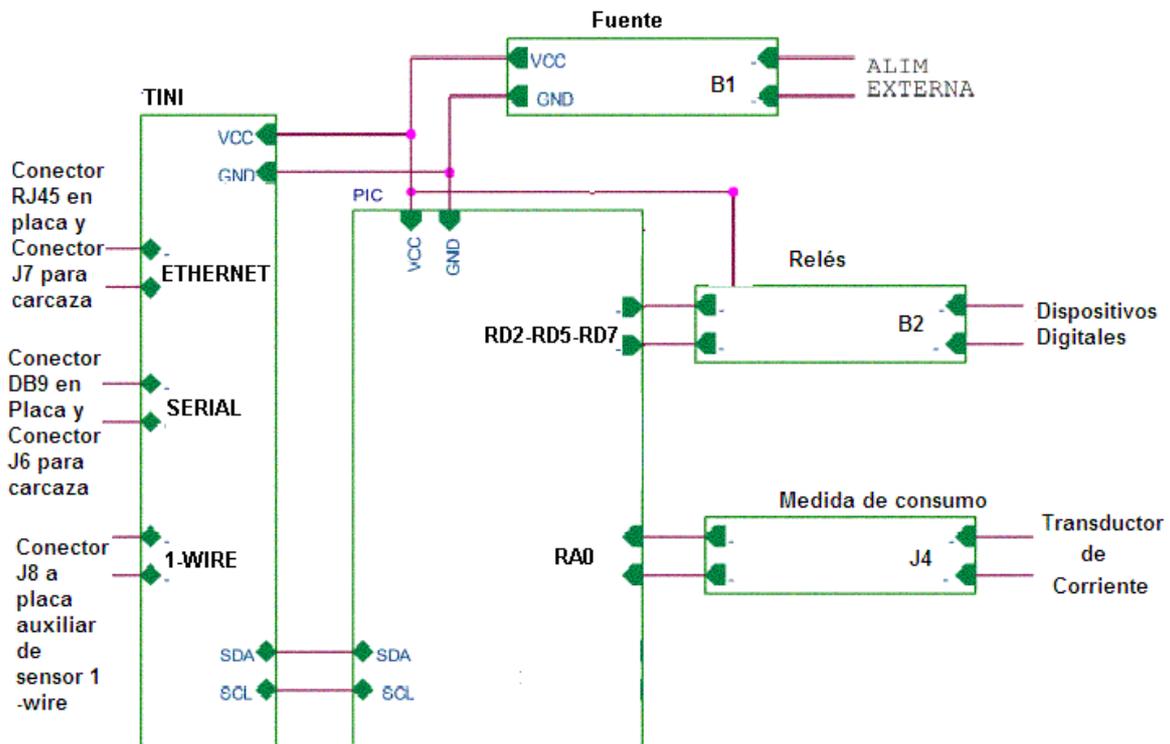


Figura 3. ESQUEMA GENERAL DE HARDWARE

El hardware del sistema esta constituido por los siguientes partes:

- x Fuente de alimentación
- x TINI
- x DS1820 (Sensor de temperatura 1-Wire)

- x PIC
- x Circuitos auxiliares de adaptación y protección
- x Bornera de relés
- x Transductor de corriente ( Externo)

### 3.2.2 - FUENTE DE ALIMENTACION

La fuente de alimentación es un módulo independiente que se monta sobre el circuito principal.

Las características principales son las siguientes:

Vin: 220 VAC rms

Vout: 5 VDC

Iout max: 1000 mA (limitado por el transformador de entrada y el regulador U1)

Se recomienda el uso de un fusible de 500 mA.

A continuación se muestra el circuito correspondiente:

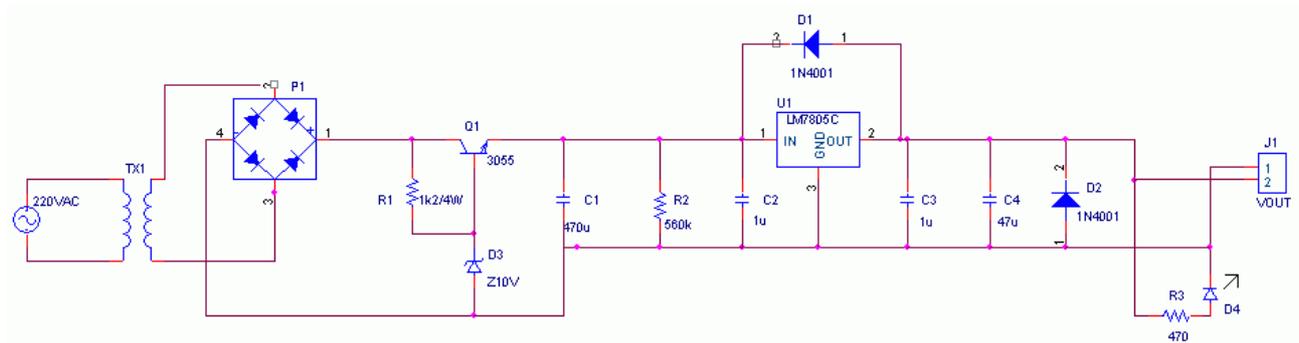


Figura 4. ESQUEMATICO DE LA FUENTE DE ALIMENTACION

La fuente consta de un transformador de entrada de 220VAC a 24VAC.

Se entra con 24VAC al puente de diodos P1.

La regulación de voltaje se realiza en dos etapas de forma de dividir la disipación de potencia entre los componentes.

La primera se basa en Q1 y D3, transistor de potencia 3055 y zener de 10V respectivamente.

Q1 trabaja en zona activa, de forma que el voltaje de emisor es mantenido en aproximadamente 9VDC. De esta forma el transistor Q1 y R1 disipan la potencia excedente de pasar de 24VAC rectificados a la salida del puente de diodos a los 9VDC que se tienen a la entrada del regulador.

El Filtro RC compuesto por R2 y C1 atenúa el ripple a la salida de Q1 para que llegue una onda más limpia al regulador U1.

La segunda etapa baja el voltaje a 5VDC mediante U1, un regulador LM7805C.

El diodo D1 actúa como protección de U1, evitando que el voltaje de salida supere el de entrada de éste. El diodo D2 protege al circuito de corrientes producidas por cargas inductivas.

Los capacitores C1 y C4 evitan caídas del voltaje suministrado a la carga en condiciones de alto consumo por períodos cortos. Los capacitores C2 y C3 filtran espurios producidos por interferencias o cargas no lineales.

La resistencia R3 limita la corriente del LED D4 de indicación de alimentación a 10 mA.

En el Apéndice A se pueden ver en detalle los esquemáticos del circuito.

### 3.2.3 - TINI

El TINI (Tiny InterNet Interface) es una plataforma de desarrollo compuesta de varios chip set que brindan entre otras cosas buena capacidad de procesamiento, control y varios niveles de comunicación a través de diferentes interfaces.

El hardware central del TINI se basa en chips LSI y básicamente son:

- x Micro controlador DS80C390
- x Flash ROM (512 KB)
- x Flash Static-RAM (expansible hasta 1 MB)

Además el TINI incluye entre otros módulos de hardware los siguientes:

- x Controlador 10Base-T Ethernet
- x Real-time clock
- x Interfaz para redes 1-Wire
- x Controlador CAN (Controller Area Network)
- x Puerto serial RS232
- x Expansión para trabajar con I/O de datos en paralelo
- x TTL I/O de propósito general
- x Puertos seriales síncronos

Puede trabajar con osciladores de hasta 40 Mhz. Está montado en una tarjeta SIMM de 72 pines y requiere solo +5V de alimentación.

El software para desarrollo de aplicaciones para el TINI se divide en dos categorías:

- x En código nativo (o código assembler) que es ejecutado por el propio microcontrolador.
- x En código escrito en lenguaje Java, a diferencia de muchos otros microcontroladores que solo pueden programarse en lenguaje C.

El TINI consta de un Application Programming Interface (API) de Java que puede usarse con cualquier herramienta de programación que soporte dicho lenguaje. Más en detalle el API esta formado de una parte de los packages definidos por Sun's Java Developer's Kit (JDK) 1.1 y otra parte packages con clases que pueden manejar todas las características propias del TINI.

Respecto al API del JDK soporta los siguientes packages:

- x java.io
- x java.lang
- x java.lang.reflect
- x java.net
- x java.util
- x javax.comm

Además la Java Virtual Machine (JVM) del TINI soporta múltiples threads, tipos primitivos de datos, strings, etc.

Las clases específicas del TINI están definidas en subpackages que se encuentran bajo el package com.dalsemi:

- x com.dalsemi.system
- x com.dalsemi.tininet
- x com.dalsemi.shell
- x com.dalsemi.comm
- x com.dalsemi.onewire

Estas clases son las que permiten entre otras cosas implementar por ejemplo el bus I<sup>2</sup>C.

### 3.2.4 - DS1820 ( Termómetro digital 1 – Wire )

Para implementar el uso de sensores que podría haber en una casa se utiliza un termómetro digital Maxim DS1820, con el que se pueden realizar mediciones de temperatura en grados centígrados. También posee una función de alarma con valores no volátiles programables por el usuario.

El DS1820 se comunica a través de un bus 1-Wire, que por definición requiere solamente una línea de datos (y tierra) para la comunicación con un microprocesador central que en este caso es el TINI.

Tiene un rango de temperatura de operación desde  $-55^{\circ}\text{C}$  hasta  $+125^{\circ}\text{C}$  y con una precisión de  $0.5^{\circ}\text{C}$  en el rango de  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ .



Adicionalmente, el sensor puede obtener energía directamente desde la línea de datos, eliminando la necesidad de alimentación externa (esto se conoce como “alimentación parásita”).

Cada sensor del estilo posee un código de serie de 64 bits, lo que permite conectar múltiples sensores en un mismo bus 1-Wire, y por consiguiente, con un único microprocesador pueden ser controlados una gran cantidad de sensores.

La funcionalidad básica del DS1820 es entregar la temperatura directamente en formato digital (no requiere un convertor analógico-digital adicional). La salida del sensor tiene una resolución de 9 bits, lo que equivale a pasos de 0.5°C. Para realizar una medición de temperatura y su conversión, el TINI debe enviar un comando que viene incluido en el API nativo. Luego de la conversión, el valor medido es guardado en un registro de 2 bytes en la memoria del sensor. El TINI debe entonces leer la memoria del sensor para obtener los datos.

### **3.2.4.1 - COMUNICACION TINI - 1-WIRE**

El sistema de bus 1-Wire usa un único dispositivo maestro para controlar uno o más dispositivos esclavos, en este caso el TINI (que puede supervisar todas las familias de dispositivos 1-Wire).

El DS1820 es siempre un esclavo. Cuando hay un solo dispositivo esclavo en el bus, el sistema se denomina «single-drop», si hay múltiples dispositivos, el sistema se denomina «multi-drop».

La siguiente descripción del bus 1-Wire está dividida en tres partes: Configuración del hardware, secuencia de transacción y señalización del bus 1-Wire.

#### **-Configuración del hardware**

El bus 1-Wire tiene por definición una sola línea de datos. Cada dispositivo (maestro o esclavo) se conecta a la línea de datos por un puerto «open drain» o «3-state». Esto permite que cada dispositivo libere la línea de datos cuando no está transmitiendo para que el bus quede disponible para ser usado por otro dispositivo.

El bus 1-Wire requiere de un resistor de pull-up externo de aproximadamente 5 kΩ, por consiguiente el estado de reposo del bus es alto. Si el bus es mantenido en bajo durante un período de más de 480 ms, todos los componentes del bus serán reseteados.

#### **- Secuencia de transacción**

La secuencia de transacción para acceder al DS1820 es la siguiente:

Paso 1: Inicialización, que consiste en un pulso de reset transmitido por el dispositivo maestro, seguido por un pulso de presencia transmitido por el esclavo. El pulso de presencia le permite al dispositivo maestro saber que hay dispositivos (como el DS1820) conectados al bus 1-Wire, y que están listos para operar.

Paso 2: Comando ROM (seguido por un intercambio de datos), después que el dispositivo maestro del bus 1-Wire detectó un pulso de presencia, puede enviar un comando ROM. Estos comandos permiten seleccionar un único dispositivo dentro de una red 1-Wire, a través de su código ROM único de 64 bits. Además permiten determinar la cantidad y tipo de dispositivos que se encuentran dentro de la red.

Paso 3: Comando de función (seguido por un intercambio de datos), luego que el dispositivo maestro envió un comando ROM para direccionar el DS1820 al que quiere comunicarse, puede

enviar uno de los comandos de función. Estos comandos permiten al dispositivo maestro leer y escribir, iniciar conversiones de temperatura y determinar el modo de alimentación.

Es sumamente importante seguir esta secuencia cada vez que se accede al DS1820, ya que no responderá si cualquiera de estos pasos faltan o son cambiados de orden.

Para comunicarnos con un dispositivo 1-Wire se utilizan clases proporcionadas en el API de TINI. Si bien en el API permite interactuar con todas las familias 1-Wire, en nuestro caso utilizamos la familia 10 que son sensores de temperatura con opción de alarma compuesta por los dispositivos: DS1820, DS18S20 y DS1920.

### **3.2.5 - PIC**

El manejo de los dispositivos digitales se implementa a través de los puertos de un PIC de Microchip Technology.

El mismo pertenece a la familia PIC16F7XA, están disponibles en encapsulados de 28 y 40 pines, y todos disponen de convertidores A/D de 10 bits, periféricos de comunicaciones que soporta SPI e I<sup>2</sup>C, tres contadores, etc.

Se programa con un juego de 35 instrucciones y pueden ser programados y reprogramados en circuito hasta 100 veces.

Al igual que para el resto de familias de PIC, la familia PIC16F7XA soporta un conjunto de herramientas de desarrollo, edición, ensamblado, compilación, grabación, emulación, incluidas en el entorno de desarrollo MPLAB-IDE.

Particularmente usamos el PIC 16F877A, trabajando con un oscilador es un cristal de 4 MHz.

Este dispositivo tiene suficiente capacidad para manejar un gran número de dispositivos a través de sus puertos digitales y gran capacidad para poder sensor varias señales por sus puertos analógicos; que evidentemente exceden el número utilizado para el proyecto. Sus entradas y salidas son TTL estándar.

#### **3.2.5.1 - COMUNICACION TINI - PIC**

Para realizar acciones sobre los relés, se utilizan los puertos digitales del PIC.

Para la comunicación con éste se utiliza el bus I<sup>2</sup>C (diseñado por Philips) que permite el intercambio de información entre circuitos integrados o similares. Funciona con una arquitectura maestro - esclavo ( en nuestra configuración TINI-maestro , PIC-esclavo), el maestro es quien inicia y maneja la comunicación El bus además permite manejar varios esclavos cada uno con una dirección que el maestro conoce. Utiliza comunicación serie y trabaja con la velocidad estándar del protocolo que es 100 Kbps.

El bus se basa en tres señales:

- SDA (System Data) por donde viajan los datos entre los dispositivos
- SCL (System Clock) por donde transitan los pulsos del reloj que sincronizan el sistema
- GND (Tierra) debe ser común a los dispositivos conectados en el bus

El protocolo es simple de usar, y el TINI y el PIC implementan el hardware y software necesario para trabajar con él.

Las ordenes se codificaron en un byte y se listan a continuación:

<i>Comando</i>	<i>Palabra(byte)</i>
<i>Encendido digital 0</i>	<i>0x10</i>
<i>Apagado digital 0</i>	<i>0x20</i>
<i>Encendido digital 1</i>	<i>0x11</i>
<i>Apagado digital 1</i>	<i>0x21</i>
<i>Encendido digital 2</i>	<i>0x12</i>
<i>Apagado digital 2</i>	<i>0x22</i>
<i>Encendido digital 3</i>	<i>0x13</i>
<i>Apagado digital 3</i>	<i>0x23</i>
<i>Lectura de consumo</i>	<i>0x40</i>

*Tabla 1: CODIFICACION COMANDO I2C*

Cuando se enciende o apaga un relé (y en consecuencia el dispositivo ahí conectado) se ejecuta una orden de escritura desde el TINI al PIC que provoca que baje o suba el voltaje de la pata del PIC (según la anterior codificación) que tiene dicho relé.

