



Sistema de notificación de incidencias de analizadores para dispositivos móviles

Nombre Estudiante: Rubén Carmona Zurita

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Consultor: Eduard Martín Lineros

Profesor responsable de la asignatura: Carles Garrigues Olivella

Fecha de Entrega: 08/06/2016

© Rubén Carmona Zurita

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

*A Rocío, sin tu apoyo y
comprensión no conseguiría
ninguna de las metas que nos
hemos propuesto.*

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Sistema de notificación de incidencias de analizadores para dispositivos móviles</i>
Nombre del autor:	<i>Rubén Carmona Zurita</i>
Nombre del consultor/a:	<i>Eduard Martín Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2016
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave:	<i>Analizadores, alarmas y Push.</i>
Resumen del Trabajo:	
<p>Durante el transcurso de su trabajo un facultativo debe realizar consultas sobre el estado de los analizadores que dispone en su laboratorio: análisis que se están realizando, incidencias o alarmas. Para realizar dichas consultas el facultativo debe de desplazarse físicamente al analizador y consultar la información deseada ya que actualmente no hay posibilidad de notificar de manera remota cualquier incidencia o alarma del analizador.</p> <p>Este proyecto pretende cubrir la necesidad de los facultativos de conocer en tiempo real el estado de los analizadores de su laboratorio de manera remota. De esta manera se pretende facilitar y agilizar su trabajo evitando desplazamientos innecesarios y el sobrecoste de no atender a una incidencia o alarma en el menor tiempo posible.</p> <p>Para dar respuesta a estas necesidades este proyecto aporta:</p> <ul style="list-style-type: none">• Sistema de notificaciones para las incidencias o alarmas que se puedan producir en los analizadores.• Consulta del estado de los analizadores del laboratorio en tiempo real.• Consulta del trabajo realizado en los analizadores en tiempo real.• Acceso a modificar configuraciones, no intrusivas ni críticas, de los analizadores través de la propia aplicación.	

Abstract:

During the course of his work, a physician should check the state of the analyzers that has in his laboratory: analysis being performed, incidences or alarms. To perform such queries, the physician must physically go to the analyzer and check the desired information because now is not possible to perform a remote notification of an analyzer`s incident or alarm.

This project aims to meet the need for physicians to know in real time the status of their laboratory analyzers remotely. In this way is intended to facilitate and accelerate their work avoiding unnecessary displacements and the extra cost of not responding to an incident or alarm in the shortest possible time.

To deal with these needs this project provides:

- System notifications for incidents or alarms that may occur in the analyzers.
- Check laboratory analyzers status in real time.
- Check analyzers' work in real-time.
- Access to modify configurations, not intrusive or critical, of the analyzers through the app.

Índice

1.	Introducción	1
1.1	Contexto y justificación del Trabajo	1
1.2	Objetivos del Trabajo	3
1.2.1	Objetivos generales	3
1.2.2	Objetivos específicos	3
1.3	Enfoque y método seguido	4
1.4	Planificación del Trabajo	5
1.4.1	PEC 1: Elaboración del Plan de Proyecto (<i>Sprint 0</i>)	5
1.4.2	PEC 2: Análisis, diseño y prototipo de la aplicación (<i>Sprint 1</i>)	6
1.4.3	PEC 3: Implementación de las funcionalidades definidas (<i>Sprint 2</i>)	6
1.4.4	Entrega final (<i>Sprint 3</i>)	7
1.4.5	Planificación Temporal	7
1.5	Breve resumen de productos obtenidos	9
1.6	Breve descripción de los otros capítulos de la memoria	9
1.7	Herramientas de desarrollo	11
1.7.1	Hardware	11
1.7.2	Software	12
2.	Análisis	13
2.1.	Usuarios y Contexto de Uso	13
2.2.	Diseño Conceptual	15
2.2.1.	Escenarios de uso	15
2.2.2.	Diagramas uso	16
2.3.	Diseño Arquitectura	25
2.3.1.	Diagrama de Base de Datos	25
2.3.2.	Diagrama UML de Clases	26
2.3.3.	Arquitectura del Sistema	26
2.4.	Diseño de la persistencia	27
2.4.1.	Persistencia en la parte servidor	28
2.4.2.	Persistencia en la aplicación móvil	28
3.	Prototipado	29
3.1	Prototipo Simulador Analizador	29
3.2	Prototipo Aplicación Móvil	30
3.3	Prototipo Aplicación Móvil (alta definición)	31
3.4	Diseño gráfico e interfaces	32
1.1.1.	Action Bar	32
1.1.2.	Drawer Menu	32
1.1.3.	Detalle Analizador	33
1.1.4.	Detalle Worklist	33
1.1.5.	Detalle Pruebas	34
1.1.6.	Detalle de las notificaciones	34
1.1.7.	Configuración de notificaciones	35
1.1.8.	Información mostrada en una notificación	35

4.	Evaluación	36
5.	Conclusiones y líneas de futuro	37
5.1	Conclusiones.....	37
5.2	Líneas de futuro.....	37
6.	Glosario	39
7.	Bibliografía y referencias.....	40
8.	Anexo 1 - Entrevistas.....	41
9.	Anexo 2 – Guía de instalación y uso del sistema	43
9.1	Instalación del Servidor de Aplicaciones y Base de datos.....	43
9.1.1	Requisitos.....	43
9.1.2	Instalación Sistema Operativo Linux.....	44
9.1.3	Instalación Software	46
9.2	Uso del Simulador de Analizadores	53
9.2.1	Pantalla Principal	53
9.3	Uso de la Aplicación Móvil	57
9.3.1	Login.....	58
9.3.2	Lista de Analizadores	58
9.3.3	Lista de Trabajo	59
9.3.4	Lista de Trabajo – Completa	59
9.3.5	Configuración de pruebas del Analizador.....	60
9.3.6	Notificaciones del Analizador	60
9.3.7	Menú de opciones	61
9.3.8	Configuración de Notificaciones	61
9.3.9	Notificación de un Evento.....	62

Lista de figuras

Ilustración 1 – Representación gráfica de la propuesta	2
Ilustración 2 – Diagrama de Gantt Sprint 0: Plan de Trabajo	8
Ilustración 3 – Diagrama de Gantt Sprint 1: Diseño	8
Ilustración 4 – Diagrama de Gantt Sprint 2: Implementación	8
Ilustración 5 – Diagrama de Gantt Sprint 3: Entrega Final	9
Ilustración 6 – Casos de Uso: Aplicación Móvil	16
Ilustración 7 – Casos de Uso: Simulador de Analizadores	24
Ilustración 8 – Diagrama de Base de Datos	25
Ilustración 9 – Diagrama UML de Clases	26
Ilustración 10 – Arquitectura del Sistema	27
Ilustración 11 – Prototipo Simulador de Analizadores	29
Ilustración 12 – Prototipo Aplicación Móvil	30
Ilustración 13 – Prototipado Aplicación Móvil (alta definición)	31
Ilustración 17 – VirtualBox: Configuración máquina virtual	44
Ilustración 18 – VirtualBox: Creación disco duro	44
Ilustración 19 – VirtualBox: Configuración de las tarjetas de red	45
Ilustración 20 – VirtualBox: Configuración aceleración 3D	45
Ilustración 21 – VirtualBox: Selección imagen ISO	45
Ilustración 22 – Aplicación Simulador Analizadores (AnSim)	53
Ilustración 23 – AnSim: Detalle cabecera	54
Ilustración 24 – AnSim: Pestaña Worklist	54
Ilustración 25 – AnSim: Pestaña Worklist iniciar lista de trabajo	55
Ilustración 26 – AnSim: Pestaña Notificaciones	55
Ilustración 27 – AnSim Pestaña de configuración de pruebas	56
Ilustración 28 – App Móvil: Login	58
Ilustración 29 – App Móvil: Lista de analizadores	58
Ilustración 30 – App Móvil: Lista de trabajo	59
Ilustración 31 – App Móvil: Lista de trabajo – Completa	59
Ilustración 32 – App Móvil: Pruebas configuradas	60
Ilustración 33 – App Móvil: Notificaciones del Analizador	60
Ilustración 34 – App Móvil: Menú de opciones	61
Ilustración 35 – App Móvil: Configuración de notificaciones	61
Ilustración 36– App Móvil: Notificaciones en la pantalla principal del móvil	62

1. Introducción

1.1 Contexto y justificación del Trabajo

En el trabajo diario de un Laboratorio Clínico se realizan cientos de pruebas sobre muestras de pacientes a través de equipos Analizadores. Estos equipos son los encargados de realizar análisis y dar resultados sobre las muestras (sangre, orina o heces) de manera que los facultativos puedan determinar una enfermedad, una infección o un resultado crítico que requiera de una intervención inmediata al paciente.

El trabajo realizado por estos analizadores es crítico dentro de un laboratorio ya que el incorrecto funcionamiento o parada puede acarrear severos problemas tanto para los pacientes como para el propio laboratorio. La ejecución de pruebas es constante y según el volumen de trabajo del laboratorio puede requerir que el analizador trabaje 24/7.

En complemento a los trabajos diarios, es obligatorio que los equipos realicen “Controles de Calidad” que permitan certificar, mediante muestras estabilizadas, que los resultados obtenidos sobre las diferentes muestras sean correctos y no den falsos o erróneos resultados.

Los analizadores realizan dichas pruebas mediante reactivos que permiten evaluar las muestras de los pacientes. Estos analizadores pueden trabajar de manera autónoma (conectados a la red del hospital) y/o conectados entre sí a través de lo que se conoce como una “cadena de analizadores”; un sistema de gestión de tubos que permite tener conectados distintos analizadores a través de unas guías por las que se transportan los tubos de manera que se permita dividir el trabajo y las pruebas para poder trabajar en paralelo y poder gestionar un volumen más elevado de pruebas a realizar.

La monitorización y la gestión de dichos analizadores se realiza in situ donde estos estén ubicados dentro del laboratorio a través de un software propietario; esto implica que los facultativos, técnicos del laboratorio, etc. tengan que desplazarse físicamente donde se encuentre el analizador (o la cadena de analizadores) y consultar:

- la lista de trabajo o pruebas que está realizando el analizador en un determinado momento y que tiene planificadas.
- las pruebas que están próximas a finalizar o han finalizado (el tiempo puede variar de 2, 5 a 30 minutos),

- realizar una modificación sobre la lista de trabajo del analizador para poder adelantar la ejecución de pruebas con mayor prioridad, los conocidos como pruebas críticas,
- si el nivel de reactivos es suficiente o está próximo a acabarse, lo que impediría la realización de determinadas pruebas,
- el estado del analizador por si ha tenido algún problema o alarma que requiera de una intervención.

Estos tres últimos puntos son considerados de los más importantes ya que implica un desplazamiento del facultativo para comprobar el estado del analizador.

Actualmente el facultativo utiliza un software propietario de gestión y consulta de resultados y peticiones instalado en un ordenador al lado del analizador. Este software permite consultar y configurar el analizador para realizar los trabajos a realizar y utilizando un middleware, también propietario, enviar los resultados al LIS del laboratorio.

La solución que se quiere aportar con este producto es la posibilidad de que el facultativo pueda consultar a través de su dispositivo móvil el estado de los analizadores para poder consultar:

- el estado del analizador: incidencias, desconexiones, estado de los reactivos, etc.,
- consultar la lista de trabajo del analizador y conocer el trabajo realizado y el pendiente de realizar,
- ser notificados de cualquier incidencia de ocurrida con el analizador a través de mensajes o notificaciones.

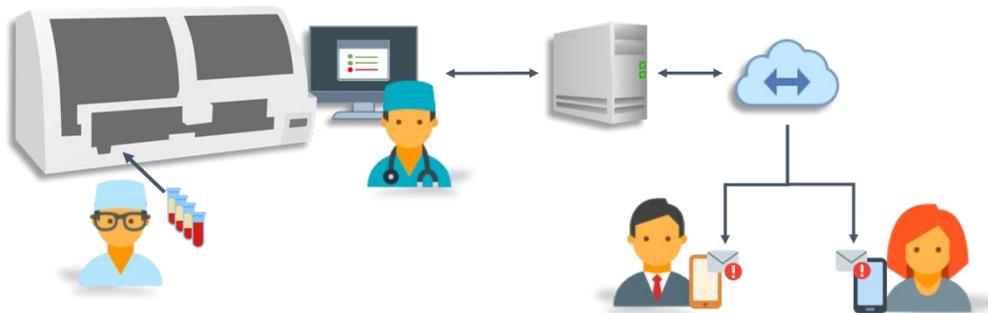


Ilustración 1 – Representación gráfica de la propuesta

1.2 Objetivos del Trabajo

El objetivo principal de este Trabajo Final de Master (TFM) es desarrollar una infraestructura de servidor de notificaciones *Push* que permita la notificación de información relacionada con los analizadores en tiempo real. De esta manera se pretende que los facultativos puedan ser notificados de cualquier incidencia que requiera su atención inmediata.

Al desarrollarse este proyecto en el entorno de una empresa este ha de integrarse con el servidor de aplicaciones, base de datos y el entorno de seguridad y gestión de usuarios ya disponible.

1.2.1 Objetivos generales

- a) Poner en práctica todos los conocimientos adquiridos a lo largo de la titulación.
- b) Realizar un aprendizaje sobre el funcionamiento de los servidores de notificaciones mediante *Tecnología Push* utilizadas para el envío de notificaciones a los dispositivos móviles. Estudiar diferentes alternativas existentes para implementar esta tecnología.
- c) Realizar un estudio sobre qué alarmas e incidencias, que actualmente los analizadores generan, deben de ser notificadas a los usuarios. Este estudio tendrá en cuenta el protocolo de comunicaciones actual implementado para la comunicación con los analizadores.
- d) Realizar una valoración sobre cuáles interacciones sobre los analizadores son interesantes de poder ser realizadas a distancia y que no requieran de la presencia física del facultativo.
- e) Valorar qué notificaciones se deberían de tener en cuenta para futuras versiones de la aplicación tomando como base las necesidades de los usuarios y los nuevos protocolos de comunicaciones que están siendo desarrollados en la actualidad para los analizadores.

1.2.2 Objetivos específicos

- a) Implementar un servidor de notificaciones *Push* para dispositivos móviles que permita realizar notificaciones de eventos que requieran la atención inmediata del facultativo. Este servidor se ha de integrar de manera transparente al sistema de aplicaciones que gestiona la información del laboratorio.
- b) Desarrollar un simulador de analizadores con el que realizar las interacciones básicas para suplir la carencia de un analizador real para las pruebas y evaluaciones del producto.

- c) Desarrollar una aplicación móvil para Android que permita a los usuarios:
- Definirse un perfil que le dé acceso a consultar y recibir notificaciones de los analizadores. Este perfil se integrará con el entorno de seguridad y gestión de usuarios ya disponible en la empresa.
 - Recibir notificaciones de alarmas/incidencias de los analizadores.
 - Consultar el detalle de la notificación.
 - Disponer del estado y los datos reales de los analizadores.
- d) Realizar un diseño técnico de la aplicación abordando la definición de casos de uso y el diseño de la arquitectura de la aplicación.
- e) Obtener una aplicación intuitiva, sencilla y eficiente en la gestión de las notificaciones de incidencias.

1.3 Enfoque y método seguido

Este proyecto tiene los requisitos lo suficientemente definidos para implicar poca incertidumbre a la hora de su desarrollo. Estando además los principales hitos del proyecto definidos por la propia Evaluación Continua (EC), que permite llevar a cabo una evaluación y revisión constante por parte del consultor, obteniendo por ello unas sugerencias que supongan una mejor calidad del producto final.

De esta manera tendremos unos hitos finales marcados por fechas, pudiéndose desglosar estos en sub-tareas (algunas independientes entre sí) para poder abarcar todo el conjunto de manera más eficiente y efectiva.

Dentro de todas las metodologías disponibles se escoge la utilización del desarrollo en *Scrum*, una metodología que pertenece al grupo conocido como *Metodologías Ágiles*, y se ha decidido por esta metodología en detrimento de otras porque:

- permite desarrollar el proyecto de manera incremental;
- permite definir una lista de requisitos priorizada (*Product Backlog*) con fechas iniciales y finales;
- permite realizar tareas no relacionadas simultáneamente;
- el hecho de definir entregas o *Sprints* con fechas establecidas casa perfectamente con los hitos del proyecto.

Con este tipo de metodología se pretende abordar los distintos hitos y requisitos, aprovechando que se pueden diferenciar claramente las tareas a realizar (y a su vez las sub-tareas), teniéndolas de esta manera más controladas y mejor planificadas (en tiempo y coste); incluso aquellas en las que se pueden trabajar en paralelo al ser independientes entre sí.

Otro tipo de metodologías ágiles que se evaluó fue la también metodología ágil *Kanban*. Esta metodología se desestimó por la naturaleza del proyecto y a la necesidad de cumplir con tiempos fijados que no se tienen en cuenta en *Kanban*.

1.4 Planificación del Trabajo

La planificación del proyecto está marcada por las distintas entregas o hitos de la propia EC con lo que podemos desglosar el proyecto en:

PEC 1	Elaboración del Plan de Proyecto.	Sprint 0
PEC 2	Análisis, diseño y prototipo de la aplicación.	Sprint 1
PEC 3	Implementación de las funcionalidades definidas.	Sprint 2
Entrega final	Entrega de la memoria y presentación del trabajo.	Sprint 3

En consecuencia de la metodología ágil escogida, estos hitos se corresponden con *Sprints* a realizar, iniciando el *Sprint 0* en el hito de la *Elaboración del Plan de Proyecto* al ser contemporánea a la elección de la metodología seleccionada.

