

Mensajería directa orientada al uso inteligente de la información



Andrés Roco Conde

Universidad de Extremadura

2013/14

Resumen

Actualmente existen una gran cantidad de aplicaciones móviles que permiten la comunicación entre diferentes usuarios. Estas aplicaciones, como por ejemplo Whatsapp, permiten crear grupos con una gran cantidad de participantes para poder hablar de diversos temas. El principal problema que provoca es que si un participante desea plantear una duda en un grupo, al principio de esta conversación cada usuario irá aportando sus conocimientos que posiblemente sean totalmente diferentes al resto de los usuarios. A lo largo de la conversación, probablemente se iniciará una discusión entre los diferentes usuarios, donde cada uno intentará defender su opinión. Finalmente nos encontraríamos con un grupo de conversación con un gran porcentaje de mensajes que no resolverían la duda del usuario y que posiblemente el tema de conversación se haya desviado completamente del inicio. Esta situación provocaría una gran pérdida de tiempo por parte del usuario, debido a que finalmente la duda seguiría en pie.

El objetivo que se propone es ofrecer una aplicación, orientada al ámbito académico, donde los usuarios puedan lanzar una conversación con un tema específico, como por ejemplo una discusión sobre las Tecnologías Software.

Esta aplicación deberá decidir automáticamente quienes serán los participantes de esta conversación, que deberán ser los usuarios expertos en dicho tema y el usuario que plantea la cuestión. La selección de estos expertos se calculará en función de las conversaciones en las que participan y principalmente en su contenido para poder ofrecer al usuario la mejor solución.

En este Trabajo de Fin de Grado se pretende conseguir dichos objetivos con la aplicación del futuro: ***JET!***

Esta plataforma determina automáticamente los expertos de un tema concreto, permitiendo la creación de canales temáticos y ofreciendo al usuario la oportunidad de preguntar numerosas

dudas con la intención de obtener, por parte de los expertos, la mejor solución de manera anónima.

Es un avance que actualmente no tiene ninguna aplicación de mensajería y facilita la adquisición de nuevos conocimientos.

Estos canales serán totalmente accesibles para el resto de los usuarios, que les permitirán enriquecerse de las cuestiones planteadas, pero con el condicionante de que no pueden transmitir sus opiniones evitando convertir este canal en un foro ajeno al tema de conversación.

Gracias a este análisis, la plataforma ofrece un servicio de sugerencia de amigos que le permite al usuario comunicarse con aquellos que tengan temas de conversación semejantes permitiendo una mayor satisfacción.

Finalmente, los buenos resultados obtenidos han permitido presentar un artículo científico en una conferencia internacional demostrando que **JET!** es una aplicación única y totalmente orientada a las necesidades de los usuarios.

Índice

Mensajería directa orientada al uso inteligente de la información	1
Andrés Roco Conde	1
1- Introducción	11
1.1 Organización de la documentación	11
1.2 Situación	13
1.3 Motivación	14
1.4 Alcance	16
1.5 Objetivos	18
1.5.1 Objetivos desglosados	18
1.6 Metodología de desarrollo	20
1.6.1 Metodología elegida	20
1.7 Productos similares	21
1.7.1 WhatsApp	21
1.7.2 Telegram	23
1.7.3 Line	25
1.7.4 Mi propuesta	27
2- Análisis inicial	28
2.1 Análisis de requerimientos	29
2.1.1 Requerimientos funcionales	29
2.1.2 Requerimientos no funcionales	30

2.2 Planificación	30
2.3 Análisis tecnológico	31
2.3.1 Dispositivo móvil	31
2.3.2 Plataforma de servicios web	36
2.3.3 Plataforma Web	39
3-Especificación	41
3.1 Casos de uso	41
3.1.1 Actores	42
3.1.2 Diagrama de casos de uso	42
3.1.3 Descripción de casos de uso	43
3.2 Modelo conceptual	54
3.2.1 Academic Catalogs	57
3.2.2 Users	58
3.2.3 Connection Conversation-Academic	59
3.2.4 Login	60
3.2.5 Conversations	60
3.2.6 Analysis Message	61
3.2.7 Channel	62
3.3 Modelo de comportamiento	63
4-Diseño	77
4.1 Arquitectura física	78
4.2 Tecnologías usadas	81
4.2.1 Android	81
4.2.2 WebRatio	86
4.3.3 Comunicación entre la aplicación móvil y los servicios web	87

4.3 Arquitectura lógica	90
4.3.1 Interfaz gráfica	90
5-Implementación	94
5.1 Web Service	95
5.1.1 Requisitos necesarios	95
5.1.2 Desarrollo de las funcionalidades principales	100
5.1.3 Despliegue	118
5.2 Aplicación Web	119
5.2.1 Requisitos necesarios	120
5.2.2 Desarrollo de las funcionalidades principales	123
5.2.3 Despliegue	131
6- Pruebas	131
7- Ampliaciones y mejoras	134
7.1 Ampliaciones	134
7.1.1 Aplicación web	134
7.1.2 Web Service	134
7.2 Mejoras	135
7.2.1 Aplicación web	135
7.2.2 Web Service	135
8- Conclusión	136
9- Agradecimientos	137
10- Bibliografía	138

1- Introducción

Este documento pretende explicar todo el desarrollo del proyecto fin de grado **JET!** razonando al detalle cada decisión tomada para llevar a cabo el proyecto de la mejor manera posible y para que pueda quedar perfectamente reflejado en este documento.

1.1 Organización de la documentación

La documentación se ha estructurado de la siguiente manera:

En el primer capítulo, la introducción, se presenta el proyecto con las motivaciones y objetivos que se pretenden conseguir al desarrollar dicho proyecto así como la metodología que se va a llevar a cabo durante todo el desarrollo.

También en esta sección se realizará una comparativa con otras aplicaciones ya existentes en el mercado y cómo JET! consigue superar a todas estas aplicaciones proporcionando nuevos servicios orientados a los usuarios.

Seguidamente se realiza un análisis inicial de todos los elementos necesarios al empezar cualquier desarrollo de estas características.

Empezaremos explicando el análisis de requerimientos donde analizaremos que requisitos tenemos que cumplir una vez finalizado el proyecto donde podemos diferenciar los requerimientos funcionales y los requerimientos no funcionales.

Para finalizar este análisis inicial, haremos un análisis de la tecnología usada para seleccionar cual es la que mejor se adapta para este proyecto en concreto.

A continuación, explicaremos al detalle todas las fases mas comunes en el desarrollo de un proyecto software.

Se pueden observar todas las fases en la siguiente figura.

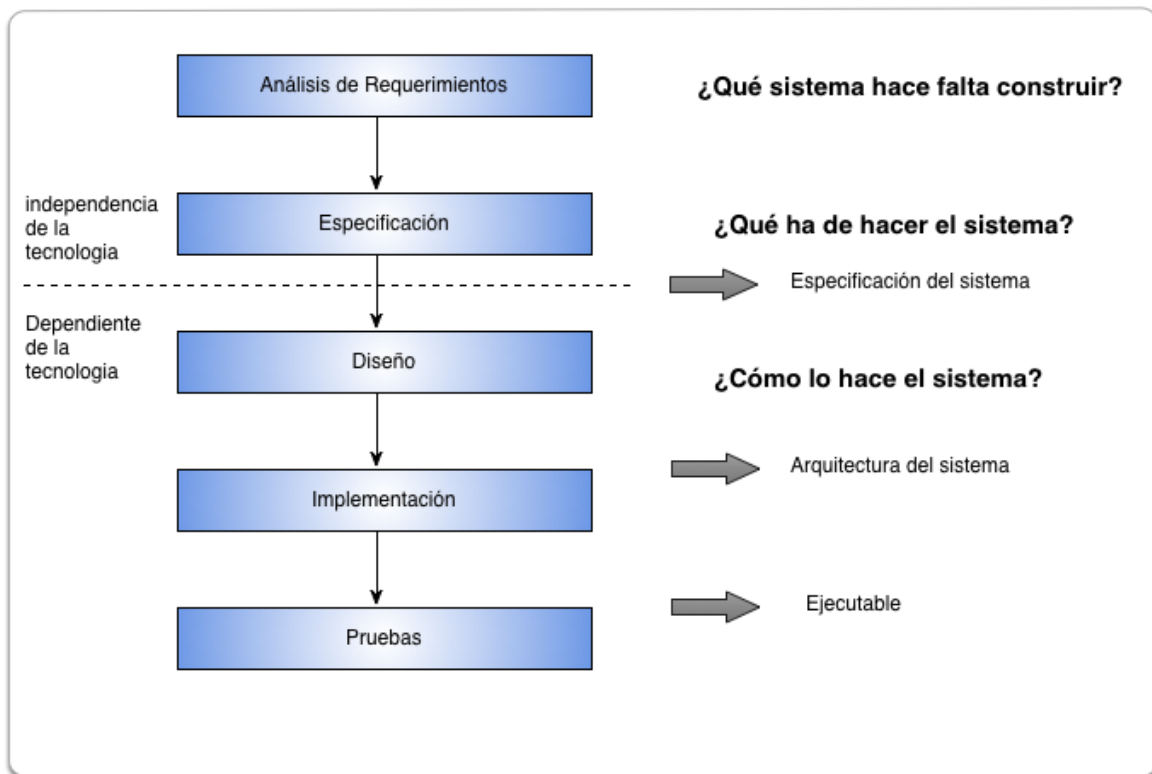


Figura 1.1: Fases de desarrollo software

En primer lugar se presenta la especificación que incluye los casos de uso de cada funcionalidad que ofrece este sistema, así como su modelo conceptual y el modelo de comportamiento.

Después nos encontramos con la fase de diseño donde explicaremos la arquitectura física de nuestro proyecto, las tecnologías elegidas para poder desarrollar nuestra plataforma y la arquitectura lógica del sistema que describe como están diseñado cada subsistema de nuestro proyecto y como se relacionan los distintos componentes.

Finalmente se explicará la implementación y las pruebas de nuestro proyecto, es decir, como se han desarrollado cada una de las funcionalidades de nuestro proyecto y como se han realizado las pruebas para poder encontrar el mayor número de errores existentes y poderlos solucionar.

Una vez descrito las etapas más comunes en el desarrollo de software, se da un repaso a toda la planificación inicialmente indicada y comprobando si definitivamente se han conseguido todos los objetivos previstos para este proyecto.

También se explicará como ha sido la experiencia al realizar este proyecto y futuras ampliaciones y mejoras del proyecto que permitan que esta plataforma se convierta en una de las mas grandes en el sector de las comunicaciones.

Finalmente, nos encontramos con la bibliografía utilizada para conseguir dichos objetivos

1.2 Situación

A lo largo del tiempo han ido mejorando de manera exponencial las herramientas que facilitan el desarrollo de un proyecto software. Hace unos años realizar un proyecto de estas características en el tiempo estipulado sería algo imposible debido a la escasez de información.

Hoy en día, con internet, tenemos al alcance toda la información necesaria para poder conseguir cualquier objetivo software y a lo largo de la carrera se han realizado numerosos proyectos que permiten definir el tiempo que va a ser necesario para conseguir nuestro objetivos.

El tiempo es un concepto fundamental en el desarrollo de un proyecto software porque es un recurso que puede variar sustancialmente y una variación negativa puede producirse en pérdidas de beneficios. Para poder evitar este desastre es necesario saber el tiempo que se dispone y que objetivos deben cumplirse y para ello debe realizarse un análisis inicial.

En la siguiente figura se puede mostrar el porcentaje del tiempo que es necesario para cada fase de un desarrollo de un proyecto software.

● Análisis ● Especificación ● Diseño ● Código ● Test
● Mantenimiento

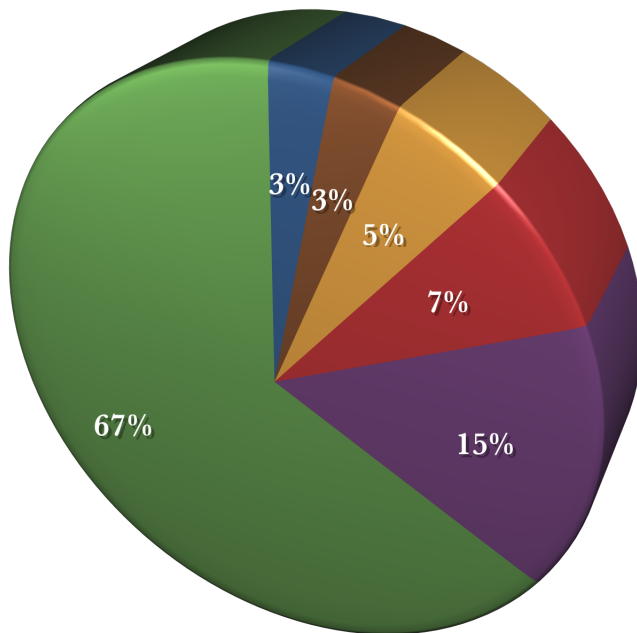


Figura 1.2: Gráfica porcentaje de tiempo para cada fase

1.3 Motivación

Actualmente las principales empresas del mundo se centran en el desarrollo de aplicaciones móviles y si concretamos más se centran en las aplicaciones móviles que permiten el intercambio de información entre diferente usuarios, bien sea mensajes, fotos, vídeos...

Hoy en día un dispositivo móvil es la identidad de una persona, es donde se encuentra toda la información necesaria para conocer a un usuario y así poder proporcionarle los mejores servicios que le puedan ofrecer ciertas facilidades en la vida cotidiana.

El crecimiento de estos dispositivos es exponencial y nunca se imaginaba que fuera a llegar a tanto y pasar de ser un capricho a ser una necesidad donde muchas personas no podrían vivir sin el.

En la siguiente figura se puede observar como aumentan el número de ventas de los dispositivos móviles, las tablet y los ordenadores.

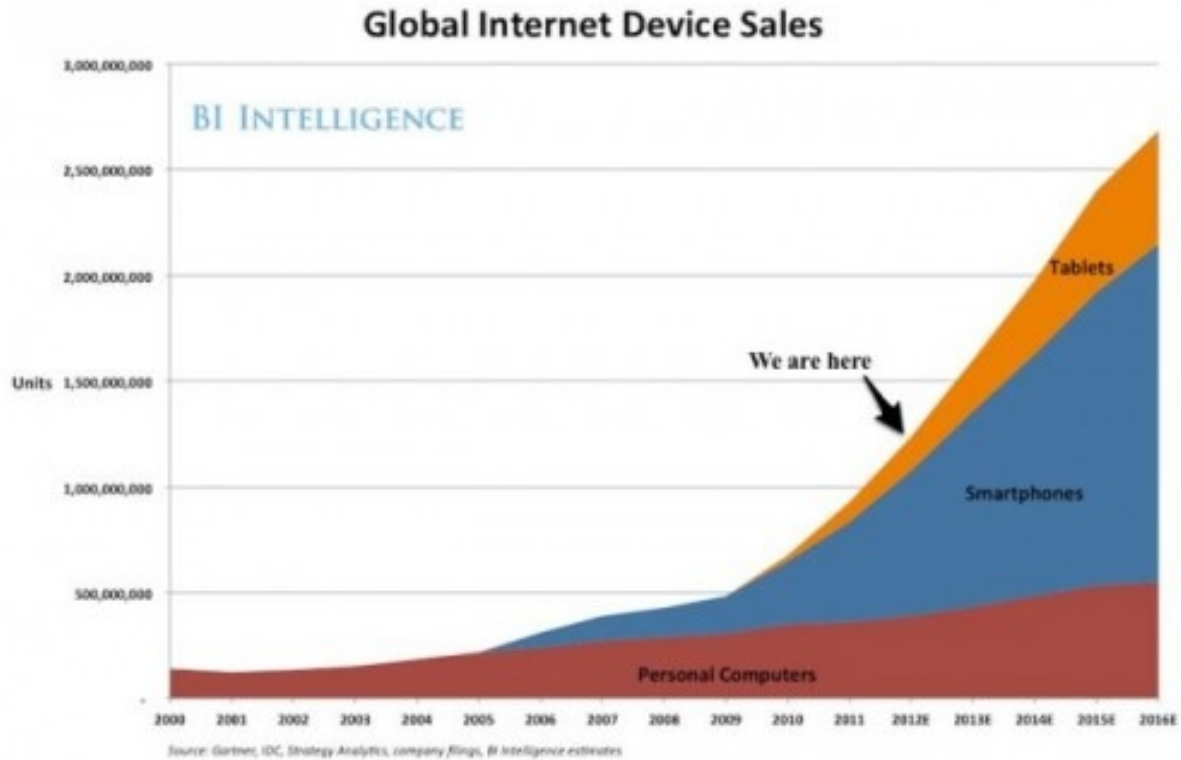


Figura 1.3: Gráfica del crecimiento de uso de los diferentes dispositivos.

Como se puede comprobar el crecimiento es inimaginable y dentro de unos años utilizaremos los dispositivos como un identificador de una persona donde podremos comprar todo desde el móvil, viajar identificándonos con el móvil, manejar objetos con el móvil en definitiva hacer todo con el móvil.

El desarrollo de esta aplicación me permite involucrarme en este mundo y poder aportar mis ideas a la sociedad facilitando la vida de cualquier usuario.

Con este proyecto los usuarios podrán conseguir todo tipo de información en tiempo real y ser proporcionada por otros usuarios de la aplicación sin tener que ir a fuentes externas que no aseguran que sea correcta esta información, en cambio, con esta plataforma los usuarios podrán discutir sobre los diferentes temas.

También es muy productivo realizar un proyecto totalmente funcional y llevarlo a cabo en una empresa como GLOIN donde he podido aprender nuevas metodologías para el trabajo en grupo y nuevos conocimiento por parte de personas que son profesionales en este mundo.

1.4 Alcance

Una de las primeras tareas de todo proyecto es la definición del alcance, es decir, delimitar las funcionalidades a desarrollar para poder cumplir con todos los objetivos al finalizar el proyecto.

Si esta fase no se hace bien puede producir problemas en la planificación, en el tiempo, en el propio funcionamiento del proyecto y en el control de errores.

Es necesario definir un alcance claro para **JET!** porque puede convertirse en un proyecto interminable y sobre todo cuando nos encontramos con un proyecto de estas características donde se le pueden añadir infinitas funciones a un aplicación móvil.

Con este proyecto se pretende ofrecer una serie de servicios al usuario que le permitan mejorar sus conocimientos gracias a otros usuarios de la propia aplicación.

Para ello en este proyecto se va a desarrollar la parte servidora de dicha aplicación, que estará alojado en un servidor de la universidad para poder estar accesible para todos los usuarios y dispositivos móviles.

En un principio este servidor fue desplegado en una Raspberry Pi localizada en el propio hogar pero las características del mismo y la conexión a internet no permitieron seguir adelante con este dispositivo debido a la carga de trabajo que debía soportar, más adelante en la parte de implementación se explicará con detalle porque este dispositivo no puede usarse como servidor para unos servicios web de estas características.

Se pretende que este servidor ofrezca servicios para llevar a cabo un registro de usuario con todo lo que conlleva y más en este caso que se utiliza una plataforma externa como es Nimbees, debe permitir poder seleccionar las características de un usuario respecto a sus estudios para poderlos registrar en una carrera, en un grado y en numerosas asignaturas.

Otro servicio que debe ofrecer es la opción de poder crear canales y suscribirse en ellos pero, ¿Que son los canales? De manera abstracta los canales son agrupaciones de usuarios que tengan algo en común por ejemplo una asignatura, una carrera, una localización, un tema...

Una vez que el usuario se encuentra en un canal el servidor ofrece un servicio al cliente que le permite crear conversaciones dentro de dicho canal.

Estas conversaciones, de manera abstracta, se crean para definir subgrupos dentro de un canal para permitirle a los usuarios, por ejemplo crear una conversación, dentro de un canal de una asignatura, que haga referencia a las dudas sobre un proyecto que se tenga que realizar.

Dentro de estas conversaciones es donde se debe poder enviar mensajes que serán analizados para una posterior búsqueda de información referente a la temática.

La **función principal de este proyecto** es la creación de canales según una temática permitiendo ofrecer un nuevo servicio inexistente actualmente a los usuarios, que permite agrupar diversos usuarios de manera totalmente anónima, que entiendan de un tema en concreto y poder ayudar a aquellas personas que necesitan de dicha información, pues **JET!** lo conseguirá



Figura 1.4: Imagen representativa del entusiasmo.

Por último, para facilitar la gestión al administrador, se va a desarrollar una página web sencilla que le permita añadir nuevas universidades, nuevos grados y nuevas asignaturas.

Respecto a la temática, el vocabulario está en constante cambio y pueden aparecer nuevos temas, pues en esta página web le ofrecemos la posibilidad al administrador de añadir nuevos temas y nuevo vocabulario referente a dicho tema.

1.5 Objetivos

El objetivo principal de **JET!** es permitir **la comunicación entre diversos usuarios** en tiempo real. Estas comunicaciones se llevarán a cabo cumpliendo ciertos requisitos como una localización específica, una asignatura, un grado, una universidad o un tema en concreto.

Para ello es necesario conseguir los siguientes 4 objetivos:

- Desarrollar una aplicación móvil.
- Desarrollar una aplicación web.
- Desarrollar un Web Service.
- Desarrollar una conexión con Nimbees.

1.5.1 Objetivos desglosados

- De la aplicación móvil:
 - * Registro de un usuario.
 - * Selección de universidad.
 - * Selección de grado.
 - * Suscribirse a una asignatura y a su vez a un canal.
 - * Crear un nuevo canal según la localización.
 - * Crear un nuevo canal con los expertos de un tema concreto.
 - * Poder crear una conversación dentro de un canal.
 - * Poder observar las medallas del usuario
- De la aplicación web:
 - * Crear una nueva universidad.
 - * Crear un nuevo grado.
 - * Crear una nueva asignatura.
 - * Crear un nuevo tema.

- * Crear nuevo vocabulario para un tema en concreto.

● Del Webservice:

- * Almacenar un registro de un usuario.
- * Obtener todas las universidades.
- * Obtener todos los grados.
- * Obtener todas las asignaturas.
- * Suscribirse a un usuario a un canal.
- * Crear un canal según un tipo de localización.
- * Crear un canal según unas coordenadas.
- * Añadir un nuevo amigo.
- * Mostrar sugerencias de amistades.
- * Crear una nueva conversación.
- * Crear un nuevo mensaje.
- * Analizar toda la información relevante a la mensajería de un usuario.
- * Gestionar el uso de certificados.
- * Gestionar el uso de autenticación.
- * Poder enviar mensajes custom que sean analizados por la aplicación.
- * Gestionar las localizaciones de un usuario.
- * Actualizar la información de un usuario.

● Desarrollar una conexión con Nimbees

- * Crear una nueva aplicación en la plataforma Nimbees.
- * Registrar un dispositivo móvil en Nimbees.
- * Enviar los mensajes a Nimbees para que esta plataforma se encargue de enviarlos a los dispositivos a través de la tecnología push.
- * Enviar mensajes custom a través de Nimbees.

1.6 Metodología de desarrollo

Existen muchas metodologías para llevar a cabo un proyecto software, y dependiendo las características del proyecto se utiliza uno u otra. En esta sección se explicará cuales son las metodologías más usadas y cual de ellas se adapta más a nuestro proyecto.