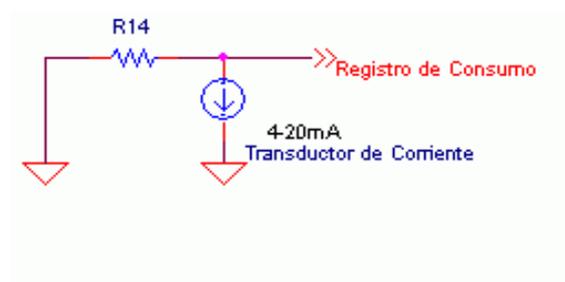
Cuando se mide consumo, a través del transductor que esta conectado a una entrada analógica del PIC, el TINI manda una orden de lectura. Luego lee los datos del buffer de salida de datos que tiene el PIC.

El protocolo I<sup>2</sup>C es manejado por los pines 10 y 11 del TINI (reloj y datos respectivamente) y es manejado por los pines 18 y 23 del PIC (reloj y datos respectivamente ). Este bus también requiere resistencias de pull-up.

### 3.2.6 - CIRCUITOS AUXILIARES

#### *CIRCUITO AUXILIAR PARA TOMA DE CONSUMO*

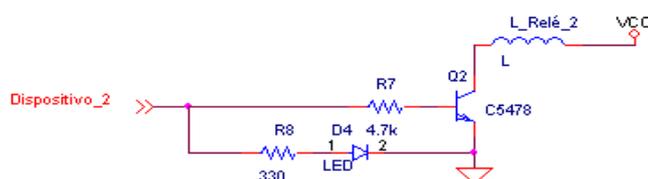
Se implementó un circuito auxiliar para la toma consumo como muestra la fig.5, en el cual la salida del transductor de corriente que trabaja en el rango de 4 a 20mA de corriente continua genera una tensión en bornes de R14 , de forma que llegue una señal analógica en voltaje a la entrada del conversor A/D del PIC ( en el rango de 1 a 5 VDC).



*Figura 5. CIRCUITO PARA MEDIR CONSUMO*

*CIRCUITO AUXILIAR PARA DISPOSITIVOS DIGITALES*

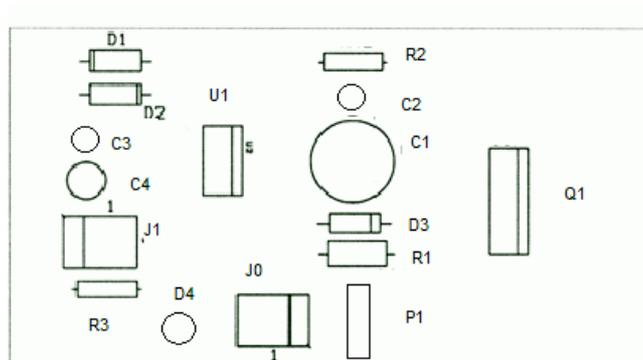
Para los dispositivos digitales se utilizó un circuito basado en un transistor bipolar NPN en una configuración de llave trabajando en corte y saturación. Se colocó una resistencia (R7 en la fig.6) para fijar la corriente de base, de forma que la salida digital del puerto del PIC ( max. corriente que puede entregar es 25mA) se conecta a la base del transistor utilizando a éste de llave para dejar pasar la corriente de colector  $I_c$  que activa la bobina del relé. También se puso un circuito indicador con LED y resistencia para indicar cuando la salida correspondiente del PIC esta activada.



*Figura 6. CIRCUITO ADAPTADOR DE PUERTOS DIGITALES*

**3.2.7 - CONEXIONADO INTERNO DEL SISTEMA**

A continuación se muestran los planos de ubicación de los componentes en el impreso principal y en el de la fuente de alimentación.



*Figura 8. ESQUEMA DE LA FUENTE DE ALIMENTACION*

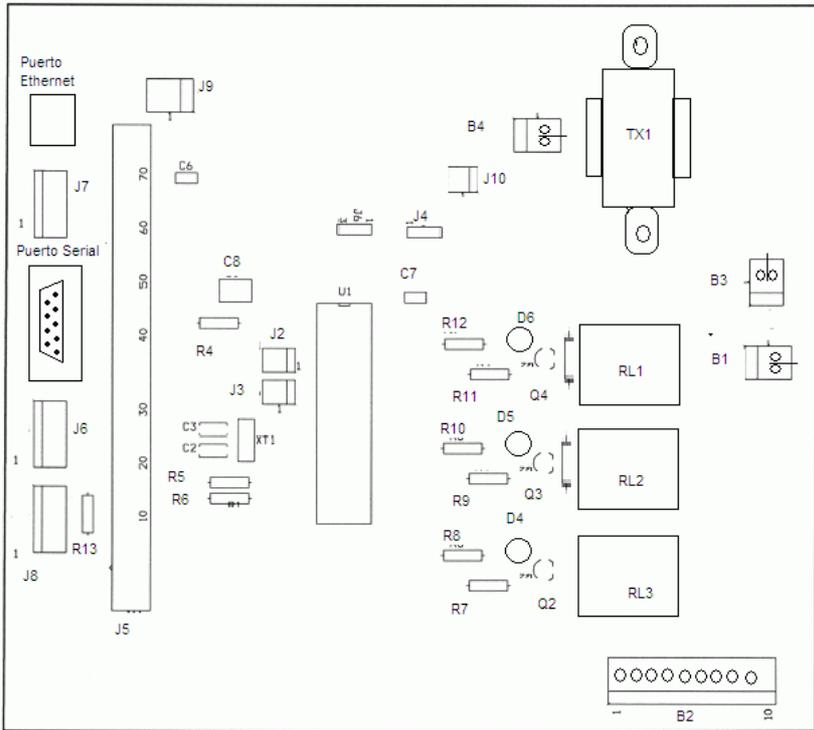


Figura 7. ESQUEMA DE LA PLACA PRINCIPAL

Las tablas siguientes muestran el pinout de los conectores en la fuente de alimentación (Conector en placa de fuente) y el correspondiente en la placa principal (Conector interior).

Conector en Placa Fuente	Conector interior
<b>J0</b>	<b>J10</b>
1	2
2	NC
3	NC
4	1

Tabla 2: PINOUT CONECTOR AC

Conector en Placa Fuente	Conector interior	Referencia
<b>J1</b>	<b>J9</b>	
1	NC	
2	1	VCC
3	2	GND
4	NC	

Tabla 3: PINOUT CONECTOR DC

## DESCRIPCION DEL SISTEMA

En los cuadros de abajo se muestra el conexionado de los conectores de red, 1-wire, serial y de los relés con su correspondiente en la placa principal (Conector Interior). Al tener un conector en la placa principal permite la adaptabilidad del equipo a una carcasa, donde solamente se tendrían que poner los puertos en la misma y conectarlos contra el jumper adecuado en la placa principal.

Conector en Placa RJ45 Puerto Ethernet	Conector interior
	<b>J7</b>
1	4
2	3
3	1
4	NC
5	NC
6	2
7	NC
8	NC

**Tabla 5:** PINOUT CONECTOR DE RED ETHERNET RJ45

Conector en Placa 1-wire	Conector interior
	<b>J8</b>
1	5
2	4
3	NC
4	6
5	NC
6	NC
7	NC
8	NC

**Tabla 4:** PINOUT CONECTOR 1-WIRE

Conector en Placa DB9 Puerto Serial	Conector interior
	<b>J6</b>
1	1
2	2
3	3
4	4
5	NC
6	NC
7	NC
8	NC
9	NC

**Tabla 7:** PINOUT CONECTOR SERIAL DB9

Bornera en Placa	Referencia
<b>B1</b>	
1	L1
2	L2
<b>B2</b>	
1	Común RL3
2	NC RL3
3	NA RL3
4	Común RL2
5	NA RL2
6	NC RL2
7	Común RL1
8	NA RL1
9	NC RL1
<b>B3</b>	
1	L1 Primario TX1
2	L2 Primario TX1
<b>B4</b>	
1	L1 Secundario TX1
2	L2 Secundario TX1

**Tabla 6:** PINOUT BORNERA RELÉS

### 3.3 - SOFTWARE

#### 3.3.1 - ESQUEMA Y DIAGRAMA DE FLUJOS

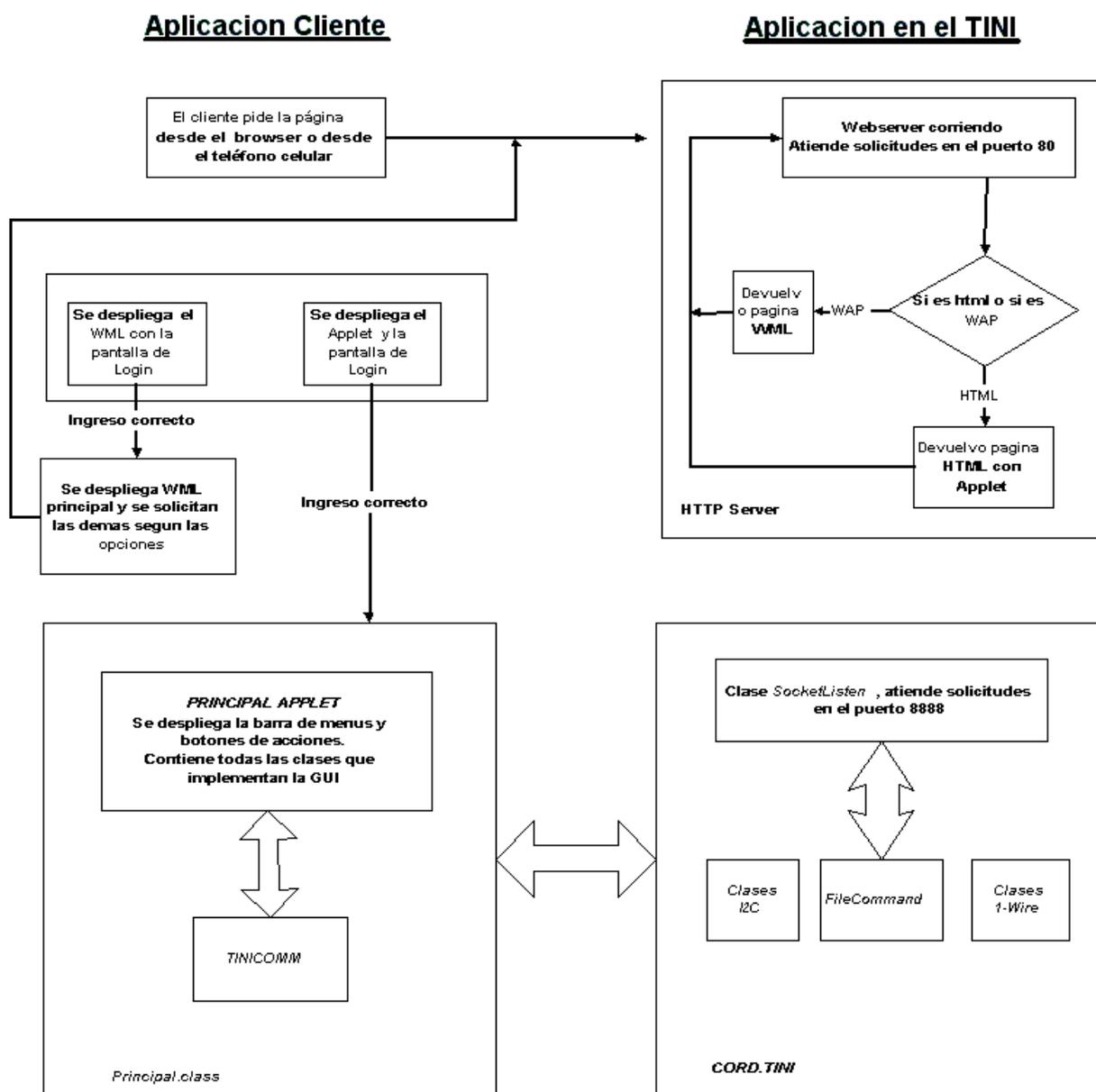


Figura 9. DIAGRAMA DE INTERACCION DE LOS ELEMENTOS DE SOFTWARE

### 3.3.2 - HERRAMIENTAS Y DESARROLLO DE SOFTWARE

Los programas usados para la programación, configuración e instalación del software del sistema; así como también los programas utilizados para interactuar con el TINI y el PIC se enumeran a continuación:

- x Java (con JCreator)
- x TINI SDK
- x MPLAB
- x Nokia Internet Kit
- x Emuladores y navegadores WAP

#### 3.3.2.1 - Java

Java es un lenguaje de programación de alto nivel y orientado a objetos, creado por Sun Microsystems. Tiene como gran virtud el hecho de crear aplicaciones multiplataforma, que no depende del sistema operativo utilizado ya que los programas creados en este lenguaje se ejecutan sobre un “maquina virtual” llamada JVM (Java Virtual Machine) que interpreta el código neutro y lo transforman en código propio del sistema operativo.

Para cualquier aplicación que se desarrolle en Java, se cuenta con un número de clases preexistentes que forman parte del lenguaje y se conocen como el API (Application Programming Interface).

Java 2 es la tercera versión de este lenguaje; y es la que se utilizó para el desarrollo de los módulos de software.

Para desarrollar, compilar y ejecutar programas se utilizó el JDK(Java Development Kit) distribuido por Sun. Además Sun distribuye una versión reducida de JDK conocida como JRE(Java Runtime Enviroment) destinada únicamente para ejecutar código Java (pero no permite compilar). Esto nos impone un requerimiento en el computador del usuario y es que dicho equipo debe tener instalado el JRE para poder ejecutar el applet.

La elección del idioma de programación se decidió por ser soportado por el TINI que cuenta con una JVM. Esto hace posible integrar un programa con el API de Java y el API propio del TINI para el acceso los recursos de su propia plataforma.

El desarrollo de software se realizó completamente en JCreator v3.50, un editor de código Java. Dicho programa se integra con el JDK, brindando una interfaz más cómoda para trabajar.

#### 3.3.2.2 - TINI SDK

A nivel de hardware se usó una placa de desarrollo TINI Board Model 390 (TBM390) de Dallas Semiconductor la que implementa todo lo necesario para desarrollo del TINI. Consta de un socalo SIMM de 72 pines, 2 puertos seriales, un puerto RJ45 y un puerto RJ11 y alimentación DC a través de un regulador.

Para la comunicación con el TINI contamos con una programa llamado MTK v2.30 (MicroController Tool Kit) que se trata de una consola de comunicación por puerto serial (entre el puerto serial del computador y uno de los puertos seriales controlados por el TINI). El MTK puede acceder al TINI en una instancia de SETUP (como en un computador estándar) o en una instancia similar al telnet ingresando al sistema operativo que tiene incorporado el TINI, y que será explicado más adelante.

El TINI permite también actualización de firmware y sistema operativo. Se descargan los archivos necesarios \*.tbin y utilizando el MTK siguiendo unos sencillos pasos se logra dicha actualización. Hoy día se cuenta con varias versiones del firmware y del sistema operativo.

A nivel de software tiene un sistema operativo llamado Slush con un shell similar al sistema operativo UNIX (actualmente también existe la versión 1.17 de éste y es la que se utilizó durante el proyecto).

Dentro de los comandos que pueden ejecutarse se pueden destacar:

- login: Permite el acceso a usuarios con distintos niveles de privilegios, por defecto trae definidos *root* y *guest*
- ipconfig: Permite ver y configurar todos los parámetros de red
- date: Permite la configuración de la fecha y la hora del sistema
- gc: Garbage collector
- startserver: Permite iniciar los servicios del TINI como ftp, telnet, etc.
- java: Es la JVM que permite ejecutar aplicaciones compiladas para el TINI (más adelante se explica el proceso)
- sendmail: Similar al demonio de UNIX, implementa un cliente SMTP para enviar correo electrónico.

Después de tener compilada un programa Java, por ejemplo *HelloWorld.class* es necesario crear un ejecutable que es el que la JVM del TINI interpreta y puede ejecutar. Para esto se utiliza una aplicación Java llamada *TiniConvertor* que transforma el archivo \*.class en \*.tini. Luego se ejecuta la aplicación creada, por ejemplo: `java HelloWorld.tini`.

En nuestro caso fue necesario la utilización de *BuildDependency* (otra aplicación Java) que implementa el *TiniConvertor* y aparte permite utilizar otros módulos como el 1-wire. La línea de compilación general del programa es:

```
java -classpath %TINI_HOME%\bin\tini.jar; BuildDependency -p
%TINI_HOME%\bin\owapi_dependencias_TINI.jar -x %TINI_HOME%\bin\owapi_dep.txt -add
Thermometers -d %TINI_HOME%\bin\tini.db -path %TINI_HOME%\bin\modules.jar -add MAILTO -p
%TINI_HOME%\bin\modules.jar -n mod_i2c.tlib -f%1 -o cord.tini
```

### 3.3.2.3 - MPLAB

Herramienta de desarrollo para PICs de Microchip, actualmente en su versión v7.22 utilizada para este proyecto.

