



Soporte Móvil para la Monitorización de la Integración de Sistemas Intensivos Software

Alejandro Morán Jiménez:

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Eduard Martín Lineros:

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

08/06/2016



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Soporte Móvil para la Monitorización de la Integración de Sistemas Intensivos Software</i>
Nombre del autor:	<i>Alejandro Morán Jiménez</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>Máximo 3 palabras clave, validadas por el director del trabajo (dadas por los estudiantes o en base a listados, tesauros, etc.)</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
<p>En la actualidad, el desarrollo de software se ha convertido en una compleja tarea que requiere la participación de muchos componentes externos, tanto humanos como otros sistemas informáticos. Estos problemas requieren el uso de sofisticadas herramientas de gestión de diversa índole, como la gestión de recursos humanos, validaciones de estándares, gestión de la calidad, pruebas unitarias o integración,</p> <p>Desarrollar estos sistemas software puede conllevar la intervención de múltiples equipos, de distintas compañías o incluso de distintos países. Además, el mercado actual no proporciona las suficientes herramientas para suplir esta necesidad de gestión de múltiples elementos del desarrollo de software, y las utilidades existentes para este fin presentan múltiples carencias.</p> <p>En particular, PositioningSystem es una herramienta de especial ayuda y apoyo en la construcción de grandes proyectos de desarrollo, donde participan diferentes empresas y/o socios, y que pueden incluso estar distribuidos geográficamente.</p> <p>El objetivo de este Trabajo de Fin de Máster es el de desarrollar una aplicación para dispositivos Android que cubra una de las carencias de Positioning System, la nula adaptación a plataformas móviles. De esta manera se podrán llevar a cabo las tareas de gestión de forma ubicua necesitando únicamente una conexión a internet. Estas tareas comprenden la creación de proyectos, familias de productos, subfamilias de productos y productos finales, que podrán ser modificados para reflejar su estado de integración final y proporcionar así</p>	

gráficos para comprender la marcha de uno o múltiples proyectos.

Abstract (in English, 250 words or less):

Today, software development has become a complex process that requires the participation of many human and external components such as other computer systems. These problems require the use of sophisticated management tools of various types, such as human resources management, validation standards, quality management, unit testing or integration,

Developing these software systems can lead to the involvement of multiple teams from different companies or even different countries. In addition, the current market does not provide sufficient tools to cover this need for managing multiple elements of software development, and existing utilities for this purpose have multiple deficiencies.

In particular, PositioningSystem is a tool of special help and support in the construction of major development projects, where different companies and partners are involved, and may even be distributed geographically.

The aim of this thesis is to develop an application for Android devices covering one of the deficiencies in Positioning System, the null adaptation to mobile platforms. This way out are to be taken management tasks ubiquitously requiring only an Internet connection. These tasks include creating projects, product families, subfamilies and final products, which may be modified to reflect their status and final integration and provide graphics to understand the progress of one or multiple projects.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	2
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	4
1.5 Breve resumen de productos obtenidos.....	1
1.6 Breve descripción de los otros capítulos de la memoria.....	1
2. Diseño Técnico.....	3
2.1 Usuarios y Contextos de Uso.....	3
2.2 Diseño Conceptual.....	4
2.3 Prototipado.....	11
2.4 Casos de Uso.....	13
2.5 Modelo de Datos.....	15
2.6 Arquitectura del Sistema.....	1
2.7 Desarrollo.....	2
3. Conclusiones.....	1
4. Glosario.....	2
5. Bibliografía.....	3
6. Anexos.....	4

Lista de figures

Figura 1 - Arquitectura del Sistema	2
Figura 2 - Planificación Temporal.....	1
Figura 3 - Caso de Uso: Gestión de Proyectos	13
Figura 4 - Caso de Uso: Gestión de Familias, Subfamilias y Productos	14
Figura 5 - Modelo de Datos	1
Figura 6 - Interfaz del Servicio Web	1
Figura 7 - EazeGraph Bar Chart.....	3
Figura 8 - EazeGraph Stacked Bar Chart.....	4
Figura 9 - EazeGraph Pie Chart	4
Figura 10 - EazeGraph Line Chart Mode	4

1. Introducción

1.1 Contexto y justificación del Trabajo

En la actualidad, el desarrollo de software se ha convertido en una compleja tarea que requiere la participación de muchos componentes externos, tanto humanos como otros sistemas informáticos [1]. Estos problemas requieren el uso de sofisticadas herramientas de gestión de diversa índole, como puede ser la gestión de recursos humanos, pruebas unitarias, pruebas de integración, validaciones de estándares o la gestión de la calidad [2].

Desarrollar estos sistemas software puede conllevar la intervención de múltiples equipos, de distintas compañías o incluso de distintos países, esto supone un esfuerzo extra a la hora de gestionar cada uno de los componentes desarrollados de forma independiente que componen el sistema software final.

Además, el mercado actual no proporciona las suficientes herramientas para suplir esta necesidad de gestión de múltiples elementos del desarrollo de software, y las utilidades existentes para este fin presentan múltiples carencias.

Una de las herramientas web actuales de código libre que presenta características útiles para este fin se denomina *Positioning System*, cuya finalidad es la de dar soporte a la gestión y monitorización de la construcción e integración de sistemas intensivos software, siendo extensible también a proyectos de otra índole. En particular, PositioningSystem es una herramienta de especial ayuda y apoyo en la construcción de grandes proyectos de desarrollo, donde participan diferentes empresas y/o socios, y que pueden incluso estar distribuidos geográficamente.

El objetivo de este Trabajo de Fin de Máster es el de desarrollar una aplicación para dispositivos Android que cubra una de las carencias de Positioning System, la nula adaptación a plataformas móviles. De esta manera se podrán llevar a cabo las tareas de gestión de forma ubicua necesitando únicamente una conexión a internet. Estas tareas comprenden la creación de proyectos, familias de productos, subfamilias de productos y productos finales, que podrán ser modificados para reflejar su estado de integración final y proporcionar así gráficos para comprender la marcha de uno o múltiples proyectos.

1.2 Objetivos del Trabajo

El principal objetivo de Trabajo de Fin de Máster es el desarrollo de una aplicación para la plataforma Android que trasladará la funcionalidad de una herramienta web existente al entorno nativo de esta plataforma móvil. Para ello se hará uso del servicio de backend proporcionado por la herramienta web, con el que esta aplicación se podrá comunicar para gestionar datos, obteniendo la siguiente arquitectura:

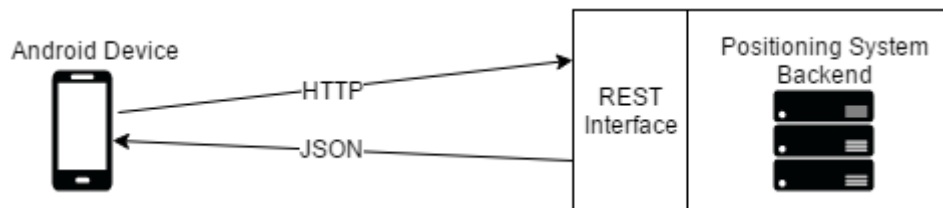


Figura 1 - Arquitectura del Sistema

Para lograr este objetivo se han definido los siguientes requisitos funcionales y no funcionales:

Requisitos Funcionales (FR):

- FR1: Sistema de registro y login que permita a los usuarios acceder y gestionar sus datos personales.
- FR2: Sistema de gestión de componentes, es decir, operaciones de creación, lectura, actualización y borrado de proyectos, familias de productos, subfamilias de productos y productos finales.
- FR3: Funcionalidad básica de red social, invitación de colaboración a los proyectos a otros usuarios de la plataforma y mensajería.
- FR4: Información estadística, módulo que permitirá observar la marcha de los proyectos en cualquier punto de su integración, a nivel general de proyecto o básico de productos.

Requisitos No Funcionales (NFR):

- NFR1: Escalabilidad, el producto ha de ser fácilmente escalable para añadir más funcionalidad al usuario como gráficos y campos de datos.
- NFR2: Fiabilidad, el producto ha de contener el mínimo posible de errores en tiempo de entrega haciendo una cobertura de pruebas unitarias mínima del 50%.
- NFR3: Confidencialidad, el producto ha de preservar la confidencialidad de los datos de los usuarios así como de sus proyectos.
- NFR4: Alto Rendimiento: el sistema ha de realizar una rápida presentación y actualización de datos para proporcionar la mejor experiencia de usuario posible de forma nativa.

1.3 Enfoque y método seguido

Debido a la naturaleza de este Trabajo de Fin de Máster, el enfoque de la aplicación a realizar será guiado por la interfaz del servicio web de la herramienta en la que se basa, *Positioning System*. Es decir, se realizará una adaptación de un sistema web a una plataforma móvil, mejorándola y añadiendo una mejor experiencia de usuario. Se ha optado por realizar la adaptación para una plataforma nativa en lugar de realizar una aplicación híbrida ya que la plataforma nativa proporciona un mejor rendimiento a la hora de presentar gráficos en pantalla, atendiendo al requisito no funcional de rendimiento.

A la hora de realizar este Trabajo de Fin de Máster se ha planteado un proceso de desarrollo iterativo e incremental, elaborando un primer prototipo con funcionalidades básicas y comenzando a iterar realizando un conjunto de tareas básicas, un número n de veces hasta llegar al producto final. Estas tareas básicas planteadas han sido las siguientes:

- Diseño del concepto o funcionalidad: En esta etapa se elabora un diseño del requisito funcional a implementar.
- Validación de diseño: Se realiza una validación del diseño elaborado.
- Implementación del diseño: Una vez validado el diseño se procede a la codificación de sus componentes e interacciones.
- Pruebas unitarias: Durante la implementación se ha de dar cobertura mediante pruebas unitarias.
- Validación de la implementación: Se validará la implementación mediante las pruebas unitarias realizadas.

Se ha decidido hacer uso de este proceso iterativo e incremental debido a la modularidad del desarrollo para aplicaciones Android, que permite realizar una excelente separación entre componentes e interacciones, permitiendo así un rápido desarrollo de integración con el resto de componentes.

1.4 Planificación del Trabajo

La planificación temporal se ha realizado de acuerdo a los hitos establecidos en cada una de las Pruebas de Evaluación continua (ver Ilustración 2). Cada tarea en un día concreto significa 2 horas de trabajo en ese día, es decir, si en un día coinciden las líneas temporales de dos tareas, significan 4 horas de trabajo. Siguiendo estas reglas el calendario resultante es el siguiente:

Desde el 24 de Febrero de 2016 al 9 de Marzo de 2016 se realiza el paquete de trabajo “Plan de Trabajo” que supone cinco tareas diferenciadas. En esta fase se invierte un total de 12 horas de trabajo en días festivos y 4 horas en días laborales.