- **Modelo en cascada:** Es una metodología que se centra en una serie de fases que se desarrollan de manera secuencial. Tras finalizar cada fase se lleva a cabo un control para verificar que el proyecto, en ese punto, está en perfectas condiciones para pasar a la siguiente fase. En el caso de encontrar algún error en la fase de pruebas provocaría un rediseño y una nueva programación del error ocurrido.
- **Prototipado:** Esta metodología consiste en desarrollar un prototipo del proyecto en poco tiempo y utilizando los mínimos recursos posibles y una vez que el prototipo este aprobado es cuando se inicia el verdadero proyecto. Esta metodología puede provocar la pérdida de la calidad del proyecto al tener la necesidad de tener un prototipo en un tiempo limitado y sin dedicarlo los recursos necesarios.
- **Incremental:** Esta metodología consiste en desarrollar partes de un proyecto una vez que ya han sido revisadas, es decir, se van desarrollando “miniproyectos” que una vez finalizado se desarrolla un nuevo “miniproyecto” a partir del anterior permitiendo poder retrasar los requisitos que pueden ir definiéndose a lo largo del proceso según las necesidades.
- **Espiral:** Esta metodología conforma una espiral en la que cada bucle representa una serie de actividades que se eligen en función del análisis de riesgo.
- **Metodologías ágiles:** Esta metodología consiste en desarrollar un proyecto en iteraciones que cada una se compone del ciclo de vida completo (Requisitos, Especificación...) y al final de cada una se realiza una detallada evaluación. Estas metodologías están en constante contacto con el cliente para cumplir completamente el contrato previamente definido.

1.6.1 Metodología elegida

En este proyecto desde el primer momento se tenían claro cuales eran los requisitos que se debían cumplir pero a lo largo del proceso iban apareciendo nuevos requisitos que mejoraban

el funcionamiento de la plataforma o simplemente añadía una nueva funcionalidad que fuera de gran interés para el usuario.

Con esta información, el proyecto se llevará a cabo con una metodología incremental que nos permite poder definir nuevos requisitos según las nuevas necesidades que iban apareciendo.

Estas definiciones se llevaban a cabo semanalmente tras analizar todo el progreso del proyecto y los nuevos servicios que pueden mejorar la plataforma.

También puede identificarse como una metodología ágil porque semanalmente se desarrollaba un proyecto desde la primera fase del ciclo de desarrollo analizando todos los posibles errores o funcionalidades que podrían ser mejoradas.

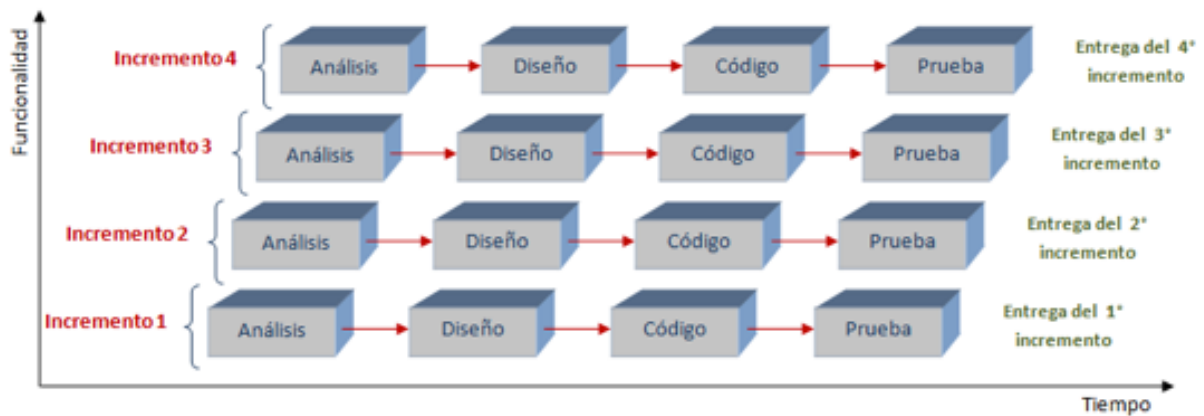


Figura 1.5: Modelo incremental.

1.7 Productos similares

En este apartado se explicará diferentes aplicaciones ya existentes en el mercado y con gran auge en la sociedad así como sus beneficios e inconvenientes y como nuestro proyecto puede destacar ante aplicaciones de gran nivel.

1.7.1 *WhatsApp*

Actualmente es la aplicación móvil de mensajería más usada por la sociedad y fue la primera que dio el “boom” en este sector que posteriormente fueron apareciendo nuevas aplicaciones semejantes a esta.

Actualmente siguen obteniendo nuevos usuarios de todo el mundo donde alcanza cifras de 430 millones de usuarios y va incrementando de inimaginable.

Cuando surgió esta aplicación estaba centrada en aquel sector de usuarios que deseaban ahorrar dinero en el envío de sms y actualmente se ha convertido en un aplicación necesaria en cualquier dispositivo móvil.

Además se caracteriza por ser un aplicación sencilla y multiplataforma lo que ha permitido ser tan atractivo para los usuarios.

En la siguiente figura, perteneciente a “Statista” se puede observar como ha aumentado el número de usuarios de esta aplicación.

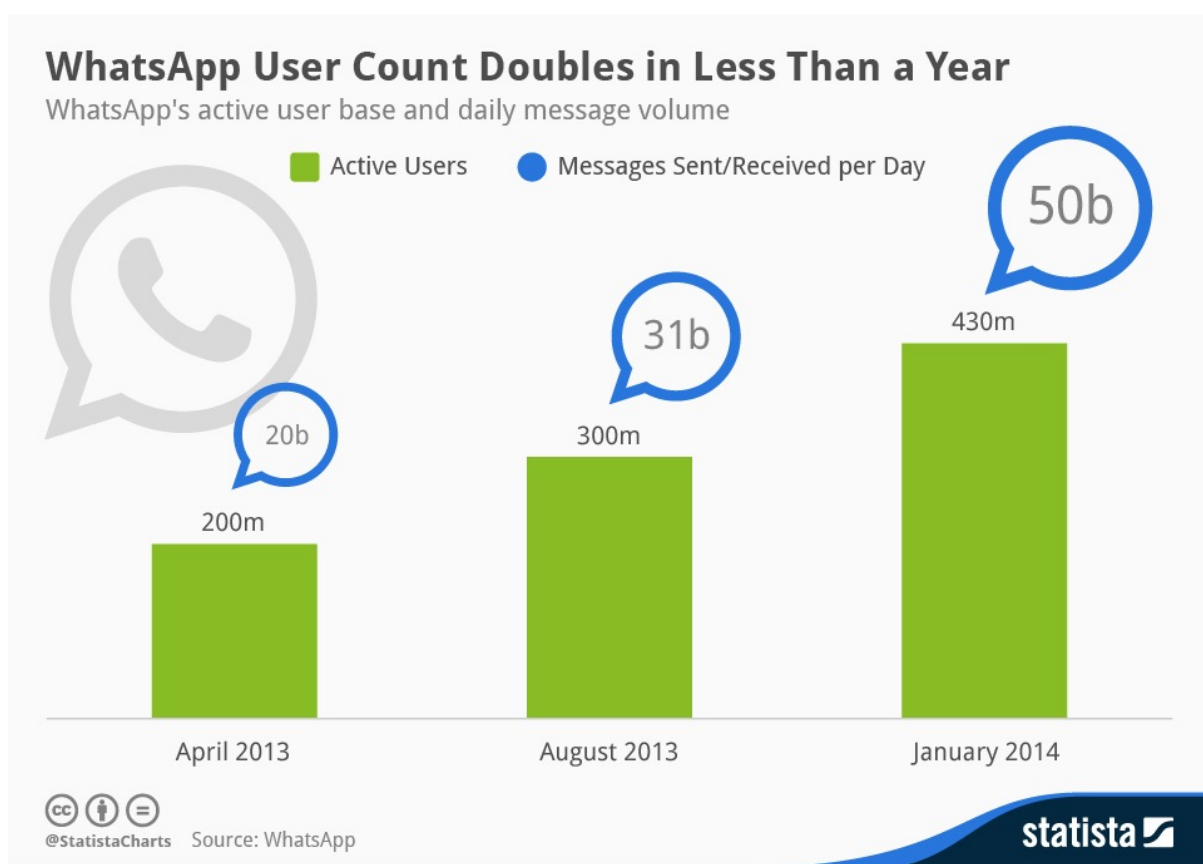


Figura 1.6: Gráfica de uso actual de la aplicación WhatsApp

A continuación se definirán los beneficios e inconvenientes de esta aplicación.

Ventajas

- Número de usuarios: Como se ha explicado anteriormente es una cifra bastante golosa y tiene casi controlado el sector de las comunicaciones a través de dispositivos móviles.

- Sencilla y simple: Esta aplicación tiene las funciones justas pero bien identificadas donde los nuevos usuarios puedan conocer su funcionamiento sin necesidad de un manual.
- Transferencia de archivos: Esta aplicación permite compartir diferentes archivos (imágenes, sonidos, vídeos...) con otros usuarios.
- No pagas tarifa por mensaje enviado: El envío de mensajes no tiene un coste adicional al precio de la aplicación.
- Multiplataforma: Esta aplicación está disponible para todos los grandes sistemas operativos para dispositivos móviles.

Desventajas

- Privativa: Esta aplicación es de pago y muchos usuarios buscan servicios alternativos a los sms que sean gratuitos.
- Problemas de seguridad: Esta aplicación desde que surgió ha llevado encima muchos fallos de seguridad al comunicar la información sin cifrar.
- Necesidad de introducir teléfono móvil: Para poder registrarse en esta aplicación es necesario registrar un teléfono para poder utilizar sus servicios.

1.7.2 Telegram

Telegram es la segunda aplicación con mayor éxito en el mercado siendo prácticamente el reflejo de whatsapp pero con bastantes avances que la convierten en superior a whatsapp aunque el número de usuarios no lo demuestre.

Esta aplicación tiene grandes mejoras en seguridad y para ello diseñaron un nuevo protocolo MTProto que cuenta con una encriptación AES de 256bits.

Es aplicación a diferencia de otras es que permite crear conversaciones seguras las cuales no son almacenadas en el servidor y únicamente se almacenan en los dispositivos donde los usuarios son responsables de esta información.

Otra característica es todos los clientes oficiales son de código abierto aunque el código del servidor será cerrado por cuestiones de seguridad.

En la siguiente gráfica se puede observar como ha ido creciendo telegram a lo largo del tiempo.

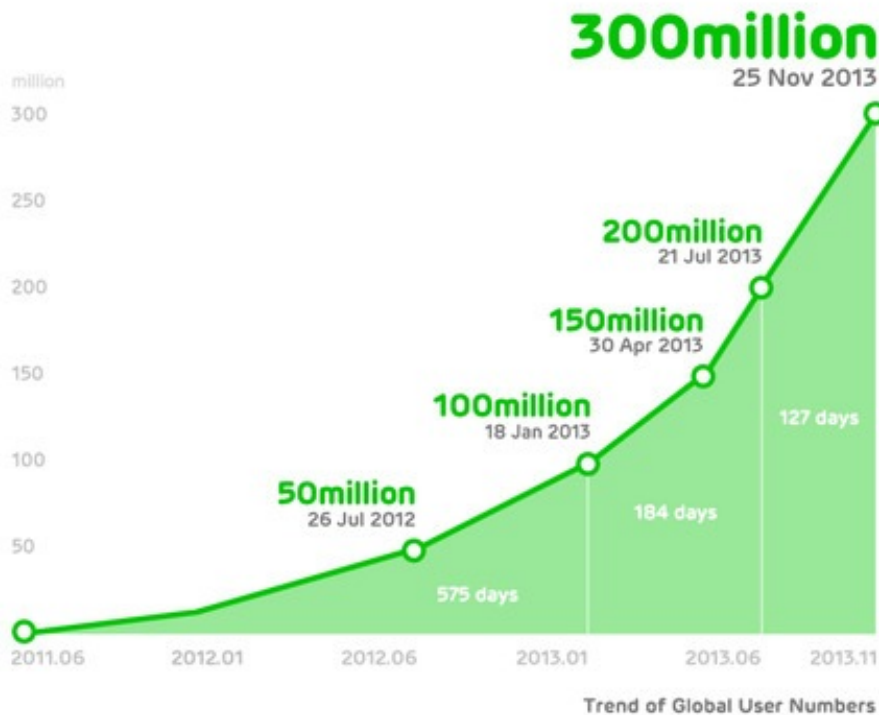


Figura 1.7: Gráfica de uso de la aplicación telegram

A continuación se detallarán las ventajas y desventajas de esta aplicación:

Ventajas

- Mensajería segura: Como he explicado anteriormente toda la información va encriptada y permite crear conversaciones privadas donde el servidor no almacena nada de la conversación.

- Grupos de gran tamaño: En esta aplicación pueden crearse grupos de hasta 200 usuarios .
- Transferencia de archivos de gran tamaño: Los usuarios pueden compartir archivos de hasta 1 GB.
- Clientes abierto: Cada desarrollador puede crear su propio cliente utilizando una API proporcionada por telegram pero no de la parte servidora.
- Gratuito: Esta aplicación es totalmente gratuita.
- Disponibles para ordenadores: A diferencia de whatsapp existe una aplicación para los ordenadores siendo compatible con los grandes sistemas operativos.
- Autodestrucción: Se puede activar la destrucción de una conversación en un intervalo de tiempo específico.

Desventaja

- La baja cantidad de usuarios: Como se ha mostrado en la gráfica esta aplicación a lo largo del tiempo va creciendo a gran velocidad el número de usuarios pero aun no puede competir con whatsapp.

1.7.3 Line

Line es una aplicación japonesa de mensajería móvil semejantes a las explicadas anteriormente.

Una de las características de esta aplicación son las pegatinas, básicamente son emoticonos con un tamaño bastante mayor que reflejan comportamientos más expresivos pero parte de estos son de pago y otro gratuitos.

También esta aplicación permite realizar llamadas gratuitas cosa que ni whatsapp ni telegram lo permiten.

Para poderte registrar en esta aplicación no es necesario revelar tu número de teléfono para poder tener una conversación con otros usuarios.

Un elemento bastante divertido es que permite agregar amigos simplemente con agitar un móvil cerca de otro y automáticamente se agregan como contactos en la aplicación.

A continuación se puede observar una gráfica con el número de usuarios registrados en la aplicación.

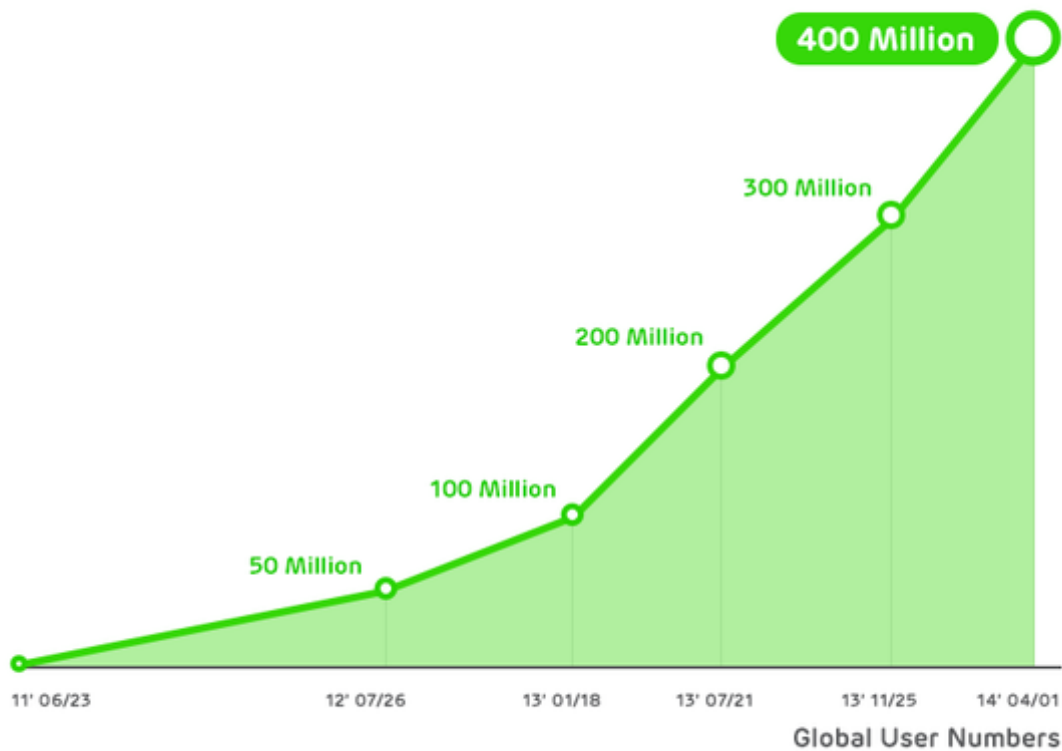


Figura 1.8: Gráfica de uso de la aplicación Line.

Ventajas

- Llamadas gratuitas: Esta aplicación permite realizar llamadas totalmente gratuitas a través de internet.
- Stikers: Emoticonos de mayor tamaño, mayor colorido y mayor expresividad que los básicos del resto de aplicaciones.

- Compatible con ordenadores: Se puede instalar en los grandes sistemas operativos de ordenadores permitiendo comunicarse sin tener un dispositivo móvil.

Inconvenientes

- Número de usuarios: Al igual que telegram tiene el inconveniente de tener un menor número de usuarios que whatsapp .
- Publicidad: Line constantemente te ofrece publicidad para que compres emoticonos en la aplicación.

1.7.4 Mi propuesta

Estas aplicaciones proporcionan lo necesario para una comunicación entre diferentes usuarios pero faltan diversos servicios que permitan una conversación más concreta y con un objetivo definido más allá de una simple comunicación. En las aplicaciones existentes los usuarios se crean un grupo compuesto por varios usuarios donde se hablan de diversos temas y esto puede causar bastante confusión entre los usuarios.

Otro defecto de las aplicaciones actuales es que no aprovechan la información para proporcionar servicios a los usuarios que les permita obtener nuevos conocimientos, sin necesidad de volverse loco buscando en internet información que ha podido ser escrita hace años y encontrar datos incoherentes.

Mi propuesta es **JET!**, una aplicación de mensajería inteligente que esta totalmente orientada a mejorar la vida de los usuarios. Se acabaron las confusiones en la creación de grupos de las aplicaciones actualmente en el mercado, el futuro son los canales donde un usuario puede definir que participantes va a tener o la opción de suscribir a los participantes según un criterio.

Un componente destacable de esta aplicación es la posibilidad de crear canales basándose en una temática, es decir, con esta aplicación si tienes un duda o quieres obtener conocimientos sobre un tema concreto o simplemente quieres debatir con usuarios sobre algún tema esta aplicación lo permite y con la sencillez de indicar el tema y la propia aplicación define un nuevo canal con aquellos usuarios que tengan mayor nivel de conocimiento de ese tema, por ejemplo, si una persona necesita realizar un trabajo sobre el antiguo Egipto no

necesita meterse en ningún buscador para buscar esta información simplemente con indicar este tema en la aplicación se crea un canal con los mayores expertos para poder debatir sobre esta temática.

Esta aplicación a diferencia de otras, no solo se centra en la comunicación, sino que permite desarrollar la competitividad entre los usuarios con adquisición de medallas que se consiguen cumpliendo unos requisitos como por ejemplo ser la persona que más a interactuado con la aplicación en un mes.

Esta aplicación puede cambiar completamente el concepto de las aplicaciones de mensajería simples y aburridas con la aparición de **JET!** la aplicación inteligente.

2- Análisis Inicial

En este capítulo se detalla el análisis realizado durante la fase inicial del proyecto. En primer lugar se realizó un análisis de requerimientos basados en los objetivos del proyecto y que nos definen como deberá ser nuestro sistema al finalizar dicho desarrollo.

Finalmente se realizó un análisis de las tecnologías que se podían utilizar y así justificar las tecnologías usadas para llevar a cabo el proyecto.

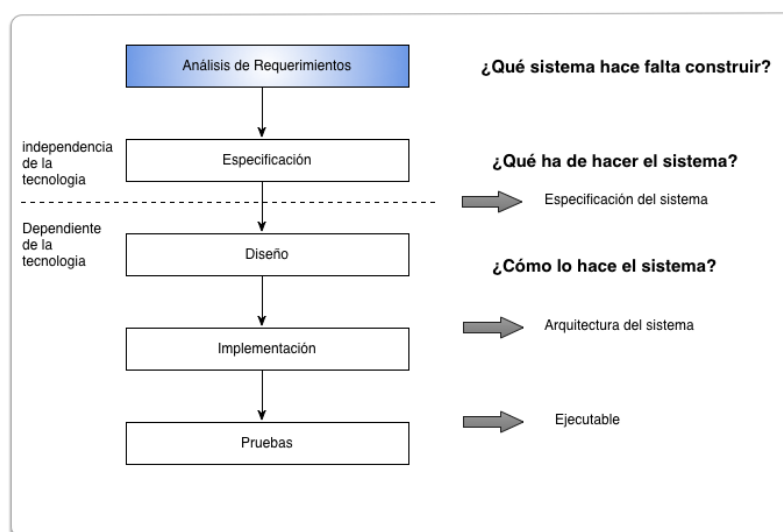


Figura 2.1: Fase de análisis de Requerimientos

2.1 Análisis de requerimientos

2.1.1 *Requerimientos funcionales*

- Poder registrar un usuario.
- Obtener la información de un usuario.
- Acceso de un usuario registrado.
- Creación de nuevas universidades.
- Creación de nuevos grados.
- Creación de nuevas asignaturas.
- Visualizar todas las universidades.
- Visualizar todos los grados.
- Visualizar todas las asignaturas.
- Creación de un canal.
- Creación de una conversación.
- Creación de un nuevo mensaje.
- Visualizar los canales.
- Visualizar las conversaciones.
- Visualizar los mensajes.
- Visualizar datos del usuario más influyente.
- Visualizar datos del usuario más influido.
- Visualizar datos del intervalo de tiempo que más interactúa con la aplicación.
- Visualizar datos de a que hora se duerme el usuario.
- Visualizar datos de a que hora se despierta el usuario.
- Visualizar datos de que temas es el más hablado por el usuario.
- Creación de un nuevo tema léxico.
- Creación de un nuevo vocabulario.
- Mostrar sugerencias de amistad.
- Añadir nuevas amistades.

2.1.2 Requerimientos no funcionales

Es necesario la elaboración de un manual de usuario a pesar de ser una aplicación sencilla.

- Usabilidad: Debe tener una interfaz intuitiva y simple.
- Escalabilidad: Debe ser posible añadir nuevas funcionalidades de forma sencilla.
- Utilidad: Las funcionalidades de la aplicación han de ser útiles para el usuario.
- Desplegar en Servidor web para permitir el acceso constante al los servicios web.
- Diseño: Un diseño atractivo de la aplicación.
- Seguridad: Permitir comunicaciones seguras a través de la red.
- Utilización de Nimbees, plataforma de la empresa Gloin.
- Compatibilidad con navegadores como internet explorer, safari,firefox y opera.

2.2 Planificación

Una vez definido el alcance del proyecto y sus requisitos es necesario elaborar una planificación que se lleva a cabo en la fase inicial del proyecto donde se incluyen todas las tareas a llevar a cabo para cumplir con éxito los objetivos marcados.

El tiempo invertido en este proyecto se puede dividir en dos grandes secciones:

Documentación: En este documento se encuentra toda la información relacionada con el proyecto y su elaboración está presente desde el inicio del proyecto con el objetivo de justificar todas la decisiones tomadas a lo largo del desarrollo del proyecto como la tecnología utilizada, las herramientas usadas para el desarrollo y así como toda la documentación técnica que facilita el mantenimiento de dicho proyecto.