Este software presenta una plataforma completa de desarrollo, permite editar, compilar, ensamblar, grabar, emular y demás tareas a realizar con un microprocesador.

Los pasos para instalar un programa en el PIC son los siguientes:

1. Crear un proyecto nuevo, siguiendo los pasos que propone el programa.
2. Se edita el programa en código assembler para el PIC, siguiendo la nomenclatura correspondiente para programar estos dispositivos: variables, librerías, etc. El archivo deberá tener la extensión \*.asm
3. Luego se puede proceder a compilar el código a partir del archivo \*.asm donde el software, se encarga de generar una lista de errores.  
Si se desea, el programa puede emularse para ver el comportamiento del mismo, también tiene ejecución paso por paso y demás opciones de debug.
4. Después de esto y cuando ya no halla errores el programa crea un archivo \*.hex el cual se procede a grabar en el PIC. Para la grabación del programa se utilizó el grabador de Microchip PICSTART Plus.
5. Después MPLAB permite verificar la grabación del programa en el PIC. También es posible ejecutarlo y ver el funcionamiento del mismo así como los registros, memorias y demás componentes internos del PIC.

Después de esto el PIC esta pronto para poner en la placa, con la alimentación, el oscilador y los circuitos auxiliares correspondientes.

### **3.3.2.4 - NOKIA INTERNET KIT**

Es un poderoso kit de desarrollo de la empresa Nokia, para desarrollar todo tipo de aplicaciones para celulares y similares. Tiene un editor de archivos WML y WMLScript con corrector de errores. Este software permite compilar los tipos de archivos antes mencionados para obtener los correspondientes compilados WMLC y WMLSC.

### **3.3.2.5 - NAVEGADORES WAP – EMULADORES DE TELEFONOS CELULARES**

Dado que desde un pc sólo es posible “emular” el funcionamiento de un teléfono celular y probar que las aplicaciones WML desarrolladas fueran soportadas por la mayoría de celulares se utilizaron varios navegadores y emuladores de páginas WAP que existen en Internet: Wapsilon, Ttag, OpenWave; siendo este último uno de los mejores además de venir con un kit de desarrollo para aplicaciones. Otras aplicaciones a mencionar son:

#### *WAPTOR*

Este software es un editor de código WML, también permite tener una pre-visualización de la página así como verla en distintos modelos de teléfonos celulares.

#### *WINWAP*

Permite la navegación de páginas WML, se puede dar una URL válida que atienda WML. Se

puede navegar y soporta todos los tipos de archivos que se utilizan para trabajar con celulares: WML, WMLC, WMLScript, WMLSC y WBMP. También se puede ver el código fuente de la página.

### 3.3.3 - TINI

#### 3.3.3.1 - DEMONIOS

El sistema inicia en el TINI varios demonios automáticamente al comenzar el sistema operativo (para esto sera necesario agregar en el archivo ".startup" de configuración del Slush las líneas correspondientes, p.e. "*java cord.tini &*"). Estos procesos se mantienen corriendo mientras el sistema este funcionando.

Los procesos son:

- x *Cord*
- x *WebServer*
- x *Diaria*
- x *SocketListen*
- x *Log*

Para distintas funcionalidades del sistema se pueden tener otros procesos corriendo como la medición de consumo eléctrico y/o el sensado de temperatura. Estos son iniciados por el demonio principal *Cord* y generalmente son temporales.

#### - *CORD*

Es el proceso principal del sistema, inicializa el *Webserver*, *Diaria*, *SocketListen* y *Log*. Esta contenido en el archivo "*cord.tini*". En éste se encuentran las clases para el acceso al sistema, manejo de archivos, comunicación entre applet y TINI, comunicación con el PIC para la ejecución de los comandos I<sup>2</sup>C y comunicación con los dispositivos 1-Wire entre otros.

#### - *WEBSERVER*

Este proceso implementa un servidor web, para atender las solicitudes de los clientes HTTP y WAP.

El TINI tiene en el API del sistema un package que implementa un HTTP Server. A partir de este package se creó uno nuevo llamado "*cord.server.http*" , con la finalidad de tener la capacidad de distinguir una solicitud HTML o una WML.

El parámetro que se chequea es la propiedad MIME y que el navegador setea en el archivo pedido para desplegar, si es HTML o WML. Con esa distinción el sistema entrega el archivo correspondiente por ejemplo: index.html o index.wml.

En el caso de WAP, existe un equipo en la red celular conocido como "puerta de enlace WAP" que es el que se encarga de traducir la página en lenguaje WML a protocolo WAP para que el móvil pueda desplegarla.

Parámetros MIME aceptados por el servidor:

- x *"text/html"*
- x *"text/plain"*
- x *"text/vnd.wap.wml"*
- x *"text/vnd.wap.wmlscript"*
- x *"image/gif"*
- x *"image/jpeg"*
- x *"image/bmp"*
- x *"application/octet-stream"*
- x *"application/vnd.wap.wmlscriptc"*

#### **- DIARIA**

Es la agenda principal del sistema y la que se encarga de generar y controlar las tareas diarias programadas para el mismo. Para esto todos los días a la cero hora el programa principal lee el "perfil" que tiene que aplicarse a los dispositivos y se genera las tareas diarias que son leídas por dicha aplicación "*Diaria.class*". Cuando llega la hora ejecuta lo que corresponde, ya sea un comando I<sup>2</sup>C o 1-Wire.

#### **- SOCKETLISTEN**

Este proceso implementa la conexión entre las aplicaciones del TINI y del applet. Al iniciar el sistema inicia un SocketServer para atender consultas en el puerto 8888. Cuando el applet quiere interactuar con el sistema envía tramas que el *Socketlisten* recibe, procesa y realiza las acciones en el TINI. La comunicación entre TINI y applet, y las tramas utilizadas en la comunicación se explican en el punto 3.3.5

#### **-LOG**

Es el registro del sistema, guarda los eventos generados en el archivo "*registro.log*".

Se ejecuta como un proceso independiente para optimizar el rendimiento del sistema, ya que cualquier otro proceso descarga en éste lo que necesita registrar y continua con su tarea.

### **3.3.3.2 - CLASES AUXILIARES**

Las clases usadas por los demonios que se ejecutan en el TINI son:

- x *Archivo.class*
- x *FileCommand.class*
- x *Perfil.class*

- x *ClaseI2C.class*
- x *ClaseWire.class*
- x *Consumo.class*
- x *Notificaciones.class*

#### **- ARCHIVO**

Clase para crear y escribir archivos “circulares” en el sistema de archivos del TINI.

La clase permite definir el nombre del archivo, la cantidad de líneas y el largo de las mismas.

Un ejemplo de un archivo manejado por esta clase es el *"registro.log"*, ya que si el archivo no fuera “circular” crecería ocupando todo el espacio libre del TINI. Cuando se llega a la última línea del archivo, vuelve a la primera (que es la más vieja) para continuar escribiendo.

También son archivos de la clase los que guardan los datos sensados para el consumo o la temperatura.

#### **- FILECOMMAND**

Clase que permite leer, escribir y listar archivos que están en el TINI y que son modificados por procesos y/o eventos.

Cuando un usuario solicita ver los perfiles guardados en el sistema, el *FileCommand* se encarga de ello, igual cuando se quieren guardar datos en los archivos de configuración del sistema.

#### **- PERFIL**

Clase que maneja los archivos \*.prf que son usados por el administrador de dispositivos para configurar las acciones en la vivienda.

#### **- CLASEI2C**

Clase que implementa todo lo necesario para la comunicación con el PIC utilizando el bus I<sup>2</sup>C.

Permite tanto la escritura (a través de las salidas digitales del PIC para accionar los dispositivos digitales) como la lectura (a través de la entrada analógica del PIC para medir consumo).

En caso de falla de la comunicación, deja registro de lo ocurrido.

#### **- CLASEIWIRE**

Clase que implementa todo lo necesario para la comunicación con el bus 1-Wire, y en particular controla el sensor de temperatura DS1820.

En caso de falla de la comunicación, deja registro de lo ocurrido.

#### **- CONSUMO**

Clase que se encarga de la medida del consumo, básicamente funciona como la agenda *"Diaria"* pero atendiendo según lo solicite el usuario la lectura de la entrada analógica del PIC destinada para tal propósito

También crea un objeto de la *ClaseI2C* para la lectura de los datos. Además genera el archivo con los datos medidos que luego puede ser visto desde el applet.

#### **- NOTIFICACIONES**

Clase que implementa un cliente SMTP, similar al sendmail del sistema operativo del TINI y envía correos electrónicos a las direcciones de notificación configuradas por el usuario. Puede ser invocada por cualquier proceso del sistema para avisar de los eventos según el nivel seleccionado.

El sistema tiene dos niveles de notificaciones: INFORMATIVO y CRITICO.

El nivel INFORMATIVO contiene todos los mensajes que el sistema puede brindar incluyendo los del nivel CRITICO.

Eventos de nivel informativo:

- x Se activó/desactivó dispositivo por la agenda diaria.
- x Se inició/finalizó medida de consumo y/o temperatura.
- x Ingreso de usuarios: *admin* y *guest* a través del applet.
- x Agenda diaria pasó a modo cronómetro.
- x Cambios en archivos del sistema (*agenda.cfg*, *notifica.cfg*, etc...).

Eventos de nivel crítico:

- x Archivo de agenda "*agenda.cfg*" no se encontró o esta dañado.
- x Archivo de perfil "*perfilX.prf*" no se encontró o esta dañado.
- x Archivo de notificaciones "*notifica.cfg*" no se encontró o esta dañado.
- x Archivo "*dis2dig.txt*" no se encontró o esta dañado.
- x Falló en la comunicación con puertos digitales.
- x No se encontró dispositivo 1-wire.
- x Activación/Desactivación de dispositivo por comando directo.
- x Activación/Desactivación de la agenda diaria por comando directo.
- x Error en la medida de consumo.
- x Consumo por encima/debajo del umbral configurado.
- x Temperatura por encima/debajo del umbral configurado.
- x Cambió la dirección IP del sistema. Nueva IP X.X.X.X

### 3.3.4 - APPLET

Cuando el servidor recibe una solicitud desde un navegador web, devuelve a la página *index.html*. En ella se indican los archivos necesarios para visualizar el applet. Se descarga entonces el archivo *CordApplet.jar* (creado con el utilitario *jar* del JDK de Sun) que contiene todas las clases de Java necesarias. Las clases son interpretadas por el JRE (Java Runtime

Environment).

### 3.3.4.1 - CLASES AUXILIARES

Las clases que componen el *Cord.jar* y utilizadas por el applet son:

- x *CordApplet.class*
- x *CordGUI.class*
- x *TiniComm.class*
- x *Grafico.class*

#### - CORDAPPLET

Al iniciar el applet aparece un pantalla de login con usuario y contraseña para permitir el acceso al sistema. Cuando el usuario es validado (ya sea *root* o *guest*) quedan determinados sus permisos sobre el sistema. Se permite el acceso de un usuario *root* al mismo tiempo y en caso de intentar ingresar como tal desde otro equipo se notifica que ya existe un usuario conectado.

Al tener un ingreso exitoso se ejecuta la Interfaz Gráfica de Usuario (GUI).

Para salir se ejecuta un logout que cierra la aplicación principal, “limpiando” el navegador y volviendo a la pantalla de login esperando el ingreso de nuevos usuarios.

#### - CORDGUI

Cuando se ingresa al sistema se inicia el applet *CordGUI.class*. Este applet es la GUI del sistema.

En la aplicación se implementan las siguientes funcionalidades:

- x Control sobre la agenda diaria, se permite iniciarla y detenerla
- x Control sobre el proceso de medida de consumo, se permite iniciarlo y detenerlo
- x Ejecución de comandos directos sobre los dispositivos declarados en el sistema
- x Administración de agenda general de funcionamiento y de “perfiles” de dispositivos
- x Configuración de los parámetros de red del TINI y de las notificaciones del sistema
- x Visualización de gráficas históricas de consumo eléctrico y de temperatura de los sensores
- x Visualización del registro del sistema

#### CONFIGURACION DE RED

## DESCRIPCION DEL SISTEMA

Permite cambiar el nombre del host, la dirección ip, mascara de red y puerta de enlace del equipo.

### *NOTIFICACIONES*

Para las notificaciones se utiliza el sistema de eventos manejados por el registro pudiendo configurarse el nivel de evento a notificar.

Se puede configurar una dirección de e-mail y un número celular a los que se mandaran las mensajes con el nivel de notificaciones seleccionado.

Las notificaciones a un teléfono celular se implementan como un e-mail que se envía al móvil (que tiene una casilla asociada del proveedor de telefonía correspondiente), el sistema soporta los tres proveedores actuales de telefonía celular en Uruguay.

Para mandar los e-mail es necesario saber los servidores que tienen el servicio SMTP de cada proveedor de casillas de correo. Por esta razón y para evitar una rutina de resolución de dominios es que fijan alguno de los dominios de correo más usados:

- |                                      |   |
|--------------------------------------|---|
| 1. 9_celular_codigo@ancelinfo.com.uy | ( servidor SMTP: mail.ancelinfo.com.uy )      |
| 2. 94_celular@sms.movistar.com.uy    | ( servidor SMTP: sms.movistar.com.uy )        |
| 3. 59896_celular@sms.ctimovil.com.ar | ( servidor SMTP: sms.ctimovil.com.ar )        |
| 4. adinet.com.uy                     | ( servidor SMTP: mx-adinet.adinet.com.uy )    |
| 5. hotmail.com                       | ( servidor SMTP: mx1.hotmail.com )            |
| 6. gmail.com                         | ( servidor SMTP: gmail-smtp-in.l.google.com ) |
| 7. yahoo.com                         | ( servidor SMTP: mx1.mail.yahoo.com )         |
| 8. fing.edu.uy                       | ( servidor SMTP: mailhost.fing.edu.uy )       |

### *VISUALIZACION DE CONSUMO*

El sistema permite comenzar a tomar datos para tener un histórico de consumo.

La vista de los datos se hace a través de una gráfica de puntos leídos desde el archivo “consumo.txt” generado por la aplicación y que reside en el TINI.

### *VISUALIZACION DE TEMPERATURA*

Es análoga a la visualización de consumo y se genera a partir del archivo “temperatura.txt” que también reside en el TINI.

### *VISUALIZACION DE LOG DEL SISTEMA*

Todos los mensajes generados en el sistema se guardan en el archivo “registro.log” y puede ser leído a través del applet.

### *AGENDA GENERAL Y PERFILES DEL SISTEMA*

Es donde reside el corazón del sistema.

Nos muestra los distintos dispositivos del hogar, según una distribución de árbol.

Para cada dispositivo podemos ver las horas y acciones configuradas en el perfil actual.

Se pueden modificar los parámetros de los dispositivos, agregar y remover dispositivos y habitaciones.

Se pueden editar otros perfiles guardados así como también crear nuevos perfiles.

El sistema permite crear hasta 15 perfiles y los nombres de los archivos guardados por el sistema tienen la nomenclatura "*perfilX.prf*". Los archivos de perfiles están en el directorio "*/perfiles*" en el TINI.

#### **- TINICOMM**

Es la contraparte del proceso *SocketListen* que se ejecuta en el TINI y también se ejecuta como un demonio en paralelo con el applet.

Cuando se produce un evento en alguna pantalla del applet que requiera iniciar una transacción con el TINI entonces esta clase se encarga de armar la trama que corresponda, mandarla al TINI, esperar la respuesta y procesarla devolviéndole algún tipo de evento al usuario.

Los detalles de la comunicación se pueden ver en el punto 3.3.5

#### **- GRAFICO**

La clase *Grafico.class* se utiliza como una herramienta para dibujar en el applet una gráfica ( independiente de quien utilice la clase, puede representar datos de consumo o de temperatura).

Los datos de la gráfica se toman de los archivos del sistema "*consumo.txt*" y "*temperatura.txt*". Éstos son archivos circulares (para no ocupar mucho espacio en el sistema de archivos) por lo tanto vemos una ventana de tiempo determinada, se optimizó para ver con un promedio de 5 minutos las muestras de un día.

### **3.3.5 - COMUNICACION TINI - APPLLET**

Para poder acceder a los recursos del TINI desde un applet, que se ejecuta en el computador del cliente es necesario establecer una comunicación entre aplicaciones.

El applet siempre inicializa la conexión hacia el TINI, abriendo una conexión TCP al puerto 8888. La aplicación abre dos sockets sobre esta conexión. Los sockets tienen seteado un TIMEOUT igual a 1 minuto, y en caso de falla en la comunicación después de pasado este período se cierran.