Una descripción más detallada del contenido de cada uno de estos hitos es:

1.4.1 PEC 1: Elaboración del Plan de Proyecto (*Sprint 0*)

Este *Sprint* tiene como objetivo definir el marco general del proyecto en base a los objetivos, decisiones globales, el alcance y la planificación específica de las actividades del mismo.

Actividad	Descripción
Investigación	<ul style="list-style-type: none"> Búsqueda de una necesidad a cubrir. Comparación y selección de la tecnología a utilizar para resolver la necesidad.
Definir Objetivo y alcance	<ul style="list-style-type: none"> Listado de los objetivos del proyecto. Establecer el alcance del proyecto.
Planificación	<ul style="list-style-type: none"> Temporización del proyecto conforme a la metodología seleccionada y a las entregas definidas por los hitos de la EC.
Preparación entornos de Trabajo	<ul style="list-style-type: none"> Instalación de Android Studio. Instalación de Eclipse. Instalación <i>MySQL Workbench</i>.
Revisión y entrega PEC 1	<ul style="list-style-type: none"> Revisión y entrega de la documentación realizada.

1.4.2 PEC 2: Análisis, diseño y prototipo de la aplicación (Sprint 1)

Este Sprint tiene como objetivo definir en detalle los requisitos así también el cómo se tienen que cubrir estos y las necesidades planteadas en el punto anterior.

Actividad	Descripción
Análisis de Requisitos	<ul style="list-style-type: none">• Especificación de requisitos funcionales.• Especificación de requisitos no funcionales.
Investigación	<ul style="list-style-type: none">• Evaluación, selección y aprendizaje de la tecnología servidor a utilizar para las notificaciones <i>Push</i>.
Diseño del modelo de datos	<ul style="list-style-type: none">• Diagrama de Casos de Uso.• Modelo Conceptual Aplicación Cliente.• Modelo Conceptual del Servidor <i>Push</i>.• Modelo de Base de Datos.
Diseño de la Aplicación	<ul style="list-style-type: none">• Estudio de usuario y el contexto de uso: perfiles de usuario, escenarios de uso.• Guías de estilo y prototipo de la interfaz de usuario para los principales casos de uso.• Diseño del prototipo en alta fidelidad.• Evaluación del prototipo con pruebas de usabilidad.
Revisión y entrega PEC 2	<ul style="list-style-type: none">• Revisión, actualización y entrega de la documentación realizada.

1.4.3 PEC 3: Implementación de las funcionalidades definidas (Sprint 2)

Durante este *Sprint* se lleva a cabo la implementación del software a partir de los requerimientos y el diseño establecido en la etapa anterior, realizando a su vez pruebas de aceptación para garantizar el cumplimiento de los requisitos. Este sistema se ha de incorporar

Actividad	Descripción
Implementación servidor notificaciones	<ul style="list-style-type: none">• Implementar el servidor de notificaciones.• Integrar la tecnología del servidor de notificaciones con el actual servidor de aplicaciones.
Desarrollo aplicación cliente	<ul style="list-style-type: none">• Implementación de la interfaz gráfica.• Implementación de las distintas funcionalidades.

	<ul style="list-style-type: none"> • Pruebas Unitarias básicas. • Pruebas de integración con el servidor.
Desarrollo simulador Analizador	<ul style="list-style-type: none"> • Desarrollo del simulador de analizadores que permita recrear el protocolo de mensajería entre el analizador y la aplicación servidor. • Pruebas de integración con el servidor y la aplicación cliente.
Depuración	<ul style="list-style-type: none"> • Depuración del código en pro de tener el producto lo más eficiente y estable posible.
Revisión y entrega PEC 3	<ul style="list-style-type: none"> • Revisión, actualización y entrega de la documentación realizada.

1.4.4 Entrega final (Sprint 3)

El último *Sprint* queda reservado a recopilar y revisar la documentación generada, realizar la presentación y defensa del mismo.

Actividad	Descripción
Elaboración de la memoria	<ul style="list-style-type: none"> • Recopilación y revisión de los entregables realizados durante el proyecto.
Revisión y entrega de la memoria	<ul style="list-style-type: none"> • Entrega final de la memoria final del proyecto. • Entrega del software desarrollado y las herramientas necesarias para su validación.
Presentación del proyecto	<ul style="list-style-type: none"> • Elaborar la presentación del proyecto. • Entrega del vídeo demostración del funcionamiento de la aplicación.

1.4.5 Planificación Temporal

Habiendo convenido que parte del desarrollo de este proyecto se realizará en el entorno de trabajo de la empresa (para la cual se desarrolla el proyecto), se establece una dedicación diaria en horario laboral de 2 horas, añadiendo a estas de 1 a 2 horas de dedicación fuera de dicho horario, sumando pues una dedicación de 3 a 4 horas diarias.

A estas se suman las horas a dedicar en fines de semana pudiendo añadir 6 horas los sábados.

	Fecha Inicio	Fecha Fin	Horas
PEC 1 – <i>Sprint 0</i>	24/02/2016	09/03/2016	60
PEC 2 – <i>Sprint 1</i>	09/03/2016	06/04/2016	104
PEC 3 – <i>Sprint 2</i>	06/04/2016	18/05/2016	156
Entrega Final – <i>Sprint 3</i>	18/05/2016	08/06/2016	70
		Total	390

En los siguientes diagramas de Gantt se puede apreciar el detalle de cada uno de los Sprints y las tareas que los componen.

Sprint 0 – Plan de Trabajo



Ilustración 2 – Diagrama de Gantt Sprint 0: Plan de Trabajo

Sprint 1 – Diseño

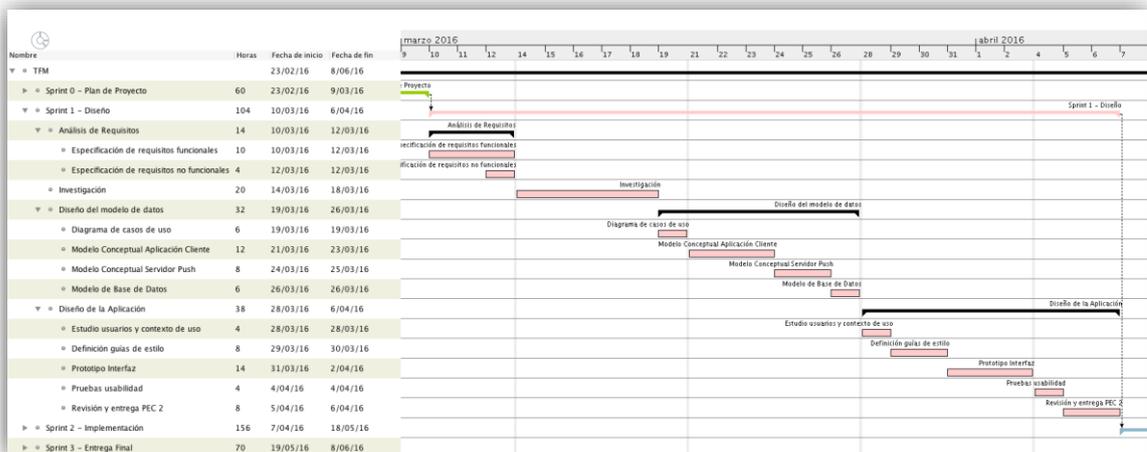


Ilustración 3 – Diagrama de Gantt Sprint 1: Diseño

Sprint 2 – Implementación

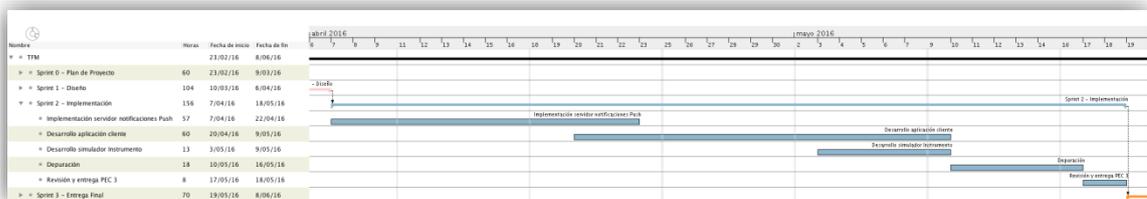


Ilustración 4 – Diagrama de Gantt Sprint 2: Implementación

Sprint 3 – Entrega Final

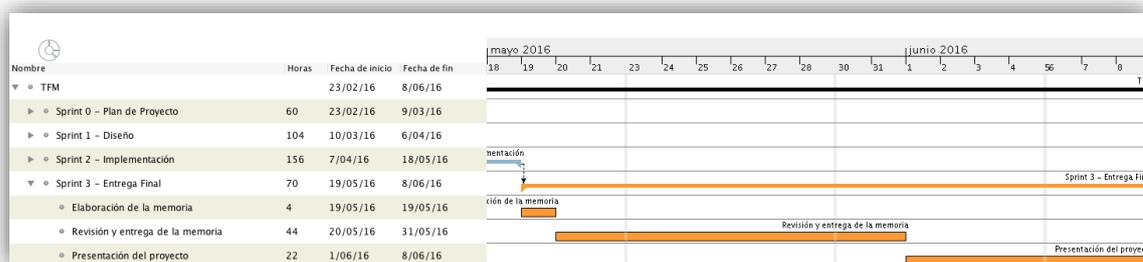


Ilustración 5 – Diagrama de Gantt Sprint 3: Entrega Final

1.5 Breve resumen de productos obtenidos

A la finalización de este proyecto se espera obtener:

- Servidor de notificaciones *Push* encargado de gestionar todas las notificaciones sobre los dispositivos móviles generadas por los eventos surgidos en los analizadores.
- Simulador de un analizador para generar los eventos a notificar a través del servidor de notificaciones y mostrar las acciones realizadas desde la aplicación móvil sobre los analizadores.
- Aplicación móvil que recibirá dichas notificaciones y permitirá realizar ciertas acciones sobre los analizadores que se mostrarán en el simulador.

1.6 Breve descripción de los otros capítulos de la memoria

El resto de capítulos de la memoria se desglosan en los distintos conceptos:

Diseño General

En este apartado se realizará una descripción detallada del funcionamiento de la solución a implementar. Centrándonos en las necesidades que este proyecto cubrirá y que necesidades o funcionalidades serán cubiertas en futuras revisiones del mismo.

Análisis de Requisitos y Diseño del Servidor de Notificaciones

Dentro de este apartado se especificarán los requisitos funcionales y no funcionales especificables en referencia al Servidor de Notificaciones, destacando:

- requerimientos funcionales y no funcionales;
- actores;
- casos de uso a cubrir;
- entidades, diagrama de clases, etc.

Análisis de Requisitos y Diseño de la Aplicación Cliente

Dentro de este apartado se especificarán los requisitos funcionales y no funcionales especificables en referencia a la Aplicación cliente, destacando:

- requerimientos funcionales y no funcionales;
- un estudio de las guías de estilo a aplicar;
- actores;
- casos de uso a cubrir;
- modelo y arquitectura;
- estructura de la aplicación;
- diseño de la interfaz gráfica del usuario;
- etc.

Análisis y Diseño del Simulador Analizador

Por la naturaleza del proyecto y que este se ha de conectar con analizadores a los que físicamente no se tiene acceso fuera de un laboratorio de un hospital, es necesario crear un simulador de estos. Para ello en este apartado se especificarán:

- requerimientos mínimos, funcionales y no funcionales, a cubrir;
- estructura de la aplicación;
- diseño de una interfaz gráfica de consulta e interacción.

Diseño del modelo de Datos

Este proyecto se nutre de cierta información de un sistema ya existente y se comunica con analizadores obteniendo de ellos información que actualmente no es procesada.

Es por ello que este proyecto ha de ampliar información dentro del sistema actual y se ha de nutrir con cierta información de dicho sistema. Para ello se especificarán que añadidos al modelo de datos actual se realizarán y qué

información del actual modelo de datos serán imprescindibles de consultar para el correcto funcionamiento de la aplicación.

Metodología de Pruebas de la Aplicación

En este apartado se especificarán el tipo de pruebas a realizar en cada uno de los *Sprints* siendo el *Sprint 1* el centrado en las pruebas sobre el prototipo y las propuestas y restando pruebas de funcionamiento de los distintos desarrollos (servidor, aplicación móvil y simulador) más detallados en los *Sprints 2 y 3*.

Propuestas de ampliaciones de Funcionalidades

A parte de las funcionalidades o características comentadas en el apartado de *Descripción General* y que no serán cubiertas por el proyecto, también se especificarán las mejoras o ampliaciones de funcionalidades pensadas para siguientes revisiones teniendo en cuenta que el protocolo de comunicaciones con los analizadores del cual se nutre este proyecto está en revisión y permitirá obtener una información más detallada y efectiva de los analizadores.

Otros apartados disponibles en esta memoria y que no hace falta especificar ya que su título es auto explicativo son:

- Conclusiones y líneas de futuro.
- Glosario.
- Bibliografía y referencias.
- Anexos

1.7 Herramientas de desarrollo

A lo largo de este apartado se describirán las diferentes herramientas empleadas para el desarrollo del proyecto. A causa de mi movilidad laboral los elementos hardware utilizados han ido variando en el transcurso del desarrollo del trabajo final de master.

1.7.1 Hardware

- Ordenador portátil Macbook Pro 13'
- Disco duro secundario Western Digital para la realización de copias de seguridad.
- Monitor Dell 21'
- Monitor HP 2420
- Móvil Motorola Moto G 2013:
 - Diagonal de pantalla: 4,5 pulgadas

- Resolución: 1280x720 píxeles (329ppp)
- Sistema Operativo: Android Lollipop 5.1

1.7.2 Software

- OSX El Capitan: sistema operativo propietario de Apple®
- Microsoft Office Word 2016: documentación de los entregables.
- Microsoft Office PowerPoint 2016: presentaciones y diseño de las diapositivas.
- Microsoft Office OneNote 2016: notas breves y ordenación de conceptos
- Gantt Project 2.7.2 Ostrava: planificación del proyecto.
- Gimp 2.8: tratamiento de las imágenes diseñadas para la aplicación.
- Inkscape 0.91: creación de imágenes vectoriales
- MySQL 5.5.47 Community Edition: gestor de la base de datos
- Vertabelo.com: diseño del esquema de base de datos y su generación.
- yEd 3.15.0.2: herramienta para el desarrollo de diagramas.
- Wildfly 8.2: servidor de aplicaciones.
- AeroGear 1.1.2: servidor de notificaciones *Push* bajo *Wildfly*.
- Axure RP 7: herramienta de diseño de los prototipos de la aplicación.
- Trello: herramienta de planificación de tareas para metodologías ágiles.
- Git: software de control de versiones.
- VirtualBox 5.0.20: software de virtualización para el servidor de aplicaciones y el servidor *Push*.
- Ubuntu 14.04: software del servidor donde se encuentran el servidor de base de datos y el de aplicaciones.
- Eclipse Mars.2 Release 4.5.2: para el desarrollo del simulador de Instrumento

2. Análisis

2.1. Usuarios y Contexto de Uso

A través de las diferentes entrevistas con Técnicos Especialistas en Implantaciones (TEI – ver [Anexo – Entrevistas](#)) se ha conseguido extrapolar un solo tipo de usuario al que puede estar orientado el producto a desarrollar. Este tipo de usuario son los Facultativos que están en los Laboratorios de análisis clínicos.

Las responsabilidades del Facultativo dentro del laboratorio son las de, entre otras muchas tareas, gestionar todo lo relacionado con los Analizadores y el trabajo a realizar con estos. Esto quiere decir que son los encargados de:

- Controlar que los analizadores se encuentren en condiciones óptimas para poder realizar las pruebas: niveles de reactivos, configuraciones, etc.
- Introducir muestras en los equipos para ser analizadas.
- Gestionar listas de trabajo a ejecutar por los analizadores. Por listas de trabajo se entienden un conjunto de pruebas a realizar para un grupo de muestras introducidas en el analizador.
- Ha de gestionar eficientemente dichas listas de trabajo de manera que le permita tener preparadas las siguientes muestras a tratar en el momento justo en el que las anteriores muestras hayan sido evaluadas. Para ello ha de tener en cuenta el tiempo requerido por parte del analizador para realizar las pruebas; este tiempo puede variar de 5 a 30 minutos por prueba.
- Dar solución a las posibles incidencias acaecidas en el transcurso del trabajo del analizador.
- En caso de trabajar con Cadenas de Analizadores, configurar dicha Cadena de Analizadores para que las pruebas se realicen en el menor tiempo posible y de la manera más eficiente.

Durante su trabajo diario, un Facultativo está en continuo movimiento por todo el laboratorio, esto quiere decir que no está constantemente delante de los analizadores para comprobar su funcionamiento, sino que pueden desplazarse y cambiar incluso de planta, en función de sus necesidades y responsabilidades. Igualmente, parte de su trabajo consiste en controlar el estado de los analizadores para detectar y estar presente por si:

- El analizador está próximo a terminar de realizar las pruebas sobre muestras suministradas. Esto requiere que el Facultativo se acerque al analizador para retirar las muestras con las que ha trabajado y reemplazarlas (en caso necesario) por el siguiente grupo de muestras a tratar.

- El nivel de reactivos para determinadas pruebas está próximo a agotarse. Lo que implica parar el analizador para reponer los reactivos
- Comprobar si el analizador se encuentra desconectado por algún motivo y no puede informar de resultados al LIS.
- Realizar una evaluación del estado del analizador en un determinado momento:
 - Estado o modo en el que se encuentra el analizador: calibración, mantenimiento o en funcionamiento.
 - Validar y dar respuesta a alarmas surgidas en el analizador que pueden indicar que:
 - Se encuentra parado por un error de hardware.
 - Se han obtenido errores en resultados de pruebas ejecutadas sobre la muestra, p.ej.: error de coagulación, valores de prueba fuera de rango, etc.
- Necesita consultar el trabajo que están realizando los distintos analizadores y cuál es el que queda pendiente de finalizar. Puede requerir realizar esta consulta también por si ha de modificar dicha lista de trabajo para realizar otras pruebas más prioritarias, conocidas como pruebas críticas o urgentes, de manera que han de ser realizadas en el menor tiempo posible.

