

# SIMIC

## Sistema Médico de Insuficiencia Cardíaca

Alejandro Cardone  
Rodrigo González  
Viterbo García  
Tutor: Franco Simini  
Co-Tutores: Gabriela Ormaechea  
Pablo Álvarez

## **Agradecimientos**

En primer lugar, queremos agradecer a nuestras familias y amigos por el apoyo incondicional recibido desde que comenzamos con esta carrera.

Al tutor del proyecto, Ing. Franco Simini, quien demostró compromiso, aportando su experiencia y sus conocimientos para guiarnos en el transcurso de esta etapa tan importante para nuestras carreras.

A la Universidad de la República, y a todo su equipo docente, a quienes debemos la formación que nos permitió afrontar con seguridad los desafíos que surgieron durante el proyecto.

A los integrantes de la Unidad Multidisciplinaria de Insuficiencia Cardíaca (UMIC) del Hospital de Clínicas, en especial a sus médicos Gabriela Ormaechea, Pablo Álvarez y Gabriela Silveira, por su compromiso, dedicación y enseñanzas obtenidas en el tiempo compartido junto a ellos.

A todos ellos, nuestro más sincero agradecimiento.

Rodrigo González  
Alejandro Cardone  
Viterbo García

## Resumen

SIMIC es una aplicación web que tiene como fin ayudar a los integrantes de la Unidad Multidisciplinaria de Insuficiencia Cardíaca (UMIC) del Hospital de Clínicas a realizar el seguimiento sus pacientes.

Busca que el médico disponga de información en la consulta y no tenga que estar re-preguntando al paciente, omitiendo datos que el paciente no sabe o no trajo de la atención con otro especialista.

Desde una misma interfaz se puede gestionar el historial médico de los pacientes y detallar sus antecedentes, enfermedades y tratamientos.

Cuenta con un sistema de alertas que facilitan el seguimiento del paciente. Por ejemplo, cuando un paciente ha pasado mucho tiempo sin concurrir a una consulta o cuando un paciente vive solo. Además de un sistema de indicadores en tiempo real claves en la evolución de pacientes y el servicio general de la unidad.

Mediante el uso de una aplicación móvil el paciente puede aportar información adicional al sistema para contribuir con su seguimiento (por ej. su peso diario), realizar un seguimiento de su medicación e informarse sobre su enfermedad.

SIMIC utiliza estándares definidos por el programa SALUD.UY y se integra con el repositorio nacional de historias clínicas ofrecido por el índice maestro de personas.

Dadas las características del proyecto, se optó por la utilización de una metodología híbrida para la gestión de su desarrollo: SCRUM [1], [2] (Metodologías Ágiles) para la fase de construcción del producto y un híbrido de este y aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado (Metodologías Tradicionales) para el resto.

1	Contenido	
2	Índice de Tablas.....	9
3	Índice de Figuras .....	10
4	Introducción .....	12
4.1	Introducción .....	12
4.2	Motivación .....	12
4.3	Propósito y alcance.....	13
4.4	Descripción del “cliente” .....	13
4.5	Equipo de proyecto y otros actores.....	13
4.6	Organización del documento .....	14
5	Descripción del proyecto.....	17
5.1	Introducción .....	17
5.2	Descripción general .....	17
5.3	Objetivos del proyecto .....	17
5.4	Alcance del sistema.....	18
5.4.1	Descripción del sistema y su entorno.....	18
5.4.2	Descripción resumida de funciones del sistema.....	18
5.5	Descripción de forma de trabajo .....	19
5.6	Proceso de desarrollo .....	19
5.6.1	Características del proyecto.....	19
5.6.2	Selección del proceso.....	20
5.6.3	Planificación del proceso de control y seguimiento del proyecto .....	20
5.6.4	Iteración 0: Elaboración de la lista de requerimientos y planificación .....	21
5.6.5	Iteraciones de desarrollo.....	22
6	Inicio del proyecto .....	25
6.1	Introducción .....	25
6.2	Presentación del proyecto.....	25
6.3	Investigación y capacitación .....	25
6.4	Organigrama del proyecto.....	26
7	Estado del arte.....	27
7.1	Introducción .....	27
7.1.1	Insuficiencia Cardíaca.....	27
7.1.2	Unidad Multidisciplinaria de Insuficiencia Cardíaca.....	27
7.1.3	Importancia de la unidad de insuficiencia cardíaca .....	28
7.1.4	Objetivos de la UMIC .....	29

7.1.5	Software en la medicina.....	29
7.2	Casos de estudio.....	30
7.2.1	Caso 1.....	30
7.2.2	Caso 2.....	31
7.2.3	Caso 3.....	33
7.2.4	Caso 4.....	33
7.2.5	Caso 5.....	35
7.2.6	Caso 6.....	36
7.3	Software actual.....	38
7.3.1	Kareo.....	38
7.3.2	MediTouch.....	39
7.3.3	AdvanceMD.....	40
7.3.4	NueMD.....	40
7.3.5	iSalus Healthcare.....	40
7.3.6	Epi Info.....	41
7.4	Observaciones.....	42
8	Ingeniería de requerimientos.....	43
8.1	Introducción.....	43
8.2	Estrategia de relevamiento.....	43
8.3	Situación inicial.....	43
8.4	Inconvenientes.....	44
8.5	Situación proyectada.....	45
8.6	Beneficios.....	46
8.7	Actores.....	47
8.7.1	Cliente.....	47
8.7.2	Roles de negocio.....	47
8.7.3	Usuarios de la solución.....	47
8.8	Especificación de requerimientos (ESRE).....	48
8.8.1	Requerimientos funcionales.....	48
8.8.2	Requerimientos no funcionales.....	58
9	Especificación del diseño.....	60
9.1	Introducción.....	60
9.2	Arquitectura del sistema.....	60
9.2.1	Arquitectura de la aplicación web.....	61
9.2.2	Arquitectura de la aplicación móvil.....	61
9.2.3	Arquitectura del FrontEnd.....	63

9.3	Servicios REST .....	64
9.3.1	Operaciones .....	64
9.3.2	La estructura de la URI .....	64
9.3.3	Tratamiento de errores .....	66
9.4	Otros patrones de diseños .....	66
9.5	Vista física .....	66
9.5.1	Diagrama de despliegue .....	66
9.5.2	Diagrama de componentes .....	68
9.5.3	Diagrama de Clases – Servidor Central .....	69
9.6	Resolución de otros problemas arquitectónicos .....	69
9.6.1	Integración con estándares de SALUD.UY .....	70
9.7	Decisión de tecnologías .....	73
9.7.1	Servidor .....	73
9.7.2	Capa de datos .....	74
9.7.3	FrontEnd .....	75
9.7.4	Otras tecnologías .....	78
10	Calidad y migración de datos .....	80
10.1	Testing .....	80
10.1.1	General .....	80
10.1.2	Test de carga .....	80
10.1.3	Test de seguridad .....	87
10.1.4	Ingeniería de Muestra .....	89
10.1.5	Integración proyecto de extensión DONOCARDIOFACIL .....	89
10.2	Migración de datos .....	90
10.2.1	Inicialización del sistema .....	90
10.2.2	Exportación de datos de Epi Info .....	92
10.2.3	Relacionamiento de tablas exportadas con las entidades de SIMIC .....	92
10.2.4	Migración de Pacientes .....	92
10.2.5	Migración de Resultados de exámenes .....	93
10.2.6	Migración de Eventos y Controles .....	93
10.3	Satisfacción del cliente .....	93
11	Gestión de la configuración .....	95
11.1	Gestión del repositorio .....	95
11.1.1	Estructura de repositorio de documentación .....	95
11.1.2	Estructura de repositorio de código .....	96
11.1.3	Proceso de gestión de repositorio .....	96

12	Gestión.....	98
12.1	Introducción .....	98
12.2	Gestión del alcance .....	98
12.2.1	Planificación del alcance.....	98
12.2.2	Cambios del alcance.....	98
12.2.3	Control del alcance .....	99
12.2.4	Avance del alcance.....	99
12.3	Gestión del tiempo.....	100
12.4	Gestión del esfuerzo .....	101
12.4.1	Planificación y control del esfuerzo .....	101
12.4.2	Avance del esfuerzo .....	102
12.5	Gestión de los Recursos Humanos.....	106
12.5.1	Normas de funcionamiento .....	106
12.5.2	Capacitación.....	106
12.6	Gestión de la comunicación .....	108
12.6.1	Comunicación entre los integrantes del equipo.....	108
12.6.2	Comunicación con el tutor.....	109
12.6.3	Comunicación con los integrantes de UMIC .....	109
12.7	Gestión de riesgos.....	109
13	Conclusiones.....	111
13.1	Análisis de gestión.....	111
13.1.1	Alcance.....	111
13.1.2	Calidad .....	111
13.1.3	Tiempo.....	111
13.1.4	Esfuerzo .....	111
13.2	Conclusiones generales.....	112
13.3	Lecciones aprendidas .....	112
14	Trabajo futuro .....	114
15	Referencias .....	115
16	Anexos .....	117
16.1	Anexo 1: Glosario .....	117
16.2	Anexo 2: Planificación y contrato.....	117
16.3	Anexo 2: Investigación de SCRUM.....	117
16.4	Anexo 4: Modelo de datos .....	117
16.5	Anexo 5: Resumen de seguridad.....	117
16.6	Anexo 6: Ingeniería de muestra.....	117

16.7	Anexo 7: Folleto de ingeniería de muestra.....	117
16.8	Anexo 8: Encuesta para conocer el uso de la tecnología por parte de pacientes.	117
16.9	Anexo 9: Encuesta de satisfacción del cliente.....	117
16.10	Anexo 10: Bitácora de reuniones.....	117
16.11	Anexo 11: Horas.....	117

## 2 Índice de Tablas

TABLA 1 Métodos Http .....	64
TABLA 2 Representación uso de las URI.....	65
TABLA 3 Comparación de popularidad entre distintos Frameworks JavaScript .....	78

### 3 Índice de Figuras

Figura 1 Proceso que se siguió en la etapa de desarrollo, se resalta el bucle entre Prueba y Desarrollo en el cual no pasa a la etapa de revisión sin una aprobación. ....	23
Figura 2 Modelo de ciclo de vida que se llevó para la ejecución del proyecto. ....	24
Figura 3 Se muestran para los diferentes rangos de edad la comparación entre el total de los pacientes y los pacientes que usan computadoras.....	32
Figura 4 Frecuencia del uso de tecnología.....	32
Figura 5 Arquitectura de SIMS, se observa la comunicación entre SIMS, los pacientes y diferente personal de salud.....	35
Figura 6 Flujo de funcionamiento de CCCP, se aprecia los diferentes medios que puede usar el paciente para su monitoreo .....	36
Figura 7 Proceso de comunicación entre los participantes del proyecto .....	38
Figura 8 Proceso de atención de la UMIC.....	44
Figura 9 Situación proyectada de la solución al inicio del proyecto, se muestran el acceso de los diferentes usuarios. Los estándares que implementara y algunas de las funcionalidades. ....	46
Figura 10 Diagrama básico que muestra los componentes de la aplicación, notar que la api REST se consume desde la aplicación web y la aplicación móvil .....	60
Figura 11 Comunicación entre el FrontEnd y BackEnd, notar que en el FrontEnd también existen los controladores, servicios y persistencia. ....	61
Figura 12 Arquitectura de la aplicación móvil, se aprecia la comunicación de sus componentes. ....	63
Figura 13 Se muestra el uso del patrón MVC en el FrontEnd .....	63
Figura 14 Diagrama de despliegue de SIMIC.....	67
Figura 15 Diagrama de componentes de SIMIC.....	68
Figura 16 extraída de google trends, que demuestra la baja de interés por JSF .....	75
Figura 17 extraída de google trends que demuestra el creciente interes de la comunidad por Spring MVC .....	76
Figura 18 Cantidad tareas reportadas, pendientes de solución y realizadas. No todos son bugs ya que muchos son comentarios para posibles mejoras.....	80
Figura 19 Tiempo de ejecución en ms de cada uno de las tareas que ejecuta el hilo.....	82
Figura 20 Programa radial "En Perspectiva", participación de Alejandro.....	89
Figura 21 Proceso para gestión de repositorio de documentos, observar que para el borrado de documentos se debe de tener aprobación de todo el equipo .....	97
Figura 22 Apariencia identificativa de la herramienta Trello, board de Plan de Proyecto...	100
Figura 23 Board Desarrollo extraído de la herramienta Trello, se pueden observar las diferentes tareas y asignaciones a los integrantes .....	101
Figura 24 Burndown Chart, se puede observar la poca experiencia previa con la que contaba el equipo para realizar estimaciones precisas .....	102
Figura 25 Total de horas por integrante del equipo .....	102
Figura 26 Evolución en el tiempo de las horas dedicadas por el equipo .....	103
Figura 27 Evolución de las horas dedicadas por el equipo para implementación.....	103
Figura 28 Distribución total de las horas según las diferentes tareas, se aprecia la cantidad de horas dedicadas a implementación .....	104

Figura 29 Distribución de horas de Rodrigo .....	104
Figura 30 Distribución de horas Viterbo .....	105
Figura 31 Distribución de horas Alejandro.....	105
Figura 32 Planificación del equipo, se observa las cards usadas para el Poker Planning .	110

## 4 Introducción

### 4.1 Introducción

En este documento se describe el proceso del proyecto académico SIMIC realizado como requisito para la obtención del título de grado Ingeniero en Computación de los estudiantes involucrados. Se realiza bajo la tutoría del Ing. Franco Simini.

El objetivo es el desarrollo de una solución informática que permita el seguimiento de pacientes que asisten a la Unidad Multidisciplinaria de Insuficiencia Cardíaca (UMIC) y ayude a los integrantes de esta unidad a un mejor control en la atención por medio la centralización de la información del tratamiento, alertas, indicadores e integración con los estándares propuestos por el gobierno en el ámbito de la salud.

Con la ejecución de este proyecto académico el equipo obtuvo como valor agregado el aprendizaje de nuevas tecnologías y metodologías de trabajo. Así como también la profundización en el conocimiento de las técnicas de ingeniería de software y las tareas de apoyo necesarias para la concreción exitosa de un proyecto.

Comprendimos las dificultades a las que se enfrentan los médicos en el sistema de salud de Uruguay y el compromiso que ellos asumen.

### 4.2 Motivación

Al comenzar la búsqueda de un proyecto para desarrollar como trabajo de fin de carrera, el equipo debió discutir acerca de qué tipo de proyecto realizaría. Así fue que se definió buscar un proyecto que fuera un desafío tanto del punto de vista tecnológico, incorporando nuevas tecnologías, como del punto de vista de las metodologías, permitiendo ensayar nuevas metodologías de trabajo poco vistas en la carrera. Además, aportar a la sociedad un granito de arena para su mejora.

Tomando las consideraciones anteriores es que el equipo optó por el proyecto SIMIC, luego de buscar un proyecto que cumpliera las características anteriormente mencionadas en las diferentes propuestas presentadas por los institutos de la facultad.

Una de las cosas que nos llamó la atención de esta propuesta fue tener la oportunidad de poder encontrar una solución a un problema real de nuestro sistema de salud y ayudar a los médicos a mejorar la atención de los pacientes en UMIC. También el poder conocer e involucrarnos en un área nueva para el equipo, como lo es la medicina, donde se usa un lenguaje específico, leyes, estándares, entre otras cosas. Además, el desafío de adaptarse a un entorno de requerimientos cambiantes, con la consiguiente posibilidad de explorar metodologías de trabajo menos tradicionales.

## 4.3 Propósito y alcance

El propósito del proyecto SIMIC es desarrollar una solución informática que facilite la gestión y seguimiento de los pacientes de la UMIC. La solución abarca tanto la obtención de datos necesarios, como la generación de indicadores y estadísticas que permitan facilitar la toma de decisiones en la gestión de la atención. Además, incluye alertas que ayudan a los integrantes de la unidad a tener un mejor control del seguimiento del paciente.

El software cuenta con un módulo web orientado a los integrantes de la UMIC, médicos referentes en la atención del paciente y también a médicos de otros centros de salud que actúan como médicos tratantes. Este módulo permitirá ingresar todos los datos de los pacientes, exámenes, valores de signos vitales, medicación y eventos. Además, se mostrarán diferentes indicadores que ayudarán con la toma de decisiones en la gestión de la atención de los pacientes. De forma complementaria el módulo móvil (desarrollado en Android) permite a los pacientes ingresar datos que pueden ser beneficiosos sus conocimientos por parte del médico para detectar en forma temprana posibles problemas en la enfermedad. En este módulo el paciente podrá también conocer su medicación, también tendrá disponible información de la enfermedad para poder mejorar el cuidado que se necesita.

## 4.4 Descripción del “cliente”

La UMIC está integrada por un grupo de profesionales de la salud pertenecientes al Hospital de Clínicas “Dr. Manuel Quíntela” de la República Oriental del Uruguay. Conforman un equipo multidisciplinario de trabajo interdisciplinario bajo la concepción de los programas de manejo de enfermedades crónicas para atender a los pacientes portadores de IC crónica por disfunción sistólica.

El método de trabajo es el de una policlínica especializada. Participan médicos internistas, cardiólogos, psiquiatra, ecocardiografías, nutricionista, licenciada en enfermería y asistente social.

## 4.5 Equipo de proyecto y otros actores

El equipo se encuentra integrado por 3 estudiantes de la carrera de Ingeniería en Sistemas, un tutor y un cliente (UMIC). A continuación se describen los perfiles de los Integrantes:

El equipo se encuentra integrado por 3 estudiantes de la carrera de Ingeniería en Computación, un tutor y un cliente (UMIC). A continuación, se describen los perfiles de los integrantes:

**Ing. Franco Simini**

Tutor del proyecto

**Ing. Paulo Sande**

Asistente del NIB  
Responsable de HCE en SALUD.UY, AGESIC

**Dr. Walter Callero**  
AGESIC

**Dra. Gabriela Ormaechea**  
Integrante de UMIC

**Dr. Pablo Álvarez**  
Integrante de UMIC

**Dra. Gabriela Silvera**  
Integrante de UMIC

**Dr. Martín Collares**  
Integrante de UMIC

**Rodrigo González**

Estudiante avanzado de Ingeniería en Computación con más de 3 años de experiencia en desarrollo de software, orientado principalmente a aplicaciones Web. Posee especial interés en temas relacionados al diseño de interfaces gráficas, metodologías ágiles y mejora en los procesos de trabajo.

Actualmente trabaja como desarrollador en java en Sofis Solutions.

**Alejandro Cardone**

Estudiante avanzado de la carrera Ingeniería en Computación. Posee experiencia en relevamiento y especificación de requisitos, habiéndose dedicado también al desarrollo de aplicaciones. Se encuentra interesado en temas relacionados a la ingeniería de requerimientos, metodologías ágiles, diseño de arquitectura y gestión de proyecto de software.

Actualmente trabaja como analista funcional en UTE, colaborando en la especificación del sistema SIO.

**Viterbo García**

Estudiante avanzado de la carrera Computación en Sistemas.

Trabaja como analista funcional en UTE, colaborando en la especificación del sistema de información geográfica EGEO, así como participando también de las pruebas del sistema y posterior dictado de cursos a usuarios.

Interés por temas relacionados a la gestión de proyectos y mejora de procesos, diseño de arquitectura, metodologías ágiles.

## 4.6 Organización del documento

A continuación, se presenta una breve descripción de los principales capítulos contenidos en el documento.

## **Glosario**

Se define la terminología usada en el documento.

## **Introducción**

Se describe muy brevemente el proyecto y los actores involucrados en el mismo y se explican los principales lineamientos del documento.

## **Descripción del proyecto**

Se realiza una descripción del proyecto incluyendo sus objetivos, su alcance y la metodología de trabajo utilizada. También se realiza una introducción al proceso utilizado durante la ejecución del proyecto.

## **Constitución del proyecto**

Se describen las tareas realizadas durante las primeras fases del proyecto.

## **Estado del arte**

Se describe la investigación que se realizó para conocer el área y soluciones al problema planteado.

## **Ingeniería de requerimientos**

Se describe el proceso de ingeniería de requerimientos, indicando la metodología utilizada. También se incluye un resumen de los requisitos del sistema desarrollado.

## **Diseño arquitectónico**

Se muestra el diseño arquitectónico elaborado para la solución y se justifican las principales decisiones de diseño tomadas. Además, se describen los problemas a los que se enfrentó el equipo y las soluciones encontradas.

## **Testing y migración de datos**

Se describen las distintas actividades realizadas para el aseguramiento de la calidad durante el transcurso del proyecto, enfocadas principalmente en el producto.

Se describen las principales actividades del proceso para realizar la migración de los datos actuales de Epi Info a SIMIC.

## **Gestión de la configuración**

Se describen los distintos aspectos de la configuración del sistema, entre los que se encuentran el proceso de control de cambios y la gestión de repositorios.

**Gestión**

Se incluye un resumen general del proyecto dividido según las distintas áreas de gestión.

**Conclusiones**

Se extraen conclusiones de los datos recolectados durante el proyecto y se elabora un compendio de lecciones aprendidas.

**Trabajos a futuro**

Se describen mejoras que podrían aportar una solución a problemas que no se incluyeron en la solución por prioridades o por la gestión del tiempo de culminación.

**Bibliografía**

Se listan las fuentes y recursos a los cuales accedieron los integrantes del proyecto durante su transcurso.

**Anexos**

Se incluyen documentos relevantes para complementar la información brindada en el presente documento.

# 5 Descripción del proyecto

## 5.1 Introducción

En el presente capítulo se brinda información general del proyecto. En particular, se incluye una descripción general del proyecto, los objetivos del mismo y una síntesis del alcance del sistema.

Como complemento a esta información se incluye también una descripción del proceso y ciclo de vida utilizado, y sus fundamentos.

Como complemento a esta información se incluye también una descripción del proceso y ciclo de vida utilizados y la fundamentación de los mismos.

## 5.2 Descripción general

SIMIC es un proyecto académico de desarrollo de software cuyo objetivo es la construcción de un sistema informático que permita ayudar a los médicos de la UMIC a mejorar el seguimiento de los pacientes. En la medicina se maneja gran cantidad de información, proveniente de fuentes muy variadas, sobre los pacientes. El uso de software para la manipulación de esta información aporta un acceso rápido y fiable (sin introducción de errores humanos en la comprensión de formularios) y permite realizar respaldos.

El proyecto soporta los estándares del programa de SALUD.UY impulsado por el gobierno para viabilizar mejoras en la calidad de las prestaciones de salud recibidas por los ciudadanos.

## 5.3 Objetivos del proyecto

El principal objetivo es brindar la ayuda necesaria a los integrantes de UMIC brindando una herramienta que le permita mejorar el seguimiento de sus pacientes. La siguiente frase extraída de [3] nos parece apropiada para definir el objetivo propuesto por el equipo para la realización del proyecto “El objetivo no son mejores hospitales, ni mejores prácticas clínicas, ni equipamiento más sofisticado, sino pacientes más sanos y más felices”.

Se desarrollan dos aplicaciones, una aplicación web y otra móvil (nativa para Android), que son soporte de los médicos para el seguimiento del paciente. La aplicación web cumple los requisitos determinados en la especificación y se puede instalar en un entorno con servidor Apache Tomcat con una base de datos MySQL. La aplicación Android cumple los requisitos determinados en la especificación y se puede ser instalada en un dispositivo con una versión superior a Android 4.0.3.

Las aplicaciones desarrolladas cuentan con documentación y Además, incorporan ayudas en pantalla para su utilización por parte de los futuros usuarios.

## 5.4 Alcance del sistema

### 5.4.1 Descripción del sistema y su entorno

En la medicina se maneja gran cantidad de información, proveniente de fuentes muy variadas, sobre los pacientes. El uso de software para la manipulación de esta información brinda un acceso rápido y fiable (sin introducción de errores humanos en la comprensión de formularios) y permite realizar respaldos.

UMIC está integrada por un grupo de profesionales de la salud pertenecientes al Hospital de Clínicas “Dr. Manuel Quíntela” de la República Oriental del Uruguay. Conforman un equipo multidisciplinario de trabajo interdisciplinario bajo la concepción de los programas de manejo de enfermedades crónicas para atender a los pacientes portadores de IC crónica por disfunción sistólica. El método de trabajo es el de una policlínica especializada. Participan médicos internistas, cardiólogos, psiquiatra, ecocardiografistas, nutricionista, licenciada en enfermería y asistente social.

Uno de los problemas que tenía la unidad era el seguimiento del paciente para poder tener un mejor control en su enfermedad. SIMIC se encuentra orientado a facilitar la realización de este seguimiento a los integrantes de UMIC. El sistema es configurable, permitiendo a usuarios administradores la creación de diferentes variables necesarias para lograr el objetivo del seguimiento del paciente.

Para los pacientes se desarrolló una aplicación móvil que permite el ingreso de valores importantes para su seguimiento. Además, brinda información sobre la medicación recetada por el médico e información sobre la IC con el fin de concientizar al paciente sobre la importancia del tratamiento.

### 5.4.2 Descripción resumida de funciones del sistema

El sistema desarrollado proporciona diversos beneficios a los usuarios. A continuación se citan los principales requerimientos que satisface el producto:

Médicos, mediante aplicación web:

- Consulta de pacientes
- Indicadores de gestión de la atención
- Agenda de médicos
- Evolución de variables del paciente
- Notificaciones para el seguimiento del paciente
- Integración con estándar de historia clínica electrónica (XDS [4])
- Integración con identificador único para personas (EMPI [5])
- Integración con servidor terminológico (SNOMED [6])
- Base de datos auditable
- Seguridad mediante https

- Ayudas de usuario en pantalla

Pacientes, mediante aplicación Android:

- Registro de la variable peso
- Lista de medicación recetada al paciente
- Información de la enfermedad
- Seguridad mediante https

## 5.5 Descripción de forma de trabajo

Al comenzar el proyecto el equipo realizó una reunión con el objetivo de establecer reglas de trabajo de forma de evitar problemas a futuro. Entre las reglas acordadas se encuentran el esfuerzo esperado por cada uno, la agenda de eventos importantes de los integrantes (exámenes, licencias de trabajo, entre otras).

Para documentar los requerimientos relevados durante el proyecto se elaboró una Especificación de Requerimientos (ESRE), incluyendo requerimientos funcionales y no funcionales, supuestos y restricciones de la aplicación. Se realizó una descripción detallada de las funcionalidades utilizando historias de usuario como lo indica el proceso de desarrollo utilizado (SCRUM), además de bosquejos de interfaz en funcionalidades que lo ameritaba para su comprensión.

Junto con el ESRE el equipo realizó la Especificación de Diseño (ESDI) definiendo la arquitectura a desarrollar.

Para la construcción de los productos requeridos se utilizó una metodología ágil basada en SCRUM. En cada iteración se priorizaron los requerimientos balanceando el valor que cada requisito aporta al negocio frente al coste estimado que tiene su desarrollo, es decir, basándose en el Retorno de la Inversión (ROI).

Al finalizar cada ciclo de SCRUM se presentó un incremento del producto al cliente y luego de cada presentación se definieron los requisitos a implementar para la siguiente iteración.

## 5.6 Proceso de desarrollo

### 5.6.1 Características del proyecto

La definición del proceso se realiza teniendo en cuenta las principales características del proyecto, las cuales se listan a continuación:

- Tecnologías nuevas para los integrantes del equipo
- Integrantes de UMIC con disponibilidad limitada por los horarios de atención de los pacientes de la unidad
- Necesidad de que las interfaces cuenten con una gran facilidad de uso generando así la necesidad de feedback temprano

- Requerimientos inestables durante la mayor parte del proyecto a causa de la introducción de los integrantes del equipo en un área desconocida
- Restricciones de fechas, el sistema debe estar implantado para el correr del año 2016
- Dedicación al proyecto en horas por parte de los integrantes será irregular (parciales, exámenes, licencias laborales, entre otras)

### 5.6.2 Selección del proceso

Teniendo en cuenta las características anteriormente mencionadas se decidió por parte del equipo la utilización de SCRUM como metodología para el desarrollo de software. Algunos de los factores que incidieron positivamente en la decisión fueron:

- Tener resultados a corto plazo
- Cambiar a menudo los requisitos del proyecto
- Ver si se cumplen sus expectativas
- Solucionar los problemas que impiden que los equipos progresen
- Utilizar un proceso de gestión ligero aún en proyectos complejos
- Gestión regular de las expectativas del cliente
- Flexibilidad y adaptación
- Mitigación sistemática de los riesgos

En este contexto se realizó el documento de contrato, que se adjunta como [Anexo 2](#), para formalizar el proceso de control y seguimiento del proyecto. También se define en dicho documento la relación entre los integrantes de UMIC e integrantes de SIMIC en la ejecución del proyecto. Se utiliza la metodología ágil SCRUM principalmente durante la etapa de implementación.

El proceso consistió en una fase primaria de constitución del proyecto, donde se delimitó su alcance en forma conjunta con los integrantes UMIC, se identificaron los principales riesgos y se definió cómo se realizaría la gestión del proyecto.

Luego se realizó la investigación en el área seguido a una fase de análisis, se realizó un diseño de alto nivel de la aplicación para tener como referencia en la fase de construcción. Paralelamente a la etapa anterior se realizó una etapa de formación en la que los integrantes tomaron contacto con las tecnologías y metodologías necesarias para poder realizar el proyecto.

### 5.6.3 Planificación del proceso de control y seguimiento del proyecto

El proyecto se ejecutó en iteraciones incrementales con una demostración del producto al finalizar cada iteración. De esta manera se pudo conocer de forma objetiva el estado del proyecto (si el desarrollo de los requisitos cumple con las expectativas, si la calidad es la esperada o si hay retrasos), con lo que UMIC pudo tomar decisiones en forma temprana.

Cada requisito se desarrolló priorizado según un valor definido conjuntamente con la UMIC, de modo que en las primeras iteraciones se obtuvieron los objetivos más importantes del proyecto y se pudieron realizar ajustes al respecto con suficiente antelación.

El control y seguimiento del proyecto se basó en los requisitos completados en cada iteración. Se entiende un requisito como completado si incluye todos los entregables asociados realizados (documentación, análisis funcional, diseño técnico, código y pruebas funcional) e integrados con los entregables de las iteraciones anteriores, de manera que el producto pueda ser entregado a UMIC con el mínimo esfuerzo.

Por esta razón, cada requisito debió ser:

- Independiente del resto de requisitos, en la medida de lo posible.
- Demostrable, de manera que sea claro cómo comprobar con UMIC que el requisito está completado y que se cumplen sus expectativas.
- De un grado de esfuerzo para ser completado semejante al del resto de requisitos, para poder realizar un seguimiento del avance del proyecto.

#### 5.6.4 Iteración 0: Elaboración de la lista de requerimientos y planificación

Duración: 3 ½ meses.

##### 5.6.4.1 Objetivos:

- Planificar y distribuir los objetivos y alcance del proyecto en iteraciones, de manera que los requisitos estén priorizados balanceando el beneficio que aportan a UMIC, su coste de desarrollo y los riesgos del proyecto. De esta manera, las primeras iteraciones del proyecto podrán acomodar los requisitos más importantes y mitigar los riesgos más altos.

##### 5.6.4.2 Actividades:

- Identificación de los objetivos del proyecto y de los requerimientos iniciales de alto nivel que permiten la consecución de estos objetivos.
- Priorización de los requerimientos en iteraciones y entregas considerando los siguientes criterios:
  - El valor aportado por cada requerimiento para UMIC. Deberá ser explícito quién es el actor o usuario beneficiario de cada requerimiento y qué valor le aporta.
  - El esfuerzo necesario para desarrollar cada uno de los requerimientos, de manera que en las primeras iteraciones se desarrollen los requerimientos que proporcionen el máximo valor con el mínimo esfuerzo y que puedan encajar en la periodicidad de las iteraciones.
  - Las dependencias inevitables entre requerimiento.

- Minimizar los riesgos del proyecto respecto a desarrollo de los requerimiento, disponibilidad y grado de implicación de los actores y beneficiarios implicados.
- Maximizar la cohesión del contenido de cada iteración, identificando los puntos de acoplamiento y las dependencias entre los diferentes incrementos de manera que sean mínimos, para poder dar por realmente completados los requerimientos desarrollados en cada una de las iteraciones.
- Calcular la duración de cada uno de los incrementos desarrollados de manera que puedan encajar en la periodicidad de las iteraciones.

## 5.6.5 Iteraciones de desarrollo

Duración: Todas las iteraciones de desarrollo no serán de tiempo fijo sino que el equipo adaptará a problemas que puedan surgir para completar la interacción.

### 5.6.5.1 Objetivos:

- Completar un incremento de producto que sea demostrable a UMIC al finalizar la iteración, de manera que se pueda tomar decisiones informadas y objetivas sobre el estado del proyecto (si el desarrollo de los requisitos cumple con las expectativas de UMIC, si la calidad es la esperada o si hay retrasos).

### 5.6.5.2 Actividades:

1. Planificación: al comienzo de cada sprint se realizó una planificación en la que los integrantes de UMIC priorizan la lista de requerimientos a resolver en el sprint. Los requerimientos se seleccionaron del product backlog.
2. Desarrollo: el equipo de SCRUM implementó las historias detalladas en el product backlog.
3. Prueba: se ejecutaron pruebas de la aplicación, principalmente testing funcional.
4. Revisión: al finalizar el sprint el equipo presentó a los integrantes de UMIC los logros alcanzados durante el sprint con el objetivo de validarlos.
5. Retrospectiva: luego de la revisión los integrantes del equipo realizaban una reunión con de fin de identificar problemas y oportunidades de mejora surgidas durante el sprint, Además, de la correcciones y cambios de requerimientos surgidas en la revisión.