Desde el 10 de Marzo de 2016 al 6 de Abril de 2016 se realiza el paquete de trabajo “Diseño” que supone cinco tareas diferenciadas. En esta fase se invierte un total de 24 horas de trabajo en días festivos y 12 en días laborales.

Desde el 7 de Marzo de 2016 al 18 de Mayo de 2016 se realiza el paquete de trabajo “Implementación” que supone cinco tareas diferenciadas. En esta fase se invierte un total de 40 horas de trabajo en días festivos y 24 en días laborales.

Desde el 19 de Mayo de 2016 al 6 de Junio de 2016 se lleva a cabo el paquete de trabajo final de este Trabajo de Fin de Master, denominado “Entrega Final” que supone 3 tareas, conlleva una inversión en horas de trabajo de 12 horas de trabajo en días festivos y 20 horas de trabajo en días laborales.

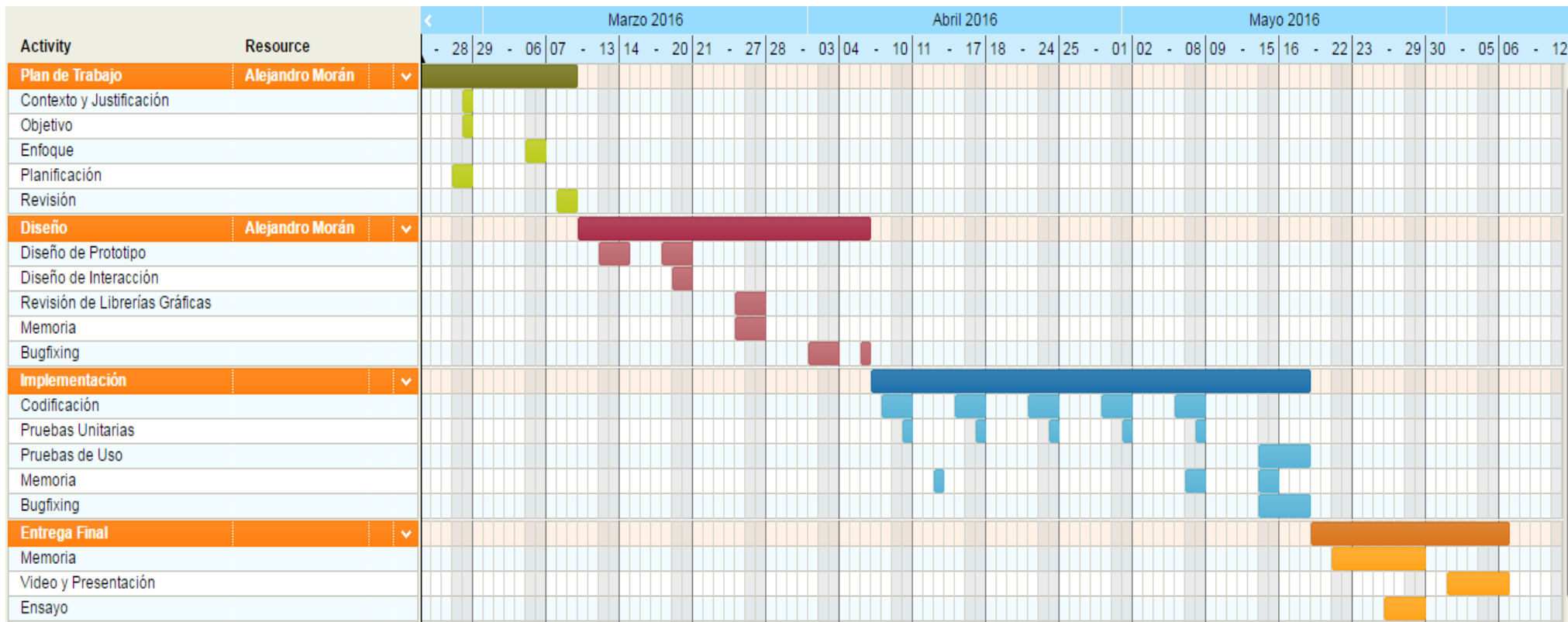


Figura 2 - Planificación Temporal

1.5 Breve resumen de productos obtenidos

- Memoria del Trabajo de Fin de Máster
Documentación completa y detallada del trabajo realizado
- Prototipo de diseño
Documentación y prototipo de la aplicación
- Modelo de Diseño
Diseño completo de la aplicación
- Modelo de Diseño con Interacción
Diseño completo de la aplicación que permite interacción con los componentes y transiciones.
- Aplicación Final
Paquete de producción listo para entrega a cliente con la aplicación final desarrollada.
- Video Demostrativo y Presentación
Vídeo screencast demostrativo del trabajo realizado y presentación Power Point.

1.6 Breve descripción de los otros capítulos de la memoria

Este TFM se divide en seis capítulos diferenciados. En este primer capítulo se ha presentado la motivación y objetivos para la realización del mismo.

En el segundo capítulo se definen conceptos previos necesarios para el correcto entendimiento del resto del proyecto.

El tercer capítulo expone detalladamente la herramienta *Positioning System*, el caso de estudio en el que se basa este TFM, una breve introducción precede la explicación de los detalles técnicos del sistema.

El cuarto capítulo expone el análisis realizado para construir la aplicación para Android *Positioning System*, las historias de usuarios que recogen los requisitos para desarrollo del proyecto, casos de uso y diagrama de clases de nivel de análisis.

El quinto capítulo describe la herramienta desarrollada y su funcionamiento como aplicación móvil.

El sexto y último capítulo expone el conjunto de conclusiones obtenidas durante y después de la construcción de la aplicación para Android PositioningSystem, así como los trabajos futuros que se pueden realizar a partir de este trabajo.

2. Diseño Técnico

2.1 Usuarios y Contextos de Uso

Este Trabajo de Fin de Máster ha sido realizado siguiendo el proceso de Diseño Centrado en el Usuario mediante el cuál se pone el foco de interés en el usuario, el uso que hará del sistema y los resultados que espera del mismo contestando a las siguientes preguntas:

¿Quiénes son los usuarios?

Positioning System es un sistema creado para la monitorización de la integración de Sistemas Intensivos Software, por tanto los usuarios serán directivos, gestores o empleados de compañías que realicen integraciones de grandes productos software, compuestos por múltiples módulos o componentes independientes, distribuidos en múltiples equipos que incluso podrían estar distribuidos en diferentes localizaciones geográficas.

¿Cuáles son sus tareas y metas?

Las tareas de las que los usuarios se han de ocupar en el sistema es de la inserción y el actualizado de los datos con el objetivo de proporcionar la capacidad de generar estadísticas a nivel de proyecto que aporten información sobre la integración específica de cada rama de productos o del proyecto completo.

¿Qué nivel de experiencia tienen los usuarios?

El sistema debe ser lo suficientemente intuitivo para usuarios no técnicos que puedan introducir datos en formularios sencillos, y generar estadísticas de forma sencilla.

¿Qué funciones se necesitan?

Los usuarios de la aplicación necesitarán un sistema de soporte de cuentas de usuario, un sistema de creación de proyectos, familias de productos, subfamilias de productos y productos finales, un sistema de invitación a otros usuarios a proyectos y un sistema de generación de estadísticas de los proyectos.

¿Qué información necesitan los usuarios y de que manera?

Además de las credenciales de acceso, los únicos datos necesarios para los usuarios son los propios a introducir en el sistema referentes a cada proyecto, familia, subfamilias, productos o socios. Así como la responsabilidad de mantener actualizados esos datos para que las estadísticas proporcionadas sean veraces.

¿Cómo se espera funcione?

Los usuarios han de instalar una aplicación en dispositivos móviles Android que proporcionará todas las funcionalidades, una vez realicen el acceso al sistema, dispondrán de todas las características del sistema en una aplicación nativa.

2.2 Diseño Conceptual

Para esta fase se ha decidido hacer uso de dos técnicas que ayudan a detectar los requisitos funcionales de la aplicación, los escenarios de uso y las historias de usuario, se han definido dos de los escenarios más relevantes de uso de la aplicación como historias de usuario de usos de elementos comunes de la misma.

Escenario 1:

Tarea: Obtención de estadísticas generales de la integración de todos los proyectos

Personaje: Miguel

Grupo de Usuarios: Directivos

Contexto:

Miguel es un alto directivo de una compañía de software que desarrolla una gran variedad de productos de alta complejidad. Para Miguel es necesario en todo momento poder acceder a un sistema que muestre la marcha actual de todos los proyectos en los que su empresa se encuentra involucrada y además el nivel de rendimiento de cada uno de los socios comerciales que integran o gestionan la integración de alguno de los productos desarrollados en estos proyectos. Miguel usará estos informes especialmente en las reuniones con sus socios comerciales, junta de accionistas y resto de la directiva para realizar decisiones estratégicas de la compañía. Además, Miguel viaja constantemente y necesita acceder a las estadísticas desde cualquier parte del mundo de forma rápida y fácil, ya que no posee demasiados conocimientos técnicos.

Escenario:

Miguel accede a la aplicación desde su dispositivo móvil Android, después de acceder al sistema con sus credenciales este visualiza la pantalla de “*Dashboard*” en la que obtiene información sobre el número de proyectos totales y el número de productos en cada estado (integrado, integración en curso o integración finalizada). Después de visualizar estos datos, Miguel pulsa sobre la pestaña de “*Overall View*” y es redirigido a una pantalla en la que puede visualizar dos gráficos rápidamente, estos gráficos son un *Gauge Chart* que muestra el nivel de integración general de todos los productos bajo todos los proyectos, además visualizará otro gráfico de tipo *Columns Chart* que mostrará el nivel de rendimiento de todos los socios en todos los proyectos.

Escenario 2:

Tarea: Obtención de estadísticas generales de la integración de un proyecto específico

Personaje: Laura

Grupo de Usuarios: Project Managers

Contexto:

Laura es la responsable de muchos proyectos de Tecnologías de Información de una empresa dedicada al desarrollo y mantenimiento de un *framework* de desarrollo web cuya principal ventaja es la división en componentes independientes y atómicos de todos los elementos de diseño web. Desarrollando así los componentes en diferentes equipos de trabajo e integrando todos en cada *release* del *framework*. En el día a día de Laura se hace necesaria la obtención de estadísticas específicas de los proyectos que se desarrollan bajo su cargo para ofrecer informes detallados en las reuniones tanto con supervisores como subordinados.

