

SIMIC

Sistema Interdisciplinario para el Manejo de la Insuficiencia Cardíaca

Rodrigo Gonzalez, Alejandro Cardone y Viterbo García

Tutor: Prof. Ing. Franco Simini

Co-Tutor: Prof. Dra. Gabriela Ormaechea y Prof. Dr. Pablo Álvarez

Ayudante: Dra. Gabriela Silvera

Agenda

— — —

- Motivación
- Objetivos
- Proceso
- Pruebas
- DEMO
- Gestión
- Trabajo futuro
- Conclusiones

Motivación

— — —

- UMIC (Unidad Multidisciplinaria de Insuficiencia Cardíaca)
- Atención de pacientes portadores de Insuficiencia Cardíaca
- NIB (Núcleo de Ingeniería Biomédica)
- Equipo multidisciplinario de trabajo interdisciplinario
- Hospital de Clínicas

Objetivos

— — —

- Principal: “pacientes más sanos y más felices” T. Ferguson and E. Gabarrón, “El libro blanco de los e-pacientes”
- Generar herramienta
 - Útil para la UMIC, logrando una satisfacción de sus integrantes
 - Perdure en el tiempo
- Desafíos personales del equipo estudiantes
 - Área desconocida
 - Tecnológico

Problemas principales

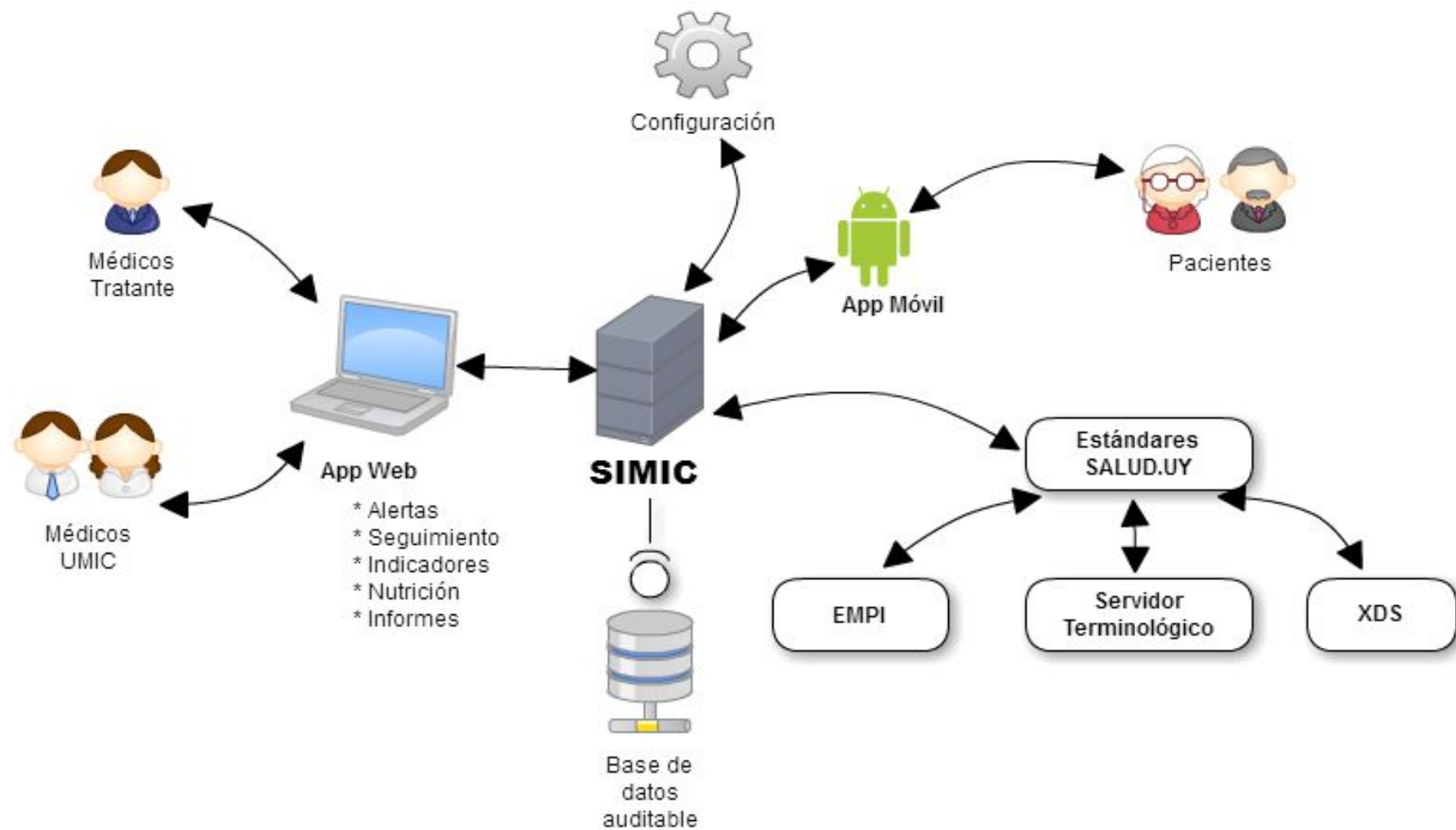
— — —

- Seguimiento de pacientes
- Datos
 - Disponibilidad
 - Integridad y fidelidad
 - Dificultad en el mantenimiento y pérdida de información de asistencias a centros externos
- Alertas
- Carencia de indicadores de gestión en la atención en tiempo real

Propuesta

— — —

- Gestión
 - Historia clínica
- Alertas configurables por el médico
- Indicadores en tiempo real
- Aplicación móvil para paciente
- Estándares Salud.uy
 - Cross Enterprise Document Sharing (XDS) - Clinical Document Architecture (CDA)
 - Índice maestro de personas (EMPI)
 - Servidor Terminológico (SNOMED)