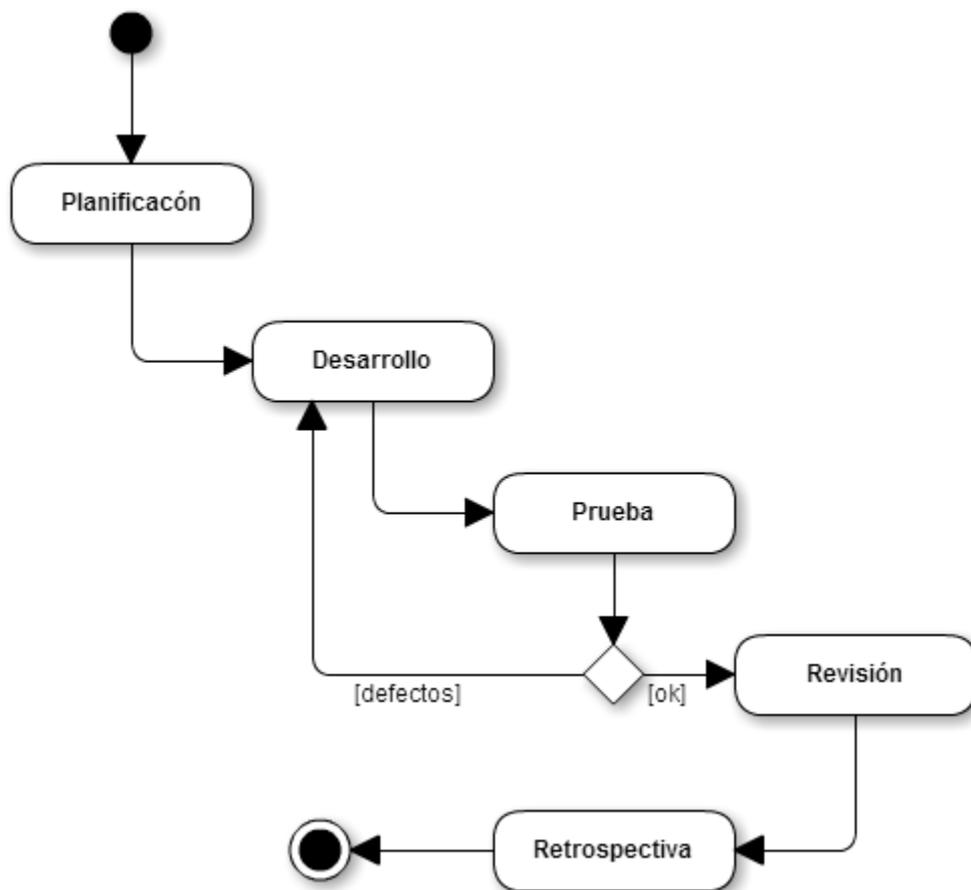
### 5.6.5.3 Entregables

- El producto desarrollado hasta ese momento, incluyendo todos los entregables asociados completados (documentación, análisis funcional, diseño técnico y código).

### 5.6.5.4 Cambios en objetivos/requisitos

Para que esta cláusula sea efectiva, UMIC se compromete a colaborar con los integrantes de SIMIC en todas las iteraciones y, especialmente, en las reuniones de especificación de requisitos y en las reuniones de demostración.

- Los cambios en prioridades de la lista de requisitos no implicarán ningún coste adicional a UMIC siempre que se mantenga el cómputo total de horas.
- UMIC podrá realizar adición de nuevos requisitos (tras las demostraciones), siempre que se retiren del contrato requisitos no iniciados que computen las mismas horas.



*Figura 1 Proceso que se siguió en la etapa de desarrollo, se resalta el bucle entre Prueba y Desarrollo en el cual no pasa a la etapa de revisión sin una aprobación.*

El proyecto culminó con una fase de “Cierre del proyecto” donde se finalizó con la documentación requerida, se evaluó la ejecución del proyecto y se extrajeron las lecciones aprendidas del mismo.

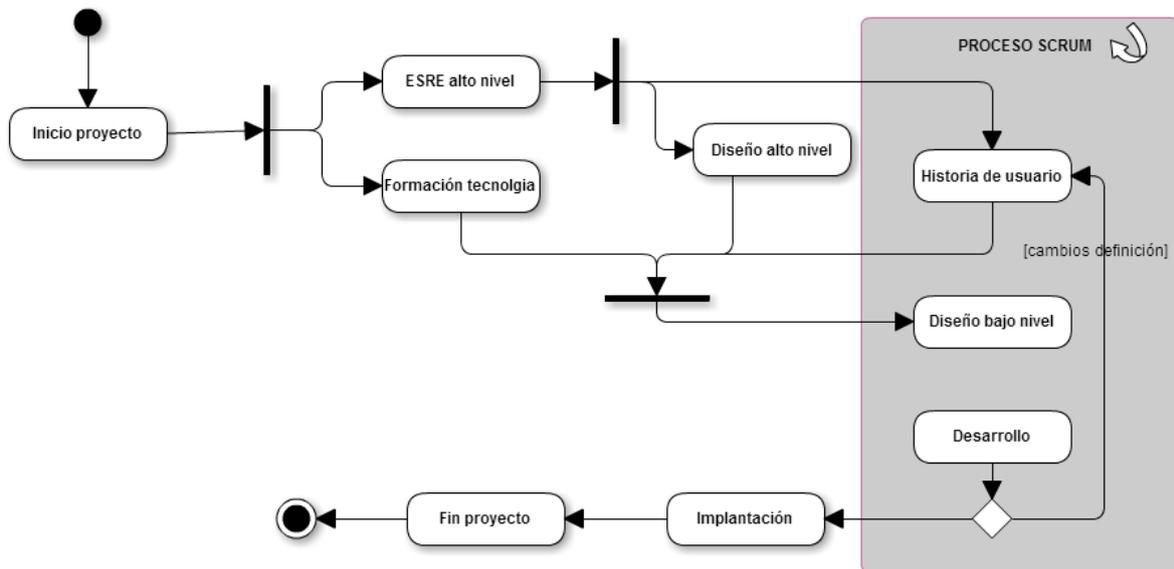


Figura 2 Modelo de ciclo de vida que se llevó para la ejecución del proyecto.

Durante el transcurso del proyecto se utilizó un ciclo de desarrollo de tipo incremental iterativo. Inicialmente se ejecutó una fase de constitución de proyecto, seguido de una fase de análisis y una de diseño ejecutadas en cascada. Luego se realizaron 7 sprints de desarrollo. La primera iteración realizada tuvo una duración más extensa que las siguientes a causa del aprendizaje de la tecnología. Los siguientes sprints fueron de 3 semanas en promedio. El objetivo de esta medida se fundamentó en la necesidad de realizar reuniones de retrospectiva con mayor frecuencia, permitiendo así tomar acciones correctivas con mayor celeridad en requerimientos importantes del sistema. Además, se buscó proponer objetivos a más corto plazo como forma de mantener constante la motivación del equipo.

Luego de la fase de construcción se procedió a la implantación del producto y la entrega de la documentación requerida.

## 6 Inicio del proyecto

### 6.1 Introducción

El objetivo de este capítulo es describir cómo se realizó la selección del proyecto a desarrollar y las necesidades de investigación y capacitación que el mismo implicó.

### 6.2 Presentación del proyecto

Al conformarse el grupo del proyecto se realizó una serie de reuniones con el objetivo de definir los criterios que debía cumplir el proyecto a comenzar. Entre lo discutido en dichas reuniones cabe destacar un interés en común entre los integrantes en que se cumplieran las siguientes características:

- Que el producto a desarrollar resulte un desafío para el equipo
- Un proyecto que permita explorar nuevas metodologías de trabajo
- Un cliente (integrantes de UMIC) comprometido con el proyecto
- El producto a desarrollar debe tener características innovadoras
- El producto tiene que ser robusto y estable.
- El producto tiene que satisfacer las necesidades las necesidades del cliente.

### 6.3 Investigación y capacitación

Teniendo en cuenta las características del proyecto, resultó clave definir un plan de formación que cubriera todas las áreas del proyecto donde el equipo necesitaba ampliar sus conocimientos. En particular, se definieron las siguientes necesidades de formación:

- Metodología SCRUM ([Anexo 3](#))
- Manejador de versiones GIT
- Normas y estándares de salud [7], [5], [8], [9] (XDS, EMPI, servidor terminológico, SNOMED)
  - AGESIC dictó cursos para el aprendizaje de los estándares.
- Protección de datos, base auditable
- Herramientas ETL para la migración de datos
- Framework AngularJS para JavaScript
- Diseño de interfaz “Material Design”
- Investigación de opciones de repositorios para poder realizar la implantación en la nube.

Estas necesidades se dividieron dentro del equipo según el rol a cumplir y el tiempo disponible de cada integrante. Al participar todos como desarrolladores, resultó indispensable también que cada uno se forme en profundidad en las tecnologías involucradas como son AngularJS, Spring Framework, MySQL, Hibernate y Bootstrap entre otras.

## 6.4 Organigrama del proyecto

Para este proyecto se definieron los siguientes roles, con sus responsabilidades asociadas:

- Gerente del Proyecto: encargado de la planificación del proyecto, seguimiento y control de su ejecución, y cierre del proyecto.
- Arquitecto: encargado del diseño arquitectónico y diseño de alto nivel de la solución.
- SCM: encargado de la gestión de la configuración del proyecto (configuración de herramientas y control de versiones) y su gestión.
- Ingeniero de requerimientos: encargado de relevar los requerimientos con el cliente y de elaborar una especificación.

El resto de las tareas, incluyendo la ejecución de pruebas, diseño detallado y desarrollo de la solución es una responsabilidad compartida por todos los integrantes del proyecto.

Los roles definidos se asignaron de la siguiente manera:

### GEPRO

- Titular: Viterbo García
- Suplente: Rodrigo González y Alejandro Cardone

### ARQ/ EXP. TEC

- Titular: Rodrigo González
- Suplente: Viterbo García y Alejandro Cardone

### SCM

- Titular: Alejandro Cardone
- Suplente: Viterbo García y Rodrigo González

### IREQ

- Todos

## 7 Estado del arte

En cuanto a la investigación realizada fue mediante ciertos criterios que el equipo considero importantes y que podrían ayudar a realizar un producto con las características planteadas, poder observar problemáticas y soluciones en el área de la salud desconocida por los integrantes del equipo.

### 7.1 Introducción

#### 7.1.1 Insuficiencia Cardíaca

[10] La IC (Insuficiencia Cardíaca) es un complejo síndrome clínico vinculado a alteraciones fisiopatológicas que ocurre cuando una anomalía de la función cardíaca determina la incapacidad del corazón para cumplir con las necesidades metabólicas tisulares, o lo hace a expensas de un incremento en las presiones diastólicas finales. Puede deberse a múltiples potenciales etiologías, siendo la posible vía final común que alcanzan las diferentes cardiopatías estructurales según su gravedad y tiempo de evolución.

Según un artículo publicado sobre el estudio del comportamiento de la IC en pacientes ingresados en el Centro Médico de Diagnóstico Integral de Venezuela las mayores dificultades en el manejo de los pacientes con IC surgen por la falta de comunicación entre las distintas partes interesadas. Por lo que es aconsejable una coordinación eficaz tanto entre atención primaria y especializada como entre enfermos, cuidadores y profesionales de la salud. Actualmente existen programas de gestión en los que personal de enfermería especializado en IC proporciona tal coordinación y continuidad de cuidados.

Se establece que es responsabilidad del médico proveer al paciente y a sus cuidadores con suficiente información y apoyo para que desarrollen y se adhieran a un plan de autocontrol de su enfermedad. Otras de sus funciones consisten en el seguimiento de guías clínicas, la monitorización del cumplimiento con el tratamiento, el seguimiento de los síntomas y complicaciones, el apoyo social y psicológico y la formación de otros profesionales de la salud. Existe evidencia clínica de que tales programas de gestión mejoran la supervivencia y calidad de vida de los pacientes, reducen los reingresos hospitalarios y acortan la estancia de las hospitalizaciones.

El papel del médico es fundamental para identificar a los pacientes con riesgo de desarrollar IC. Principalmente aquellos con antecedentes familiares de enfermedades cardiovasculares o cardiomiopatías. Así como enfatizar en la vigilancia y hacer el diagnóstico oportuno en pacientes con Hipertensión Arterial sistémica, Diabetes Mellitus, dislipidemia y síndrome metabólico ya que son pacientes con alta probabilidad de desarrollar Insuficiencia Cardíaca Congestiva (ICC).

#### 7.1.2 Unidad Multidisciplinaria de Insuficiencia Cardíaca

La Unidad Multidisciplinaria de Insuficiencia Cardíaca (UMIC) está integrada por un grupo de profesionales de la salud pertenecientes al Hospital de Clínicas “Dr. Manuel Quíntela” de la República Oriental del Uruguay. Conforman un equipo multidisciplinario de trabajo interdisciplinario bajo la concepción de los programas de manejo de enfermedades crónicas para atender a los pacientes portadores de IC crónica por disfunción sistólica [10]

El método de trabajo es el de una policlínica especializada. Participan médicos internistas, cardiólogos, psiquiatra, ecocardiografistas, nutricionista, licenciada en enfermería y asistente social.

### 7.1.3 Importancia de la unidad de insuficiencia cardiaca

La IC es la única enfermedad cardiovascular que ha aumentado su incidencia y prevalencia en los últimos años. Varios estudios han demostrado el impacto negativo en la evolución de la IC que tiene un pobre seguimiento médico así como un tratamiento inadecuado. Por otra parte en los últimos 10 años ha tomado fuerza el concepto de la IC como un síndrome clínico complejo que requiere el control periódico de un equipo multidisciplinario lo cual, llevado a la práctica, ha demostrado claros beneficios en la reducción de reingresos hospitalaria por esta patología.

La definición precisa de Unidades de IC aún no está completamente desarrollada. Su objetivo es el manejo integrado de la enfermedad. Busca, mediante un equipo multidisciplinario trabajando en forma coordinada, encargarse del control y seguimiento de los pacientes con IC en las diferentes etapas de la enfermedad. Fundamentalmente en las etapas más complejas y en los diferentes niveles de la cadena de atención sanitaria que vive el paciente ya sea extra e intrahospitalaria.

Estas unidades han mostrado a nivel internacional grandes beneficios. Fundamentalmente en temas relacionados al mayor conocimiento de la enfermedad por parte del paciente y su familia, adhesión al tratamiento, disminución franca de las hospitalizaciones y número de días de internación. Aún no se pueden extraer datos en cuenta al impacto en la sobrevida.

La dirección del Hospital de Clínicas, así como el Consejo de la Facultad de Medicina, definieron como imagen objetivo y estrategia de este Hospital ser un centro de “alta complejidad” en lo asistencial con una descentralización de la atención ambulatoria de primer nivel en coordinación con los servicios de atención externa en el ámbito de un programa nacional de salud. Incluyendo la coordinación de todos los efectores periféricos tanto sean de ASSE como de las intendencias u otros servicios. El Hospital cuenta para este objetivo con ciertas ventajas comparativas en el área cardiológica dada la existencia de un servicio de hemodinámica y un servicio de cirugía cardiaca con vasta experiencia. Complementado con una unidad de cuidados cardiológicos avanzados. A esto se agrega un servicio de electrofisiología que viene desarrollando con éxito los procedimientos más actuales en esta materia y un servicio de ecocardiografía y de medicina nuclear que facilitan el manejo diagnóstico.

En el hospital de clínicas existía hasta hace algunos años una gran diversificación en el seguimiento de los pacientes con IC en los distintos servicios clínicos de medicina y el departamento de cardiología.

Desde octubre del 2003 comienza a funcionar en el servicio de atención ambulatoria del departamento de Cardiología del Hospital la Policlínica de Insuficiencia Cardíaca (PIC) con el objetivo de generar pautas comunes de diagnóstico, tratamiento y seguimiento de los pacientes con IC. Así como también crear un registro para el manejo de datos estadísticos. Su creación se dio en el contexto histórico de la formación del Departamento Clínico de Medicina del Hospital de Clínicas que lleva a la natural unificación de criterios en el accionar de todos los servicios no quirúrgicos de este Hospital.

Es así que la PIC funciona desde su comienzo como un equipo multidisciplinario con médicos internistas y cardiólogos habiéndose integrado en forma paulatina una licenciada en enfermería, una nutricionista y una asistente social. Luego de estos sucesos se definió transformar las características de la PIC para pasar a llamarse Unidad Multidisciplinaria de Insuficiencia Cardíaca (UMIC)

#### 7.1.4 Objetivos de la UMIC

1. Lograr un manejo integral del paciente con insuficiencia cardíaca.
2. Establecer sistemas educativos hacia los pacientes para promover el autocuidado.
3. Servir como nexo entre la atención primaria y la atención de tercer nivel.
4. Disminuir el número de ingresos y reingresos así como la duración de la internación por IC abatiendo los altos costos de la atención de dicha patología.
5. Facilitar el manejo del paciente internado en el hospital en coordinación con el servicio en el que sean internados los pacientes en vistas a su resolución en plazos aceptables y disminuyendo los costos por internación evitando la superposición de estudios, logrando, Además, una continuidad en los regímenes terapéuticos personalizados en cada paciente.
6. Propender a lograr una estandarización en el diagnóstico y tratamiento de la ICC siendo motor de la promoción de pautas, su publicación y su correcto cumplimiento por parte de la población médica.

#### 7.1.5 Software en la medicina

En la medicina se maneja gran cantidad de información sobre los pacientes, proveniente de fuentes muy variadas. El uso de software para la manipulación de esta información aporta un acceso rápido, fiable (sin introducción de errores humanos en la comprensión de los formularios usados en la recolección de datos) y Además, la realización de respaldos que servirán para la posibilidad de pérdidas.

En Uruguay en su visión de Gobierno Electrónico, AGESIC [11] entendió prioritario abordar el área de la Salud con el fin de aportar una fuerza propulsora para la modernización de los procesos, avanzar en la aplicación de las políticas de gobierno en el área y viabilizar mejoras en la calidad de las prestaciones de salud recibidas por los ciudadanos.

Para implementar esta estrategia, se firmó un Convenio entre Presidencia, AGESIC, el Ministerio de Salud Pública y el Ministerio de Economía y Finanzas, a través del cual se creó el Programa Salud.uy fuerza ejecutiva a cargo de la instrumentación de la iniciativa.

El objetivo del Programa Salud.uy es fortalecer el Sistema Nacional Integrado de Salud (SNIS) apoyando la conformación de la red asistencial a través del uso de las Tecnologías de la Información y la Comunicación (TIC), creando herramientas que contribuyan a mejorar el acceso de los ciudadanos a servicios de salud de calidad.

Su finalidad es desarrollar los mecanismos, las acciones transversales y la infraestructura tecnológica necesaria, a los efectos de generar las condiciones para que los prestadores de salud puedan prestar sus servicios en forma integrada, complementaria y centrada en el usuario.

Para la primera etapa de implementación del Programa Salud.uy, el gobierno definió tres componentes en los que se trabaja con un enfoque único a nivel nacional: Historia Clínica Electrónica Nacional; Sistema Nacional de Teleimagenología y Fortalecimiento Institucional del Ministerio de Salud Pública como ente rector.

Historia Clínica Electrónica Nacional (HCEN), se trata de una herramienta que permite el almacenamiento, transferencia y consulta de información sobre la prestación de servicios de salud y datos clínicos del usuario. Entre sus cometidos está promover la continuidad de la atención sanitaria y la calidad del registro a través de la normalización de las estructuras clínicas, así como generar una base sustantiva de información clínica que permita complementar los servicios asistenciales y su prestación a distancia.

Uno de sus principales objetivos es garantizar que la información clínica vital del ciudadano esté disponible y accesible para el profesional de la salud, de forma oportuna, segura y en línea.

Sistema nacional de teleimagenología, el objetivo es desarrollar e implantar aplicaciones de TIC en la gestión de servicios de análisis y diagnóstico en base a imágenes radiológicas. La digitalización de los servicios de diagnóstico por imágenes permite desarrollar en el país una red integrada de prestaciones que acompaña la modernización de la atención. En este sentido, la teleimagenología hace posible que la imagen obtenida en la localidad donde se encuentra el equipo y el paciente sea informada a través de las telecomunicaciones por un médico radiólogo en otra parte del país.

## 7.2 Casos de estudio

### 7.2.1 Caso 1

#### **A Domain-Oriented Software Development Environment for Cardiology**

Los equipos de desarrollo de software encuentran problemas para trabajar en áreas que no le son familiares. Según las experiencias de la Universidad Federal de Río de Janeiro, descritas en este artículo [12], los principales problemas al desarrollar software para el tratamiento de enfermedades cardiológicas pasa por no estar familiarizados al área. En el

artículo se plantea el problema de lo tedioso que se le resulta a los especialistas en cardiología explicar a los desarrolladores conceptos básicos. Además, a esto se le suma la reiteración de las explicaciones por el constante cambio que en general existe en los integrantes de un equipo de desarrollo.

Para hacer frente a ese problema crearon Domain-Oriented Software Development Environment (DOSDE), un modelo de desarrollo de software orientado al dominio en que se está trabajando, con el fin que este pueda dar soporte en el desarrollo de sistemas de software en un dominio específico. DOSDE considera el conocimiento específico que le fue incorporado para guiar a los desarrolladores en las diferentes fases del proceso de desarrollo del software.

Este modelo de desarrollo de software requiere dos características esenciales:

- a) tener dominio de teoría.
- b) el uso de este conocimiento durante el desarrollo de software.

En cuanto al “Dominio de Teoría” propuesto por DOSDE, es el modelo de dominio que contiene la información necesaria para ayudar a los desarrolladores. El Dominio de Teoría se puede dividir en sub-teorías. En el caso de la cardiología se podría subdividir, por ejemplo:

- i. anatomía del corazón.
- ii. resultados.
- iii. terapia, diagnóstico y patologías.

Las tareas identificadas para la cardiología son: diagnóstico, planificación terapéutica, simulación y monitoreo. El mapeo de las teorías con las tareas proporciona una mejor visión de los conceptos que son importantes en el dominio y en los cuales el equipo de desarrollo debe poner más atención. Por ejemplo, para la tarea de diagnóstico, es importante tener en cuenta los resultados, patología y sub-teorías de diagnóstico.

Actualmente el equipo está trabajando en extensiones y algunas de las herramientas necesarias para hacer el DOSDE de Cardiología una realidad.

## 7.2.2 Caso 2

### **CasSelf-Reported “Communication Technology” Usage in Patients Attending a Cardiology Outpatient Clinic in a Remote Regional Hospital**

[13] Este estudio tiene como objetivo evaluar la percepción del uso, y actitudes hacia, tecnologías de la comunicación (teléfono móvil y mensajes de texto, email y la web) en pacientes que acuden a una clínica de cardiología con el fin de observar y guiar el futuro rediseño de los servicios de salud.

Se llevó a cabo en el hospital Raigmore en el norte de Escocia, el mismo atiende alrededor de 300.000 pacientes, muchos de estos pacientes presentan dificultades para asistir a la clínica por la barrera geográfica.

Se realizó una encuesta a los pacientes mayores de 16 años que concurrieron a la clínica en cierto rango de fechas. El diseño del cuestionario se realizó de forma iterativa en varias etapas, hasta llegar a la versión final.

A los datos obtenidos se le realizaron análisis estadísticos para evaluar la influencia de la edad, género y distancia al hospital, en el uso de internet, teléfono móvil, mensajes de texto y email. También se realizaron pruebas para ver las frecuencias de uso.

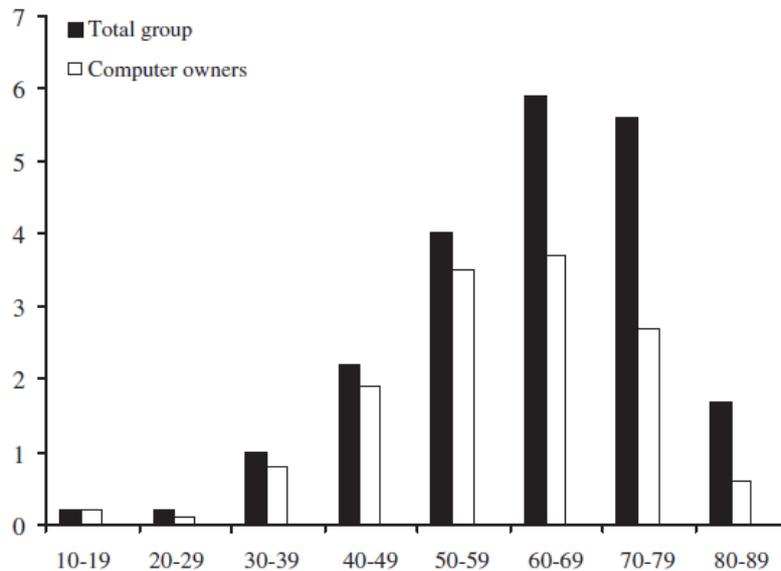


Figura 3 Se muestran para los diferentes rangos de edad la comparación entre el total de los pacientes y los pacientes que usan computadoras

TECHNOLOGY USED	NUMBER (%)				
	DAILY	WEEKLY	MONTHLY	NEVER	UNREPORTED
E-mail	74 (36)	26 (12)	17 (8)	70 (32)	34 (15)
Internet	67 (30)	53 (24)	16 (7)	64 (29)	21 (10)
Mobile phone	102 (46)	31 (14)	25 (11)	29 (13)	34 (15)
Mobile phone text	51 (23)	36 (16)	14 (6)	76 (34)	46 (21)

Figura 4 Frecuencia del uso de tecnología

Según los datos obtenidos se pudo realizar la discusión de la influencia de la edad, y de la zona (rural o urbana) de afincamiento de los pacientes para el uso de la tecnología de comunicación.

Se puede determinar que gran parte del uso de la tecnología de la comunicación va a depender de los pacientes, en el estudio se pudo determinar que solo el 62% de los pacientes tenía acceso a una computadora en su hogar, el 50% tenían conectividad de banda ancha. Las consultas médicas cara a cara son una forma común de evaluación y seguimiento de los pacientes y han estado en vigor durante muchos años, pero muchos de los pacientes a los cuales se le agendan consultas de rutinas, ven el asistir a la clínica con poco o ningún resultado positivo. Todos estos inconvenientes encontrados se acentúan aún más en los pacientes que viven en zonas rurales.

Por todo esto el artículo plantea a modo de discusión el beneficio del uso de la tecnología en la comunicación entre los pacientes y el personal médico. Se presentan ejemplos de beneficios, como no perder el día laboral por asistir a la consulta, descongestionar las salas de espera, el aumento de la adhesión al tratamiento por medio de notificaciones y alertas a los pacientes, Además, en zonas rurales se propone que la comunicación con el paciente por intermedio de las clínicas rurales si es que éste no posee acceso a banda ancha (las denominadas policlínicas en Uruguay) en donde se tenga acceso por ejemplo a una aplicación web.

Respecto a la aceptación que hay por parte de los pacientes y médicos en el uso de la tecnología en la comunicación, el artículo hace referencia a que los pacientes aceptaban recibir notificaciones respecto a citas u otro tipo de notificación por parte del médico. Los beneficios que aportaría una comunicación más frecuente con el médico parecían ser más importante que posibles reducciones sutiles en la calidad de la consulta. La seguridad y confidencialidad son una preocupación especial, en las consultas a distancias se puede perder cierta información, se podría obtener luego mediante una consulta personal.

En pacientes de edades avanzadas, el estudio demostró que son menos propensos a utilizar tecnologías de comunicación, y existe la preocupación de que los pacientes de edad avanzada se nieguen a los beneficios que implicaría el uso de tecnologías. Sin embargo, estudios han demostrado que si se da apoyo, las nuevas tecnologías pueden ser utilizadas con éxito incluso en esta población.

Como conclusión el estudio identificó que el uso de la tecnología de la comunicación no está muy extendida en la atención de la cardiología. Uno de las barreras más importantes es la edad de la población de pacientes.

### 7.2.3 Caso 3

#### **Software para diagnosticar la insuficiencia cardiaca**

[14] Un estudiante (Fabián Torres Robles) del Instituto Politécnico Nacional de México, realizó junto con la institución un software para el diagnóstico de la insuficiencia cardiaca. El sistema se centra en el procesamiento de imágenes digitales, técnicas de lógica difusa y redes neuronales, y con estos métodos puede realizar el diagnóstico en menos de un minuto. El software prevé ayudar a los médicos que no tienen la especialidad en Cardiología. Los médicos de primer contacto podrían establecer un diagnóstico con un alto grado de confiabilidad.

### 7.2.4 Caso 4

#### **Sistema de Control Inteligente para pacientes de insuficiencia cardiaca**

[15] SIMS (Self-adaptive Intelligent Monitoring System for Supporting Home-based Heart Failure Patients) es un sistema que toma ventajas de una serie de tecnologías avanzadas

de inteligencia de software, recuperación de datos, minería de datos y base de datos. SIMS es capaz de proporcionar una serie de funciones avanzadas. Se puede priorizar eficazmente a los pacientes, proporcionar recomendación automática para el control de la frecuencia y la evaluación de riesgos, y llevar a cabo análisis de correlación con el fin de identificar las relaciones entre los factores externos y el desarrollo de la condición del corazón del paciente. Todas estas funciones pueden reducir significativamente la cantidad de consultas de un paciente, aumentar la confianza del paciente sobre su estado de salud y mejorar su calidad de vida en general.

El desarrollo de SIMS tuvo como fin abordar tres limitaciones asociadas los servicios IC:

1. La falta de priorización automática de pacientes y evaluación de riesgos. Los servicios de IC carecen mecanismos automáticos de priorización y evaluación de riesgos de pacientes. La priorización de pacientes y la evaluación de riesgos médicos apuntan a priorizar pacientes en función de su estado de salud y tratamientos con el fin de optimizar el despliegue de recursos para los pacientes que están más necesitados.
2. La falta de predicción automática y recomendación. Los servicios de IC actuales no proporcionan la predicción automática y la recomendación con el fin de apoyar la toma de decisiones de alto nivel. SIMS proporciona seguimiento selectivo basado en datos de los pacientes (auto-reporte), los datos de seguimiento, el control del cumplimiento y a partir de esto ofrece recomendaciones importantes en cuanto a la frecuencia de las pruebas que los pacientes tienen que deben seguir y la medicación necesaria que debe proporcionarse. Este trabajo se basó en un estudio sobre la recomendación en base a dos criterios: la necesidad de servicios de atención de emergencia y la probabilidad de la producción de un resultado sub-óptimo, mediante la explotación de técnicas de minería de datos de tratamiento. SIMS ayuda a los médicos a evaluar el riesgo de deterioro clínico a través de múltiples dominios.
3. La falta de descubrimiento inteligente de patrones de correlación para los factores externos. Los servicios de extensión de enfermedades crónicas actuales no proporcionan funciones de análisis que puedan ofrecer evidencia basada en datos y conocimiento respecto a las correlaciones entre las condiciones del paciente y varios factores de estilo de vida. Estos factores externos influyentes incluyen la nutrición, la obesidad, el consumo de alcohol, ejercicio físico, la calidad del sueño, y otros.



Figura 5 Arquitectura de SIMS, se observa la comunicación entre SIMS, los pacientes y diferente personal de salud

### 7.2.5 Caso 5

#### **Insuficiencia Cardíaca Monitoreo Remoto: Evidencia de la evaluación retrospectiva de un Programa de Monitoreo Remoto del mundo real**

[16] Dada la magnitud del aumento de la mortalidad por IC, los enfoques multidisciplinarios, en forma de programas de gestión de enfermedades y otros modelos de integración de la atención, se recomienda para optimizar los resultados del tratamiento el uso de la monitorización remota, ya sea como soporte telefónico estructurado o tele monitorización o una combinación de ambos. Aunque el control remoto de pacientes se está convirtiendo rápidamente en una parte integral de muchos programas de control de enfermedades, los estudios que informan sobre la evaluación de los programas de monitoreo remoto de insuficiencia cardíaca en el mundo real son escasos.

Se realizó un estudio con el objetivo evaluar el efecto de un programa de tele monitorización de insuficiencia cardíaca, Connected Cardiac Care Program (CCCP), de hospitalización y mortalidad en una base de datos de revisión retrospectiva de las historias clínicas de los

pacientes con insuficiencia cardíaca que reciben atención en el Hospital General de Massachusetts.

Los pacientes inscritos en el programa de vigilancia de la IC de CCCP fueron emparejados uno a uno con los pacientes de cuidados habituales. Los pacientes de cuidados habituales recibieron atención de situaciones clínicas similares a los pacientes CCCP y fueron identificados a partir de un gran registro de datos clínicos. El criterio principal de valoración fue la mortalidad por todas las causas y las hospitalizaciones durante la duración (4 meses) del programa. Los resultados secundarios incluyeron las tasas de hospitalización y mortalidad (obtenidos mediante el seguimiento de los pacientes durante un período adicional de 8 meses después de la finalización del programa, duración total de 1 año), riesgo de múltiples hospitalizaciones y la duración de la estancia.