En base a su movilidad y a las necesidades de un control en tiempo real de la situación de los analizadores (o cadena de analizadores) el Facultativo necesita:

- Identificarse dentro del sistema para poder consultar la información relativa a los analizadores.
- Consultar el estado del analizador en tiempo real: conexión, volumen de los reactivos, estado de funcionamiento del analizador (modo calibración, modo mantenimiento o en funcionamiento), etc.
- Ser notificado de cualquier evento o alarma relacionada con un analizador que requiera de su supervisión personal.
- Configurar la notificación de dichos eventos.
- Consultar la configuración aplicada a los analizadores.
- Consultar el trabajo asignado a un analizador: el trabajo que está realizando en un preciso instante y el trabajo planificado.

Para dar soporte a pruebas, se necesita la creación de un simulador de analizadores que permita dar respuesta a todas las necesidades a resolver con la aplicación, es decir, que permita simular un analizador de manera que:

- Permita consultar su estado: conexión, volúmenes de reactivos, estado de funcionamiento (calibración, mantenimiento, funcionamiento).

- Permita generar eventos o alarmas de notificación para ser enviadas a la aplicación.
- Permita gestionar listas de trabajo: lista de pruebas en tratamiento y lista de pruebas en espera.

2.2. Diseño Conceptual

2.2.1. Escenarios de uso

En base al perfil definido del usuario de la aplicación a continuación se definen las funcionalidades de mayor frecuencia a realizar por parte del usuario.

Escenarios de uso	
Escenario 1	Tareas
Comienza su jornada laboral en el laboratorio y necesita consultar el estado de los analizadores del laboratorio.	<ul style="list-style-type: none"> • Consultar en la lista principal el estado de los analizadores disponibles en el laboratorio.
Escenario 2	Tareas
Ha comenzado su jornada laboral y necesita preparar las muestras y los reactivos para esa mañana. Necesita conocer la lista de trabajo de los analizadores para ir preparando las muestras y los reactivos para la siguiente lista de trabajo.	<ul style="list-style-type: none"> • Accede a la lista de analizadores y consulta de cada uno de ellos su lista de trabajo, accediendo a ella desde la propia lista de analizadores.
Escenario 3	Tareas
Se encuentra alejado de la zona de analizadores y le comenta un compañero que no está seguro de si la configuración que han realizado a un analizador se corresponde con la que debería de tener.	<ul style="list-style-type: none"> • Acceder al detalle de la configuración del analizador a través de la lista de analizadores.
Escenario 4	Tareas
Tiene una hoja impresa con la configuración de un analizador con el trabajo a realizar por este durante las próximas horas. Necesita conocer el trabajo que está realizando en ese momento el analizador y si está configurado correctamente con dicha lista de trabajo.	<ul style="list-style-type: none"> • Acceder al detalle del analizador a través de la lista de analizadores y consulta la actual lista de trabajo que está realizando el mismo.

Escenario 5	Tareas
Se encuentra alejado de la zona de analizadores y recibe una notificación de una incidencia acaecida en un analizador.	<ul style="list-style-type: none"> • Consultar qué tipo de notificación ha recibido. • En caso de ser una notificación de desconexión o error se desplaza al analizador afectado o avisa de que sea tratada. • En caso de ser una alarma, si esta es de falta de reactivos, se desplaza al analizador afectado o avisa de que alguien la solvente.
Escenario 6	Tareas
El usuario quiere dejar de recibir notificaciones de un analizador en concreto porque ese analizador durante las próximas semanas será utilizado por otro departamento del laboratorio.	<ul style="list-style-type: none"> • El usuario accede a la lista de notificaciones de analizadores y desmarca dicho analizador de ser notificado.

2.2.2. Diagramas uso

Partiendo de los escenarios de uso, el análisis de los usuarios y la evaluación del contexto de uso que requieren, se han determinado los siguientes casos de uso para el usuario Facultativo:

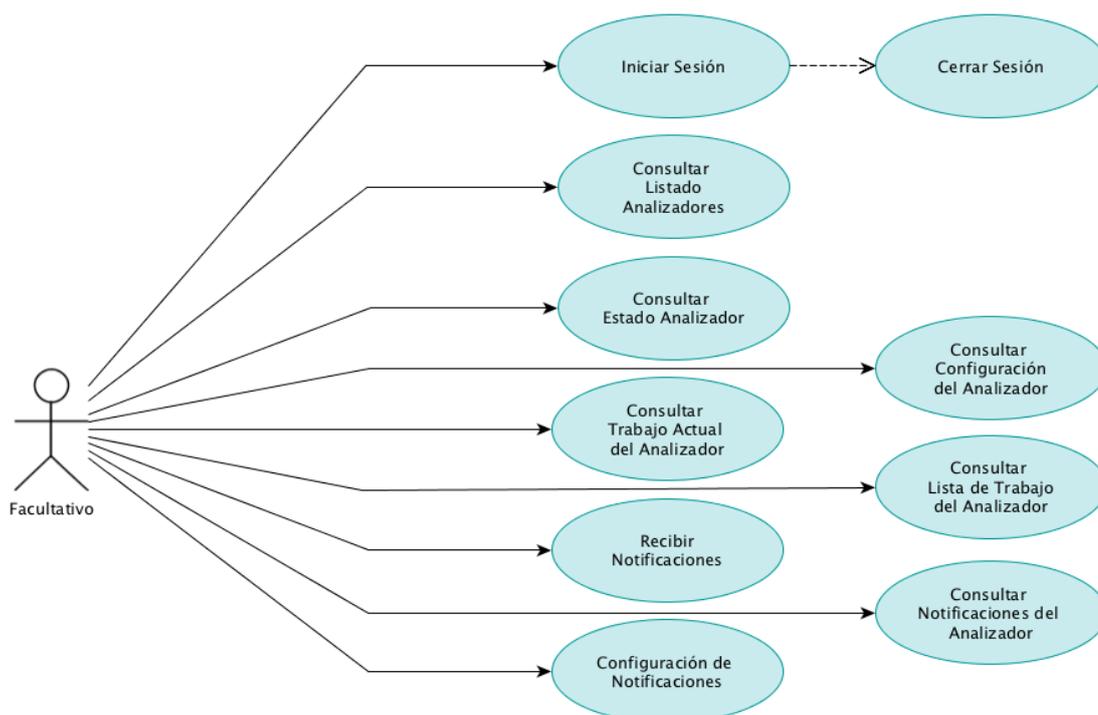


Ilustración 6 – Casos de Uso: Aplicación Móvil

A continuación se realiza una explicación detallada de cada uno de los casos de uso representados en la ilustración 6.

Caso de Uso	
UC-1	Iniciar Sesión
Descripción	Acción que permite al usuario iniciar sesión en el sistema.
Actor	Facultativo.
Pre-Condiciones	<ul style="list-style-type: none"> No estar autenticado en el sistema a través de la aplicación. El usuario y la contraseña a utilizar han de estar dadas de alta en el sistema de gestión de usuarios.
Escenario Principal	<ol style="list-style-type: none"> El usuario recibe una notificación o necesita realizar consultas del estado de un analizador. El usuario accede a la aplicación e introduce un Usuario y una Contraseña. El sistema comprueba la validez del Usuario y la Contraseña introducidos. El usuario accede al detalle del analizador que ha generado la notificación o a la pantalla principal de resumen de estado de los analizadores.
Escenario Adicional	Si el usuario no consigue iniciar sesión, el sistema le permitirá le permitirá intentarlo de nuevo hasta un máximo de 5 intentos. Pasados ese número de intentos el sistema bloqueará al usuario.
Post-Condiciones	El usuario ha iniciado sesión correctamente dentro del sistema y la aplicación.
Frecuencia de Uso	Muy alta
Estado	Desarrollado
Prioridad	Alta

Caso de Uso	
UC-2	Cerrar Sesión
Descripción	Acción que permite cerrar la sesión del usuario en el sistema.
Actor	Facultativo.
Pre-Condiciones	<ul style="list-style-type: none"> Tener una sesión activa en el sistema a través de la aplicación.
Escenario Principal	<ol style="list-style-type: none"> El usuario seleccionará la opción del menú principal "Cerrar Sesión". El sistema registrará el cierre de la sesión por parte del usuario. La aplicación mostrará la pantalla de inicio de sesión.
Escenario Adicional	Si el usuario no consigue "Cerrar Sesión", el sistema automáticamente cerrará la sesión del usuario y la aplicación mostrará la pantalla de inicio de sesión.
Post-Condiciones	El usuario se ha autenticado correctamente en la aplicación.
Frecuencia de Uso	Muy alta
Estado	Desarrollado

Prioridad	Alta
------------------	------

Caso de Uso UC-3		Consultar Listado Analizadores
Descripción	Funcionalidad que permite visualizar en una lista el resumen del estado de los analizadores disponibles en el sistema.	
Actor	Facultativo.	
Pre-Condiciones	<ul style="list-style-type: none"> • Tener una sesión activa en el sistema a través de la aplicación. • Tener dado de alta al menos un analizador dentro del sistema. 	
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario visualiza en modo de lista la información más relevante relacionada con los analizadores. <ul style="list-style-type: none"> • Nombre del Analizador. • Estado de conexión del analizador. • Resumen del número de alarmas o notificaciones del analizador. • Nombre de la prueba que actualmente está realizando el analizador. 2. Pulsando sobre un analizador se mostrará el estado del analizador en detalle. 3. Pulsando sobre la opción de menú el usuario puede acceder a: <ul style="list-style-type: none"> • Configurar las notificaciones a recibir. • Cerrar la sesión iniciada. 	
Escenario Adicional	En caso de no haber ningún analizador creado en el sistema se mostrará un mensaje informativo.	
Post-Condiciones	El usuario ha podido consultar de forma breve y efectiva el estado de los analizadores disponibles en el sistema.	
Frecuencia de Uso	Muy alta	
Estado	Desarrollado	
Prioridad	Alta	

Caso de Uso UC-4		Consultar Estado Analizador
Descripción	Funcionalidad que permite al usuario conocer el estado en tiempo real de un analizador.	
Actor	Facultativo.	
Pre-Condiciones	<ul style="list-style-type: none"> • Tener una sesión activa en el sistema a través de la aplicación. • Haber seleccionado un analizador. 	
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario visualizará en detalle el estado del analizador. Esto quiere decir: <ul style="list-style-type: none"> • Nombre y descripción del tipo de analizador. • Estado de conexión del analizador. 	

	<ul style="list-style-type: none"> • Número de alarmas disparadas en el analizador. • Prueba en proceso dentro del analizador. • Tiempo estimado de la finalización de la prueba. <ol style="list-style-type: none"> 2. El usuario puede visualizar un listado resumen de las pruebas que están pendientes de realizar en el analizador. <ul style="list-style-type: none"> • Nombre de la prueba. • Previsión del tiempo en el que se realizará. 3. El usuario podrá consultar el listado de alarmas y/o errores generados en el analizador. 4. El usuario puede volver al listado de analizadores a través del acceso en la cabecera de la aplicación. 5. El usuario puede acceder al detalle de la configuración de pruebas del analizador a través del acceso en el detalle. 6. Pulsando sobre la opción de menú el usuario puede acceder a: <ul style="list-style-type: none"> • Configurar las notificaciones a recibir. • Cerrar la sesión iniciada.
Escenario Adicional	Si el sistema no consigue obtener la información del estado del analizador, el sistema mostrará un aviso.
Post-Condiciones	El usuario ha podido consultar el detalle del estado del analizador seleccionado.
Frecuencia de Uso	Muy alta
Estado	Desarrollado
Prioridad	Alta

Caso de Uso	
UC-5	
Consultar Configuración Analizador	
Descripción	Funcionalidad que permite al usuario conocer la configuración de las pruebas de un analizador.
Actor	Facultativo.
Pre-Condiciones	<ul style="list-style-type: none"> • Tener una sesión activa en el sistema a través de la aplicación. • Haber seleccionado la opción de configuración de un analizador
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario visualiza el detalle de la configuración de las pruebas del analizador. <ul style="list-style-type: none"> • Nombre de la prueba. • Unidades de la prueba. • Volumen de pruebas que el analizador puede todavía realizar sin necesitar un cambio de reactivos. • Última vez que se realizó dicha prueba en el analizador. 2. El usuario puede volver al listado de analizadores a través del acceso en la cabecera de la aplicación. 3. Pulsando sobre la opción de menú el usuario puede acceder a: <ul style="list-style-type: none"> • Configurar las notificaciones a recibir. • Cerrar la sesión iniciada.

Escenario Adicional	Si el sistema no consigue obtener la información del estado de la configuración de pruebas del analizador, el sistema mostrará un aviso.
Post-Condiciones	El usuario ha conseguido consultar la información referente al estado de la configuración de las pruebas del analizador.
Frecuencia de Uso	Media
Estado	Desarrollado
Prioridad	Media

Caso de Uso	
UC-6 Consultar Trabajo Actual del Analizador	
Descripción	Funcionalidad que permite al usuario consultar la prueba que está realizando actualmente el analizador.
Actor	Facultativo.
Pre-Condiciones	<ul style="list-style-type: none"> Tener una sesión activa en el sistema a través de la aplicación.
Escenario Principal	<ol style="list-style-type: none"> El usuario visualiza cual es la prueba que está realizando actualmente el analizador. <ul style="list-style-type: none"> Nombre de la prueba. Unidades. Tiempo de inicio de la prueba. Tiempo de previsión de ejecución de la prueba. Reactivos que se utilizan en la prueba. Gasto de los reactivos que se utilizan en la prueba. El usuario puede volver al listado de analizadores o a la lista de trabajo del analizador a través del acceso en la cabecera de la aplicación. Pulsando sobre la opción de menú el usuario puede acceder a: <ul style="list-style-type: none"> Configurar las notificaciones a recibir. Cerrar la sesión iniciada.
Escenario Adicional	<p>Si el sistema no consigue recuperar la información referente a la prueba que está actualmente realizando el analizador mostrará un mensaje de error.</p> <p>Si el analizador está desconectado, está en mantenimiento, calibración o no está realizando ninguna prueba el sistema mostrará un mensaje de aviso.</p>
Post-Condiciones	El usuario ha conseguido consultar el detalle de la prueba que se está realizando en ese preciso instante en el analizador
Frecuencia de Uso	Alta
Estado	Desarrollado
Prioridad	Alta

Caso de Uso UC-7		Consultar Lista de Trabajo del Analizador
Descripción	Funcionalidad que permite consultar la lista de pruebas que tiene pendientes de realizar el analizador.	
Actor	Facultativo.	
Pre-Condiciones	<ul style="list-style-type: none"> • Tener una sesión activa en el sistema a través de la aplicación. • Haber seleccionado un analizador que estuviera funcionando con una lista de trabajo activa. 	
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario visualiza en modo de lista la información más relevante relacionada con las pruebas pendientes de realizar. <ul style="list-style-type: none"> • Nombre de la prueba. • Hora prevista de inicio de la prueba.. 2. Pulsando sobre una prueba se mostrará el detalle de la prueba. 3. El usuario puede volver al listado de analizadores o a la lista de trabajo del analizador a través del acceso en la cabecera de la aplicación 4. Pulsando sobre la opción de menú el usuario puede acceder a: <ul style="list-style-type: none"> • Configurar las notificaciones a recibir. • Cerrar la sesión iniciada. 	
Escenario Adicional	<p>Si el sistema no consigue recuperar la información referente a la lista de trabajo del analizador mostrará un mensaje de error.</p> <p>Si el analizador está desconectado, está en mantenimiento, calibración o no está realizando ninguna prueba el sistema mostrará un mensaje de aviso.</p>	
Post-Condiciones	El usuario consigue visualizar la lista de trabajo del analizador seleccionado.	
Frecuencia de Uso	Normal	
Estado	Desarrollado	
Prioridad	Normal	

Caso de Uso UC-8		Consultar Notificaciones del Analizador
Descripción	Funcionalidad que permite al usuario consultar las notificaciones que ha ido enviando el analizador.	
Actor	Facultativo.	
Pre-Condiciones	<ul style="list-style-type: none"> • Tener una sesión activa en el sistema a través de la aplicación. 	
Escenario Principal	<ol style="list-style-type: none"> 1. El usuario visualiza en modo de listas las diferentes notificaciones agrupadas por su categoría. Estas categorías son: <ul style="list-style-type: none"> • Notificaciones informativas. • Alarmas. • Advertencias. 	

	2. Pulsando sobre el título de cada uno de los tabuladores podrá consultar la categoría correspondiente.
Escenario Adicional	Si no existen notificaciones, en cada una de las pestañas se mostrará un mensaje informativo.
Post-Condiciones	El usuario ha consultado las notificaciones del analizador.
Frecuencia de Uso	Alta
Estado	Desarrollado
Prioridad	Alta

Caso de Uso	
UC-9	Recibir Notificaciones
Descripción	Acción que permite al usuario recibir notificaciones enviadas por parte del sistema.
Actor	Facultativo.
Pre-Condiciones	<ul style="list-style-type: none"> El usuario ha de haberse registrado al menos una vez dentro del sistema a través de la aplicación. Un analizador ha generado una notificación.
Escenario Principal	<ol style="list-style-type: none"> El usuario visualiza en la pantalla del móvil la notificación del analizador. El usuario accede a la aplicación a través de la notificación. En caso de no disponer de una sesión iniciada se le mostrará la pantalla de inicio de sesión. Una vez iniciada la sesión, o si esta ya estuviera iniciada, se le mostrará al usuario el detalle del analizador con un mensaje explicativo de la notificación.
Escenario Adicional	Si el usuario no tiene acceso a la red del servidor se mostrará un mensaje de error.
Post-Condiciones	El usuario ha sido notificado de un evento sobre un analizador.
Frecuencia de Uso	Muy alta
Estado	Desarrollado
Prioridad	Alta

Caso de Uso	
UC-10	Configuración de Notificaciones
Descripción	Funcionalidad que permite al usuario definir sobre qué eventos quiere ser notificado.
Actor	Facultativo.
Pre-Condiciones	<ul style="list-style-type: none"> Tener una sesión activa en el sistema a través de la aplicación.