Escenario:

Laura accede a la aplicación mediante su dispositivo móvil, después de acceder al sistema con sus credenciales es redirigida a la pantalla de "Dashboard" en la que se muestra el número de proyectos en los que se encuentra involucrada y el número de productos integrados bajo esos proyectos organizados por su estado (integrado, integración en curso o integración finalizada). Después de esto, Laura pulsa sobre el botón "Projects" el cual le redirigirá a un listado con todos sus proyectos acompañados de una breve descripción. Al pulsar sobre uno de estos proyectos listados Laura es redirigida a la pantalla de información general del proyecto dónde se ofrecen datos detallados de las familias de productos, subfamilias de productos y productos finales que se han de integrar bajo ese proyecto. Después de esto, Laura pulsa sobre el botón del menú "Statistics" que le redirigirá a una pantalla en la que podrá ver información del proyecto organizada en un gráfico del tipo *Columns Chart*, esta información será el nivel de integración por cada familia de productos, además dispondrá de un gráfico de tipo *Pie Chart* mediante el cual se mostrará la información de la distribución de los países a los que pertenecen los socios que participan a cualquier nivel del proyecto.

Historias de Usuario

En las metodologías ágiles, las historias de usuario son requisitos funcionales que responden a tres preguntas: ¿A quién se beneficia?, ¿qué se quiere? y ¿cuál es el beneficio?. Este proceso de definición de historias de usuario puede ser realizado durante cualquiera de las iteraciones del proyecto, pudiendo modificar o eliminar alguna de las ya existentes o añadir nuevas. Se ha definido un modelo para la definición de historias de usuario el cual contiene un identificador de la historia de usuario, nombre, descripción, conversación y criterios de aceptación.

H.U. X.X. - Nombre de la historia de usuario	
Descripción:	
Descripción extendida de la historia de usuario	
Conversación	
Número del paso	Pasos a realizar por el usuario
Criterios de Aceptación	
Número del criterio de aceptación	Criterios de aceptación

Clasificación de las historias de usuario:

- Gestión de Cuenta
 - Crear una cuenta
 - Acceso al sistema
 - Eliminar una cuenta
- Gestión de proyectos
 - Crear un proyecto
 - Crear una familia
 - Crear una subfamilia
 - Crear un producto
 - Gestión de contactos
 - Invitar un contacto a proyecto

Gestión de Cuentas:

H.U. 1.1. - Crear una cuenta	
Descripción:	
Como usuario anónimo necesito una pantalla para la creación de una cuenta que habilite el acceso al sistema usando las credenciales establecidas en el mismo.	
Conversación	
1	Accedo a la aplicación
2	Accedo a la pantalla de registro pulsando sobre el texto enlazado "Create an account"
3	Introduzco mis datos personales y mis credenciales de acceso (nombre, apellidos, email, repetición de email, password y repetición de password) enviando el formulario de registro al finalizar.
4	
5	La aplicación verifica la validez de los datos introducidos, la dirección de correo electrónico introducida ha de ser una dirección válida y no debe existir previamente en la base de datos, debe coincidir con el campo de repetición así como la contraseña introducida ha de coincidir con su campo de repetición.
6	Los datos son enviados al backend mediante una petición http invocando la operación POST con los datos del formulario adjuntos.
7	El sistema de backend introduce los datos del usuario en la base de datos devolviendo un mensaje en JSON con un campo denominado success con valor true.
	La aplicación recibe la respuesta de registro exitoso y redirige al usuario a la pantalla de login anterior.
Criterios de Aceptación	
1	Si puedo crear una cuenta con unas determinadas credenciales de acceso
2	Si una vez creada la cuenta, soy redirigido a la pantalla de login.

H.U. 1.2. - Acceso al sistema	
Descripción:	
Como usuario anónimo necesito una pantalla que me permita el acceso al sistema mediante el uso de unas credenciales previamente creadas.	
Conversación	
1	Accedo a la aplicación.

2	Visualizo la pantalla de login.
3	Introduzco mis credenciales en los campos de texto habilitados para ello.
4	Pulso el botón “ <i>Login</i> ”
5	La aplicación envía mis credenciales de acceso al backend mediante una petición http invocando el método GET con los datos introducidos.
6	
7	El backend recibe las credenciales y comprueba su validez en la base de datos.
8	El backend crea una sesión en el servidor para el usuario y devuelve una respuesta JSON a la aplicación con un campo denominado <i>success</i> con valor <i>true</i> .
	Soy redirigido a la pantalla principal de la aplicación.
Criterios de Aceptación	
1	Si, usando mis credenciales de acceso previamente establecidas se me permite el acceso al sistema.
2	Si, una vez iniciada la sesión en el sistema soy redirigido a la pantalla principal de la aplicación.

H.U. 1.3. - Eliminar una cuenta

Descripción:

Como usuario necesito una pantalla que me permita eliminar mi cuenta y mis datos del sistema

Conversación

1	Accedo a la configuración de la aplicación en el menú superior de la pantalla principal después de acceder al sistema.
2	Selecciono la opción “Delete my account”.
3	Acepto el aviso de borrado de datos.
4	Soy redirigido a la pantalla de login.

Criterios de Aceptación

1	Mis datos son eliminados de la base de datos.
---	---

Gestión de proyectos:

Para mejorar la legibilidad de esta sección se han omitido las historias de usuario 2.3 (Creación de una subfamilia) y 2.4 (Creación de un producto) dada su similitud a la historia de usuario 2.1 la cual define el comportamiento genérico de estas otras.

H.U. 2.1. - Crear un proyecto	
Descripción:	
Como usuario necesito una pantalla para la creación de proyectos	
Conversación	
1	Accedo a la sección de proyectos de la aplicación.
2	En el menú superior pulso " <i>New Project</i> ".
3	Soy redirigido a la pantalla de creación de un nuevo proyecto.
4	Introduzco los datos relativos al proyecto en el formulario mostrado.
5	Pulso en el botón " <i>Create</i> "
6	Soy redirigido a la pantalla de proyectos.
Criterios de Aceptación	
1	Si, al crear un proyecto puedo visualizar sus datos en el listado de proyectos.

H.U.2.2. - Crear una familia	
Descripción:	
Como usuario necesito una pantalla para la creación de familias dentro de un proyecto existente	
Conversación	
1	Accedo al proyecto al que añadiré la familia.
2	En el menú superior pulso el botón " <i>New Family</i> ".
3	Soy redirigido a la pantalla de creación de una nueva familia.
4	Relleno los datos solicitados para la creación de una nueva familia.
5	Pulso el botón " <i>Create Family</i> ".
6	Soy redirigido a la pantalla del proyecto.
Criterios de Aceptación	
1	Una vez creada la familia aparece listada bajo las familias

pertenecientes al proyecto en el que se creó.

H.U.3.1. - Invitar un contacto al proyecto

Descripción:

Como usuario necesito una pantalla que me permita realizar invitaciones al proyecto a otros usuarios de la aplicación conociendo su dirección de correo electrónico.

Conversación

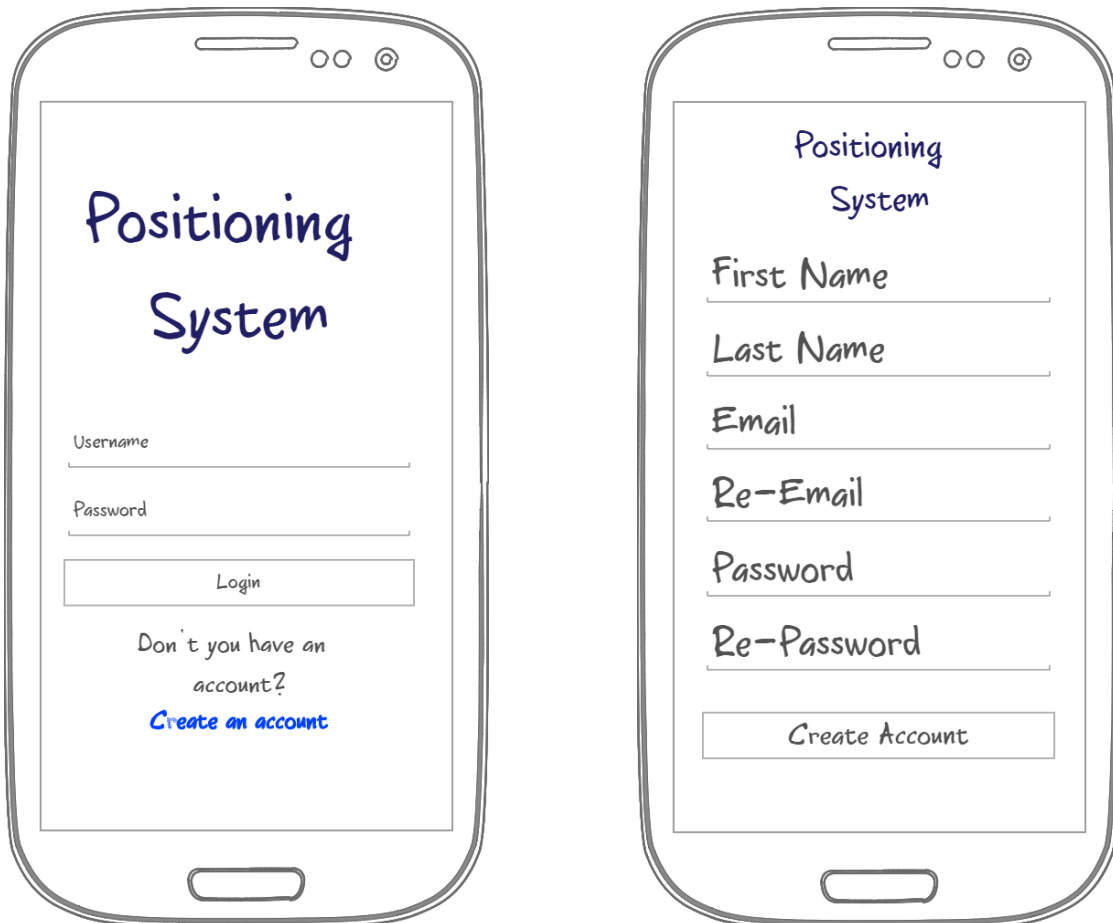
- | | |
|---|--|
| 1 | Accedo al proyecto al que invitaré a otro usuario. |
| 2 | En el menú superior pulso el botón <i>"Invite"</i> . |
| 3 | Soy redirigido a la pantalla de invitación de usuarios. |
| 4 | Introduzco la dirección de correo electrónico del usuario a invitar. |
| 5 | Pulso el botón <i>"Send Invitation"</i> . |
| 6 | Soy redirigido a la pantalla del proyecto. |