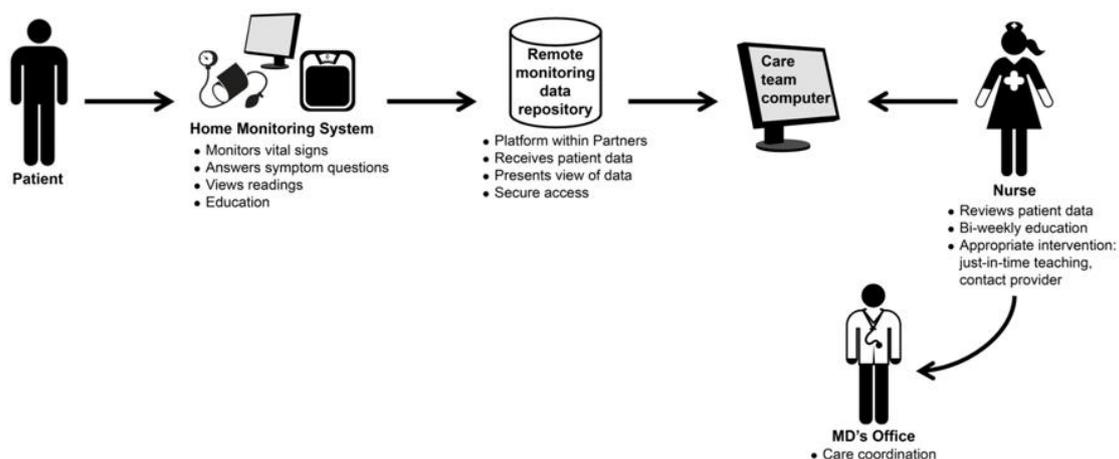


Figura 6 Flujo de funcionamiento de CCCP, se aprecia los diferentes medios que puede usar el paciente para su monitoreo

Un total de 348 pacientes fueron incluidos en los análisis de eventos. Las cantidades de hospitalizaciones previas para programar la inscripción no difirieron significativamente por grupo. En comparación con los controles, las tasas de hospitalización disminuyeron en los primeros 30 días de inscripción en el programa. El efecto diferencial en las tasas de hospitalización se mantuvo constante hasta el final del programa de 4 meses. El programa también se asoció con menores tasas de mortalidad al final del programa de 4 meses. Al extender 8 meses de seguimiento después de la finalización del programa, no se observó efectos beneficiosos extras del programa CCCP en la mortalidad o las hospitalizaciones.

CCCP se asoció con tasas de hospitalización significativamente más bajas de hasta 90 días y las tasas de mortalidad significativamente más bajas más de 120 días de programa. Sin embargo, estos efectos no persistieron más allá de la duración del programa de 120 días.

## 7.2.6 Caso 6

**The American College of Cardiology-National Cardiovascular Data Registry™ (ACC-NCDR™): building a national clinical data repository**

[17] El diagnóstico cardíaco, la caracterización e intervenciones coronarias percutáneas (PCI) son componentes críticos de la diagnóstico y tratamiento de pacientes con enfermedad en la arteria coronaria.

El PCI ha evolucionado rápidamente, llegando a ser un gran éxito en la revascularización miocárdica. Con la llegada de nuevos dispositivos y terapias farmacológicas, el riesgo de resultados adversos asociados al PCI ha disminuido.

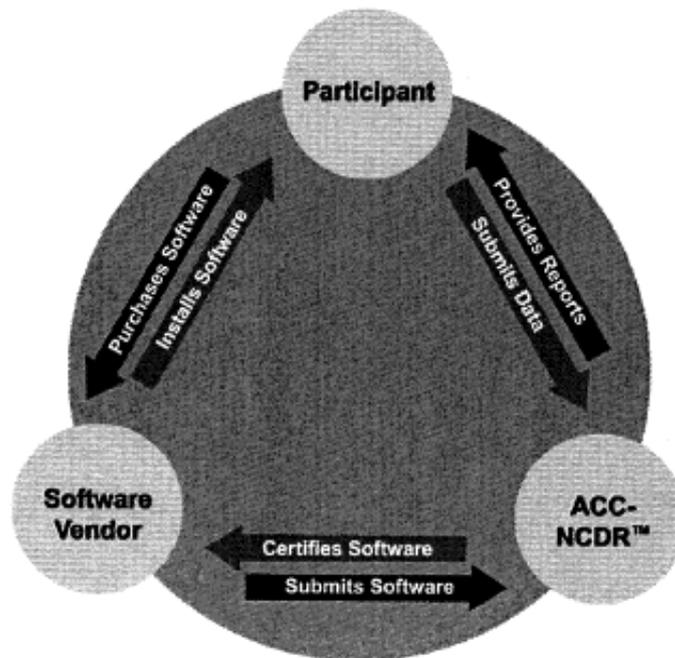
Durante más de 24 años se han estado registrando datos a gran escala, utilizados para investigar sobre PCI y, en menor medida, cateterismos cardíacos. Se estudian las características del paciente, procedimientos, tratamientos post-procedimiento, resultados inmediatos (en el hospital), y los resultados a largo plazo. Muchos análisis diferentes de estos datos se han realizado en un esfuerzo para mejorar la calidad de la atención que reciben los pacientes coronarios.

Se puede observar que entre los mayores problemas para el análisis de datos cardiovasculares es el frecuente cambio en los tratamientos, confundiendo terminología, definiciones no estándar, y la falta de estándares de datos aplicados. Reconociendo estos desafíos el Colegio Americano de Cardiología (ACC) tomó como desafío crear una base de datos que estandariza qué y cómo se recoge la información para los pacientes que recibieron cateterismos cardíacos y PCI. Esta iniciativa de ACC se inició con la formación de un comité de base de datos bajo el liderazgo de Suzanne B. Knoebel, MD, FACC. El comité exploró diversas estrategias para mejorar la recolección y evaluación de datos cardiovasculares. Como resultado, en 1990 el ACC conjuntamente con Summit Medical Systems [18] propusieron distribuir software para gestionar electrónicamente la recolección de datos importantes en un repositorio central. Logrando obtener los datos actualizados a través de 5 pasos fundamentales:

1. El establecimiento de National Cardiovascular Data Registry™ (ACC-NCDR™), que se encuentra en la Casa del Corazón, Bethesda, MD (sede del ACC).
2. Permitir que la certificación de software ACC abierta a todos los proveedores comerciales viables y comprometidos.
3. La contratación de personal clínicamente capacitado para apoyar y reclutar participantes ACC-NCDR™.
4. Uso de la ACC-NCDR™ como una instalación de procesamiento de datos y los datos de retroalimentación de la calidad.
5. El aumento de la frecuencia de los envíos de datos participante de cada año a trimestral, lo que permite una información más oportuna.

El objetivo principal de la ACC-NCDR™ es convertirse en la base de datos comparativa más completa tanto de pacientes hospitalizados, ambulatorios y atención cardiovascular disponible en los EE.UU. Desde que las instituciones comenzaron a inscribirse en el ACC-NCDR™ en noviembre de 1998, ha crecido hasta representar a 350 instituciones (hospitales, laboratorios externos, entre otros).

Los contratos ACC-NCDR™ con proveedores de software comerciales que han accedido a proporcionar software certificado (adhiriéndose a los estándares de datos clínicos y de codificación). De esta forma las instituciones participantes logran obtener un software que les permite recoger de forma sistemática datos sobre cuidados prestados a los pacientes sometidos a cateterismo cardiaco diagnóstico o ICP certificados.



*Figura 7 Proceso de comunicación entre los participantes del proyecto*

Después de obtener algún software certificado, los participantes son capaces de enviar datos al ACC-NCDR™ y las instituciones son capaces de obtener reportes. La base de datos utiliza identificadores cifrados, un sistema que preserva la confidencialidad del paciente y el médico al tiempo que permite una vista de los datos, orientada al paciente. Además, los datos son almacenados en un sistema de información seguro en el ACC.

## 7.3 Software actual

### 7.3.1 Kareo

[19] Es un software de gestión médica y que incluye facturación. La compañía ofrece una suite basada en la nube con productos y servicios para la gestión de la HCE, gestión de la práctica médica y de facturación de servicios. Este sistema es especialmente adecuado para las prácticas de las pequeñas empresas.

Kareo ofrece tres niveles de servicios de gestión de la práctica:

1. Patient Engagement/Practice Marketing: una plataforma del paciente basado en la nube que permite a los usuarios programar citas, crear recordatorios móviles, y gestionar su reputación online.
2. Free EHR: basada en la nube, compatible con iOS, permite a los usuarios revisar los registros de pacientes, visitas carta de pacientes, prescribir medicamentos e incentivar el “uso significativo”.
3. Practice Management: un servicio de gestión de la práctica combinando con Free EHR con funciones de apoyo a la gestión de la práctica, incluyendo la gestión de documentos, procesamiento de reclamos, el manejo y almacenamiento de la información del paciente, el envío de reclamo de seguros y la generación de balances.

4. Billing Services: servicios de Practice Management combinados con seguimiento financiero adicional y servicios de gestión de reclamo tercerizados.

De la experiencia expuesta por Kareo en su blog podemos rescatar una serie de consejos, por ejemplo la hora de agendar una cita con un paciente recomiendan tener en cuenta las siguientes parámetros, nombre del paciente, número de teléfono celular, email y cumpleaños del paciente (el hecho de decir feliz cumpleaños al paciente en su día hace que se sienta más importante).

También han dado su experiencia sobre lo que según ellos ha mejorado el uso de la HCN, alguno de estos beneficios que enumeran son: legibilidad de notas, accesibilidad a gráficas (es fácil ver y analizar los datos en forma de grafica), múltiples usuarios (con el registro en papel un solo médico a la vez podía acceder a la historia del paciente), recuperación por desastres (muchas personas tienen miedo a un accidente en la HCN, si bien es un tema delicado un "desastre" en una HCN es mucho menos grave que una en papel, incluso es mucho más fácil crear copias de seguridad y almacenarlas en distintos sitios).

### 7.3.2 MediTouch

[20] MediTouch es una aplicación web basada en un sistema de registros médicos con capacidades de personalización e interoperabilidad. Esta empresa de software médico también ha integrado el software de gestión en la práctica.

Entre los servicios que ofrece brinda la posibilidad de que los pacientes pueden utilizar un portal para realizar tareas, tales como solicitar repetición de recetas.

Los pacientes también pueden programar citas a través del software, usar herramientas de mensajería y consultar su información médica, incluyendo los resultados de laboratorio.

El sistema MediTouch conecta con casi 200 laboratorios en todo Estados Unidos. Esta interoperabilidad permite enviar pedidos directamente a un laboratorio y recibir resultados de vuelta a través del sistema. También incluye una característica ePrescribe por la que puede enviar recetas a una farmacia elegida por el paciente. El sistema también integra con SureScripts, una red ePrescribing que muchos sistemas de HCE populares utilizan para establecer contactos con las farmacias y otras instalaciones. Permite descargar y realizar un seguimiento de la historia completa de medicamentos de un paciente.

La función de receta electrónica también está disponible en los dispositivos móviles.

MediTouch atiende actualmente a aproximadamente veinte especialidades (incluyendo la cardiología), y su sistema se puede personalizar para adaptarse a otras especialidades o prácticas específicas. Al iniciar el sistema en un principio, se tiene acceso a las plantillas precargadas que se pueden personalizar a las áreas que se necesitan para lograr un flujo de trabajo.

MediTouch es un sistema basado en la nube, por lo que el usuario no tiene que gestionar el mantenimiento de hardware, como servidores, para albergar el sistema por sí mismo, y tampoco se tiene que actualizar el software manualmente.

### 7.3.3 AdvanceMD

[21] Es un Electronic Health Record (EHR) y solución de la gestión en la práctica que dispone la programación, documentación, recetas electrónicas, gestión de ingresos, y las capacidades de información financiera. Los usuarios obtienen una experiencia informática de grado industrial con una gran velocidad, acceso seguro y soporte. No hay software que instalar o mantener. Actualizaciones automáticas hacen que la gestión de los cambios regulatorios, como la implementación de CIE-10 sea más fácil.

Esta plataforma EHR Proporciona acceso a los datos del paciente en cuestión de segundos y carga los resultados de laboratorio en las historias clínicas. En su nube y a partir de un inicio de sesión único se brinda un acceso inmediato desde cualquier lugar, seguridad en los datos, lo que permite a los usuarios mantenerse conectados.

AdvancedMD permite al personal completar el registro de entrada en menos de un minuto, la transferencia de datos de los pacientes desde cualquier sistema, y brinda la posibilidad de personalizar para adaptarse a las necesidades de cada institución. Advanced Insight Reporting proporciona herramientas para medir el desempeño financiero de los médicos y las prácticas; con una interfaz donde se tiene que solamente arrastrar y soltar hace que la generación y ejecución de reportes sea fácil, rápida y precisa.

### 7.3.4 NueMD

[22] Esta plataforma ofrece una suite de software y servicios para equipos médicos de distintos tamaños. Ofrece posibilidad de programación de citas, así como software de gestión/facturación médica y HCN.

También ofrece servicios de facturación y ofrece la experiencia por un gran número de Certificados.

NueMD EHR es certificada como ONC-OCTA, ofrece una funcionalidad robusta para flujos de trabajo clínicos. También incluye la receta electrónica, el seguimiento del informe de laboratorio y fax seguro.

### 7.3.5 iSalus Healthcare

[23] iSalus ofrece un software de gestión médica que incluye una solución para la planificación y facturación médica. Con el fin de que la práctica médica se pueda realizar fácilmente con los pacientes desde el momento en que se registren hasta al momento de enviar las reclamaciones de seguros y recibir el reembolso.

Como software de planificación médica, el sistema de gestión iSalus ofrece muchas opciones de personalización. Por ejemplo personalizar la vista de calendario para mostrar los horarios de acuerdo a la ubicación, el médico y o día o la semana. De esta forma trata que el personal de recepción puede acceder rápidamente a la información del paciente directamente desde el programa.

Este software también les brinda a los pacientes el acceso a un portal. Los pacientes pueden rellenar formularios demográficos elegidos por personal médico. Esto significa que los pacientes no tienen que gastar tiempo llenando formularios en papel cuando vienen a sus citas.

También brinda a los pacientes la posibilidad de que puedan ver sus declaraciones, solicitar citas y utilizar el sistema de mensajería segura del portal para comunicarse con los médicos y el resto del personal.

Una de las características más importantes de este software es su sistema de facturación, pero éste no es algo que sea relevante para nuestro trabajo.

iSalus al igual que la mayoría de los software de estas características que hemos encontrado sus servicios son ofrecidos en la nube de esta forma el personal médico está desentendido del hardware, servidores, o de contratar a alguien para mantener el sistema.

Se accede al software principal a través de un navegador web. La compañía también ofrece una aplicación para iPhone con funciones más reducidas.

### 7.3.6 Epi Info

[24] Epi Info es una suite de herramientas de software que proporciona funcionalidad básica sin la complejidad o el alto costo de grandes aplicaciones empresariales. Está orientada a médicos, enfermeras, epidemiólogos y otros trabajadores de la salud pública carecen de experiencia en tecnología de la información y a menudo tienen una necesidad de herramientas sencillas que permiten la rápida creación de instrumentos de recolección de datos y análisis de datos, visualización y presentación de informes utilizando métodos epidemiológicos.

Epi Info es flexible, escalable y permite la recogida de datos, análisis estadísticos avanzados, y un sistema de información geográfica (SIG) con capacidad de mapeo.

Epi Info se utiliza en todo el mundo para la evaluación rápida de brotes de enfermedades; para el desarrollo de los pequeños a los sistemas de vigilancia de enfermedades de tamaño medio; como componentes especiales integrados con otros sistemas de información de salud pública a gran escala o de toda la empresa; y en la formación continua de los profesionales de salud pública el aprendizaje de la ciencia de la epidemiología, herramientas y técnicas.

El módulo Diseñador de formularios de Epi Info permite a los usuarios crear cuestionarios y formularios de entrada de datos en Epi Info. Con Diseñador de formularios, los usuarios ponen preguntas y campos de entrada de datos en una o muchas páginas y adaptan el proceso de entrada de datos con los patrones de salto condicional, validación de datos, cálculos y personalizados programados por el usuario utilizando el comprobador de código del Diseñador de formularios.

El módulo de Enter de Epi Info crea automáticamente la base de datos del cuestionario en el Diseñador de formularios. Los usuarios introducen los datos, modifican datos existentes o buscan registros. Con Enter, se muestran los formularios y los usuarios realizan la entrada

de datos, mientras que el Código Comprobar valida los datos o realiza cualquier cálculo automático que se especificaron en Diseñador de formularios.

El módulo de análisis se utiliza para leer y analizar los datos introducidos con el módulo de introducir datos o importados de 24 formatos de datos diferentes. Epidemiológicos, estadísticas, tablas, gráficos y mapas se producen con comandos simples.

El módulo Epi Map muestra mapas geográficos con datos de Epi Info. Epi Map muestra archivos shape que contiene los límites geográficos en capas con datos de los resultados del módulo de análisis.

## 7.4 Observaciones

Luego de la investigación realizada, el equipo pudo contar con los siguientes conocimientos.

Uno de los principales problemas en la elaboración de software en la medicina es la intercomunicación que debe existir entre los profesionales de medicina y computación. La diferencia de conocimientos de las diferentes áreas y vocabulario usado por las los integrantes cada parte, hace que la comunicación no sea una tarea fácil.

Otro factor importante a tener en cuenta para el desarrollo de software en medicina es el avance que puede producir, provocando cambios en el proceso hasta el momento realizado por los profesionales, aparición de nuevos conceptos y falta de estándares de datos.

Al desarrollar software destinado a pacientes es necesario contar la evaluación previa de esta población respecto al uso de tecnología y el acceso que pueda llegar a tener. Una barrera importante puede ser es la edad de las personas.

Se pudo comprobar que el uso de software para el seguimiento de pacientes disminuye las tasas de hospitalización y mortalidad, ya que los médicos pueden trabajar de forma conjunta y lograr un mejor control sobre el paciente. Aunque la mayoría no son capaces de realizar: priorización automática de pacientes, evaluación de riesgos, predicciones automáticas y recomendaciones.

**Como observación importante extra de investigación** es que 2 integrantes del grupo realizaron el “seminario de ingeniería biomédica” (que se dicta en facultad de ingeniería como materia de posgrado) para poder incorporar conocimiento del área de salud. En el seminario se pudo extraer ideas para poder incorporar en el proyecto. Una de ellas fue extraída del sistema PRAXIS, sistema que se adapta a cada usuario (médico), en el proyecto se usó esa idea para la funcionalidad del ingreso de datos nutrición con el objetivo de facilitar la gran cantidad de datos a la nutricionista.

# 8 Ingeniería de requerimientos

## 8.1 Introducción

En esta etapa se trabajó en el relevamiento de las principales funcionalidades del sistema junto a los integrantes de UMIC.

El ESRE (documento de especificación de requerimientos) fue desarrollado a partir de reuniones con los responsables de la UMIC, integrantes de AGISC y tutor del proyecto.

## 8.2 Estrategia de relevamiento

Consistió en la conceptualización de la visión del producto de los integrantes de la UMIC. Se realizó en base a reuniones con integrantes de la UMIC y documentos usados para la atención de los pacientes.

Se enfatizó en obtener las necesidades particulares de los integrantes de la UMIC y en buscar la forma en que la solución pueda simplificar y mejorar su operativa.

Durante el desarrollo del ESRE también se validaron con los integrantes de la UMIC los requerimientos principales del sistema a través de MockUps. El objetivo de ello fue evaluar que estos requerimientos principales estuvieran contemplados.

Finalmente los requerimientos fueron trasladados a historias de usuario debido a la metodología de trabajo elegida. El Product Backlog era en cada sprint trasladado a un Sprints Backlog y este último priorizado con la ayuda de los integrantes de UMIC.

## 8.3 Situación inicial

En la actualidad los integrantes de la UMIC cuentan con formularios físicos estandarizados que ayudan al médico a tener una consulta ordenada con el paciente.

La consulta de cada paciente sigue un flujo por el que son atendidos por diferentes integrantes de UMIC. El problema que encontró el equipo es que los formularios estaban desactualizados y contenían información redundante.

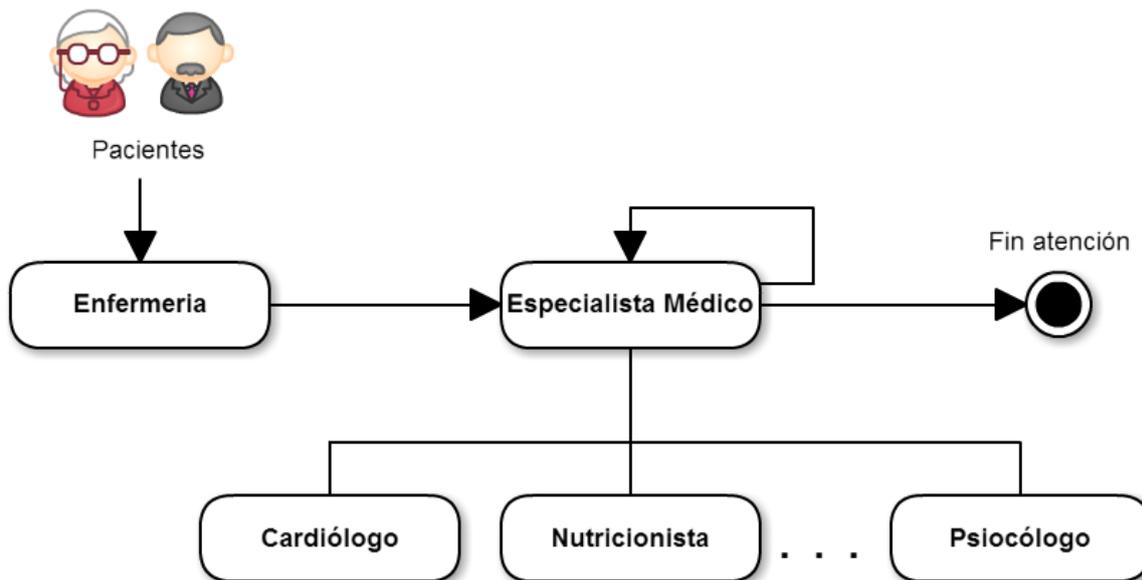


Figura 8 Proceso de atención de la UMIC

Con SIMIC cada integrante de UMIC va a ingresar los datos relevantes del paciente que le corresponde y podrá observar de forma unificada estos datos obteniendo así una visión global del paciente.

Además, de estos formularios, la unidad tiene implementada una versión del software Epi Info en la que van a ingresando los datos recolectados. Un problema que los integrantes de la UMIC plantearon al inicio, fue poder sacar estadísticas a partir de la información recabada para mejorar por ejemplo en la gestión de la atención. Epi Info no les permitía realizar esto de una forma clara. SIMIC apunta a resolver este problema.

## 8.4 Inconvenientes

A continuación se listan los principales inconvenientes que presentan los integrantes de la UMIC en la atención y seguimiento de los pacientes de la unidad.

**Pérdidas en la atención de los pacientes:** los pacientes no se adhieren al tratamiento y no toman la importancia que esto implica en su enfermedad. El control de los pacientes se realiza mediante comunicación telefónica una vez que se observa que el paciente no asiste a las consultas.

**Controles en cuanto a los resultados de los exámenes:** al tener la información de forma física mediante formularios en papel es difícil poder tener una visión global del paciente a partir de los datos.

**Consultas en periodos largos:** muchas veces los pacientes no asisten a consultas durante largos periodos de tiempo, los médicos quisieran poder tener el valor de ciertas variables importantes en periodos más cortos (no esperar a la próxima consulta).

**Contar con indicadores en tiempo real:** indicadores que ayuden en la gestión de la atención de los pacientes por parte de la UMIC. Actualmente la unidad no cuenta con estos indicadores en tiempo real, se tiene que recurrir a personal externo para obtener indicadores que ayuden a tomar decisiones en la gestión de su atención.

**Inconvenientes en el mantenimiento y fidelidad de la base de datos:** los integrantes de la UMIC tienen inconvenientes a la hora de realizar estadísticas en cuanto a los datos que actualmente se registran en el sistema Epi Info. El proceso para el registro a entender del equipo tiene mucho overhead. Cuenta con una instancia donde el médico en el momento de la atención del paciente por intermedio de diferentes formularios completa información (se usan muchos formularios con información repetida). Luego los formularios llegan a un integrante de la unidad que realiza el ingreso de los datos al sistema Epi Info. Durante la etapa de migración se tuvo un mejor acercamiento al problema de la fidelidad de los datos, se pudo encontrar errores de incongruencia. Pueden ser ejemplos de estos errores, la siguiente lista:

- Diferentes pacientes con el mismo documento de identidad
- Controles sin fechas
- Diferentes nombres para la misma medicación

## 8.5 Situación proyectada

La solución que se desarrolló permite centralizar toda la información que se obtiene actualmente a través de documentos físicos a un sistema, lo que mejora sustancialmente el seguimiento del paciente. Se realiza de forma remota y en tiempo real. Permitiendo ver una visión global del paciente, observar graficas de sus variables, entre otras cosas. Además a partir de los indicadores de gestión del proceso de atención el medico podrá tomar decisiones al médico al mismo tiempo que se está realizando la consulta.

El sistema contará con una versión web y una versión que se ejecutará en dispositivos móviles Android. La versión web será accedida por médicos y estudiantes de la cathedra dictada por la UMIC, tendrá diferentes perfiles de usuarios dentro de los cuales el perfil administrador permitirá ingresar y editar la configuración base del sistema (ingreso de variables nuevas, centros de atención, entre otras).

La versión móvil será usada por los pacientes, podrán acceder a la medicación recetada por el médico en la consulta, ingresar variables importantes para el medico en el control de la enfermedad (actualmente se registra el peso de la persona) y además el paciente accederá a información de interés sobre la IC ingresadas desde la aplicación web por los integrantes de la UMIC con el perfil adecuado.

Utilizando los datos ingresados en el sistema se generarán advertencias para los médicos de la UMIC, serán de ayuda en la toma de decisiones y en el seguimiento del paciente.

La solución se enmarca en la implementación de diferentes estándares propuestos por el programa de SALUD.UY que se viene desarrollando por parte del gobierno para mejorar el sistema de salud.

Algunos de estos estándares son: sistema de identificación de persona única llamado EMPI, historia clínica electrónica centralizada XDS y servidor terminológico SNOMED.

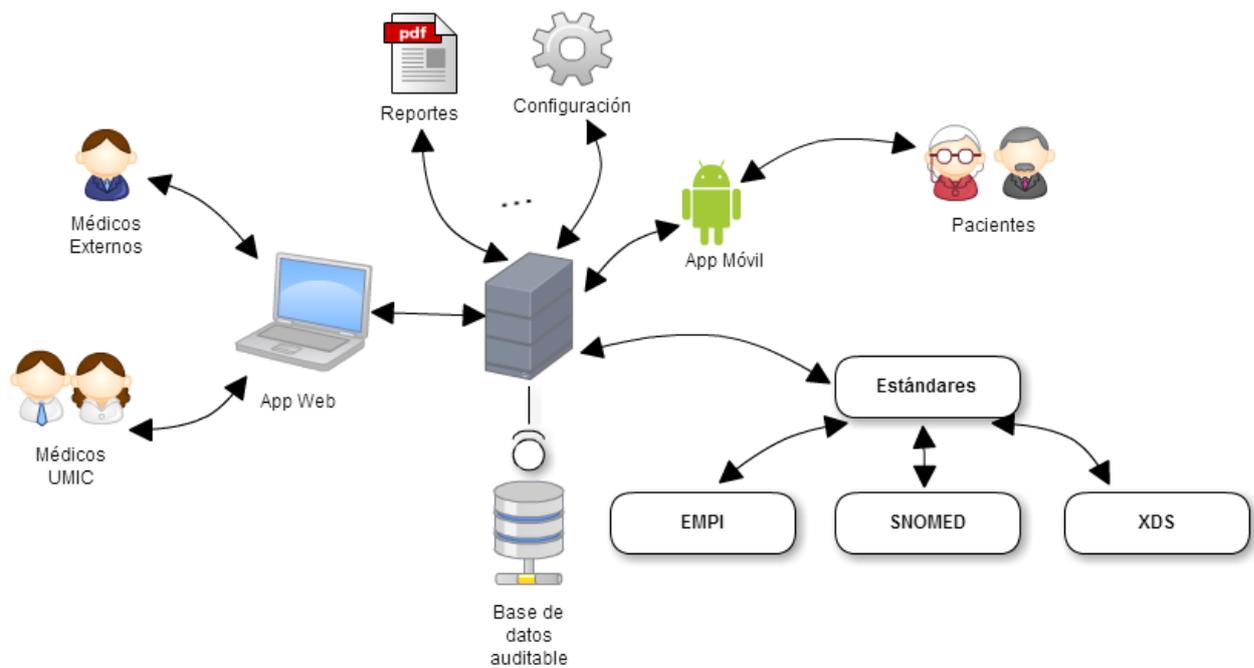


Figura 9 Situación proyectada de la solución al inicio del proyecto, se muestran el acceso de los diferentes usuarios. Los estándares que implementara y algunas de las funcionalidades.

## 8.6 Beneficios

A continuación se describen los principales beneficios de la solución.

**Calidad de la información:** mejora la fidelidad de los datos, menos overhead que el proceso actual. Registró de todas las modificaciones al utilizar una base de datos auditable, se puede verificar cuando y quien realizó los cambios.

**Digitalización y centralización de la información:** toda la información pasa a formar parte de una base de datos centralizada lo que mejora el registro en comparación con el formulario físico.

**Seguridad:** mediante diversos roles se restringe el acceso a la información a aquellas personas que están autorizadas para verla.

**Uso de datos:** al contar con los datos digitalizados es viable la generación de reportes y estadísticas que aportarán a la toma de decisiones en el seguimiento del paciente y en la gestión de la atención de UMIC.

**Histórico:** permite ver y analizar los datos del paciente de forma visual fácilmente a través del tiempo.

**Datos aportados por el paciente:** el paciente aporta datos de variables importantes para la detección de alertas en su enfermedad, lo que permite al médico poder tener un mejor control aún no realizadas las consultas en persona.

**Alertas:** avisos automáticos a los médicos sobre pacientes que no reciben consultas durante un tiempo determinados, controles en valores de variables, medicamentos contraindicados son ejemplos de alertas.

**Personalización del sistema:** el sistema permite la configuración por parte de los usuarios que cuente con el rol apropiado, el ingreso y des-habilitación de variables, modificación de interfaz de la visión global del paciente. Además, en ciertas funcionalidades el sistema cuenta con personalizaciones para cada médico en base a uso frecuentes de términos que son ingresados como datos.

## 8.7 Actores

### 8.7.1 Cliente

UMIC está integrada por un grupo de profesionales de la salud pertenecientes al Hospital de Clínicas “Dr. Manuel Quintela” de la República Oriental del Uruguay, quienes conforman un equipo multidisciplinario, de trabajo interdisciplinario, bajo la concepción de los programas de manejo de enfermedades crónicas, para atender a los pacientes portadores de insuficiencia cardiaca (IC) crónica por disfunción sistólica.

El método de trabajo es el de una policlínica especializada en la cual participan médicos internistas, cardiólogos, psiquiatra, ecocardiografistas, nutricionista, licenciada en enfermería y asistente social.

Los integrantes de UMIC que tendrán el rol de clientes son: Prof. Dra. Gabriela Ormaechea, Prof. Adgo. Dr. Pablo Álvarez y Dra. Gabriela Silvera.

### 8.7.2 Roles de negocio

**Médicos:** son los usuarios principales del sistema, registran consultas y toda la información necesaria para mejorar la atención del paciente.

**Paciente:** son pacientes de la UMIC, estos van a poder tener acceso mediante aplicación móvil información respecto a la medicación del tratamiento y a su enfermedad. Además, podrá aportar datos de variables.

### 8.7.3 Usuarios de la solución

**Médico UMIC:** son los médicos que trabajan en UMIC en el hospital de clínicas. Podrán ver todos los pacientes ingresados en el sistema.

**Médico Externo:** médicos que atienden a los pacientes que ingresaron a UMIC y Además, realizan controles en policlínicas externas a UMIC. Solo podrán ven los pacientes asignados tanto como médico responsable o tratante.

**Estudiantes pasantes:** los estudiantes pasantes realizan pasantías en UMIC colaborando en la atención de los pacientes. Van a tener permisos restringidos.

**Administrador:** son los usuarios que controlan la configuración del sistema, altas de nuevos usuarios, ingreso de nuevas variables, cambios de permisos de usuarios, y otras funcionalidades.

## 8.8 Especificación de requerimientos (ESRE)

A continuación se listan las historias de usuario que se realizaron para cumplir con los requerimientos establecidos.

Estas historias fueron administradas durante cada sprint mediante la herramienta online Trello.