Desarrollo tanto de los servicios web como de la página web utilizando las diversas librerías y la definición de las pruebas necesarios para comprobar el correcto funcionamiento. Este desarrollo siempre tiene que estar relacionado con la aplicación móvil que será la encargada

de consumir dichos servicios y para ello debe existir un previo contrato entre los dos desarrollos.

2.3 Análisis tecnológico

Dados los diferentes requisitos anteriormente definidos es necesario seleccionar que tecnología vamos a utilizar para conseguir nuestros objetivos y para ellos necesitamos realizar una serie de comparaciones para elegir la tecnología más adecuada.

2.3.1 Dispositivo móvil

En los últimos años los dispositivos móviles han obtenido una gran relevancia en el mundo actual.

Antiguamente tener un móvil se consideraba como un capricho pero actualmente se considera que tener un dispositivo móvil es una necesidad y muchas personas y empresas no se pueden despegar de dicho dispositivo.

Actualmente se invierte mucho dinero en mejorar estos dispositivos proporcionando nuevas tecnologías que permitan facilitar la vida a las personas y ofrecer nuevos métodos de entretenimiento como cámara de fotos, GPS, Bluetooth, wifi y las redes 3G y 4G.

El “boom” de estos dispositivos móviles se inició con la aparición del primer iPhone de la compañía de Apple que posteriormente apareció el sistema operativo Android para dispositivos móviles.

Android destaca por ser un sistema multiplataforma, es decir, que permite instalarse en dispositivos totalmente diferentes manteniendo todas las compatibilidades. Esto permitió un gran crecimiento de este sistema operativo que más adelante sería el sistema más utilizado tanto por los usuarios como por las empresas.

Hay otros sistemas para dispositivos móviles como Symbian que actualmente está en decadencia y terminará por extinguirse, también tenemos Windows Phone un sistema desarrollado por Microsoft y es uno de los grandes competidores por el mercado de los dispositivos móviles junto con Android e iOS de Apple.



Figura 2.2: Conjunto de iconos de los grandes sistemas móviles.

Para poder seleccionar un sistema operativo u otro es necesario realizar una serie de comparaciones que nos permitan tener claro que sistema va a ser el más idóneo para nuestro proyecto.

A continuación se muestra una tabla comparativa entre los sistemas operativos móviles actualmente más grandes del mercado.

	iOS 	Android 	Windows  Phone
Fabricante	Apple	Google	Microsoft
Kernel	OS X	Linux con Máquina Virtual Dalvik	Windows CE
Licencia	Código cerrado	Apache y GPL	Cerrada
Lenguaje de desarrollo	Objetive C	Java	Visual C++ en .NET
Herramienta de desarrollo	Xcode	Android SDK	Visual studio, Windows Phone Dev Tools
Executable	.app	.apk	.xap
Tienda de aplicaciones	App Store	google play	windows phone market

Figura 2.3: Tabla de características de los sistemas operativos móviles.

Kernel

Es el núcleo del sistema operativo, es decir, el software encargado de gestionar los recursos del sistema.

La diferencia entre un kernel de libre distribución a un kernel propietario, radica en el hecho de que los primeros cuentan con gran cantidad de foros y comunidades de desarrollo que proporcionan un soporte que permite solucionar problemas de compatibilidad, seguridad, realización de mejoras...

En el caso de los kernel propietario son los propios desarrolladores del sistema los que se encargan de solucionar estos errores y son bastante más costosos. En este aspecto Android destaca respecto a iOS y windows phone ya que hay mucha más información para desarrollar las aplicaciones permitiendo muchas facilidades como por el ejemplo la cantidad de librerías desarrolladas por estas comunidades de desarrollo orientadas a nuevas tecnologías o funcionalidades.

Otro aspecto es la gran adaptabilidad que tiene Android para adaptarlo a diferentes dispositivos independiente de sus características y de ahí el gran auge que tiene en comparación con iOS que solo se puede adaptar a los dispositivos fabricados por apple o windows phone que solo se adapta con aquellos dispositivos con los que tengan cierto convenio como puede ser Nokia.

Licencia

Android es un sistema totalmente abierto, esto permite que todas las versiones mejoradas seguirán siendo totalmente libres y así se evitaría competir con alguna versión mejorada privativa.

Lenguaje de desarrollo

A la hora de desarrollar una aplicación móvil es fundamental conocer el lenguaje de programación para ese sistema. En el caso de las empresas Apple y Microsoft han desarrollado su lenguaje propio que dependiendo su dificultad de aprendizaje puede complicarse más el desarrollo o no.

En el caso de Android utiliza java, un lenguaje usado a lo largo de la carrera y que puede utilizarse para numerosos ámbitos de la programación. Hay que destacar que este lenguaje de programación tienes una gran cantidad de librerías que facilitan el desarrollo de un proyecto y de ahí que programar para Android sea mucho más sencillo y simple.

Interfaz de usuario

Hoy en día es el aspecto más importante para los usuarios porque si una aplicación no tiene un buen aspecto los usuarios no la usan porque es su contacto con la aplicación en el día a día.

En este aspecto iOS con el iphone van por delante en este aspecto como por ejemplo las librerías para representar gráficas esta mucho más avanzada que las librerías de Android. En cambio Android se centra más en una interfaz no tan llamativa si no una interfaz simple e intuitiva que permita a las usuarios navegar por cualquier aplicación sin tener grandes conocimientos de la dicha.

En el caso de windows phone se centra en el uso del puntero y es una interfaz totalmente diferente que los sistemas explicados anteriormente.

Estadísticas de uso

Actualmente el sistema móvil más utilizado es Android principalmente por ser multiplataforma que permite a los usuarios poder comprar móvil de características hardware totalmente diferentes, permitiendo comprar un móvil de bajo costo y o un móvil de alto precios pero que pueden tener el mismo software garantizando la compatibilidad entre un dispositivo y otro.

Otra característica es su facilidad de uso en comparación con otros dispositivos y este uso tan diferenciado entre los diferentes sistemas se puede comprobar en la siguiente figura.

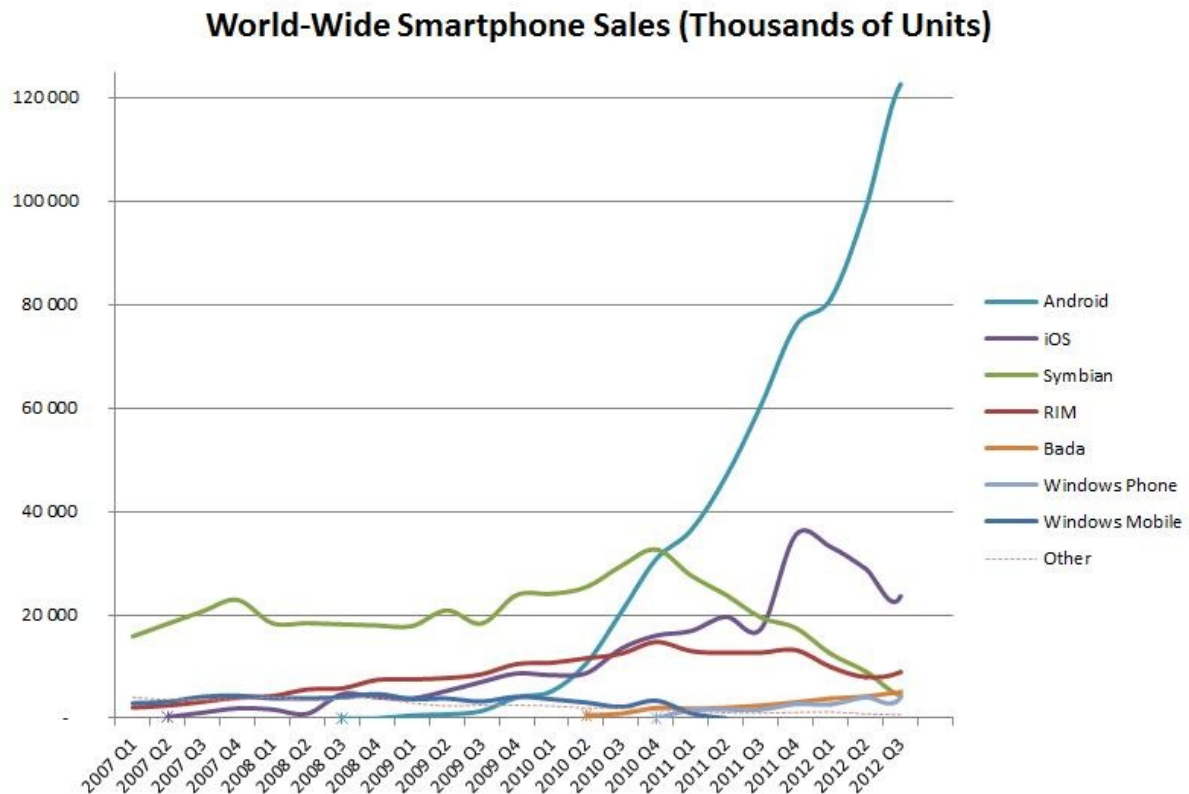


Figura 2.4: Gráfica de uso de los principales sistemas operativos móviles.

Android, la mejor solución

Después de esta comparación entre los diferentes dispositivos móviles podemos comprobar que Android se adapta más a nuestros objetivos con el deseo de que la aplicación llegue al mayor número de usuarios.

Otra cuestión fundamental para esta selección es la facilidad que proporciona este sistema para desarrollar aplicaciones y el coste necesario para llevarlo a cabo.

En conclusión he seleccionado Android como sistema para la aplicación **JET!**.

A continuación se mostrará una serie de motivos por los que se ha elegido este sistema:

- La licencia es abierta, por lo que el código está accesible.
- Bajo coste para desarrollar la aplicación.
- La experiencia a lo largo de la carrera con el lenguaje de programación Java.
- La gran abundancia de documentación en internet.
- La interfaz sencilla e intuitiva.



Figura 2.5: Imagen ilustrativa del sistema operativo móvil seleccionado.

2.3.2 Plataforma de servicios web

Los dispositivos móviles se conectan con nuestra plataforma consumiendo servicios web que son unos “programas” que nos permiten el intercambio de información entre el servidor y los dispositivos móviles.

Podemos diferenciar distintas formas de ofrecer estos servicios y distintos tipos de protocolos de comunicaciones, los más destacados son SOAP y REST, y distintos lenguajes de implementación como pueden ser Java, C++...

Tipos de protocolos de comunicación

En primer lugar vamos a diferenciar los distintos tipos de protocolos de comunicación para poder dejar claro porque hemos seleccionado uno y no otro.

En el caso de SOAP es un protocolo estándar que permite comunicar dos objetos a través del intercambio de datos XML. Básicamente es un paradigma de mensajería de una dirección sin estado, que suele utilizarse para servicios web más complejos.

El formato del intercambio es en XML que es un lenguaje basado en etiquetas que tiene un formato muy estructurado y que permite la facilidad de extensión con la adición de nuevas etiquetas según las necesidades y basta con un único analizador para poder procesar la información de dicho formato.

Un posible ejemplo de los barajados para utilizar en el proyecto se puede observar en la siguiente figura como quedaría de manera gráfica la estructura del XML de un mensaje.

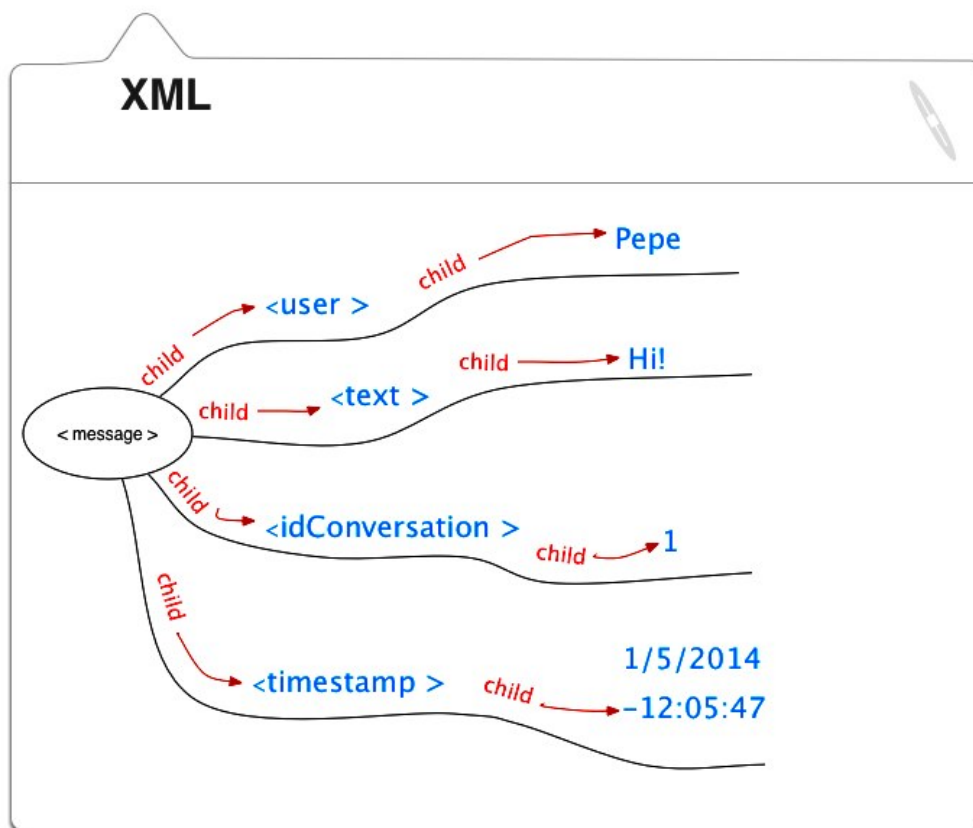


Figura 2.6: Representación del posible XML de la comunicación en nuestro proyecto.

Otra opción es REST, un estilo de arquitectura de software para sistemas distribuidos tales como la web que se centra en el uso de estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional. Las operaciones (o funciones) se solicitan

mediante GET, POST, PUT y DELETE por lo que no requiere implementaciones especiales para consumir estos servicios.

Un sustituto de XML como contenedor de información existe JSON que nos permite mejorar el rendimiento, que es aconsejable cuando disponemos de escasos recursos como en nuestro caso los dispositivos móviles, y desde el punto de vista del desarrollador es un formato mucho más sencillo que el típico XML.

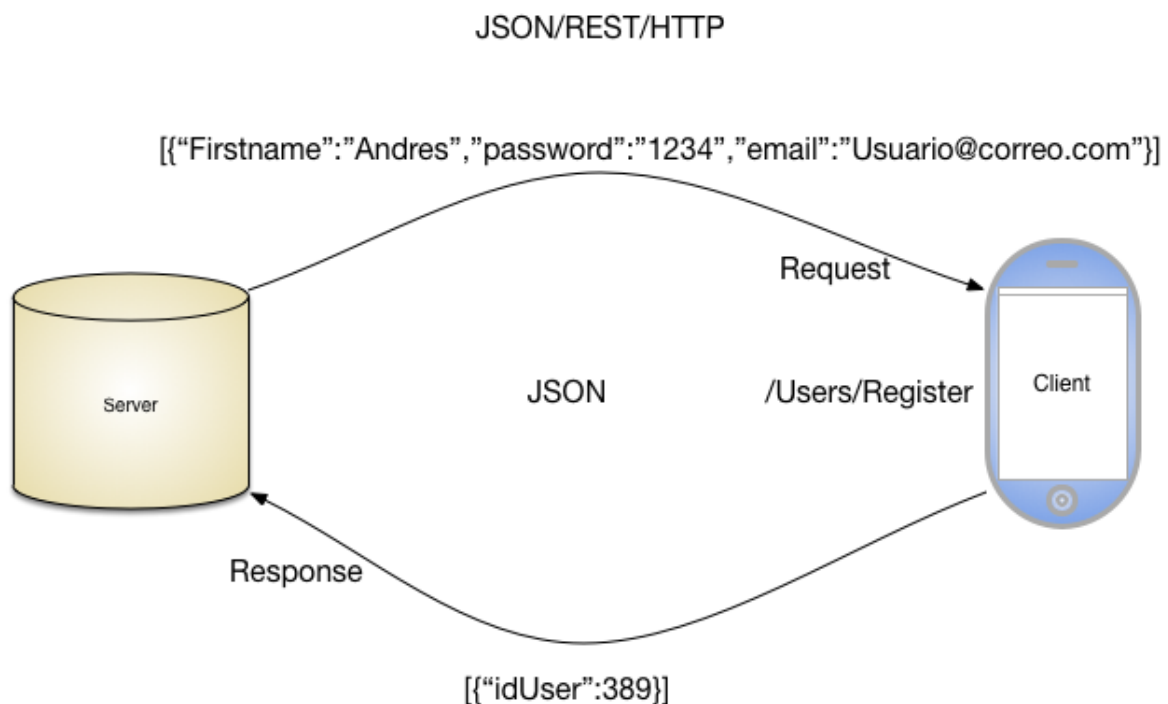


Figura 2.7: Representación del funcionamiento de Json en nuestra aplicación.

JSON/REST/HTTP, la mejor solución

Como se puede comprobar después de la comparación anterior la mejor opción es utilizar JSON y REST y comunicar los datos a través de http.

La principal razón es que REST con JSON es mucho más eficiente que SOAP con XML y para un dispositivo móvil, que tiene poco recursos, la eficiencia es algo esencial en las comunicaciones entre el dispositivo móvil y el servidor.

Otra razón importante es que para el desarrollador es más fácil de entender REST al ser un patrón de servicios web muy simple a comparación con SOAP que opta por un patrón jerárquico basado en etiquetas que es más complicado definir un analizador para XML que para JSON.

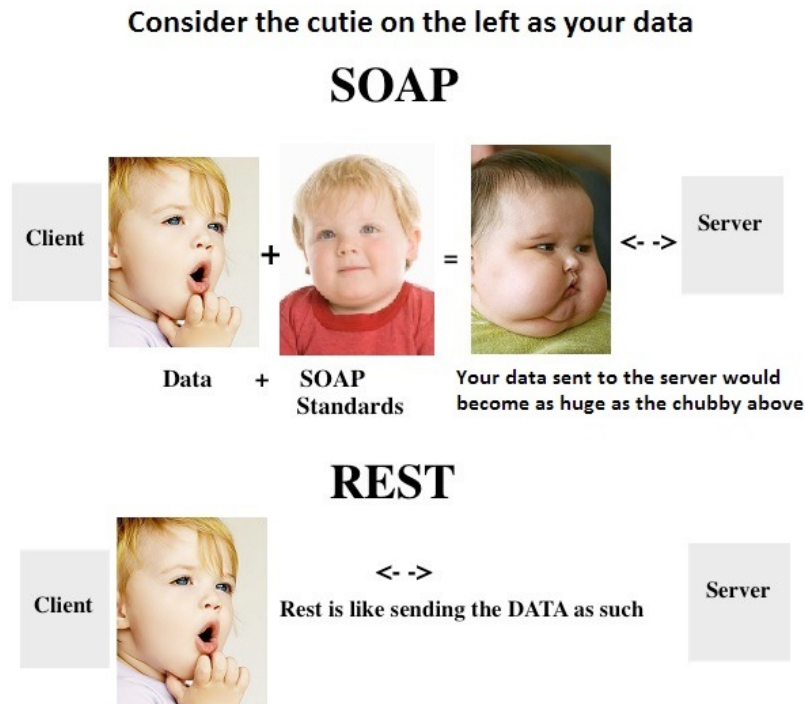


Figura 2.8: Comparación entre Soap y Rest.

2.3.3 Plataforma Web

Para la parte administradora se va a desarrollar una página web que permita realizar operaciones sobre la base de datos de manera fácil y sencilla sin tener que acceder de manera directa.

Para el desarrollo de paginas web hay numerosas formas y herramientas pero en este caso voy a comparar Spring con webratio herramientas dadas a lo largo de la carrera que ofrecen un servicio totalmente diferente pero que tienen el mismo objetivo, desarrollar una pagina web.

Tipos de herramientas

Spring es una herramienta de desarrollo de cualquier aplicación desarrollada en java y existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. Este framework no ofrece un modelo de programación concreto pero actualmente es uno de los más populares.

Esta herramienta es recomendable cuando la lógica de negocio es bastante compleja y se necesite de un lenguaje, como java, que permita realizar procesos con un nivel de complejidad alto. Hay que destacar que esta herramienta es gratuita y no tiene ningún coste para desarrollar en ella.



Figura 2.9: Icono representativo de Spring

Webratio es una herramienta que permite desarrollar paginas web a través de modelos sin tener que definir el código directamente. Esta herramienta se automáticamente de monitorear las aplicaciones web.

Hay que destacar que esta herramienta es de pago pero existe una versión de prueba para estudiantes.



Figura 2.10: Icono representativo de WebRatio.

Webratio, la mejor solución

Finalmente la herramienta que más se identifica con nuestras necesidades es webratio que nos facilita el desarrollo de una aplicación web de manera sencilla utilizando modelos.

La única pega de esta herramienta es el precio para poderla usar y para este proyecto se va a utilizar una versión de estudiante que caduca en cierto tiempo y al ser una herramienta poco conocida la documentación es escasa.

3-Especificación

La especificación es un documento que describen las necesidades y funciones específicas de un sistema determinado.

En primer apartado voy a definir los casos de uso, técnica que permite definir como interactúa el cliente con el servidor y viceversa y que surgen de la captura de requisitos previamente definidos.

Primeramente mostraré los casos de uso de manera gráfica para su mejor entendimiento y seguido una descripción detallada de cada uno.

En el segundo apartado describo los modelos conceptuales o diagrama de clases que describen la estructura del sistema, mostrando clases, atributos y relaciones entre ellos.

Finalmente se describirá el modelo de comportamiento del sistema, formado por los diagramas de secuencia del sistema.

3.1 Casos de uso

En esta sección se mostrará los actores (Es toda entidad externa al sistema que tiene cierta relación con este y utiliza parte de los servicios que este proporciona) y diagramas que intervienen en nuestro sistema. Hay que destacar que el proyecto global esta formado por dos sistemas totalmente definidos que sería la parte móvil y la parte servidora.

Esta definición se centra en la parte servidora.

3.1.1 Actores

En este sistema se pueden encontrar dos actores claramente identificados, el usuario de la aplicación móvil que tendrá acceso a todas las funcionalidades de la mensajería y el usuario administrador que será el encargado de controlar la frecuencia de las actualizaciones y del mantenimiento de la información léxica y sintáctica para una mejor categorización de la información.