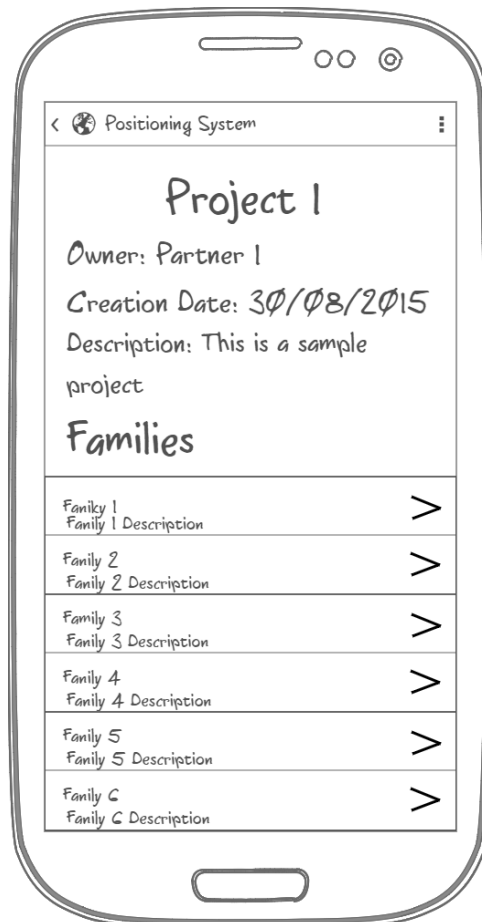
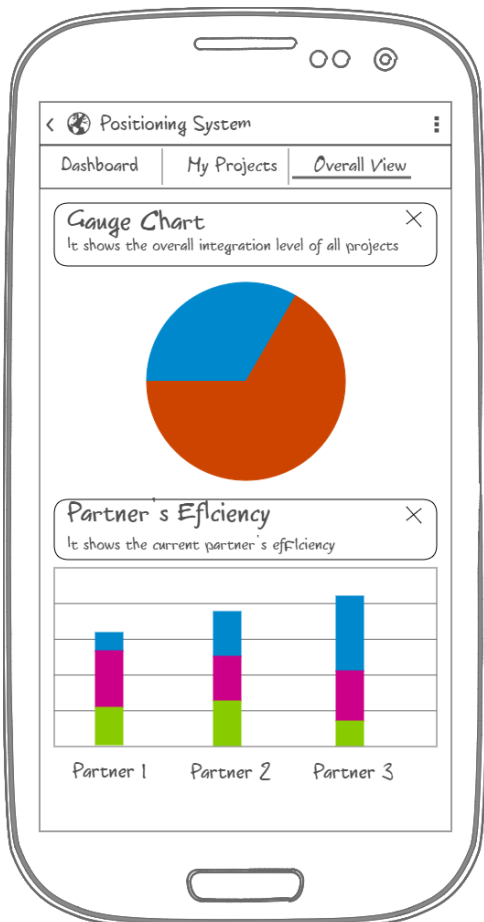
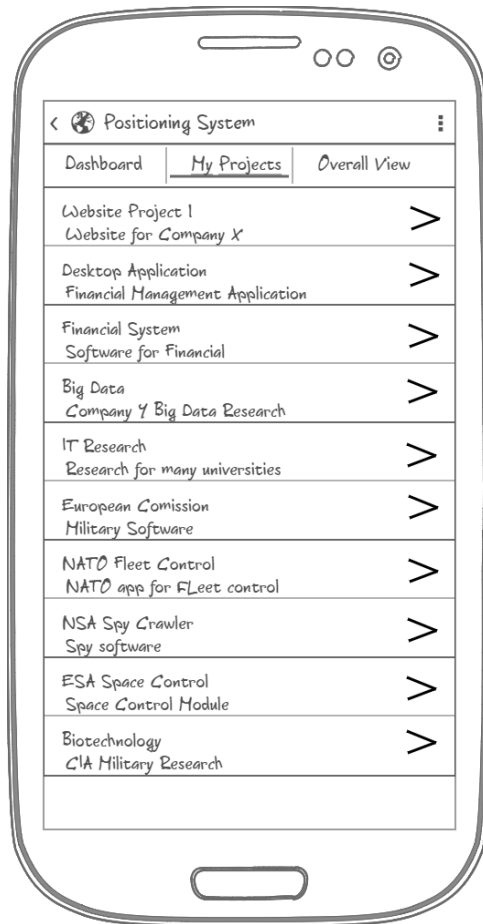
Criterios de Aceptación

- | | |
|---|---|
| 1 | Una vez invitado, el usuario receptor puede visualizar el proyecto al que ha sido invitado dentro de su listado de proyectos. |
|---|---|

2.3 Prototipado

Los siguientes prototipos muestran la interacción general de la aplicación, omitiendo prototipos repetidos como la visión de familias, subfamilias y productos ya que su funcionamiento y distribución es similar al general de proyectos. Los siguientes prototipos se corresponden a la pantalla de inicio, registro de usuario, dashboard, proyectos, visión general y vista de un proyecto.





2.4 Casos de Uso

Para la presentación de los requisitos funcionales extraídos en este Trabajo de Fin de Máster se ha hecho uso de diagramas de casos de uso, que proporcionan una visión global de los *stakeholders* que interactúan con el sistema. Una de las ventajas de estos diagramas es su facilidad de interpretación, que permite mostrar las diferentes posibilidades de interacción con el sistema y los actores que intervienen en cada caso. A continuación se muestran los procesos que el usuario de la aplicación móvil puede realizar en la misma como único actor. Estos diagramas de casos de uso han sido divididos y agrupados por funcionalidad para mejor visualización.

Gestión de proyectos:

Después de acceder al sistema, el usuario puede gestionar sus proyectos, y este proceso implica las tareas asociadas de creación, edición y borrado. El usuario puede consultar las estadísticas de su proyecto y además puede invitar a otros usuarios a formar parte del proyecto y permitirles visualizarlo.

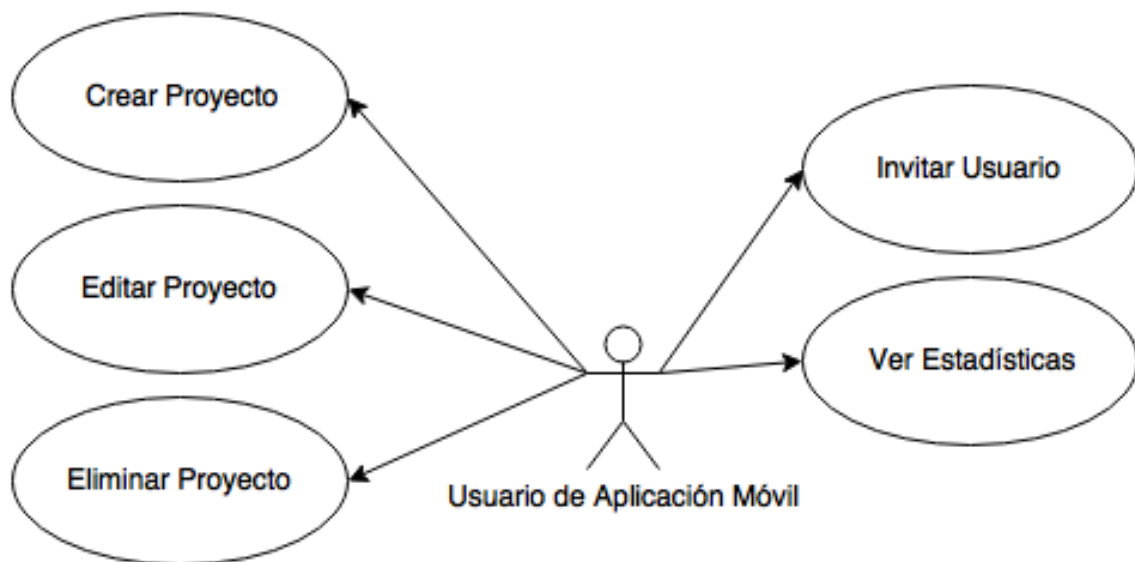


Figura 3 - Caso de Uso: Gestión de Proyectos

Gestión de familias, subfamilias y productos:

Los procesos de gestión de familias y subfamilias permiten la creación, modificación y borrado de las mismas, además el proceso de gestión de subfamilias da acceso al proceso de gestión de productos. Este proceso de gestión de productos representa la gestión de los elementos atómicos de los proyectos.

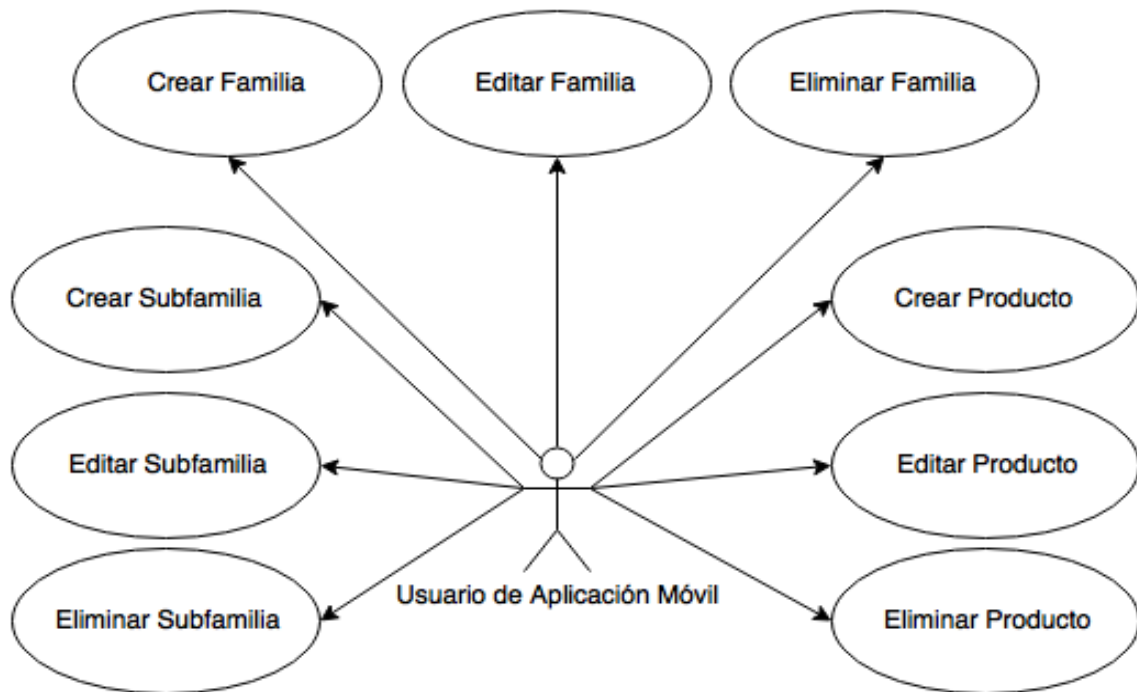


Figura 4 - Caso de Uso: Gestión de Familias, Subfamilias y Productos

2.5 Modelo de Datos

Como parte del análisis del sistema de backend se ha realizado un diagrama similar al estándar de facto entidad-relación usando la herramienta MySQL Workbench. A continuación se proporciona un resumen de las entidades desarrolladas en el modelo de datos así como sus relaciones:

Usuarios:

Entidad que representa a los usuarios de la aplicación junto a sus datos personales. Estos usuarios pueden enviar mensajes a otros usuarios, ser y formar contactos con otros usuarios, crear proyectos, socios para sus proyectos y pueden poseer privilegios de visibilidad sobre proyectos creados por otros usuarios. Además, los usuarios pueden enviar mensajes a otros usuarios, generando así la entidad derivada de la relación, "mensajes".