Si la conexión se establece correctamente se inicializan dos canales de comunicación (streams) por los que se escriben y leen bytes. Tanto los procesos del applet como los del TINI,

transforman la información con la que están trabajando en un conjunto de bytes que es lo intercambiado entre ambos. La otra parte, debe leer esos bytes, y procesarlos.

Para optimizar el manejo de las aplicaciones se creó un protocolo de entramado para el intercambio de información.

Una trama típica de este protocolo es la siguiente:

### TRAMA CARACTERISTICA PARA COMUNICACION ENTRE LAS APLICACIONES

Largo máximo 64 KB

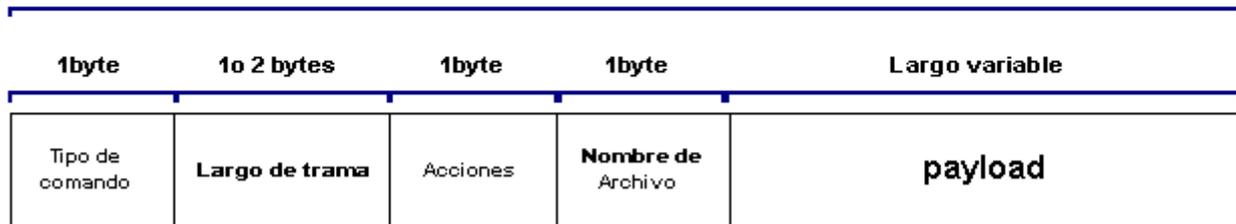


Figura 10. TRAMA DE COMUNICACION ENTRE TINI - APLET

Las distintas codificaciones de comandos, acciones y archivos son las siguientes:

Tipo de Comando	Palabra(byte)
De Archivo	0xF1
De Agenda	0xF2
De Consumo	0xF3
Perfil Activo	0xF4
De Red	0xF5
Login / Logout	0xF6
De Comando	0xF7

Tabla 8: TIPOS DE COMANDO

ACK de Acciones	Palabra(byte)
Lectura de Archivo	0x82
Escritura de Archivo	0x83
Iniciar Agenda	0x82
Parar Agenda	0x83
Iniciar Consumo	0x82
Iniciar Consumo	0x83
Lectura Param. Red	0x82
Escritura Param. Red	0x83
Login valido	0x82
Login invalido	0x83
Lista de Perfiles	0x8B

Tabla 9: ACK DE ACCIONES

Acciones	Palabra(byte)
Lectura de Archivo	0x80
Escritura de Archivo	0x81
Iniciar Agenda	0x80
Parar Agenda	0x81
Iniciar Consumo	0x80
Iniciar Consumo	0x81
Lectura Param. Red	0x80
Escritura Param. Red	0x81
Login al Sistema	0x80
Logout al Sistema	0x81
Lista de Perfiles	0x8A

Tabla 9: ACCIONES DE COMANDOS

Nombre de archivo	Codificación(byte)
Registro.log	0xA0
Consumo.txt	0xA1
Notifica.cfg	0xA2
Temperatura.txt	0xA3
Agenda.cfg	0xA4
Dis2dig.txt	0xA6
Perfil01.prf	0xF1
Perfil02.prf	0xF2
...	...
Perfil15.prf	0xFF

Tabla 10: CODIFICACION DE ARCHIVOS

El TINI siempre responde con un ACK de la trama recibida y del comando correspondiente.

El campo de acciones se especifica en los casos que son necesarios.

*Ejemplo de comunicación APPLLET-TINI - Lectura de archivo "registro.log" :*

Cuando desde el applet se selecciona MOSTAR => "Registro del sistema", se dispara un evento de lectura de archivo. El applet llama a la clase *TiniComm* para que genere la siguiente trama:

#### TRAMA DE LECTURA DE ARCHIVO "registro.log"

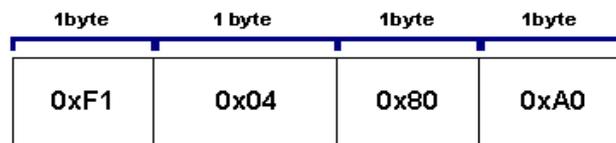


Figura 11. TRAMA DE LECTURA

Luego el método *txrxBytes()* del *TiniComm* abre la conexión con el TINI toma la trama y la envía. Del otro lado la clase *SocketListen* (que implementa el *SocketServer*) acepta el pedido de conexión estableciendo así el vínculo. El método *socketServer()* del *SocketListen* recibe los bytes e interpreta la trama invocando al método *leer()* de la clase *Cord* con el nombre de archivo *0xA0*.

Si la lectura fue exitosa se guarda el archivo en un array de bytes, y se invoca al método *ackLeer()* del *SocketListen* que arma la trama de respuesta de lectura poniendo en el payload el archivo leído.

#### TRAMA DE ACK DE LECTURA DE ARCHIVO

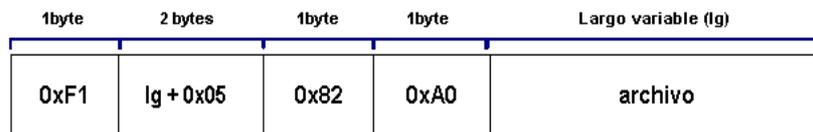


Figura 12. TRAMA DE ACK DE LECTURA

Ahora el *TiniComm* recibe la trama de respuesta con el archivo leído, el applet lee el archivo y lo despliega en pantalla (ver figura 21 ).

### 3.3.6 - CODIGO WML PARA WAP

Para acceder desde un celular se desarrollo un portal WAP, que consta de varias páginas WML y un archivo WMLSC con funciones útiles, como la autenticación de los usuarios.

Actualmente el portal WAP permite solamente la visualización del estado y de la configuración del sistema.

Breve reseña de WML:

WML es un lenguaje similar a HTML (ambos derivados de XML), tiene las mismas

funcionalidades y se agregan algunas otras específicas para el funcionamiento en el teléfono celular.

Una página WML consta de un encabezado, que siempre tiene que estar presente en las primeras líneas del código. Además consta de tags indicando el comienzo y fin del código WML ( lo que se encuentra dentro de estos tags es la página propiamente dicha ), y el cuerpo de la página.

*Ejemplo de una página WML:*

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card id="inicio" title="Menu inicio">

    <p><big><u><strong> Menu principal </strong></u></big><br/>

    <do type="accept" label="LOG SISTEMA">

    <go href="registro.wml#registro"/>

    </do>

    </p>

    .....

</card>

</wml>
```

Generalmente se dividen en *cartas* o “*cards*”, que representa cada pantalla a visualizar (al archivo WML completo se le denomina *baraja*), se puede hacer referencia a otras “*cards*” del mismo archivo o de otro archivo por medio del operador # y antecediendo el tag “< go href = ...>” es posible usar los hipervínculos de HTML para esta acción.

Cada archivo WML tiene que tener por lo menos una “*card*” para poder ser visualizado.

Para poner un botón para disparar un evento se utiliza el tag “< do type= y </do>”.

El otro tipo de archivo utilizado es WMLSC, que es un archivo de WMLScript compilado. Estos archivos le dan cierta “inteligencia” a la página WML y permite realizar algunas funciones utilizando el lenguaje proporcionado por WML.

WMLScript es una versión reducida de JavaScript, siendo limitadas las acciones y controles que pueden hacerse pero para poder realizar algunas operaciones básicas como la autenticación es suficiente.

*Ejemplo de un código WMLS:*