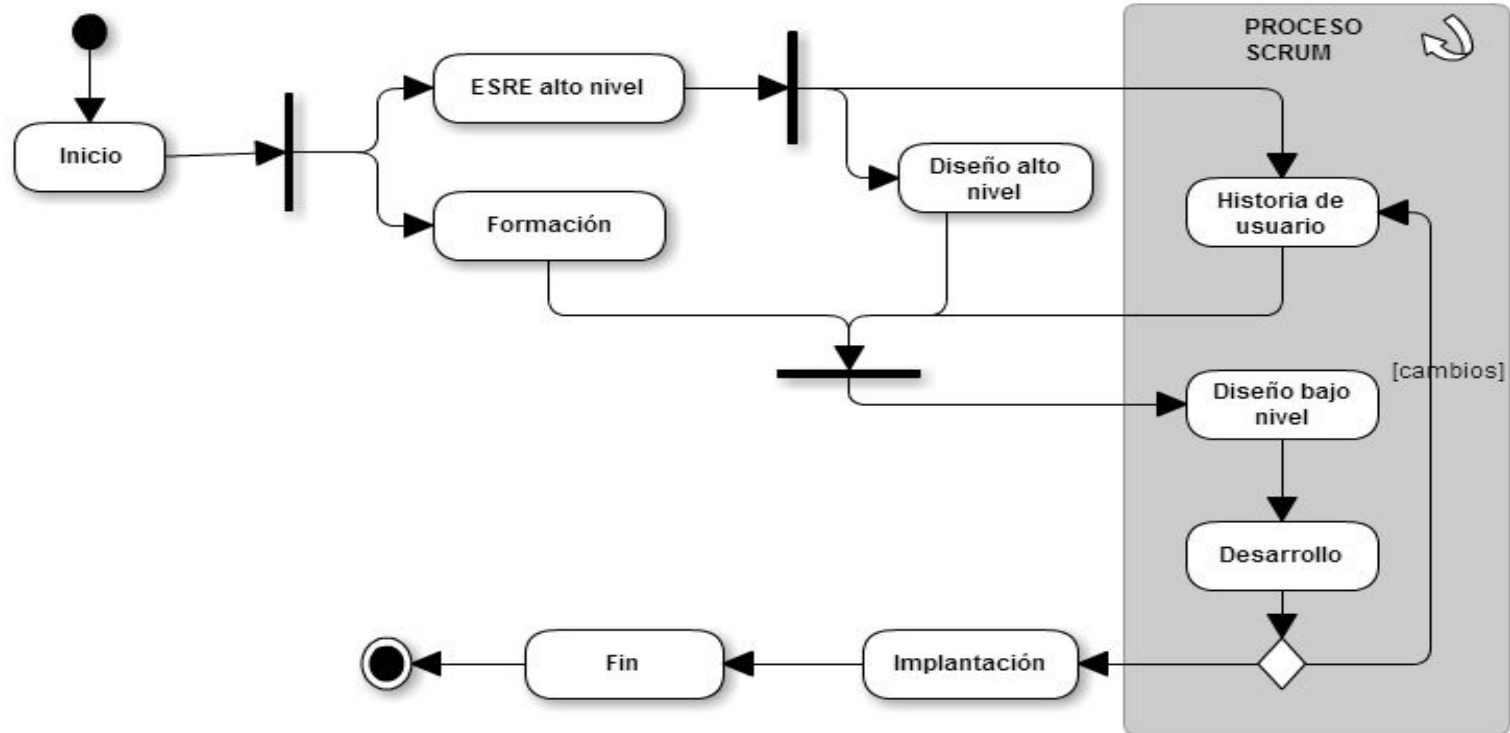


Proceso

— — —

- Elección de metodología
 - Híbrida: tradicional y SCRUM (principalmente en la etapa de desarrollo)
- Características
 - Requerimientos cambiantes: área desconocida y dificultad en la comunicación entre el profesional médico y informáticos
 - Resultados a corto plazo
 - Gestión regular de las expectativas: demostraciones y feedback temprano