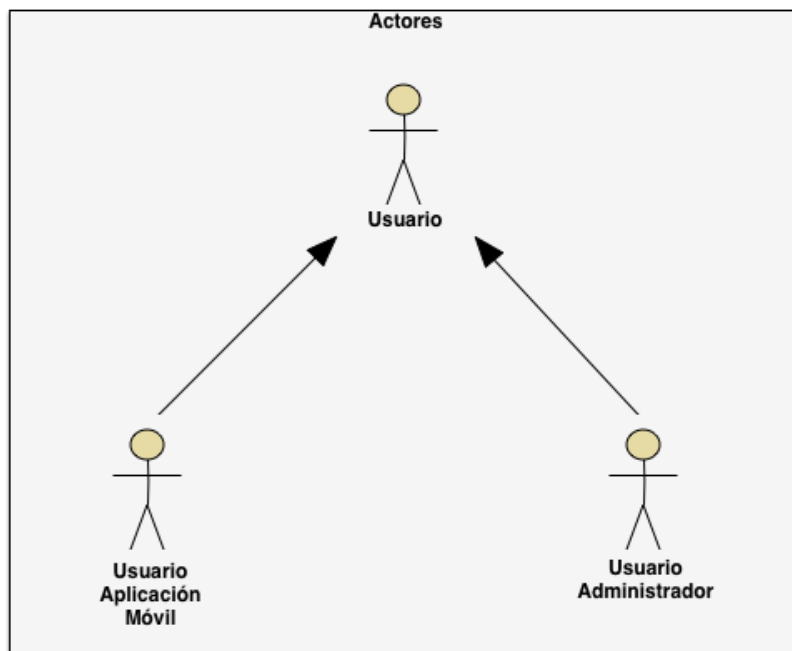


Figura 3.1: Esquema representativo de los actores de nuestra plataforma

3.1.2 Diagrama de casos de uso

Un diagrama de caso de uso es una descripción del comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio. Esta descripción se enfoca en el valor suministrado por el sistema a entidades externas como los usuarios

A continuación podemos ver en la siguiente figura los casos de uso de la plataforma.

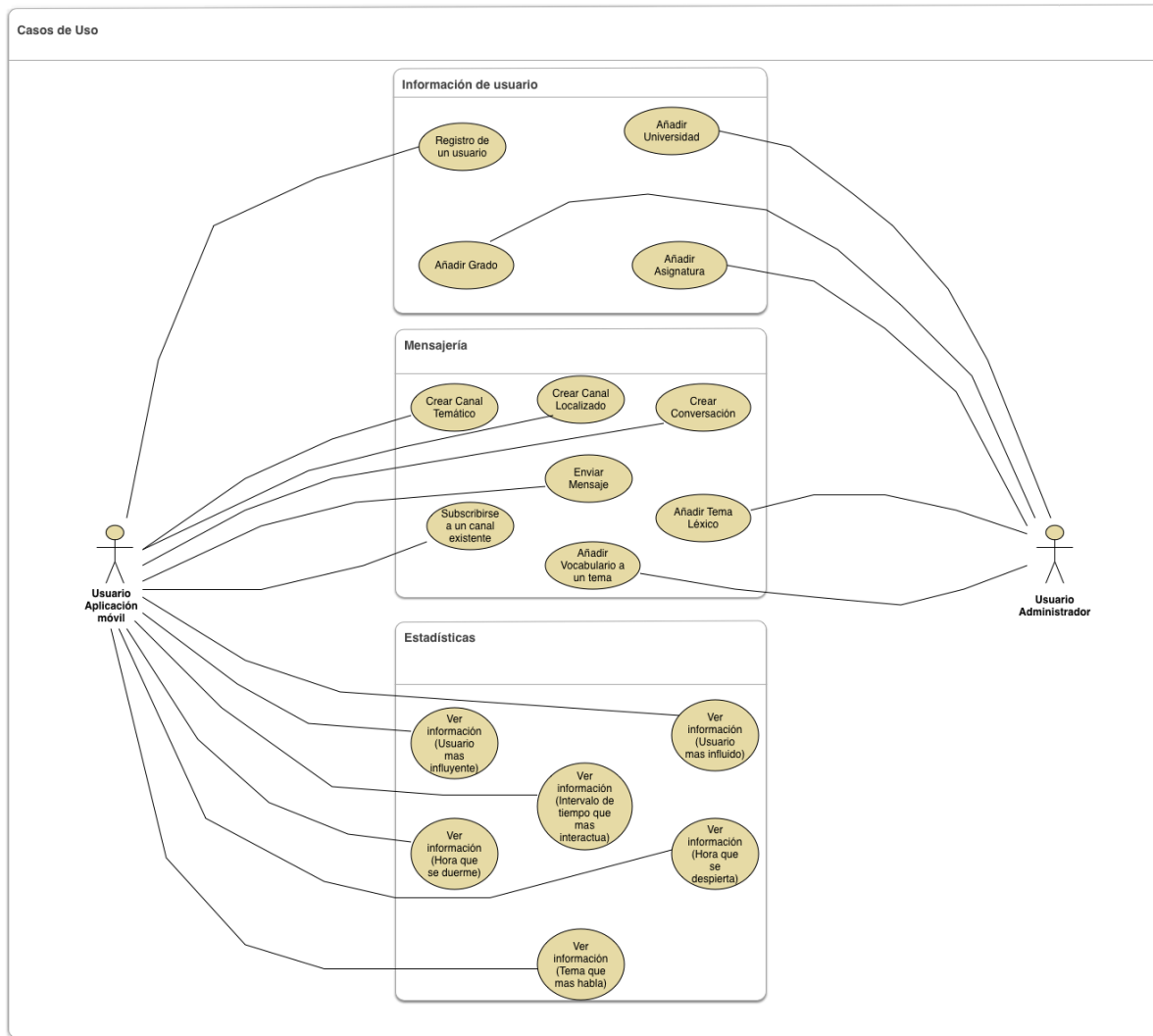


Figura 3.2: Esquema representativo de los casos de uso de nuestro proyecto.

3.1.3 Descripción de casos de uso

Seguidamente se detallan cada uno de los casos de uso de la plataforma.

Información de usuario

El registro de un usuario se encarga de registrar a un usuario en la aplicación y para ello es necesario almacenar toda la información del usuario y posteriormente registrar el dispositivo móvil en la plataforma Nimbees.

Registro de un usuario
<p>Descripción: Se desea registrarse en la aplicación Jet!</p>
<p>Autores: Usuario Aplicación Móvil</p>
<p>Precondiciones: Tener almacenado en el servidor Universidades, Grados y asignaturas</p>
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario accede a la aplicación móvil 2-El usuario selecciona el Username 3-El usuario selecciona el nombre 4-El usuario selecciona los apellidos 5-El usuario selecciona el email 6-El usuario selecciona una contraseña 7-El usuario selecciona la opción registrarse 8-El dispositivo móvil envía una petición al servidor para que le devuelva las universidades. 9-Almacena las universidades en la base de datos del dispositivo móvil. 10-El dispositivo móvil muestra todas las universidades. 11-El usuario selecciona una universidad. 12-El dispositivo móvil envía una petición al servidor para que le devuelva los grados de esa universidad. 13-Almacena los grados en la base de datos del dispositivo móvil. 14-El dispositivo móvil muestra todos los grados. 15-El usuario selecciona un grado. 16-El dispositivo móvil pide información de las asignaturas de ese grado. 17-El dispositivo móvil almacena en la base de datos las asignaturas. 18-El dispositivo móvil envía la información al servidor. 19-El servidor almacena la información y devuelve un id del usuario. 20-El dispositivo móvil almacena el id del usuario. 21-El dispositivo móvil se registra en la plataforma NimBees. 22-Finaliza el caso de uso <p>Flujo Alternativo:</p> <p>{7-Error de los datos}</p> <ol style="list-style-type: none"> 1-El móvil comprueba que hay campos vacíos o incorrectos 2-El caso de uso vuelve al paso 2 <p>{19-Error al almacenar los datos}</p> <ol style="list-style-type: none"> 1-El servidor comprueba que ya existe ese Username 2-El caso de uso vuelve al paso 2

Figura 3.3:Tabla descriptiva del caso de uso Registro de un usuario.

Un usuario que este registrado como administrador podrá añadir diversas universidades a través de la pagina web.

Añadir Universidad
Descripción: Se desea añadir una nueva universidad a la base de datos
Autores: Usuario Administrador
Precondiciones: Creada cuenta administrador
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario se logea en la pagina web. 2-El servidor devuelve todas las universidades existentes. 3-El usuario introduce una nueva universidad. 4-El servidor almacena la nueva universidad. 5- Finaliza el caso de uso. <p>Flujo alternativo:</p> <p>{1-El usuario no existe}</p> <ol style="list-style-type: none"> 1-Vuelve al paso 1. <p>{4-La universidad ya existe}</p> <ol style="list-style-type: none"> 1-Vuelve al paso 3.

Figura 3.4:Tabla descriptiva del caso de uso Añadir Universidad.

Un usuario que este registrado como administrador podrá añadir diversos grados a través de la pagina web.

Añadir Grado
Descripción: Se desea añadir un nuevo grado a la base de datos
Autores: Usuario Administrador
Precondiciones: Creada cuenta administrador
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario se logea en la pagina web. 2-El servidor devuelve todas las universidades existentes. 3-El usuario selecciona una universidad. 4-El servidor devuelve todos los grados de esa universidad 5-El usuario introduce un nuevo grado. 6-El servidor almacena el nuevo grado. 7- Finaliza el caso de uso. <p>Flujo alternativo:</p> <p>{1-El usuario no existe}</p> <ol style="list-style-type: none"> 1-Vuelve al paso 1. <p>{5-El grado ya existe}</p> <ol style="list-style-type: none"> 1-Vuelve al paso 5.

Figura 3.5:Tabla descriptiva del caso de uso Añadir Grado.

Un usuario que este registrado como administrador podrá añadir diversas asignaturas a través de la pagina web.

Añadir Asignatura
Descripción: Se desea añadir una nueva asignatura a la base de datos
Autores: Usuario Administrador
Precondiciones: Creada cuenta administrador
Flujo principal: 1-El usuario se logea en la pagina web. 2-El servidor devuelve todas las universidades existentes. 3-El usuario selecciona una universidad. 4-El servidor devuelve todos los grados de esa universidad 4-El usuario selecciona un grado. 5-El servidor devuelve todas las asignaturas de ese grado 6-El usuario introduce una nueva asignatura. 7-El servidor almacena la asignatura. 8- Finaliza el caso de uso.
Flujo alternativo: {1-El usuario no existe} 1-Vuelve al paso 1. {5-El grado ya existe} 1-Vuelve al paso 5.

Figura 3.6:Tabla descriptiva del caso de uso Añadir Asignatura.

Mensajería

Un usuario desde el dispositivo móvil puede crear una canal según la localización.

Crear Canal Localizado
Descripción: Se desea crear una canal según la localización
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado

Flujo principal:

- 1-El usuario selecciona de que tipo quiere el canal (Cercanía a su hogar,trabajo...)
- 2-El dispositivo móvil envía el tipo de canal al servidor.
- 3-El servidor devuelve los usuarios que cumplen esa característica.
- 4-El dispositivo crea un canal con esos usuarios.
- 5-El servidor almacena ese canal.
- 6-El servidor devuelve el id del canal.
- 7-El dispositivo almacena el id devuelto.
- 8-Finaliza el caso de uso.

Figura 3.7:Tabla descriptiva del caso de uso Crear Canal Localizado.

El usuario desde el dispositivo móvil puede crear una canal totalmente anónimo con aquellos usuarios que hablen más de un tema concreto.

Crear Canal Temático
Descripción: Se desea crear una canal según un tema específico
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario selecciona de que tipo quiere el canal y el tema 2-El dispositivo móvil envía el tipo de canal al servidor. 3-El servidor comprueba que no exista ese canal. 4-El servidor calcula que usuarios cumplen esa característica. 5-El servidor crea y almacena ese canal. 6-El servidor devuelve un id del canal. 7-El dispositivo almacena el id devuelto. 8-Finaliza el caso de uso. <p>Flujo alternativo:</p> <p>{4-No existe usuarios que cumplan esa característica}</p> <ol style="list-style-type: none"> 1-Vuelve al paso 1 <p>{3-El canal ya existe}</p> <ol style="list-style-type: none"> 1-Va directamente al paso 6

Figura 3.8:Tabla descriptiva del caso de uso Crear Canal Temático.

El usuario puede crear una conversación desde el dispositivo móvil seleccionando un canal. Estas conversaciones son las que los usuarios utilizan para transmitirse mensajes y pueden crearse numerosas conversaciones dentro de un canal.

Crear conversación
Descripción: Se desea crear una conversación dentro de un canal.
Autores: Usuario Aplicación Móvil

<p>Precondiciones: Usuario registrado y suscrito a un canal</p>
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario selecciona un canal. 2-El usuario crea una nueva conversación. 3-El dispositivo móvil envía la información de la conversación y del canal. 4-El servidor almacena la conversación. 5-El servidor devuelve el id de la conversación. 6-El dispositivo almacena el id de la conversación. 7-Finaliza el caso de uso.

Figura 3.8: Tabla descriptiva del caso de uso Crear Conversación.

El usuario puede enviar un mensaje dentro de una conversación específica para que este llegue a los diversos usuarios que se encuentran en dicha conversación.

Enviar Mensaje
<p>Descripción: Se desea crear un mensaje dentro de una conversación.</p>
<p>Autores: Usuario Aplicación Móvil</p>
<p>Precondiciones: Usuario registrado y suscrito a un canal que contenga alguna conversación creada.</p>
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario selecciona una conversación. 2-El dispositivo muestra todos los mensajes de esa conversación. 3-El usuario crea un nuevo mensaje. 4-El dispositivo móvil envía la información del mensaje al servidor. 5-El servidor almacena el mensaje. 6-El servidor envía el mensaje a la plataforma Nimbees. 7-La plataforma Nimbees envía el mensaje a todos los dispositivos registrados. 8-Cada dispositivo móvil comprueba si le pertenece el mensaje. 9-El servidor procesa el mensaje. 10-Finaliza el caso de uso.

Figura 3.9: Tabla descriptiva del caso de uso Enviar Mensaje

Un usuario se puede suscribir en aquellos canales que se refieren a una universidad, un grado o una asignatura debido a que se definen cuando se crea una nueva asignatura, universidad o grado por lo que un usuario solamente necesita suscribirse a ese canal. Hay que destacar que muchos canales se encuentran en una jerarquía y si un usuario se suscribe a un canal automáticamente se suscribe a los canales padre. Una jerarquía por ejemplo podría ser la siguiente.

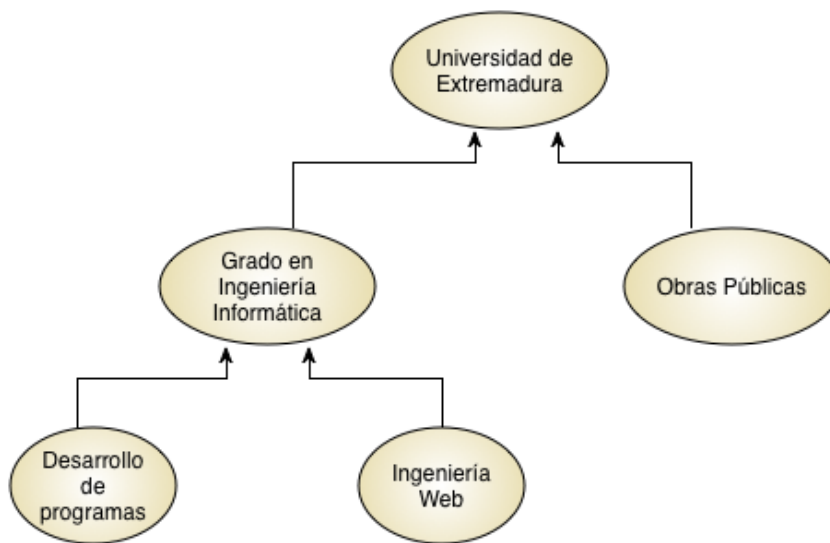


Figura 3.10: Representación de la jerarquía académica.

En este caso si un usuario se suscribe a la asignatura de Desarrollo de programas automáticamente se suscribe al canal de Grado en Ingeniería Informática y al canal de Universidad de Extremadura.

Subscribirse a un canal existente
Descripción: Se desea suscribirse a un canal ya existente.
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1-El usuario selecciona un canal que quiera suscribirse. 2-El dispositivo móvil envía la información de la suscripción al servidor. 3-El servidor almacena la suscripción de ese canal y devuelve el id del padre. 4-El dispositivo móvil crea una nueva suscripción con el id del padre 5-El servidor almacena la suscripción de ese canal. 6-Finaliza el caso de uso. <p>Flujo alternativo:</p> <p>{3- No devuelve un id padre válido}</p> <ol style="list-style-type: none"> 1- Finaliza en el paso 6 <p>{5- Devuelve un id padre válido}</p> <ol style="list-style-type: none"> 1-El servidor devuelve el id padre. 2-Vuelve al paso 4

Figura 3.11: Tabla descriptiva del caso de uso Subscribirse a un canal existente

El administrador puede crear nuevos temas que permitan mejorar la búsqueda de aquellas personas tengan cierto conocimientos de un tema. Cuanto más detallado sea el vocabulario que tengamos almacenado, mejor será la búsqueda de los usuarios.

Añadir Tema Léxico
Descripción: Se desea añadir un nuevo tema.
Autores: Usuario Administrador
Precondiciones: Registrado administrador
Flujo principal: 1-El usuario se loguea en la página. 2-El sistema le devuelve todos aquellos temas ya definidos. 3-El usuario inserta un nuevo tema. 4-El sistema almacena el nuevo tema en la base de datos. 5-Finaliza el caso de uso.
Flujo alternativo: {1-El usuario no esta registrado} 1-Vuelve al paso 1 {4-El tema ya esta definido} 1-Vuelve al paso 3

Figura 3.12: Tabla descriptiva del caso de uso Añadir Tema Léxico.

El administrador puede añadir nuevo vocabulario a los temas anteriormente definidos para saber el contexto de un tema concreto.

Añadir vocabulario un tema concreto
Descripción: Se desea añadir una palabra a un tema concreto.
Autores: Usuario Administrador
Precondiciones: Registrado administrador

Flujo principal:

- 1-El usuario se loguea en la página.
- 2-El sistema le devuelve todos aquellos temas ya definidos.
- 3-El usuario selecciona un tema concreto.
- 4-El sistema le devuelve todas las palabras de ese tema.
- 5-El usuario inserta la nueva palabra.
- 6-El sistema almacena la nueva palabra en la base de datos.
- 7-Finaliza el caso de uso.

Flujo alternativo:

- {1-El usuario no esta registrado}
- 1-Vuelve al paso 1
- {6-La palabra ya esta definida}
- 1-Vuelve al paso 5

Figura 3.13: Tabla descriptiva del caso de uso Añadir Vocabulario a un tema concreto.

Estadísticas

Los usuarios pueden observar quien es el usuario que más mensajes escribe en la aplicación permitiendo una pequeña competitividad entre los usuarios.

Ver Información (Usuario más influyente)
Descripción: Se desea ver quien es el usuario que más mensajes envía en la aplicación
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
Flujo principal:
1-El servidor envía la información a los dispositivos.
2-El dispositivo actualiza la base de datos.
3-El dispositivo muestra la información al usuario.
4-Finaliza el caso de uso.

Figura 3.14: Tabla descriptiva del caso de uso Ver Información(Usuario más influyente)

Los usuarios pueden observar quien es el usuario que más mensajes recibe en la aplicación permitiendo una pequeña competitividad entre los usuarios.

Ver Información (Usuario más influido)
--

Descripción: Se desea ver quien es el usuario que más mensajes recibe en la aplicación
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
Flujo principal: 1-El servidor envía la información a los dispositivos. 2-El dispositivo actualiza la base de datos. 3-El dispositivo muestra la información al usuario. 4-Finaliza el caso de uso.

Figura 3.15: Tabla descriptiva del caso de uso Ver información (Usuario más influido)

Los usuarios pueden observar en que intervalos de tiempo utilizan más la aplicación.

Estos datos son de carácter informativo.

Ver Información (Intervalo de tiempo que más interactua)
Descripción: Se desea ver cual es el intervalo de tiempo que más se comunica un usuario.
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
Flujo principal: 1-El servidor envía la información a los dispositivos. 2-El dispositivo actualiza la base de datos. 3-El dispositivo muestra la información al usuario. 4-Finaliza el caso de uso.

Figura 3.16: Tabla descriptiva del caso de uso Ver información (Intervalo de tiempo que más interactua)

Los usuarios pueden observar a que hora se suelen dormir, es decir, a que hora suelen dejar de interactuar con la aplicación.

Ver Información (Hora que se duerme)
Descripción: Se desea ver cual es el intervalo de tiempo que el usuario se suele dormir
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado

Flujo principal:

- 1-El servidor envía la información a los dispositivos.
- 2-El dispositivo actualiza la base de datos.
- 3-El dispositivo muestra la información al usuario.
- 4-Finaliza el caso de uso.

Figura 3.17: Tabla descriptiva del caso de uso Ver información (Hora de dormir)

Los usuarios pueden observar a que hora se suelen despertar, es decir, a que hora suelen empezar a interactuar con la aplicación.

Ver Información (Hora que se levanta)
Descripción: Se desea ver cual es el intervalo de tiempo que el usuario se suele levantar.
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
Flujo principal: 1-El servidor envía la información a los dispositivos. 2-El dispositivo actualiza la base de datos. 3-El dispositivo muestra la información al usuario. 4-Finaliza el caso de uso.

Figura 3.18: Tabla descriptiva del caso de uso Ver información (Hora de despertarse)

Los usuarios pueden observar que temas son los que más hablan y así poder relacionarlos con otros usuarios.

Ver Información (Tema de mayor interés para el usuario)
Descripción: Se desea saber cual es el tema de mayor interés para el usuario
Autores: Usuario Aplicación Móvil
Precondiciones: Usuario registrado
Flujo principal: 1-El servidor envía la información a los dispositivos. 2-El dispositivo actualiza la base de datos. 3-El dispositivo muestra la información al usuario. 4-Finaliza el caso de uso.

Figura 3.19: Tabla descriptiva del caso de uso Ver información (Tema de mayor interés)

3.2 Modelo conceptual

Una vez explicado los casos de uso, se debe determinar el modelo de datos para el sistema. Este modelo de datos esta formado por clases, atributos y relaciones, que representan diferentes objetos de nuestra aplicación.

Como he explicado anteriormente la aplicación se divide en dos partes totalmente identificadas, el servidor y la aplicación móvil. En el caso de este proyecto está orientada a la parte servidora.

En la siguiente figura se puede observar un esquema completo de nuestra plataforma con cada uno de los componentes necesarios para llevar a cabo todos los objetivos marcados anteriormente.