```

extern function login(){
    var nombre = WMLBrowser.getVar("usuario");
    var contra = WMLBrowser.getVar("password");
    if (nombre == "guest"){
        if(contra == "wap"){
            WMLBrowser.go("./wml/inicio.wml#inicio");
        }
    }
    else {
        WMLBrowser.go("index.wml#error");
    }
}

```

WMLS cuenta con librerías que se describen a continuación:

- x Lang
- x Float
- x String
- x Url
- x Dialogs
- x WMLBrowser

Utilizando estas librerías es que se pueden realizar acciones como leer y configurar variables leídas de la página, leer datos ingresados por el usuario, tomar decisiones según esos datos ingresados, etc.

Dentro del código WMLS también se pueden hacer llamadas a otros archivos y a determinados “cards” de un archivo WML, esto nos permite agregar un poco de lógica al portal WAP.

Todo esto corre directamente en el teléfono celular lo cual impone ciertas limitaciones. A pesar de ser un estándar cada teléfono tiene particularidades diferentes lo que hace un desafío realizar páginas WML visualizables en todas las marcas y modelos de teléfonos.

Para nuestro caso se trató de utilizar el código más sencillo y básico posible facilitando así el procesamiento en la mayoría de los celulares.

También es pertinente destacar que el pedido de la página WML se realiza a través de un servidor del proveedor de telefonía y estos tienen diferencias a la hora de solicitar las páginas y aplicaciones lo que obligó a adaptar el servidor para atender el mayor rango de solicitudes.

Las opciones disponibles desde el celular son:

- x Log del sistema, últimas 5 líneas
- x Visualización de datos de consumo, últimos 5 valores
- x Visualización de datos de temperatura, últimos 5 valores
- x Visualización del "perfil activo" del día
- x Visualización de la configuración de las notificaciones.

Toda la información vista a través del portal WAP es la misma que la vista a través del applet; cada vez que la información se modifica, los cambios se actualizan casi simultáneamente a través de procesos internos del sistema para que la información que se muestre sea la misma vista desde el portal WEB o desde el portal WAP.

### 3.3.7 - PROGRAMA PIC

Una vez instalado el PIC en la placa del C.O.R.D. y ser alimentado, éste se inicializa.

El programa principal es un loop infinito donde su única misión es resetear el watchdog. A través del watchdog timer se procura que el sistema no se vaya a un estado desconocido, entonces si dicho contador no es puesto a cero periódicamente el PIC se resetea.

Cuando el TINI (que es el maestro en la comunicación I<sup>2</sup>C) solicita ejecutar una acción la rutina de atención a interrupciones ejecuta dicha acción y luego devuelve el control al programa principal.

Las acciones a ejecutar por el PIC son:

#### - ESCRITURA

Se traduce en acciones sobre la bornera de relés.

Cuando se le ordena al PIC escribir en una de sus patas depende el valor que se escriba será el efecto sobre el circuito de la vivienda.

Si se manda un comando "*Encendido digital x*", el PIC levanta 5 voltios en la pata asociada a ese "digital" lo que es suficiente para accionar el relé que tiene asociado en la bornera.

Si se manda un comando "*Apagado digital x*", el PIC baja la pata a tierra del "digital" lo que desmagnetiza el relé conectado.

#### - LECTURA

En esta ocasión sólo se va a utilizar para leer el consumo eléctrico del hogar.

Primero con el comando "*Leer consumo*", se le dice al PIC en que pata leer. Éste lee el dato en esa pata y lo deja en un buffer de salida, que el TINI después lee y traduce en el valor de potencia consumida.

El valor que es leído en la pata correspondiente es entregado por una bobina de corriente que

hace de transductor para llevar los valores de potencia reales a un rango que el PIC puede leer.

Se procesa una instrucción a la vez, y durante este proceso se deshabilitan las interrupciones del PIC. Por esto durante la atención de interrupciones no es posible establecer comunicación con el mismo y será necesario separar tareas que envíen comandos I<sup>2</sup>C seguidos dando tiempo para terminar la orden en proceso. Fallará cualquier intento de comunicación que se produzca antes de retornar al estado normal.

### 3.3.8 - ARCHIVOS DE CONFIGURACION DEL SISTEMA

El sistema utiliza diferentes archivos de configuración que son los siguientes:

- *registro.log* - Es el archivo donde se escriben todos los eventos del sistema.
- *consumo.txt* - Guarda los datos de consumo leídos por el transductor.
- *temperatura.txt* - Guarda los datos de temperatura leídos por el sensor.
- *agenda.cfg* - Contiene la agenda de perfiles que el sistema debe asignar a la "*agenda diaria*" cada día.
- *dis2dig.txt* - Contiene la correspondencia entre los puertos digitales y 1-Wire instalados en la casa con los dispositivos creados por el usuario en la interfaz gráfica.
- *notifica.cfg* - Contiene la dirección de e-mail y el celular al cual se va a notificar así como también el nivel de notificaciones.
- *perfilX.prf* - En estos archivos está la información de la casa bajo una estructura de árbol, con las habitaciones y los dispositivos. Estos últimos con las horas y acciones que se configuraron para dicho perfil.

## 4 - INSTALACION

### 4.1 - INSTALACION FISICA

Los requerimientos para la instalación del sistema C.O.R.D. en la casa del usuario son mínimos.

#### *PLACA Y BORNERA DE RELES*

La placa del sistema se recomienda instalarla cerca del tablero general, ésta cuenta con una bornera con nueve polos para conectar tres dispositivos digitales ( tres polos por dispositivo a

conectar: Común , NA y NC) . Estos dispositivos son activados por medio de un relé auxiliar, el cual es inversor. Por este motivo el dispositivo puede ser conectado en 2 configuraciones como se ve en la fig. 13.

**Configuración Normal Abierta (NA):** El dispositivo esta sin alimentación mientras la bobina del relé correspondiente de la placa también esta sin alimentación. Cuando la bobina es alimentada (por la acción del PIC), se activa el contacto de salida del relé que permite cerrar el circuito del dispositivo. En esta configuración se conecta el Borne Común a un polo de la alimentación de la vivienda, el borne NA a un polo del dispositivo y el otro polo del dispositivo al polo restante de la alimentación según muestra el diagrama superior de la fig.13.

**Configuración Normal Cerrada (NC):** El dispositivo esta con alimentación mientras la bobina del relé correspondiente de la placa está sin alimentación. Cuando la bobina es alimentada (por la acción del PIC), se desactiva el contacto de salida del relé que permite abrir el circuito del dispositivo. En esta configuración se conecta el Borne Común a un polo de la alimentación de la vivienda, el borne NC a un polo del dispositivo y el otro polo del dispositivo al polo restante de alimentación según muestra el diagrama inferior de la fig.13.

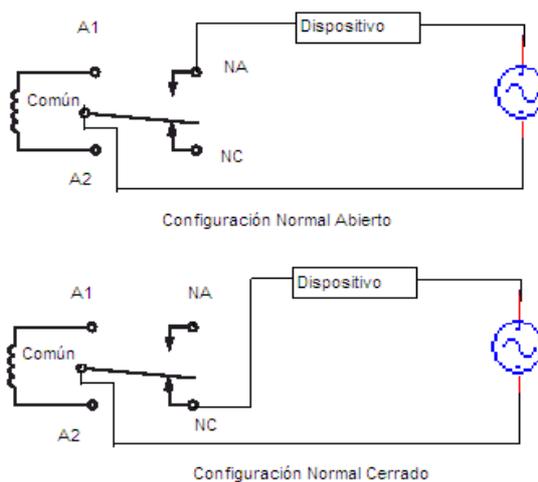


Figura 13. CONEXION RELE - DISPOSITIVO

El lugar en la bornera de las conexiones de los dispositivos digitales son los que definen el vínculo entre los dispositivos y su asignación en la numeración que tiene el software de los dispositivos que maneja. Debido a esto es importante registrar en un plano que dispositivos están conectados a cada polo de la bornera.

### TRANSDUCTOR DE CONSUMO

Para medir consumo de potencia activa se debe conectar a C.O.R.D. un transductor de corriente que posea una salida análogica en corriente 4-20mA con alimentación autónoma. El equipo con el que se realizaron las pruebas de funcionamiento es el modelo 8526610000 de la marca Weidmuller ( se adjunta ficha técnica en Apendice) que consta de un toroide por el cual pasa el conductor en el cual se quiere medir la corriente circulante. El equipo tiene un rango configurable de hasta 10A de entrada y la salida análogica proporcional a la corriente de entrada en el standard 4-20mA ( 4mA = 0% del rango de entrada, 20mA = 100% del rango de entrada).

Esta salida analógica en 4-20mA es acondicionada en la entrada a la plaqueta para que el conversor A/D del PIC pueda procesarla.

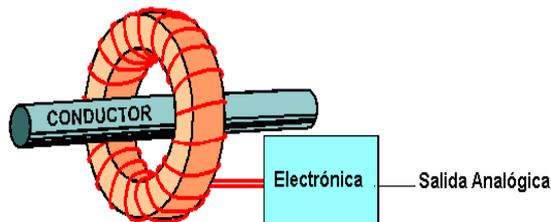


Figura 14. ESQUEMA TRANSDUCTOR DE CORRIENTE

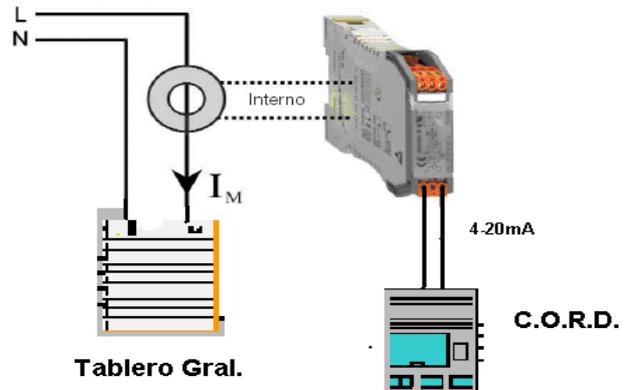


Figura 15. CONEXIONADO TRANSDUCTOR

Teniendo el consumo de corriente entregado por el transductor , una tensión de alimentación de 220VAC rms y asumiendo un factor de potencia de 0,92 el sistema calcula y gráfica la potencia activa consumida en la vivienda, en una habitación ó un dispositivo, según en que punto de medición se conecte el transductor.

La salida de corriente del transductor se conecta al conector J4 de la placa según la tabla.

Conector Transductor	Conector interior
	<b>J4</b>
Salida 4-20mA (+)	1
Salida 4-20mA (-)	2

Tabla 11: PINOUT TRANSDUCTOR

Es importante tener en cuenta que el lazo de corriente de 4-20mA debe ser alimentado desde el transductor, la entrada de C.O.R.D. es pasiva.

### SENSOR DE TEMPERATURA

El sensor de temperatura se instala en la zona que se quiere medir la temperatura teniendo, si fuera necesario, que realizar una conexión con un conector de cobre.

### CONEXION A INTERNET

Para la conectividad es necesario conectar el puerto ethernet de la placa con la conexión dedicada a Internet.

## 4.2 - INSTALACION DE SOFTWARE

A continuación se describen los requerimientos de software y hardware para instalar el sistema.

### 4.2.1 - REQUERIMIENTOS - COMPUTADOR PERSONAL

Estas recomendaciones se basan en pruebas realizadas con diferentes equipos y las especificaciones dadas son necesarias para una buena performance de trabajo.

#### *HARDWARE*

- x Procesador: Intel Pentium III o similar de 500 Mhz
- x Memoria: 128 MB de RAM
- x Tarjeta de red 10Base-T (mínimo)
- x Conexión a Internet dedicada (ADSL o similar) con una velocidad de bajada o downstream superior a 128 Kbps (ver nota adjunta)

Nota: Se realizaron pruebas de acceso a través por módem de 56 Kbps por la línea telefónica, pero las demoras son bastante superiores.

También se realizaron pruebas con un acceso a Internet inalámbrico con IP dinámica pedida a través de DHCP para lo que el TINI ya tiene implementado un cliente.

Si se quisiera conectar en una conexión ADSL con IP dinámica se necesitaría desarrollar un cliente PPPoE para el TINI.

#### *SOFTWARE*

- x Sistema Operativo (ver nota adjunta)
- x Navegador Web: Internet Explorer, Mozilla Firefox, Galeón, etc.