### 8.8.1 Requerimientos funcionales

#### 8.8.1.1 FAMILIAS DE MEDICAMENTOS

FM1	<b>Agregar Familia de Medicamento</b>	Como administrador deseo poder agregar familias de medicamentos para luego poder asociarlas a un medicamento. Indicando el nombre de la familia, una descripción y un conjunto de alertas.
FM2	<b>Editar Familia de Medicamento</b>	Como administrador deseo poder editar las familias de medicamentos para corregir posibles errores al ingreso.
FM3	<b>Consultar Familia de Medicamento</b>	Como administrador deseo poder consultar las familias de medicamentos ingresadas al sistema.
FM4	<b>Listar Familias de Medicamento</b>	Como administrador deseo ver la lista de las familias de medicamentos ingresadas para poder gestionarlas.
FM5	<b>Deshabilitar Familia de Medicamento</b>	Como administrador quiero poder deshabilitar una familia de medicamentos para no usarla más en el sistema. Se deberán deshabilitar también todos los medicamentos que tengan esa familia asociada.
FM6	<b>Habilitar Familia de Medicamento</b>	Como administrador quiero poder habilitar una familia de medicamentos deshabilitada para volver a usarla en el sistema. No se deberán habilitar los medicamentos asociados a esta familia.

#### 8.8.1.2 MEDICAMENTOS

M1	<b>Agregar Medicamento</b>	Como administrador deseo poder agregar medicamentos para poder recetarlos a un paciente. Indicando un nombre, una descripción, a cuál de las familias de medicamentos previamente ingresadas pertenece y un conjunto de alertas de la familia.
M2	<b>Editar Medicamento</b>	Como administrador deseo poder editar los medicamentos para corregir posibles errores al ingreso.

M3	<b>Consultar Medicamento</b>	Como administrador deseo poder consultar los medicamentos ingresados al sistema.
M4	<b>Listar Medicamentos</b>	Como administrador deseo ver la lista de los medicamentos ingresados para poder gestionarlos.
M5	<b>Deshabilitar Medicamento</b>	Como administrador quiero poder deshabilitar un medicamento para no usarlo más en el sistema.
M6	<b>Habilitar Medicamento</b>	Como administrador quiero poder habilitar un medicamento deshabilitado para volver a usarlo en el sistema.

### 8.8.1.3 SIGNOS VITALES

SV1	<b>Agregar Signo Vital</b>	Como administrador quiero poder agregar signos vitales para luego usarlos en la consulta del paciente.
SV2	<b>Editar Signo Vital</b>	Como administrador quiero poder modificar los signos vitales ingresados.
SV3	<b>Habilitar Signo Vital</b>	Como administrador quiero poder habilitar un signo vital para usarlo en el sistema.
SV4	<b>Deshabilitar Signo Vital</b>	Como administrador quiero poder deshabilitar un signo vital para no usarlo en el sistema.
SV5	<b>Consultar Signo Vital</b>	Como administrador quiero poder consultar la información de un signo vital ingresado.
SV6	<b>Listar Signos Vitales</b>	Como administrador quiero poder consultar la lista de signos vitales ingresados al sistema para poder gestionarlos.

### 8.8.1.4 EXÁMENES

EX1	<b>Agregar Examen</b>	Como administrador quiero poder agregar exámenes para luego usarlos en la consulta del paciente.
EX2	<b>Editar Examen</b>	Como administrador quiero poder modificar los exámenes ingresados.
EX3	<b>Habilitar Examen</b>	Como administrador quiero poder habilitar un examen para usarlo en el sistema.
EX4	<b>Deshabilitar Examen</b>	Como administrador quiero poder deshabilitar un examen para no usarlo en el sistema.
EX5	<b>Consultar Examen</b>	Como administrador quiero poder consultar la información de un examen ingresado.
EX6	<b>Listar Exámenes</b>	Como administrador quiero poder consultar la lista de exámenes ingresados al sistema para poder gestionarlos.

### 8.8.1.5 VALORES CALCULADOS

VC1	<b>Agregar Calculado</b>	<b>Valor</b> Como administrador quiero poder agregar valores calculados para luego usarlos en la consulta del paciente.
VC2	<b>Editar Valor Calculado</b>	Como administrador quiero poder modificar los valores calculados ingresados.

VC3	<b>Habilitar Calculado</b>	<b>Valor</b>	Como administrador quiero poder habilitar un valor calculado para usarlo en el sistema.
VC4	<b>Deshabilitar Calculado</b>	<b>Valor</b>	Como administrador quiero poder deshabilitar un valor calculado para no usarlo en el sistema.
VC5	<b>Consultar Calculado</b>	<b>Valor</b>	Como administrador quiero poder consultar la información de un valor calculado ingresado.
VC6	<b>Listar Valor Calculados</b>		Como administrador quiero poder consultar la lista de valores calculados ingresados al sistema para poder gestionarlos.

#### 8.8.1.6 EVENTOS

E1	<b>Agregar paciente</b>	<b>Evento</b>	<b>a</b>	Como médico deseo poder agregar un evento a un paciente indicando el tipo de evento, el subtipo, una fecha y un comentario.
E2	<b>Consultar paciente</b>	<b>Evento</b>	<b>de</b>	Como médico quiero poder consultar la información de un evento asociado a un paciente.
E3	<b>Editar paciente</b>	<b>Evento</b>	<b>de</b>	Como médico deseo poder evitar el subtipo, fecha y descripción de un evento ingresado a un paciente.
E4	<b>Eliminar paciente</b>	<b>Evento</b>	<b>de</b>	Como médico deseo poder eliminar un evento asociado a un paciente.
E5	<b>Listar paciente</b>	<b>Eventos</b>	<b>de</b>	Como médico deseo poder consultar la lista de eventos de un paciente.

#### 8.8.1.7 COMORBILIDADES

CO1	<b>Agregar Comorbilidad</b>			Como administrador quiero poder agregar comorbilidades para luego usarlos en la consulta del paciente. Para esto quiero poder ingresar un nombre y un valor para luego poder calcular el Índice de Charlson.
CO2	<b>Editar Comorbilidad</b>			Como administrador quiero poder modificar las comorbilidades ingresados.
CO3	<b>Habilitar Comorbilidad</b>			Como administrador quiero poder habilitar una comorbilidad para usarlo en el sistema.
CO4	<b>Deshabilitar Comorbilidad</b>			Como administrador quiero poder deshabilitar una comorbilidad para no usarlo en el sistema.
CO5	<b>Consulta Comorbilidad</b>		<b>de</b>	Como administrador quiero poder consultar la información de una comorbilidad ingresado.
CO6	<b>Listar Comorbilidades</b>			Como administrador quiero poder consultar la lista de comorbilidades ingresados al sistema para poder gestionarlos.

#### 8.8.1.8 OTROS ANTECEDENTES

AN1	<b>Listar antecedentes paciente</b>	<b>Otros de</b>		Como médico quiero ver la lista de otros antecedentes del paciente (que no sean comorbilidades y FDR).
AN2	<b>Agregar Antecedente a paciente</b>	<b>Otro</b>		Como médico quiero poder agregar un antecedente nuevo a un paciente seleccionándolo de la lista de otros antecedentes del

---

sistema.

---

AN3	<b>Eliminar Antecedente a paciente</b>	<b>Otro</b>	Como médico quiero poder eliminar un antecedente asignado a un paciente.
-----	--	-------------	--

### 8.8.1.9 FACTORES DE RIESGO

FR1	<b>Agregar Riesgo</b>	<b>Factor</b>	<b>de</b>	Como administrador quiero poder agregar factor de riesgo ingresando su nombre y una descripción para luego usarlos en la consulta del paciente.
-----	-----------------------	---------------	-----------	---

FR2	<b>Editar Factor de Riesgo</b>			Como administrador quiero poder modificar los factores de riesgo ingresados.
-----	--------------------------------	--	--	--

FR3	<b>Habilitar Riesgo</b>	<b>Factor</b>	<b>de</b>	Como administrador quiero poder habilitar un factor de riesgo para usarlo en el sistema.
-----	-------------------------	---------------	-----------	--

FR4	<b>Deshabilitar Riesgo</b>	<b>Factor</b>	<b>de</b>	Como administrador quiero poder deshabilitar un factor de riesgo para no usarlo en el sistema.
-----	----------------------------	---------------	-----------	--

FR5	<b>Consultar Riesgo</b>	<b>Factor</b>	<b>de</b>	Como administrador quiero poder consultar la información de un factor de riesgo ingresado.
-----	-------------------------	---------------	-----------	--

FR6	<b>Listar Riesgos</b>	<b>Factor</b>	<b>de</b>	Como administrador quiero poder consultar la lista de factor de riesgos ingresados al sistema para poder gestionarlos.
-----	-----------------------	---------------	-----------	--

### 8.8.1.10 CENTROS DE SALUD

CS1	<b>Agregar Salud</b>	<b>Centro</b>	<b>de</b>	Como administrador quiero poder ingresar centros de salud al sistema para poder asociarlos a un paciente o médico. Para esto quiero poder ingresar un nombre, información del centro y ubicación en Google Maps.
-----	----------------------	---------------	-----------	--

CS2	<b>Editar Centro de Salud</b>			Como administrador quiero poder editar los datos de un centro de salud ingresado al sistema.
-----	-------------------------------	--	--	--

CS3	<b>Habilitar Salud</b>	<b>Centro</b>	<b>de</b>	Como administrador quiero poder habilitar un centro de salud para usarlo en el sistema.
-----	------------------------	---------------	-----------	---

CS4	<b>Deshabilitar Salud</b>	<b>Centro</b>	<b>de</b>	Como administrador quiero poder deshabilitar un centro de salud para no usarlo en el sistema.
-----	---------------------------	---------------	-----------	---

CS5	<b>Consultar Salud</b>	<b>Centro</b>	<b>de</b>	Como administrador quiero poder consultar los datos de un centro de salud ingresado al sistema.
-----	------------------------	---------------	-----------	---

CS6	<b>Listar Centros de Salud</b>			Como administrador quiero ver la lista de centros de salud ingresados al sistema para poder gestionarlos.
-----	--------------------------------	--	--	---

### 8.8.1.1

### 8.8.1.12 PACIENTES

P1	<b>Alta Paciente</b>			Como médico quiero poder dar de alta pacientes en el sistema, para eso quiero indicar sus datos patronímicos y sus datos clínicos.
----	----------------------	--	--	--

P2	<b>Modificar Paciente</b>	<b>datos</b>		Como médico quiero poder modificar los datos ingresados al dar de alta un paciente.
----	---------------------------	--------------	--	---

P3	<b>Consulta Paciente</b>			Como médico quiero poder consultar los datos de un paciente ingresado al sistema.
----	--------------------------	--	--	---

P4	<b>Asignar</b>	<b>Médico</b>		Como médico quiero poder asignar un médico referente a un
----	----------------	---------------	--	---

---

	<b>Referente a Paciente</b>		paciente. Para esto quiero indicar el médico desde la lista de médicos ingresados al sistema.
P5	<b>Desasignar Referente a Paciente</b>	<b>Médico</b>	Como médico quiero poder desasignar un médico referente a un paciente.
P6	<b>Asignar Tratante a Paciente</b>	<b>Médico</b>	Como médico quiero poder asignar un médico tratante a un paciente. Para esto quiero indicar el médico desde la lista de médicos ingresados al sistema.
P7	<b>Desasignar Tratante a Paciente</b>	<b>Médico</b>	Como médico quiero poder desasignar un médico tratante a un paciente.
P8	<b>Agregar Paciente</b>	<b>Contacto a</b>	Como médico quiero poder agregar un nuevo contacto al paciente indicando su nombre, apellido, parentesco, mail y número de teléfono.
P9	<b>Editar Paciente</b>	<b>Contacto a</b>	Como médico quiero poder editar un contacto agregado a un paciente para mantenerlo actualizado.
P10	<b>Listar Paciente</b>	<b>Contactos de</b>	Como médico quiero poder ver la lista de contactos de un paciente.
P11	<b>Editar Paciente</b>	<b>Contacto a</b>	Como médico quiero poder editar la información de un contacto ingresado a un paciente.
P12	<b>Agregar Paciente</b>	<b>Ítem checklist</b>	Como médico quiero poder agregar un nuevo ítem al checklist del paciente.
P13	<b>Marcar completado checklist Paciente</b>	<b>como ítem</b>	Como médico quiero poder marcar como completado un ítem del checklist del paciente.
P14	<b>Indicar de Paciente</b>	<b>Fallecimiento</b>	Como médico quiero ingresar el fallecimiento de un paciente indicando la fecha, la causa y el modo de certificación.
P15	<b>Ver Inicio (resumen) de Paciente</b>		Como médico quiero ver un resumen del paciente en una sola página para poder agilizar el seguimiento. El resumen deberá incluir datos sobre los últimos ingresos de eventos, medicación, comentarios de consultas y contraindicaciones medicamentosas. Además de una lista con sus factores de riesgos y comorbilidades. También se debe mostrar su nombre, apellido, score, etiología, edad, si vive solo o no, número de historia clínica, su foto y su lista de tareas relacionadas (checklist).
P16	<b>Listar Tratantes</b>	<b>Médicos</b>	Como médico quiero ver la lista de médicos tratantes de un paciente.
P17	<b>Listar Referentes</b>	<b>Médicos</b>	Como médico quiero ver la lista de médicos referentes de un paciente.
P18	<b>Agregar Situación Vital</b>		Como médico quiero poder agregar la situación vital de un paciente, para esto quiero ingresar su estado (censurado, vivo o fallecido), una fecha y una observación.
P19	<b>Editar Situación Vital</b>		Como médico quiero poder editar la situación vital de un paciente.

### 8.8.1.13 CONSULTA

CON1	<b>Agregar Consulta Paciente</b>	<b>Signo Vital</b>	Como médico deseo poder ingresar un valor de un signo vital para un paciente en una consulta.
CON2	<b>Editar Consulta Paciente</b>	<b>Signo Vital</b>	Como médico deseo poder editar un valor ingresado de un signo vital para un paciente en una consulta.
CON3	<b>Listar Consulta Paciente</b>	<b>Signos Vitales</b>	Como médico deseo poder ver la lista de valores de signos vitales ingresados en una consulta de un paciente.

CON4	<b>Listar Calculados Paciente</b>	<b>Valores Consulta</b>	Como médico deseo poder ver la lista de valores calculados de un paciente en una consulta.
CON5	<b>Agregar Examen Paciente</b>	<b>Resultado Consulta</b>	Como médico deseo poder ingresar el resultado de un examen de un paciente en una consulta. Para esto quiero seleccionar el examen de una lista de exámenes ingresados al sistema, indicar los valores de cada ítem, la fecha del examen y un comentario.
CON6	<b>Editar Examen Paciente</b>	<b>Resultado Consulta</b>	Como médico deseo poder editar los valores ingresados de un examen de un paciente en una consulta.
CON7	<b>Listar Examen Paciente</b>	<b>Resultados Consulta</b>	Como médico deseo ver la lista de exámenes que fueron ingresados en la consulta de un paciente.
CON8	<b>Consultar Examen Paciente</b>	<b>Resultado Consulta</b>	Como médico deseo consultar los valores de un examen ingresado en la consulta de un paciente.
CON9	<b>Agregar Consulta Paciente</b>	<b>Medicación</b>	Como médico deseo poder recetar un medicamento a un paciente. Para esto quiero seleccionar la familia a la que pertenece el medicamento para luego ver la lista de medicamentos asociados a esa familia y seleccionar el medicamento que quiero recetar. Luego quiero ingresar un conjunto de dosis (hora y cantidad) y el propósito pro el cual fue recetado.
CON10	<b>Editar Consulta Paciente</b>	<b>Medicación</b>	Como médico deseo poder editar la medicación recetada a un paciente en una consulta.
CON11	<b>Eliminar Consulta Paciente</b>	<b>Medicación</b>	Como médico deseo poder eliminar la medicación recetada a un paciente en una consulta.
CON12	<b>Consultar Consulta Paciente</b>	<b>Medicación</b>	Como médico deseo poder consultar las medicaciones recetadas a un paciente en una consulta.
CON13	<b>Listar Consulta Paciente</b>	<b>Medicación</b>	Como médico deseo ver la lista de prescripciones de un paciente en una consulta.
CON14	<b>Repetir Medicación</b>		Como médico deseo poder repetir una prescripción realizada en la consulta anterior a la consulta actual. Para esto quiero ver la lista de prescripciones de la consulta anterior y seleccionar la prescripción a repetir. Quiero que se me permita modificar las dosis y el propósito de la nueva prescripción.
CON15	<b>Contraindicar Medicamento Paciente</b>	<b>Familia</b>	Como médico deseo poder contraindicar una familia de medicamento a un paciente para luego alertarlo cuando se le quiera recetar algún medicamento de dicha familia. Para esto quiero seleccionar una familia de la lista de familias del sistema e indicar un motivo.
CON16	<b>Contraindicar Medicamento Paciente</b>	<b>a</b>	Como médico deseo poder contraindicar un medicamento a un paciente para luego alertarlo cuando se le quiera recetar ese medicamento. Para esto quiero seleccionar el medicamento de la lista de medicamentos del sistema e indicar un motivo.
CON17	<b>Listar Contraindicaciones Paciente</b>		Como médico deseo ver la lista de contraindicaciones del paciente.
CON18	<b>Eliminar Contraindicación Paciente</b>	<b>a</b>	Como médico deseo poder eliminar una contraindicación a un paciente.
CON19	<b>Mostrar Medicamento</b>	<b>Alertas</b>	Como médico deseo ver las alertas de un medicamento al seleccionarlo para recetárselo a un paciente.

CON20	<b>Mostrar Alertas Familia Medicamento</b>	Como médico deseo ver las alertas de una familia de medicamento al recetar un medicamento de dicha familia.
CON21	<b>Mostrar Alerta al Contraindicación al medicar</b>	Como médico deseo ver si un medicamento / familia de medicamento está contraindicado/a y el motivo al seleccionarlo/a para recetárselo/a a un paciente.
CON22	<b>Agregar Comentario Consulta Paciente</b>	Como médico quiero poder agregar un comentario en una consulta de un paciente.
CON23	<b>Listar Comentarios Consulta Paciente</b>	Como médico quiero ver los comentarios ingresados a un paciente en una consulta
CON24	<b>Exportar Consulta en PDF</b>	Como médico quiero poder descargar todos los datos ingresados en una consulta en formato PDF.
CON25	<b>Consultar Históricos de cambios en Consulta</b>	Como médico quiero poder ver todos los cambios que se realizaron en una consulta, que usuario lo realizó y cuando.
CON26	<b>Listar Consultas</b>	Como médico deseo ver una lista con todas las consultas del paciente permitiéndome navegar entre ellas (anterior y siguiente).

#### 8.8.1.14 CUERPO MÉDICO

CM1	<b>Agregar Médico</b>	<b>Cuerpo</b>	Como administrador quiero poder agregar un nuevo integrante del cuerpo médico, para esto quiero indicar su nombre, mail, perfil de usuario y si es administrador o no. Al usuario le tiene que llegar un mail con una contraseña para que pueda ingresar al sistema y completar su registro.
CM2	<b>Habilitar Médico</b>	<b>Cuerpo</b>	Como administrador quiero poder habilitar un médico para que pueda usar el sistema.
CM3	<b>Deshabilitar Médico</b>	<b>Cuerpo</b>	Como administrador quiero poder deshabilitar un médico para que no pueda usar el sistema.
CM4	<b>Marcar Administrador Médico</b>	<b>como Cuerpo</b>	Como administrador quiero poder marcar un integrante del cuerpo médico como administrador.
CM5	<b>Desmarcar Administrador Médico</b>	<b>como Cuerpo</b>	Como administrador quiero poder desmarcar un integrante del cuerpo médico como administrador.

#### 8.8.1.15 NUTRICIÓN

N1	<b>Agregar Datos Clínicos Nutricionales</b>	Como médico quiero agregar datos nutricionales del paciente, para esto quiero ingresar su peso ideal, peso habitual, su tránsito intestinal, su estado dentario y su apetito.
N2	<b>Editar Datos Clínicos Nutricionales</b>	Como médico quiero poder editar los datos clínicos nutricionales ingresados para un paciente.
N3	<b>Agregar Anteriores Regímenes</b>	Como médico quiero poder agregar un nuevo régimen anterior, para esto quiero ingresar la fecha de inicio y fin del régimen, si fue asignado por un médico o un nutricionista y un comenario.
N4	<b>Editar Anteriores Regímenes</b>	Como médico quiero poder editar los regímenes anteriores.
N5	<b>Agregar Alimentaria Anamnesis</b>	Como médico quiero poder agregar la Anamnesis Alimentaria de un paciente. Para esto quiero indicar el lugar, la hora y los alimentos que ingiere en cada una de las comidas (desayuno, almuerzo, merienda, cena, colación y fin de semana). Además, quiero ingresar preferencias, intolerancias, prescripción

			nutricional, cálculo de requerimientos y observaciones.
N6	<b>Editar Anamnesis Alimentaria</b>		Como médico quiero poder editar los datos agregados en la anamnesis alimentaria.
N7	<b>Agregar Diagnostico</b>		Como médico quiero poder agregar un diagnóstico.
N8	<b>Editar Diagnostico</b>		Como médico quiero poder editar el diagnostico de un paciente.
N9	<b>Agregar Comentario</b>		Como médico quiero poder agregar comentarios sobre la nutrición del paciente.
N10	<b>Editar Comentario</b>		Como médico quiero poder editar los comentarios sobre la nutrición del paciente.
N11	<b>Consultar Datos Clínicos Nutricional</b>		Como médico quiero poder consultar la información agregada en datos clínicos nutricional de un paciente.
N12	<b>Consultar Regímenes Anteriores</b>		Como médico quiero poder consultar la información sobre Regímenes Anteriores de un paciente.
N13	<b>Consultar Anamnesis Alimentaria</b>		Como médico quiero poder consultar la información sobre Anamnesis Alimentaria de un paciente.
N14	<b>Consultar Diagnóstico</b>		Como médico quiero poder consultar el diagnóstico de un paciente.
N15	<b>Consulta Comentarios</b>		Como médicos quiero poder consultar los comentarios sobre la nutrición del paciente.

#### 8.8.1.16 EVOLUCIÓN

EV1	<b>Evolución Consulta</b>		Como médico quiero ver un resumen con los valores de las variables (exámenes, signos vitales y valores calculados) ingresadas en las últimas consultas para poder tomar decisiones sobre el paciente.
EV2	<b>Evolución Variable</b>		Como médico deseo poder ver la evolución de una variable para un paciente, para esto quiero seleccionar la variable de una lista. Además, quiero ver información sobre la variable y sus ítems. Los datos se deben mostrar en forma de tabla y graficados.

#### 8.8.1.17 INDICADORES

I1	<b>Indicador 1</b>		Como médico quiero ver el siguiente indicador: Asistencias desde [fecha inicio] hasta [fecha fin]
I2	<b>Indicador 2</b>		Como médico quiero ver el siguiente indicador: Tiempo dedicado a pacientes desde [fecha inicio] hasta [fecha fin]
I3	<b>Indicador 3</b>		Como médico quiero ver el siguiente indicador: Pacientes nuevos desde [fecha inicio] hasta hoy.
I4	<b>Indicador 4</b>		Como médico quiero ver el siguiente indicador: Promedio de asistencias por pacientes activos (vivos e incluidos) desde [fecha inicio] hasta hoy.
I5	<b>Indicador 5</b>		Como médico quiero ver el siguiente indicador: Cantidad de pacientes activos (vivos e incluidos) sin asistencia, desde [fecha inicio] hasta hoy.
I6	<b>Indicador 6</b>		Como médico quiero ver el siguiente indicador: Cantidad de pacientes fallecidos desde [fecha inicio] hasta hoy.
I7	<b>Indicador 7</b>		Como médico quiero ver el siguiente indicador: Cantidad de pacientes con el evento internación desde [fecha inicio] hasta hoy.

18	<b>Indicador 8</b>	Como médico quiero ver el siguiente indicador: Cantidad de pacientes con el evento internación desde [fecha inicio] hasta hoy.
19	<b>Indicador 9</b>	Como médico quiero ver el siguiente indicador: Cantidad de pacientes activos (vivos e incluidos) sin registrar ninguna variable desde [fecha inicio] hasta hoy.

#### 8.8.1.18 CALENDARIO

CA1	<b>Agregar Calendario</b>	<b>Cita</b> Como médico quiero poder agregar una cita (consulta) con un paciente al calendario, para esto quiero ingresar el día, la fecha de inicio y fin de la cita y el paciente.
CA2	<b>Modificar Calendario</b>	<b>Cita</b> Como médico quiero poder modificar los datos de una cita con un paciente.
CA3	<b>Eliminar Cita Paciente</b>	Como médico quiero poder eliminar una cita con un paciente.
CA4	<b>Ver Mi Calendario</b>	Como médico deseo poder ver mi calendario con las citas de mis pacientes agendadas.
CA5	<b>Ver calendario de otro médico</b>	Como administrador quiero poder ver el calendario de otro médico del sistema.

#### 8.8.1.19 GUIAS DE PACIENTES

GP1	<b>Listar Guías Pacientes</b>	Como administrador quiero poder ver la lista de guías de paciente para poder gestionarlas.
GP2	<b>Agregar Guía Paciente</b>	Como administrador quiero poder agregar una nueva guía de paciente.
GP3	<b>Editar Guía Paciente</b>	Como administrador quiero poder editar una guía de paciente.
GP4	<b>Eliminar Guía Paciente</b>	Como administrador quiero poder eliminar una guía de paciente.
GP5	<b>Habilitar Guía Paciente</b>	Como administrador quiero poder habilitar una guía de paciente para que se muestre en la app móvil.

#### 8.8.1.20 CONFIGURACIÓN

CF1	<b>Listar Configuración</b>	<b>Ítems</b> Como administrador quiero ver una lista de valores configurables (ítem de configuración) del sistema junto a su nombre y descripción para poder gestionarlos.
CF2	<b>Consultar Configuración</b>	<b>Ítem</b> Como administrador quiero poder consultar la información sobre un ítem de configuración.
CF3	<b>Editar Configuración</b>	<b>Ítem</b> Como administrador del sistema quiero poder modificar el valor de un ítem de configuración y su descripción.

#### 8.8.1.21 APLICACIÓN MÓVIL

AP1	<b>Iniciar sesión en móvil</b>	Como paciente quiero iniciar sesión en la app móvil ingresando un pin por única vez.
AP2	<b>Consultar guías de usuario</b>	Como paciente quiero consultar las guías de usuario ingresadas en el sistema.
AP3	<b>Consultar Medicación</b>	Como paciente quiero ver la medicación que me recetó el médico en la última consulta.

AP4	<b>Marcar Medicación como tomada</b>	Como paciente quiero marcar una prescripción (de la última consulta) como completada para indicar que ya la tomé.
AP5	<b>Desmarcar medicación como no tomada</b>	Como paciente quiero desmarcar una prescripción (de la última consulta) como completada para indicar que aún no la he tomado.
AP6	<b>Consultar información de contacto sobre UMIC</b>	Como paciente quiero consultar información de contacto de la UMIC desde la app.
AP7	<b>Ver alerta al ingresar peso</b>	Como paciente quiero ver una alerta cuando mi peso hay variado más de lo estipulado.
AP8	<b>Ver alerta si no se ha ingresado el peso diario</b>	Como paciente quiero saber si ya he ingresado mi peso del día.
AP9	<b>Ingresar Peso Diario</b>	Como paciente quiero ingresar mi peso diario para conocer su evolución.

#### 8.8.1.22 SEGUIMIENTO DE PACIENTES

SM1	<b>Generar PIN</b>	Como médico quiero poder generar un pin para un paciente para que éste pueda ingrese a la app móvil. Para esto debo ingresar un mail (del paciente) al cual le llegará el pin generado.
SM2	<b>Consultar cantidad de guías vistas por paciente</b>	Como médico quiero ver la cantidad de guías de paciente que un paciente ha visto desde la app.
SM3	<b>Consultar ingresos paciente</b>	<b>Pesos por</b> Como médico quiero poder consultar los pesos ingresados por un paciente desde la app móvil.
SM4	<b>Consultar último inicio de sesión de paciente</b>	Como médico quiero ver cuando fue el último inicio de sesión de un paciente en la app móvil.
SM5	<b>Modificar variación de alerta Paciente</b>	<b>valor peso</b> Como médico quiero poder modificar el valor de variación de peso para un paciente por el cual el paciente va a ser alertado.
SM6	<b>Consultar variación de alerta Paciente</b>	<b>valor peso</b> Como médico quiero poder consultar el valor de variación de peso para un paciente por el cual el paciente va a ser alertado.
SM7	<b>Volver a generar Pin Paciente</b>	Como médico quiero poder volver a generar un pin para un paciente.

#### 8.8.1.23 OTROS

O1	<b>Conexión con SPSS</b>	Los datos deben poder ser tratados fácilmente a través del SPSS.
O2	<b>Migración de datos</b>	Los datos de la base del Epi Info que actualmente utiliza la UMIC deben ser migrados a SIMIC en la medida que sea compatibles.
O3	<b>Paginación</b>	Los datos de las tablas se deben presentar paginados. Con la posibilidad de ir a la página siguiente, anterior o a una específica.
O4	<b>Filtros</b>	Los datos de las tablas se deben poder filtrar mediante filtros personalizados para cada tabla.
O5	<b>Búsqueda</b>	Los datos de las tablas se deben poder filtrar mediante una búsqueda en al menos un atributo.
O6	<b>EMPI</b>	Se debe de implementar la integración con el estándar EMPI, del programa de SALUD.UY.

O7	<b>SNOMED</b>	Se debe de implementar la integración con el estándar SNOMED, del programa de SALUD.UY
O8	<b>XDS</b>	Se debe de implementar la integración con el estándar HCE, del programa de SALUD.UY

## 8.8.2 Requerimientos no funcionales

A continuación se describen los principales requerimientos no funcionales y los atributos de calidad más relevantes tomados en cuenta en la arquitectura.

### 8.8.2.1 Seguridad

Uno de los aspectos que más preocupó fue la seguridad, confidencialidad y mantenimiento de los datos. El sistema prevé el acceso no autorizado a la información y cuenta con mecanismos que permiten el registro de las acciones realizadas dentro del sistema, logrando saber quién las hizo.

### 8.8.2.2 Confidencialidad

El sistema soporta diferentes perfiles de usuarios y éstos tienen ciertas restricciones respecto a la información a la que podrán tener acceso. La arquitectura cuenta con mecanismos que aseguran que quien acceda a los datos tenga los privilegios necesarios para su visualización, edición o creación según corresponda.

Un aspecto importante dentro este requerimiento fue que se debió manejar en la base de datos la información personal del paciente (datos patronímicos) separada de los datos clínicos respetando así la ley de protección de datos personales.

### 8.8.2.3 Integridad

Todos los datos utilizados en SIMIC deben ser fidedignos. La información manejada por el sistema debe ser cuidadosamente protegida de estados inconsistentes. Se debe tener presente en todo momento que cualquier dato equívoco que despliegue el sistema, puede significar la toma de una decisión incorrecta que puede repercutir sobre la atención del paciente. El sistema realiza las validaciones correspondientes sobre ciertos datos ingresados por el usuario de los que su integridad sea importante. Estas validaciones fueron definidas conjuntamente con los integrantes de la UMIC que en ciertos casos manifestaron la necesidad de no incluir validaciones, por resultar quizás tedioso en momentos críticos el ingreso de forma rápida. Una de los conocimientos adquirido por el equipo es la frase “al médico no se le puede obligar a ingresar nada”.

### 8.8.2.4 Usabilidad

Este es un punto que muchas veces se menosprecia por parte del equipo de desarrollo, para SIMIC se presenta como crítico y puede determinar el éxito o fracaso del sistema. Por ser un área donde no son muy aceptados los sistemas informáticos, como se pudo observar en la investigación que se realizó al comienzo del proyecto, uno de los factores que entiende el equipo necesarios para captar la atención del usuario final es su usabilidad. El sistema debe presentar una interfaz sencilla e intuitiva, que ayude y no entorpezca el trabajo por parte del médico.

SIMIC tampoco reinventará la rueda en este tema, ya que no se dispone de los recursos necesarios para poder realizar una interfaz, ni es la finalidad del proyecto.

El equipo decidió seguir los estándares de usabilidad y diseños propuestos por la empresa Google [25] en sus guías de “Material Design” especificadas en [26]. Éste estandar se aplica para la aplicación web y para la aplicación móvil.

#### 8.8.2.5 Integración

Actualmente al desarrollar un sistema sin proveer la fácil integración con otros, el resultado sería un sistema obsoleto. Por lo que para SIMIC es muy importante este punto.

El gobierno definió en la primera etapa de implementación del Programa Salud.uy, algunos componentes en los que se trabaja con un enfoque único a nivel nacional.

Para SIMIC incorporar estos servicios que aún están en desarrollo y existe poca documentación, es un gran desafío. Se realizó la implementación de las siguientes integraciones:

- Repositorio de XDS para el envío de CDA (Historia Clínica Electrónica Nacional)
- Integración con el Sistema Nacional de Terminológica (Servidor terminológico que incluye SNOMED, y otros estándares).
- El sistema de identificación única de personas (EMPI)

#### 8.8.2.6 Legislación actual

Por tratarse de un sistema de salud es muy importante cumplir con la legislación vigente. Durante el proyecto se tuvo que tener siempre presente la legislación existente en el área y poder brindar con la solución su cumplimiento.

Vale referenciar las leyes N° 18.335 [27] y N° 18.331 [28] y también cabe destacar “la cartilla de derechos y deberes de los usuarios de los servicios de salud” [7].

# 9 Especificación del diseño

## 9.1 Introducción

El propósito de este capítulo es brindar la información técnica necesaria para entender el funcionamiento del sistema.

Se detallan los principales problemas arquitectónicos a los que se enfrentó el equipo y la forma de resolver los mismos, la justificación de las tecnologías utilizadas así como las principales vistas y diagramas de la solución.