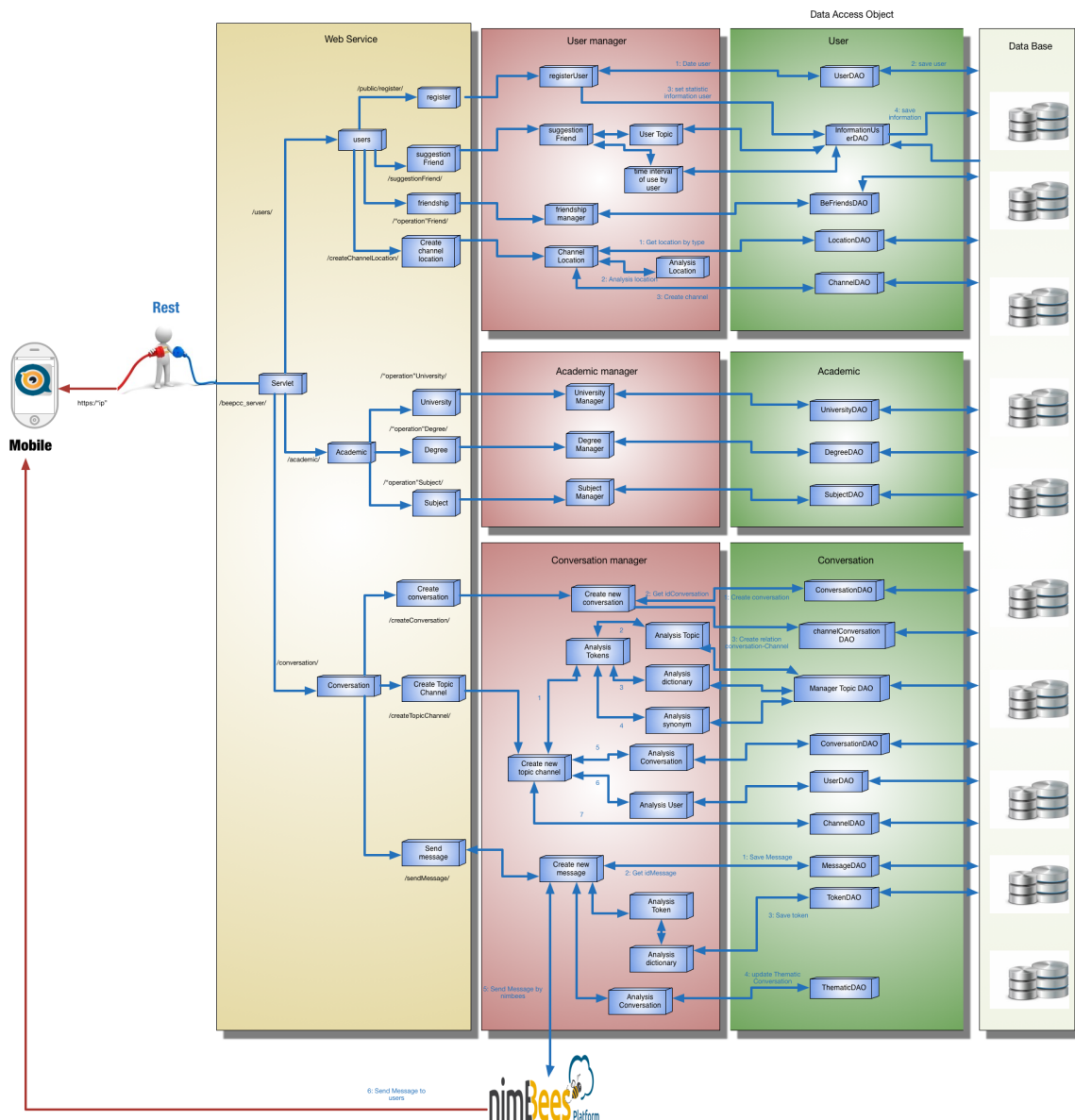


Figura 3.20: Arquitectura completa de nuestra plataforma

A continuación se muestra el modelo conceptual de la parte servidora y seguido se irá explicando parte a parte el modelo conceptual con mayor detalle.

Los nodos de este modelo hacen referencia a los objetos de nuestro proyecto como por ejemplo los usuarios, un atributo es una información que se desea almacenar de un objeto en concreto por ejemplo en el caso de un usuario almacenaremos el nombre, email... y las relaciones son las conexiones que existen entre varios objetos por ejemplo los usuarios tienen relación con las conversaciones, un usuario puede crear muchas conversaciones pero una conversación solo puede ser creada por un único usuario.

En la siguiente figura se puede observar en el modelo conceptual de la parte servidora encargada de gestionar toda la comunicación y el análisis de la información.

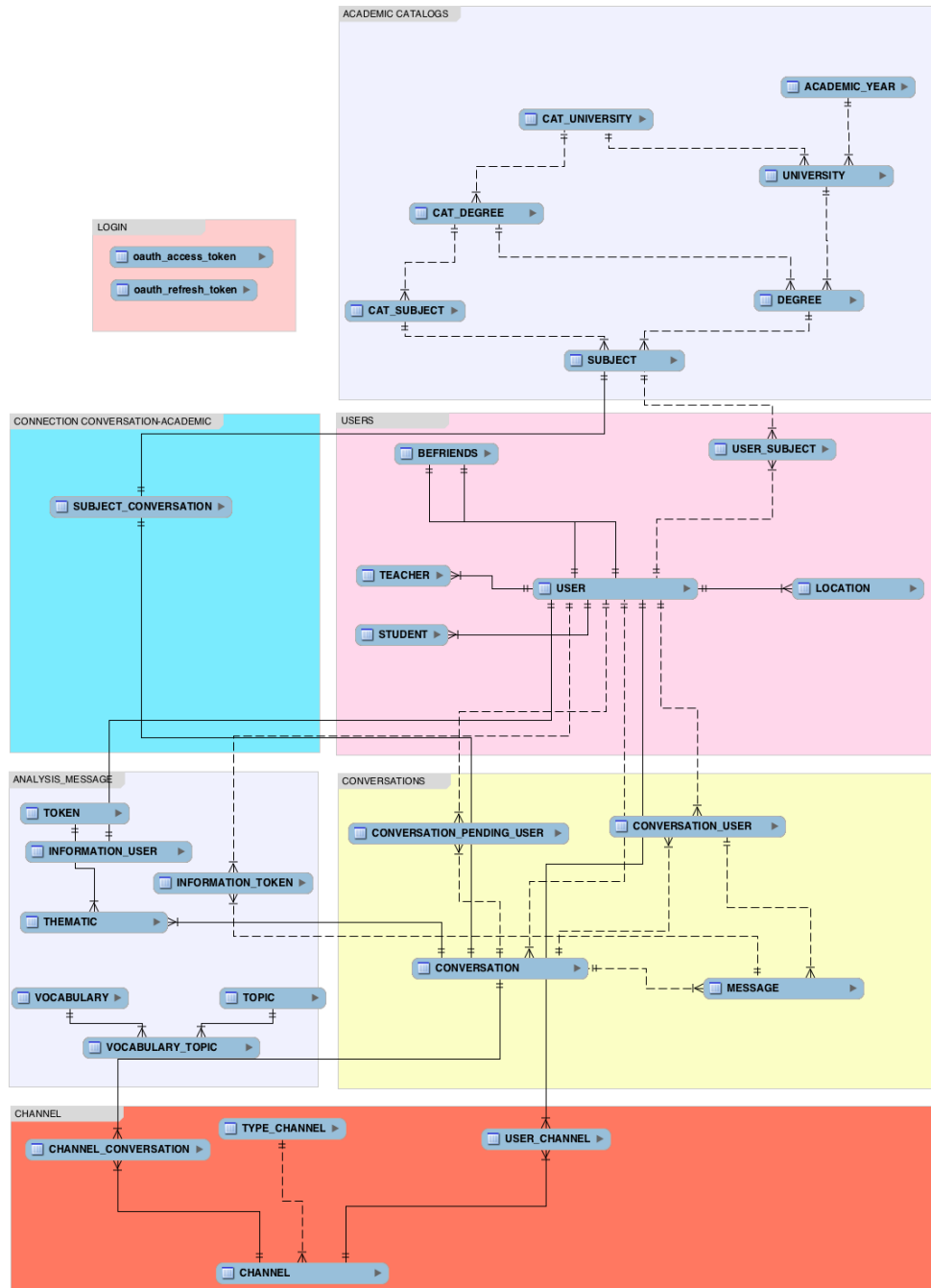


Figura 3.21: Modelo conceptual global

Para poder entender mejor este modelo conceptual lo he dividido en paquetes según su funcionalidad permitiendo abstraer los diferentes servicios que proporciona este servidor.

3.2.1 Academic Catalogs

El objetivo de este paquete es la gestión de todos los datos académicos que tendremos en nuestra aplicación para que los usuarios se puedan suscribir a dichos elementos.

Estos datos tienen que ser gestionados por un administrador y tienen que tener relación con los datos reales de las universidades.

En la siguiente figura se muestra este paquete con mayor detalle.

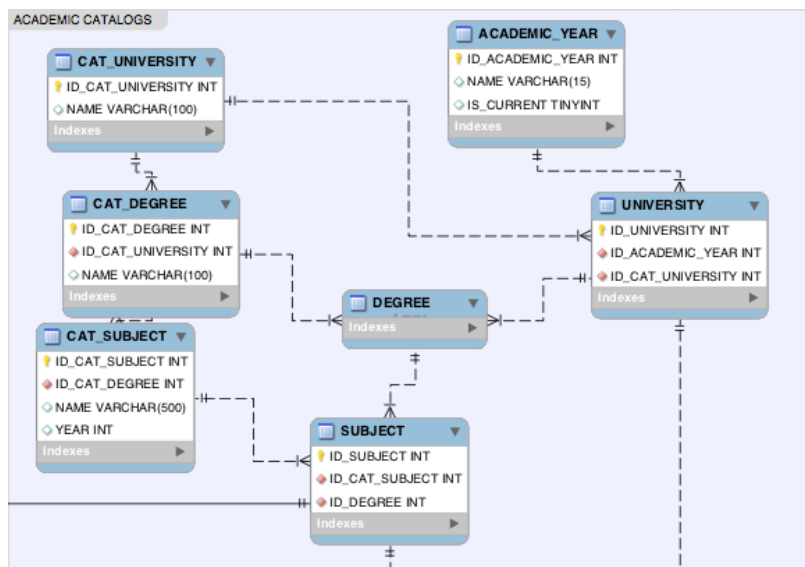


Figura 3.22: Modelo conceptual Academic Catalogs

En este paquete observamos que tenemos diferenciado las entidades que únicamente contienen los identificadores y las relaciones con otras entidades por ejemplo “UNIVERSITY” y aquellas entidades que contienen toda la información complementaria para esa entidad y que estas definidas como “CAT_’Nombre entidad”.

El objetivo de esta estructura es de poder ampliar la información correspondiente pudiendo almacenar nuevos tipos de datos de una entidad sin tener que realizar grandes modificaciones en nuestro proyecto.

3.2.2 Users

El objetivo de este paquete es tener gestionado toda la información referente a los usuarios. Tenemos una entidad principal que es “Users” que almacenará los datos principales del usuario teniendo relaciones con diversas entidades que nos permiten complementar esta información como por ejemplo tenemos la entidad “Location” que almacenará la localización de un usuario dependiendo un tipo [Casa, Trabajo...].

En este paquete también nos encontramos la relación que tiene los usuarios con las asignaturas a las que está suscrito.

Una de las ventajas que nos permite esta aplicación es la de añadir amigos para poder crear canales y a su vez conversaciones que nos permitan comunicarnos con aquellas personas que deseemos sin tener nada en común.

En el ámbito académico tenemos una jerarquía que nos permite diferenciar aquellos usuarios que sean profesores y aquellos que sean alumnos para tener en un futuro diferentes formas de administración en la aplicación.

En la siguiente figura se muestra el paquete con mayor precisión.

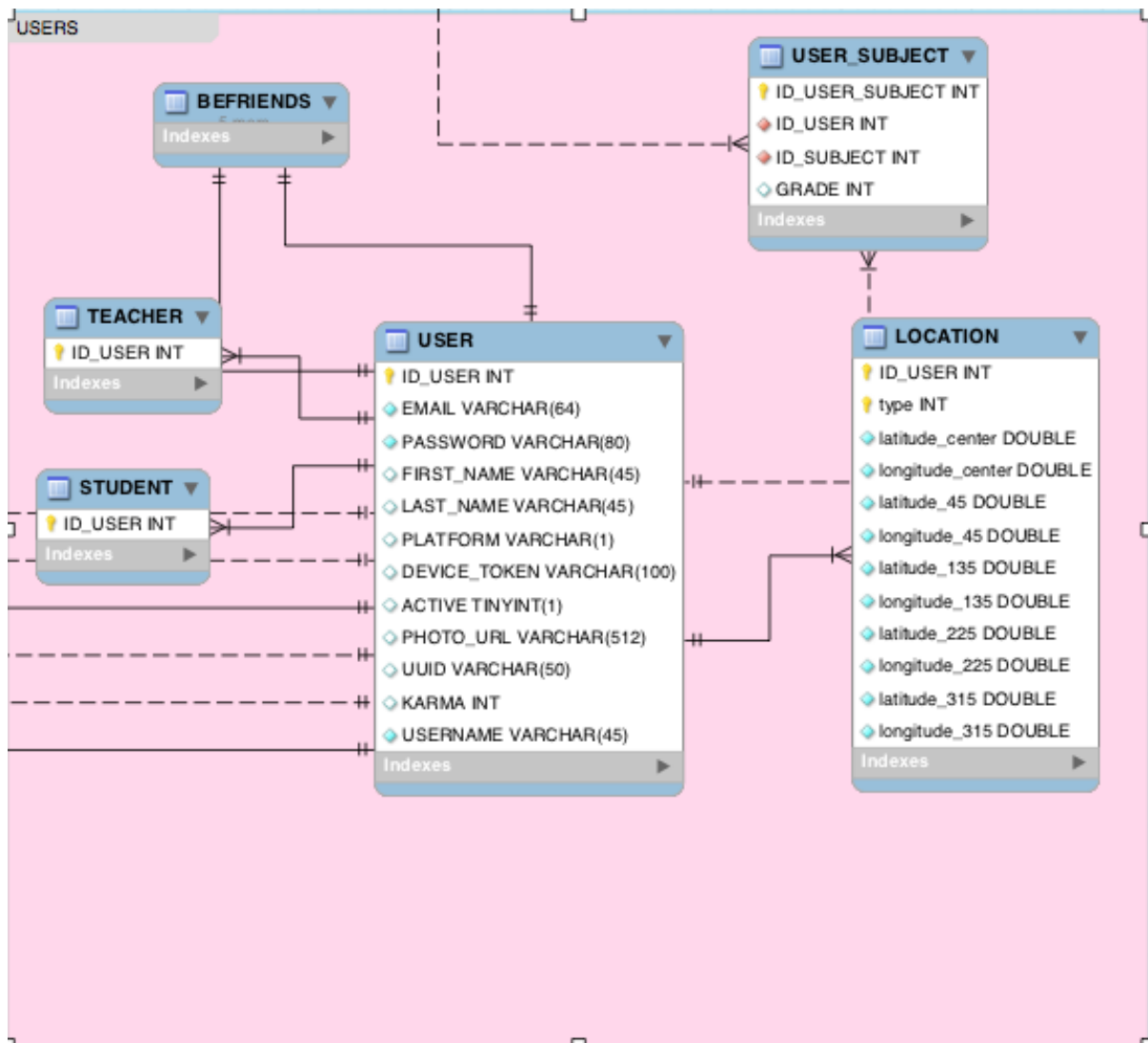


Figura 3.23: Modelo conceptual Users

3.2.3 Connection Conversation-Academic

El objetivo de este paquete es gestionar la relación que hay entre las asignaturas y las conversaciones que se crean dentro de dicha asignatura. Por ejemplo cuando un usuario quiere crear un foro dentro de la asignatura.

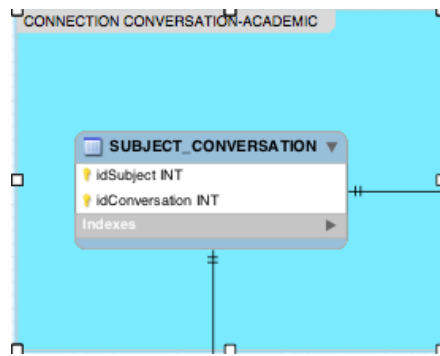


Figura 3.24: Modelo conceptual Connection Conversation-Academic

3.2.4 Login

Esta aplicación contiene un protocolo de seguridad que permite autorizar únicamente a los usuarios registrados a ciertos datos y servicios y para ellos necesitamos el paquete “login” para almacenar todos los datos necesarios para llevar a cabo correctamente la gestión de las autorizaciones.

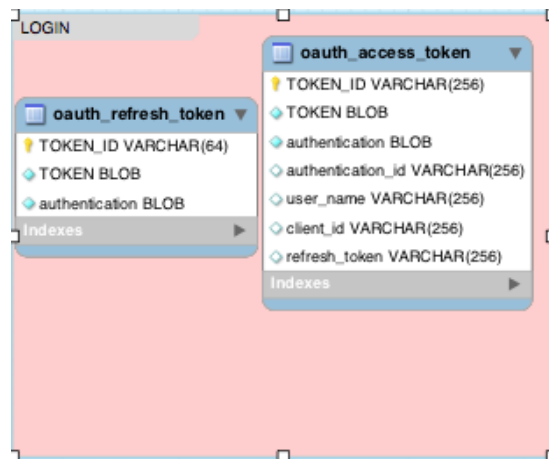


Figura 3.25: Modelo conceptual Login

3.2.5 Conversations

Este paquete se encarga de gestionar todo lo referente a las conversaciones permitiendo almacenar toda su información y la información de los mensajes que se escriben en las conversaciones.

Un usuario cuando desea enviar un mensaje es necesario que en primer lugar exista una conversación y este la seleccione y ya en su interior poder escribir el mensaje en concreto.

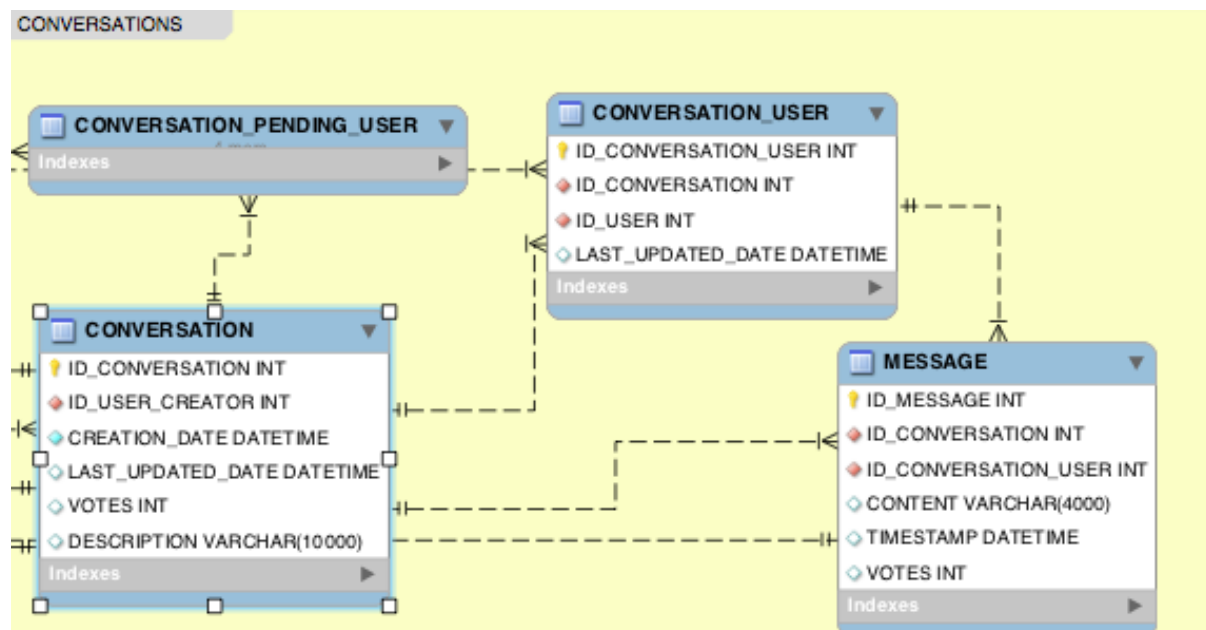


Figura 3.26: Modelo conceptual Conversations

3.2.6 Analysis Message

Este paquete se encarga de toda la gestión del análisis de los mensajes para poder proporcionar los diversos servicios, relacionados con la temática, con los usuarios.

Podemos identificar diversas entidades como “Token” que se encarga de almacenar todo aquel vocabulario que aparece por la aplicación. Hay que destacar que no todo el vocabulario es válido, es decir, tenemos un diccionario que nos indica si una palabra o token es correcto o no. Relacionado con la información que obtenemos de los mensajes tenemos una entidad que almacena quien a escrito más a un usuario, de quien a recibido más mensajes... y esta información es la que se envía a la aplicación móvil para que este actualice su base de datos y el usuario pueda observar la información de manera actualizada.

Otra entidad de gran importancia es “Thematic” que nos permite obtener la información de los temas hablado en una conversación en concreto y a su vez de todos sus integrantes.

Finalmente tenemos una parte, que es la encargada de gestionar el administrador que es la relación “Vocabulary”->”Vocabulary-Topic”->”Topic” que se encarga de almacenar todos aquellos temas y el vocabulario relacionado a dichos temas para cuando un usuario realice

una búsqueda de todos aquellos usuarios que hables de algo en concreto no se base únicamente en la palabra en concreto si no que se pueda concretar más en la búsqueda permitiendo mejores resultados.

Este paquete se puede observar con mayo detalle en la siguiente figura.

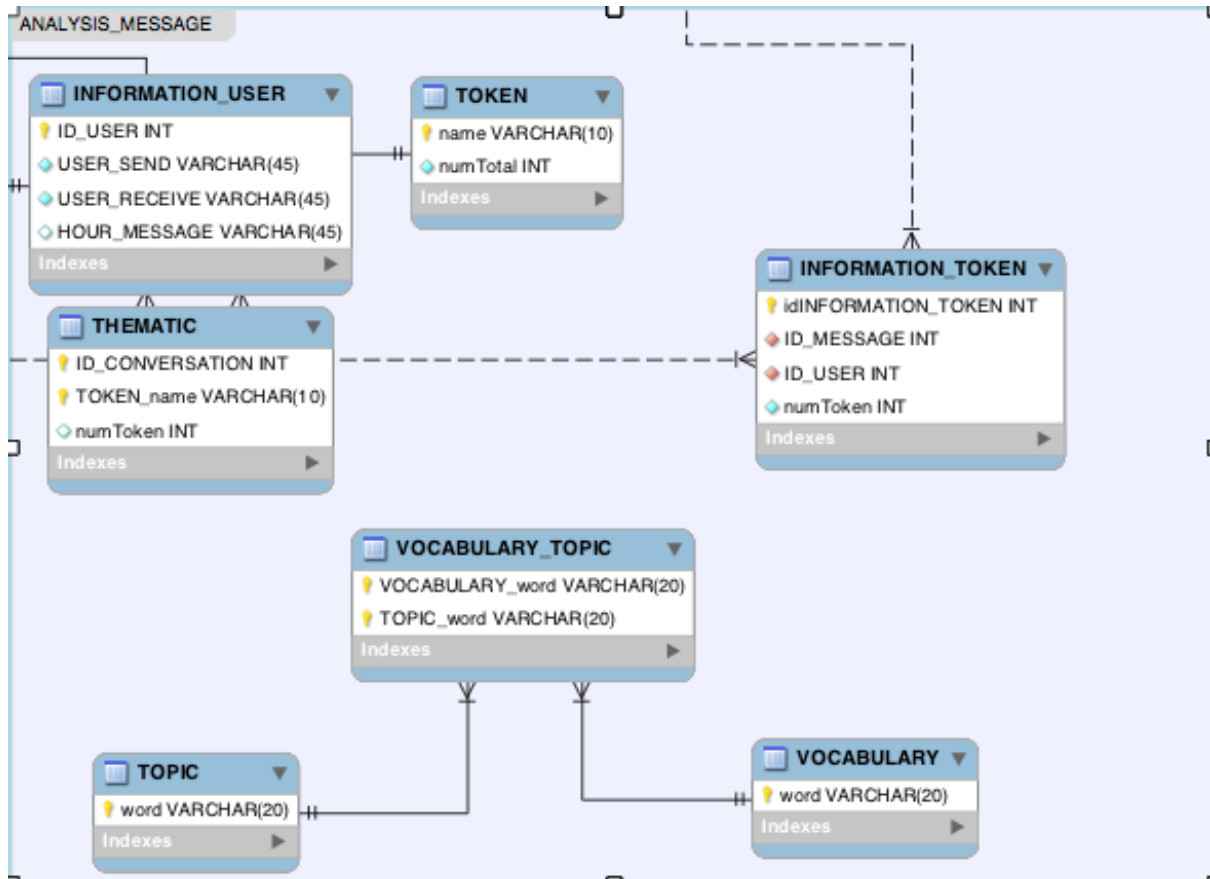


Figura 3.27: Modelo conceptual Analysis_Message

3.2.7 Channel

Este paquete se utiliza para gestionar los diversos canales creados en los dispositivos móviles permitiendo relacionarlos con los usuarios que participan en cada canal. Hay que destacar que los canales se pueden diferenciar según su tipo por ejemplo puede haber canales creados para una asignatura, para un tema concreto, para una localización...