 - Proceso de gestión ligero

Ciclo de vida del proyecto



* ESRE: especificación de requerimientos

Ingeniería de requerimientos

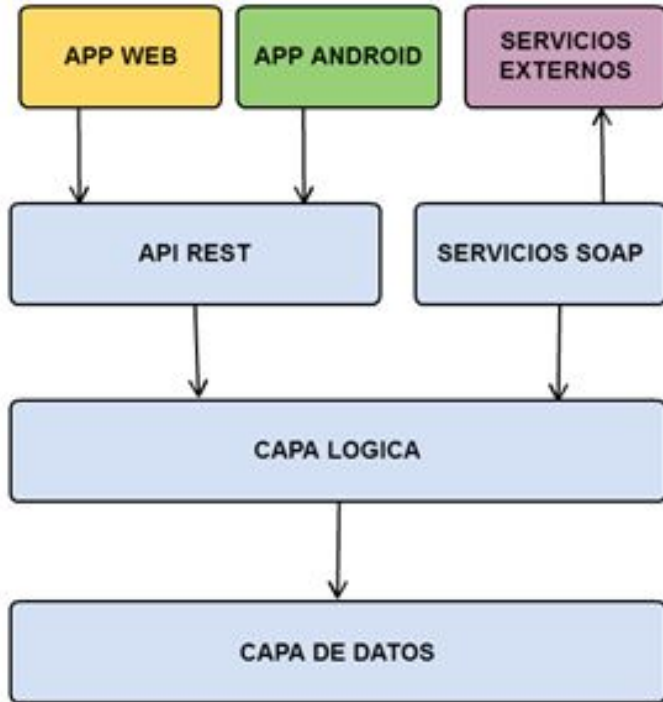
- Investigación en el área
 - Problemas de interacción con los profesionales de la salud y computación
 - Leyes
- Reuniones con responsables de la UMIC
 - Historias de usuarios
- Uso de mockups
 - Mejor interacción
- Generación de documentación