Escenario Principal	<ol style="list-style-type: none"> 1. El usuario visualiza en modo de lista las diferentes opciones de notificaciones a ser informado, a modo de ejemplo: <ul style="list-style-type: none"> • Desconexión de un analizador. • Cambios de estado del analizador: mantenimiento, calibración, etc. • Errores en la ejecución de las pruebas. • Alarmas relacionadas con el estado de los reactivos de un analizador. • Finalización de la ejecución de una prueba. • Finalización de listas de trabajo. 2. Pulsando sobre un seleccionador en cada una de las opciones se activaran o desactivarán dichas notificaciones. 3. Pulsando sobre la opción de menú "Guardar" el usuario registrará sus nuevas preferencias de notificación. 4. Pulsando sobre la opción de menú el usuario puede acceder a: <ul style="list-style-type: none"> • Configurar las notificaciones a recibir. • Cerrar la sesión iniciada.
Escenario Adicional	Si el sistema no puede guardar las preferencias de notificaciones se mostrará un mensaje de error.
Post-Condiciones	El usuario guarda correctamente las preferencias de notificaciones.
Frecuencia de Uso	Muy alta
Estado	Desarrollado
Prioridad	Alta

Tal y como se ha ido explicando a lo largo de los capítulos anteriores, este proyecto requiere de un simulador de analizadores para poder realizar las pruebas de integración. En este caso, el usuario que se define es un usuario genérico, ya que no son necesarios los conocimientos médicos para hacer uso del mismo.

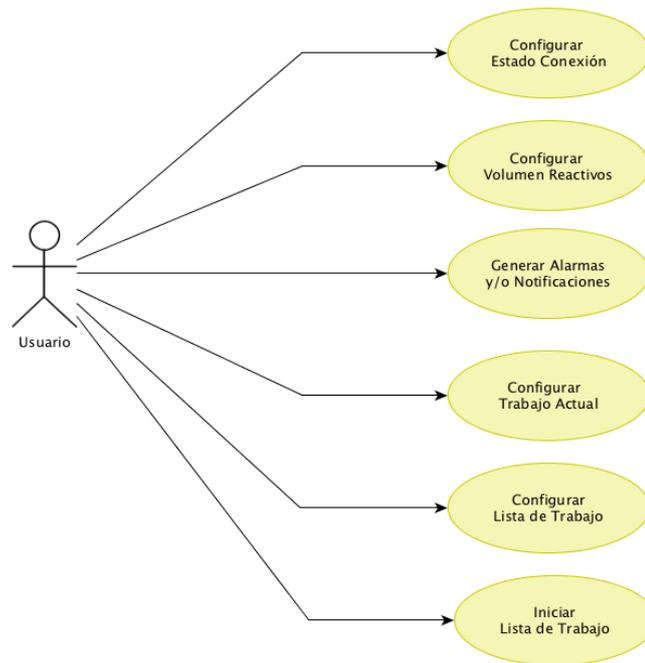


Ilustración 7 – Casos de Uso: Simulador de Analizadores

Debido a que el propósito principal del proyecto no es la realización de este simulador, sino la aplicación en si misma y la implantación del servidor de notificaciones *Push*, la definición de estos casos de uso se realizará una descripción breve.

1. Configurar Estado Conexión: a través de la pantalla de detalle del analizador permitirá simular la conexión o desconexión del mismo de la red.
2. Configurar Volumen Reactivos: permitirá registrar el volumen que se quiere mostrar de reactivos por tal de informar al usuario del número de ensayos a poder realizar de las distintas pruebas configuradas.
3. Generar Alarmas y/o Notificaciones: a través de la pantalla de detalle del analizador, el usuario podrá forzar las alarmas o notificaciones a enviar a los usuarios registrados. Estas alarmas o notificaciones están relacionadas con eventos de estado del analizador así como eventos de realización de las pruebas. También se generará una notificación automática en el momento en el que una lista de trabajo comience la última de las pruebas en un analizador.
4. Configurar Trabajo Actual: permite al usuario indicar qué ensayo está realizando actualmente el analizador.
5. Configurar Lista de Trabajo: permite al usuario crear una lista de ensayos a realizar para que el usuario pueda consultar el trabajo pendiente del analizador a través de la aplicación.
6. Iniciar Lista de trabajo: permite al usuario comenzar a ejecutar las pruebas de la lista de trabajo de manera que se vayan realizando y al final se notifique la finalización de la lista mediante una notificación *Push*.

2.3. Diseño Arquitectura

En este apartado definiremos la arquitectura del sistema identificando las entidades que se representan en el modelo de base de datos, el modelo de dominio de la aplicación y la estructura empleada para realizar las llamadas al servidor desde la aplicación cliente y desde el simulador de analizadores.

2.3.1. Diagrama de Base de Datos

En el siguiente diagrama se recoge las entidades utilizadas por el sistema (incluido el simulador de analizadores).

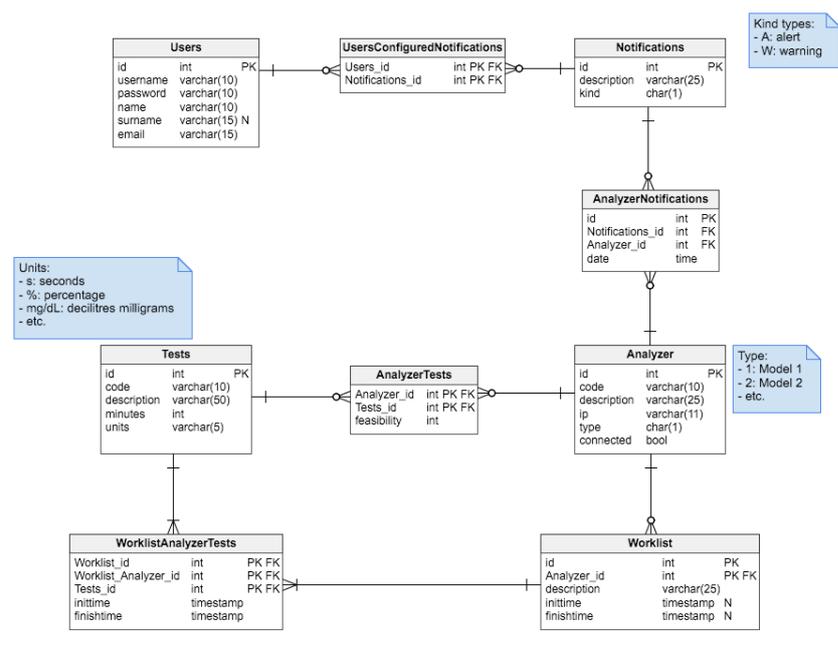


Ilustración 8 – Diagrama de Base de Datos

2.3.2. Diagrama UML de Clases

En el siguiente diagrama muestra las principales clases identificadas en el dominio de la aplicación.

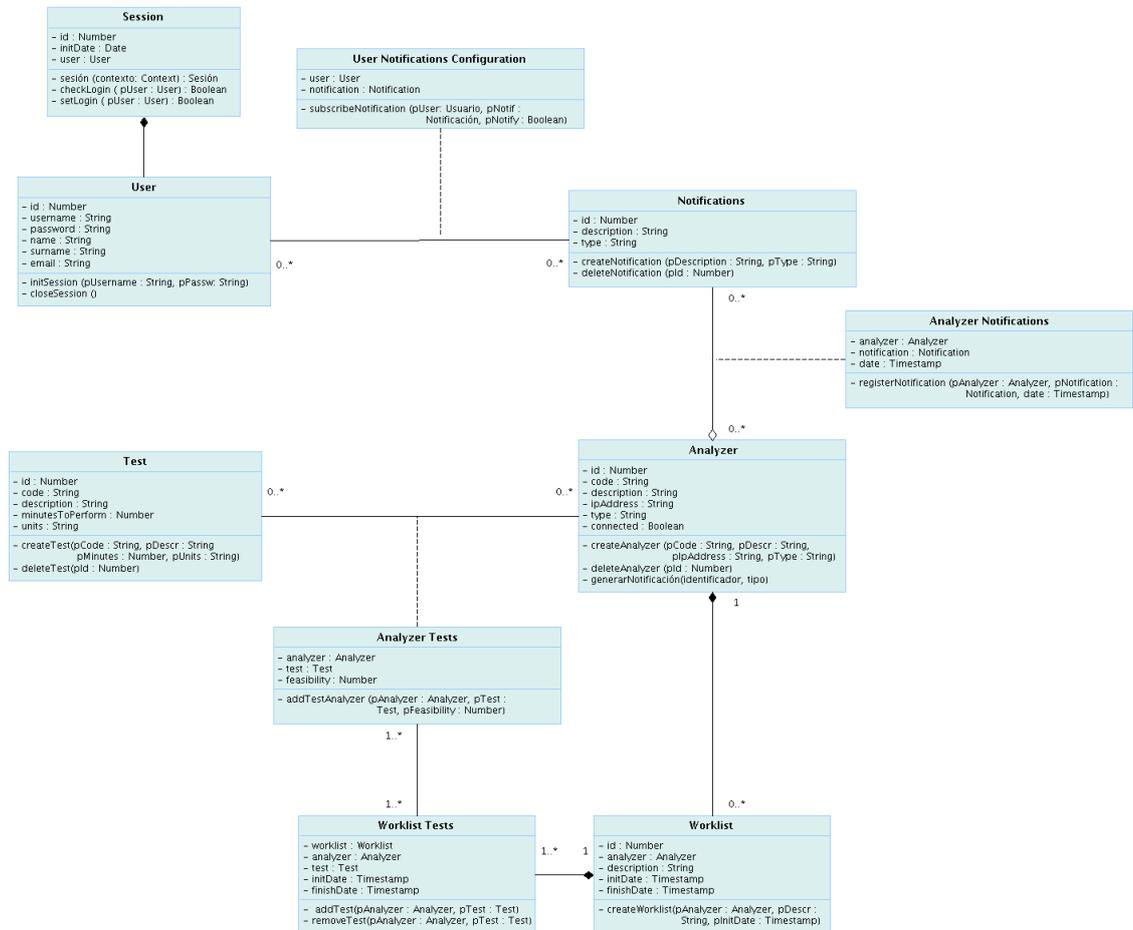


Ilustración 9 – Diagrama UML de Clases

2.3.3. Arquitectura del Sistema

Con la intención de obtener un software robusto donde prime la facilidad de mantenimiento, reutilización de código y la separación de conceptos se opta por emplear una arquitectura Modelo-Vista-Controlador (MVC).

- **Modelo:** es responsable de acceder a la capa de almacenamiento de base de datos. Para este proyecto, los datos permanecerán en una base de datos *MySQL* al que accederemos a través de un patrón Singleton para limitar inyecciones SQL. Como caso excepcional, este proyecto dispone de dos modelos independientes que gestionarán dos bases de datos: la propia con la información del proyecto y la utilizada para gestionar las notificaciones. La gestión de este segundo modelo recae sobre el proyecto OpenSource AeroGear.

- **Vista:** es responsable de recibir los datos del modelo y mostrarlos al usuario. En este proyecto habrán dos tipos de vistas, las utilizadas por la aplicación cliente y las utilizadas por el simulador de analizadores. Desde estas dos vistas se realizarán las peticiones de datos y será responsable de generar la salida correspondiente.
- **Controlador:** recibe los eventos de entrada y gestiona el flujo de información entre la vista y el modelo. En este proyecto, se responsabiliza de gestionar tanto los eventos de la aplicación cliente como de los del simulador de analizadores. Esto quiere decir las consultas realizadas a través de la aplicación cliente y las consultas y modificaciones desde el simulador de analizadores.

En la siguiente ilustración se muestra, de forma esquemática, el funcionamiento básico.

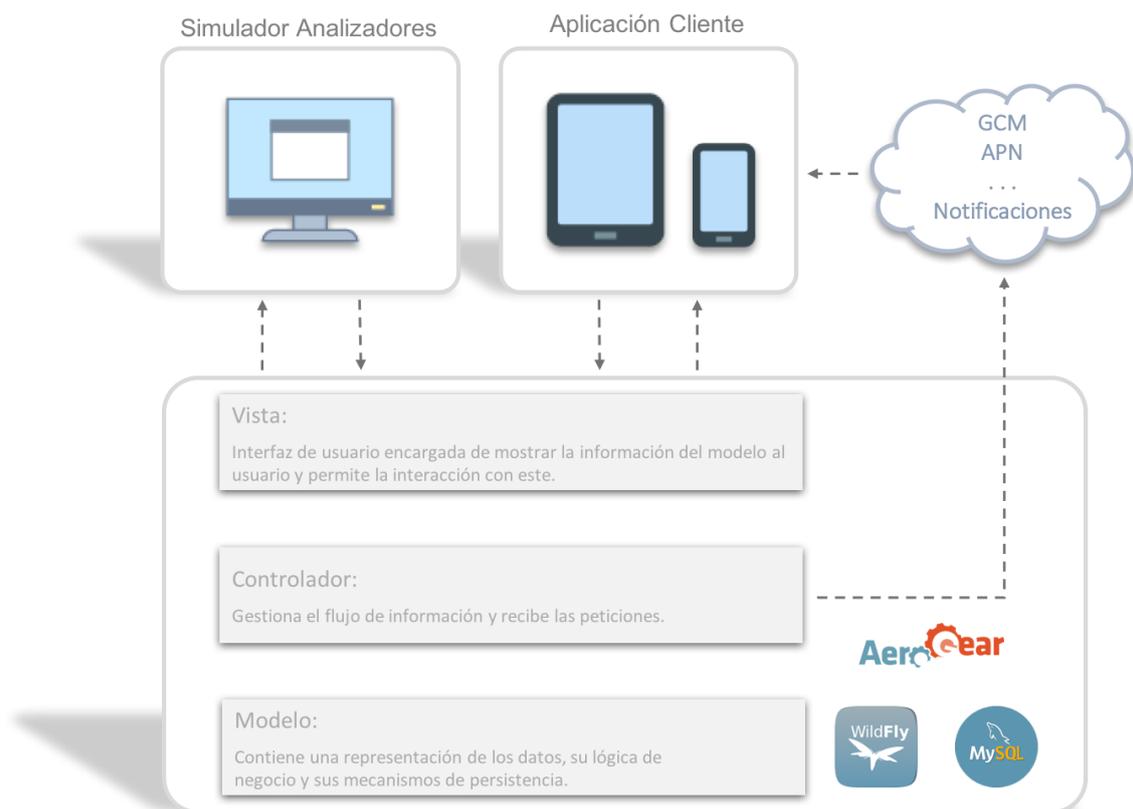


Ilustración 10 – Arquitectura del Sistema

2.4. Diseño de la persistencia

Este proyecto, al tener que realizar una simulación de equipos analizadores, ha requerido de un diseño de la persistencia que difiere del que el producto tendrá en futuras versiones, siendo la intención del producto de integrarse en un sistema de aplicaciones real. Es por ello que tanto la aplicación móvil como la aplicación de simulador de analizadores hacen uso de una base de datos MySQL que es la base de datos que utiliza el servidor de notificaciones.

2.4.1. Persistencia en la parte servidor

Tal y como se comenta al inicio de este capítulo, la capa de persistencia relativa a la información de los analizadores, el servidor de notificaciones y la base de datos que simula ser la correspondiente a un servidor de aplicaciones (gestión de usuarios, datos de analizadores, etc.) se ha decidido implementar sobre el Sistema de Gestión de Base de Datos (SGBD) MySQL 5.5.

La información almacenada en la parte de servidor se compone de:

- Información relativa a los usuarios con accesos a la aplicación.
- Información relativa a los analizadores:
 - Datos descriptivos.
 - Listas de trabajo.
 - Pruebas configuradas.
 - Notificaciones realizadas.
- Información relativa al sistema de notificación.

La implementación desarrollada permite traspasar toda la información relativa al sistema de notificaciones a otro sistema de manera que se permitan realizar implementaciones alternativas.

2.4.2. Persistencia en la aplicación móvil

Siendo la información gestionada por la aplicación móvil proveniente del servidor es sumamente, se ha decidido guardar la mínima información indispensable dentro de la propia aplicación. De este modo la única información que se ha consensado de guardar es la referente a las configuración de las notificaciones a recibir.

3. Prototipado

3.1 Prototipo Simulador Analizador

Este proyecto tiene la característica de necesitar un simulador de analizadores, la presente ilustración muestra la manera en que el usuario puede interactuar con el simulador de analizadores y generar las notificaciones necesarias.