- x Plugin de Java: Es necesario tener el plugin (JRE) instalado en el navegador para poder visualizar la aplicación. Este software se puede descargar de la página de Sun Microsystems.

Nota: Dado que la programación se hizo en lenguaje Java el único requerimiento es que exista el plugin de Java (con una JVM) para el sistema operativo utilizado.

Si bien las pruebas se hicieron con el software y hardware antes dicho, no quita que no pueda usarse con menos requerimientos (no es recomendable), teniendo en cuenta que la performance disminuye notoriamente.

#### **4.2.2 - REQUERIMIENTOS - TELEFONO CELULAR**

Para poder acceder desde una red celular es necesario:

- x Tener un terminal con capacidad para navegar en páginas WAP  
( muchos de los modelos de teléfonos que hay en plaza lo tienen ).
  
- x Tener acceso a Internet desde el teléfono, para lo cual debe contactarse con el proveedor correspondiente ( las empresas de telefonía celular que actualmente residen en el país brindar este servicio).

## 5 - MODO DE OPERACION

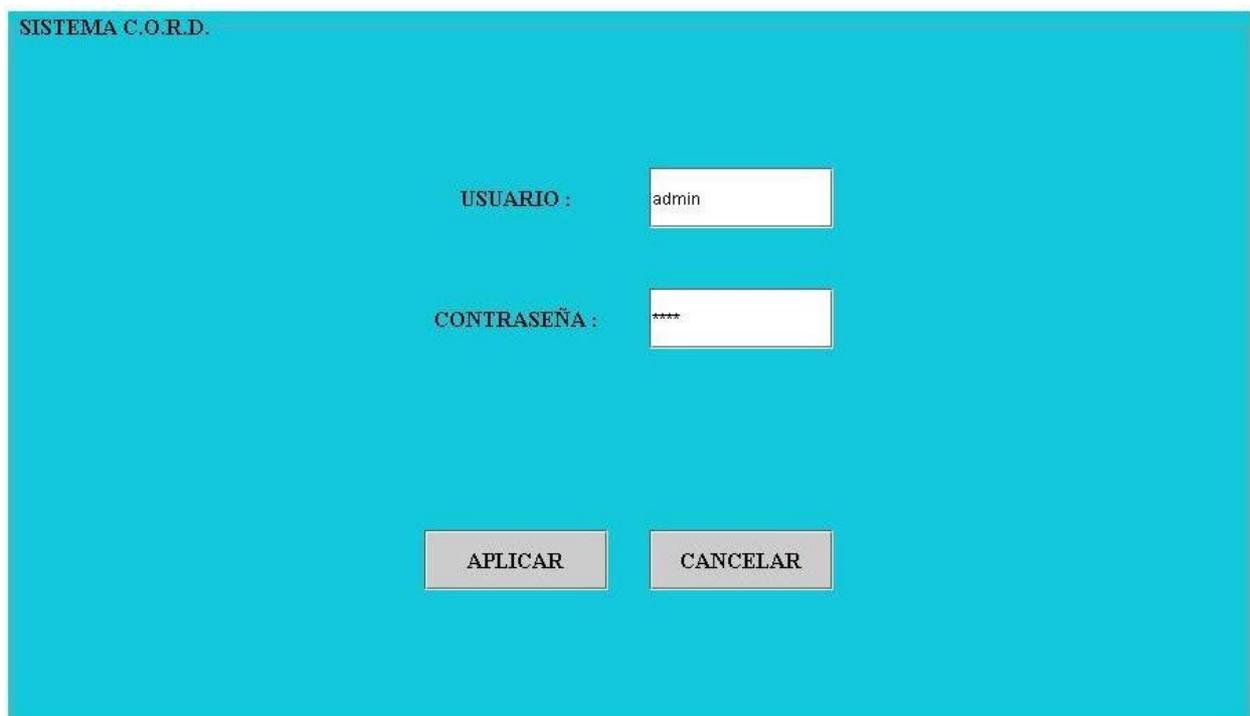
### 5.1 - INTERFAZ GRAFICA - WEB

#### ACCESO AL SISTEMA (INICIO DE SESION)

La pantalla de inicio de sesión es la primer pantalla que se observa al iniciar el applet y permite a los usuarios ingresar al programa principal desde donde se interactua con el resto del sistema.

Los usuarios son: *admin* y *guest*, el acceso con uno u otro determina los privilegios para operar el sistema.

El usuario *admin* tiene derechos de administración total, mientras el usuario *guest* tiene únicamente derechos de visualización.



The image shows a login interface for 'SISTEMA C.O.R.D.' on a blue background. It features two input fields: 'USUARIO' with the text 'admin' and 'CONTRASEÑA' with four asterisks '\*\*\*\*'. Below these fields are two buttons: 'APLICAR' and 'CANCELAR'.

Figura 16. PANTALLA DE INICIO DE SESION

## PANTALLA PRINCIPAL Y BARRA DE MENU

En la pantalla principal tenemos una barra donde se muestra una lista de los menús disponibles en el sistema, así como una etiqueta con el usuario conectado y un botón LOGOUT para finalizar la sesión actual.

Los menús disponibles son:

- x MENU DE ACCIONES
- x MENU DE CONFIGURACION
- x MENU MOSTRAR
- x MENU AYUDA

## MENU DE ACCIONES



*Figura 17. MENU ACCIONES*

### *INICIAR / DETENER AGENDA DIARIA*

Este ítem permite iniciar y detener la agenda diaria. Además dependiendo de la situación en la que esta la agenda el ítem indica la acción que es posible ejecutar (si ya esta iniciada el ítem indica “Detener agenda diaria” y si esta detenida el ítem indica “Iniciar agenda diaria”).

Un ejemplo donde podría utilizarse esta acción es cuando el perfil activo es modificado, para ejecutar las nuevas tareas sera necesario detener la agenda e iniciarla nuevamente.

En caso de intentar iniciar la agenda un día donde el sistema no tiene perfil asociado, el sistema lo notifica con un mensaje en pantalla.

### *INICIAR / DETENER MEDIDA DE CONSUMO*

Es similar al ítem anterior, solo que para la acción de medida de consumo. El ítem también indica la acción que puede realizarse sobre la medida de consumo (iniciarla si esta detenida o viceversa).

*EJECUTAR COMANDO DIRECTO*

Esta pantalla muestra una lista de todos los dispositivos definidos (independiente de los perfiles en que fueron declarados). Se puede seleccionar cualquiera de ellos y ejecutar acciones sobre los mismos utilizando los botones que aparecen en pantalla. El sistema da aviso del resultado de la tarea, ya sea que se ejecuto exitosamente o no. No se contempla la situación actual del dispositivo por lo que podría seleccionarse una acción de encendido siendo que el dispositivo ya estuviera encendido, esto no produce ningún efecto contraproducente sobre el sistema.

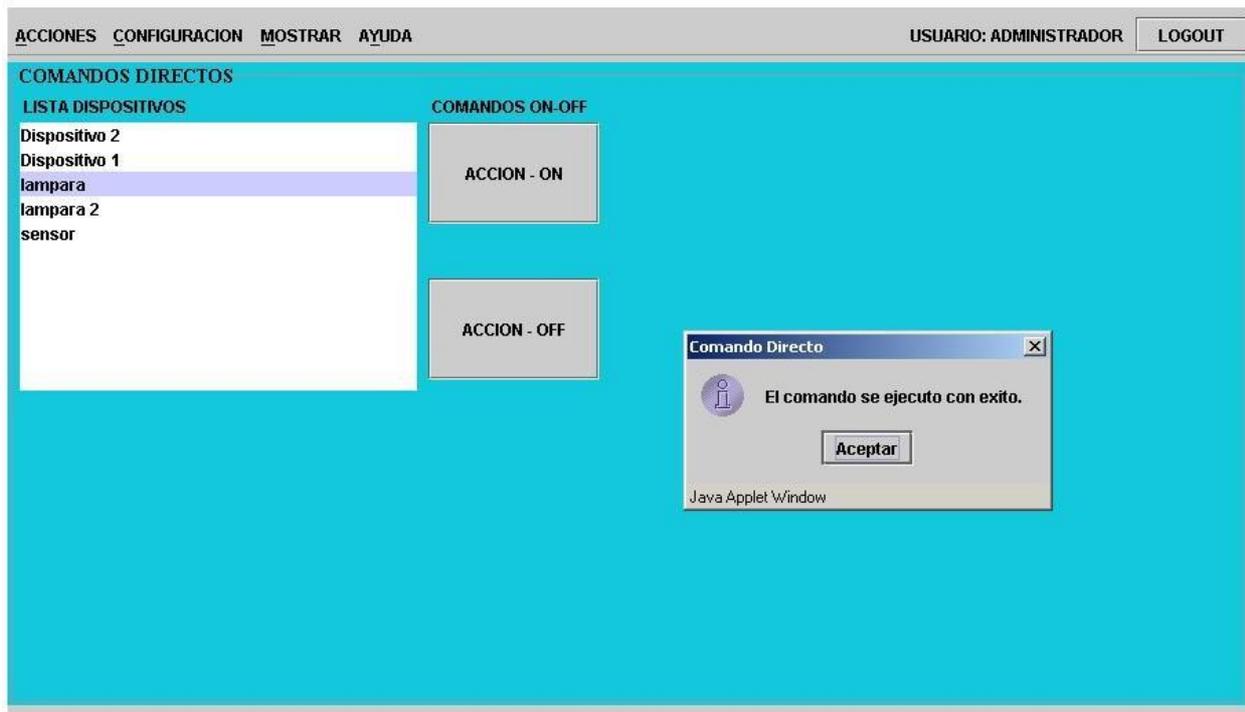


Figura 18. PANTALLA DE COMANDOS DIRECTOS

**MENU CONFIGURACIONES**



Figura 19. MENU CONFIGURACION

*AGENDA*

Esta pantalla permite crear la agenda de perfiles que el sistema deberá iniciar cada día.

Permite ingresar períodos de uno o mas días con el perfil seleccionado.

Además controla que no halla incoherencias entre las opciones ingresadas.

*Ejemplo:*

Si se define un intervalo del **"01/Marzo 01/Abril – perfil01"**, no se puede agregar otro intervalo que se superponga con el anterior con otro perfil diferente como ser **"10/Marzo 15/Marzo – perfil04"**(el segundo intervalo esta contenido en el primero y tienen perfiles diferentes).

También une períodos consecutivos si el perfil es el mismo.

*Ejemplo:*

Si se define un intervalo del **"01/Marzo 15/Marzo – perfil01"**, y se ingresa otro intervalo como ser **"10/Marzo 28/Marzo – perfil01"** (el segundo intervalo se superpone con el primero y lo extiende con el mismo perfil), entonces se concatenan los mismos y se define uno solo de la siguiente manera **"01/Marzo 28/Marzo – perfil01"**.

Se muestra el comentario de los perfiles existentes en el sistema a modo de ayuda para el usuario.

La pantalla se inicia con la lista actual de períodos definidos en el sistema, que son leídos de un archivo de configuración *"agenda.cfg"*. Cuando se oprime el botón de finalizar se guarda en dicho archivo la nueva lista.

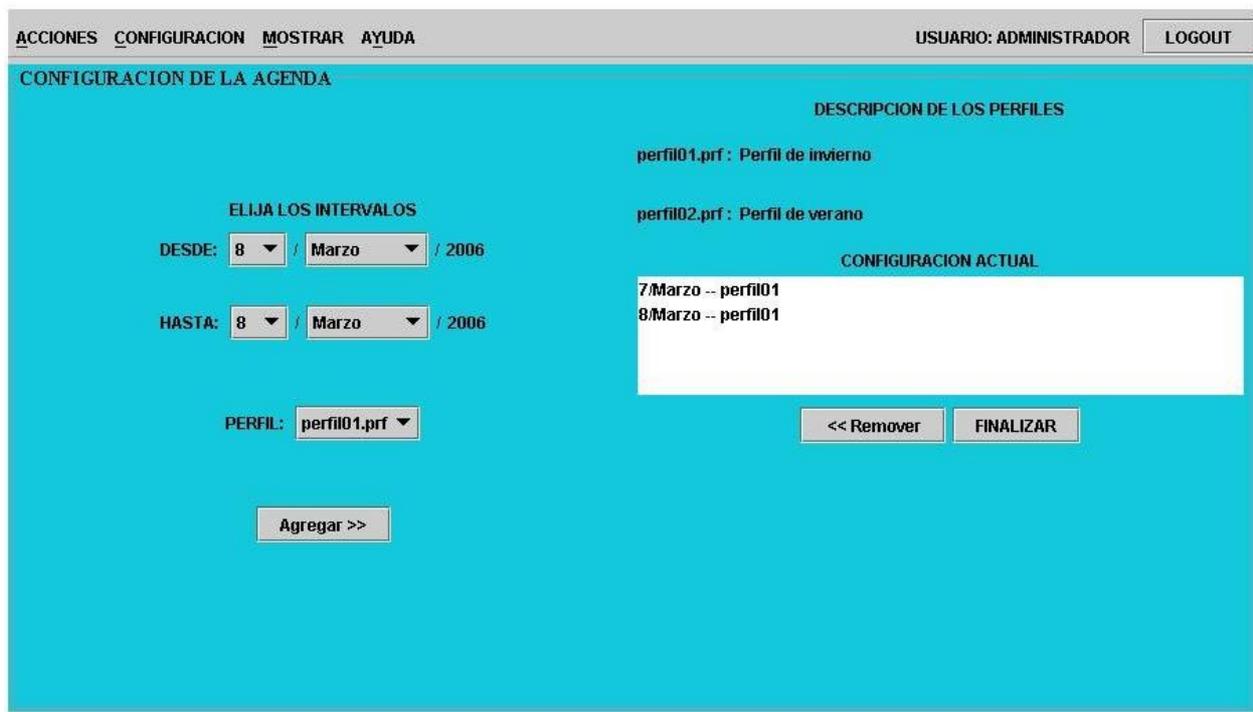


Figura 20. PANTALLA DE AGENDA GENERAL

## PERFILES

Se administran los dispositivos del sistema y perfiles de funcionamiento.

ACCIONES CONFIGURACION MOSTRAR AYUDA USUARIO: ADMINISTRADOR LOGOUT

Arbol => Agregar Remove

Propiedades de Dispositivo 1

Puerto Asociado: digital02

Hora del Día	Minutos	Acción
11	11	on
13	09	off
15	17	on
18	04	off

Nueva Línea Insertar Línea Borrar Líneas Aplicar

EDITANDO: perfil01.gdf COMENTARIO: Perfil de invierno

Nuevo Perfil Editar otro Perfil Guardar Perfil Borrar Perfil

Figura 21. PANTALLA DE CONFIGURACION DE PERFILES

Por defecto nos muestra el "Perfil Activo" del día (si ese día tiene perfil asignado). De un lado tenemos la descripción gráfica del perfil en un árbol de habitaciones y dispositivos que es completamente modificable.

Al seleccionar la raíz del árbol (Casa) se puede agregar o borrar habitaciones con sus dispositivos. Al seleccionar una habitación se pueden agregar o borrar dispositivos para esa habitación. Seleccionando un dispositivo se accede a su pantalla de configuración en donde se pueden ingresar y/o modificar las acciones y horas para el comportamiento deseado del mismo. Se controla además que no se ingresen horas incompatibles o incompletas.

Se pueden editar perfiles o realizar nuevos. Para cada perfil se puede ver una breve descripción del mismo si así fue ingresado por el usuario al momento de crearlo. La idea es que el usuario tenga una referencia del perfil que esta modificando. Solo cuando se de la opción de guardar, el archivo del perfil editado será guardado por el sistema y quedará pronto para usar.

### CONFIGURACION DE RED

En esta pantalla se pueden ver los principales parámetros de red que actualmente tiene configurado el TINI, nombre del host, dirección IP, máscara de subred y puerta de enlace.

The screenshot shows a web interface for network configuration. At the top, there is a navigation bar with links for 'ACCIONES', 'CONFIGURACION', 'MOSTRAR', and 'AYUDA'. On the right side of the navigation bar, it displays 'USUARIO: ADMINISTRADOR' and a 'LOGOUT' button. Below the navigation bar, the main content area is titled 'PARAMETROS DE RED'. It contains four rows of configuration fields: 'HOSTNAME' with the value 'cord', 'DIRECCION IP' with the values '200', '108', '221', and '241', 'MASCARA DE RED' with the values '255', '255', '255', and '192', and 'PUERTA DE ENLACE' with the values '200', '108', '221', and '193'. At the bottom of the form, there are two buttons: 'APLICAR' and 'CANCELAR'.

Figura 22. PANTALLA DE CONFIGURACION DE RED

En el caso de haber ingresado al sistema como usuario *admin* cualquiera de estos parámetros puede ser modificado. El programa controla que se ingresen valores válidos.

Para guardar la nueva configuración se oprime el botón "APLICAR" y los parámetros son modificados en el momento. Es importante tener en cuenta que al momento de guardarse los nuevos parámetros se pierde automáticamente la conexión con el sistema y habrá que conectarse con la nueva dirección IP que recientemente se asignó al sistema.

El botón "CANCELAR" solo deja los campos en blanco.

### CONFIGURACION DE NOTIFICACIONES

El sistema tiene la posibilidad de enviar mensajes al usuario para avisar de diferentes eventos y es a través de esta pantalla que se ingresan las direcciones para tales notificaciones.

Ingresando como *admin* estos datos pueden modificarse.

ACCIONES CONFIGURACION MOSTRAR AYUDA USUARIO: ADMINISTRADOR LOGOUT

NOTIFICACIONES DEL SISTEMA

E-MAIL TINI : bord@tini.com

NIVEL NOTIFICACIONES : NIVEL CRITICO

CELULAR NOTIFICACIONES : 099213377

E-MAIL NOFITICACIONES : fdrodri@adinet.com.uy

APLICAR CANCELAR

Figura 23 . PANTALLA DE CONFIGURACION DE NOTIFICACIONES

En la parte superior de la pantalla se observa la dirección con la que llegan los mensajes. A continuación de debe seleccionar el nivel de notificación que se desea recibir (INFORMATIVO o CRITICO). Se puede ingresar un número de teléfono celular (con su código si es necesario) y/o una dirección de correo electrónico donde se pretende recibir las notificaciones.

Si bien el sistema no obliga a ingresar ninguna dirección para recibir avisos, siempre es conveniente tener configurada alguna, por lo menos con un nivel crítico para estar al tanto del funcionamiento del sistema.

El sistema también verifica que el número celular, el código y las direcciones de e-mail ingresadas sean válidas.

## MENU MOSTRAR



Figura 24. MENU MOSTRAR

### REGISTRO DEL SISTEMA

Permite ver el archivo de log del sistema, donde se van registrando los eventos trascendentes del mismo. A cada evento que se escribe en este archivo se le adjunta su correspondiente "marca" de fecha y hora.

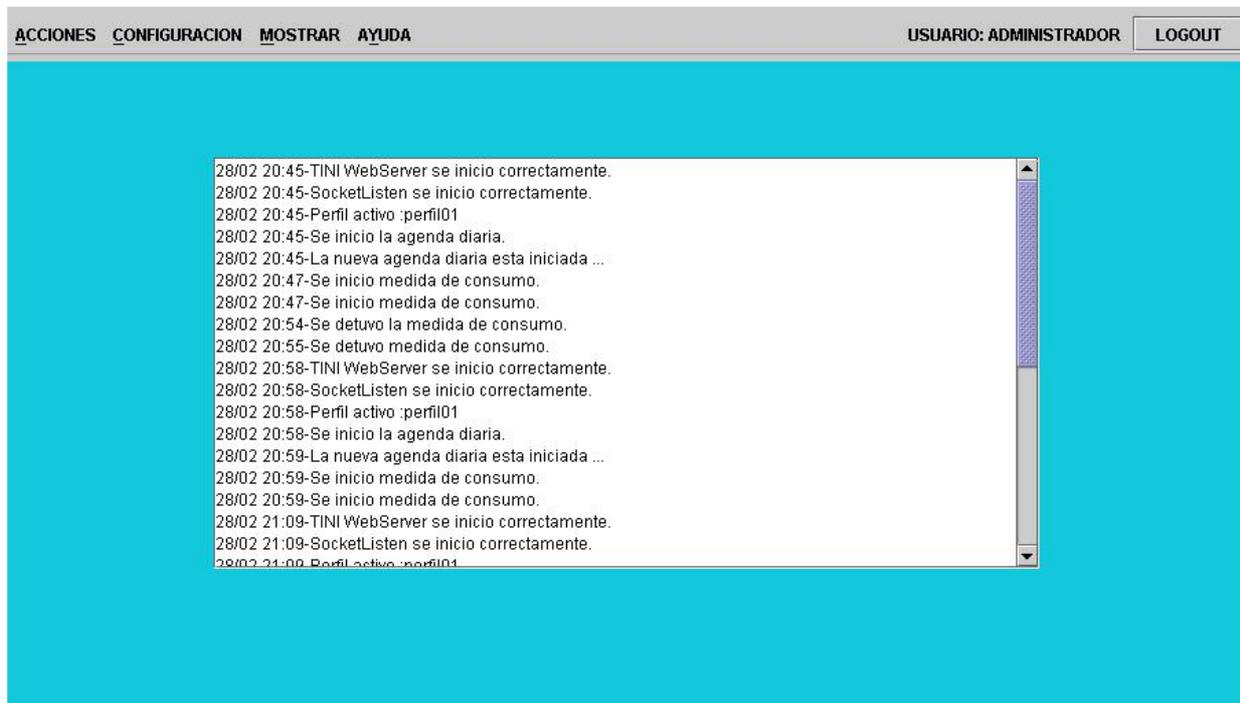


Figura 25. PANTALLA DE LOG DEL SISTEMA

*REGISTRO DE CONSUMO*

Permite ver una gráfica del consumo de potencia activa.



Figura 26. PANTALLA DE GRAFICA DE CONSUMO

*REGISTRO DE TEMPERTURA*

Permite ver una gráfica de la temperatura medida por el sensor.

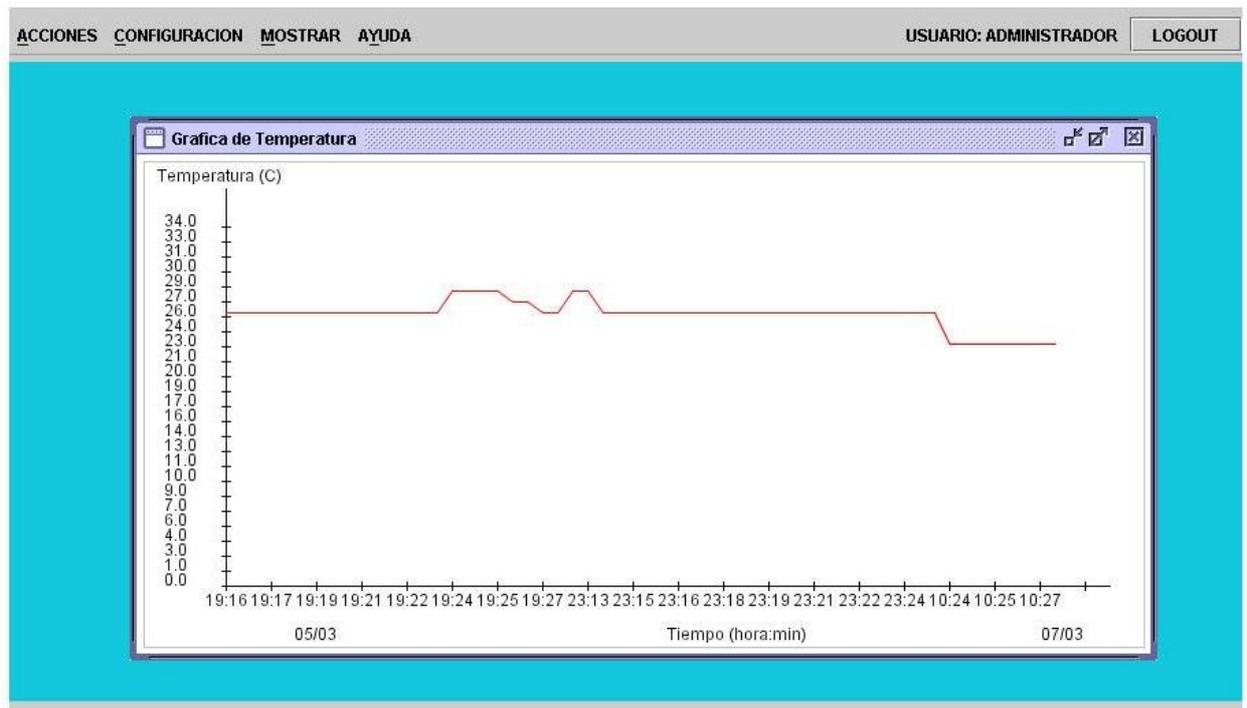


Figura 27. PANTALLA DE GRAFICA DE TEMPERATURA

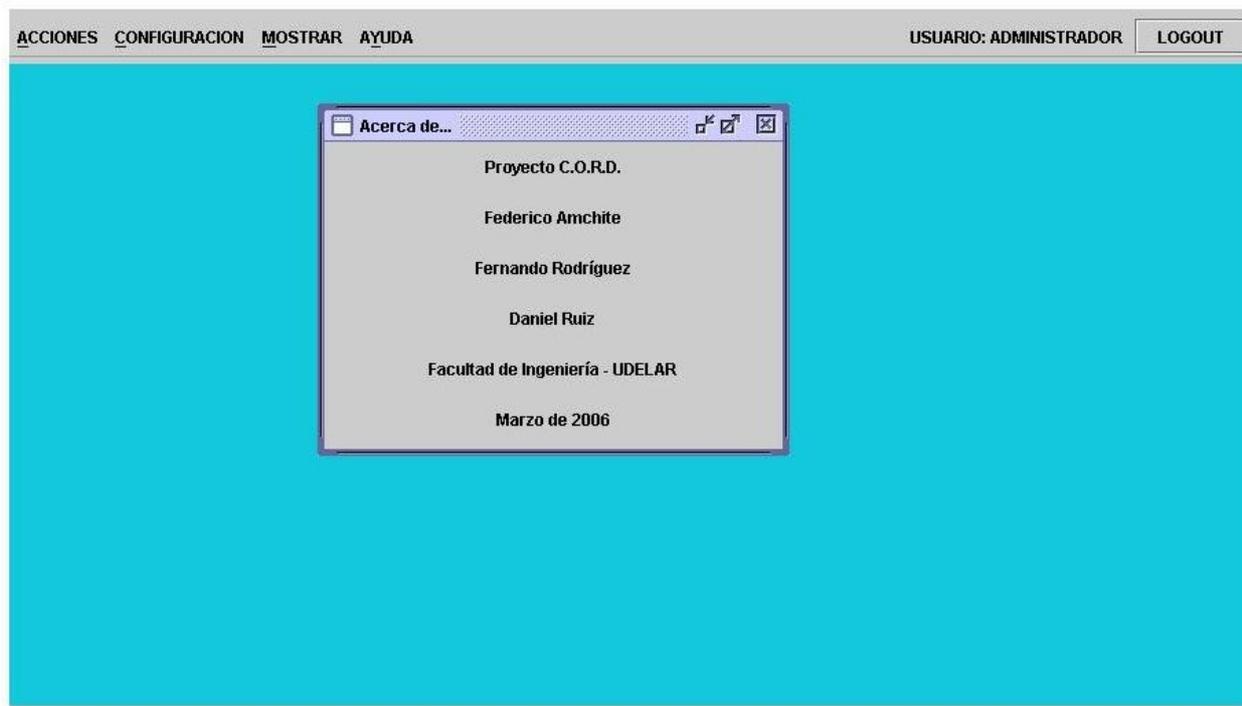
**MENU AYUDA**



**Figura 28. MENU AYUDA**

*ACERCA DE ...*

Acerca de este proyecto, muestra los datos de los estudiantes, nombre del proyecto y fecha de culminación.