Requerimientos no funcionales

- Seguridad
 - Registro de acciones por parte del usuario
 - Seguridad básica.
- Confidencialidad
 - Diferentes perfiles y control de acceso a los datos
- Integridad
 - Todos los datos utilizados en SIMIC deben ser fidedignos.
- Usabilidad
 - Interfaz: intuitiva y fácil de usar
- Integración
 - Integración con los servicios proporcionados de la iniciativa Programa Salud.uy
- Legislación
 - Cumplir con la legislación actual.

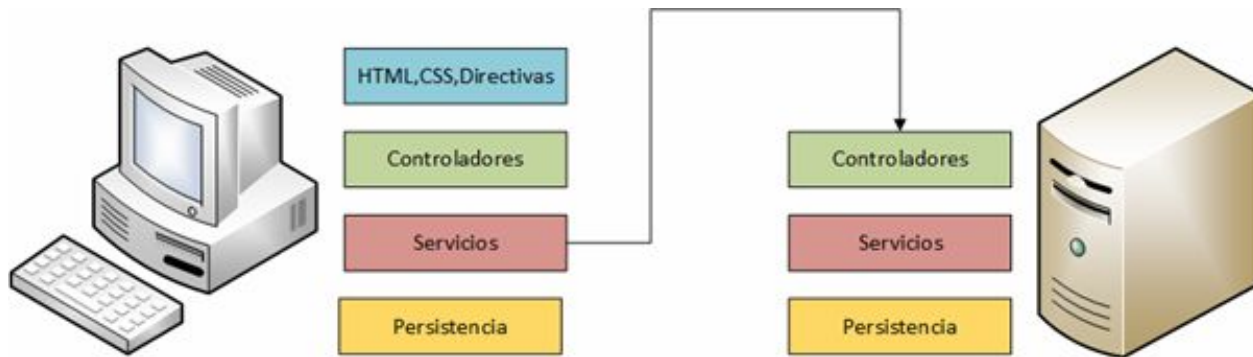
Arquitectura

— — —



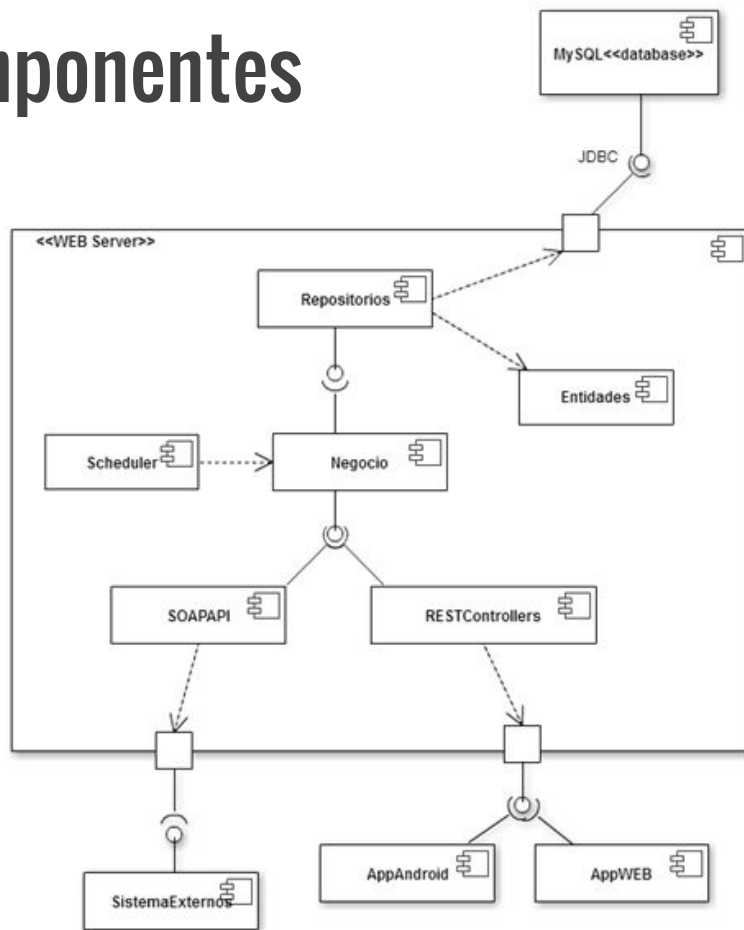
- App web para uso del médico
- App móvil para uso del paciente
- Arquitectura basada en recursos
- REST como interfaz para la comunicación (app web y móvil)
- Interoperabilidad con otros sistemas (basada en SOAP)