## 9.2 Arquitectura del sistema

La Arquitectura a utilizar será Cliente-Servidor. Se desarrollará una sola aplicación integrada, en la que solo se permitirá el acceso a los usuarios registrados en el sistema y a las tareas a las cuales estén autorizados. Se empleará un servidor centralizado, se podrá extender fácilmente a un clúster o granja, por ejemplo la exclusión mutua no será a nivel de código sino a nivel de base de datos.

Se tendrá una arquitectura basada en capas, que publicará servicios REST que se serán la forma de comunicación para la aplicación web Single Page Application (SPA) [29], como para la aplicación móvil de los pacientes. La interoperabilidad con otros sistemas se realizará a través de servicios SOAP.

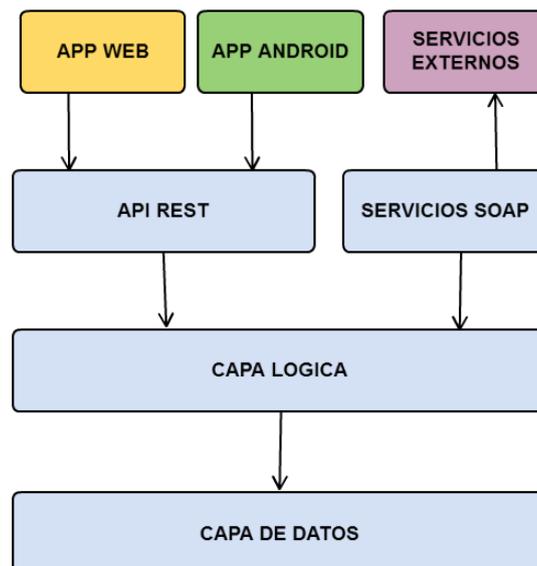


Figura 10 Diagrama básico que muestra los componentes de la aplicación, notar que la api REST se consume desde la aplicación web y la aplicación móvil

## 9.2.1 Arquitectura de la aplicación web

La aplicación web será basada en AngularJS. A diferencia de las arquitecturas clásicas en donde los terminales son tontos, y en cada petición se envía la página con todos los datos. El FrontEnd será realmente una aplicación por sí solo, al momento de entrar a la página se cargarán los archivos imprescindibles de la aplicación, luego se realizarán comunicaciones con el servidor para traer datos cuando sea necesario.

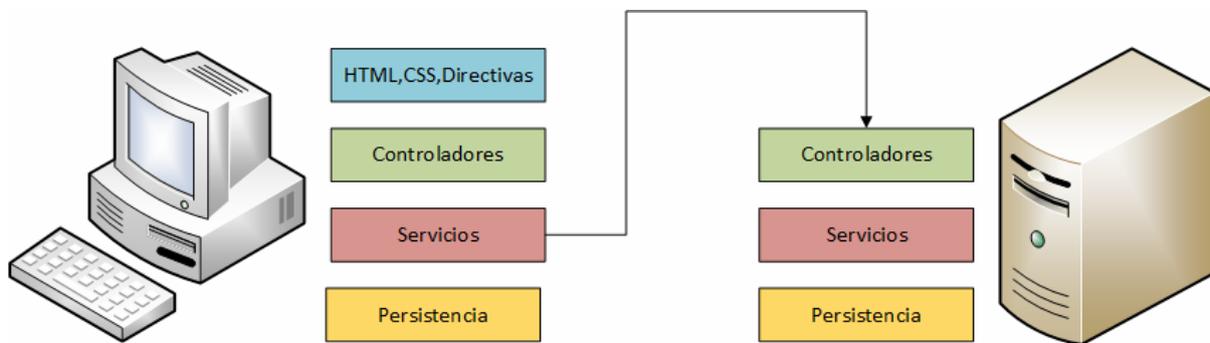


Figura 11 Comunicación entre el FrontEnd y BackEnd, notar que en el FrontEnd también existen los controladores, servicios y persistencia.

Del lado del servidor encontramos:

- Controladores: Son los encargados de brindar los servicios REST que serán consumidos tanto por la aplicación web, o móvil.
- Servicios: implementa la lógica propiamente dicha del sistema y la interacción con la capa de acceso a datos.
- Persistencia: Esta capa es la encargada de implementar el acceso a datos.

Del cliente podemos observar que existen controladores, servicios y persistencia. Con esto se puede apreciar el potencial que nos ofrece AngularJS, conjuntamente usado con el patrón MVC brinda la posibilidad de realizar una interfaz más inteligente.

La arquitectura del servidor será descrita en forma detallada a través de distintas vistas en los siguientes capítulos.

## 9.2.2 Arquitectura de la aplicación móvil

Se realizó una aplicación móvil (nativa para Android) para uso de los pacientes.

La comunicación es a través del protocolo HTTP al igual que la aplicación web.

Para realizar una aplicación más eficiente y robusta se decidió el uso de los SyncAdapter [30] que provee Android, los cuales encapsulan las tareas de transferencia de datos entre el dispositivo y el servidor. Podemos decir que SyncAdapter es un plug-in que se encarga de las sincronizaciones de datos de la aplicación con el servidor. Este plug-in será registrado en el administrador de sincronización de la plataforma, que es el encargado de ejecutarlo. Se activa cuando se solicita y además el sistema automáticamente lo activará una vez al día.

Algunas de las ventajas que podemos encontrar en la documentación de Android para los desarrolladores son:

- Eficiencia de la batería, porque al ser manejado por el administrador de sincronización este tiene en cuenta diferentes factores para decidir el momento en el que se realiza la sincronización.
- Interface, el usuario en una misma pantalla puede deshabilitar la sincronización.
- Actualización de contenido.
- Mecanismo de reintentos, provee un mecanismo de reintento que se activa en intervalos de tiempo que crecen de forma exponencial.

A continuación se listan los principales elementos de arquitectura de la aplicación móvil:

ContentProvider:

- Es el corazón de los DAOS.
- Es el encargado de proveer el contenido a las actividades y manejarlo con eficiencia.
- Cuando hay un dato guardado que necesita ser sincronizado, es el encargado de inicializar la sincronización del SyncAdapter.

SyncAdapter:

- Es un plug-in que nos brinda Android.
- Maneja la sincronización en una cola de ejecución.
- Se ejecuta en segundo plano cuando hay red.

NetworkLayer

- Será la encargada de realizar la comunicación con el servidor.
- Será invocada por el SyncAdapter, a excepción de los casos en los que los datos no son locales y son obtenidos desde el servidor.

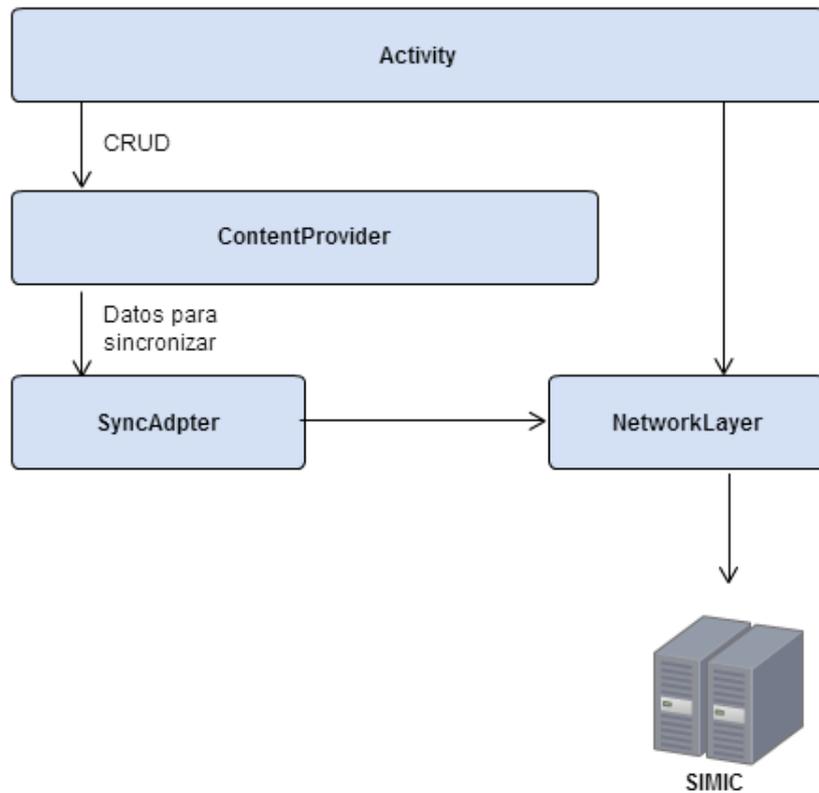


Figura 12 Arquitectura de la aplicación móvil, se aprecia la comunicación de sus componentes.

### 9.2.3 Arquitectura del FrontEnd

El FrontEnd se ejecuta en el navegador del usuario final, se encarga de mostrar la información de una forma visualmente atractiva y se comunica a través de una API REST con el BackEnd para la creación y obtención de datos.

Se utilizó AngularJS con el patrón MVC, esto nos ayudó a mejorar la implementación de JavaScript gracias a la modularización del código y así evitar la poca legibilidad que generalmente caracteriza a JavaScript.

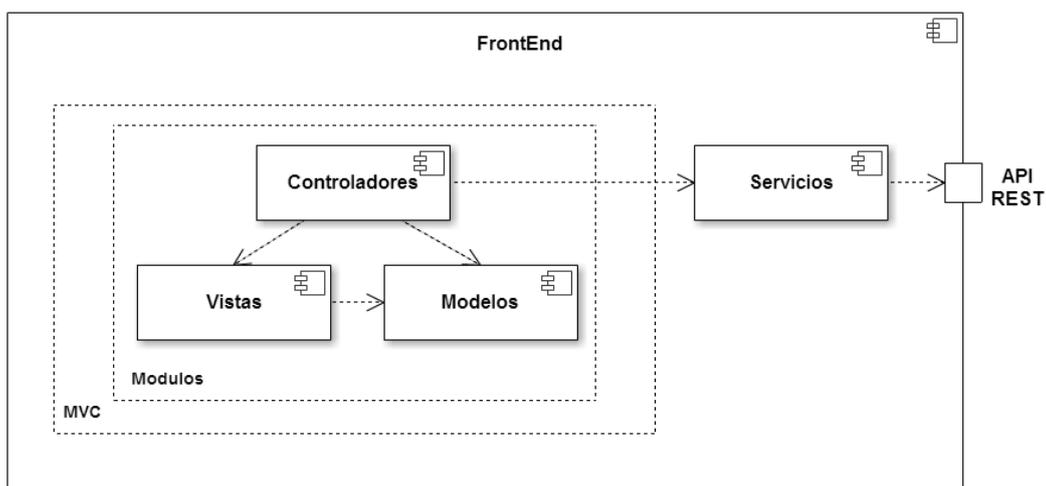


Figura 13 Se muestra el uso del uso del patrón MVC en el FrontEnd

## 9.3 Servicios REST

La comunicación entre el servidor y las distintas aplicaciones, tanto la aplicación web y el cliente móvil será través de HTTP [31].

REST siempre se diferenci6 por su sencillez ya que utiliza todas las características que puede de HTTP en vez de reinventar lo que ya tiene HTTP:

- Las operaciones a realizar
- La Estructura de la URI
- Tratamiento de errores
- El formato de los datos
- El estado de la aplicación
- Cache
- Otros.

A continuación enumeramos las principales características que usará SIMIC, ya que es muy común el mal uso de REST y a cualquier servicio HTTP se le llame mal REST.

Dado las características de SIMIC, donde toda comunicación se realizará a través de REST es muy importante enfatizar su buen uso [32], [33]

### 9.3.1 Operaciones

Por ejemplo en cuanto a las operaciones a realizar vamos a utilizar 4 métodos HTTP que coinciden con los 4 métodos de un CRUD

TABLA 1 Métodos Http

Método HTTP	Descripción	Método CRUD
GET	Este método HTTP lo usaremos para cuando queremos leer datos del servidor	Read
POST	Este método HTTP lo usaremos para añadir datos al servidor	Create
PUT	Este método HTTP lo usaremos para actualizar datos del servidor	Update
DELETE	Este método HTTP lo usaremos para borrar datos del servidor	Delete

Por ejemplo, para borrar un médico del servidor no se usa el método GET con una URL del estilo `http://miapp.com/medicos/borrar/medico` sino que utilizamos el método DELETE. Lo mismo para los restantes métodos.

### 9.3.2 La estructura de la URI

Las URI de SIMIC tienen las siguientes características:

- La URI no debe contener la acción. Así por ejemplo no debería ser /api/medico/delete, para indicar la acción ya está el método HTTP. Es decir no puede haber verbos en la URI como borrar, copiar, imprimir, cerrar, etc.
- La URI no contiene el formato de los datos que solicitamos. Esto es una característica común de Spring Framework, es decir que nuestras URI no acaban con ".pdf" ni por ".json" ni ".xml" , etc. Para solicitar distintos formatos se hace con la cabecera Accept. Esta cabecera indica el formato en que se quiere que el servidor envíe los datos. Por ejemplo si se incluye la cabecera Accept: application/json significa que lo queremos en formato JSON.
- La propia URI hace referencia al recurso sin necesidad de parámetros. Por ejemplo si se quiere el médico con id=34, la URI no es de la forma /api/medico?idMedico=34 sino que es de la siguiente manera /api/medico/34.
- Las URI pueden seguir una jerarquía, por ejemplo /api/consulta/4567/examen/67. Es decir queremos el examen 67 pero de la consulta 4567.
- Para las búsquedas o filtrado sí se permite el uso de los parámetros en las URI. Por ejemplo /api/medico/?nombre=Juan&ape1=Cano

Con todas estas reglas vamos a mostrar el ejemplo de las URI con la entidad "Usuario"

TABLA 2 Representación uso de las URI

Descripción	URL	Método HTTP	JSON Enviado	JSON Retornado
Leer un usuario	/api/Usuario/{idUsuario}	GET	Ninguno	Usuario leído
Buscar usuarios	/api/Usuario/?columnaBusqueda1=valor1&columnaBusqueda2=valor2&....	GET	Ninguno	Array de usuarios
Añadir un usuario	/api/Usuario	POST	Usuario a insertar	Usuario insertado
Actualizar un usuario	/api/Usuario/{idUsuario}	PUT	Usuario a actualizar	Usuario actualizado
Borrar un usuario	/api/Usuario/{idUsuario}	DELETE	Ninguno	Ninguno

Como se puede ver en la tabla 1 el "JSON Enviado" es el JSON que se debe enviar con los datos al hacer esa petición. Como podemos ver solo se envía al insertar o al actualizar. Es decir es el JSON del usuario a insertar o el JSON con los nuevos datos del usuario a modificar. El "JSON Retornado" es lo que nos retornará el servidor. Como se ve retorna un JSON con los datos en todos los casos excepto en el borrado, y no lo hace ya que no existe ningún dato a retornar ya que lo hemos borrado.

### 9.3.3 Tratamiento de errores

También haremos la gestión de errores siguiendo los códigos de estado de HTTP. Poder controlar todos estos códigos que brinda HTTP en la aplicación es un poco excesivo, así que nosotros vamos a simplificar el número de códigos que retornará nuestra aplicación. Por lo que usaremos 3 códigos.

- 200 OK: Si la petición ha sido realizada con éxito
- 500 Internal Server Error: Si ha habido algún error interno en nuestro propio servidor.
- 400 Bad Request: Si los datos enviados por el cliente no eran correctos. Normalmente son datos escritos por el usuario y que son erróneos. Se retorna un objeto List<BusinessMessage> en formato JSON

## 9.4 Otros patrones de diseños.

Otros patrones de diseños [34] que se tuvieron en cuenta al momento de diseñar la arquitectura y cabe destacar son:

- MVC - Modelo Vista Controlador con el objetivo de separar las distintas responsabilidades entre los componentes.
- Business Delegate, con el objetivo de ocultar los detalles de la implementación de la interfaz que presenta la lógica de negocio.
- DAO -Data Access Object -El subsistema de datos abstrae al negocio de la resolución de acceso a datos, delegando a DAOs específicos (Database Access Object). Cada DAO será responsable de implementar las altas, bajas, modificaciones y consultas sobre los datos de una entidad o grupo de entidades relacionados.

## 9.5 Vista física

En esta sección se describe el escenario general de distribución esperado para los componentes de software, las características de los nodos presentes y la comunicación entre los mismos.

### 9.5.1 Diagrama de despliegue

En la siguiente figura se presenta el escenario de distribución esperado para la instalación tanto de SIMIC Web como el SIMIC Móvil.

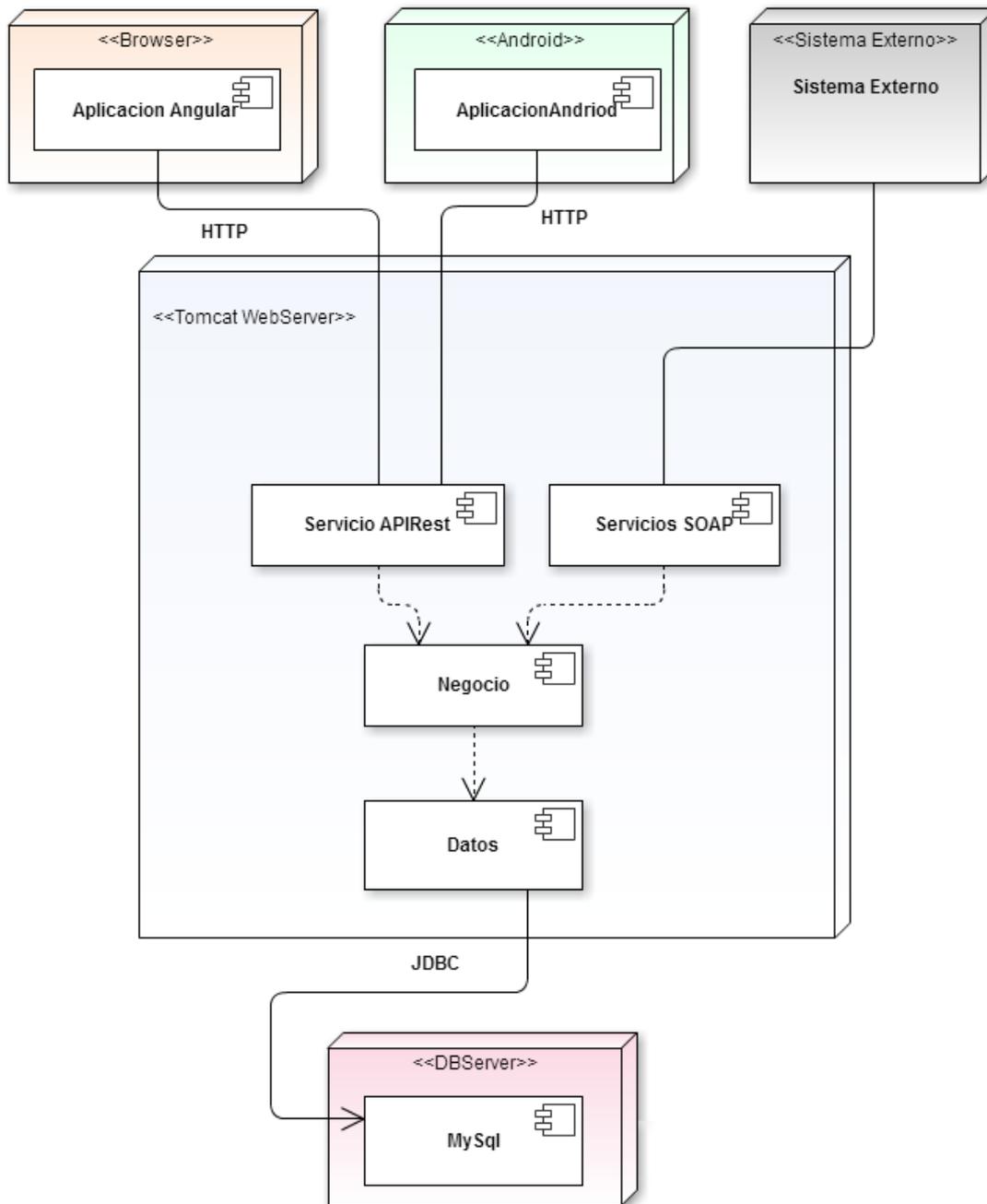


Figura 14 Diagrama de despliegue de SIMIC.

Como principal componente de la aplicación podemos destacar el web. Cuyos módulos están dispuestos en un único archivo tipo WAR, conteniendo:

- Un módulo encargado de brindar los servicios REST, tanto para los móviles como para el FrontEnd.
- Un módulo de servicios SOAP dedicado a la interoperabilidad.
- Un módulo que encargado de la lógica de negocio.
- Un módulo Data que contendrá todos los DAO y las entidades del modelo de datos.

## 9.5.2 Diagrama de componentes

El siguiente esquema muestra la interacción entre los componentes integrantes de la arquitectura del sistema.

En un primer nivel se pueden observar 5 grandes componentes:

- Sistema Externo: Representa los sistemas externos de los que consume servicios SIMIC.
- AppWeb: Aplicación web implementada con el framework AngularJS para el uso de los médicos.
- AppAndroid: Aplicación Android para el uso de los pacientes
- AppServer: Servidor donde está ejecutando el aplicativo
- Motor de base de datos MySQL: Donde se encuentra el modelo relacional del aplicativo.

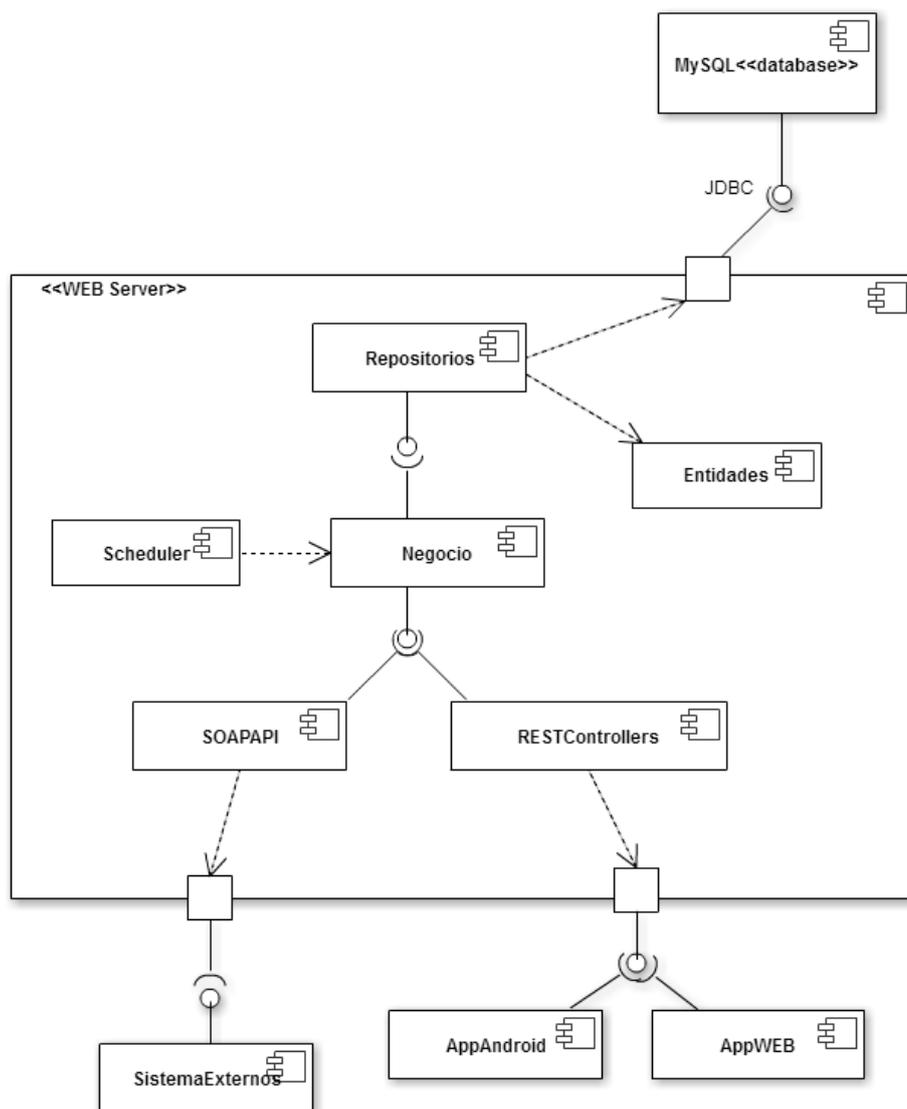


Figura 15 Diagrama de componentes de SIMIC

### 9.5.3 Diagrama de Clases – Servidor Central

El modelo de datos se adjunta como [Anexo 4](#).

#### 9.5.3.1 Descripción del modelo

En el modelo de dominio se puede apreciar como todos los datos que son considerados privados del paciente se encuentra en una entidad llamada Patronímico, la separación de los datos personales se realizó para que si un paciente solicita se pueda borrar de forma fácil toda su información personal que guarda el sistema, otra ventaja es que es una forma fácil de restringir a alguien la información que accede del paciente.

Otro desafío que se enfrentaron los integrantes de SIMIC a la hora de representar el modelo fue la representación de los distintos tipos de ítems. Una vez realizado el análisis se detectaron 2 tipos de ítems (calculados y numéricos) SIMIC optó por hacer un diseño que sea fácilmente extensible y le permitiera añadir nuevos tipos en el futuro por ejemplo archivos. Esto fue un acierto, ya que a la hora de realizar la migración apareció un tipo de Ítem nuevo, Opciones de textos.

En cuanto al Ítem calculado cabe destacar que se optó por una implementación en la que cada vez que se registra un valor en la consulta se vuelve a recalcular el ítem y se almacena este valor.

No se optó por una opción que se calcule sólo al momento de visualizar el valor.

Se prefirió la solución descrita, ya que van a ser mucho más visualizaciones que modificaciones sobre la consulta, entonces de esta forma se realizará una menor carga al sistema.

Otro punto a tomar en cuenta es que en esta solución es mucho más fácil obtener un histórico de la variación en un período de tiempo, en Ítems calculados complejos (como el score) utilizar una solución de cálculo en el momento de visualización se volvería poco menos viable.

Y el motivo más importante que justifica esta solución es lograr visualizar un histórico de un valor (con esto nos referimos al histórico auditable, no a la variación del ítem), ya que esto es un dato importante, por ejemplo saber cómo estaba el valor calculado del índice de masa corporal de un paciente cuando un médico tomó una decisión.

## 9.6 Resolución de otros problemas arquitectónicos

Un punto importante en SIMIC es la seguridad del sistema. Esto se pudo encontrar en todas las capas del mismo.

En la capa de servicios, se pudo decir que se optó por una solución en la que se utiliza HttpsSession para almacenar los datos de seguridad, confiando a los clientes la responsabilidad de guardar y enviar las cookies. Se optó por esta solución ya que la básica para un sistema en producción, y el equipo se encuentra cómodo con esto porque le permite concentrarse en las soluciones de los problemas de negocio.

Aunque dado la buena integración de Spring Framework y AngularJS, se cuenta con un abanico de opciones amplio a la hora de elegir el sistema de autenticación, y fácilmente

cambiables en caso que en algún momento se decida por ejemplo, utilizar un servidor de recursos, distinto al de servicios.

También se puede decir que la aplicación Web provee diferentes tipos de perfiles de usuarios, los que son generados por usuarios con permiso de administrador.

Dependiendo de estos perfiles el usuario accede a ciertas funcionalidades e información.

En cuanto a la Capa de negocio se puede decir que restringe la información según las reglas del negocio propiamente dichas, por ejemplo la restricción de un paciente a un médico que no está asignado.

Y respecto la capa de datos, SIMIC cuenta con una base de datos auditable, brindando la facilidad de obtener los históricos de datos.

Para el almacenamiento de la contraseña se utiliza como función de hash a bcrypt dado su gran eficiencia y la transparencia que brinda su librería a la hora de utilizar todo lo relativo al hasheo de contraseñas.

### 9.6.1 Integración con estándares de SALUD.UY

La integración con estándares fue uno de los retos con los que se encontró el equipo. Fundamentalmente los problemas que surgieron fueron a causa que los mismos están siendo evaluados para su implantación por el gobierno, al no estar en producción la información con la que se cuenta están en continuo cambio.

Al tratarse de un proyecto de grado de la UDELAR y no propio del ámbito de salud, se tuvo que tener consideraciones especiales para poder generar diferentes identificadores.

Se logró la integración con el Repositorio XDS (Historia Clínica Electrónica) luego de trabajar en conjunto con otros grupos que están o estaban realizando proyectos de grado con el mismo tutor del proyecto.

Al ser la implementación de los estándares algo nuevo, la información con la que se contaba variaba mucho, se tomó la decisión de realizar la implementación de la integración con el Repositorio XDS en su primera versión distribuida al inicio.

Otro de los estándares implementado fue el uso del servidor terminológico en ciertas funcionalidades del sistema.

No solo la poca documentación y lo poco maduro que están los estándares fueron problemas a los que se enfrentó el equipo. También se tuvieron que tomar algunas decisiones a nivel arquitectónico.

Uno de ello fue a la hora de obtener algunos datos del servidor terminológico, por ejemplo los parentescos. Estos datos tienen la particularidad de ser un conjunto bastante reducido que no varían, y son consultados muy seguidos en la aplicación.

Por lo que se decidió que corra un hilo aparte que inicialmente está configurado para que ejecute cada 10 días y actualizar esos códigos que se almacenarán en el sistema.

Y de esta forma obtener una mejor experiencia en el uso de SIMIC.

Un caso distinto es por ejemplo los datos de los medicamentos, estos son muchos y varían con más frecuencia, por lo que a la hora de realizar la consulta al sub-set de medicamentos del servidor terminológico lo hace en el instante que el usuario escribe el nombre del medicamento.

Un caso similar es el envío de los CDA al repositorio XDS. Dado que una consulta un paciente va pasando por una serie de etapas, donde primero una enfermera toma algunos datos, luego el medico puede tomar o modificar otros, y por último la nutricionista también podría agregar datos nuevos o modificar. También cabe destacar que en cualquier momento del tiempo se podría modificar una consulta existente.

En este caso no solo se tiene que definir cuándo envía el CDA, sino que también se debe definir como describir el autor de cada acto médico sin que pierda coherencia lo que se informa.

Antes de tomar una decisión el equipo evaluó las siguientes opciones:

- **Enviar el CDA con toda la información de la consulta indicando en cada entry el autor**

Ventajas:

1. El CDA tiene más sentido ya que se ve la consulta entera.
2. Se ven todos los ítems que se tomaron en una misma consulta juntos, esto puede ser importante ya que el hecho de indicar una medicación puede estar condicionado por un signo vital o un examen. O en un comentario puede dar una aclaración de un signo vital.

Desventajas:

1. Creemos que no es posible diferenciar el autor de las acciones. Supongamos que informamos el autor en cada entry del cda. No encontramos una solución conviente para informar el siguiente caso a través de esta solución: Un médico asigna una medicación, y otro médico distinto modifica el valor de la dosis de esa medicación. ¿quién es el que se la asignó? ¿a quién podríamos como autor de esa medicación?

- **Enviar un CDA por cada acto médico**

Ventaja

1. Se puede distinguir el autor de cada acto médico.

Desventaja

1. El CDA enviado carece de sentido, por ejemplo se estaría informando un CAD con los signos vitales, y la medicación en otro, o los comentarios realizados en otro. Tratar de darle sentido a la información y que sea interpretada por una persona pensamos que sería muy difícil de esta manera. Aparte de requerir mucho más procesamiento del sistema ya que la cantidad de CDA generados podría ser excesiva

- **Enviar toda la información consulta cada vez que hay un acto médico.**

Esta forma es una combinación de las dos anteriores, Cada consulta se enviaría como una actualización del documento ya enviado.

Por ejemplo, supongamos que la enfermera toma el peso y altura de un paciente. Luego el médico modifica el peso y escribe un comentario, esto se reflejaría con los siguientes CDA:

CDA1

autor:enfermera1

peso: 20kg

altura: 1.8m

CDA2

actualización de CDA1 por medico1

autor:enfermera1, medico1

peso:200kg

altura: 1.8m

comentario por medico1: necesita adelgazar.

Ventajas

1. Se puede llegar a identificar quien hizo que, teniendo todos los cda, se puede ver cómo fue evolucionando la consulta y quien hizo que.
2. Los CDA enviados tienen sentido por sí solos, no hay que juntarlos todos para ver si se envió una medicación por un signo vital tomado por la enfermera.

Desventajas

1. Se envía información repetida por ejemplo la altura no fue modificada por el médico y se envió en los dos CDA.
2. Existe una generación excesiva de CDA. Esta fue la primera opción probada por SIMIC, pero en una consulta normal se estarían generando 25 CDA, y enviando mucha información repetida en cada uno, este fue el principal motivo por el que cambiamos nuestra implementación.

Finalmente se optó por una solución que envíe los CDA disponibles cada 30 minutos con toda la información disponible en la consulta, lo que obtenemos es que la consulta este mucho más armada, y se informen más datos juntos. Ósea cada CDA enviado tiene coherencia por sí solo.

Al informar cada cierto tiempo esperamos obtener una solución mucho más estable, incluso a nivel de la coherencia del CDA.

En cuanto al cuerpo médico que interviene se envían todos los que realizaron algún acto de la consulta en el cabezal del CDA, pero sin especificar a quién corresponde cada acción.