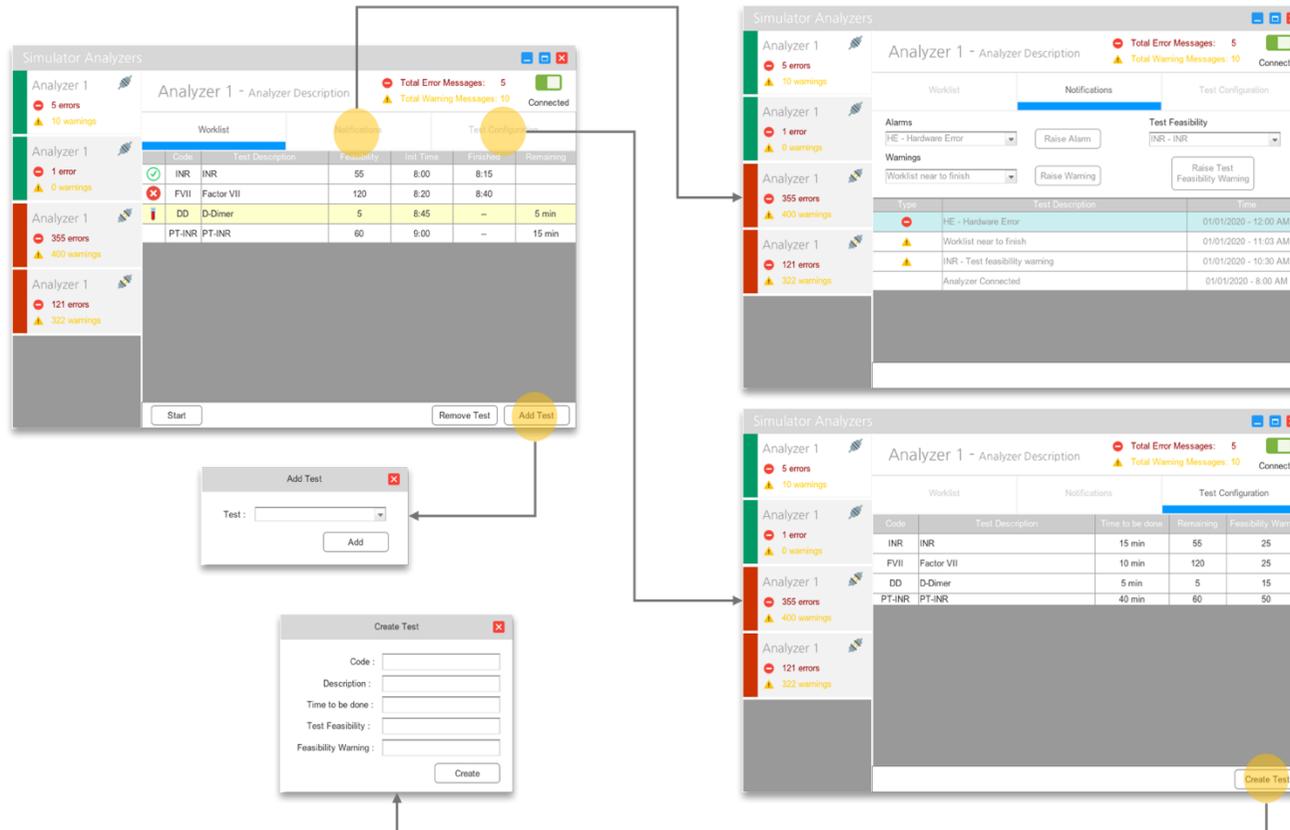


Ilustración 11 – Prototipo Simulador de Analizadores

3.2 Prototipo Aplicación Móvil

En base a la estructura general de la aplicación en función de los flujos de iteración mostrados en el capítulo anterior, la siguiente ilustración muestra la idea del proyecto de manera breve, haciendo hincapié en la sencillez de uso y la fácil visualización de los datos.

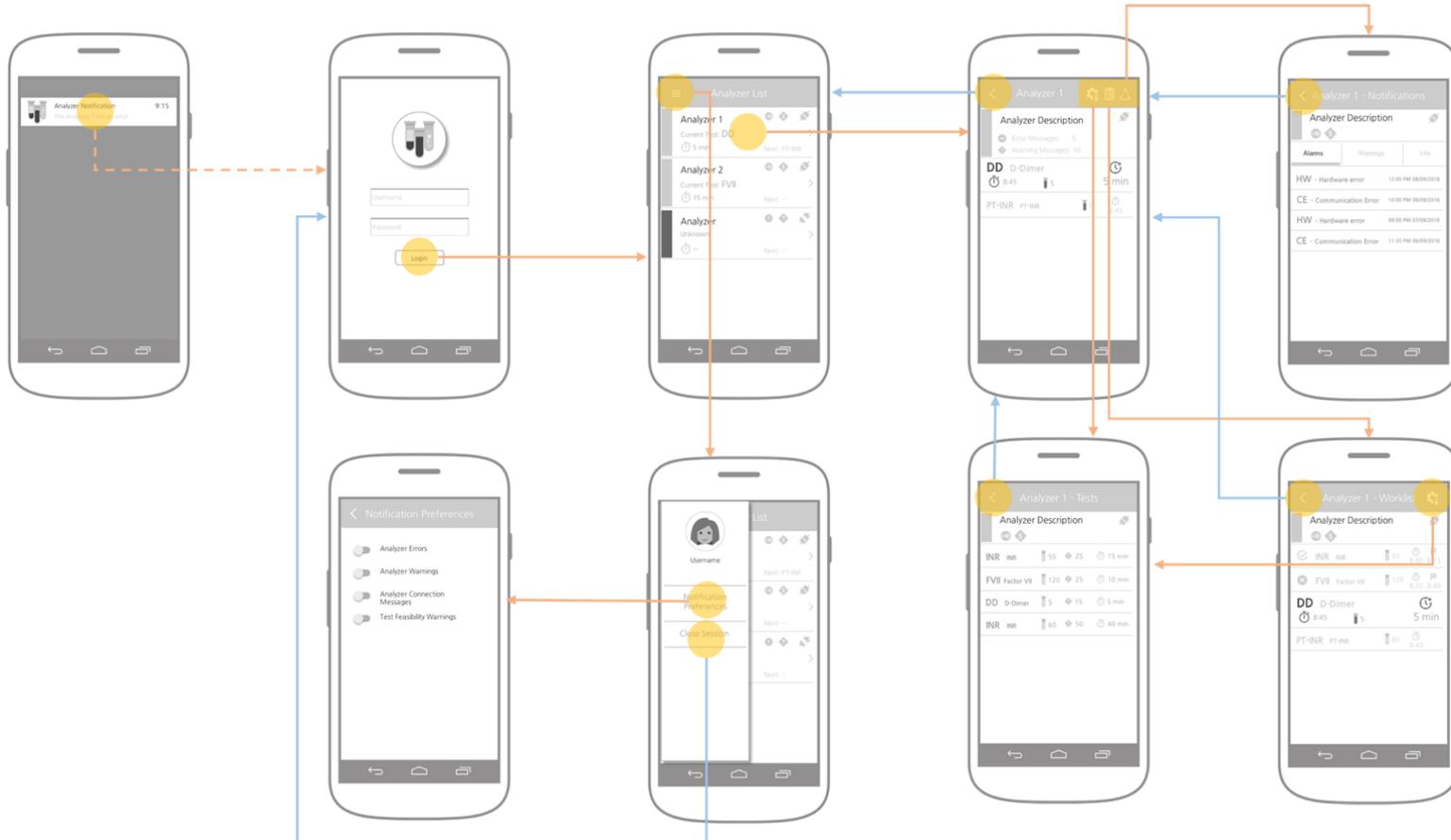


Ilustración 12 – Prototipo Aplicación Móvil

3.3 Prototipo Aplicación Móvil (alta definición)

En base al esquema presentado en el apartado anterior, el resultado del prototipado aplicado a la aplicación final es el siguiente:

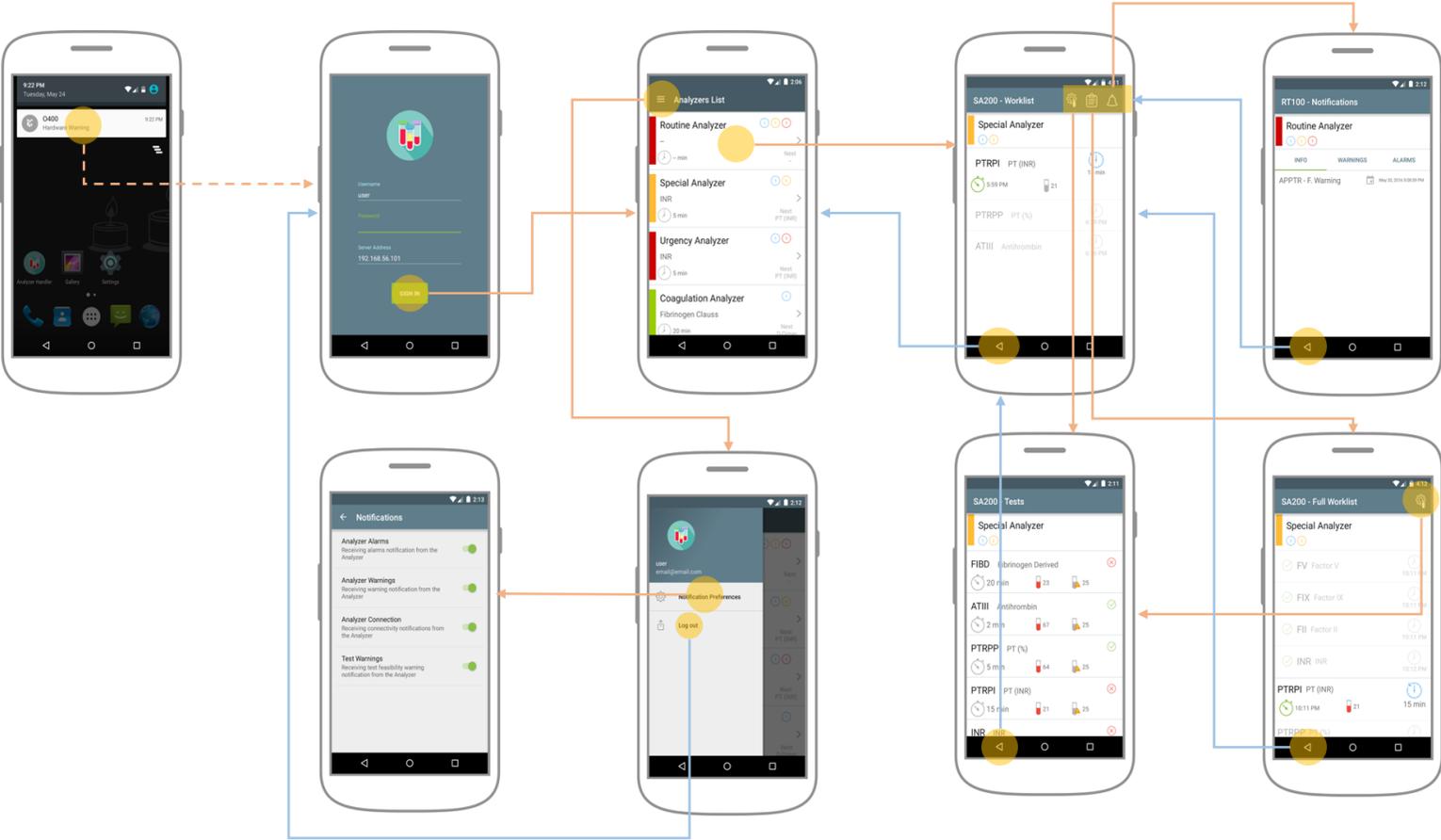


Ilustración 13 – Prototipado Aplicación Móvil (alta definición)

3.4 Diseño gráfico e interfaces

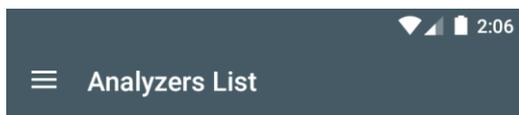
En este apartado se describen los diferentes elementos que forman parte del diseño gráfico de la aplicación. Siendo la simplicidad y el minimalismo las bases que han imperado en el desarrollo del diseño de la aplicación, los siguientes elementos gráficos y funcionales han sido escogidos bajo esas premisas.

Para el diseño gráfico se ha intentado seguir las pautas establecidas en la normativa de diseño de Google *Material design*. Este tipo de diseño permite obtener un apartado visual más limpio en el que predominan las animaciones y transiciones, así como el relleno y los efectos de profundidad como la iluminación y las sombras. Aunque no se ha seguido estrictamente las propuestas de este diseño, se ha obtenido un resultado limpio y de fácil uso por parte del usuario.

1.1.1. Action Bar

Este recurso de diseño se utiliza para proporcionar al usuario de una navegación sencilla e intuitiva.

En concreto este recurso es utilizado en varias de las pantallas ofreciendo acceso a diferentes funcionalidades de la aplicación:

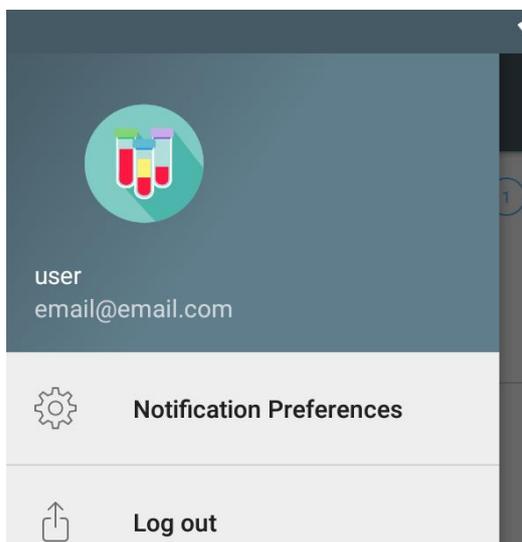


Da acceso al menú de la aplicación que permite configurar las notificaciones y salir de la aplicación.



Da acceso a diferentes funcionalidades relacionadas con un analizador: lista de pruebas configuradas, lista de trabajo completa y notificaciones recibidas.

1.1.2. Drawer Menu

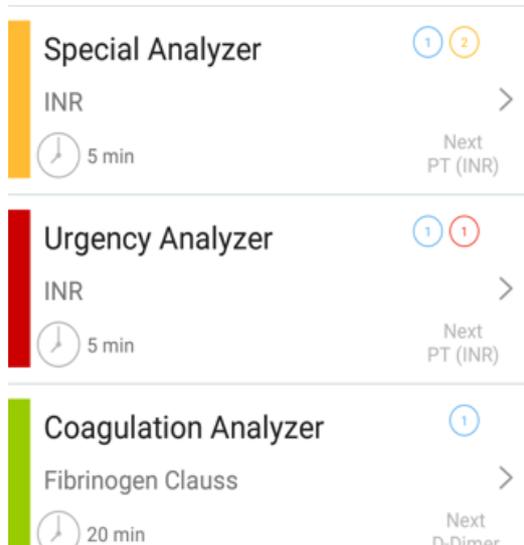


Tal y como se describe en el punto anterior, desde recurso se puede acceder a configurar las notificaciones que se desean recibir, así como

A futuro este menú puede albergar más posibilidades en cuanto a funcionalidades.

1.1.3. Detalle Analizador

Para el detalle del analizador, se ha intentado ofrecer toda la información que es relevante de la manera más sencilla y entendible.



De esta manera, visualizando el contenido mostrado de izquierda a derecha, podemos saber:

4. Barra de estado del analizador (rojo, naranja o verde).
5. Nombre del analizador.
6. Cuantas notificaciones muestra el analizador.
7. El nombre de la prueba que está ejecutando en ese instante el analizador.
8. El tiempo restante que queda hasta que la prueba esté finalizada.
9. El nombre de la siguiente prueba que tiene pendiente de realizar el analizador, si es que la hubiera.

1.1.4. Detalle Worklist

En el detalle de una lista de trabajo se intenta mostrar de manera clara cuales han sido las pruebas ya realizadas y cuál es la que se está realizando en ese preciso instante.



Entrando en detalle sobre la información mostrada podemos observar para cada una de las pruebas:

10. El código y descripción de la prueba.
11. Hora cuando se finalizó y si esta lo hizo satisfactoriamente o no.
12. En caso de ser la prueba que se esté realizando:
 - Código y descripción.
 - Cuando comenzó la prueba.
 - El tiempo de finalización.
 - El número de pruebas restantes.

1.1.5. Detalle Pruebas

Para la pantalla de detalle de pruebas se ha intentado resaltar de manera visual toda la información relativa a las mismas.

FIBD Fibrinogen Derived	⊗	
20 min	23	25
ATIII Antihrombin	✓	
2 min	67	25

El detalle de la información se compone de:

13. El código de la prueba.
14. La descripción de la prueba.
15. Tiempo necesaria para finalizarla.
16. Número de pruebas restantes.
17. Límite en el que se ha de realizar la notificación de falta de muestras.
18. Marca visual de si la prueba puede realizarse.

1.1.6. Detalle de las notificaciones

Para el detalle de las notificaciones de un analizador se ha optado por utilizar el recurso de las pestañas en tabuladores. Al considerarse todas como notificaciones, este recurso nos permite diferenciar de manera clara la información de cada una de ellas.

Routine Analyzer		
1	1	1
INFO	WARNINGS	ALARMS
APPTR - F. Warning May 20, 2016 5:08:59 PM		

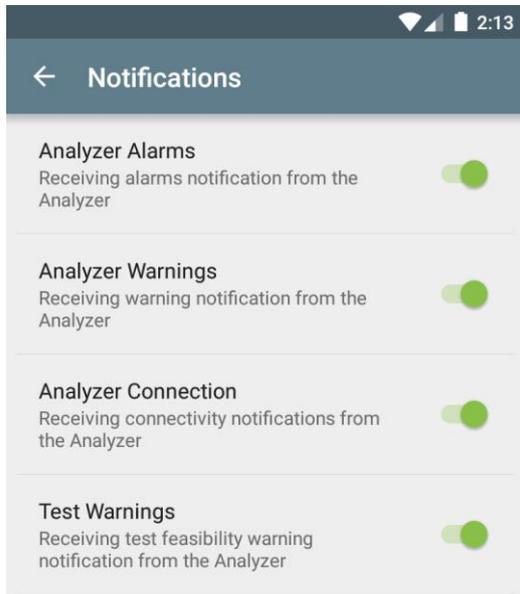
El mensaje a mostrar en cada una de las notificaciones se corresponde con:

19. El código de la notificación.
20. La fecha y hora de cuando se generó dicha notificación.

En el código de la notificación puede variar dependiendo del tipo de notificación (información, alarma o advertencia) y de las necesidades.