Este paquete se puede observar detalladamente en la siguiente figura.

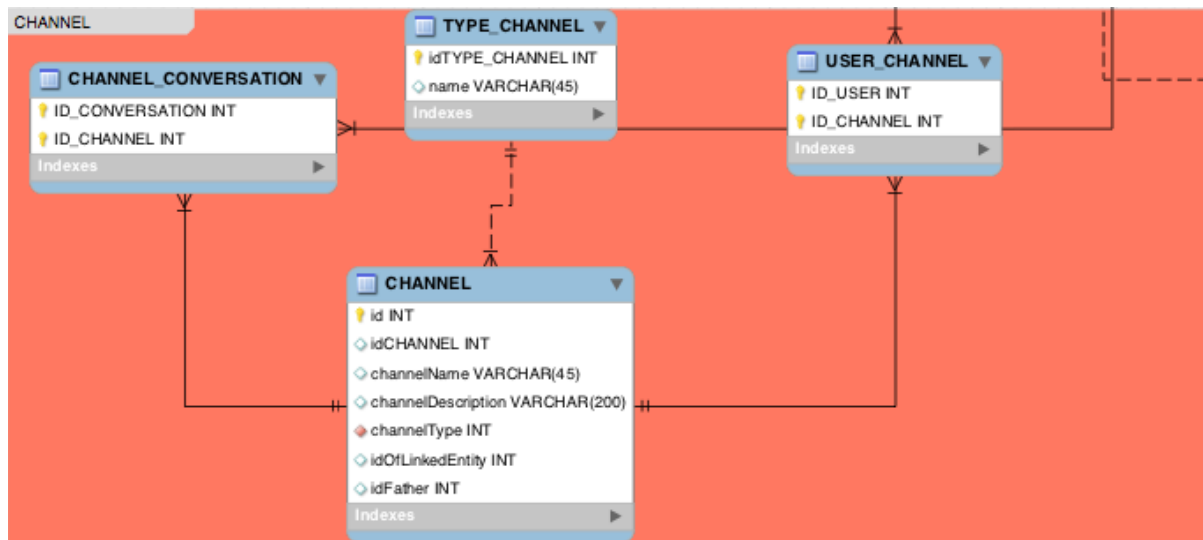


Figura 3.28: Modelo conceptual Channel

3.3 Modelo de comportamiento

Los objetos del sistema se comunican entre si utilizando operaciones de otros objetos. Con este modelo se pretende representar los diagramas de secuencia del sistema para identificar las diversas operaciones y como interactúan los actores y el sistema.

También se mostrará los contratos de las operaciones, describiendo lo efectos que puedan tener cada una, ya sean cambios en las base de información, las salidas de cada operación...

Seguidamente se detallan cada uno de los diagramas de secuencia junto con los contratos de las operaciones de los casos de uso descrito anteriormente definidos.

Hay que destacar que todas las peticiones se realizan utilizando JSON pero en los diagramas de secuencia se definirán como funciones normales y los parámetros son los datos enviados en el body de la petición.

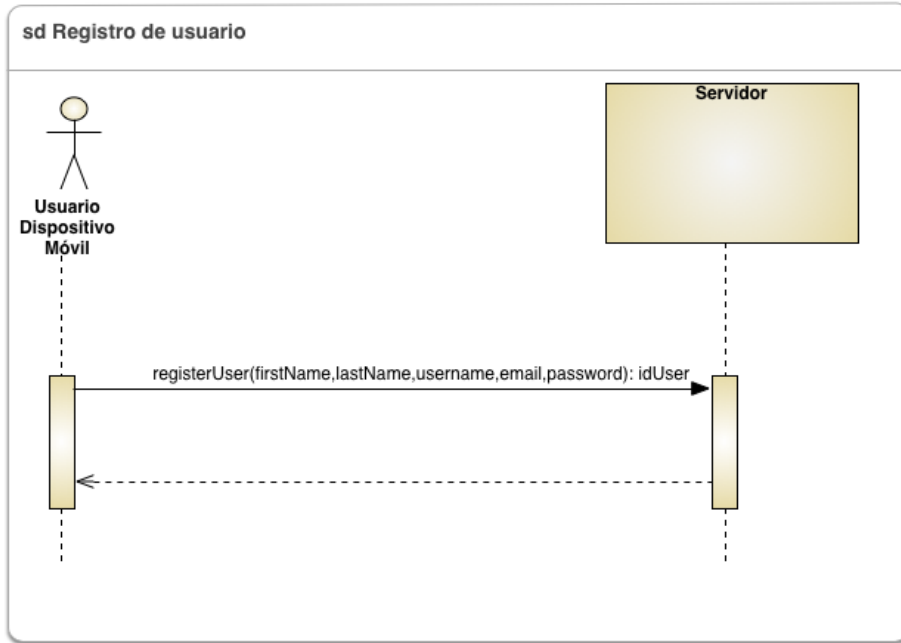


Figura 3.29: Diagrama de secuencia Registro de un usuario

Registro de un usuario	
Precondiciones	
Poscondiciones	Se ha creado un nuevo usuario en la base de datos con los datos de "email", "password", "firstname", "lastname" y "username"
Salida	Se devuelve el id del usuario creado

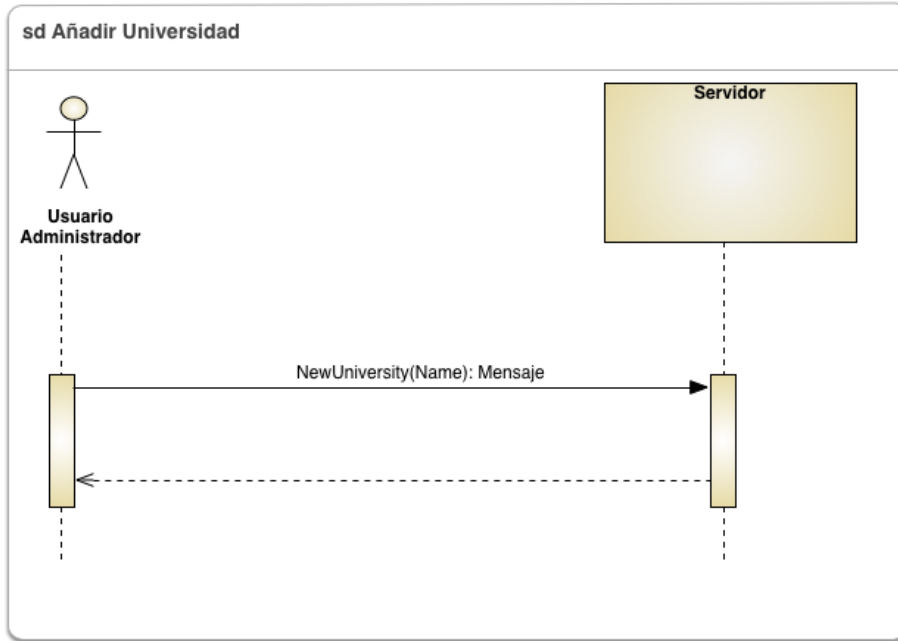


Figura 3.30: Diagrama de secuencia Añadir Universidad

Añadir una universidad	
Precondiciones	
Poscondiciones	Se ha creado una nueva universidad en la base de datos con nombre "Name".
Salida	Se devuelve el id de la universidad almacenada.

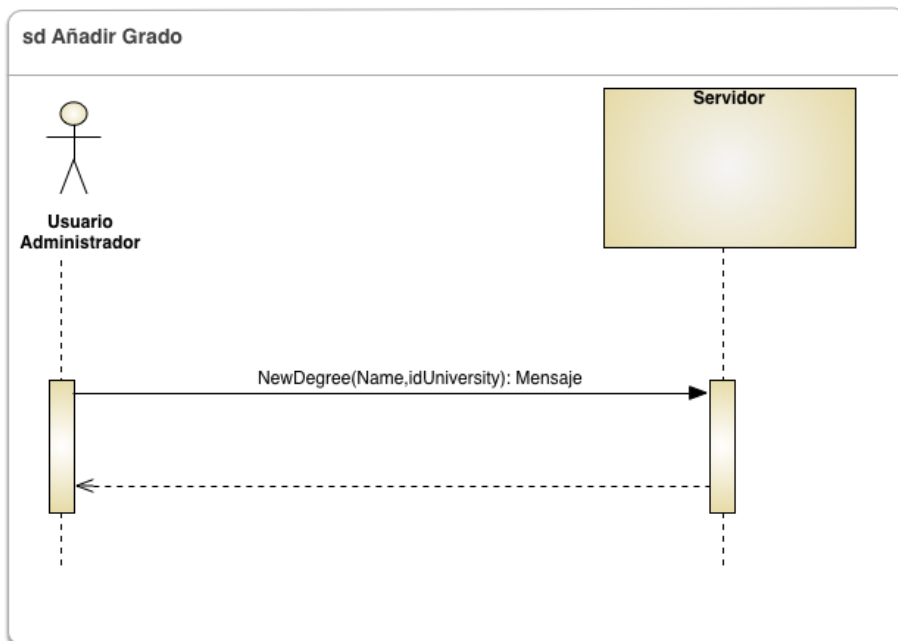


Figura 3.31: Diagrama de secuencia Añadir Grado

Añadir un grado	
Precondiciones	Ha de existir alguna universidad
Poscondiciones	Se ha creado un nuevo grado en la base de datos con nombre "Name" y que pertenece a la universidad "idUniversity"
Salida	Se devuelve el id del grado almacenado.

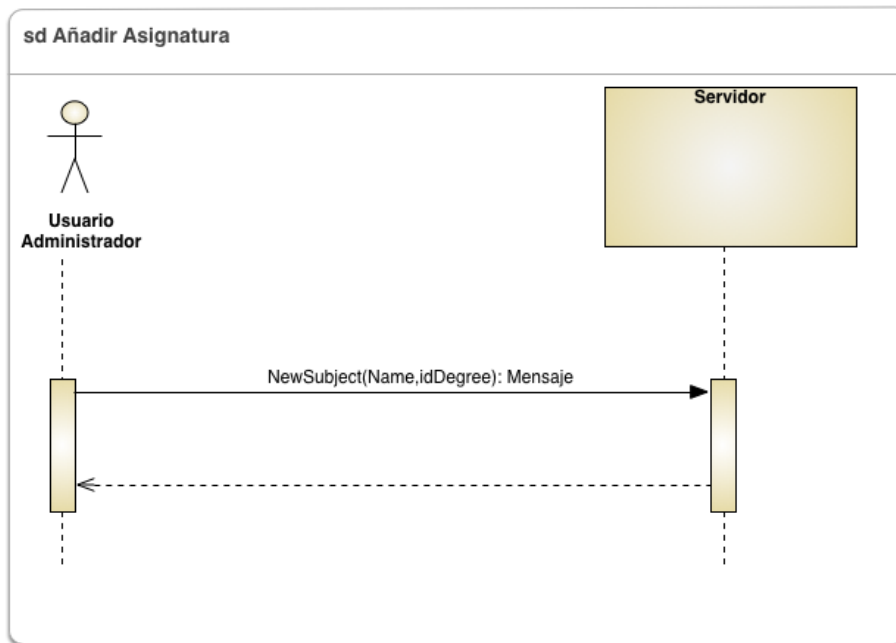


Figura 3.32: Diagrama de secuencia Añadir Asignatura

Añadir una asignatura	
Precondiciones	Ha de existir algún grado
Poscondiciones	Se ha creado una nueva asignatura en la base de datos con nombre "Name" y que pertenece a un grado "idDegree".
Salida	Se devuelve el id de la asignatura almacenada.

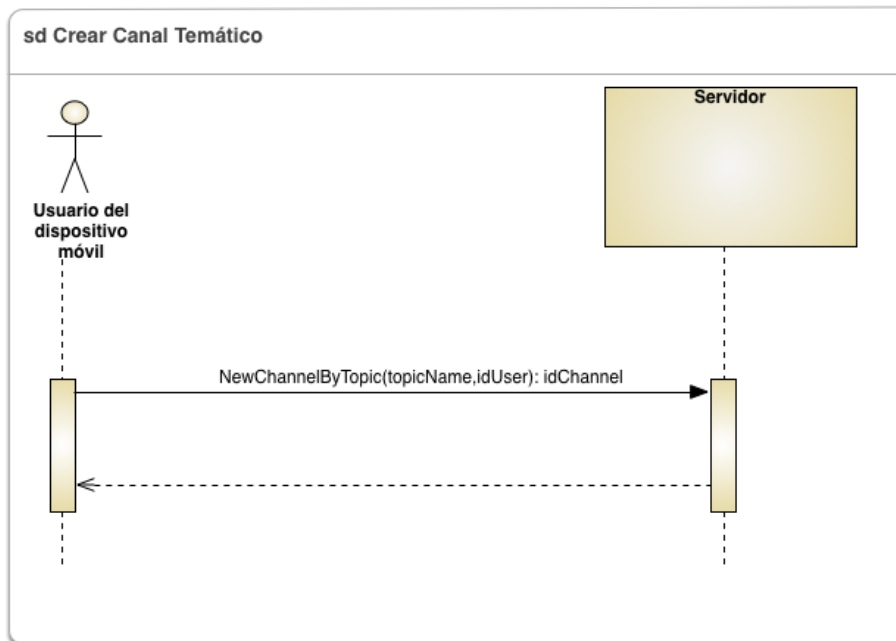


Figura 3.33: Diagrama de secuencia Crear canal temático

Crear canal temático	
Precondiciones	Ha de existir usuarios que tengan información del tema en concreto.
Poscondiciones	Si no existe se crea un canal nuevo con dicho tema "topicName" con los usuarios que más hablen de ese tema y el usuario que realiza la petición "idUser" en concreto en la aplicación y si existe no se realiza ninguna acción en la base de datos
Salida	Se devuelve el id del canal creado o ya existente.

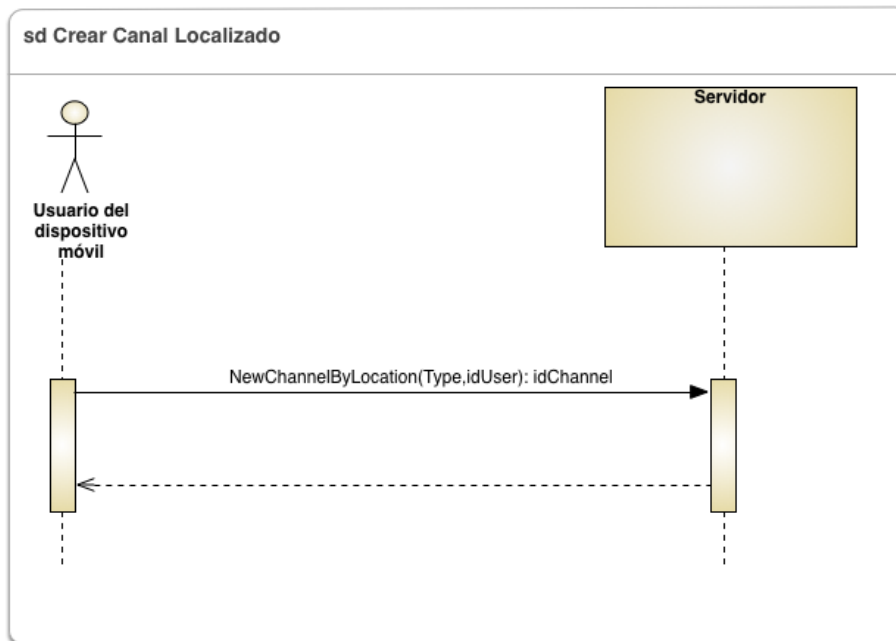


Figura 3.34: Diagrama de secuencia Crear canal localizado

Crear canal localizado	
Precondiciones	Ha de existir usuarios que tengan relación con la localización del usuario .
Poscondiciones	Se crea un nuevo canal con aquellos usuarios que cumplan una serie de características semejantes a dicho usuario "idUser" en relación con su localización y un tipo indicado "Type".
Salida	Se devuelve el id del canal creado.

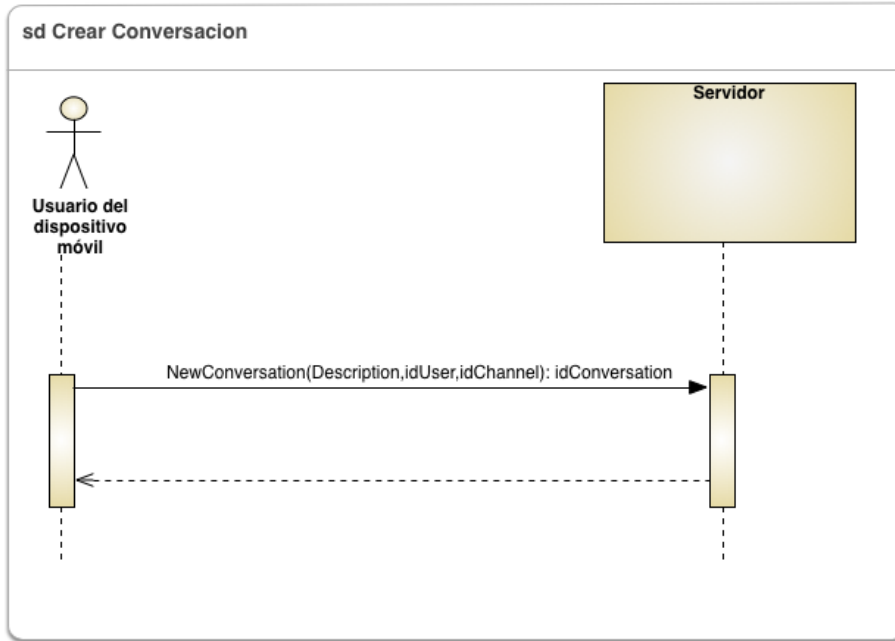


Figura 3.35: Diagrama de secuencia Registro de un usuario

Crear conversación	
Precondiciones	Ha de existir algún canal creado con usuarios registrados en dicho canal.
Poscondiciones	Se crea una nueva conversación en nuestra base de datos con una descripción "Description", un autor "idUser" y en relación a un canal específico "idChannel".
Salida	Se devuelve el id de la conversación.

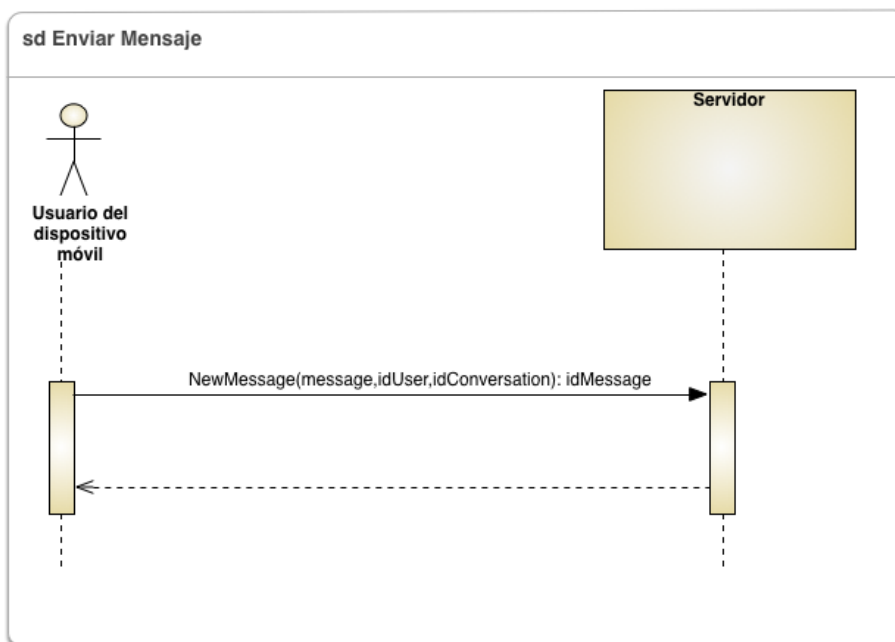


Figura 3.36: Diagrama de secuencia Enviar mensaje

Enviar mensaje	
Precondiciones	Ha de existir alguna conversación.
Poscondiciones	Se crea un nuevo mensaje con texto "message", con origen de un usuario "idUser" y que pertenece a un conversación "idConversation"
Salida	Se devuelve el id del mensaje.

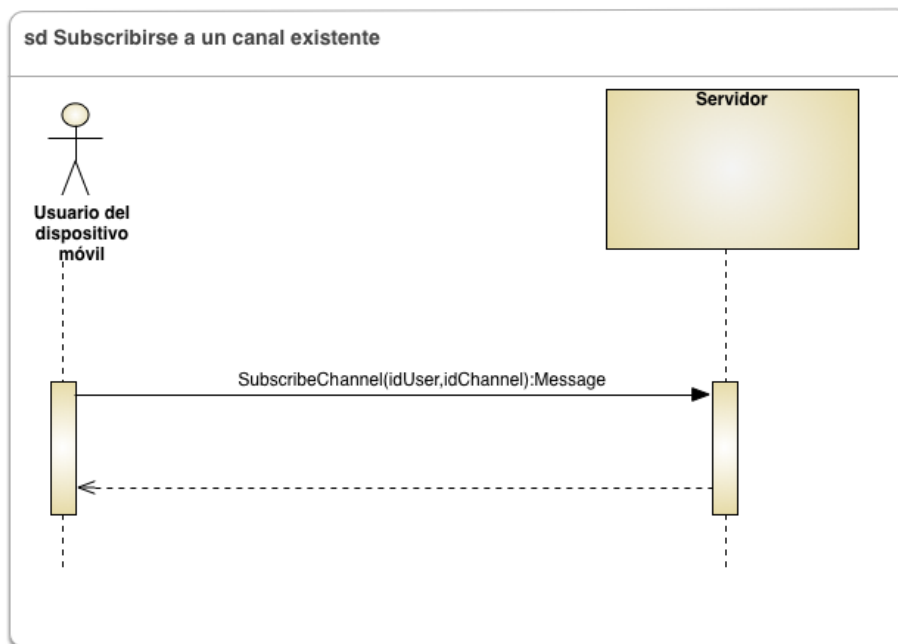


Figura 3.37: Diagrama de secuencia Suscribirse a un canal existente

Subscribirse a un canal existente	
Precondiciones	Ha de existir alguna canal.
Poscondiciones	Se añade una nueva relación entre el usuario "idUser" y el canal ya almacenado en la base de datos "idChannel"
Salida	Se devuelve un mensaje de confirmación.