Aplicación Web

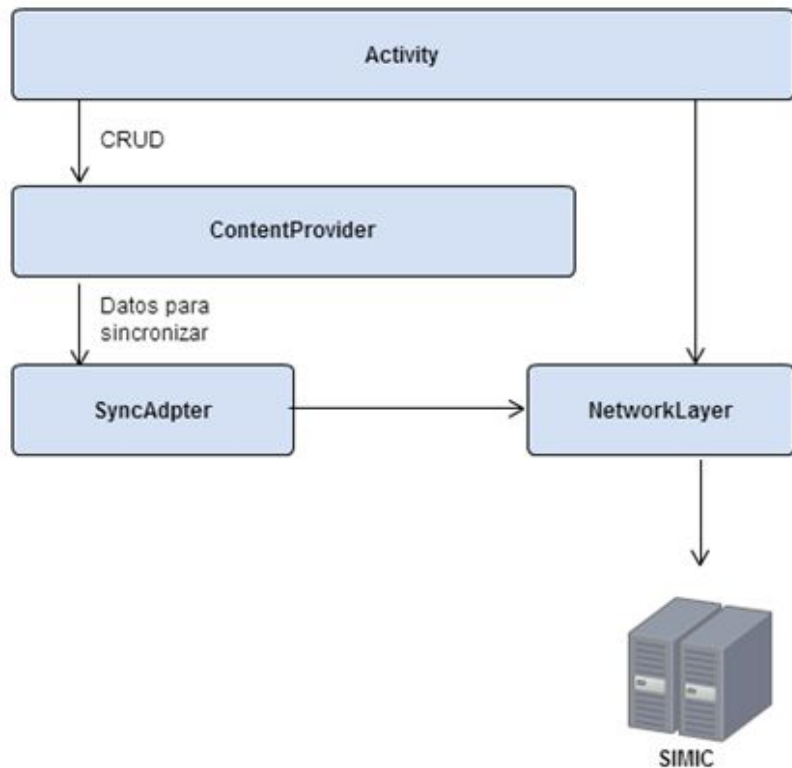


- Arquitectura en capas
- Aplicación web SPA (basada en AngularJS)
- Tanto de lado del cliente como del servidor encontramos
 - Controladores
 - Servicios
 - Persistencia