Socios:

Los socios son la pieza fundamental para la creación del resto de componentes de la aplicación, ya que a la hora de crear un proyecto se ha de asignar un socio previamente creado al mismo. Además, estos socios están asociados a un país concreto.

Proyectos:

Entidad fundamental de la aplicación que representa el contenedor más grande de elementos propiamente objetivo de este proyecto.

Familias, Subfamilias y Productos:

Definen los componentes de los proyectos, cada proyecto está formado por familias de productos, que a su vez están compuestos de subfamilias de productos que integran múltiples productos

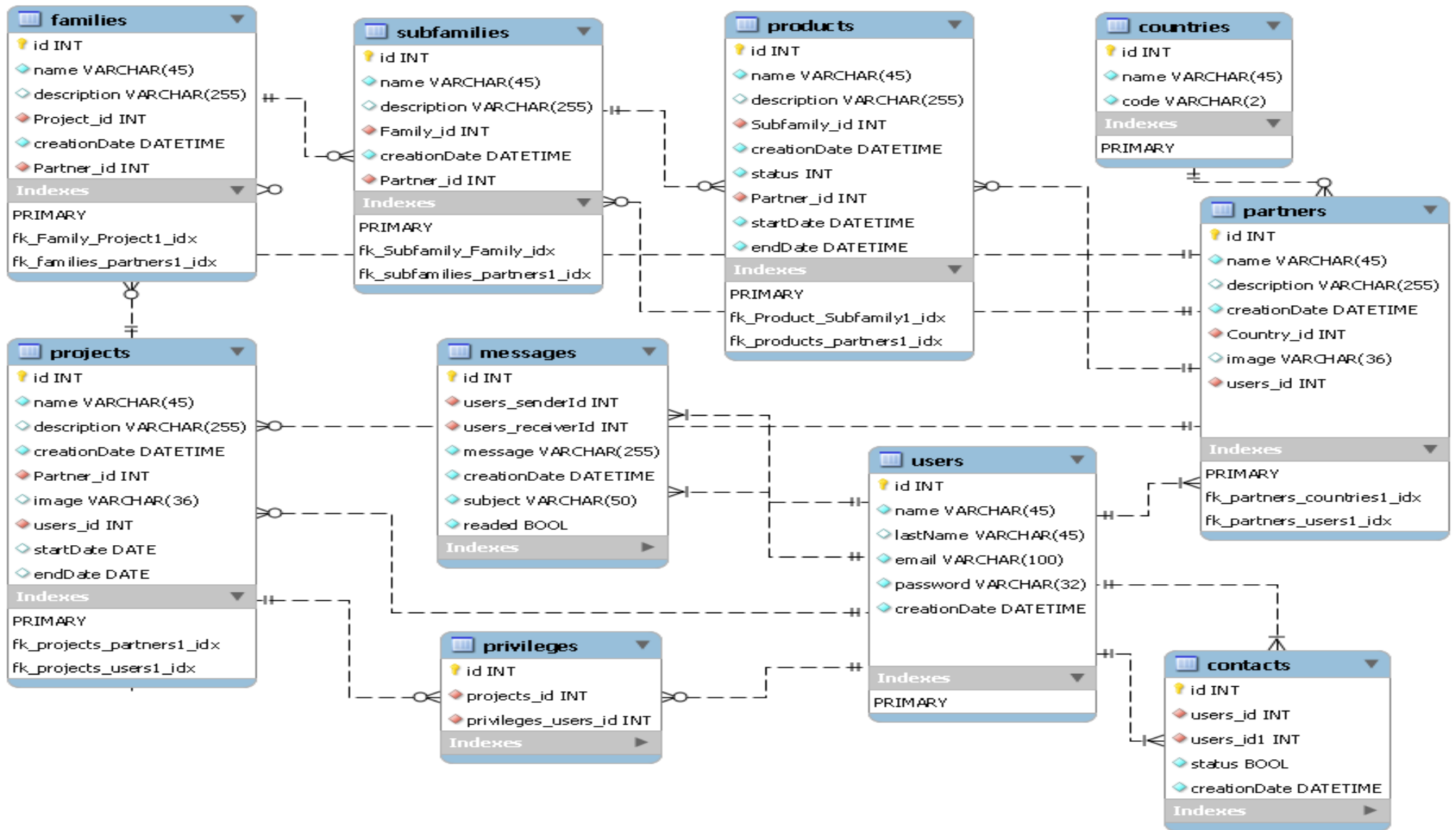


Figura 5 - Modelo de Datos

2.6 Arquitectura del Sistema

En la comunicación entre módulos de la arquitectura, encontramos en un extremo la aplicación móvil, en un punto intermedio el servicio web que proporciona datos del otro extremo, la base de datos MySQL.

El servicio web será el principal módulo con el que se comunicará la aplicación móvil, este proporciona una interfaz pública de servicios a los que realizar llamadas http, cabe destacar la no estandarización del servicio web, emula el funcionamiento de un servicio REST sin serlo, devuelve objetos serializados en formato JSON como resultado de la comunicación. La interfaz pública que proporciona el servicio es la siguiente:

REST API Interface

Contacts	Families
/contacts/ /contacts/requests /contacts/add/{email} /contacts/accept/{id}	/family/{id} /families/{projectId}
	Users
	/user
	Subfamilies
	/subfamily /subfamilies/{familyId}
Products	Partners
/products/{subfamilyId} /products/{productId} /products/status/{status}	/partners /partners/{partnerId}
Projects	Messages
/projects /projects/snapshot /projects/others/{id} /projects/spider/{projectId}/{snapshotId} /projects/invite/{projectId}	/messages /messages/unread /messages/markAsReaded/{messageId} /messages/markAsReaded/{messageId} /messages/send/{contactId}

Figura 6 - Interfaz del Servicio Web

2.7 Desarrollo

Librerías Externas

El desarrollo de la aplicación Android ha requerido el uso de las tres siguientes librerías externas para facilitar tareas de trabajo en red, optimización de recursos y renderizado gráfico:

Volley:

Volley es una librería HTTP que facilita el trabajo en red en Android y además mejora su velocidad. Volley está disponible a través del repositorio abierto AOSP [3].

Esta librería ofrece los siguientes beneficios:

- Planificación automática de las peticiones de redirigida
- Múltiples conexiones concurrentes
- Caché de respuestas transparente en memoria y disco de acuerdo al estándar HTTP (cache coherence)(referencia bibliográfica aquí)
- Soporte para la priorización de peticiones
- API para la cancelación de peticiones. Se puede cancelar una única petición o se pueden establecer bloques o ámbitos de peticiones a cancelar.
- Proporciona la robustez necesaria para rellenar la interfaz de usuario con datos extraídos de forma asíncrona de la red.
- Herramientas para el depurado y trazado de errores.

Además, Volley destaca en las operaciones de tipo RPC usadas para rellenar datos de la interfaz de usuario, como extraer una página de resultados de búsqueda en datos estructurados. Se integra fácilmente con cualquier protocolo y permite el soporte para cadenas de texto, imágenes y JSON. Volley libera al desarrollador del trabajo necesario para codificar estructuras de gestión de red y permite poner el foco de desarrollo en la lógica de la aplicación. Volley no es la solución adecuada para grandes descargas o operaciones de streaming ya que almacena las respuestas en memoria mientras estas son procesadas.

La librería principal de Volley se desarrolla en el repositorio abierto de AOSP bajo /frameworks/volley y contiene el manejador principal de peticiones así como un conjunto de utilidades comunes disponibles en el toolbox de Volley. La manera más sencilla para añadir Volley a un proyecto es clonar el repositorio y añadirlo como una librería del proyecto.

ButterKnife:

Butterknife es una librería para la "inyección" de Vistas en Android. Esto significa que la librería escribe código el código repetitivo común de las vistas

por ti basándose en anotaciones para ahorrar tiempo de desarrollo y reducir significativamente el código repetitivo a escribir [4].

Usa anotaciones en tiempo de compilación, lo que significa que no añade ningún coste adicional en tiempo de ejecución. Haciendo llamadas a delegados vinculados al código generado este se puede ver y depurar. Esto significa que Butterknife no ralentiza las aplicaciones en ningún caso.

EazeGraph:

EazeGraph es una librería Android para la creación de bonitos y vistosos gráficos. Su objetivo principal es el de crear una librería ligera de fácil uso y altamente personalizable con un look actual.

Esta librería no está orientada a propósitos de tipo matemático como las librerías achartengine o androidplot. El propósito principal es el de crear una presentación visual atractiva para el usuario de datos relacionados donde se proporciona el valor a representar y el resto se calcula de forma dinámica. Las aplicaciones deben soportar un nivel mínimo de API de 9 para hacer uso de esta librería [5].