**Figura 29. PANTALLA DE ACERCA DE ...**

## 5.2 - INTERFAZ GRAFICA – WAP

Al acceder desde un teléfono celular la interfaz gráfica del sistema cambia sustancialmente.

Como primera pantalla tenemos la de acceso al sistema, el usuario es *guest* y contraseña *wap*.

En caso de verificar los datos de usuario, se accede a una pagina de menú principal que presenta las opciones disponibles en el sistema.

Si no se verifican los datos de usuario muestra un mensaje de error.

### INICIO DE SESION

A continuación se presentan la página del menú principal y luego las demás páginas a las que se permite acceder.



Figura 30. LOGIN

### MENU PRINCIPAL

#### *LOG SISTEMA*

Permite ver las últimas cinco líneas del log del sistema, cada vez que se agrega un línea al log se actualiza el archivo WML correspondiente.



Figura 31. LOG SISTEMA

#### *LOG CONSUMO*

Permite ver una tabla con los últimos cinco valores del consumo eléctrico de la casa, este archivo también se actualiza cuando el archivo de consumo del sistema es modificado.



Figura 32. LOG CONSUMO

*PERFIL ACTIVO*

Permite ver el "Perfil Activo" del día, viendo sus dispositivos con las configuraciones de horarios y acciones correspondientes.



Figura 33. PERFIL ACTIVO

*NOTIFICACIONES*

Permite ver la configuración de notificaciones de los eventos del sistema. También se actualiza cuando el archivo correspondiente de configuración se modifica.



Figura 34. NOTIFICACIONES

*ACERCA DE ...*

Acerca de este proyecto ...



Figura 35. ACERCA DE ...

## 6 - OBSERVACIONES

### 6.1 - MEJORAS EN HARDWARE

1. Para la instalación se podría realizar la comunicación con los componentes por medio de RF, la línea de 220V usando protocolo X-10, bus 1-wire o algún otro bus conocido.

Para eso sería necesario actualizar el sistema con los módulos correspondientes de hardware y software. Estas opciones sustituirían la bornera de relés por el módulo de hardware correspondiente.

A la hora de utilizarlo en una casa real cualquiera de estas soluciones son buenas y facilitarían la instalación sin tener que preveer una pre-instalación domótica.

2. Una fuente switcheada mejoraría el rendimiento y también reduciría el tamaño de la placa de fuente.
3. Sistemas de monitoreo de acciones o realimentación, para verificar que al ejecutar una acción sobre un dispositivo pueda verificarse el estado del mismo . Se podría configurar otro puerto del PIC, con entradas analógicas para testear señales eléctricas.

Esto también implica agregar el software necesario con un proceso que implemente la realimentación leyendo del puerto del PIC y tome las decisiones correspondientes.

### 6.2 - MEJORAS EN SOFTWARE

1. La implementación de otros clientes para la conexión de Internet ( PPPOE principalmente ) sería una mejora en la adaptabilidad del sistema a distintos entornos con proveedores de Internet en diferentes tecnologías.
2. A nivel de WAP, implementar la ejecución de acciones sobre el sistema ( iniciar o detener agenda, comandos directos, etc. ) ya que actualmente solo permite la supervisión del sistema.
3. Todo el manejo de archivos de perfiles del sistema, su vinculación los dispositivos reales con la complejidad de lógica de software que conlleva podría simplificarse agregando un motor de base de datos que trabaje al respecto.
4. Permitir acciones sobre dispositivos a partir de estados y/o mediciones de otros. Por ejemplo controlar la temperatura de una habitación en función del sensor de temperatura y las acciones ON/OFF que se realicen sobre la calefacción.
5. Otra mejora sería realizar perfiles por dispositivo (actualmente son perfiles diarios del sistema) o sea la administración de dispositivos en el tiempo. Esto haría el manejo de la "Agenda Diaria" diferente, se configuraría el comportamiento de un dispositivo en el día y en la agenda se agregaría el dispositivo los días en el que se quiere tener ese perfil de comportamiento.

Con la implementación al día de hoy para realizar esto es necesario agregar al dispositivo en el perfil del día marcado en la agenda. Si bien con perfiles de dispositivos sería más flexible para el usuario, el código para tener este comportamiento es más complejo.

6. Con respecto a las notificaciones, la resolución de los servidores SMTP de cada dominio de mail se realiza por programa y solamente para los dominios de correo más usados (a nuestro criterio). Esto se podría mejorar, es decir, permitir al usuario ingresar cualquier casilla de correo válida, e implementar una rutina de resolución de nombres de dominios. El TINI contiene un comando en el shell para realizar consultas de DNS "*nslookup*", se podría realizar una consulta (de registro MX) por el dominio de la casilla y leer la salida del comando obteniendo el nombre del servidor SMTP correspondiente.
7. Los nombres de los archivos donde se guardan los perfiles *perfil01.prf*, *perfil02.prf*, etc. son fijos, dado que se implementa la comunicación a través de un protocolo de entramado de bytes se fijo los nombres de los archivos para la fijar la codificación. Otra mejora es permitir nombres de archivos de perfil variables (a elección del usuario), lo que significa modificar el software para realizar este control.
8. En las gráficas, se mejoraría la visualización incluyendo un campo de ingreso de los límites superiores del eje Y. Esto oficiaría en cierta manera como un "zoom", ya que se podría ver una rango determinado.

## 7 - CONCLUSIONES

- Consideramos que se llegó a los objetivos planteados en el comienzo de la nueva etapa del proyecto, en la cual nos definimos por realizar un sistema domótico. El proyecto original consistía en el control de antenas de TV y radio via GPRS. En diciembre del 2004 se agotaron las posibilidades de conseguir en préstamo el módem GPRS con Ancel. Debido a esto se planteó realizar el control a través de una red ethernet trabajando sobre el transmisor de canal 10. En febrero del 2005 por razones de fuerza mayor canal 10 decidió que no se realizaran pruebas sobre su transmisor y nos vimos obligados a cambiar la idea de los dispositivos a controlar. Ahí comenzamos a madurar el nuevo sistema para el proyecto.
- Tomando en cuenta que teníamos una menor disponibilidad de tiempo y que el sistema no iba a ser utilizado sobre el control ó monitoreo de un equipo específico, se decidió invertir más tiempo en poder desarrollar un Software que pudiera ser más completo ( manejo de dispositivos 1-wire, señales analógicas, acceso vía web, acceso vía celular) y realizar un Hardware más acotado que tuviera como fin poder mostrar el funcionamiento de las distintas funcionalidades del software.

A nivel de Hardware se realizó una placa sencilla pero completa y escalable que permite probar y utilizar las funcionalidades del sistema.

A nivel global se cumplió con los requerimientos planteados de controlar dispositivos ON/OFF, tener puertos analógicos para medición e implementar el uso de dispositivos 1-wire (como el sensor usado para medir temperatura). También se logró el acceso al sistema a través de Internet y de un equipo móvil (celular).

- Consideramos la experiencia muy buena como proyecto de ingeniería. El sistema puede verse como un producto "cerrado" (se parte de una idea, se investigan y plantean soluciones, se limita el alcance y se llega a un sistema en hardware y software funcionando). Esto nos permitió trabajar y desarrollar varias áreas de la ingeniería, lo que fue motivante. También el poder haber resuelto distintas situaciones, que se fueron presentando, agotando todas las opciones de estudio e investigación dentro del grupo de proyecto (los tres integrantes) nos permitió aplicar conocimientos adquiridos a lo largo de la carrera. Valoramos el hecho de haber resuelto enteramente el proyecto por nuestro propios medios.
- Como proyecto en sí nos dejó varias enseñanzas ya que el trabajo en grupo además del crecimiento intelectual conlleva otros aspectos como el intercambio de ideas y opiniones, discusiones, asignación de tareas y mucho tiempo compartido, que hace a la parte humana de cada uno. También aprendimos que dentro de un proyecto de ingeniería no hay que subestimar el tiempo que se debe invertir en cuestiones no estrictamente técnicas como son: logística, coordinación de tiempos y economía.

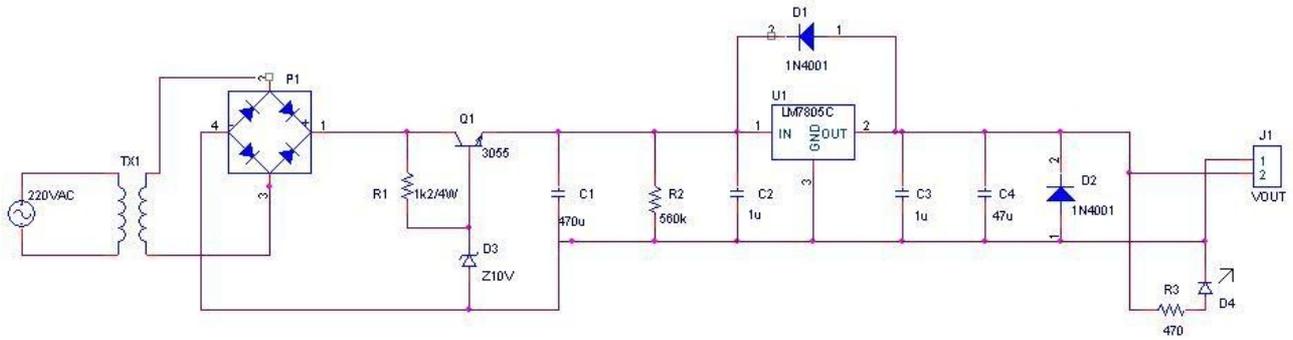
## 8 - REFERENCIAS Y BIBLIOGRAFIA

- [1] **"Página oficial DALLAS SEMICONDUCTOR - TINI"**;  
<http://www.maxim-ic.com/TINIplatform.cfm>
- [2] **"Getting Started with TINI"**;  
[http://www.maxim-ic.com/products/tini/pdfs/TINI\\_GUIDE.pdf](http://www.maxim-ic.com/products/tini/pdfs/TINI_GUIDE.pdf)
- [3] **"The TINI Specification and Developer's Guide"**;  
<http://www.maxim-ic.com/products/tini/pdfs/tinispec.pdf>
- [4] **"TINI API Documentation"**;  
<http://files.dalsemi.com/tini/javadocs/firmware/index.html>
- [5] **"Foro de DALLAS SEMICONDUCTOR"**;  
<http://discuss.dalsemi.com>
- [6] **"Application Note 198: Networked Temperature Monitoring"**;  
<http://www.maxim-ic.com/appnotes10.cfm/>
- [7] **"Proyecto STAS - nfxr – UDELAR (2003)"**;  
Dario Parente, Adrián Rancaño, Federico Viñar
- [8] **"Página oficial MICROCHIP – PIC16F877A "**;  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1335&dDocName=en010242](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010242)

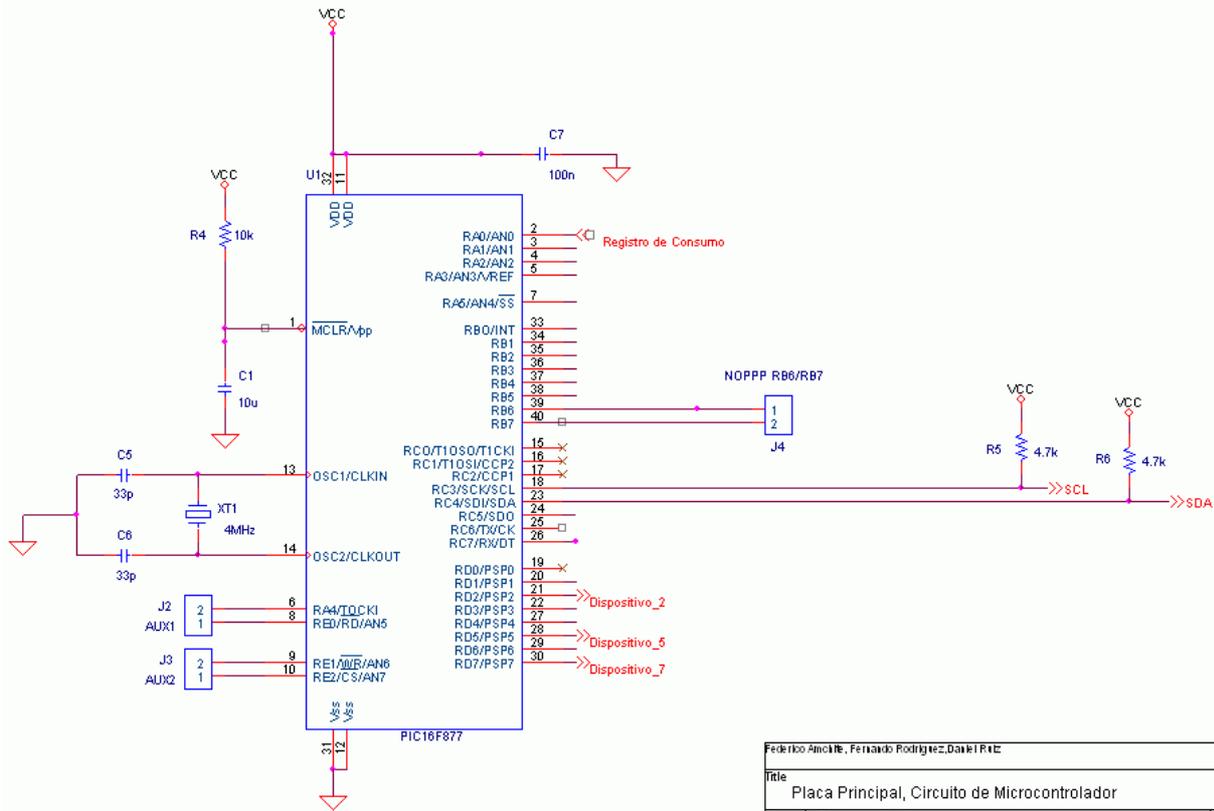
- [9] **"PIC16F87XA Data Sheet"**;  
<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>
- [10] **"MPLAB® IDE User's Guide "**;  
<http://ww1.microchip.com/downloads/en/DeviceDoc/51519a.pdf>
- [11] **"Circuitos Microelectrónicos"**;  
 Sedra – Smith
- [12] **"Página oficial de Java (Sun Microsystems)"**;  
<http://www.java.com/>
- [13] **"The Java Tutorial (Sun Microsystems)"**;  
<http://java.sun.com/docs/books/tutorial>
- [14] **"Java™ 2 Platform, Standard Edition, v 1.4.1 API Specification"**;  
<http://java.sun.com/j2se/1.4.1/docs/api/index.html>
- [15] **"Thinking in Java, 2<sup>nd</sup> edition"**;  
 Bruce Eckel
- [16] **"JCreator Xinox Software"**;  
<http://www.jcreator.com/>
- [17] **"Foro de Nokia - Nokia Mobile Internet Toolkit"**;  
<http://www.forum.nokia.com/main.html>
- [18] **"Openwave WAP Simulator"**;  
<http://www.openwave.com/>
- [19] **"Tutorial WML"**;  
<http://www.wmlclub.com/tutorialwml/index.htm>
- [20] **"Tutorial WMLScript"**;  
<http://www.wmlclub.com/tutorialwmls/index.htm>
- [21] **"Emuladores WAP (varios)"**;  
<http://www.wapsilon.com>,  
<http://emulator.tagtag.com/wapemulator.cgi>