## 9.7 Decisión de tecnologías

En esta sección haremos una comparación de las distintas tecnologías disponibles con sus ventajas y desventajas, para la implementación de SIMIC. Justificando las tecnología utilizada en cada caso.

### 9.7.1 Servidor

En el caso de la tecnología que se usó en el servidor, el equipo evaluó dos opciones (Ruby y Java) ya que son las tecnologías en que el equipo tiene experiencia.

Dado los siguientes aspectos:

- La importancia del servidor
- El conocimiento de los integrantes de SIMIC
- El objetivo de realizar una solución de calidad.

El equipo no se propuso innovar en este tema, ya que no se dispone con los recursos de realizar una capacitación de un lenguaje de servidor desconocido como para llegar a una solución de la calidad deseada, y el hacerlo sería muy peligroso.

#### 9.7.1.1 JAVA

Cada integrante del equipo de SIMIC cuenta alrededor de 4 años de experiencia en JAVA. Siendo este el principal lenguaje usado en las experiencias laborales que han tenido. El hecho de conocer esta tecnología más que ninguna les permitirá desarrollar soluciones de calidad. Los integrantes del equipo conocen bien sus debilidades. Este fue el principal motivo para decidirse a usar java como lenguaje de programación del lado del servidor.

Otro punto a favor es la gran cantidad de código ejemplos y lenguajes disponibles, probablemente mayor a cualquier otro lenguaje. Cuenta con una gran cantidad de bibliotecas conocidas por los integrantes del equipo, que brindan facilidades a la solución de problemas descritos en la arquitectura.

#### 9.7.1.2 RUBY

Ruby fue la segunda opción que tomaron los integrantes del equipo a la hora de elegir el lenguaje del servidor. El equipo también cuentan con experiencia en Ruby en mayor o menor medida, habiendo desarrollado aplicaciones, siendo esta experiencia menor a la que se tiene en Java.

Aunque no se puede asegurar llegar a una solución de calidad, como con la que podría lograr con Java con la utilización de Java. En este caso también se tendría que considerar la capacitación para lograr llegar a la mejor solución.

Como ventaja de Ruby podemos decir que es un lenguaje enfocado al desarrollo, y es un lenguaje interpretado a diferencia de Java donde se debe de compilar el código. Característica que se va a extrañar en la etapa de desarrollo.

Como desventaja, en base a la experiencia del equipo, se puede decir que el acceso a los datos en base no es de los más eficiente, lo que es una característica importante a tener cuidado.

### 9.7.1.3 JAVA EE vs Spring Framework

Dentro del mundo Java los integrantes del equipo evaluaron dos opciones el uso de Java EE o el uso de Spring Framework.

Cuando se utilizan los estándares como por ejemplo los EJB se tiene la ventaja de que nuestro servidor de aplicaciones nos aporta todo lo que necesitamos. Sin embargo no se tiene una gran capacidad para poder extenderlo y añadir nuevas funcionalidades. El servidor de aplicaciones provee de cosas buenas y malas. Esto lo convierte en un arma de doble filo, por ejemplo el uso de los EJB en algún caso puede resultar no conveniente, ya que se requeriría más recursos en el servidor. En cambio con Spring Framework podemos ser capaces de lograr soluciones para ambientes con menos recursos. Dado las características de la UMIC, el hecho de que la aplicación sea muy eficiente y pueda funcionar en un servidor de bajos recursos puede ser un aspecto muy importante, esta causa fue una de las importantes en la decisión del uso de Spring Framework.

Otro punto a tener en cuenta es que Spring Framework funciona de una forma muy distinta a los Estándares (EE) e incluye muchos puntos de conectividad/extensibilidad que permite que otros frameworks se acoplen a él. Esto hace que en las nuevas tecnologías siempre exista una solución antes para Spring Framework que Java EE.

### 9.7.2 Capa de datos

Dentro de la capa de datos, a la hora de elegir qué implementación de JPA a usar el equipo no dudo en elegir Hibernate. Dado que es de las implementaciones más eficientes, y hacen más fácil su uso.

Otra de las ventajas más grandes, que nos brinda Hibernate es Envers, con esto se logra una forma fácil de realizar la auditoría de las clases persistentes. Tan solo basta con agregar una anotación a las clases usadas.

Otros soluciones como EclipseLink también tienen sus soluciones de auditoría, pero no son tan transparentes como Envers. Uno de los principales beneficios de EclipseLink que podemos destacar es que se puede llamar a las funciones SQL nativas directamente en sus consultas JPQL. En Hibernate esto no se puede realizar directamente.

Como ventaja de Hibernate también se puede decir que cuenta con una gran comunidad, mayor y mejor documentada. Como experiencia del equipo también podemos decir que los mensajes de errores son fáciles de interpretar que los de EclipseLink.

#### 9.7.2.1 MySQL vs PostgreSQL

A la hora de elegir la base de datos el equipo evaluó las dos principales opciones de los motores de base de datos relacionales de código abierto en el mercado. Ambas son soluciones probadas en el tiempo, compiten en gran forma con los motores de base de datos pagos.

MySQL hace tiempo se supone que es el más rápido pero no ofrece tantas funciones avanzadas, mientras que PostgreSQL se supone que es un sistema de base de datos más robusto.

MySQL es más popular debido a su velocidad y facilidad de uso, mientras que PostgreSQL ha tenido su brecha en desarrolladores que provienen más de un entorno como SQL Server u Oracle.

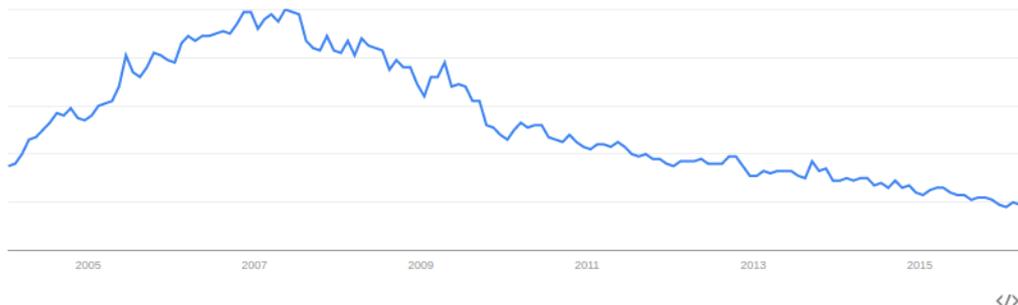
Si bien ambos proyectos han mejorado considerablemente desde que se formaron, realizar una comparación actualmente es más difícil de hacer que antes, las bases técnicas y modos de pensar de los desarrolladores que informaron estas percepciones iniciales continúan siendo una influencia dominante en ambos proyectos.

El equipo, optó por MySQL, teniendo en cuenta que es más liviano y ha obtenido muy buenas experiencias previas.

## 9.7.3 FrontEnd

### 9.7.3.1 JSF

Una vez definida la tecnología del servidor nos cabe discutir que tecnologías se usara para el FrontEnd. El estándar de java para el FrontEnd es JSF. Es evidente que el interés por JSF está decreciendo dentro de la comunidad.



*Figura 16 extraída de google trends, que demuestra la baja de interés por JSF*

JSF siempre ha gestionado su ciclo de vida a través del servidor construyendo los distintos tipos de controles que necesitamos usar. Dejando al cliente unas responsabilidades bastante reducidas.

Esto repercute en los recursos que tiene que tener disponible el servidor, aparte de ser un framework orientado a controles, es el servidor el encargado de hacer las renderizaciones de todas las páginas.

JSF facilita muchas cosas pero también tiene sus desventajas, ya que la forma de brindar sus componentes hace fácil su desarrollo pero difícil a la hora de darle estilo a la aplicación y hacerla atractiva al usuario

### 9.7.3.2 Spring MVC

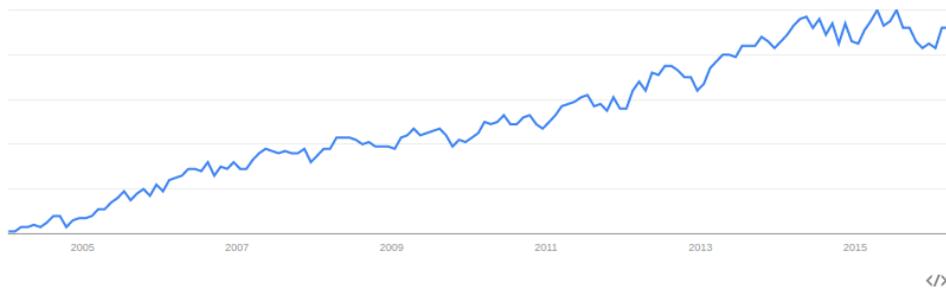


Figura 17 extraída de google trends que demuestra el creciente interes de la comunidad por Spring MVC

Spring MVC hace tiempo que compite con JSF pero su enfoque no está orientado a controles sino que se basa más en un enfoque MVC puro basado en controladores. La ventaja de este tipo de soluciones es que su integración con las nuevas arquitecturas móviles y JavaScript es más directa. También podemos decir que es mucho más acogida por la comunidad, otro punto es que los integrantes del equipo se sienten más cómodos.

### 9.7.3.3 Framework MVC cliente (AngularJS, Backbone, entre otros)

Hoy por hoy existen muchas aplicaciones que han optado por delegar una gran parte de las responsabilidades al cliente. Ejemplos claros son las plataformas móviles con HTML5, Arquitecturas SPA y frameworks como JQuery Mobile que usan simplemente el servidor para nutrirse de datos pero que todos los controles vienen definidos directamente en el cliente. Ante este tipo de situaciones JSF pierde parte de su interés.

Aparte de lo que se acaba de observar, los nuevos Frameworks MVC JavaScript que se ejecutan en el cliente se apoyan en una filosofía similar a las aplicaciones móviles a la hora de asumir responsabilidades de cliente. Con la ventaja Además, de que están orientados a desarrollar aplicaciones de gran tamaño. El hecho de delegar la parte de la lógica de la interfaz al cliente le quita carga al servidor. También se puede decir que se pueden hacer aplicaciones con un diseño de interfaz más amigable con el usuario.

Esta fue la arquitectura elegida por el equipo, a continuación vamos a comparar AngularJS con sus principales competidores

#### 9.7.3.3.1 JQuery

Podemos pensar que AngularJS debe ser algo como JQuery pero más moderno, nada más lejos de la realidad compararlo con JQuery.

JQuery realmente no tiene un mayor nivel de abstracción que el propio JavaScript estándar del navegador.

JQuery es un parche sobre los navegadores, una simple librería para que todo no sea tan tedioso de hacer pero es lo mismo que ya existe en el navegador. Con JQuery por ejemplo no existe el concepto de modelo de una aplicación o no te dice como estructurar tu aplicación etc. Por ello JQuery no es un framework sino simplemente una librería a usar por nuestra aplicación.

AngularJS juega en otra liga, es un framework que pretende definir la arquitectura la aplicación, que define claramente la separación entre el modelo de datos, la vista y el controlador, el hecho de no escribir el JavaScript como uno quiere sino como AngularJS manda le da mucho más potencia a la aplicación.

#### 9.7.3.3.2 Backbone

Backbone es uno de primeros frameworks cuyo objetivo era organizar la aplicación JavaScript. Se añade en esta comparativa ya que hace unos años era muy utilizado entre gente que quería ir más allá de JavaScript y tener una arquitectura MVC bien definida.

El problema de Backbone es que su diseño es muy minimalista y hace muy pocas cosas para comparado con otros frameworks hoy en día. Sin embargo hay que decir que Backbone más muchos plug-in se pueden hacer muchas cosas, pero eso no es la solución. Añadir tantos plug-in puede ser un dolor de cabeza. Es mucho más cómodo que el propio framework provea todas esas funcionalidades de formar central que tanto se necesitarán.

#### 9.7.3.3.3 Ember

Ember es otro framework JavaScript pero a diferencia Backbone este sí que ofrece gran cantidad de funcionalidades. Realmente es el framework contra el que compite AngularJS y nuestra elección debería ser entre uno de los dos.

Ember tiene algunas cosas buenas que no tiene AngularJS:

- Rendimiento: No es que el rendimiento de Ember sea muy bueno es que el de AngularJS no es de lo mejores sobre todo cuando hay gran cantidad de información en una pantalla.
- Rutas: El servicio de rutas de Ember es claramente superior al de AngularJS 1.2, por suerte la versión 1.3 de AngularJS mejora mucho su servicio de rutas.
- Principio de acceso uniforme: en Ember se puede acceder a un propiedad y luego cambiarla como un una función que calcula su valor y desde la vista se seguirá viendo del mismo modo. Es decir que te abstraes de si el dato es propiedad o función. Esta es una de las mejores características de Ember aunque para ello se debe pagar un precio que es no tener la sencillez de AngularJS.

Como contrapartida tenemos que el código escrito en JavaScript está lleno de código de Ember, en AngularJS siguen la filosofía de los POJOS (Plain Old JavaScript Objects) que es como los POJO de Java. Es decir que no hay que estar heredando de otras clase de AngularJS para trabajar son simplemente clases JavaScript normales. Esto fue una de las principales cosas que tomó en cuenta el equipo para usar AngularJS ya que eso nos da la oportunidad de poder hacer muchas cosas sin que pasaran cosas “raras” ya que en esencia solamente son objetos JavaScript normales y corriente.

Por supuesto no es solo lo anterior lo mejor de AngularJS, otro punto importante es la flexibilidad que da para organizar todo. Está claro que AngularJS impone unas normas y esas son de obligado cumplimiento pero son solo unas pocas. Lo que da mucha libertad.

Ember y a AngularJS a veces son comparados con el mundo Java de la siguiente manera (salvando las distancias y es simplemente para que se entienda las diferencias entre ellos). Ember es como Grails que ya ha decidido como es todo. Es mucho más fácil hacer todo pero es difícil salirte de su filosofía. Por otro lado AngularJS es como Spring Framework +Hibernate+... donde cada uno separado no hacen nada ya que cada uno no sabe mucho del otro pero los pueden juntar a tu manera y montarte tu propia arquitectura.

Por último es una cuestión de popularidad, como podemos apreciar en la tabla extraída de [35]

TABLA 3 Comparación de popularidad entre distintos Frameworks JavaScript

<b>Metric</b>	<b>AngularJS</b>	<b>Backbone.js</b>	<b>Ember.js</b>
Estrellas en Github	27.2k	18.8k	11k
Módulos de terceras partes	800 ngmodules	236 backplugs	21 emberaddons
Preguntas en StackOverflow	49.5k	15.9k	11.2k
Resultados en YouTube	~75k	~16k	~6k
Contribuidores en GitHub	928	230	393
Usuarios de extensiones Chrome	150k	7k	38.3k

Una vez elegido AngularJS otro punto a tener en cuenta es Ng-Boilerplate este es un sistema para poder generar la estructura de un proyecto de una forma muy rápida, incluye herramientas como Bootstrap, UI Bootstrap, Grunt, Less, Font Awesome y AngularJs Placeholders.

## 9.7.4 Otras tecnologías

### 9.7.4.1 En el servidor

- Maven: para el manejo de dependencias
- Jasperreports: para la generación de reportes en pdf
- Jackson: para el procesamiento de JSON
- Imgscalr: para cambiar el tamaño de imágenes
- Log4j: para la generación del log en el servidor
- JavaMail: para el envío de mail.

### 9.7.4.2 En el FrontEnd

- NodeJs: para poder ejecutar cosas como Bower y Grunt
- Grunt: como gestor de tareas a la hora de desarrollar en el frontend
- Bower: gestor de dependencias
- Twitter Bootstrap: como librería de componentes y estilos
- UI Bootstrap: implementación de los componentes de bootstrap en AngularJS

- Google Maps: para poder seleccionar la ubicación
- ngStorage: para el almacenamiento en el navegador
- moment js: para el manejo de las fechas
- fullcalendar: como calendario para el cuerpo médico.
- ChartJS: para el dibujo de gráficas.
- angular-loading-bar: para mostrar el avance de las peticiones al servidor

#### 9.7.4.3 En Android

- android.support.appcompat: como soporte de la barra de acción en las versiones anteriores
- android.support.design: para soportar elementos como el botón flotante en versiones anteriores
- butterknife: para la inyección de cosas, y resulte más fácil su desarrollo
- picasso: para el tratamiento de imágenes
- retrofit: para realizar la comunicación más fácil con el servidor
- gson converter: para convertir json a clases
- sqlite-provider: para realizar de forma fácil un provider a través de sqlite.

# 10 Calidad y migración de datos

## 10.1 Testing

### 10.1.1 General

Para la gestión de la calidad se realizó testing funcional a cada historia de usuario y pruebas cruzadas con las historias que se relaciona. Se utilizó un enfoque basado en pruebas de caja negra, comprobando que, ante el ingreso de datos, la salida de información o el comportamiento del sistema sea el correcto.

Las pruebas se ejecutaron en varias etapas, en el transcurso de cada sprint. Al momento de completar el desarrollo de una funcionalidad, el desarrollador realizó pruebas exploratorias con el fin de verificar que los principales cursos de acción de dicha funcionalidad se cumplen correctamente.

Luego, en el proceso de migración de datos, se realizó testing de todo el sistema integrado realizando la verificación de tanto de los requerimientos como de los datos cargados. Se realizó una mejor inversión del tiempo al llevar a cabo las dos tareas conjuntamente. La mayoría de los bugs encontrados en el sistema se encontraron en esta etapa.

Se dedicaron aproximadamente 100 horas al testing del producto.

A partir del tablero "Testing" en Trello creado para el registro de bugs se obtuvo la cantidad de bugs encontrados:



Figura 18 Cantidad tareas reportadas, pendientes de solución y realizadas. No todos son bugs ya que muchos son comentarios para posibles mejoras

### 10.1.2 Test de carga

Antes de empezar a describir las pruebas de rendimiento es importante dejar claro dos cosas muy importantes: 1) los valores a evaluar y 2) las dependencias.

### **Los valores para evaluar:**

- En primer lugar vamos a considerar las métricas usadas. Unos parámetros importantes que siempre deben tenerse en cuenta son: el tiempo de carga de archivos JavaScript, el tiempo de carga de imágenes, el tiempo de carga de archivos CSS y el tiempo de descarga de contenido.
- Número de solicitud y estado de respuesta HTTP.

### **Nivel de la prueba:**

Ahora bien, esto podría clasificarse ampliamente por 2 grupos:

- pruebas del lado del cliente
- pruebas del lado del servidor (a nivel API)

La mayoría de las veces las pruebas se centran del lado del servidor y API's. Dada la arquitectura de nuestra aplicación, tiene muy poco sentido solo hacer pruebas del lado del servidor ya que gran parte de la aplicación se encuentra del lado del cliente.

En aplicaciones SPA, el rendimiento depende por igual tanto en el lado del cliente y del lado del servidor. Dada la arquitectura de una aplicación SPA como la nuestra, código JavaScript y la peticiones AJAX puede incluso afectar la mayor parte del rendimiento, que la ejecución del lado del servidor.

Existen varias herramientas populares para hacer pruebas del lado del cliente como Google Page Speed o Webpagetest.org. Pero dichas herramientas no pueden probar diferentes módulos de la aplicación por separado, su funcionamiento general es probar la URL que se introduce.

Para probar diferentes secciones de su solicitud podemos seguir un enfoque diferente. Por esta razón nos decidimos a utilizar la herramienta de código abierto más popular (JMeter) para realizar las pruebas de rendimiento.

Una limitación conocida de JMeter es que no es un navegador es decir, su incapacidad para ejecutar JavaScript. Cuando JMeter hace una petición a una página las llamadas AJAX no se ejecutan automáticamente, JMeter almacena cuando se registró una solicitud Javascript y luego solo hace el GET.

Este fue el principal desafío que nos encontramos a la hora de realizar los test, para que el resultado de estos fueran representativos, para ellos utilizamos WebDriver para medir el tiempo de respuesta en un navegador tiempo real. Combinando esto con las pruebas de carga de JMeter, podemos medir la experiencia del usuario en tiempo real cuando se aplica la carga severa.

Otro punto a tener en cuenta es que todos los test se corrieron en una notebook MSI GS60 2QD (Procesador: Intel Core i7-4720HQ) con Elementary OS Freya (64bit) 0.3.2

Antes de analizar el resultado de los tests de carga es importante tener visión del comportamiento de cada hilo. En el test de carga realizado se ejecutaron distintos hilos, cada hilo realiza la siguiente iteración:

Entrar a la aplicación

Realiza el inicio de sesión y busca un paciente  
Entra al perfil del paciente, y realiza una acción.

Si analizamos estos tiempos de ejecución en un solo hilo solo obtenemos las siguientes duraciones:

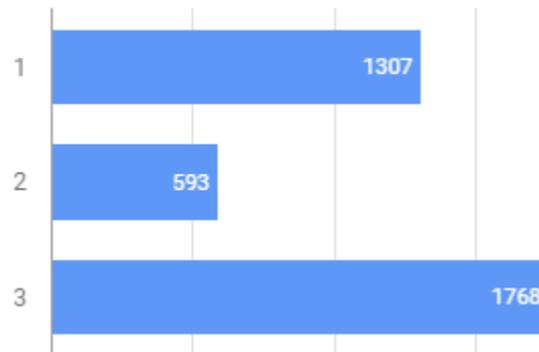


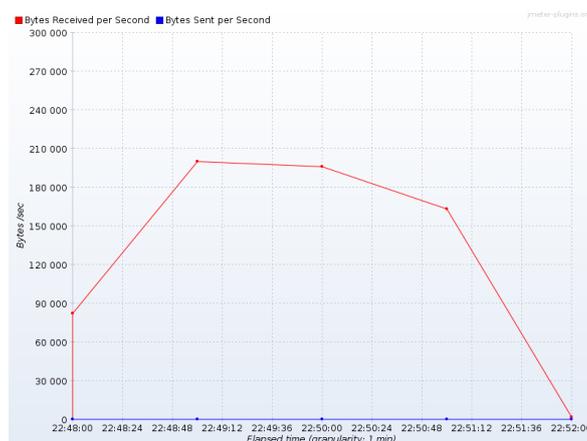
Figura 19 Tiempo de ejecución en ms de cada uno de las tareas que ejecuta el hilo

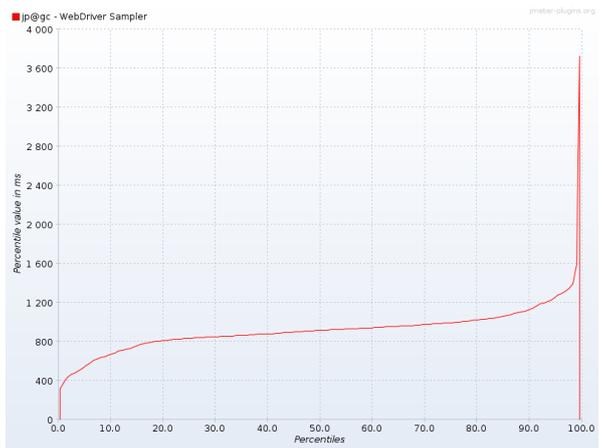
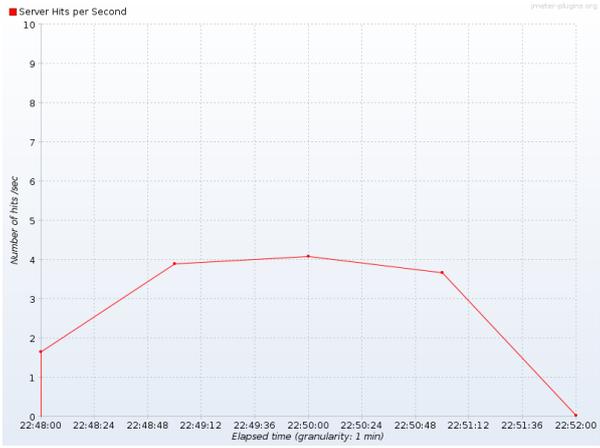
Como podemos observar uno de los tiempos más grande es el de cargar la página (1), con todos los archivos CSS y JavaScript, que se encuentran en ella.

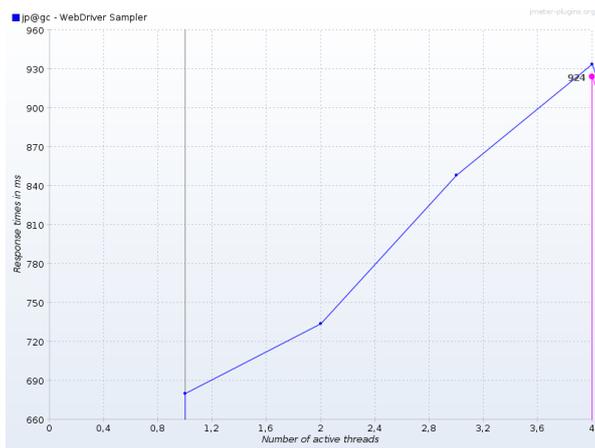
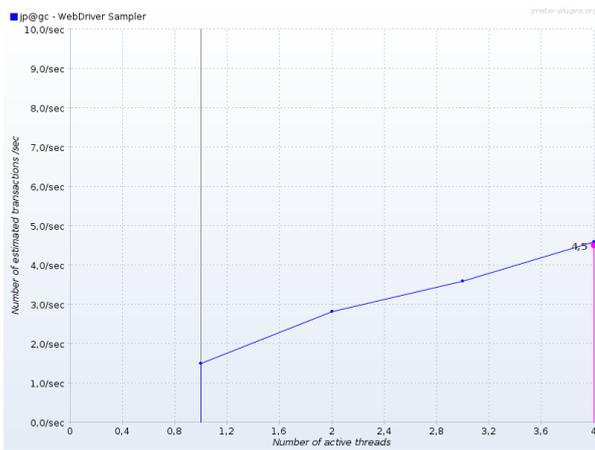
Luego de cargar la página realizar consultas sobre pacientes es extremadamente rápido ya que solo se transfieren los datos necesarios y no se vuelven a transferir todos los archivos por la red. La presentación, JavaScript, se encarga de mostrarlos de mostrarlos

Una vez obtenido el paciente se entra a su perfil y se realiza una acción. Aquí los tiempos aumentan debido a que realizar una acción requiere un procesamiento del servidor y no es un simple GET.

#### Ejecución de 4 hilos - 200 iteraciones

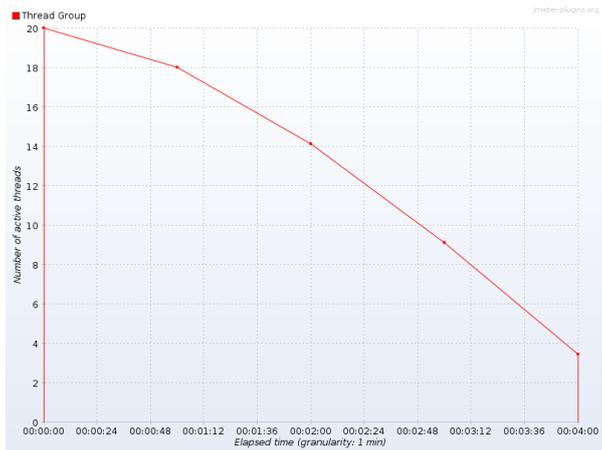
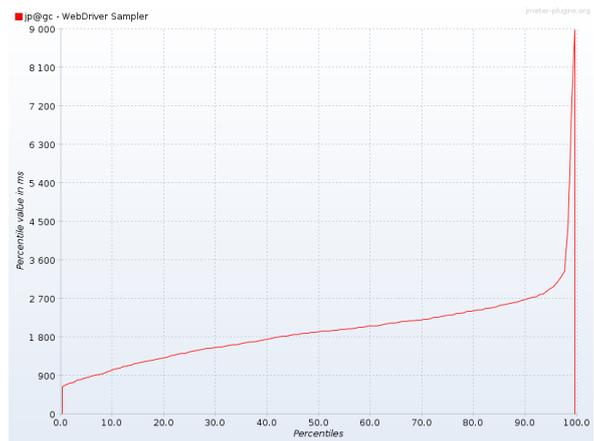


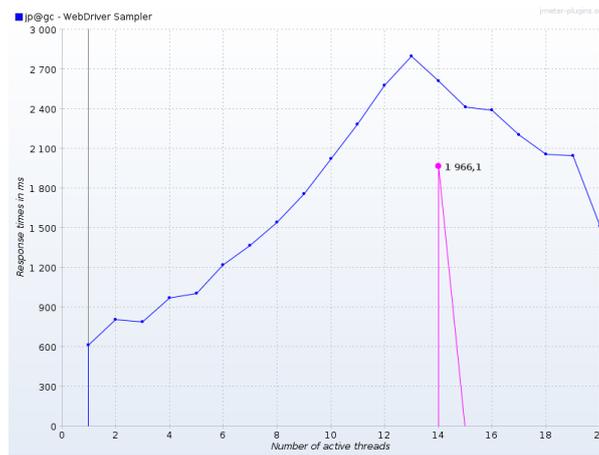
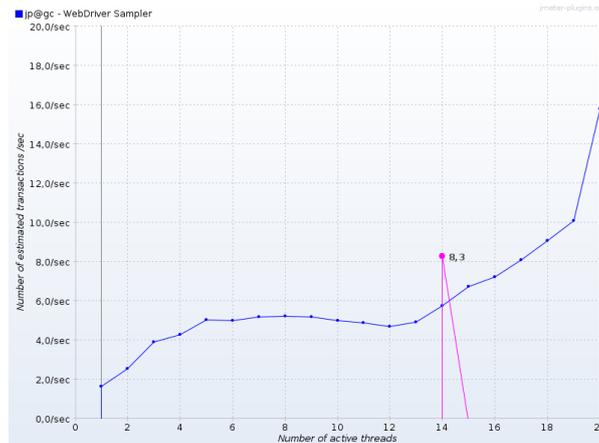




Ejecución de 20 hilos - 50 iteraciones







Tras la ejecución y análisis de los resultados obtenidos durante la ejecución de cada uno de las pruebas de carga no existe degradación en los tiempos de respuestas, ni tampoco penalización con la incorporación de nuevos hilos de ejecución, obteniendo, por tanto, las siguientes conclusiones:

- El sistema se muestra estable para periodos elevados de tiempo manteniéndose constante los tiempos de respuesta.
- El sistema garantiza que en entornos similares al utilizado durante la prueba, se asegura que para un número de usuarios concurrentes similar a 20, los tiempos medios de respuesta son inferiores a 1 segundo.

### 10.1.3 Test de seguridad

OWASP fue la herramienta que se utilizó para la realización de test de seguridad. Una de las razones que se tuvo en cuenta para su elección fue la experiencia previa de uno de los integrantes en su uso.

OWASP (Open Source Web Application Security Project) es una comunidad que produce publicaciones, metodologías, documentos, herramientas y tecnologías en el campo de la seguridad.

ZAP (Zed Attack Proxy) es una de las herramientas más importantes desarrolladas por esta comunidad. El propósito principal de esta herramienta es buscar vulnerabilidades en aplicaciones web.

Cada año, el Proyecto OWASP publica una lista de las 10 principales vulnerabilidades encontradas en las aplicaciones web, esta es una de las razones por la que también es muy conocido. Esta lista varía poco de año a año, desafortunadamente siempre se repiten los mismos errores.

Se utilizó ZIAP para obtener una mejor visibilidad del estado de SIMIC. Esta herramienta de penetración que nos permitió detectar vulnerabilidades.

Como toda herramienta automática existen vulnerabilidades que no puede detectar (como el acceso a un recurso protegido que no se protegió por omisión en la implementación) pero sí detecta otras como SQL Injection.

En primer lugar, se hizo un análisis rápido de la URL de la aplicación. Esto busca en todo lo que encuentra disponible en el sitio y testea sus vulnerabilidades. El resultado del test se encontró en los valores esperados, informando los detalles encontrados. El informe completo del uso de esta herramienta se encuentra como [Anexo 5](#).

ZAP también permite hacer un análisis mucho más profundo e incluso realizar ataques fuzzing usando la función de proxy de ZAP. Aquí es donde tiene más sentido esta herramienta.

Utilizando ZAP como proxy, se navegó a través del sitio y el equipo fue utilizando las distintas funcionalidades que provee ZIAP, como inyecciones SQL o fuzzing.

De esta forma se pudo probar mucho más la aplicación y obtener información que resultó muy útil. Cabe destacar que todos los informes informados por ZIAP fueron de un nivel medio y bajo, manteniéndose dentro de lo esperado por el equipo.

El equipo quedó muy conforme con el resultado obtenido, como se dijo anteriormente el informe generado por ZIAP se encuentra en el [Anexo 5](#), pero cabe destacar los principales detalles:

- **Manipulando Parámetros**

La manipulación de parámetros causó un error en la aplicación que no estaba contemplado. La aplicación escribió el error o Java stack trace. Esto indicaba la falta de manejo de excepciones y potenciales áreas para su posterior explotación.

- **Inclusión de Librería JS a través de CDN**

Muchas de las advertencias fueron relacionadas con algunas librerías que al momento de hacer el test se incluían a través de CDN. Estas advertencias fueron útiles ya que por omisión del equipo se había olvidado de agregar dichas librerías a SIMIC.

- **cookie without the HttpOnly**

Esta advertencia fue pasada por alto ya que indica que una cookie se ha fijado sin el indicador HttpOnly, lo que significa que la cookie se puede acceder por JavaScript. Pero en este caso dicha cookie corresponde al XSRF-TOKEN, justo es una situación en las que no se puede establecer el atributo HttpOnly, ya que cuando lo utiliza un código JavaScript (AngularJS) que requiere leer el valor de la cookie.