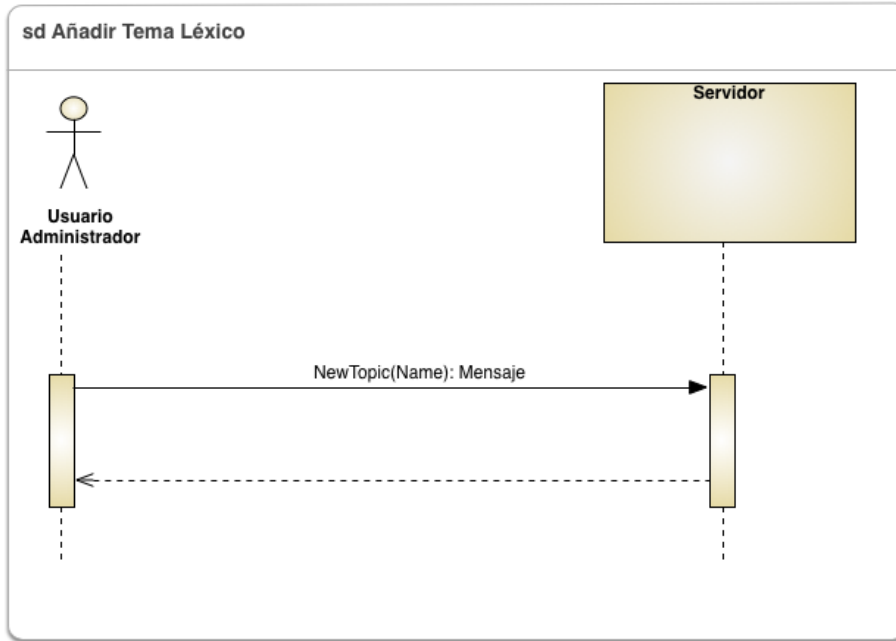


Figura 3.38: Diagrama de secuencia Añadir tema léxico

Añadir tema léxico	
Precondiciones	
Poscondiciones	Se añade un nuevo tema en nuestra base de datos con nombre "Name"
Salida	Se devuelve un mensaje de confirmación.

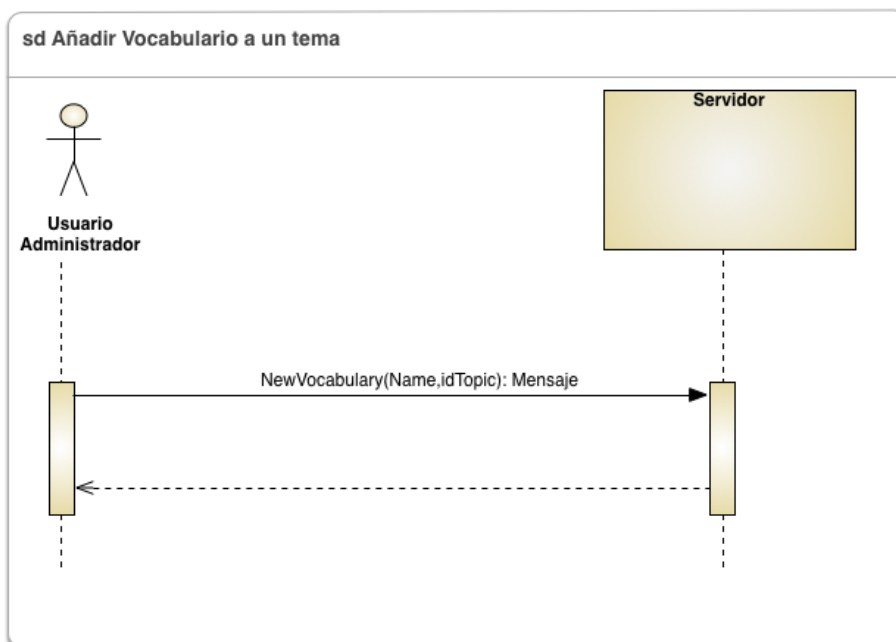


Figura 3.39: Diagrama de secuencia Añadir vocabulario a un tema

Añadir una nueva palabra a un tema léxico	
Precondiciones	Ha de existir algún tema almacenado.
Poscondiciones	Se añade una nueva palabra "Name" a nuestra base de datos en relación con un tema "idTopic"
Salida	Se devuelve un mensaje de confirmación.

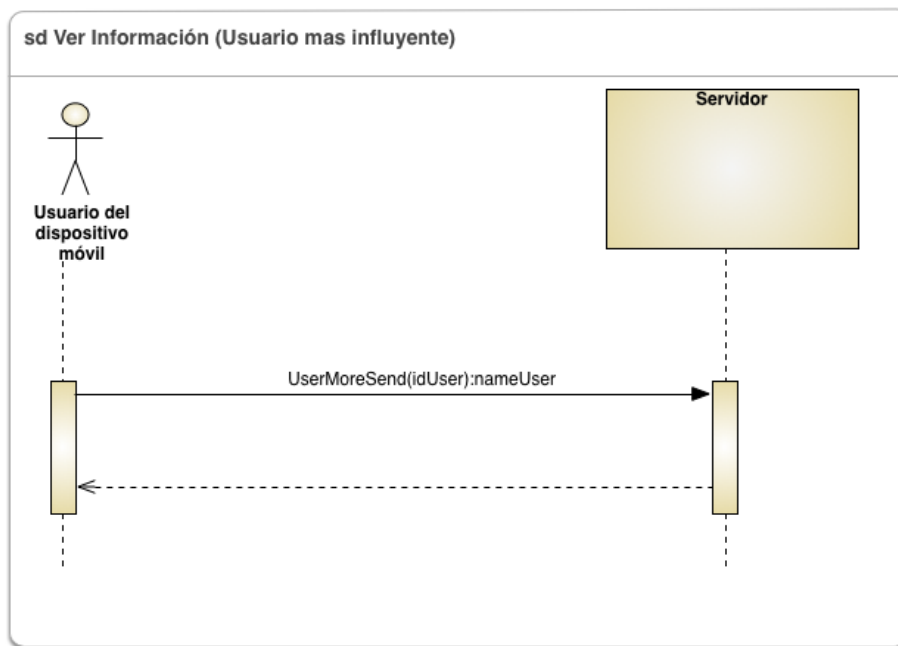


Figura 3.40: Diagrama de secuencia Ver información(Usuario más influyente)

Ver información (Usuario más influyente)	
Precondiciones	
Poscondiciones	Se modifica la información de la base de datos del usuario "idUser"
Salida	Se devuelve el nombre del usuario de quien más ha recibido mensajes el usuario "idUser"

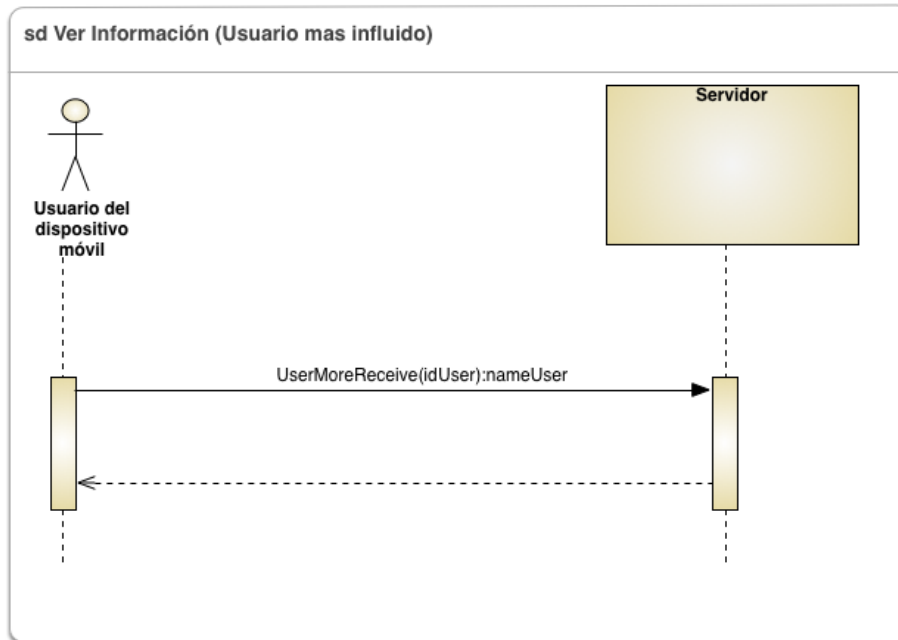


Figura 3.41: Diagrama de secuencia Ver información(Usuario más influido)

Ver información (Usuario más influido)	
Precondiciones	
Poscondiciones	Se modifica la información de la base de datos del usuario "idUser"
Salida	Se devuelve el nombre del usuario que ha recibido más mensajes del usuario "idUser"

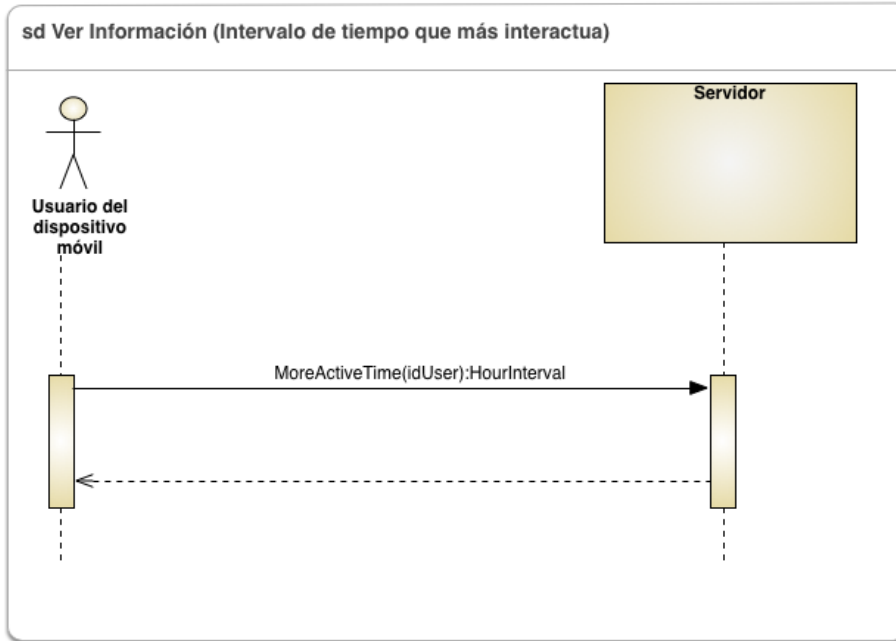


Figura 3.42: Diagrama de secuencia Ver información(Intervalo de tiempo que más interactua)

Ver información (Intervalo de tiempo que más interactua)	
Precondiciones	
Poscondiciones	Se modifica la información de la base de datos del usuario "idUser"
Salida	Se devuelve el intervalo de tiempo el cual el usuario a enviado más mensajes a lo largo del uso de la aplicación.

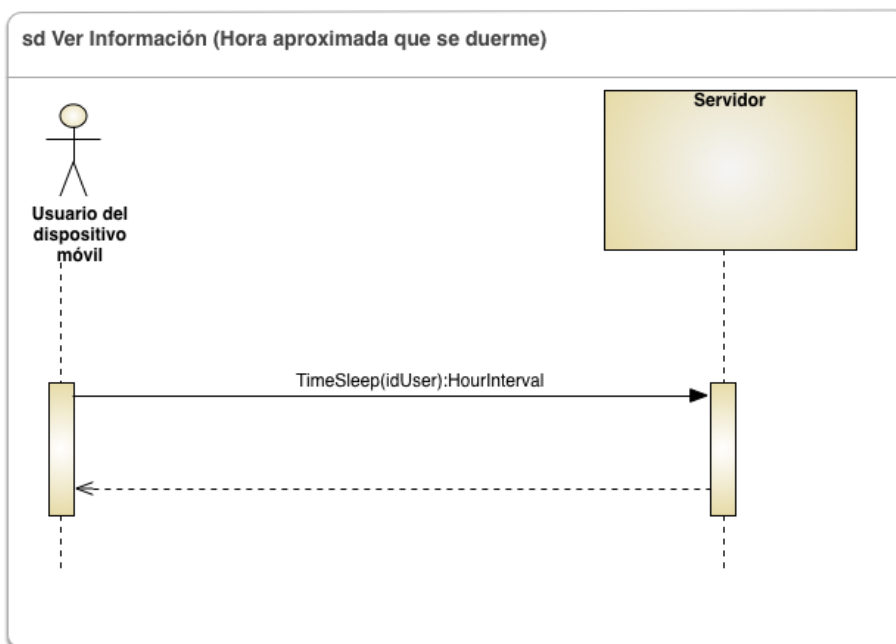


Figura 3.43: Diagrama de secuencia Ver información(Intervalo de tiempo que más interactura)

Ver información (Hora que se duerme)	
Precondiciones	
Poscondiciones	Se modifica la información de la base de datos del usuario "idUser"
Salida	Se devuelve el intervalo de tiempo aproximado en el que el usuario de duerme.

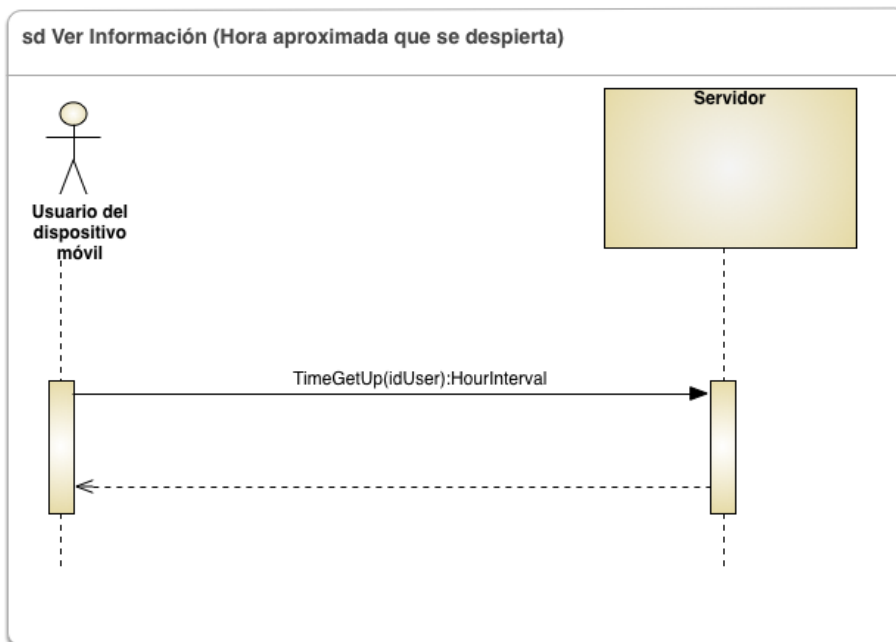


Figura 3.44: Diagrama de secuencia Ver información(Hora aproximada que se despierta)

Ver información (Hora que se despierta)	
Precondiciones	

Ver información (Hora que se despierta)	
Poscondiciones	Se modifica la información de la base de datos del usuario "idUser"
Salida	Se devuelve el intervalo de tiempo aproximado en el que el usuario se despierta.

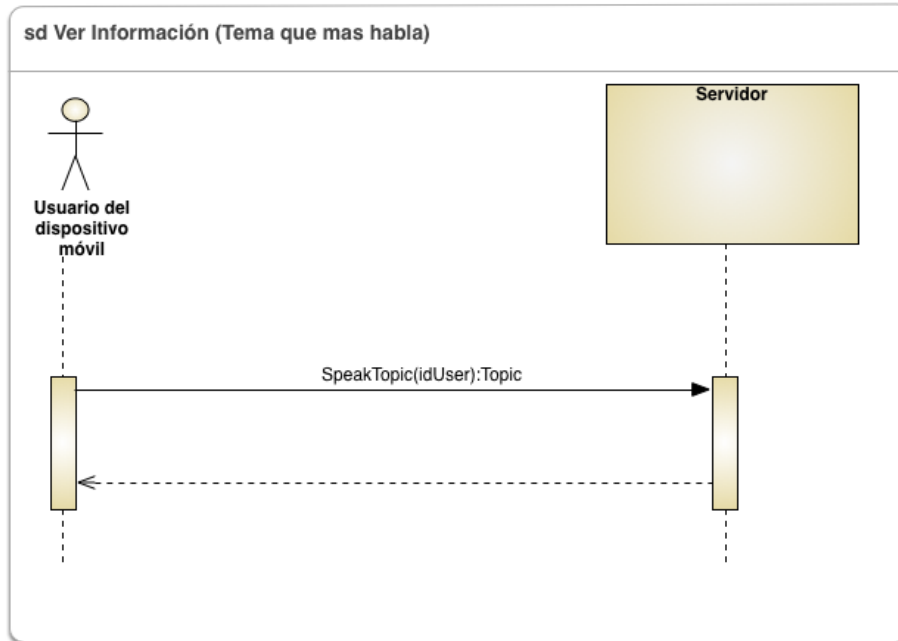


Figura 3.45: Diagrama de secuencia Ver información(Tema que más habla)

Ver información (Tema que más habla)	
Precondiciones	
Poscondiciones	Se modifica la información de la base de datos del usuario "idUser"
Salida	Se devuelve el tema más hablado por el usuario.

4-Diseño

En la fase siguiente se va a estudiar y a definir como va a funcionar el sistema, basándonos en su arquitectura, y especificar todos los detalles para permitirnos pasar a la siguiente fase que es la implementación.

Los objetivos principales del diseño son:

- Construir el sistema en base al lenguaje de programación utilizado y/o sistema.
- Analizar con suficiente detalle todos los requisitos no funcionales para permitir construir un sistema con una arquitectura que garantice la calidad deseada.
- Descomponer la fase de implementación en varias partes para poder facilitar toda la gestión de la misma y así conseguir un transcurso del proyecto más sencillo y eficiente respecto al tiempo.
- Crear una abstracción de todos los componentes del sistema para poder facilitar el mantenimiento del proyecto una vez finalizado el proceso y la facilidad de identificación de los componentes cuando se necesite hacer algún tipo de mejora.

En primer lugar se procederá a explicar la arquitectura física de los sistemas que se adapten mejor para poder conseguir nuestros objetivos.

A continuación se explicará las tecnologías usadas que estarán ligadas tanto al diseño de la plataforma móvil como a la plataforma web.

Para finalizar se realizará una descripción detallada de la arquitectura lógica del sistema global para poder identificar los componentes software de nuestro sistema.

4.1 Arquitectura física

Al encontrarnos con un sistema donde el cliente es un dispositivo móvil que utiliza diversos servicios de un web service nos encontramos con un claro modelo Cliente-Servidor.

Esta arquitectura es un modelo de aplicación distribuida en el que las tareas se reparten entre diferentes recursos en este caso el servidor que es el encargado de suministrar los servicios y el cliente que es el encargado de consumir dichos servicios.

Una de las características principales es la organización que tiene este modelo permitiendo una centralización de la gestión de la información y una separación en las responsabilidades lo que facilita el diseño de un sistema.

La separación entre cliente y servidor es una separación lógica donde el servidor no tiene porque ejecutarse en una única máquina.

Una de las opciones planteadas para este sistema es un modelo multicapa permitiendo descomponer la carga de trabajo del servidor en varias máquinas para tener un equilibrado y puede llegar a ser una opción en el caso de que la aplicación tenga un gran número de usuario accediendo al sistema en un mismo momento.

Las principales ventajas de utilizar este modelo cliente-servidor son las siguientes:

- Centralización del control: los accesos concurrentes, los recursos y la integridad de los datos son controlados por el servidor de tal forma que un cliente no autorizado no pueda dañar el sistema o en el caso de que ocurra se pueda volver a un estado anterior.
- Escalabilidad: se puede aumentar las capacidades de los clientes y servidores sin que afecte al sistema en cualquier momento.

- Fácil de mantenimiento: Al tener bien identificados los roles de cada componente se pueden mejorar, solucionar o reemplazar un componente sin que afecte de manera drástica al resto del sistema.
- Existen tecnologías que aseguran la seguridad de las transacciones, la amigabilidad de la interfaz y la usabilidad.

El sistema global esta compuesto por los clientes que son los dispositivos móviles y el administrador que accede a la pagina web de administración que utilizan los servicios ofrecidos por el servidor.

Este último es el encargado de gestionar toda la lógica de negocio y todos los datos.

En el caso de los clientes podemos dividirlos en los dispositivos móviles que utilizan los Web Service para comunicarse con el servidor que es el encargado de proporcionar dichos servicios y la aplicación web que es el intermediario entre el administrador y los datos almacenados en la base de datos.

A continuación se observa brevemente las tecnologías usadas:

- La aplicación web estará desarrollada con Web Ratio
- La aplicación móvil estará desarrollada con Android y actuará como cliente.
- Los servicios web estará desarrollada con RESTful.
- La comunicación se realizará en formato JSON.
-

Tanto la aplicación web como los servicios web estarán alojados en un servidor de la universidad para tener una disponibilidad total para todos los usuarios que utilicen esta aplicación.

A continuación se puede observar de forma esquemática como está construido el sistema de comunicación entre los dispositivos móviles y el servidor.

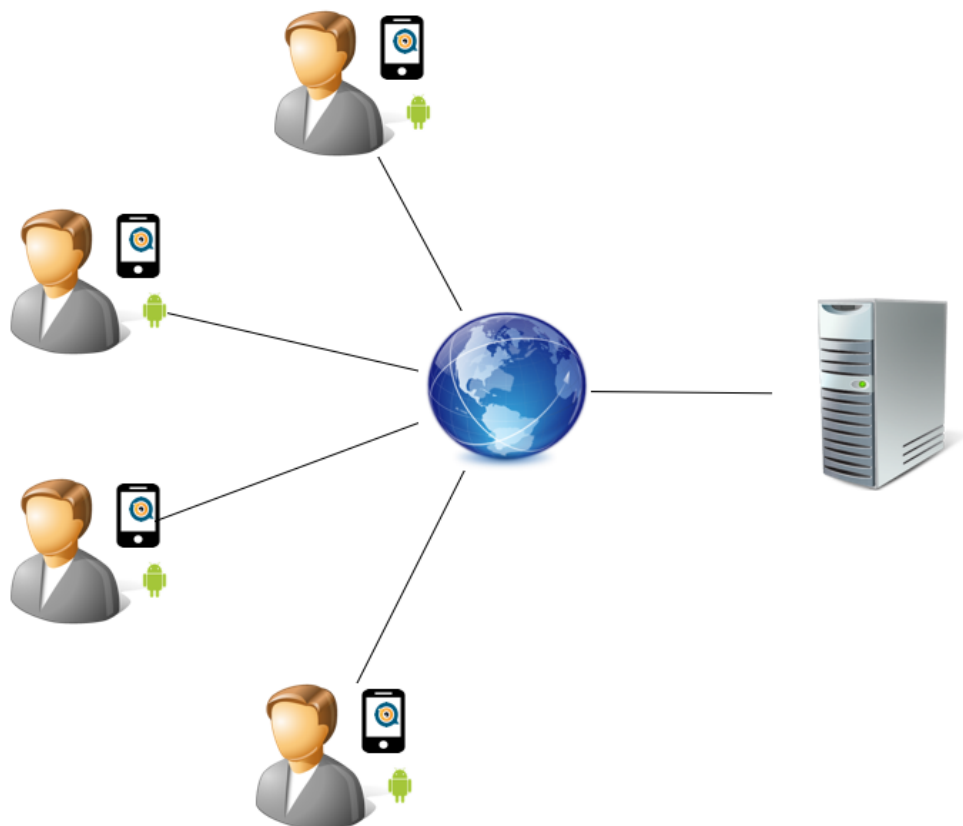


Figura 4: Descripción global de interacción entre dispositivos y servidor

Finalmente se muestra el esquema de como esta construido el sistema para la gestión del administrador.