Diagrama de componentes

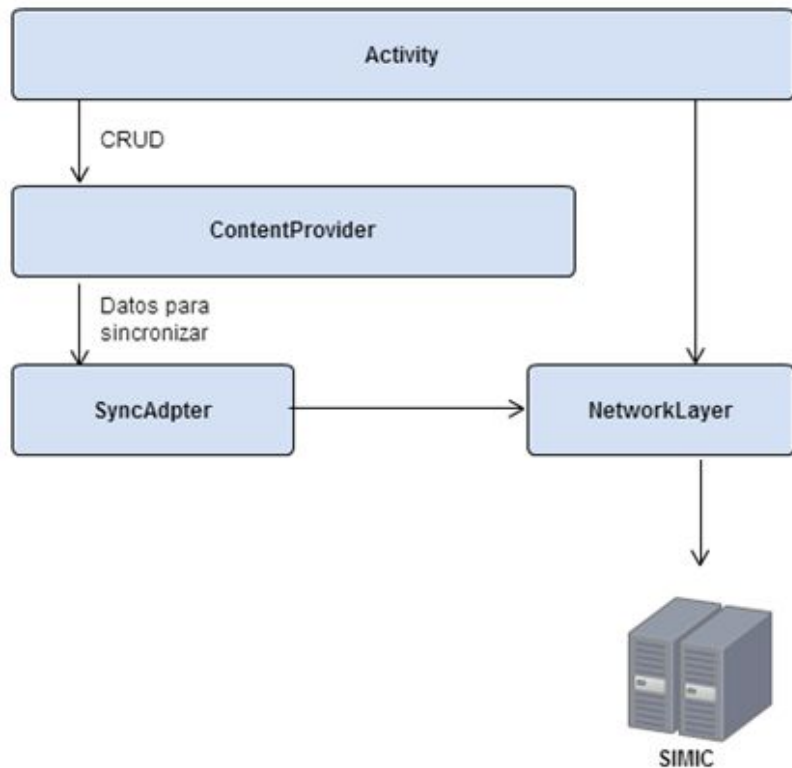


Aplicación móvil



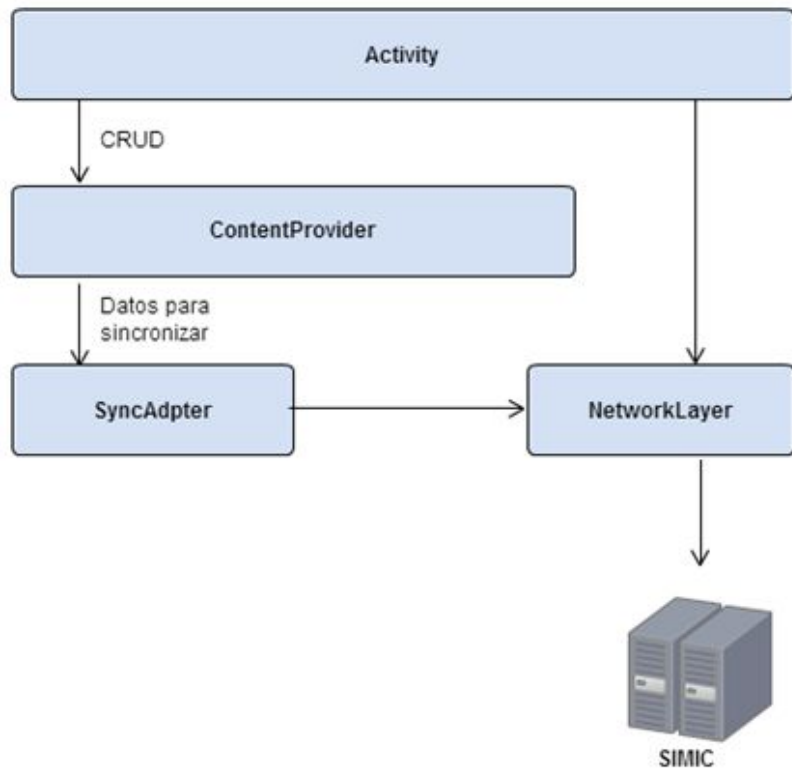
- **ContentProvider:**
 - Corazón de los DAOs
 - Encargado de proveer el contenido, llama al SyncAdapter cuando hay un dato que necesita ser sincronizado

Aplicación móvil



- **SyncAdapter:**
 - Es un plug-in que nos brinda Android
 - Maneja la ejecución en una cola de ejecución, se ejecuta en segundo plano

Aplicación móvil



- **NetworkLayer:**

- Encargada de realizar la comunicación con el servidor
- Será invocada por el sync adapter a excepción de los datos que son consumidos directamente del servidor