#### 10.1.4 Ingeniería de Muestra

Un hito importante que se presentó en el transcurso de la realización de proyecto que fue la presentación del equipo en “Ingeniería de Muestra”, donde fue la primera demostración que se realizó. La participación en el evento fue muy positiva, obteniendo una experiencia enriquecedora resultando un factor de motivación extra para la poder aportar un producto que resulte útil para los integrantes de UMIC. El equipo pudo obtener consejos por partes de médicos que concurrieron al evento, y comentarios favorables acerca del producto obtenido hasta ese momento. El proyecto fue seleccionado junto a otros para la participación en un programa radial “En perspectiva” conducido por Emiliano Cotelo. En la figura 20 se puede observar a Alejandro durante la participación en el programa [36] . Se explicó fundamentalmente la motivación del proyecto y las principales funcionalidades incluidas en el alcance. Se incluyen como [Anexo 6](#) la documentación presentada en el evento.



*Figura 20 Programa radial "En Perspectiva", participación de Alejandro*

#### 10.1.5 Integración proyecto de extensión DONOCARDIOFACIL

DONOCARDIOFACIL reúne los prototipos de proyectos de fin de carrera para su puesta en producción y ajustes en policlínicas de los hospitales de San José y de Pando para facilitar el registro de la donación de sangre con su promoción, para el seguimiento de personas con insuficiencia cardíaca mejorando su calidad de vida y para gestionar eficazmente reservas de hora en las propias policlínicas participantes.

El proyecto HEMOLOGICA incluye una aplicación web de mensajes sobre donación de sangre y ayuda al personal de salud a tratar a los donantes. El sitio de HEMOLOGICA se conecta con los sistemas de gestión de banco de sangre existentes y empodera al paciente dándole una cuenta corriente de sangre donada/recibida.

El proyecto SIMIC ayuda a asegurar el seguimiento de personas con insuficiencia cardíaca en la Policlínica especializada del Hospital de Clínicas. La vida diaria, la medicación, la sucesión de consultas y la prevención de situaciones adversas para estas personas son abordados por SIMIC en sus casas y en sus entornos naturales.

La gestión de la policlínica de cualquier especialidad obtiene con los proyectos SAMI y SAREM una elevada satisfacción del ciudadano que consulta asegurando que no se olvide, que los turnos liberados sean llenados intempestivamente y que el personal de salud tenga las tareas de preparar la lista de pacientes del día y luego redactar el parte diario muy facilitadas.

DONOCARDIOFACIL plantea instruir al personal de salud del Centro Auxiliar de Pando y del Hospital de San José por los estudiantes que le enseñan el uso de estos sistemas a la población de pacientes y acompañantes. El uso real permite que se verifiquen los postulados que dieron origen a los prototipos. Los pacientes seleccionados por los Hospitales son también capacitados en el uso de las aplicaciones en sus terminales móviles accediendo a las tres aplicaciones. Parte de la tarea de enseñanza en las policlínicas es realizada por estudiantes en “Modulo de Taller” guiados por los estudiantes autores de los sistemas de DONOCARDIOFACIL. Mediante un protocolo de documentación de indicadores de calidad de atención consensuados previamente por el grupo de extensión (estudiantes y docentes) y los gestores de los Hospitales participantes, se estimará el impacto en la mitad del período y al final del año lectivo en que se ejecutará el proyecto.

El método de trabajo incluye la formación de grupos de instrucción a cargo de los estudiantes que se enfrentan al desafío del trabajo con la sociedad (extensión universitaria) transmitiendo el uso de herramientas para familiarizarse con la práctica de centros de salud y adquirir experiencia de trabajo solidario. En la sala de espera el estudiante describe el sistema y acompaña su uso. Por ejemplo ayuda al usuario en su reserva de hora de consulta o para ver el saldo de sangre donada o en cómo registrar sus actividades diarias compatibles con su insuficiencia cardíaca.

Los tres sistemas están integrados entre sí por medio de la interoperabilidad de SALUD.uy al compartir listas de ciudadanos y repositorios de documentos médicos.

## 10.2 Migración de datos

Se explica el proceso que se siguió para efectuar la migración desde el Epi Info al SIMIC incluyendo la configuración del sistema y la migración de los datos.

### 10.2.1 Inicialización del sistema

Mediante reuniones con el cuerpo médico de la UMIC se determinó cuáles exámenes, signos vitales, Valores Calculados, comorbilidades y factores de riesgo se debían cargar en el sistema para que sea posible realizar la migración de datos desde el Epi Info a SIMIC.

#### 10.2.1.1 Factores de Riesgo

Se migraron los siguientes factores de riesgo:

- Hipertensión Arterial
- Diabetes Mellitus
- Tabaquismo
- Alcoholismo

- Dislipemia
- Exposición A Cardiotoxicos
- Hipertensión Arterial
- Diabetes Mellitus
- Tabaquismo
- Alcoholismo
- Dislipemia

#### 10.2.1.2 Comorbilidades

Se migraron las siguientes comorbilidades con sus respectivos índices de Charlson:

• Hepatopatías	1	
• Hepatopatías Moderada a Severa	3	
• Infarto De Miocardio	1	
• Insuficiencia Cardiaca	1	
• Artropatía Obstructiva	1	
• Enfermedad Cerebrovascular	1	
• Neoplasia	2	
• Enfermedad Pulmonar Crónica	1	
• Diabetes Mellitus	1	
• Diabetes Mellitus Con Lesión Órgano Blanco		2
• Enfermedad Renal Crónica	2	
• Metástasis De Tumor Solido	6	
• Plejia	2	
• Leucemia	2	
• Linfomas Malignos	2	
• Deterioro Cognitivo	1	
• Moderado Severo	1	
• Ulcera Gástrica	1	
• HIV o Sida	6	

#### Exámenes

Se migraron los siguientes exámenes:

- Caminata
- Electrocardiograma (ECG)
- Ecocardiograma (ECO)
- ECO Estrés
- Ergometría
- Centellograma perfusión
- HOLTER
- CRASIS
- Función Tiroidea
- Uricemia

- Función renal
- Hepatograma
- Serología Chagas
- Perfil lipídico
- Glicemia
- Ionograma
- Hemograma

#### Signos Vitales

Se migraron los siguientes signos vitales:

- Frecuencia cardíaca
- Peso
- ProBNP
- Talla
- Perímetro abdominal
- Clase funcional
- Presión arterial

#### Migración de Familias de medicamentos y Medicamentos

Se determinó la lista de familias de medicamentos y medicamentos que debían ser migrados al sistema para luego poder migrar los controles de los pacientes.

### 10.2.2 Exportación de datos de Epi Info

Se exportaron todas las tablas de Epi Info en formato Excel 97 / 2003.

### 10.2.3 Relacionamiento de tablas exportadas con las entidades de SIMIC

Se relacionó cada una de las tablas exportadas de Epi Info con las nuevas entidades de SIMIC. Se realizaron las correcciones de datos necesarias para la automatización de la migración.

### 10.2.4 Migración de Pacientes

Se migraron los pacientes guardando el ID que usaban en Epi Info en el campo "Key" para luego poder migrar los eventos, resultados de exámenes y controles.

En esta instancia se migraron los siguientes datos:

- Datos Personales
- Datos Clínico
  - Comorbilidades
  - Factores de Riesgo
  - Antecedentes personales

Se encontraron registros con la misma cédula de identidad, en estos casos se consultó a la UMIC que hacer con cada uno de estos registros.

### 10.2.5 Migración de Resultados de exámenes

Una vez que los pacientes estuvieron migrados se realizó la migración de sus exámenes.

### 10.2.6 Migración de Eventos y Controles

Se migró información sobre las prescripciones y valores de signos vitales ingresados en cada control. Se migraron los eventos y sus comentarios. Antes de migrar las prescripciones se tuvo que corregir datos mal ingresados en Epi Info para mantener la coherencia en el sistema.

## 10.3 Satisfacción del cliente

Una de las principales formas de determinar el éxito o fracaso de un proyecto es sin lugar a dudas la satisfacción del cliente, y por lo tanto fue una de las métricas principales. Si los integrantes de UMIC se sienten satisfechos con lo realizado durante el proyecto, se puede decir que éste ha sido exitoso.

Como se expresa anteriormente, la calidad de un producto está supeditada a la calidad del proceso con el cual se lleva a cabo. Si extrapolamos este concepto a la satisfacción del "cliente", podemos decir que la satisfacción del "cliente" no solo se colma con un producto final satisfactorio, sino que también con un proceso de trabajo en el cual se sienta a gusto. Durante el proyecto la relación con los integrantes de UMIC se fue afianzando cada vez más logrando una comunicación más fluida entre el grupo. Agradecemos el compromiso y la amabilidad de los integrantes de UMIC han tenido con el proyecto y así como con el equipo.

Al final del proyecto se realizó una medición de la satisfacción del cliente mediante una encuesta enviada a los integrantes de UMIC. La encuesta está compuesta por 10 preguntas, las cuales se puntuaron de 1 a 5, siendo; 1. Muy malo, 2. Malo, 3. Regular, 4. Bueno y 5. Muy bueno.

A continuación, se citan algunos comentarios realizados:

*"Siempre hubo buena disposición al intercambio y al dialogo"*

Pablo Álvarez

*"Hubo un gran aporte por parte del grupo de manera tal que creemos se obtendrá un producto que superará con creces nuestras expectativas"*

Pablo Álvarez

*“Será una herramienta fundamental que estar destinada a resolver grandes problemas de registro que tenemos actualmente. Me genera expectativas la posibilidad de manejar los datos en una base estadísticas”*

Pablo Álvarez

# 11 Gestión de la configuración

En cuanto a la gestión de la configuración, se utilizaron diversas herramientas que le permitió al equipo tener una adecuada gestión de los repositorios.

## 11.1 Gestión del repositorio

Se han definido dos repositorios para el proyecto, uno dedicado al almacenamiento de documentos y otro para alojar el código fuente.

Para los documentos se utilizó:

- Google Drive: fue utilizado como repositorio principal para guardar toda la documentación generada por equipo durante el proceso. Todos los integrantes cuentan con los permisos para generar nuevo contenido. Al momento de realizar el borrado de algún documento el integrante del grupo deberá solicitar a los restantes su autorización y una vez que la reciba podrá proceder al borrado. La elección de esta herramienta fue principalmente porque todos los integrantes del equipo ya la conocían y tenían previamente experiencia con ella. Además, su elección se debe a que la generación de la documentación se realiza de forma concurrente. Una vez terminada la generación de la documentación, se realizaba la corrección de detalles en la herramienta de office y luego se realizaba la subida en formato PDF.

Para el código fuente se utilizó:

- GIT: la elección de esta herramienta es que a diferencia de las restantes que se utilizan para manejar el versionado de código (ejemplo SVN) trabaja con múltiples repositorios y no con uno central. El problema de trabajar con un repositorio central es que si este se cae ninguno de los desarrolladores podrán reflejar sus cambios en el mismo hasta que éste se recupere. Trabajando con GIT cada desarrollador trabaja en su repositorio local hasta finalizado su trabajo. Se analizó la importancia de incluir su aprendizaje en el plan de formación, ya que podría solucionar posibles futuros problemas por ejemplo en cuanto a la integración de los diferentes repositorios locales con él con el repositorio en la nube. El repositorio fue alojado en Bitbucket, este fue seleccionado por el equipo por la experiencia previa de la mayoría de los integrantes en materias realizadas con anterioridad por estos integrantes. Durante el transcurso del proyecto no se tuvo caídas del repositorio, pero en el caso de que esto sucediera el equipo tenía cuenta en otros repositorios que soportan GIT, como por ejemplo GitHub.

### 11.1.1 Estructura de repositorio de documentación

A continuación se describe la estructura física del repositorio ubicado en Google Drive, en el cual tienen acceso todos los integrantes del grupo.

SIMIC: Raíz del repositorio.

- Bibliografía: Diferentes tipos de documentación usada por los integrantes del equipo para la elaboración de los diferentes documentos realizados en el proceso.
  - Marco Teórico: Documentos que relacionados a la introducción del área en la que se desarrolla el proyecto.
  - Timbó: Documentos bajados desde la plataforma Timbó.
- Reuniones: Documentos que se realizaban luego de las reuniones durante el proceso, cuentan con una breve descripción de los temas tratados por asistentes de la reunión. Además, se documentan las decisiones tomadas en las reuniones.
- Documentos: Documentos generales que se realizaron durante el proyecto, tiene la siguiente estructura
  - Documentos UMIC: Diferente documentación aportada por los integrantes de UMIC en el proceso de análisis de requerimientos (formularios, guías para paciente, entre otros).
  - Tecnología y Estándares: Documentos que se utilizaron para el aprendizaje de las diferentes tecnologías y estándares de salud que se integraron (XDS, EMPI y servidor terminológico).
  - Documentos principales: Se guardan los documentos principales generados.
    - PDF: Se guardan una vez que pasan por la validación y correcciones finales que se realizan en conjunto en la herramienta Office.
- Fotos: Fotos que se tomaron durante el proceso.
- Ing. de Muestra: Toda la documentación generada para el evento de Ingeniería de Muestra. Además, se encuentra videos e imágenes que se utilizaran en el evento.

### 11.1.2 Estructura de repositorio de código

Se tomó la decisión de tener tres repositorios para la separación del código utilizado para la generación de la solución.

- SIMIC: Se utilizó como repositorio principal del código de SIMIC Web y Además, en el mismo se encuentran diferentes estructuras de carpetas para algunas de las funcionalidades como por ejemplo para la utilización de PDF, imágenes guardadas entre otras.
- SIMIC Design: Se utiliza para el guardado de todo el código (estilos, ejemplos, entre otros) generados para la evaluación de diferentes propuestas para el diseño de interfaz de SIMC Web.
- SIMIC Móvil: Se utiliza para el código de la aplicación móvil realizada para el uso por parte de los pacientes.

### 11.1.3 Proceso de gestión de repositorio

En la siguiente figura se describe el diagrama de flujo que modela las actividades, secuencia y responsables para la detección, registro y resolución de problemas.

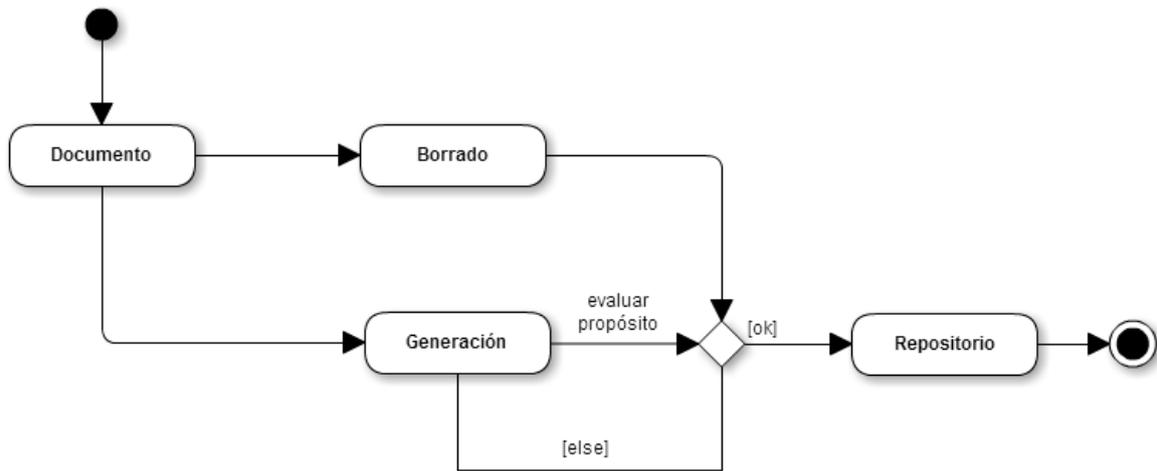


Figura 21 Proceso para gestión de repositorio de documentos, observar que para el borrado de documentos se debe de tener aprobación de todo el equipo

# 12 Gestión

## 12.1 Introducción

En el presente capítulo se presenta un resumen de las distintas áreas de gestión incluidas en el plan de proyecto, entre las que se incluyen: gestión del alcance, gestión del cronograma, gestión del esfuerzo, gestión de los recursos humanos, gestión de la comunicación y gestión de riesgos.

## 12.2 Gestión del alcance

### 12.2.1 Planificación del alcance

Inicialmente, luego de las primeras reuniones entre los diferentes actores del proyecto se elaboró una definición preliminar de un alcance a alto nivel.

Luego, se comenzó la fase de análisis de requerimientos, se pactó reuniones con de relevamiento de requerimientos con los integrantes de UMIC. Ellos tomaron los roles de cliente y usuarios finales del sistema. El lugar de las reuniones es en la propia unidad, lo cual le permitió al equipo poder conocer los requerimientos y problemas desde su lugar de trabajo.

Los requerimientos relevados en dichas reuniones se incluyen en el documento de Especificación de Requerimientos (ESRE). La fase de análisis se complementó con la generación de MockUps de diferentes partes de la interfaz que el equipo entendía importantes para la facilitar usabilidad del sistema.

Con la especificación de requerimientos, realizada como historias de usuario, las cuales luego se incorporan al “product backlog”.

Los registros del “product backlog” son priorizados en conjunto entre los integrantes de UMIC y el equipo. La priorización y selección de requerimientos se realiza al comienzo de cada sprint, la idea es que los requerimientos más priorizados (requerimiento factible de cambio, básico en el funcionamiento de sistema, entre otros) tengan mayor tiempo de desarrollo.

### 12.2.2 Cambios del alcance

Al finalizar cada sprint de SCRUM se realizará una verificación del cumplimiento del alcance determinado. Dicha verificación se realizará mediante una reunión con los integrantes de UMIC donde a través de una demo del sistema se mostrarán los requerimientos alcanzados. Los requerimientos planificados para el sprint que no llegaron a completarse deben de planificarse para el siguiente.

Se permitirá incluir cambios en el alcance durante las reuniones de planificación de Sprint, la idea es que del equipo es poder consentir a los integrantes de UMIC para poder aportar un

sistema que sea de utilidad para unidad a raíz de que la experiencia de la unidad con sistemas informáticos no ha sido buena.

El costo de cambio de dicho alcance será calculado por el equipo, y se deberá verificar la priorización de dicho requerimiento por sobre otros definidos anteriormente.

Se determinará el alcance final del proyecto con el tutor en donde se establecerá la finalización del proyecto.

### 12.2.3 Control del alcance

El alcance del proyecto fue supervisado por los integrantes del equipo, en general el integrante designado con el rol de gerente de proyecto era el que tenía la obligación de tener más presente el control.

Los integrantes del equipo actualizaban en avance de las distintas tareas que tenían asignadas por intermedio de la herramienta Trello, en la misma se tenía una categorización por intermedio de etiquetas las cuales se asignaban a la tarea. Este control se vio reforzado con las discusiones de los retrasos realizadas durante por el equipo en el proceso, fundamentalmente en su etapa de desarrollo.

Esta información es complementada Además, por el burndown chart generado automáticamente generado por intermedio de un plug-in para Trello que daba soporte para su elaboración.

Se permitió incluir cambios en el alcance durante las reuniones de planificación de Sprint entre los integrantes del UMIC y el equipo, la idea que del equipo fue poder consentir a los integrantes de UMIC en la mayoría de las solicitudes para poder obtener una solución sea de utilidad para la unidad.

### 12.2.4 Avance del alcance

Inicialmente se planificó ejecutar 5 sprints de desarrollo, permitiendo desarrollar dentro del calendario un total de 595 puntos. El Product Backlog inicial contaba con un total de X puntos. En el transcurso de las interacciones se ingresaron cambios y nuevos requerimiento y por un total de 360 puntos.

Al momento del cierre del proyecto se han completado 955 puntos, lo cual equivale a un 160% del total del proyecto.

Esta variación del alcance, podemos decir que se vio incrementado debido a errores de estimación del equipo en algunos casos y Además, en parte a la influencia de algunos requerimientos que no dependen del equipo estrictamente, como la integración de los estándares de SALUD.UY.

## 12.3 Gestión del tiempo

Para la planificación del calendario del proyecto se realizó la elaboración de un cronograma inicial, el cual se encuentra disponible en la herramienta Trello, en un tablero con el nombre de “Plan de Proyecto”. La figura 21 muestra el plan de proyecto previsto.

El seguimiento del proyecto se realizó utilizando las alarmas de fecha aportada por la herramienta.

Durante el transcurso del proyecto se produjeron algunas demoras en el cumplimiento del plan, algunas eran esperadas por equipo a causa de la falta de experiencia a la estimación, trabajar con tecnologías nuevas recientemente aprendidas y la dificultad de trabajar en un área difícil como es la salud la cual los integrantes del equipo desconocían.

Un cambio importante en reflejar fue la generación de nuevos sprint para el desarrollo llegando a realizar un total de 7 sprint, Además, se realizó un sprint denominado arreglos de interfaz que se fueron resolviendo por los integrantes durante los demás sprint.

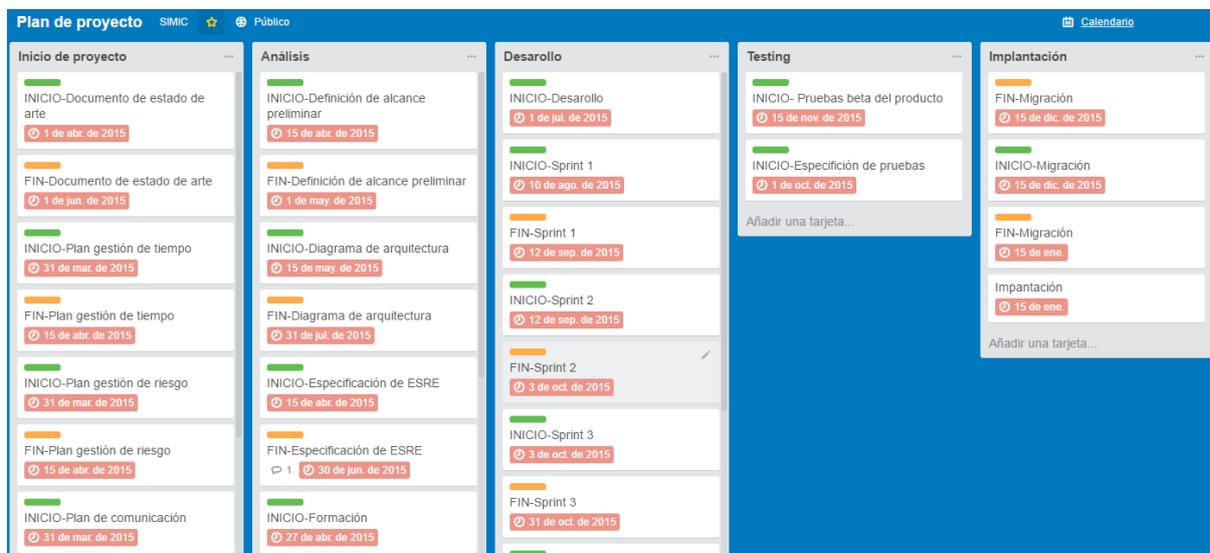


Figura 22 Apariencia identificativa de la herramienta Trello, board de Plan de Proyecto

Para realizar el seguimiento durante las iteraciones de desarrollo se creó otro tablero con el nombre de “Desarrollo” en el cual se crearon diferentes listas en las cuales se encuentran las historias de usuario con su correspondiente estimación de esfuerzo y su esfuerzo real. Para indicar el estado de desarrollo en el que se encontraba cada historia de usuario, se utilizó una categorización por medio de diferentes etiquetas las cuales eran asignadas a las mismas. La figura 22 muestra el tablero de desarrollo donde se muestran las diferentes listas de los sprints realizados.

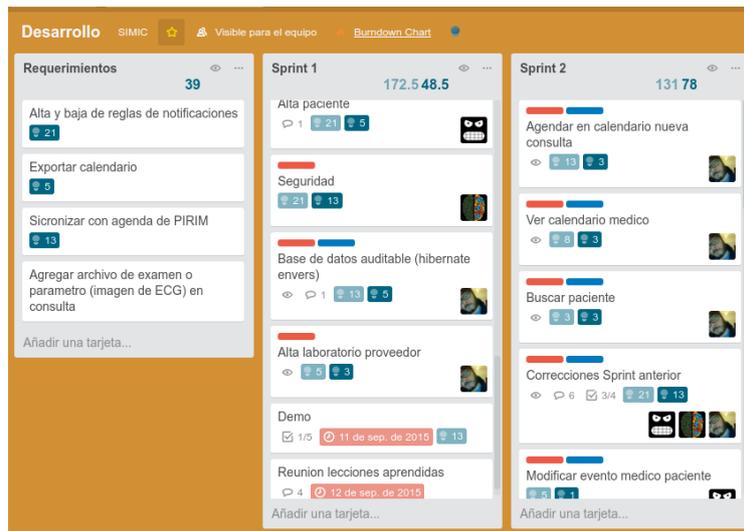


Figura 23 Board Desarrollo extraído de la herramienta Trello, se pueden observar las diferentes tareas y asignaciones a los integrantes

## 12.4 Gestión del esfuerzo

### 12.4.1 Planificación y control del esfuerzo

Al comienzo del proyecto, el equipo estableció la forma de registrar el esfuerzo de cada integrante. En esta instancia se creó un formulario a través de google por el cual cada integrante registraba la cantidad de horas dedicada diariamente a cada tarea.

Durante el transcurso y no muy avanzado el proyecto se decidió cambiar a una herramienta con más opciones que era compatible con la Trello herramienta donde se llevó la gestión del proyecto. Esta decisión no fue muy acertada a causa que la herramienta en su versión gratuita tiene una fecha de expiración la cual se terminó antes de la culminación del proyecto. Se volvió a utilizar el formulario de google forms.

La forma registrar el esfuerzo en la etapa de desarrollo (Además, de registrar las horas diariamente) fueron los “story point” unidad que se utiliza en el proceso de SCRUM. Esta forma se realizó a través de un plug-in que se integró con la herramienta Trello donde se registró un board llamado “Desarrollo” en el cual se dividieron las “historias de usuario” con la estimación de “story point” que el equipo había determinado para su resolución. Luego la resolución de cada “historia de usuario” el integrante encargado de la misma registraba la cantidad de “story point” que realmente se dedicó. De estos datos se pudo obtener la velocidad de esfuerzo del primer sprint, con esta se observó que la primera estimación fue errada, fue el momento en el que se tomó la decisión de contar con más sprint y reducir la cantidad de “story point” estimada para la elección de las historias a realizar.

Estas son algunas de las métricas que el equipo, estas métricas principalmente fueron aplicadas para la realización de la fase de implementación y como se explicó anteriormente se realizaron a base de los denominados “story point”:

- Índice de rendimiento del esfuerzo: Este indicador se calcula de la siguiente manera:
 
$$\text{índice de rendimiento del esfuerzo} = \frac{\% \text{Historias completadas} \times \text{Esfuerzo estimado}}{\text{Esfuerzo realizado} \times 100}$$
 El valor de este indicador permite identificar rápidamente si el esfuerzo dedicado para el cumplimiento del desarrollo en las historias de usuarios hasta el momento fue menor o mayor al esperado.
- Gráfico de esfuerzo estimado vs esfuerzo realizado
 Este gráfico permite identificar rápidamente y de forma visual el ajuste al esfuerzo estimado, poder ajustar en función a lo observado la distribución de este esfuerzo.

## 12.4.2 Avance del esfuerzo

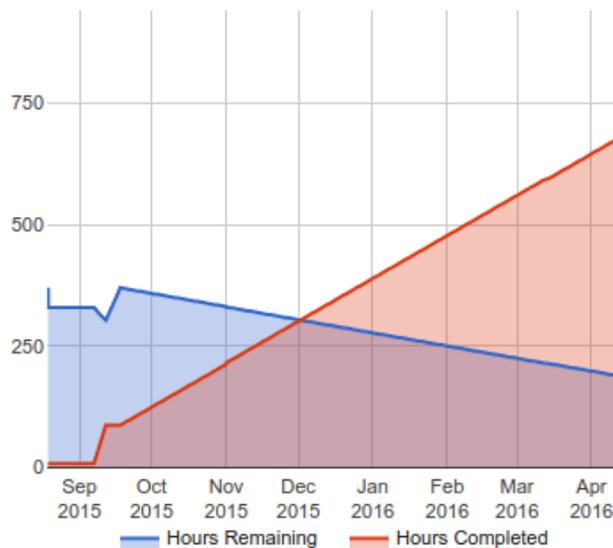


Figura 24 Burndown Chart, se puede observar la poca experiencia previa con la que contaba el equipo para realizar estimaciones precisas

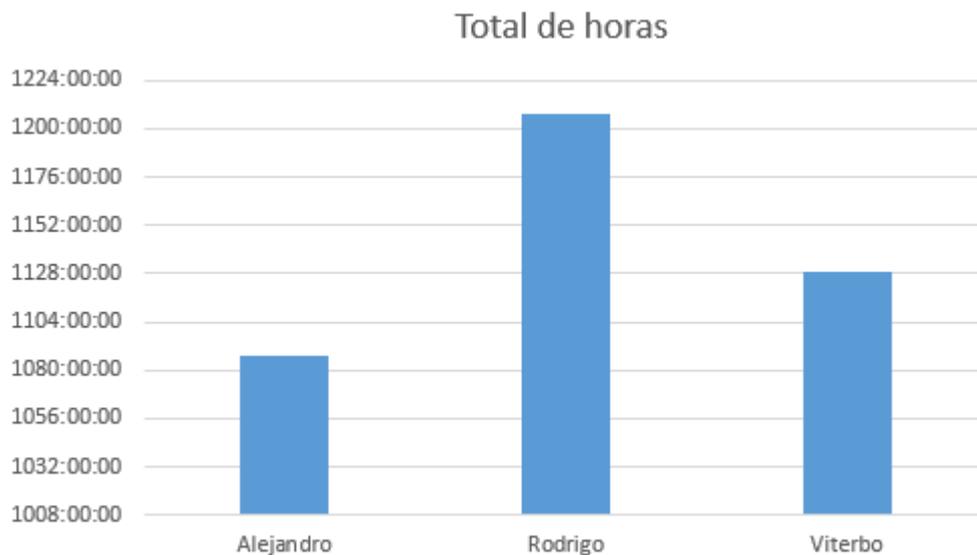


Figura 25 Total de horas por integrante del equipo

### Evolución de horas

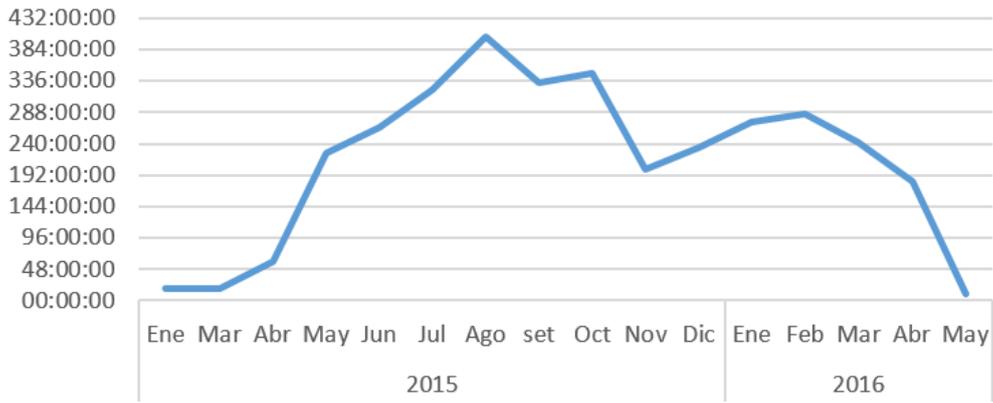


Figura 26 Evolución en el tiempo de las horas dedicadas por el equipo

### Evolución de horas Implementación

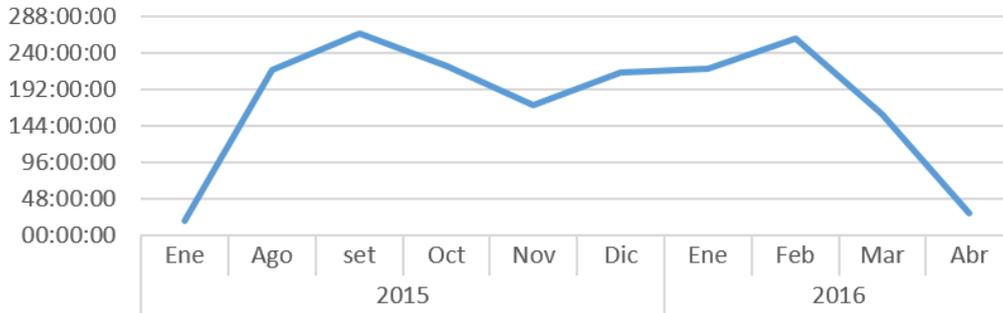


Figura 27 Evolución de las horas dedicadas por el equipo para implementación

## Distribución de horas

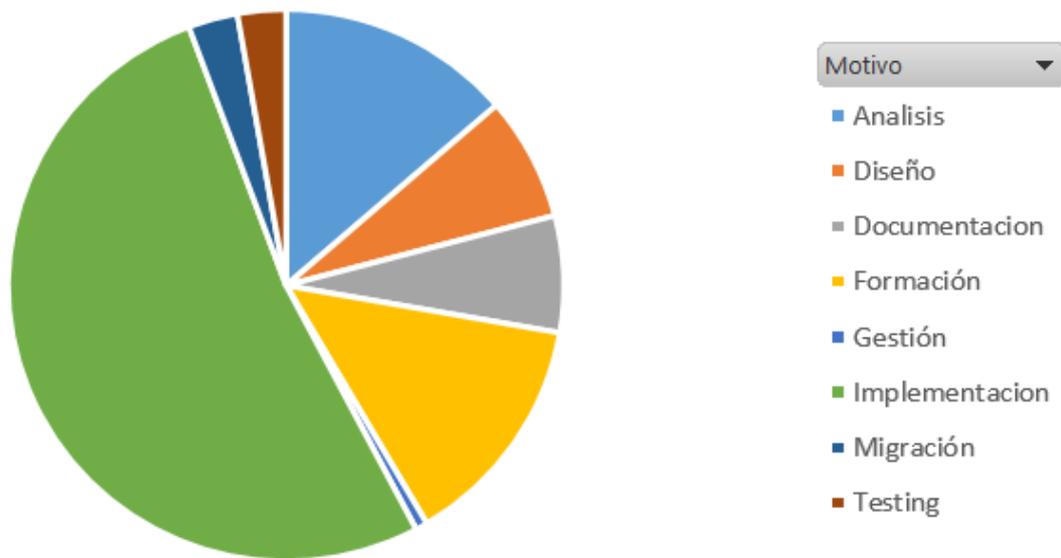


Figura 28 Distribución total de las horas según las diferentes tareas, se aprecia la cantidad de horas dedicadas a implementación