Características:

- Generación dinámica de etiquetas
- Posibilidad de usar etiquetas de leyenda propias
- Animaciones para cada uno de los gráficos
- Interacción mediante pulso manual en los gráficos PieChart y LineChart
- Varios atributos XML para personalizar los gráficos

Actualmente existen los siguientes cuatro tipos de gráficos disponibles:

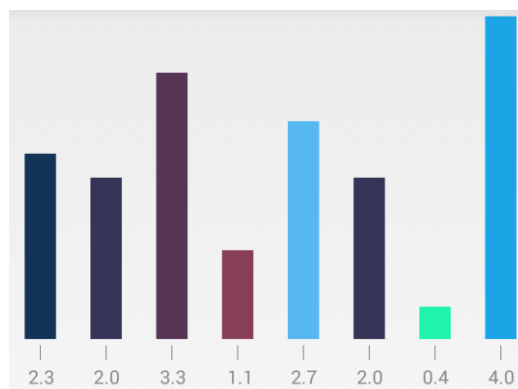


Figura 7 - EazeGraph Bar Chart

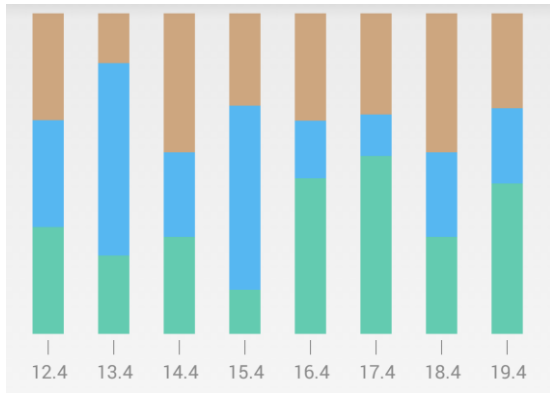


Figura 8 - EazeGraph Stacked Bar Chart

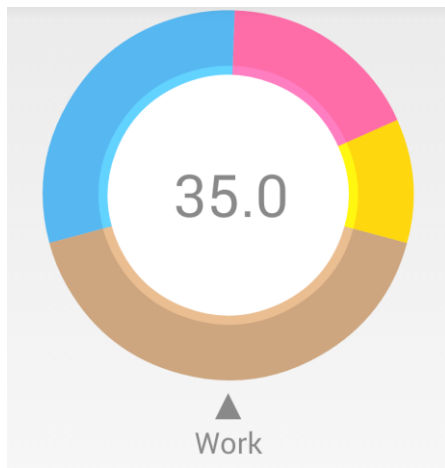


Figura 9 - EazeGraph Pie Chart

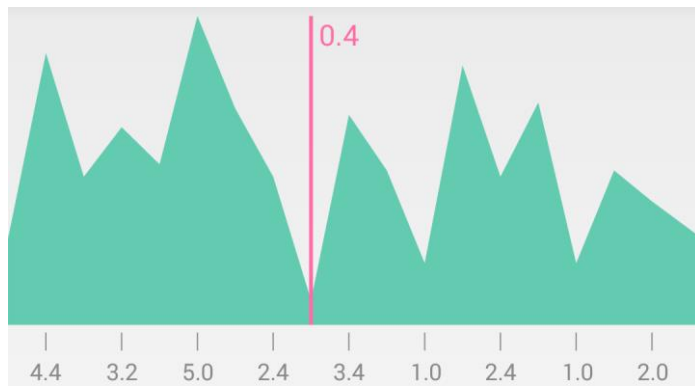


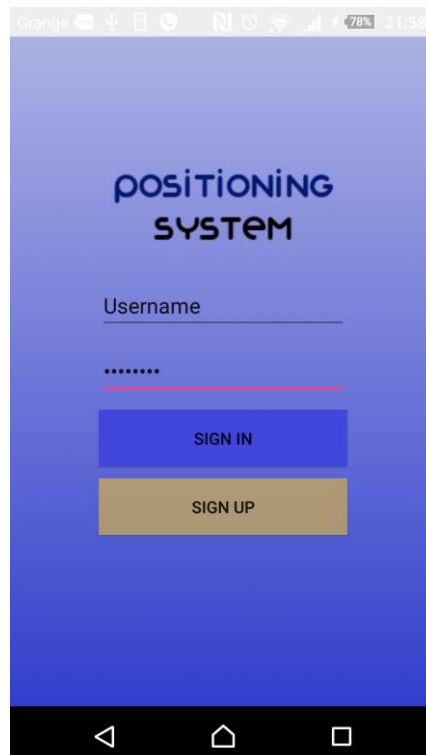
Figura 10 - EazeGraph Line Chart Mode

Producto Final

Como resultado de las diferentes iteraciones de desarrollo de este Trabajo de Fin de Máster se ha obtenido una aplicación para dispositivos Android con un total de X pantallas finales de presentación mediante las cuales los usuarios se comunicarán con el servicio web que sostiene la lógica del sistema. Estas pantallas resultantes son las siguientes:

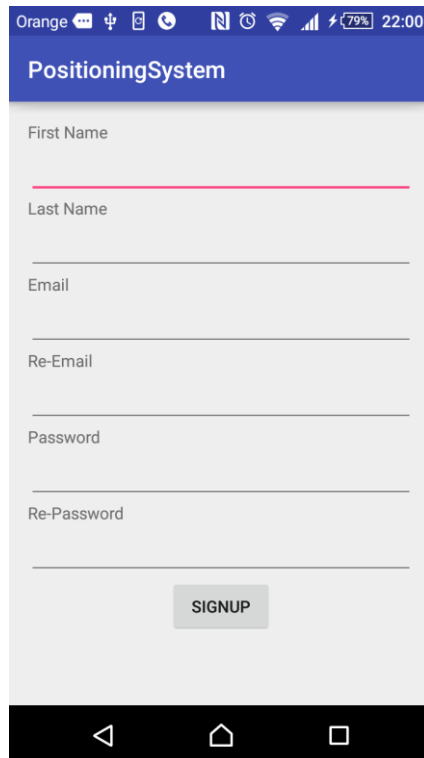
Login:

En esta pantalla se observa el logotipo de la aplicación junto a los campos de entrada de datos para acceder al sistema, en caso de no disponer de credenciales, se podrá navegar a la pantalla de registro a través del botón *Signup*. Estos datos de login son contrastados contra el servicio web a través de una operación HTTP GET con los parámetros de acceso. Este canal de comunicación debería ser HTTPS ya que en caso contrario estos datos de acceso podrían ser capturados por un intermediario.



Registro:

En esta pantalla se realizan las funciones básicas de registro de usuarios. Una vez se han rellenado todos los campos de forma correcta estos datos se envían al servicio web como cuerpo de una petición HTTP POST.

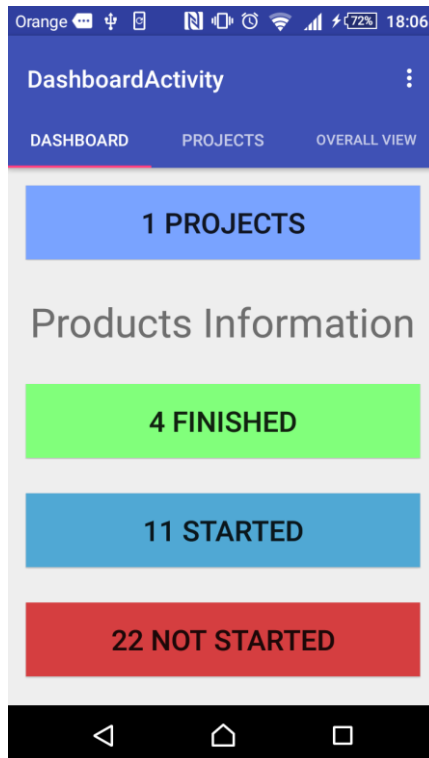


Dashboard:

Esta es la pantalla principal para los usuarios con credenciales validas, en ella se distingue un menú lateral superior de navegación y un *RelativeLayout* con botones para seleccionar las distintas opciones disponibles en la aplicación. Estos botones cambian su nombre de forma dinámica de acuerdo al número de componentes que representa su funcionalidad, es decir, aparece un número antes del texto que indica cuántos de esos componentes se encuentran en el estado definido por el texto del botón.

Para la carga dinámica de estos datos se usan llamadas al servicio web mediante la librería Volley, esta realiza peticiones HTTP GET a los servicios que devuelven la cuenta de cuantos productos se encuentran en un estado determinado.

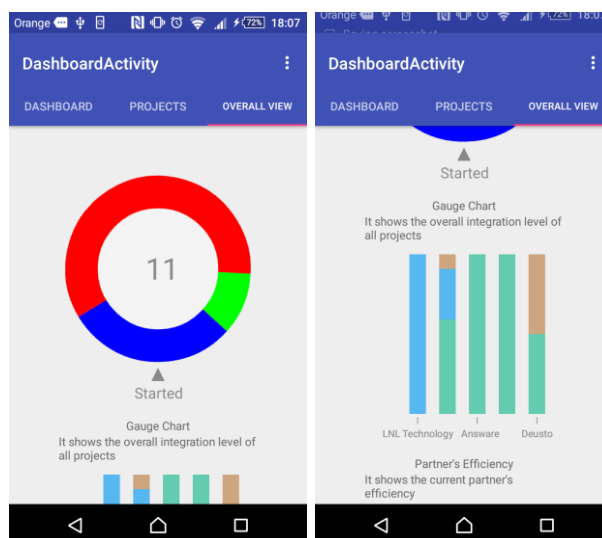
El menú de selección lateral cuenta con otras dos opciones, *Projects*, que otorga acceso a la pantalla de Proyectos y *Overall View*, que otorga acceso a la pantalla de visión general de los proyectos.



Overall View:

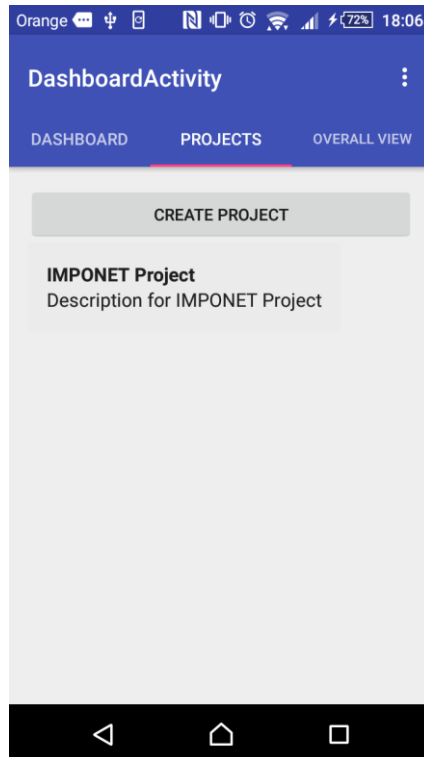
Esta vista presenta los datos generales de los proyectos de dos formas distintas, la primera, en un *Pie Chart* que representa el total de productos divididos en tres grupos dado su estado, productos que han sido totalmente integrados, productos que están en proceso de integración y productos que no han comenzado a ser integrados.

Por otro lado, se presenta un gráfico del tipo *Stacked Bar Chart* que indica en una columna por cada socio que participa en la integración de algún producto, el número de productos y el estado en el que se encuentran.