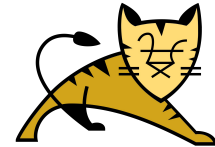
Otras decisiones de diseño

- Separación de datos privados (rigiéndose por la normativa actual)
- Distintos tipos de variables (calculadas, numéricas, con opciones)
- Seguridad básica (<https>, usuario contraseña, login a través de cookie)
- Base de datos auditable (tener registro de quién y cuándo hizo cada acción)
- Integración de estándares
 - Envío de CDA (integración con el repositorio XDS)
 - Servidor terminológico (por eje: para parentesco solución en caché y medicamentos a demanda)
 - EMPI (registro único de personas al crear paciente)
- Diccionario predictor personalizado por nutricionista (predicción con aprendizaje)

Tecnologías backend

— — —

- JAVA (Spring Framework)
- Hibernate
- Maven
- Jackson
- Imgscalr
- Jasperreports
- Log4j
- JavaMail
- MySQL



Tecnologías frontend

— — —

- AngularJS
- NodeJS
- Grunt
- Bower
- Twitter Bootstrap
- UI Bootstrap
- Google Maps
- ngStorage
- Moment js
- ...



Aplicación móvil Android

— — —

- `Android.support.appcompat`
- `Android.support.design`
- Butterknife
- Picasso
- Retrofit
- Gson converter
- `sqlite-provider`



Calidad y migración de datos

— — —

- **Testing funcional**
 - Testing funcional basado en las historias de usuario
 - Testing Cruzado a la finalización del proyecto
- **Testing de carga**
 - Utilizando la herramienta JMETER
 - (1307ms inicio de sesion, 593ms buscar paciente, 1768ms realizar una acción)
- **Testing de Seguridad**
 - Se utilizó la herramienta ZAP de OWASP
- **Migración de datos**
 - Decisiones en datos con errores
 - 940 pacientes, 16678 controles, 86806 valores, 41795 dosis, etc..
- **Ingeniería de muestra**
 - Comentarios positivos, motivación para el equipo
 - Participación en el programa radial “En perspectiva”

99 %

— — —

De los bugs encontrados durante la etapa de testing fueron corregidos

DEMO

— — —

Gestión

— — —

- Planificación
 - Etapa de análisis y formación
 - Historias de usuario
 - Desarrollo
 - Sprint
 - Testing
 - Test de carga
 - Test de seguridad
 - Migracion
 - Implantacion

Gestión de configuración

— — —



Forms



timecamp

- Planificación
 - Plug-in para SCRUM
 - Métrica de velocidad
- Documentación
 - Trabajar de forma concurrente
- Gestión de horas
 - Mala experiencia
- Gestión de código
 - Versionado local
 - Experiencia previa sin caídas

Gestión de alcance

— — —

- Planificación

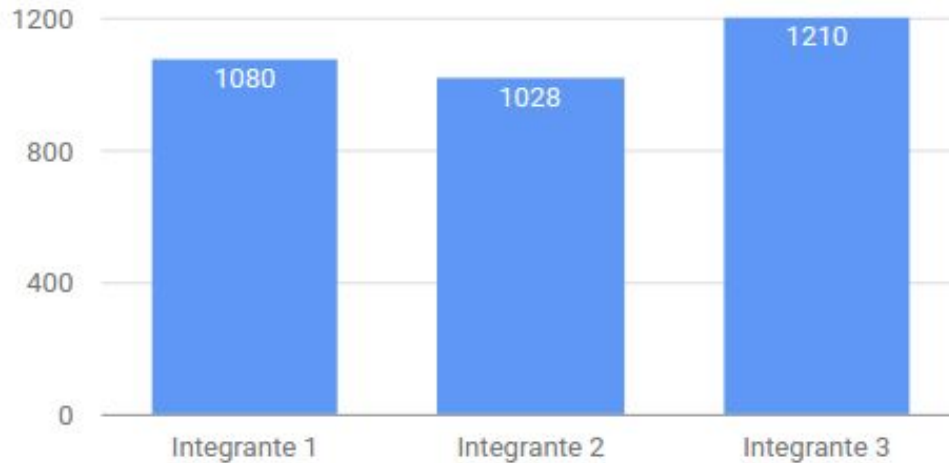
- Requerimientos cambiantes
- Sprints en la etapa de desarrollo
- Errores de estimación. Mejoras con el avance del tiempo