## 9 - APENDICES

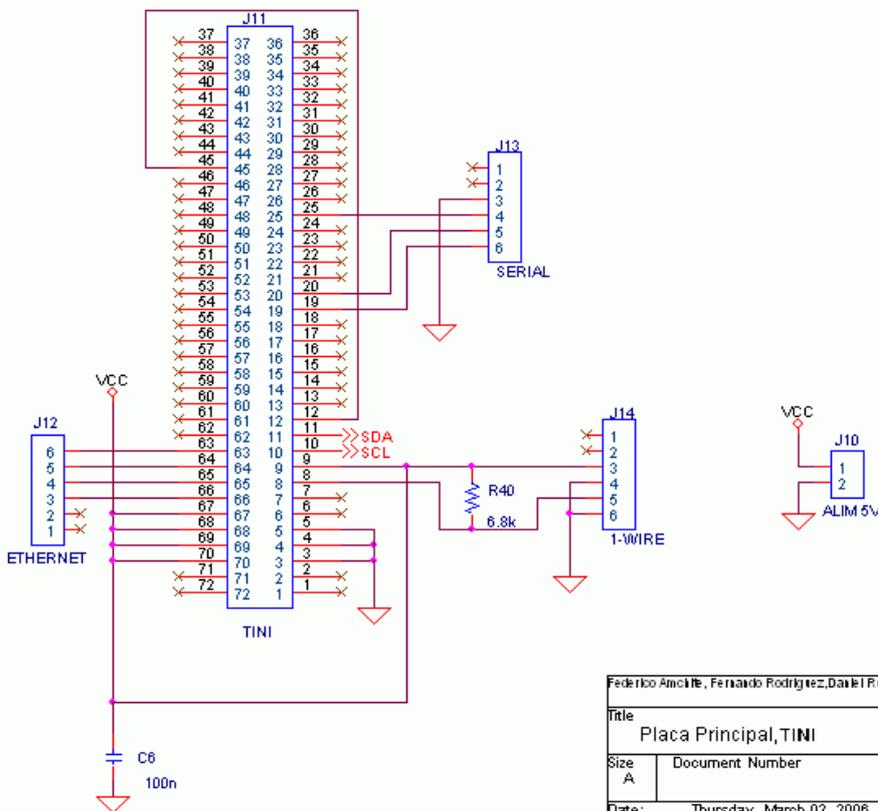
### 9.1 - APENDICE A: ESQUEMATICOS



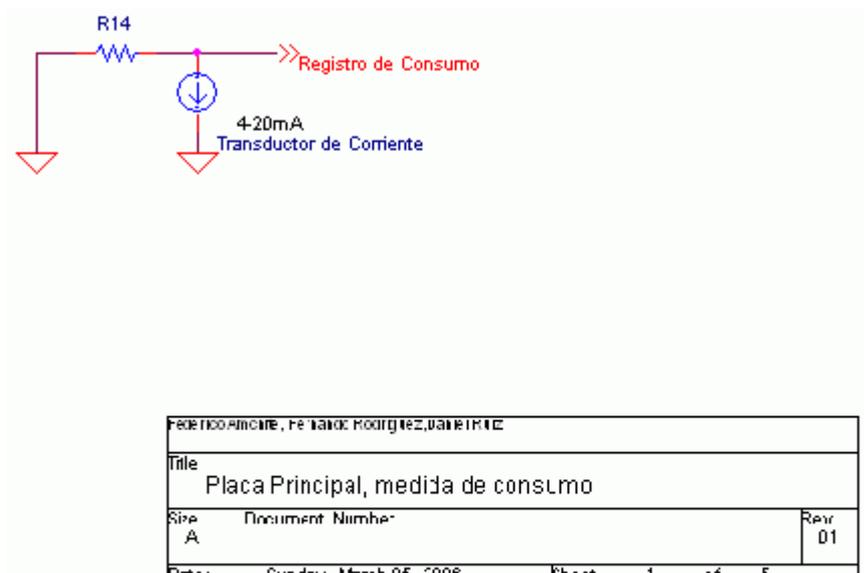
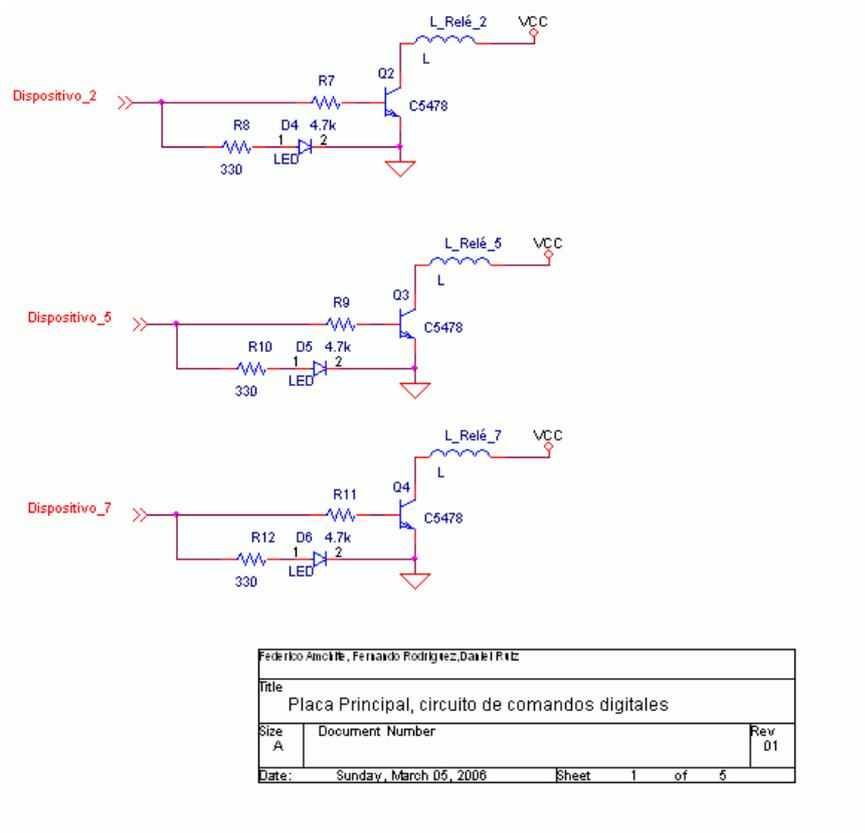
Federico Amador, Fernando Rodriguez, Daniel R. C.		
Title		
Fuente de Alimentación		
Size	Document Number	Rev
A		D1
Date:	Sunday, March 05, 2006	Sheet 1 of 4



Federico Amcibé, Fernando Rodríguez, Daniel Rite		
Title Placa Principal, Circuito de Microcontrolador		
Size A	Document Number	Rev D1
Date:	Thursday, March 02, 2006	Sheet 1 of 5



Federico Amcibé, Fernando Rodríguez, Daniel Rite		
Title Placa Principal, TINI		
Size A	Document Number	Rev D1
Date:	Thursday, March 02, 2006	Sheet 1 of 5



## 9.2 - APENDICE B: PERFIL – MANUAL DE AYUDA

### COMO CREAR Y MODIFICAR UN PERFIL

*Def.* Un Perfil es el comportamiento diario de un conjunto de dispositivos del sistema. Es donde se configura que se quiere controlar, encender o apagar en la casa en un día.

Para la configuración de los perfiles se selecciona en la barra de menú *CONFIGURACIÓN => Sistema => Perfiles*. Solamente el usuario *admin* tiene permisos para crear, modificar o borrar perfiles. El usuario *guest* puede ver el perfil activo del día.

Al ingresar, si existe un perfil configurado para la fecha actual, lo primero que se observa es la pantalla de administración con las información del perfil. Se despliega un árbol con los dispositivos y habitaciones, y los botones de “Nuevo Perfil”, “Editar otro Perfil”, “Guardar Perfil” y “Borrar perfil”.

Sobre los botones se muestra el comentario que tiene el perfil y el archivo \*.prf sobre el que se esta trabajando.

Sobre el árbol de la casa están los botones de acción, “Agregar” y “Remover”.

#### NUEVO PERFIL

Permite crear un perfil. Nos muestra un cuadro de diálogo donde podemos elegir el perfil a crear, de los archivos de perfil existentes, y agregarle un comentario de referencia.

#### EDITAR OTRO PERFIL

Permite editar los perfiles existentes. Muestra un cuadro de diálogo donde se puede elegir, dentro de los perfiles hechos, el perfil a modificar.

#### GUARDAR PERFIL

Permite guardar el perfil actual en el sistema de archivos, quedando permanente.

#### BORRAR PERFIL

Borra de forma permanente el perfil seleccionado, liberando todos los recursos (nombre de archivo, dispositivos y puertos digitales o 1-wire si corresponde).

### ***NAVEGACION EN EL ARBOL***

La pantalla de inicio contiene el árbol, mostrando el contenido del perfil (habitaciones y dispositivos) y los botones de acción sobre los archivos.

Al seleccionar sobre el árbol la “casa” (nodo de primer nivel) podemos agregar habitaciones (nodos de segundo nivel). Seleccionando las habitaciones podemos agregar dispositivos (nodos de tercer nivel).

Al situarnos sobre un dispositivo se despliega un panel conteniendo: la tabla para ingresar las horas y acciones, el nombre del dispositivo, el puerto digital o 1-wire asociado y una barra de botones de acciones sobre la tabla.

### ***BOTONES DE DISPOSITIVOS***

#### **NUEVA LINEA**

Permite agregar una línea de horario a la tabla del dispositivo.

#### **INSERTAR LINEA**

Permite insertar líneas en la tabla.

#### **BORRAR LINEA**

Permite borrar una o más líneas de la tabla.

#### **APLICAR**

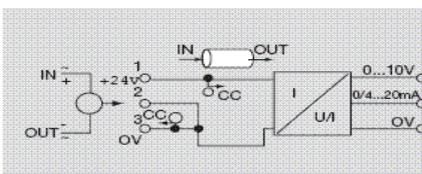
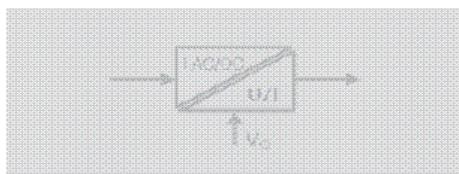
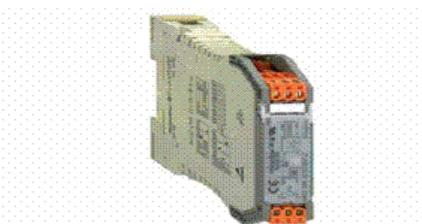
Guarda los datos de la tabla en memoria.

## 9.3 - APENDICE C: HOJAS DE DATOS

### Salida analógica

- Margen de entrada y salida a través del microinterruptor ajustable
- No precisa calibración

### 5/10A ac/dc



### Datos técnicos

<b>Entrada</b>
Corriente de entrada
Frecuencia de entrada
Corriente máx.
Tensión de circuito de medición
Sensor
Diámetro de la versión

0...5 Aac/dc/ 0...10 Aac/dc
0...2 kHz (RMS real para convertidor DC)
dependiendo de la sección del conductor
400 Vac, > 400V ac depend. del aislamiento del conductor
Sensor de Hall (interno)
8 mm

<b>Salida</b>
Intensidad de salida/Tensión de salida
Corriente de error
Límite de la señal de salida
Tensión de la resistencia de carga/Corriente
Salida de respuesta transitoria
Precisión
Coefficiente de temperatura
Indicador de estado

0(4)...20 mA/0...10 V
máx. 150 $\mu$ A
aprox. 13 V bzw. 24 mA
$\geq 1 \text{ k}\Omega / < 600 \Omega$
tip. 700 ms
0,5% FSR
$\leq 650 \text{ ppm/K}$
LED ON: OK; LED intermitente: no hay señal en el rango;
LED OFF: Error

<b>Datos generales</b>
Tensión de alimentación
Consumo
Capacidad de carga de la conex. transv.
Temperatura de servicio/Temperatura almacenamiento
Instalación de obra
Homologaciones

24 Vdc +/- 10 %
< 50 mA a Iout = 20 mA
$\leq 2 \text{ A}$
0 °C...+50 °C/-20 °C...+70 °C
0...5A, 4...20mA
CE / ESD / cURus

<b>Aislamiento</b>
Tensión nominal
Sobretensión de choque
Grado de polución
Categoría de sobretensión
Distancia de fuga y aire
Tensión de aislamiento (entrada /salida)

300 V
6 kV
2
III
$\geq 5,5 \text{ mm}$
4 kVeff / 5 s

<b>Dimensiones</b>	
Sección de embomado (nom./min./máx.)	mm <sup>2</sup>
Longitud / Anchura / Altura	mm

<b>Conexión brida-tornillo</b>	<b>Conex. directa</b>
2,5 / 0,5 / 2,5	1,5 / 0,5 / 2,5
92,4 x 22,5 x 112,4	92,4 x 22,5 x 112,4

<b>Indicación</b>
Tu=23°C, módulo simple

--

### Datos para pedido

<b>Tipo de conexión</b>	
Conexión brida-tornillo	
Conexión directa	

<b>Tipo</b>	<b>U.E.</b>	<b>Código</b>
WAS2 CMA 5/10A uc	1	852661000
WAZ2 CMA 5/10A uc	1	852662000

<b>Indicación</b>
-------------------

--

### Accesorios

<b>Indicación</b>
-------------------

Tensión de alimentación de 24V y 0V conectable transversalmente con ZGV 2,5N/2
--



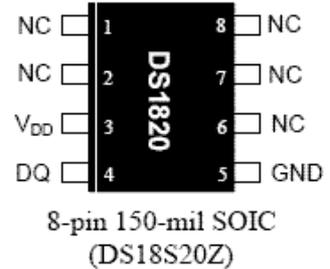
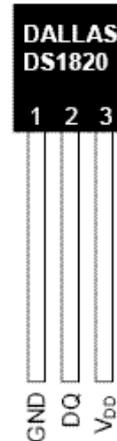
**DS18S20**  
**High Precision**  
**1-Wire<sup>®</sup> Digital Thermometer**

[www.dalsemi.com](http://www.dalsemi.com)

**FEATURES**

- Unique 1-wire interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an on-board ROM
- Multi-drop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$ )
- $\pm 0.5^{\circ}\text{C}$  accuracy from  $-10^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$
- 9-bit thermometer resolution
- Converts temperature in 750 ms (max.)
- User-definable nonvolatile alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

**PIN ASSIGNMENT**



**PIN DESCRIPTION**

- GND - Ground
- DQ - Data In/Out
- V<sub>DD</sub> - Power Supply Voltage
- NC - No Connect

**DESCRIPTION**

The DS18S20 Digital Thermometer provides 9-bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  and is accurate to  $\pm 0.5^{\circ}\text{C}$  over the range of  $-10^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ . In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.



# Relé miniatura AT AT miniature relay

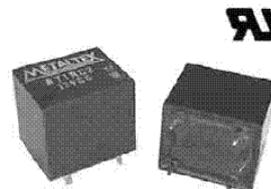
- 1 contato reversível para 15 A
- Montagem direta em circuito impresso
- Hermeticamente selado
- Baixo custo
- Homologado UL

- 15 A contact
- Direct PC mounting
- Hermetically sealed
- Low cost
- UL recognized

Chave de código / How to order

AT1R C2

Tensão nominal da bobina / Nominal voltage  
 C-5V - 5 VCC / VDC    C2 - 12 VCC / VDC  
 C1 - 6 VCC / VDC    C3 - 24 VCC / VDC  
 C-9V - 9 VCC / VDC    C4 - 48 VCC / VDC



## Especificações da bobina / Coil specifications

Modelo / Type	Tensão Nominal VCC / Nominal Voltage VDC	Resistência (±10%) Ω* / Coil resistance (±10%) Ω*	Corrente Nominal mA / Nominal Current mA	Tensão de Atraque VCC / Pick-up Voltage VDC	Tensão de Desoperação VCC / Drop-out Voltage VDC	Máx. Tensão Contínua VCC / Max. Allowable Voltage VDC
C-5V	5	70	72,0	≤ 4,0	≥ 0,5	5,5
C1	6	100	60,0	≤ 5,1	≥ 0,6	6,6
C-9V	9	225	40,0	≤ 7,2	≥ 0,9	9,9
C2	12	400	30,0	≤ 9,6	≥ 1,2	13,2
C3	24	1600	15,0	≤ 19,2	≥ 2,4	26,4
C4	48	4500	10,6	≤ 38,4	≥ 4,8	52,8

\* ±15% acima (over) de 1200Ω

## Especificações do contato / Contact specifications

Capacidade do contato / Rated current (Carga resistiva / Resistive load)	120 VCA / VAC 15 A 277 VCA / VAC 10 A 24 VCC / VDC 15 A
Corrente de comutação máx. / Maximum switching current	15 A
Tensão de comutação máx. / Maximum switching voltage	240 VCA / 110 VCC / 240 VAC / 110 VDC
Corrente de condução máx. / Maximum allowable current	15 A
Resistência de contato inicial máx. / Maximum initial contact resistance	50 mΩ
Vida mecânica / Mechanical life	10 <sup>7</sup> operações mín. / operations min. (300 operações/ minuto) (operations/minute)
Vida elétrica / Electrical life	10 <sup>5</sup> operações mín. / operations min. (20 operações/ minuto) (operations/minute)
Tempo de operação / Operate time	10 ms máx.
Tempo de desoperação / Release time	5 ms máx.
Material dos contatos / Contact material	Liga de prata / Silver alloy

## Características gerais / Characteristics

Rigidez dielétrica entre bobina e contatos / Breakdown voltage between contact and coil	1500 VCA/VAC (1 minuto/minute)
Rigidez dielétrica entre contatos abertos / Breakdown voltage between open contacts	750 VCA/VAC (1 minuto/minute)
Resistência de isolamento / Insulation resistance	100 MΩ min (500 VCC/VDC)
Temperatura de operação / Operating ambient temperature	-40 a (to) +70°C
Resistência à vibração / Vibration resistance	10 a 55 Hz dupla amplitude 1,5 mm 10 to 55 Hz d.a. 1.5 mm
Resistência a impacto / Shock resistance	10 G

Esquema elétrico / Schematic



Dimensões / Dimensions: mm

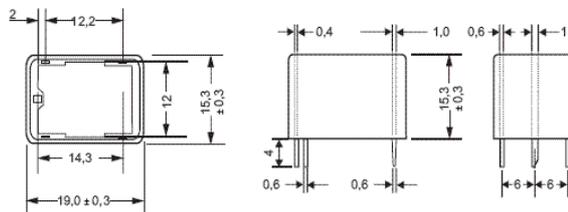
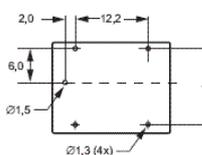


Diagrama de circuito impresso  
PC board layout



Tolerâncias não indicadas : ± 0,1 mm  
Not indicated tolerance : ± 0.1 mm



**DETAILED PIN DESCRIPTIONS Table 1**

8-PIN SOIC*	TO-92	SYMBOL	DESCRIPTION
5	1	GND	<b>Ground.</b>
4	2	DQ	<b>Data Input/Output pin.</b> Open-drain 1-wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	3	V <sub>DD</sub>	<b>Optional V<sub>DD</sub> pin.</b> V <sub>DD</sub> must be grounded for operation in parasite power mode.

\*All pins not specified in this table are "No Connect" pins.

**OVERVIEW**

Figure 1 shows a block diagram of the DS18S20, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T<sub>H</sub> and T<sub>L</sub>). The T<sub>H</sub> and T<sub>L</sub> registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Dallas' exclusive 1-wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the 1-WIRE BUS SYSTEM section of this datasheet.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C<sub>PP</sub>), which then supplies power to the device when the bus is low. This method of deriving power from the 1-wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V<sub>DD</sub>.

**DS18S20 BLOCK DIAGRAM Figure 1**