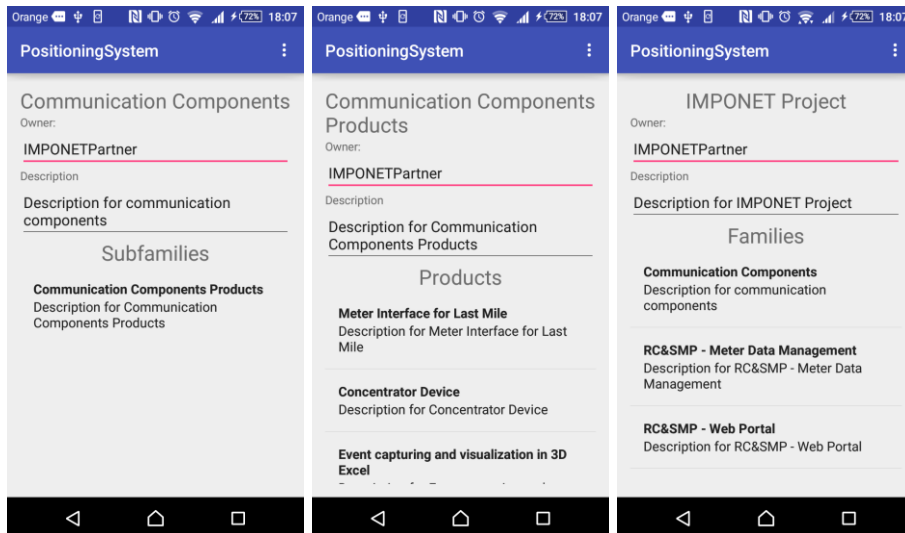
Projects:

La vista de proyectos es la principal vía de entrada a la información útil de la aplicación, a través de un listado se presentan los distintos proyectos creados por un usuario, seleccionando uno de estos proyectos se realiza una navegación a la pantalla denominada *Elements* adjuntando un parámetro que representa el identificador del proyecto.



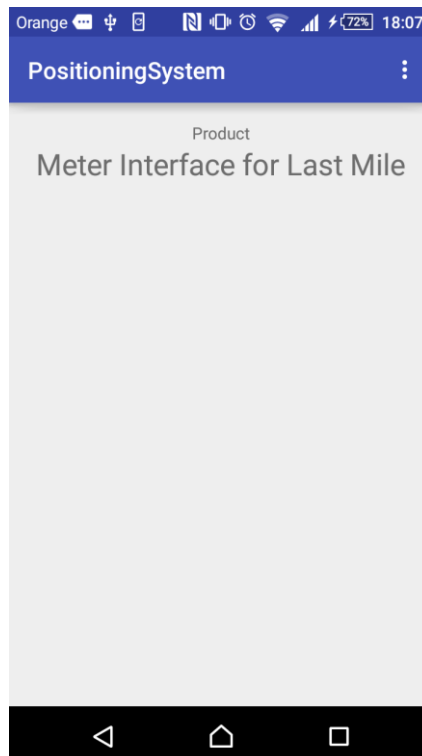
Elements:

Esta vista representa a nivel de diseño un elemento genérico, atendiendo a los parámetros que recibe de las vistas que navegan hacia ella, se representa alguno de los distintos componentes de la aplicación, estos pueden ser proyectos junto a sus familias, familias junto a sus subfamilias o subfamilias junto a sus productos. Es decir, si se navega desde la pantalla de proyectos pulsando sobre uno de los elementos listados, se obtendría esta vista *Elements* rellena con los datos de ese elemento, en este caso, un proyecto concreto.



Product:

Esta vista representa un producto final, a diferencia de los demás elementos genéricos de la aplicación, estos requieren una vista personalizada ya que disponen de campos adicionales como el estado, la fecha de inicio o la fecha de finalización de la integración de ese producto.



Statistics:

2.8 Pruebas

La ejecución de una fase de pruebas adecuada garantizará el éxito de la aplicación y evitará problemas futuros debido a posibles errores, además esto proporciona un grado mínimo de calidad de los productos y repercutirá sobre la imagen y valoración del mismo.

Pruebas de Integración:

Dada la naturaleza de este proyecto y su necesidad de velocidad de desarrollo no se han automatizado pruebas unitarias y por tanto se han definido pruebas de integración, que presentan las siguientes ventajas:

- Permiten analizar la interacción entre distintos elementos de un sistema
- Facilita la detección de errores derivados de la actividad de los usuarios y no únicamente de los métodos del código
- Facilitan las tareas necesarias para realizar pruebas de aceptación del software

Para la realización de esta fase de tests de integración se ha definido el siguiente plan de pruebas en el que se prueba cada una de las historias de usuario previamente definidas.

Historia de Usuario	Funcionalidad a probar	Prerequisitos
H.U. 1.1. - Crear una cuenta	El usuario pulsa sobre el botón para la creación de la cuenta y es redirigido a la pantalla de registro, una vez rellenos los datos el formulario se envía al servidor	Aplicación instalada
H.U. 1.2. - Acceso al sistema	El usuario accede al sistema mediante sus credenciales	Aplicación instalada y credenciales previamente creadas
H.U. 1.3. - Eliminar una cuenta	El usuario elimina su cuenta y no puede acceder mas a la aplicación	Aplicación instalada y credenciales previamente creadas
H.U. 2.1. - Crear un proyecto	El usuario crea un proyecto y este es visible en la pantalla de proyectos	Aplicación instalada y credenciales previamente creadas
H.U.2.2. - Crear una familia	El usuario crea una familia y esta es visible en la pantalla del proyecto al que está asignada	Aplicación instalada y credenciales previamente creadas
H.U.3.1. - Invitar un contacto al proyecto	El usuario es capaz de invitar a otro usuario a su proyecto a través de su email	Aplicación instalada, credenciales previamente creadas y un proyecto creado al menos

El desarrollo de las pruebas sigue los siguientes esquemas de validación y verificación:

H.U. 1.1. - Crear una cuenta				
Número de paso	Paso a seguir	Datos de entrada	Datos de salida	¿Válido?
1	Se accede a la aplicación	Pulsar el icono de la aplicación	Se muestra la pantalla de login	Si
2	Se accede la pantalla de registro	Pulsar el botón Signup	Se muestra la pantalla de registro	Si
3	Se rellenan los datos y se envía el formulario	Insertar texto en cada uno de los campos	Se redirige a la pantalla de login	Si

H.U. 1.2. – Acceso al sistema				
Número de paso	Paso a seguir	Datos de entrada	Datos de salida	¿Válido?
1	Se accede a la aplicación	Pulsar el icono de la aplicación	Se muestra la pantalla de login	Si
2	Se introducen los datos de login y se envían	Pulsar el botón Signin	Se redirige a la pantalla principal de la aplicación	Si

H.U. 1.3. - Eliminar una cuenta				
Número de paso	Paso a seguir	Datos de entrada	Datos de salida	¿Válido?
1	Se accede al menú de opciones	Pulsar el icono de la opciones	Se muestra la pantalla de opciones	Si
2	Se selecciona elimina la cuenta	Pulsar el botón de eliminar cuenta	Se redirige a la pantalla de login	Si
3	La cuenta ha sido eliminada	Se intenta acceder con las credenciales eliminadas	No se accede a la aplicación ya que las credenciales no existen	Si

H.U. 2.1. - Crear un proyecto				
Número de paso	Paso a seguir	Datos de entrada	Datos de salida	¿Válido?
1	Se accede a la pantalla de proyectos	Pulsar o desplazar el menú hacia la opción Projects	Se muestra la pantalla de proyectos	Si
2	Se crea un proyecto	Seleccionar la opción Create Project	Se redirige a la pantalla de creación de un proyecto	Si

3	Se envían los datos	Seleccionar el botón Create	Se redirige a la pantalla de proyectos y el proyecto creado aparece en el listado	Si
---	---------------------	-----------------------------	---	----

H.U. 2.2. - Crear una familia				
Número de paso	Paso a seguir	Datos de entrada	Datos de salida	¿Válido?
1	Se accede a la pantalla de un proyecto concreto	Pulsar sobre un proyecto en el listado de proyectos	Se muestra la pantalla de información de ese proyecto	Si
2	Se crea una familia	Seleccionar la opción Create Family	Se redirige a la pantalla de creación de una familia	Si
3	Se envían los datos	Seleccionar el botón Create	Se redirige a la pantalla de ese proyecto y la familia creada aparece en el listado	Si

H.U. 3.1. – Invitar un contacto al proyecto				
Número de paso	Paso a seguir	Datos de entrada	Datos de salida	¿Válido?
1	Se accede a la pantalla de un proyecto concreto	Pulsar sobre un proyecto en el listado de proyectos	Se muestra la pantalla de información de ese proyecto	Si
2	Se invita a un contacto	Seleccionar la opción Invite	Se redirige a la pantalla de invitación a un contacto	Si
3	Se envían los datos	Seleccionar el botón Invite	Se redirige a la pantalla de proyectos y el contacto invitado puede visualizar el proyecto al que ha sido invitado en su listado de proyectos	Si

3. Conclusiones

Una vez realizado este Trabajo de Fin de Máster se puede concluir a través de las distintas fases de desarrollo, la necesidad de contar con un amplio equipo a la hora de acometer este tipo de proyectos software, ya que varios miembros de diferentes disciplinas pueden aportar distintos puntos de vista y calidad en cada una de sus áreas, acelerando el proceso creativo y de desarrollo.

Además se ha dado constancia del esfuerzo que requieren los desarrollos móviles actuales ya que hacen necesario en la mayoría de los casos, el consumir una API que proporcione servicios a los distintos clientes móviles. Esto requiere también una inversión en desarrollo y mantenimiento de los conocidos sistemas backend.

En terminos generales se ha logrado una correcta ejecución de los objetivos planificados al principio del proyecto, estos objetivos han sido reducidos gradualmente a medida que el proyecto adquiría mayor complejidad. Estos reajustes han sido debido a una deficiente planificación temporal al comienzo del proyecto.

3.1 Líneas de Futuro

Aplicación Híbrida

Debido al auge y al crecimiento de los frameworks de creación de aplicaciones híbridas, este proyecto podría trasladarse a uno de estos entornos y ser generado de forma rápida para múltiples plataformas, ya que el uso de la herramienta generada en este proyecto es de tareas administrativas y de gestión, no requiere un tratamiento de gráficos avanzado ni funciones nativas de los sistemas, por tanto un desarrollo híbrido sería el más adecuado.

Mobile Backend as a Service

La aparición de las plataformas que proporcionan sistemas de backend remotos abre un nuevo abanico de posibilidades a la hora de desarrollar aplicaciones tanto web como para dispositivos móviles, aplicándose a este proyecto podría ahorrarse gran parte del coste de desarrollo de la API a la que se realizan peticiones.