1.1.7. Configuración de notificaciones

La configuración de las preferencias en notificaciones se realiza a través de elementos *checkbox* de manera que el usuario solo tendrá que .

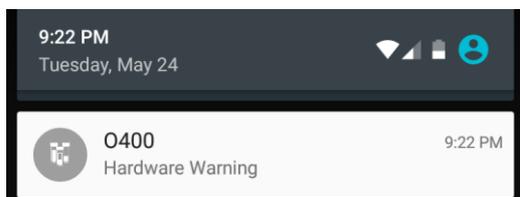


La notificaciones que en esta primera versión se permite configurar al usuario se corresponden con ser o no notificado de:

21. Alarma provenientes del analizador: error de hardware, etc.
22. Advertencias relacionadas con el analizador.
23. Desconexiones de los analizadores.
24. Notificaciones de los mensajes de advertencia sobre la imposibilidad de realizar más pruebas por falta de recursos.

1.1.8. Información mostrada en una notificación

Para las notificaciones se ha decidido informar de los eventos sin revelar información confidencial.



Por esto el mensaje de una notificación incluye:

25. El código del analizador.
26. Una definición genérica de la notificación.

4. Evaluación

Para la evaluación del producto se pretende que el usuario evaluador sea capaz de realizar las tareas planificadas de manera satisfactoria teniendo en cuenta:

- la velocidad con la que realiza cada tarea,
- la cantidad de errores que comete al realizar dichas tareas,
- el grado de satisfacción con las tareas ejecutadas.

De esta manera se pretende obtener información real de la interacción de los usuarios con la aplicación y evaluaremos si esta responde adecuadamente a sus necesidades; evaluando también si la presentación gráfica y la navegación resultan eficientes realizando anotaciones sobre las reacciones de los usuarios durante las pruebas.

Las tareas deben de enfocarse sobre los procesos de consulta de la información de los analizadores así como las notificaciones que generan. Para ello se comprobarán los siguientes puntos:

- La presentación gráfica es clara y ayuda al usuario a obtener la información.
- El proceso de consulta del estado e información de los analizadores del laboratorio es rápida y eficiente.
- Se accede de manera sencilla y efectiva a la configuración de un analizador, así como a la configuración de sus pruebas.
- La consulta del listado de trabajo de un analizador aporta la información necesaria.
- Las notificaciones mostradas ayudan a cerrar el ciclo de información relacionado con los analizadores del laboratorio.

La evaluación del Simulador de Analizadores queda fuera del alcance de este proyecto al entenderse que no se corresponde con el objetivo del mismo.

En esta fase del proyecto, el usuario solo puede evaluar el prototipo generado. Una vez finalizado el proyecto, el usuario podrá evaluarlo usando la propia aplicación móvil así como el simulador de analizadores.

5. Conclusiones y líneas de futuro

5.1 Conclusiones

Uno de los objetivos principales de este proyecto era conseguir una herramienta que permitiera la notificación de incidencias a dispositivos móviles mediante la tecnología *Push* y que pudiera ser integrada en un servidor de aplicaciones *Wildfly*. Este objetivo se ha cubierto proporcionándome el conocimiento suficiente para integrar esta herramienta con el sistema de aplicaciones que gestiona la información de un laboratorio clínico.

A nivel de desarrollo, este proyecto ha ampliado mis conocimientos relacionados con la tecnología *Push*, así como los relacionados con el desarrollo de aplicaciones Android.

A nivel de gestión de proyectos he podido constatar lo importante que es partir de una buena planificación y dedicación para poder llegar a cubrir cada una de las etapas de un proyecto con éxito. Gracias a ello, y a la comprensión de mi consultor en distintas fases del proyecto, he conseguido entregar un producto funcional que ha cubierto todos los objetivos inicialmente planteados para este proyecto. El hecho de haber marcado desde un principio las limitaciones del sistema a implementar no ha menguado en su complejidad y utilidad, cosa que permitirá en futuras versiones partir de un producto viable.

Uno de los puntos clave para la finalización del proyecto se corresponde con la metodología utilizada. El haber escogido una metodología que permite desglosar cada una de las etapas del proyecto en pequeñas tareas, algunas de ellas independientes entre sí, me ha permitido obtener un producto acorde con los requerimientos definidos en las primeras fases del mismo.

El haber tenido que desarrollar un simulador de analizadores, me ha obligado a estudiar y valorar con detalle cuál es la información más relevante a notificar sobre este tipo de equipos. La información escogida para este proyecto cubre las informaciones más relevantes y permite obtener una idea clara del potencial que ofrece.

5.2 Líneas de futuro

Durante el desarrollo de este proyecto han surgido muchas mejoras que no han sido incluidas por estar fuera del alcance del mismo, pero que sirven de base para las mejoras que en posteriores versiones se puedan realizar.

Los siguientes pasos para este proyecto serían:

1. Incorporar el sistema de notificaciones al sistema real de gestión de información de un laboratorio (LIS). De esta manera el LIS podrá realizar notificaciones de información relacionada con los analizadores de manera

transparente, ya que la implementación desarrollada permite exportar la base de datos y el servidor de aplicaciones de manera que pueda ser re-utilizada de la manera más conveniente.

2. Desarrollar la aplicación para otras plataformas o desarrollar una aplicación híbrida sería una posibilidad que dotaría al proyecto de mayor proyección y facilidad de mantenimiento.
3. Uso de la solución para informar de otro tipo de notificaciones relacionadas con el laboratorio y los analizadores. Un ejemplo sería la obtención de resultados críticos en los análisis realizados por el analizador, que dependiendo del país y la legislación vigente, requiere de un tratamiento y protocolo de actuación específico.
4. Mejoras en la aplicación móvil:
 - Poder recibir diferentes tipos de notificaciones y que en función del tipo de notificación, la aplicación te muestre una información u otra.
 - La posibilidad de dar por visualizadas las alarmas, advertencias e informaciones relacionadas con la notificación recibida.
 - Incluir más tipo de información referente al analizador que pueda ser consultada remotamente. Por ejemplo:
 - Estado de los componentes.
 - Estadísticas de trabajo.
 - Etc.

6. Glosario

Android	Sistema operativo de Google para dispositivos móviles
APK	Formato de archivo usado para empaquetar aplicaciones del sistema operativo Android.
D-Dimer	Dímero-D productos de degradación de la fibrina.
Diagrama de Gantt	Gráfico para describir actividades durante el tiempo de ejecución de un proyecto.
Factor VII	Factor de Coagulación VII.
INR	International Normalized Ratio, estandarización de los valores obtenidos a través del tiempo de protombina.
J2EE	Plataforma de programación para desarrollar y ejecutar aplicaciones escritas en el lenguaje de programación Java
Java	Lenguaje de programación orientado a objetos propiedad de Oracle.
Kanban	metodología ágil basada en un control de flujo de recursos en procesos de producción a través de tarjetas haciendo énfasis en la entrega justo a tiempo sin sobrecargar los miembros del equipo.
LIS	Sistema de Información del Laboratorio encargado de gestionar peticiones de pacientes, entrega de resultados, etc.
Metodología Ágil	es una serie de técnicas para la gestión de proyectos de manera eficiente y efectiva.
MySQL	Sistema de gestión de bases de datos.
Product Backlog	es el conjunto de todos los requisitos del proyecto y contiene descripciones genéricas de funcionalidades deseadas.
PT-INR	Tiempo de Protombina en INR.
Scrum	metodología ágil con estrategia de desarrollo incremental, solapamiento de diferentes fases de desarrollo para equipos auto organizados.
Servidor de Aplicaciones Wildfly	Servidor de aplicaciones de código abierto J2EE multiplataforma.
Sprint	Conjunto de tareas o historias que forman parte de la metodología ágil Scrum.
SGBD	Sistema Gestor de Base de Datos
Tecnología Push	forma de comunicación a través de internet en la que la petición de envío tiene origen en el servidor.

7. Bibliografía y referencias

Iconos	https://icons8.com/
Traducciones	http://www.linguee.es/espanol-ingles
Decisiones de diseño	https://blog.prototypr.io/5-ux-plays-to-start-your-design-on-good-tracks-45d00260909e#.eidd82510
GanttProject 2.7.2 Ostrava	https://www.ganttproject.biz/ostrava
Apple	http://www.apple.com
Office 2016	http://www.microsoftstore.com/
Gimp	https://www.gimp.org
Inkscape	https://inkscape.org/es/descargas
MySQL	https://www.mysql.com/products/community
Vertabelo	http://www.vertabelo.com
yEd	https://www.yworks.com/products/yed
Wildfly	http://wildfly.org/downloads
Aerogear	https://aerogear.org
Axure RP	http://www.axure.com
Trello	https://trello.com
Git	https://git-scm.com/downloads
Virtual Box	https://www.virtualbox.org
Ubuntu	http://www.ubuntu.com/download/alternative-downloads
Eclipse Mars	https://eclipse.org/mars

8. Anexo 1 - Entrevistas

Esta entrevista resume las entrevistas realizadas a dos Técnicos Especialistas en Implantaciones. Sus conocimientos a lo largo de los años, valoraciones, apreciaciones y opiniones respecto al trabajo diario realizado por un facultativo de laboratorio.

Entrevista	
Pregunta	Respuesta
¿Cuál es el trabajo diario de un facultativo dentro del laboratorio?	<ul style="list-style-type: none"> • Son responsables de la configuración y realización de pruebas sobre los analizadores del laboratorio. Esto quiere decir, preparar los reactivos a utilizar durante las sesiones de trabajo y configurar los analizadores para que estos realicen las pruebas teniendo en cuenta: <ul style="list-style-type: none"> ○ Peticiones del día. ○ Estado de los analizadores. ○ Niveles de reactivos de los analizadores. ○ Prioridad de las peticiones marcadas como urgentes.
¿Qué posibilidades de utilizar tecnología móvil tienen?	<ul style="list-style-type: none"> • Actualmente todo el mundo lleva consigo su teléfono móvil, ya sea el personal o de empresa. Por ahora no hay restricciones en los laboratorios sobre su uso.
¿Qué os suelen solicitar con respecto a poder utilizar con tecnología móvil?	<ul style="list-style-type: none"> • Al estar en continuo movimiento dentro del laboratorio, nos suelen comentar que sería interesante poder consultar el estado de un analizador en concreto a través del móvil o Tablet.
¿Qué información consideran relevante?	<ul style="list-style-type: none"> • Para ellos lo más importante sería: <ul style="list-style-type: none"> ○ Detectar cuando un analizador se ha desconectado. ○ Qué está haciendo un analizador en un momento determinado y cuánto tiempo resta para que termine. ○ Saber qué pruebas tiene "en cola" el analizador para saber así cuanto tiempo falta para que termine un determinado trabajo y tengan que ir a configurarlo de nuevo. ○ Saber el nivel de reactivos de cada una de las pruebas configuradas en el analizador. Por si tienen que remplazar los reactivos. ○ Poder ser notificados directamente a su móvil de cualquier incidencia o cambio de estado en cualquiera de los analizadores: desconexiones, alarmas, cuando se acaben de realizar las pruebas de una lista de trabajo, etc.
¿Qué funcionalidades considerarías interesantes de poder realizar?	<ul style="list-style-type: none"> • Todo lo comentado en la pregunta anterior sería un muy buen comienzo. Quizás de las notificaciones les sería más útil si en vez de notificarles cuando se finaliza la lista de trabajo, se les notificara cuando se está realizando la última de las pruebas. De esta manera podrían ir

	<p>preparando el trabajo a realizar por el analizador en las siguientes horas.</p> <ul style="list-style-type: none"> • También nos comentaron que sería interesante poder “desactivar” un analizador en función de si ven que los niveles de reactivos están próximos a agotarse.
De las funcionalidades mencionadas, ¿cuáles consideráis no necesarias?	<ul style="list-style-type: none"> • Quizás la de “desactivar” un analizador. Aunque es interesante, implica mucho más trabajo en temas de seguridad, registro de quién ha parado un analizador y porqué. Además de necesitar registrar toda esa información por temas legales y de consultoría.
¿Creéis que el aspecto visual de la aplicación debe de estar acorde con el aspecto actual de la aplicación que gestiona el analizador?	<ul style="list-style-type: none"> • Con el aspecto actual no, ya tiene sus años, quizás con el aspecto que va a tener la nueva versión, mucho más moderna acorde con los nuevos estilos. Basándote en las guías de estilo estándar de las aplicaciones actuales podría quedarte una aplicación muy atractiva.
¿Qué otras funcionalidades se os ocurren que pudieran ser incluidas a futuro?	<ul style="list-style-type: none"> • Pues explotar el tema de las notificaciones al máximo: <ul style="list-style-type: none"> ○ Informar de cualquiera de los errores o alarmas (que veo que no están todas las que se pueden generar) de los analizadores. ○ Quién ha solucionado determinado error y cuando. ○ Escoger de qué zona del laboratorio quieres estar informado mediante notificaciones. ○ Escoger que se te informe cuando ha finalizado una determinada prueba, porque es una prueba urgente y no de rutina. • También sería interesante poder buscar analizadores si la lista es elevada. Estamos pensando en laboratorios que son multicentro, que pueden controlar o gestionar analizadores de distintos laboratorios repartidos por el hospital (incluso de diferentes hospitales). • Buscar de un tipo de prueba cómo están los reactivos de todos los analizadores también sería interesante. • Ver el resultado de las pruebas que se han realizado, no solo conocer cuales han finalizado. Filtrarlas por pruebas de rutina o especiales también sería interesante. • O incluso modificar el orden en el que se ejecutan las pruebas dentro de la lista de trabajo, por si una pasa a ser prioritaria por algún motivo. • Otra posibilidad podría ser la de incluir todo lo relacionado con los controles de calidad. Quien sabe, se podrían ver incluso las gráficas de resultados y los propios resultados. • No estaría mal tampoco consultar históricos de resultados, pero quizás esto último sería más interesante en una Tablet en vez de un móvil, ¿no?
Muchas gracias por vuestro tiempo. Tomaré muy en cuenta vuestras opiniones.	<ul style="list-style-type: none"> • De nada y mucha suerte con el proyecto.

9. Anexo 2 – Guía de instalación y uso del sistema

En los siguientes puntos se describen diferentes posibilidades para hacer uso e instalar el sistema de notificaciones, el simulador de analizadores y la aplicación móvil.

En todas las posibilidades propuestas, los requisitos mínimos son los siguientes:

- Java (versión 1.7.0_79 en adelante).
- Simulador de Analizadores → **AnalyzerSimulator.jar**
- Aplicación móvil Android → **AnalyzerHandler.apk**

Los usuarios y contraseñas de las distintas aplicaciones utilizadas en el sistema son los siguientes:

Aplicación	Usuario	Contraseña
MySQL	root	
MySQL BBDD Simulador	tfm_user	tfm_user
Servidor Ubuntu	tfm-uoc	uoc
Servidor AeroGear	admin	uoc
Aplicación Android	user	123

9.1 Instalación del Servidor de Aplicaciones y Base de datos

9.1.1 Requisitos

Los requisitos mínimos para hacer uso de la máquina virtual son:

- Aplicación Virtual Box 5.0.20 descargable desde <https://www.virtualbox.org/>
- Poder dotar a la máquina de al menos 2048 de memoria.
- Disponer de una imagen ISO de un sistema operativo Linux¹.
- Servidor de aplicaciones Wildfly 8.2.0.
- Servidor de base de datos MySQL 5.5.
- Los ficheros de exportación de las diferentes bases de datos:
 - AnalyzerSimulatorDB.sql
 - keycloak.sql

¹ En esta guía la instalación se realiza sobre el sistema operativo Ubuntu 14.04 LTS descargable desde <http://www.ubuntu.com/download/alternative-downloads>

- unifiedpush.sql

9.1.2 Instalación Sistema Operativo Linux

En los siguientes pasos se muestra la configuración que se ha de utilizar en la creación de la máquina virtual Linux, con la imagen ISO descargada, para después realizar la instalación de las diferentes aplicaciones y servidores (aplicaciones y bases de datos) que se utilizarán.

- Para ello primeramente realizaremos la creación e instalación de la máquina virtual Linux con la misma configuración utilizada en el apartado anterior.



Ilustración 14 – VirtualBox: Configuración máquina virtual

- Creamos un nuevo fichero de disco virtual con un tamaño mínimo de 8Gb.