- Modificación

- Recomendaciones de tutor
- Problemas requerimientos de integración con estándares SALUD.UY
- Flexible
 - Motivo: Llegar a una solución que ayude a la UMIC

Gestión de esfuerzo

— — —



- Buen relacionamiento del equipo
- Equilibrio en la asignación de tareas

Gestión de esfuerzo

— — —

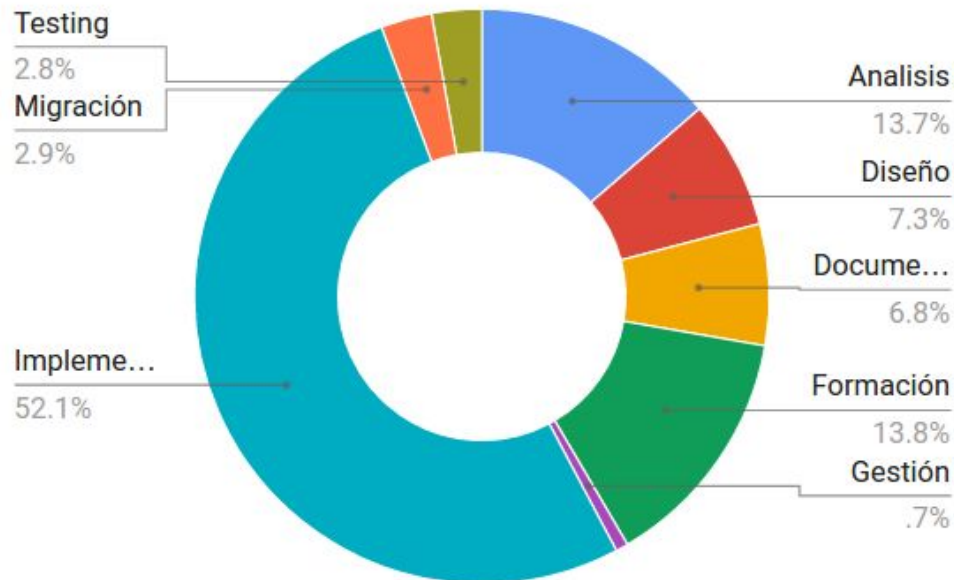


- Identificación de las etapas
- Presentación en Ing. de muestra
- Migración

Gestión de esfuerzo

— — —

- Gran cantidad en desarrollo
- Formación
 - Aprendizaje de una nueva tecnología y refuerzo de otras
- Migración
 - Principalmente en el análisis y correcciones de los datos



Trabajo futuro

— — —

- Soporte multilinguaje
- Sincronización del calendario médico de SIMIC con el calendario de registros médicos del hospital
- Integración con el laboratorio del hospital
- Gráficas normalizadas para la comparación de diferentes variables

Trabajo futuro

— — —

- Incluir archivos adjuntos al resultado de un examen
- Notificación cumpleaños del paciente
- Simulación de evolución del paciente
 - Identificación de patrones mediante uso de redes neuronales
- Análisis geográficos

Conclusiones

- Satisfacción de integrantes de UMIC
 - Encuestas a los responsables de UMIC
 - Presentación de SIMIC a la UMIC el 3/05/2015
- Desempeño del equipo
- Lecciones aprendidas
 - SCRUM, sprints más corto (se fue corrigiendo)
 - Conocimiento en el área de salud



En especial a nuestras familias.