4. Glosario

FR: Functional Requirement, Requisito Funcional

NFR: Non-Functional Requirement, Requisito No Funcional

Stakeholders: Conjunto de sujetos con algún interés en el proyecto

Front-end:

Back-end

5. Bibliografía

1. The 'Grand Challenge' in Informatics: Engineering Software-Intensive Systems. Manfred Boy, Technische Universität of München. s.l. : IEEE Computer Society, 2006.
2. Wirsing, Martin. Report of the Beyond the Horizon Thematic Group 6 on Software Intensive Systems. 2006.
3. Volley Android [En Línea] Junio de 2016.
<https://developer.android.com/training/volley/index.html>
4. ButterKnife. [En Línea] Junio de 2016.
<http://jakewharton.github.io/butterknife/>
5. EazeGraph. [En Línea] <https://github.com/blackfizz/EazeGraph>

6. Anexos

6.1 Guía de Instalación del Servicio Web

Existen dos opciones para poner en marcha el proyecto, la api puede ser usada tanto en el servidor público de pruebas como en un servidor local:

Servidor de pruebas:

El servidor de pruebas se ubica en la dirección física <http://188.166.148.89:81/> y este ha de servir a las peticiones realizadas desde el cliente una vez configurado (ver configuración Android). Para comprobar el correcto funcionamiento del servidor se dispone de un servicio de prueba en <http://188.166.148.89:81/test> que devolverá una respuesta con código HTTP 200.

Servidor local:

Para instalar el servicio web se ha de llevar la carpeta *api* a un servidor web Apache y crear un host virtual de la siguiente manera:

```
<VirtualHost *:80>
  ServerAdmin nombredominio.dev
  DocumentRoot "<ruta a la carpeta>"
  ServerName nombredominio.dev
  ServerAlias api.nombredominio.dev
  <Directory "<ruta a la carpeta>">
    Allow from all
    Options +Indexes +ExecCGI
  </Directory>
</VirtualHost>
```

Una vez creado el host virtual, en sistemas Windows debemos añadir la siguiente línea al fichero hosts bajo la ruta C:\windows\system32\drivers\etc\, mientras que en sistemas UNIX debemos modificar el fichero /etc/hosts:

```
127.0.0.1    api.nombredominio.dev
```

6.2 Guía de Instalación de la Aplicación Android

Para compilar correctamente la aplicación Android hay que modificar el parámetro "wsBaseUrl" en el fichero strings.xml de recursos, este parámetro ha de contener la dirección IP del servicio web, ya sea un servidor instalado de forma local o un servidor remoto en el caso del servidor de pruebas.

Ejemplo: <string name="wsBaseUrl"><http://188.166.148.89:81/></string>

6.3 Guía de Instalación de la Base de Datos

Con el objetivo de crear la tablas correspondientes al modelo de datos, se han de ejecutar los siguientes comandos de creación en una base de datos MySQL.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";  
SET time_zone = "+00:00";
```

```
CREATE TABLE IF NOT EXISTS `contacts` (  
  `id` int(11) NOT NULL,  
  `users_id` int(11) NOT NULL,  
  `users_id1` int(11) NOT NULL,  
  `status` tinyint(1) NOT NULL,  
  `creationDate` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `countries` (  
  `id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `code` varchar(2) NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `families` (  
  `id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `description` varchar(255) DEFAULT NULL,  
  `Project_id` int(11) NOT NULL,  
  `creationDate` datetime NOT NULL,  
  `Partner_id` int(11) NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `messages` (  
  `id` int(11) NOT NULL,  
  `users_senderId` int(11) NOT NULL,  
  `users_receiverId` int(11) NOT NULL,  
  `message` varchar(255) NOT NULL,  
  `creationDate` datetime NOT NULL,  
  `subject` varchar(50) NOT NULL,  
  `readed` tinyint(1) NOT NULL DEFAULT '0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `partners` (  
  `id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `description` varchar(255) DEFAULT NULL,  
  `creationDate` datetime NOT NULL,  
  `Country_id` int(11) NOT NULL,  
  `image` varchar(36) DEFAULT NULL,  
  `users_id` int(11) NOT NULL
```

```
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `privileges` (  
  `id` int(11) NOT NULL,  
  `projects_id` int(11) NOT NULL,  
  `privileges_users_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `products` (  
  `id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `description` varchar(255) DEFAULT NULL,  
  `Subfamily_id` int(11) NOT NULL,  
  `creationDate` datetime NOT NULL,  
  `status` int(11) NOT NULL,  
  `Partner_id` int(11) NOT NULL,  
  `startDate` datetime NOT NULL,  
  `endDate` datetime NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=46 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `projects` (  
  `id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `description` varchar(255) DEFAULT NULL,  
  `creationDate` datetime NOT NULL,  
  `Partner_id` int(11) NOT NULL,  
  `image` varchar(36) DEFAULT NULL,  
  `users_id` int(11) NOT NULL,  
  `startDate` date DEFAULT NULL,  
  `endDate` date DEFAULT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `snapshots` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `description` varchar(255) NOT NULL,  
  `creationDate` datetime NOT NULL,  
  `data` text NOT NULL,  
  `projects_id` int(11) NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `subfamilies` (  
  `id` int(11) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `description` varchar(255) DEFAULT NULL,  
  `Family_id` int(11) NOT NULL,  
  `creationDate` datetime NOT NULL,  
  `Partner_id` int(11) NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `lastName` varchar(45) DEFAULT NULL,
  `email` varchar(100) NOT NULL,
  `password` char(60) NOT NULL,
  `creationDate` datetime NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `contacts`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_users_has_users_users4_idx`
  (`users_id1`), ADD KEY `fk_users_has_users_users3_idx` (`users_id`);
```

```
ALTER TABLE `countries`
  ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `families`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_Family_Project1_idx` (`Project_id`),
  ADD KEY `fk_families_partners1_idx` (`Partner_id`);
```

```
ALTER TABLE `messages`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_users_has_users_users2_idx`
  (`users_receiverId`), ADD KEY `fk_users_has_users_users1_idx`
  (`users_senderId`);
```

```
ALTER TABLE `partners`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_partners_countries1_idx`
  (`Country_id`), ADD KEY `fk_partners_users1_idx` (`users_id`);
```

```
ALTER TABLE `privileges`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_projects_has_users_users1_idx`
  (`privileges_users_id`), ADD KEY `fk_projects_has_users_projects1_idx`
  (`projects_id`);
```

```
ALTER TABLE `products`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_Product_Subfamily1_idx`
  (`Subfamily_id`), ADD KEY `fk_products_partners1_idx` (`Partner_id`);
```

```
ALTER TABLE `projects`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_projects_partners1_idx`
  (`Partner_id`), ADD KEY `fk_projects_users1_idx` (`users_id`);
```

```
ALTER TABLE `snapshots`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_snapshots_projects1_idx`
  (`projects_id`);
```

```
ALTER TABLE `subfamilies`
  ADD PRIMARY KEY (`id`), ADD KEY `fk_Subfamily_Family_idx` (`Family_id`),
  ADD KEY `fk_subfamilies_partners1_idx` (`Partner_id`);
```

```
ALTER TABLE `users`  
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `contacts`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
ALTER TABLE `countries`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=10;  
ALTER TABLE `families`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=15;  
ALTER TABLE `messages`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
ALTER TABLE `partners`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=16;  
ALTER TABLE `privileges`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
ALTER TABLE `products`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=46;  
ALTER TABLE `projects`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=7;  
ALTER TABLE `snapshots`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=14;  
ALTER TABLE `subfamilies`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=13;  
ALTER TABLE `users`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=11;
```

```
ALTER TABLE `contacts`  
ADD CONSTRAINT `fk_users_has_users_users3` FOREIGN KEY (`users_id`)  
REFERENCES `users` (`id`) ON DELETE NO ACTION ON UPDATE NO  
ACTION,  
ADD CONSTRAINT `fk_users_has_users_users4` FOREIGN KEY (`users_id1`)  
REFERENCES `users` (`id`) ON DELETE NO ACTION ON UPDATE NO  
ACTION;
```

```
ALTER TABLE `families`  
ADD CONSTRAINT `fk_Family_Project1` FOREIGN KEY (`Project_id`)  
REFERENCES `projects` (`id`) ON DELETE NO ACTION ON UPDATE NO  
ACTION,  
ADD CONSTRAINT `fk_families_partners1` FOREIGN KEY (`Partner_id`)  
REFERENCES `partners` (`id`) ON DELETE NO ACTION ON UPDATE NO  
ACTION;
```

```
ALTER TABLE `messages`  
ADD CONSTRAINT `fk_users_has_users_users1` FOREIGN KEY  
(`users_senderId`) REFERENCES `users` (`id`) ON DELETE NO ACTION ON  
UPDATE NO ACTION,
```

```
ADD CONSTRAINT `fk_users_has_users_users2` FOREIGN KEY
(`users_receiverId`) REFERENCES `users` (`id`) ON DELETE NO ACTION ON
UPDATE NO ACTION;
```

```
ALTER TABLE `partners`
ADD CONSTRAINT `fk_partners_countries1` FOREIGN KEY (`Country_id`)
REFERENCES `countries` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
ADD CONSTRAINT `fk_partners_users1` FOREIGN KEY (`users_id`)
REFERENCES `users` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```

```
ALTER TABLE `privileges`
ADD CONSTRAINT `fk_projects_has_users_projects1` FOREIGN KEY
(`projects_id`) REFERENCES `projects` (`id`) ON DELETE NO ACTION ON
UPDATE NO ACTION,
ADD CONSTRAINT `fk_projects_has_users_users1` FOREIGN KEY
(`privileges_users_id`) REFERENCES `users` (`id`) ON DELETE NO ACTION
ON UPDATE NO ACTION;
```

```
ALTER TABLE `products`
ADD CONSTRAINT `fk_Product_Subfamily1` FOREIGN KEY (`Subfamily_id`)
REFERENCES `subfamilies` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
ADD CONSTRAINT `fk_products_partners1` FOREIGN KEY (`Partner_id`)
REFERENCES `partners` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```

```
ALTER TABLE `projects`
ADD CONSTRAINT `fk_projects_partners1` FOREIGN KEY (`Partner_id`)
REFERENCES `partners` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
ADD CONSTRAINT `fk_projects_users1` FOREIGN KEY (`users_id`)
REFERENCES `users` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```

```
ALTER TABLE `snapshots`
ADD CONSTRAINT `fk_snapshots_projects1` FOREIGN KEY (`projects_id`)
REFERENCES `projects` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```

```
ALTER TABLE `subfamilies`
ADD CONSTRAINT `fk_Subfamily_Family` FOREIGN KEY (`Family_id`)
REFERENCES `families` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
ADD CONSTRAINT `fk_subfamilies_partners1` FOREIGN KEY (`Partner_id`)
REFERENCES `partners` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION;
```