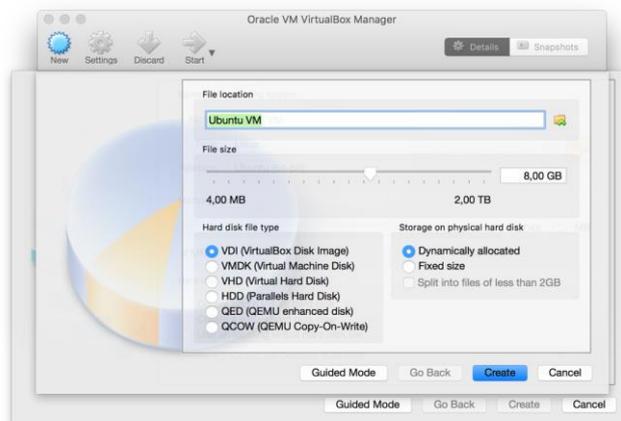


Ilustración 15 – VirtualBox: Creación disco duro

- Modificamos los parámetros de pantalla y tarjetas de red de la misma manera que en la máquina virtual descrita en el apartado anterior:

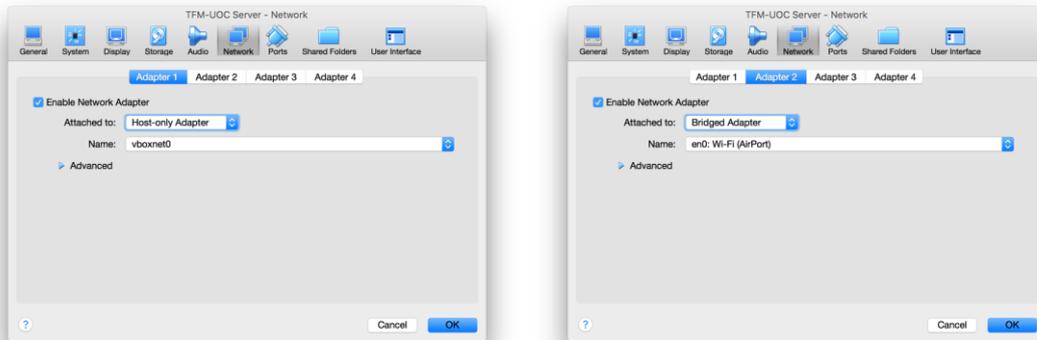


Ilustración 16 – VirtualBox: Configuración de las tarjetas de red

- Habilitar la aceleración 3D.

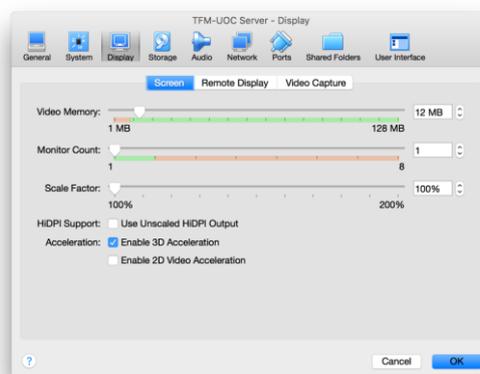


Ilustración 17 – VirtualBox: Configuración aceleración 3D

- Escogemos el fichero ISO del sistema operativo Linux.

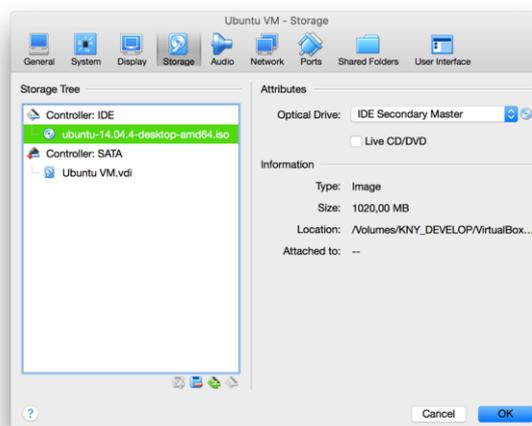


Ilustración 18 – VirtualBox: Selección imagen ISO

Realizamos la instalación del sistema operativo guardándonos el **nombre de usuario** y la **contraseña** registrados para ser utilizados en los siguientes pasos de la instalación.

9.1.3 Instalación Software

En los siguientes pasos se muestra, de manera guiada, el orden en el que se han de instalar y configurar los diferentes servidores y aplicaciones. Es importante seguir el orden para asegurar el correcto funcionamiento de las aplicaciones.

Igualmente es importante usar los usuarios y contraseñas dados para las diferentes bases de datos.

Sugerencia: es recomendable usar el comando **sudo -s** para asegurarnos de que estamos ejecutando como **root** todos los comandos y evitar problemas al realizar cualquiera de las acciones. En caso de no usarlo, se deberá de poner delante de cada línea el comando **sudo**.

1. Instalar Java 1.8

Para la instalación de la versión de Java 1.8 se deben de utilizar los siguientes comandos:

```
> apt-get install python-software-properties
> add-apt-repository ppa:webupd8team/java
> apt-get update
> apt-get install oracle-java8-installer
```

2. Instalar servidor Wildfly 8.2.0 Final

El primer paso es crear el usuario **wildfly** a utilizar por el servidor.

```
> cd /opt
> adduser --no-create-home --disabled-password --disabled-login wildfly
Adding user `wildfly' ...
Adding new group `wildfly' (1002) ...
Adding new user `wildfly' (1002) with group `wildfly' ...
Not creating home directory `/home/wildfly'.
Changing the user information for wildfly
Enter the new value, or press ENTER for the default
  Full Name []: Wildfly Server
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
```

Descargamos y descomprimos el servidor:

```
> wget http://download.jboss.org/wildfly/8.2.0.Final/wildfly-8.2.0.Final.tar.gz
> tar -xzf wildfly-8.2.0.Final.tar.gz
```

Configuramos un enlace simbólico (wildfly) a la carpeta del servidor.

```
> ln -s wildfly-8.2.0.Final wildfly
> chown -R wildfly:wildfly wildfly
> chown -R wildfly:wildfly wildfly-8.2.0.Final
```

Modificamos el archivo de configuración **wildfly.conf** para que se ejecute con los siguientes parámetros:

```
> cd /opt/wildfly/bin/init.d
> vi wildfly.conf

# General configuration for the init.d scripts,
# not necessarily for JBoss AS itself.
# default location: /etc/default/wildfly

## Location of JDK
# JAVA_HOME="/usr/lib/jvm/default-java"

## Location of WildFly
JBOSS_HOME="/opt/wildfly"

## The username who should own the process.
JBOSS_USER=wildfly

## The mode WildFly should start, standalone or domain
JBOSS_MODE=standalone

## Configuration for standalone mode
JBOSS_CONFIG=standalone-full.xml

## Configuration for domain mode
# JBOSS_DOMAIN_CONFIG=domain.xml
# JBOSS_HOST_CONFIG=host-master.xml

## The amount of time to wait for startup
STARTUP_WAIT=120

## The amount of time to wait for shutdown
SHUTDOWN_WAIT=120

## Location to keep the console log
JBOSS_CONSOLE_LOG="/var/log/wildfly/console.log"

:wq
```

Configuramos el servicio del servidor *Wildfly*:

```
> cd /etc/default
> ln -s /opt/wildfly/bin/init.d/wildfly.conf wildfly
> cd /etc/init.d
> ln -s /opt/wildfly/bin/init.d/wildfly-init-debian.sh wildfly
```

Modificamos el fichero de configuración para que el servidor escuche desde cualquier puerto:

```
> cd /opt/wildfly/standalone/configuration
> vi standalone-full.xml
:%s/127.0.0.1/0.0.0.0/g
:wq
```

Añadimos un usuario administrador para poder conectarnos a la consola de administración del servidor *Wildfly*.

```
> cd /opt/wildfly/bin
> ./add-user.sh
```

Nota: es importante guardarse el usuario y contraseña utilizados. Por ejemplo:

- **Usuario:** admin-uoc
- **Contraseña:** uoc

3. Instalar y configurar el servidor de base de datos MySQL 5.5

Para descargarnos el servidor de base de datos *MySQL* 5.5 ejecutaremos el siguiente comando:

```
> apt-get install mysql-server
```

Durante la instalación nos preguntará varias veces de cambiarle la contraseña al usuario root; no es necesario, se puede dejar el usuario sin contraseña, pero si se decide ponerle una contraseña esta tendrá que ser utilizada en los siguientes pasos.

Una vez instalado el servidor de base de datos es necesario realizar modificaciones en el archivo de configuración para que acepte conexiones desde cualquier dirección IP y no resuelva nombres cuando se realice una conexión.

Para ello tendremos que modificar el siguiente archivo modificando y añadiendo los siguientes parámetros:

```
> vi /etc/mysql/my.cnf

...
[mysqld]
#
# * Basic Settings
#
user      = mysql
pid-file  = /var/run/mysqld/mysqld.pid
socket    = /var/run/mysqld/mysqld.sock
port      = 3306
basedir   = /usr
datadir   = /var/lib/mysql
tmpdir    = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
skip-name-resolve

#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 0.0.0.0
...
```

Una vez realizado el cambio será necesario reiniciar el servicio **mysql**.

```
> service mysql restart

mysql stop/waiting
mysql start/running, process 27065
```

4. Descargar e instalar el servidor de aplicaciones AeroGear Push 1.1.2

Para poder utilizar el servidor de aplicaciones de notificaciones, es necesario descargarse tanto el código fuente como los WAR que tendremos que copiar dentro de la carpeta deployments de nuestro servidor Wildfly².

Para ello utilizaremos los siguientes comandos para obtener dichos archivos:

```
> cd /home/<user_name>/Downloads

> wget https://github.com/aerogear/aerogear-unifiedpush-server/releases/download/1.1.2.Final/aerogear-unifiedpush-server-1.1.2-dist.tar.gz

> wget https://github.com/aerogear/aerogear-unifiedpush-server/archive/1.1.2.Final.tar.gz
```

Una vez descargados descomprimiremos dichos archivos:

² Los siguientes pasos es una versión resumida de la guía de instalación de AeroGear disponible en https://aerogear.org/docs/unifiedpush/ups_userguide/index/

```
> tar -xvzf aerogear-unifiedpush-server-1.1.2-dist.tar.gz
> tar -xvzf aerogear-unifiedpush-server-1.1.2.Final.tar.gz
```

Importante: Para los siguientes pasos, es necesario que el servidor de aplicaciones *Wildfly* esté parado.

El siguiente punto de la instalación requiere tener instalado Maven en la máquina virtual. En caso de no tenerlo, se puede obtener mediante el siguiente comando:

```
> apt-get install maven
```

Una vez instalado continuaremos con la instalación del conector de *MySQL* en el servidor de aplicaciones haciendo uso de las siguientes instrucciones:

```
> cd aerogear-unifiedpush-server-1.1.2.Final/
> mvn dependency:copy -Dartifact=mysql:mysql-connector-java:5.1.18 -
DoutputDirectory=/opt/wildfly/modules/com/mysql/jdbc/main/
```

En este punto es importante comprobar que el archivo **module.xml** ha sido creado dentro de la carpeta junto con el JAR del conector:

/opt/wildfly/modules/com/mysql/jdbc/main

En caso de no existir se deberá de crear manualmente dentro de dicha carpeta con el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
JBoss, Home of Professional Open Source
Copyright Red Hat, Inc., and individual contributors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<module xmlns="urn:jboss:module:1.0" name="com.mysql.jdbc">
  <resources>
    <resource-root path="mysql-connector-java-5.1.18.jar"/>
    <!-- Insert resources here -->
  </resources>
```

```

<dependencies>
  <module name="java-apt"/>
  <module name="java-transaction-apt"/>
</dependencies>
</module>

```

Una vez configurado procederemos a la creación, configuración e importación de las diferentes bases de datos.

5. Creación, configuración e importación de las bases de datos

En los siguientes puntos realizaremos la creación de las diferentes bases de datos, usuarios y lo enlazaremos con el servidor de aplicaciones *Wildfly*.

Los siguientes pasos se deberán de realizar con el usuario **root** de *MySQL*, en caso de haber establecido una contraseña este es el momento de utilizarla:

```

> mysql -u root

mysql> create database unifiedpush default character set = "UTF8" default collate = "utf8_general_ci";
mysql> create database keycloak default character set = "UTF8" default collate = "utf8_general_ci";
mysql> create user 'unifiedpush'@'%' identified by 'unifiedpush';
mysql> GRANT SELECT,INSERT,UPDATE,ALTER,DELETE,CREATE,DROP,INDEX ON unifiedpush.* TO 'unifiedpush'@'%';
mysql> GRANT SELECT,INSERT,UPDATE,ALTER,DELETE,CREATE,DROP ON keycloak.* TO 'unifiedpush'@'%';
mysql> GRANT SELECT,INSERT,UPDATE,ALTER,DELETE,CREATE,DROP,INDEX ON unifiedpush100li.* TO 'unifiedpush'@'%';

mysql> create database analyzer_simulator default character set = "UTF8" default collate = "utf8_general_ci";
mysql> create user 'tfm_user'@'%' identified by 'tfm_user';
mysql> GRANT SELECT,INSERT,UPDATE,ALTER,DELETE,CREATE,DROP,INDEX ON analyzer_simulator.* TO 'tfm_user'@'%';

```

Para el siguiente punto debemos asegurarnos de que el servidor de aplicaciones *Wildfly* está en funcionamiento y ejecutaremos el comando que permite crear los descriptores que enlazan el servidor de aplicaciones y las bases de datos. Para lo cual haremos uso de las siguientes instrucciones:

```

> service wildfly start

* Starting WildFly Application Server wildfly
* WildFly Application Server (already running)

> cd /opt/wildfly/bin/

> ./jboss-cli.sh --file=/home/rubkrmon/Downloads/aerogear-unifiedpush-server-1.1.2/databases/mysql-database-config-wildfly.cli

```

Si la ejecución ha sido satisfactoria, veremos que en el archivo **standalone-full.xml** existen sendas referencias a las conexiones a bases de datos **unifiedpush** y **keycloak**.

```

...
<datasource jndi-name="java:jboss/datasources/UnifiedPushDS" pool-name="UnifiedPushDS" enabled="true" use-ccm="false">
...
<datasource jndi-name="java:jboss/datasources/KeycloakDS" pool-name="KeycloakDS" enabled="true" use-ccm="false">
...

```

El siguiente punto es preparar el esquema de las bases de datos, para ello ejecutaremos los siguientes comandos:

```
> cd /home/<user_name>/Downloads/aerogear-unifiedpush-server-1.1.2/migrator
> unzip ups-migrator-dist.zip
> cd ups-migrator
> cp liquibase-mysql-example.properties liquibase.properties
> ./bin/ups-migrator update
```

Una vez finalizado el proceso, deberemos detener el servidor de aplicaciones *Wildfly* para copiar las propias aplicaciones en la carpeta deployments del servidor mediante las siguientes instrucciones:

```
> service wildfly stop
> cp /home/<user_name>/Downloads/aerogear-unifiedpush-server-1.1.2/servers/unifiedpush-auth-server.war /opt/wildfly/standalone/deployments/
> cp /home/<user_name>/Downloads/aerogear-unifiedpush-server-1.1.2/servers/unifiedpush-server-wildfly.war /opt/wildfly/standalone/deployments/
```

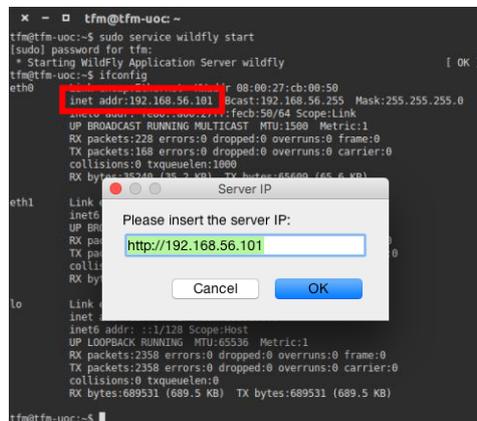
Para finalizar, realizaremos la importación de las diferentes bases de datos. Es importante tener localizados los archivos correspondientes a cada una de las bases de datos a importar. Una vez finalizada la importación podremos ejecutar el servidor de aplicaciones *Wildfly* y comenzar a utilizar las diferentes aplicaciones.