## Total de por cada integrante

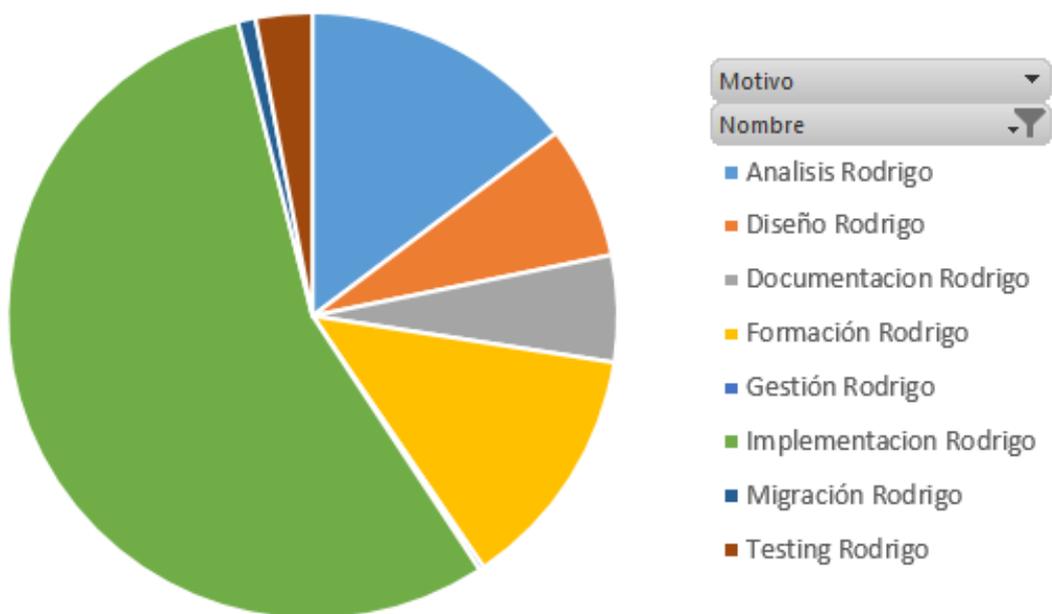


Figura 29 Distribución de horas de Rodrigo

### Total de por cada integrante

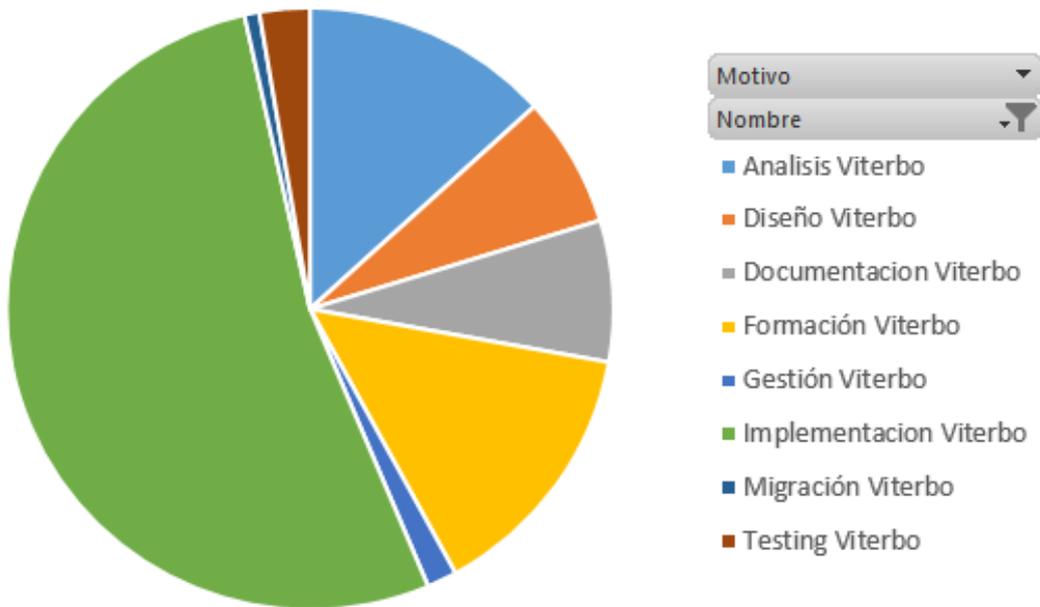


Figura 30 Distribución de horas Viterbo

### Total de por cada integrante

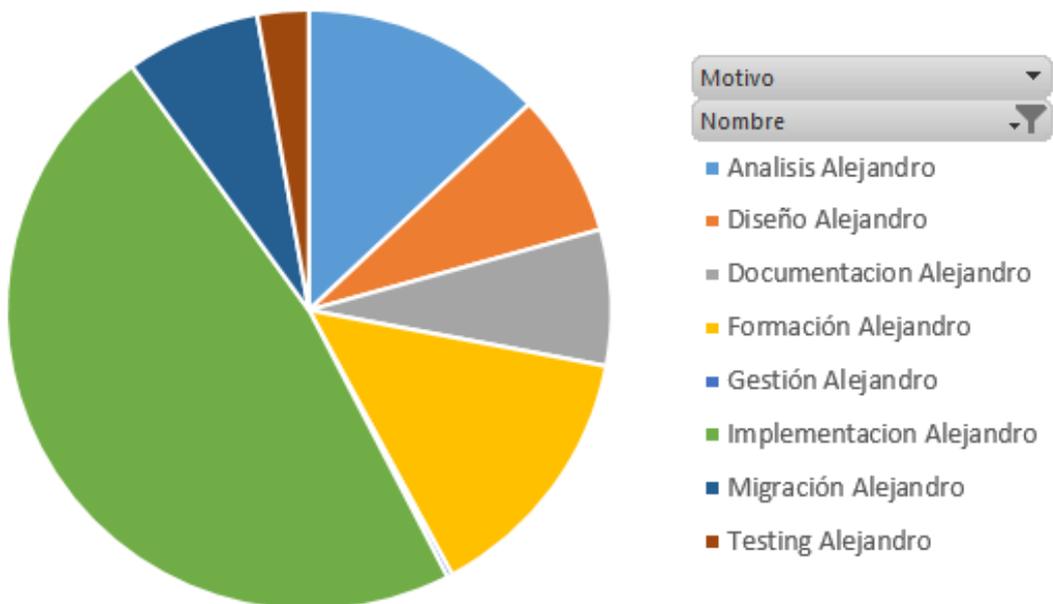


Figura 31 Distribución de horas Alejandro

De las gráficas las observaciones que resultan interesantes son:

En particular, el alto tiempo dedicado a la implementación de la solución. Se puede observar que la cantidad de horas dedicadas a la implementación de BackEnd es similar a la de FrontEnd lo que refleja el compromiso asumido por el equipo en cuanto a la importancia de la realización de una interfaz que sea intuitiva y fácil de usar. Además, se pueda acercar a

los a los profesionales del área de salud a la utilización de herramientas informáticas que sean de utilidad para la realización de su trabajo y sea de su agrado.

Otra observación de la Figura 21, en cuanto a la gráfica de la distribución de las horas dedicadas a la implementación en el tiempo es que se ve el hito importante que se presentó en el transcurso de la realización de proyecto que fue la presentación del equipo en “Ingeniería de Muestra.

En el gráfico de burndown que se utiliza, se puede observar la falta de experiencia en el equipo al momento de estimar los “story point” para las historias de usuarios. Durante el proceso se fue corrigiendo, pero igualmente se ve principalmente en el inicio que fue casi el doble el esfuerzo realizado sobre el esfuerzo estimado.

El grupo le da también su responsabilidad a esta distribución (figura XX) de horas al uso del proceso de metodologías ágiles. El uso de herramientas utilizadas como apoyo para la realización del proceso por ejemplo Trello permitió por ejemplo la simplificación de la gestión de las tareas en el proceso de desarrollo fundamentalmente, logrando una disminución de las horas dedicadas a la tarea de gestión.

También puede notarse un alto tiempo de formación, principalmente debido a la utilización de AngularJS, tecnología seleccionada Además, por sus beneficios aportados a la solución como también el desafío que se planteó al momento del inicio del proyecto de poder aprender nuevas tecnología, buscando ser una motivación extra para el equipo.

## 12.5 Gestión de los Recursos Humanos

### 12.5.1 Normas de funcionamiento

Al comenzar el proyecto se decidió también establecer ciertos acuerdos y normas de funcionamiento del equipo, de manera de organizarse de una mejor manera y evitar problemas en el transcurso del proyecto. Una ventaja para esta situación es que el equipo previamente ya había trabajado de forma conjunta. Con esta idea en mente es que se realizó una reunión en la que se intercambiaron las opiniones de cada integrante acerca de temas como asignación de tareas, “roles” del proyecto, capacitación, agenda de los integrantes, reuniones, comunicación entre los distintos actores del proyecto y compromisos asumidos.

### 12.5.2 Capacitación

El tiempo estimado para la formación fue de alrededor de 6 semanas. La capacitación será realizada en una etapa inicial donde cada integrante se formará individualmente y luego una etapa de forma conjunta en donde se discutirá y se analizará el aprendizaje de cada tecnología.

- Formación en SCRUM: Obtener el conocimiento necesario para poder gestionar el proyecto utilizando la metodología SCRUM. La formación se realizará a través de la lectura del libro “Gestión de Proyectos con SCRUM”. Se generará un Documento indicando generalidades de SCRUM y cómo se realizará su uso en el proyecto.

- Desarrollo de aplicaciones para Android: Los integrantes podrán desarrollar aplicaciones para Android e instalarlas en una Tablet. Podrán crear aplicaciones que utilicen la conexión web de la Tablet. Actualmente uno de los integrantes cuenta con conocimientos sólidos en la tecnología a modo que actúa principalmente como líder en la tecnología y será el encargado del desarrollo de la aplicación móvil. La estrategia de formación a utilizar será el autoestudio mediante libros indicados por el referente
- Formación versionado de código: Obtener los conocimientos sobre Git herramienta seleccionado por el equipo para el versionado del código. Los recursos formados deben ser capaces de obtener y realizar actualizaciones del repositorio y principalmente deberá de forma precisa poder realizar la resolución de conflictos. La formación se realizará de forma individual a través del curso disponible[37].
- Estándares de salud a ser integrados: En el inicio se realiza principalmente con la documentación que está disponible en el portal de HCEN. Luego por intermedio de reuniones que se coordinaron con los responsables Paulo Sander y Walter Callero Además, de reuniones con Lucia Grundel. Se espera el evento que se realizará por parte de HCEN denominado “Connectathon” donde se podrá verificar la intercomunicación entre los diferentes sistemas de centros de salud que están en proceso de integración.
- AngularJS: Fue donde se centralizó la formación, Además, de la propia tecnología se tuvo que tener en cuenta el uso de esta de forma correcta. Por ejemplo se tuvo que ver tecnologías como:
  - Grunt: Es una herramienta para automatizar tareas es nuestros proyectos, especialmente en el desarrollo de proyectos web. Se utilizará en las tareas fundamentalmente en las rutinarias que deben hacer en nuestro proyecto y para la detección de errores en tiempo real en el código JavaScript.
  - Gulp: igual que Grunt, se utilizó para el código de estilos ya que la tecnología que se utilizó lo recomendaba.
  - Bower: Bower es una herramienta permite encontrar proyectos y herramientas de JavaScript, y poder administrarlas dentro de nuestro proyecto. Esto permite forma fácil descargar y actualizar paquetes como JQuery, Bootstrap, entre otros.

Para la formación en AngularJS se utilizó principalmente el curso que se tiene disponible en su página [38]. Además, se formó en la integración por intermedio de servicios web de AngularJS con Spring Framework desde [33].

Luego se realizó un prototipo que incluyera un inicio de sesión y un ABM de un elemento. Con esto se pudo realizar la práctica, para ver la arquitectura y definir las capas del sistema.

- Base de datos auditable: para poder soportar los cambios en cualquier variable del sistema, ya que así lo recomiendan las leyes y normas se tuvo que realizar una formación en Hibernate Envers que provee a Hibernate del soporte para base de datos auditable.
- Normas y leyes de la salud involucradas con el proyecto.
- Diseño de interfaz “Material Design”: en la etapa inicial no fue especificada. Al tomar la decisión de contar con una interfaz fácil de usar y agradable para los integrantes

de UMIC se decidió la formación de uno de los integrantes del equipo para la formación de las guías publicadas por Google. La documentación que se uso fue la que está disponible [26].

## 12.6 Gestión de la comunicación

El plan de comunicaciones se basa mayormente en reuniones, siendo estas de dos tipos, las reuniones periódicas y las reuniones bajo demanda.

Las reuniones periódicas son las reuniones que se realizan rutinariamente según como se indica en la metodología SCRUM. En particular para este proyecto se define que se realizarán las reuniones de SCRUM de la siguiente manera:

- SCRUM planning: 2 horas de planificación del equipo a realizar el primer día del SCRUM, luego de la presentación con el cliente del sprint anterior para poder contar con los feedbacks.
- Daily meeting: No se realizarán de forma diaria debido a que durante la semana se dedicará la menor cantidad de horas del proyecto. Por tal motivo se define que las Daily meeting se realizarán los días martes, jueves, sábados y domingos. Para contrarrestar esta decisión cada integrante del equipo notificará al resto la tarea a la que estará abocado durante la semana.
- SCRUM review: 2 horas de demo al cliente y planificación del siguiente SCRUM. Se realiza una vez finalizado un ciclo de SCRUM.
- Retrospectiva: 2 horas de reunión de grupo a realizar luego del SCRUM review.

### 12.6.1 Comunicación entre los integrantes del equipo

Se utilizará correo electrónico para las comunicaciones que se realicen por escrito. Aquellos correos electrónicos que sean dirigidos a todo el grupo serán enviados a la casilla genérica.

Para las discusiones se utilizarán los comentarios en la herramienta trello, creando para cada tema en especial a tratar una nueva lista en la organización de los tableros creados al inicio del proyecto.

Para las comunicaciones que requieran interacción de las partes se utilizará Google Hangout, grupo creado de Whatsapp, sms y como última instancia si se requiere se podrá realizar la llamada telefónica.

La asignación de tareas se realizará utilizando la asociación de miembros a la tarjeta correspondiente, estas categorizaran (en proceso, urgente, completado) usando etiquetas diferentes. Esto principalmente se utilizó en la etapa de implementación para observar el avance que se iba produciendo en cada sprint y poder ajustar para cumplir con lo planificado.

## 12.6.2 Comunicación con el tutor

Se realizará la comunicación por intermedio de correo electrónico principalmente desde la cuenta creada por el equipo para el proyecto. Para que todo el equipo reciba copia de la comunicación se deberá poner como copia oculta la propia cuenta.

## 12.6.3 Comunicación con los integrantes de UMIC

Se realizará la comunicación por intermedio de correo electrónico principalmente desde la cuenta creada por el equipo para el proyecto. Para que todo el equipo reciba copia de la comunicación se deberá poner como copia oculta la propia cuenta.

Además, se cuenta con los números telefónicos de la unidad, de los integrantes responsables. Se utilizarán con la necesidad de una comunicación que deba realizarse de forma urgente. Durante el proceso no se requirió la utilización de este medio de comunicación.

## 12.7 Gestión de riesgos

Los riesgos en el proyecto fueron identificados de manera grupal, a través de la Aplicación de la técnica de Brainstorming, incorporando Además, riesgos identificados en la tarea de investigación que se realizó al inicio del proyecto documentada como “Estado de Arte”.

Luego fueron ponderados utilizando la técnica de Poker Planning (Figura 31) recomendada en el proceso de SCRUM. A partir de los riesgos identificados se realiza una priorización y cuantificación de los mismos. Para aquellos de magnitud alta o moderada se elabora un plan de respuesta y contingencia de forma de reducir el impacto.

Algunos de los riesgos de mayor magnitud durante el transcurso del proyecto son los siguientes:

- Tiempos planificados para las tareas son menores a tiempos reales por la falta de experiencia del equipo. En ciertas tareas, principalmente en la etapa de desarrollo se decidió aplicar una fecha de finalización para la misma, logrando así que pese a que la tarea requirió un esfuerzo mayor no produzca un retraso en el proyecto.
- Tareas y requerimientos no identificados. Como plan de mitigación se definió en cada sprint la posibilidad de incluir estas siempre y cuando el equipo este en fecha con el sprint.
- Requerimientos inestables. Para mitigar dicho riesgo se definió en el documento de “Planificación y contrato” donde se aclara el proceso que se tendrá en cuenta para los cambios de requerimientos. El equipo igualmente quería aportar la mejor solución posible, esto se vio reflejado en requerimientos propuestos por los mismos integrantes.
- Complejidad de tecnologías atrasa el desarrollo del producto. Se realizarán reuniones grupales para que el grupo se enfoque en el problema y poder resolver el inconveniente.

- Problemas a la integración de estándares de SALUD.UY: al no tener producción el uso de estos estándares el equipo cree que hay un gran riesgo. Para poder contar con la ayuda necesaria se acudirá al tutor principalmente para poder tener un mayor contacto con los integrantes de AGESIC responsables de los estándares.
- Problemas en la comunicación con los integrantes de UMIC: Al ser un área donde se trabaja muchas horas, se ve como un riesgo el tiempo disponible que puedan tener los integrantes de UMIC principalmente para la etapa de análisis de requerimientos. Se resolvió poder contar con más de un medio de comunicación y agregar como riesgo “requerimientos inestables”.
- Complejidad del área. El área en la que se desarrollara el proyecto es nueva para los integrantes del equipo. Lo que se resolvió en invertir más horas de formación en los temas necesarios para poder involucrarse en mayor medida con el área y realizar una comunicación más fluida con los integrantes de UMIC.
- Migración de datos. Se les explica a los integrantes de UMIC el proceso y se aclara que pueden surgir algunas inconsistencias en los datos, datos que se cuentan no son precisos y contienen errores. En forma conjunta se tomará la decisión de cómo resolver cada problema en el momento que surja.
- Uso de la aplicación móvil: Durante la investigación para introducirse en el área, se observó que un problema detectado es al uso de la tecnología por parte de los pacientes. Una de las formas de mitigar este riesgo era realizando un estudio de la población de paciente de la unidad, principalmente para observar el conocimiento de cierta tecnología y el acceso que pudieran tener a ella. El equipo elaboró un cuestionario para que los integrantes de UMIC pudieran realizarles a los pacientes y así poder contar con cierta información para poder evaluar el riesgo. En una reunión con el tutor del proyecto se resolvió no realizar dicho cuestionario y realizar la aplicación móvil con el objetivo de que las venideras generaciones la tuvieran en caso de que no se utilizara en gran forma actualmente.



Figura 32 Planificación del equipo, se observa las cards usadas para el Poker Planning

# 13 Conclusiones

## 13.1 Análisis de gestión

### 13.1.1 Alcance

El alcance al cual se llegó es superior al planificado y por tanto que el resultado del proyecto es positivo. Dicho logro se basa fundamentalmente en los siguientes aspectos:

- Una metodología que permitió adaptarse con éxito a los requerimientos inestables a los cuales estuvo sujeto el proyecto.
- El compromiso asumido por los actores del proyecto en la realización de un producto que sea de utilidad para la unidad de UMIC.

### 13.1.2 Calidad

En cuanto al producto, se piensa que está en un nivel aceptable, se llegó sin errores relevantes pendientes de corregir. Las correcciones pendientes son principalmente mejoras a la interfaz.

En cuanto a la calidad del proceso, el equipo no quedó conforme ya que por diferentes situaciones presentadas en el proceso no se pudo realizar de la forma que el equipo lo había planificado. Se verificaron y validaron correctamente todos los documentos, permitiendo alcanzar buenos niveles de calidad en los mismos. Se tuvo en cuenta en cualquier momento la evaluación de los riesgos detectados.

Por otro lado como resultado de las reuniones de retrospectiva que se realizaron durante la etapa de implementación las cuales pudieron ser corregidas en el siguiente sprint.

### 13.1.3 Tiempo

Durante el proyecto se realizó un control de cumplimiento de las tareas de forma. El equipo se tuvo que enfrentar a cambios importantes en su planificación como el que introdujo el evento de Ingeniería de muestra, una reevaluación del diseño de la interfaz, entre otros.

Estos hitos no permitieron al equipo cumplir con el calendario inicial del proyecto en tiempo y forma. Igualmente teniendo en cuenta que se logró un alcance mayor al planificado inicialmente, la gestión del tiempo puede evaluarse como positiva.

### 13.1.4 Esfuerzo

Al momento del cierre del proyecto el esfuerzo de los integrantes se encuentra por encima de la estimación del esfuerzo comprendido en la materia. Un factor importante a tener en cuenta es que el equipo estaba comprometido en realizar un producto que le fuera de utilidad a los integrantes de UMIC. Otro factor importante fue los requerimientos de

integración con los estándares de salud, que al ser estándares que se está trabajando para su implementación en el año 2018 hay aspectos que no están resueltos.

Teniendo en cuenta lo anterior se puede concluir que la gestión del esfuerzo no fue ni positiva ni negativa.

## 13.2 Conclusiones generales

Grado de satisfacción de integrantes de UMIC X: al finalizar el proyecto se realizó una encuesta de satisfacción del cliente, obteniendo el equipo X.

Equipo eficaz: durante el transcurso del proyecto se presentaron diversas dificultades en las distintas áreas del proyecto, las cuales fueron sorteadas eficazmente por parte de los miembros del equipo, permitiendo obtener un producto bueno a entender del equipo.

## 13.3 Lecciones aprendidas

Las lecciones aprendidas a lo largo del proyecto fueron las siguientes, se categorizaron en:

- Mejorar
  - Estimación de esfuerzo, fue uno de los aspectos que se mejoraron en el transcurso del proyecto. El equipo entiende que ninguno de los integrantes llegó con la experiencia previa necesaria para poder realizar una estimación acertada.
  - El lenguaje técnico usado por el equipo en diferentes reuniones principalmente en las primeras reuniones con los integrantes de UMIC. Este aspecto se trató en reuniones de retrospectiva y se fue corrigiendo en reuniones posteriores.
  - Aprovechar mejor las ventajas que el equipo entiende al usar SCRUM.
  - Realizar una investigación de poder contar con una herramienta que provea de forma gratuita la gestión del esfuerzo de los integrantes. El equipo realizó el uso de timecamp no teniendo en cuenta el periodo de tiempo en el que era gratuita.
  - Sprints más cortos. Por la experiencia adquirida en el proyecto por parte del equipo se cree que sería beneficioso realizar sprints de alrededor de 2 semanas de duración para contar con un feedback continuo por parte del cliente. Durante el transcurso del proyecto se decidió realizar sprint con mayor tiempo, principalmente porque todos los integrantes del equipo debían cumplir con sus obligaciones laborales.
- Buena adopción
  - La utilización de MockUps para validaciones de requerimientos es de gran ayuda. Al equipo le permitió mejorar la comunicación con los integrantes de UMIC. El equipo cree que se podría haber utilizado en mayor forma.
- No repetir

- No posponer tareas de testing. Tenerlas presentes a partir de un plan de calidad que involucre estas tareas desde el inicio del proyecto.

## 14 Trabajo futuro

- Soporte de multi-idioma, actualmente se encuentra realizado parte de la implementación. Se decidió de forma conjunta con el tutor del proyecto no incluir el requerimiento en el alcance final en función de tratarse de un requerimiento que no fue solicitado por los integrantes de la UMIC.
- Sincronización del calendario médico de SIMIC con el calendario manejado por registros médicos del hospital de clínicas. Durante el proyecto el equipo realizó la propuesta pero a raíz de la cantidad de trámites burocráticos que se debe realizar no se incluyó en el alcance final.
- Integración con el laboratorio del hospital, para poder obtener los resultados de exámenes en menor tiempo posible. Con esto los médicos podrían tener un análisis del estado de un paciente previo a su consulta.
- Soportar la asignación de más de un centro de salud para los médicos. Se logrará poder diferenciar el tiempo dedicado a la atención de pacientes por parte de cada centro de salud. Actualmente este tiempo es incluido en el tiempo de atención de UMIC.
- Gráficas normalizadas para la comparación de variables con diferentes definiciones.
- Incluir datos del régimen que debe de seguir el paciente en su alimentación en el plan realizado por los profesionales de nutrición integrantes de UMIC.
- Poder incluir variables que sean de tipo archivo. Con esto se podría implementar un visor de archivo DICOM. Logrando de esta forma poder ver electrocardiogramas ingresados médicos tratantes de UMIC que realizan la atención fuera de la unidad.
- Incluir una notificación para el médico en momento de la consulta que recuerde el cumpleaños del paciente. Con esto se podría lograr un acercamiento del médico con el paciente haciendo que este se sienta más cómodo.
- Simulación de evolución del paciente. Por medio de los datos con los que cuenta el sistema poder comparar con pacientes similares. También se podría utilizar redes neuronales, integrantes del equipo pudieron asistir a charlas dictadas en facultad de ingeniería en donde se veía su aplicación en el área de salud.
- Realizar análisis geográficos para obtener datos relevantes acerca de la población de pacientes que asiste a UMIC. Actualmente SIMIC registra datos con referencia geográficas las direcciones tanto de pacientes como de centros de salud. Un ejemplo podría ser obtener la cantidad de pacientes por barrios y a partir de esto poder mejorar la atención de los centros de salud de esos barrios.

## 15 Referencias

- [1] X. Q. Allue, A. Cyment, and Martín Alaimo, "What is Scrum," p. 10, 2013.
- [2] H. Kniberg, *Scrum and XP desde las trincheras*. 2007.
- [3] T. Ferguson and E. Gabarrón, "El libro blanco de los e-pacientes," p. 148, 2007.
- [4] AGESIC, "Guía de implementación estructura mínima nacional del documento clínico HL7 V3 CDA-R2 para uso en el dominio de Salud," p. 28, 2013.
- [5] P. Salud and H. CI, "Guía para la Identificación de Persona en el dominio de Salud."
- [6] N. Archivo, "Guía SNOMED CT," pp. 1–21, 2013.
- [7] "Derechos y Deberes-Usuarios Servicios Salud." .
- [8] C. H. C. E. N. Salud.uy, "Guía para la Gestión de OID," pp. 1–23, 2013.
- [9] G. Dra. Muracciole Bárbara; Dra. Rodríguez, Beatriz; Dra. Romero, "Recomendaciones para la protección de datos personales en el ámbito de la e-Salud," 2013.
- [10] "Umic," 2015. [Online]. Available: <http://www.unic.hc.edu.uy/index.php>. [Accessed: 01-Jan-2015].
- [11] "AGESIC," 2016. [Online]. Available: <http://www.agesic.gub.uy/>. [Accessed: 01-Jan-2016].
- [12] K. M. Oliveira, A. R. Rocha, and G. H. Travassos, "A Domain-Oriented Software Development Environment for Cardiology," no. i, p. 2.
- [13] T. Gandiya, A. Dua, G. King, T. Mazzocco, A. Hussain, and S. J. Leslie, "Self-Reported 'Communication Technology' Usage in Patients Attending a Cardiology Outpatient Clinic in a Remote Regional Hospital," *Telemed. e-Health*, vol. 18, no. 3, pp. 219–224, 2012.
- [14] Universia Mexico, "Crean en el IPN software para diagnosticar insuficiencia cardiaca," DF, México, 06-Sep-2011.
- [15] H. Wang, J. Zhang, J. Soar, X. Tao, and W. Huang, "Inclusive Society: Health and Wellbeing in the Community, and Care at Home: 11th International Conference on Smart Homes and Health Telematics, ICOST 2013, Singapore, June 19-21, 2013. Proceedings," J. Biswas, H. Kobayashi, L. Wong, B. Abdulrazak, and M. Mokhtari, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 322–330.
- [16] S. Agboola, K. Jethwani, K. Khateeb, S. Moore, and J. Kvedar, "Heart failure remote monitoring: Evidence from the retrospective evaluation of a real-world remote monitoring program," *J. Med. Internet Res.*, vol. 17, no. 4, p. e101, 2015.
- [17] R. G. Brindis, S. Fitzgerald, H. V. Anderson, R. E. Shaw, W. S. Weintraub, and J. F. Williams, "The American College of Cardiology-National Cardiovascular Data Registry(TM) (ACC-NCDR(TM)): building a national clinical data repository," *J. Am. Coll. Cardiol.*, vol. 37, no. 8, pp. 2240–2245, 2001.
- [18] "Summit Medical, Inc." [Online]. Available: <http://www.summitmedicalusa.com/>. [Accessed: 01-Jan-2015].
- [19] "Kareo." [Online]. Available: <http://www.kareo.com/>. [Accessed: 01-Jan-2015].
- [20] "Medi Touch." [Online]. Available: <http://www.meditouch.co.il/en>. [Accessed: 01-Jan-2015].
- [21] "AdvanceMD." [Online]. Available: <http://www.advancedmd.com>. [Accessed: 01-Jan-2015].
- [22] "NueMd." [Online]. Available: <http://www.nuemd.com/>. [Accessed: 01-Jan-2015].
- [23] "iSalus Healthcare." [Online]. Available: <http://isalushealthcare.com/>. [Accessed: 01-Jan-2015].
- [24] "Epi Info." [Online]. Available: <http://wwwn.cdc.gov/epiInfo>. [Accessed: 01-Jan-2015].
- [25] "About Google." [Online]. Available: [https://www.google.com/intl/en\\_US/about/](https://www.google.com/intl/en_US/about/). [Accessed: 01-Jan-2016].
- [26] "Guías Material Design." [Online]. Available: agregar referencia a <https://www.google.com/design/spec/material-design/introduction.html#introduction-goals>. [Accessed: 01-Jan-2015].

- [27] "Ley Xi - N° 18.335." .
- [28] "Ley Xi - N° 18.331." .
- [29] M. Mikowski and J. Powell, "Single Page Web Applications," p. 432, 2013.
- [30] V. Dobjanschi, "Developing Android REST client applications," *Google I/O 2010*, 2010.
- [31] "HTTP Specifications and Drafts." [Online]. Available: <https://www.w3.org/Protocols/Specs.html>. [Accessed: 01-Jan-2015].
- [32] R. Fielding, "REST API," 2015. [Online]. Available: <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>.
- [33] "Curso AngularJS," 2015. [Online]. Available: <http://cursoangularjs.es/doku.php>.
- [34] A. Shalloway and J. Trott, "Design Patterns – Elements of Reusable Object-Oriented Software," *A New Perspect. Object-Oriented Des.*, p. 334, 2002.
- [35] "Java Script Comparision." [Online]. Available: <https://www.airpair.com/js/javascript-framework-comparison>. [Accessed: 01-Jan-2015].
- [36] "En Perspectiva, Ingenieria de Muestra 2015," 2015.
- [37] "Guide Git," 2015. [Online]. Available: <https://git-scm.com/docs/gittutorial>. [Accessed: 01-Jan-2015].
- [38] "Curso de AngularJS by Google," 2015. [Online]. Available: <http://campus.codeschool.com/courses/shaping-up-with-angular-js/intro>. [Accessed: 01-Jan-2015].

## 16 Anexos

Los anexos se encuentran como archivos independientes dentro de la carpeta Anexo entregada junto al documento principal.

16.1 Anexo 1: Glosario

16.2 Anexo 2: Planificación y contrato.

16.3 Anexo 2: Investigación de SCRUM.

16.4 Anexo 4: Modelo de datos

16.5 Anexo 5: Resumen de seguridad.

16.6 Anexo 6: Ingeniería de muestra.

16.7 Anexo 7: Folleto de ingeniería de muestra.

16.8 Anexo 8: Encuesta para conocer el uso de la tecnología por parte de pacientes.

16.9 Anexo 9: Encuesta de satisfacción del cliente.

16.10 Anexo 10: Bitácora de reuniones.

16.11 Anexo 11: Horas.