```
> mysql -u root -p analyzer_simulator < /home/<user_name>/Downloads/AnalyzerSimulatorDB.sql
> mysql -u root -p keycloak < /home/<user_name>/Downloads/keycloak.sql
> mysql -u root -p unifiedpush < /home/<user_name>/Downloads/unifiedpush.sql
> service wildfly start
```

9.2 Uso del Simulador de Analizadores

Para ejecutar el Simulador de Analizadores simplemente se ha de ejecutar el archivo AnalyzerSimulator.jar

La primera de las opciones que nos solicitará es la dirección IP del servidor virtual. Esta dirección IP la obtendremos directamente de la máquina virtual a través del Terminal usando el comando **ifconfig**.



9.2.1 Pantalla Principal

La pantalla principal del Simulador de Analizadores se divide en las siguientes secciones:

- Lista de analizadores disponibles.
- Cabecera de detalle del analizador escogido.
- Configuración de listas de trabajo, notificaciones y pruebas del analizador escogido.

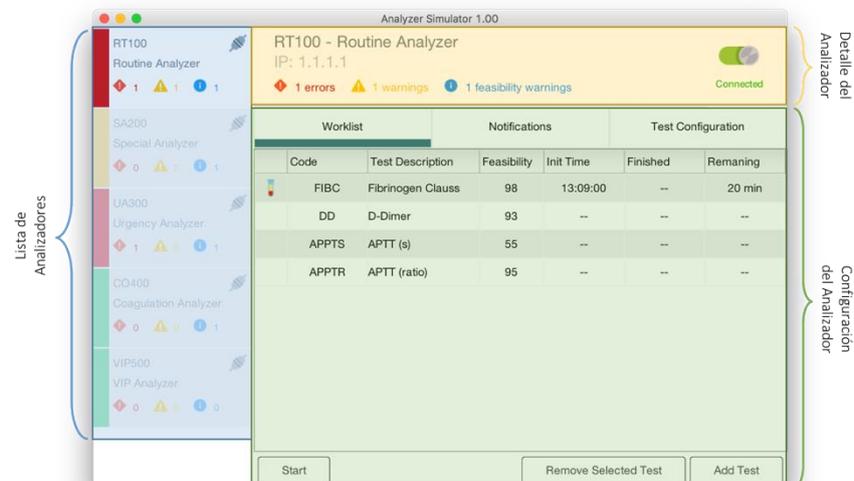


Ilustración 19 – Aplicación Simulador Analizadores (AnSim)

Pantalla Principal – Cabecera

En la cabecera de detalle se muestra la información relativa al Analizador resumiendo el número de notificaciones relativas a alarmas y avisos del analizador y de pruebas.

En esta cabecera se puede forzar la notificación de conexión y desconexión del analizador a través del check



Ilustración 20 – AnSim: Detalle cabecera

Pantalla Principal – Configuración

Pestaña Worklist

Las funcionalidades disponibles a través de esta pestaña de configuración se engloban en 2 tipos.

Con los botones **Remove Selected Test** y **Add Test** se permite modificar la lista de trabajo (si esta no está en ejecución), quitando de la lista la prueba seleccionada o añadiendo alguna de las pruebas que tenga configuradas el analizador (y que no se encuentre ya en la lista de pruebas a realizar).

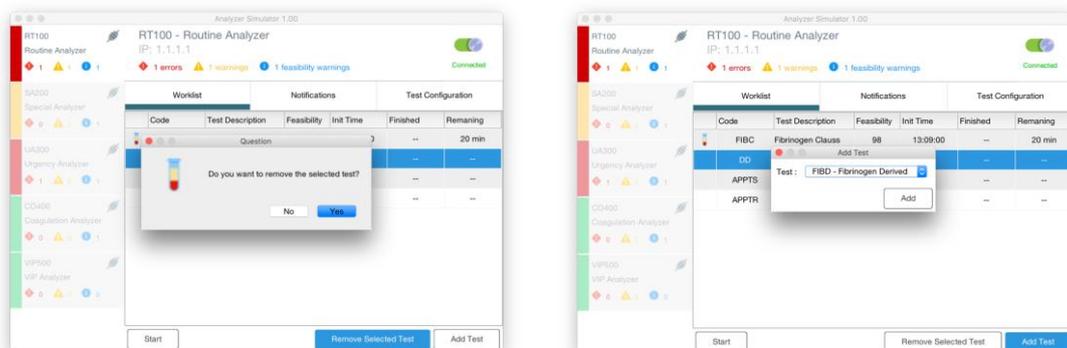


Ilustración 21 – AnSim: Pestaña Worklist

Iniciar la simulación de la ejecución de pruebas de la lista de trabajo o reiniciarla a través del botón **Start** y **Reset**.

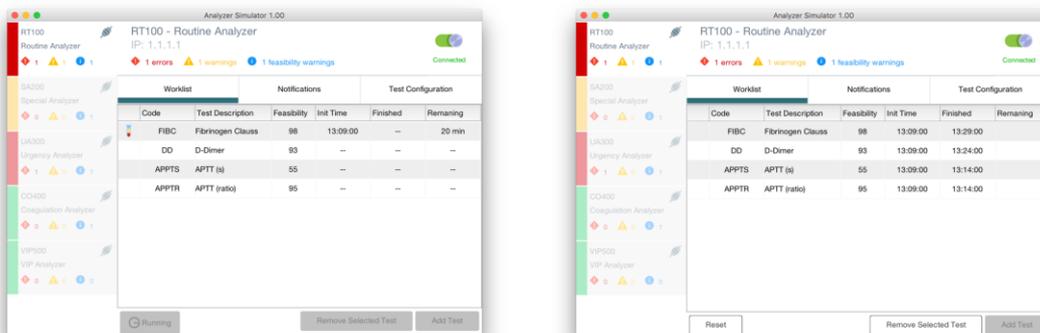


Ilustración 22 – AnSim: Pestaña Worklist iniciar lista de trabajo

El tiempo de ejecución de las pruebas simuladas está establecida en segundos aunque se muestren minutos.

Pestaña Notificaciones

En la pestaña de notificaciones se muestran todas las notificaciones generadas para el analizador. Las acciones que se pueden realizar desde esta pestaña son las de generar notificaciones de alarmas, avisos y avisos de viabilidad de pruebas.

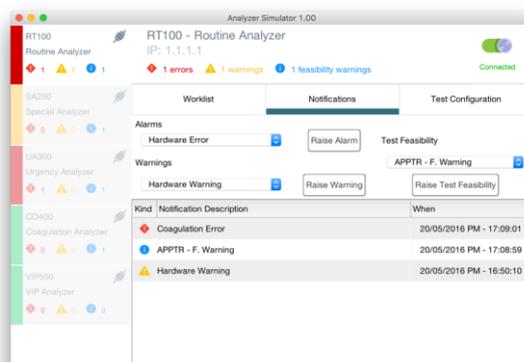


Ilustración 23 – AnSim: Pestaña Notificaciones

Pestaña Configuración de Pruebas

En la pestaña de configuración de pruebas se muestran todas las pruebas configuradas para el analizador pudiendo crear de nuevas a través del botón **Create Test**.

La información mostrada de las pruebas se corresponde con:

- Código identificador de la prueba.
- Descripción de la prueba.
- Tiempo necesario para la ejecución de la prueba.
- Número de pruebas que se pueden realizar.
- Límite en el que se debe de informar y bloquear la ejecución de dicha prueba.

En el momento de crear una prueba, aparte de la información listada, se han de definir las unidades en las que trabaja dicha prueba.

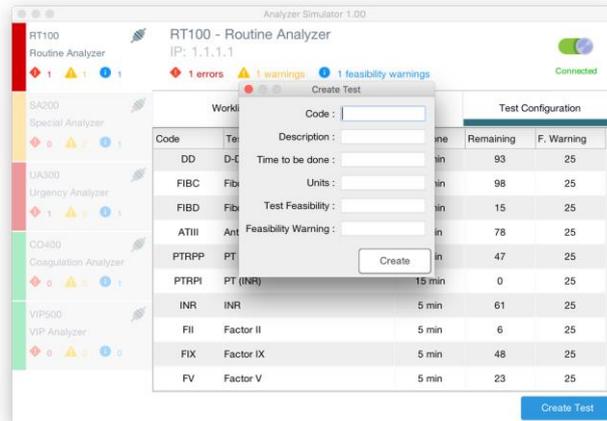


Ilustración 24 – AnSim Pestaña de configuración de pruebas

9.3 Uso de la Aplicación Móvil

Para ejecutar la aplicación móvil simplemente se ha de instalar en un dispositivo móvil. En caso de querer utilizar un simulador Android este no ha de estar ejecutándose sobre la misma máquina que el servidor de notificaciones o utilizar un gestor de máquinas virtuales Android del estilo **Genymotion** que permite la ejecución de diferentes máquinas virtuales sin entrar en conflicto.

Es importante remarcar que si se decide utilizar una máquina virtual para el simulador del dispositivo Android es necesario que ésta tenga instaladas las Google Apps. Se pueden obtener para instalar desde la dirección:

<http://www.androidrootz.com/2015/03/download-android-51x-lollipop-pa-gapps.html>

A continuación se muestra un flujo de trabajo con la aplicación que permite conocer cada una de las pantallas en detalle y las funcionalidades de las mismas. Es recomendable realizar la consulta de las diferentes pantallas utilizando en paralelo la aplicación simulador de analizadores para obtener una experiencia de usuario más completa y satisfactoria.

9.3.1 Login

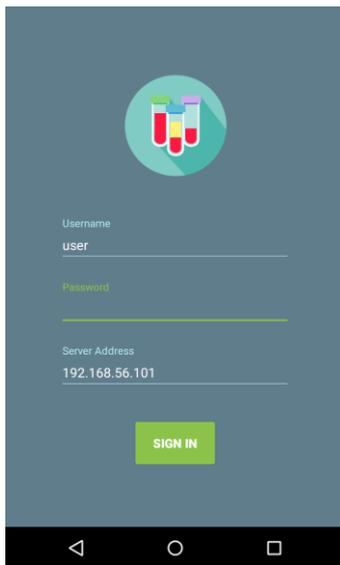


Ilustración 25 – App Móvil: Login

Utilizar la dirección IP del servidor de aplicaciones para realizar la conexión.

El usuario y contraseña a utilizar son user/123

9.3.2 Lista de Analizadores

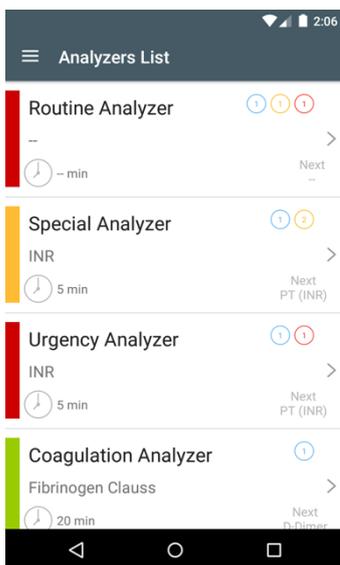


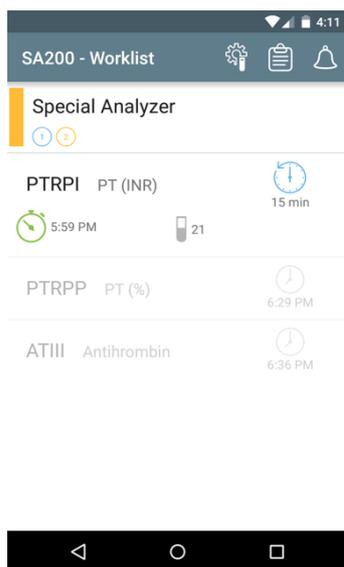
Ilustración 26 – App Móvil: Lista de analizadores

En la lista de analizadores se muestra:

- El estado de los analizadores; en caso de tener una alarma o una advertencia la barra de estado de cada uno de los analizadores varía entre verde, rojo o amarillo.
- A través de círculos se muestra el número de notificaciones recibidas agrupadas en alarmas, advertencias e información.
- En caso de estar ejecutando una lista de trabajo se muestra:
 - Código de la prueba que se está realizando.
 - Tiempo restante para finalizar la prueba.
 - En caso de tener, cuál es la siguiente prueba a realizar.

Esta lista es actualizable a través del gesto de *pull*down.

9.3.3 Lista de Trabajo



**Ilustración 27 – App Móvil:
Lista de trabajo**

En la lista de trabajo se muestra:

- El estado de los analizadores; en caso de tener una alarma o una advertencia la barra de estado de cada uno de los analizadores varía entre verde, rojo o amarillo.
- A través de círculos se muestra el número de notificaciones recibidas agrupadas en alarmas, advertencias e información.
- En caso de estar ejecutando una lista de trabajo se muestra:
 - Código y descripción de la prueba que se está ejecutando.
 - Tiempo restante para finalizar la prueba.
 - El número de pruebas restantes que puede realizar.
 - En caso de tener, cuales son las siguientes pruebas que quedan por realizar.

Esta lista es actualizable a través del gesto de pulldown.

9.3.4 Lista de Trabajo – Completa



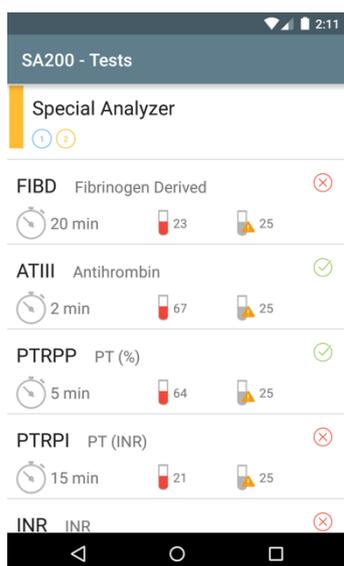
**Ilustración 28 – App Móvil:
Lista de trabajo – Completa**

En la lista de trabajo completa se muestra:

- El estado de los analizadores; en caso de tener una alarma o una advertencia la barra de estado de cada uno de los analizadores varía entre verde, rojo o amarillo.
- A través de círculos se muestra el número de notificaciones recibidas agrupadas en alarmas, advertencias e información.
- En caso de estar ejecutando una lista de trabajo se muestra:
 - Las pruebas que ya se han realizado, si se han realizado correctamente y cuando terminaron.
 - Código y descripción de la prueba que se está ejecutando.
 - Tiempo restante para finalizar la prueba.
 - El número de pruebas restantes que puede realizar.
 - En caso de tener, cuales son las siguientes pruebas que quedan por realizar.

Esta lista es actualizable a través del gesto de *pull*down.

9.3.5 Configuración de pruebas del Analizador



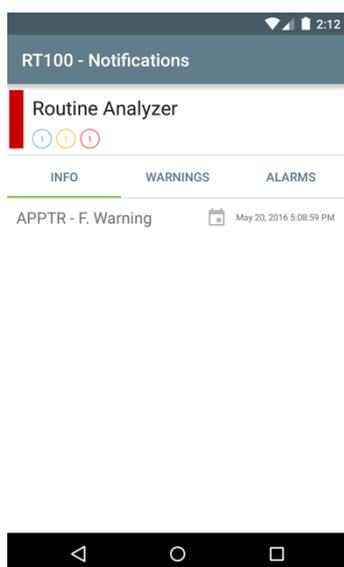
**Ilustración 29 – App Móvil:
Pruebas configuradas**

En la lista de pruebas configuradas en el analizador se muestra:

- El estado de los analizadores; en caso de tener una alarma o una advertencia la barra de estado de cada uno de los analizadores varía entre verde, rojo o amarillo.
- A través de círculos se muestra el número de notificaciones recibidas agrupadas en alarmas, advertencias e información.
- Un listado de las pruebas configuradas en el Analizador.
 - Código y descripción de la prueba.
 - Tiempo restante para finalizar la prueba.
 - El número de pruebas restantes que puede realizar.
 - El mínimo límite de pruebas que evita que se pueda realizar la prueba en el analizador.
 - En base al punto anterior, una marca de si se puede realizar la prueba o no en el analizador.

Esta lista es actualizable a través del gesto de *pull*down.

9.3.6 Notificaciones del Analizador



**Ilustración 30 – App Móvil:
Notificaciones del Analizador**

En la lista de notificaciones se muestran (en diferentes pestañas) las notificaciones realizadas por el instrumento agrupadas en: informativas, advertencias y alarmas. Mostrándose de cada una de ellas

- Descripción de la notificación.
- La fecha de cuando fueron informadas.

Estas listas son actualizables a través del gesto de *pull*down.

9.3.7 Menú de opciones

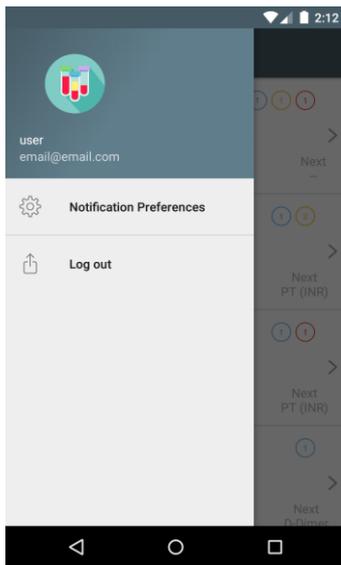


Ilustración 31 – App Móvil: Menú de opciones

Desde la pantalla principal se puede acceder al menú de opciones del usuario donde:

- Accede a la configuración de las notificaciones que quiere recibir.
- Puede salir de la aplicación.

9.3.8 Configuración de Notificaciones

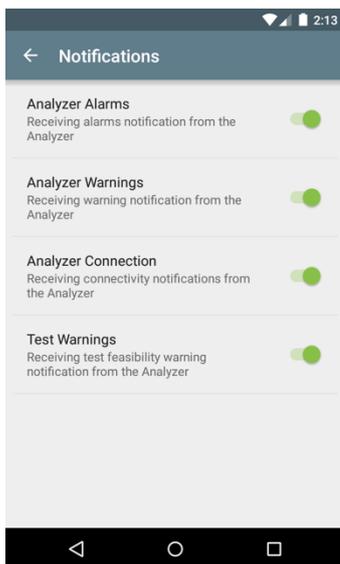
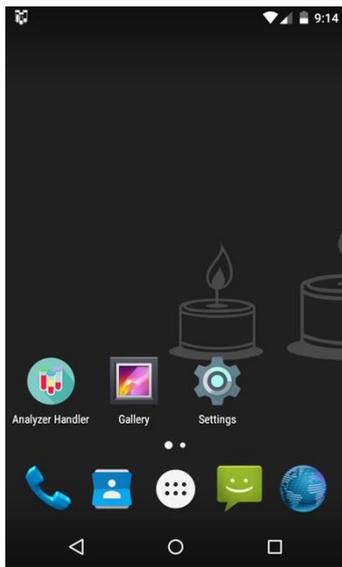


Ilustración 32 – App Móvil: Configuración de notificaciones

En la lista de configuración de notificaciones se muestra:

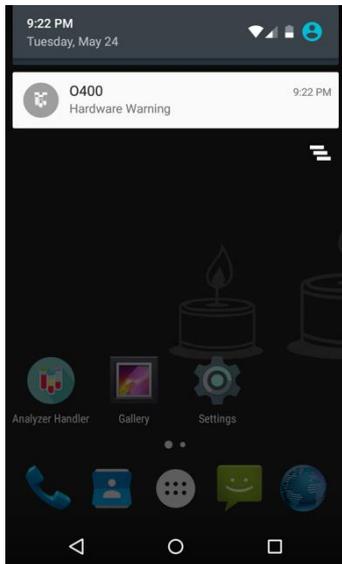
- El tipo de notificación.
- Está activa o no para el usuario.

9.3.9 Notificación de un Evento



En las notificaciones de eventos se muestra:

- El código del analizador.
- El mensaje referente a la notificación mostrada.



**Ilustración 33– App Móvil:
Notificaciones en la pantalla
principal del móvil**