Figura 4.1: Descripción global de interacción entre aplicación web y servidor

4.2 Tecnologías usadas

En esta sección se van a definir las tecnologías escogidas en el capítulo de análisis inicial, usadas tanto para la aplicación web, aplicación móvil y comunicación de datos.

4.2.1 Android

Android es un sistema operativo basado en Linux y dedicado mayoritariamente a dispositivos móviles y tablets. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. Además Google posee un mercado llamado google play donde se pueden colgar todas las aplicaciones desarrolladas, escritas en lenguaje de programación Java.



Figura 4.2: Icono Android con numerosas aplicaciones.

El anuncio del sistema Android se realizó el 5 de Noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia de Apache, una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL, y una biblioteca estándar de C. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2.8 millones de líneas de lenguaje C, 2.1 millones de líneas de Java y 1.75 millones de líneas de C++.

Para describir mejor el sistema, vamos a mencionar las partes por las que se compone Android:

- Núcleo: Android depende de Linux para los servicios base del sistema así como la seguridad, la gestión de procesos, pila de red y la gestión de los controladores.
- Bibliotecas: Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema expuestas para su uso por los desarrolladores y un conjunto de bibliotecas que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje java.
- Marco de trabajo de aplicaciones: Los desarrolladores tienen acceso completo al SDK de Android para poder facilitar la creación de nuevas aplicaciones.
 - * -Actividades
 - * -Servicios

- * -Proveedores de contenido
- * -Receptores de transmisión

- Aplicaciones: La capa de aplicaciones, es el entorno de instalación de diferentes aplicaciones de Android.

Se puede observar todas las partes en la siguiente figura.



Figura 4.3: Componentes de Android
Ciclo de vida de sus componentes

Todos los procesos tienen un tiempo de vida, desde que son iniciadas hasta el momento que son destruidos, aunque durante la vida de cada uno de estos componentes, pasan por una serie de estados.

Por ejemplo el ciclo de vida de una actividad se muestra en la siguiente figura.

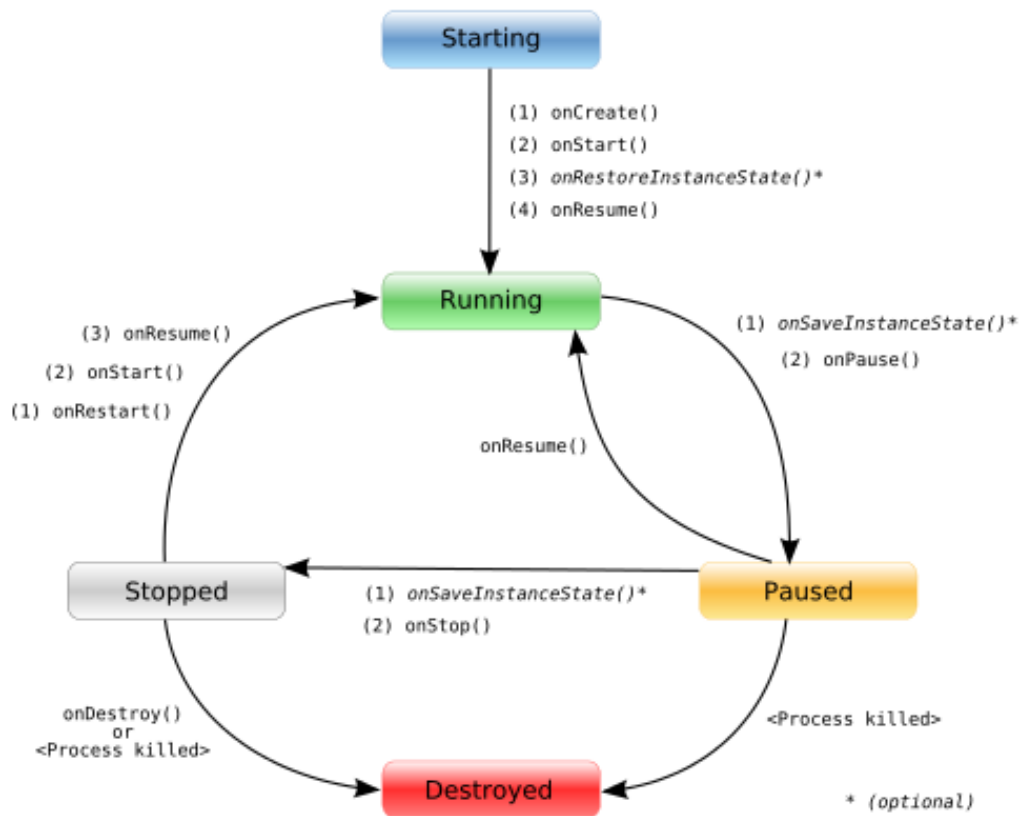


Figura 4.4: Ciclo de vida de un actividad Android

Y los métodos utilizados en las distintas transiciones de estado se pueden observar en la siguiente figura.

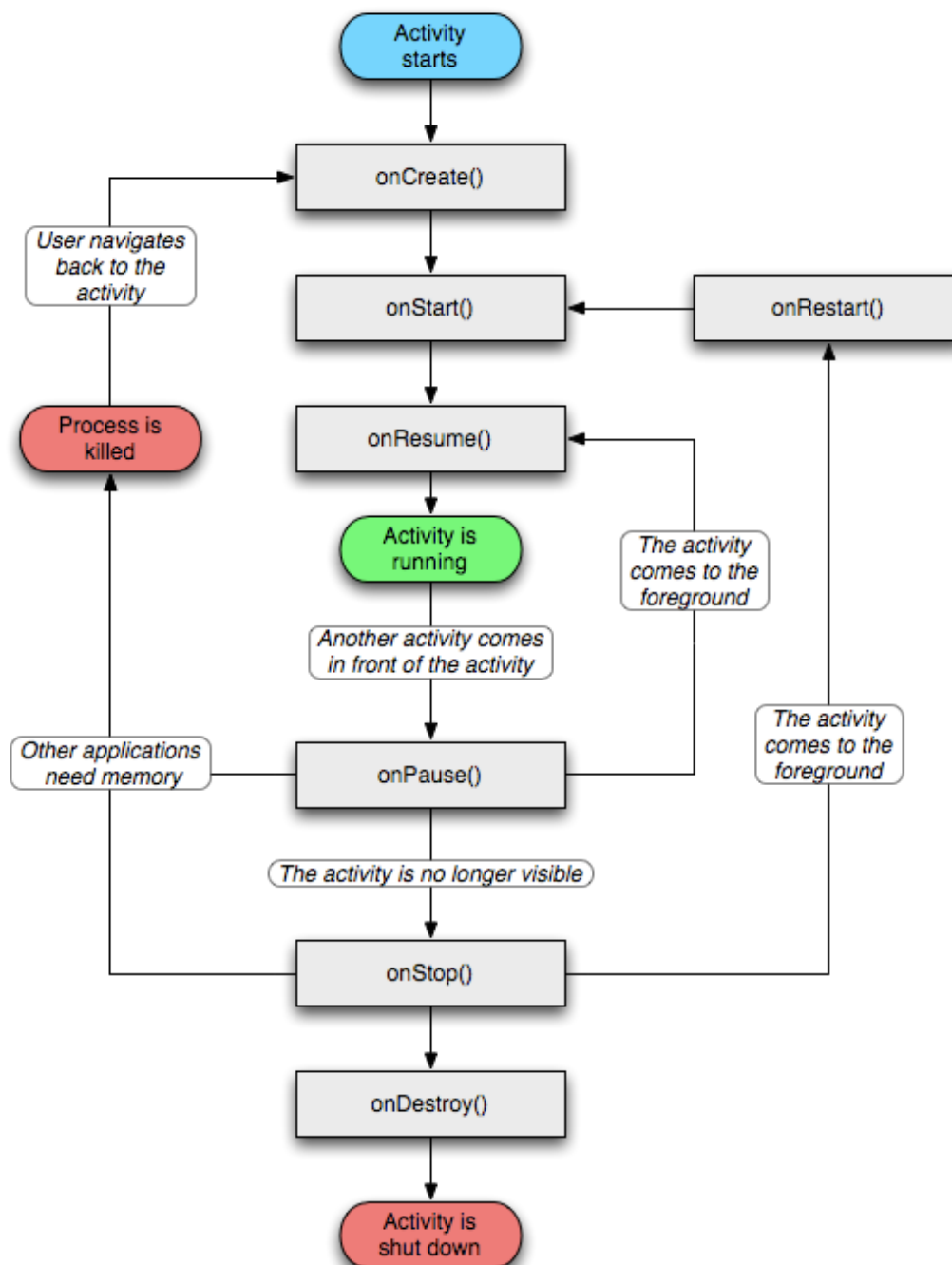


Figura 4.4: Tareas ciclo de vida de una actividad Android

4.2.2 *WebRatio*

WebRatio es una herramienta que facilita la definición de una página web y se ha seleccionado esta herramienta por una serie de motivos:

- Dispone de las funcionalidades necesarias para diseñar modelos de datos y navegacionales utilizando una metodología específica: WebML.
- WebRatio crea un único fichero XML con todos los modelos definidos. La sintaxis de estos ficheros no puede considerarse excesivamente complicada.
- Permite utilizar prácticamente cualquier gestor de base de datos.
- A petición del usuario, crea en la base de datos y de forma automática las tablas, atributos y relaciones indicados en el modelo conceptual.
- Genera automáticamente el código Java de una aplicación Web a partir del modelo navegacional.
- La metodología WebML la forman un conjunto de elementos no demasiado extenso, lo que facilita su estudio y aprendizaje.

Para poder conseguir una aplicación web de manera automática es necesario llevar a cabo 3 fases claramente identificadas:

Fase 1: Se crea un modelo de datos con webratio. Esto crea un fichero XML con el modelo de datos e información adicional.

Fase 2: El programa implementado carga este fichero XML en una estructura acorde con las necesidades y partiendo del modelo de datos anteriormente definido se crea un grafo navegacional refactorizado y su correspondiente modelo navegacional. Para finalizar el

programa crea un fichero XML con todos los datos necesarios para que pueda ser procesado por WebRatio.

Fase 3: Se abre el fichero XML resultado de la fase 2 con webratio y se genera el código de la aplicación automáticamente.

4.3.3 Comunicación entre la aplicación móvil y los servicios web

Para esta comunicación se va a utilizar RESTful que es una propuesta de Roy Fielding que permite manejar los servicios web con métodos definidos, manteniendo la simpleza del protocolo como XML, pero que cada servicio sea identificado únicamente con un solo URI.

Para comenzar debemos entender que necesitamos una clase para manejar un servicio. En esta clase solo puede haber máximo cuatro métodos públicos que son ejecutados por los cuatro métodos HTTP disponibles para RESTful.



Figura 4.4: Funciones Rest

Para el intercambio de información utilizaremos JSON que es un formato ligero para el intercambio de información. Su simplicidad y compactación ha dado lugar a la generalización de su uso, alternativamente a XML.



Figura 4.3: Símbolo JSON

Son formatos similares y cada uno cuenta con seguidores y detractores pero la principal diferencia entre ambos, es que para la transferencia de una mayor cantidad de datos, JSON es bastante más ligero que XML y para una plataforma de comunicación en la cual la transmisión de mensajes es continua necesitamos un formato de comunicación que sea eficiente.

Lenguaje de desarrollo

Para poder seleccionar un lenguaje de desarrollo es necesario realizar una comparativa entre los grandes de este sector, debido a que hay muchos lenguajes y de diferentes características y se convertiría en algo inmanejable para esta documentación.

Para comparar he seleccionado aquellos lenguajes dados a lo largo de la carrera y por lo tanto con conocimientos ya adquiridos de dichos lenguajes que nos facilita el proceso de cercanía con el lenguaje.

Tenemos lenguajes como c++, c o java que se tienen ya bastantes conocimientos de dichos lenguajes.

En el caso de c es un lenguaje ya con bastante rodamiento en el mundo de la informática pero es un lenguaje un poco feo por parte del desarrollador debido a la inexistencia de la programación orientada a objetos pero lo compensa con su alta eficiencia.

C++ pretende mejorar el lenguaje anteriormente definido incluyendo el paradigma de programación orientado a objetos algo que es indispensable para nuestro proyecto.

Por ultimo tenemos Java, un lenguaje moderno y muy cercano a los lenguajes anteriormente definidos que puede ejecutarse en cualquier sistema operativo gracias al uso de una máquina virtual (JVM).

Es un lenguaje orientado a objetos que una de las principales características es la gran cantidad de librerías ya implementadas para este lenguaje que facilitan mucho el trabajo de un desarrollador.

En la siguiente figura se puede observar el indice de popularidad de los lenguajes de programación de mayor auge.

Position Aug 2013	Position Aug 2012	Delta in Position	Programming Language	Ratings Aug 2013	Delta Aug 2012	Status
1	2	↑	Java	15.978%	-0.37%	A
2	1	↓	C	15.974%	-2.96%	A
3	4	↑	C++	9.371%	+0.04%	A
4	3	↓	Objective-C	8.082%	-1.46%	A
5	6	↑	PHP	6.694%	+1.17%	A
6	5	↓	C#	6.117%	-0.47%	A

Figura 4.4: Comparación lenguajes de programación.

Java, la mejor solución

Java es un lenguaje moderno compatible con todas las plataformas gracias a que utiliza una máquina virtual (JVM) y la gran cantidad de librerías que facilitan el desarrollo de cualquier proyecto en comparación con c y c++.

4.3 Arquitectura lógica

La arquitectura lógica describe como están diseñados cada uno de los sistemas y como se integran y organizan cada componente basándonos en el uso de patrones.

Esta arquitectura se encarga de explicar los subsistemas que puedan existir y como se comunican entre ellos para poder conseguir un objetivo, es un paso anterior a la implementación del proyecto y se considera una de las secciones fundamentales en un proyecto software porque puede considerarse una interfaz entre el código y los requisitos definidos en el análisis inicial.

4.3.1 Interfaz gráfica

La interfaz gráfica es la parte del sistema que está en contacto con el usuario y se encarga de comunicarle la información y capturar todos los gestos o movimientos que realice el usuario. Podemos dividir el diseño en dos partes, una sería la interfaz gráfica de la aplicación móvil y otra sería la interfaz gráfica de la aplicación web.

Hay que destacar que en este apartado no tendría importancia explicar la interfaz gráfica de los servicios web porque no existen como tal y no están en contacto directo con los usuarios. A continuación se describe el diseño para cada una de las aplicaciones.

Diseño de la aplicación web

Es en este apartado se explicará el diseño de nuestra pagina web y como funciona su navegación. El diseño de dicha página es simple e intuitiva teniendo únicamente las funciones necesarias para el administrador.

En la siguiente figura se muestra como sería la interfaz de nuestra aplicación web para añadir un nuevo tema y a su vez nuevo vocabulario.

Gestión de temas

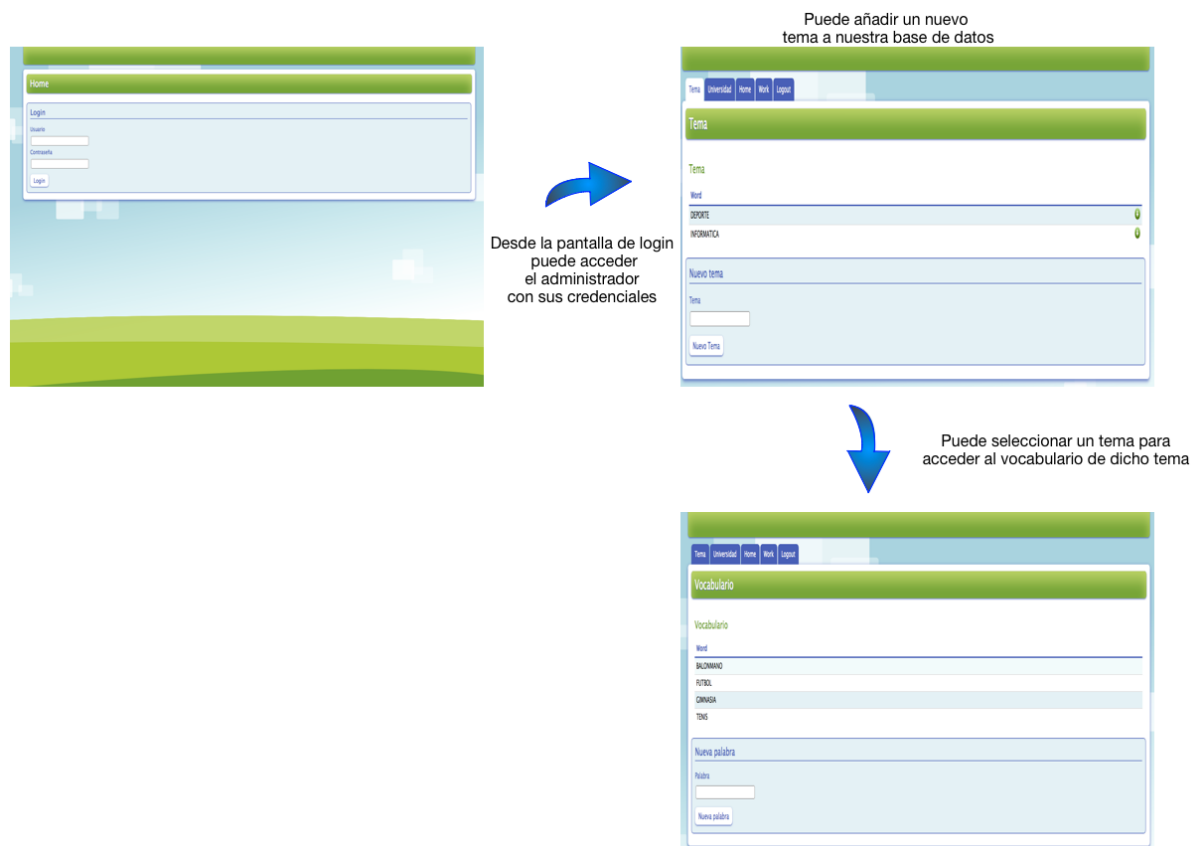


Figura 4.5: Interfaz aplicación web (Gestión temas)

A continuación se mostrará como sería la interfaz para añadir una nueva universidad, un nuevo grado o una nuevo asignatura.

Hay que destacar que para añadir un grado hay que elegir previamente una universidad y a su vez para añadir una asignatura hay que seleccionar un grado.

Gestión de universidades

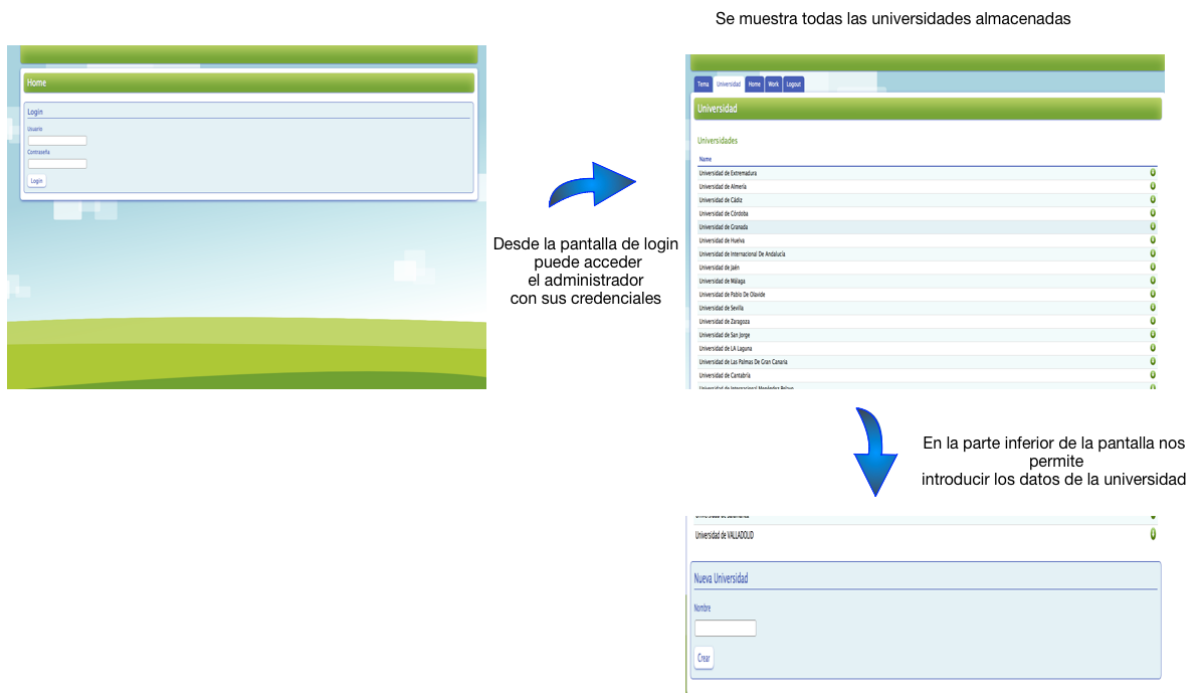


Figura 4.6: Interfaz aplicación web (Gestión universidades)

Gestión de grados

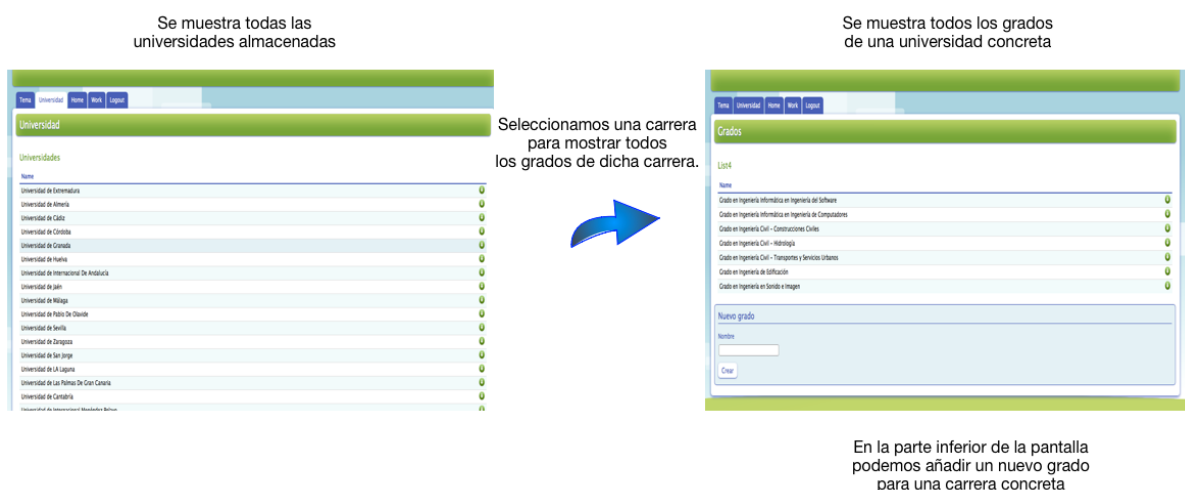
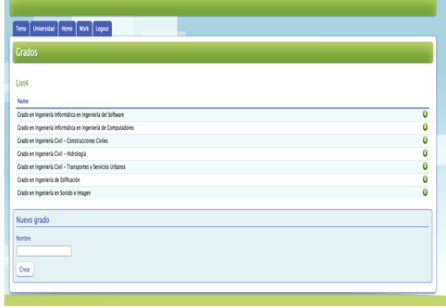



Figura 4.6: Interfaz aplicación web (Gestión grados)

Gestión de asignaturas

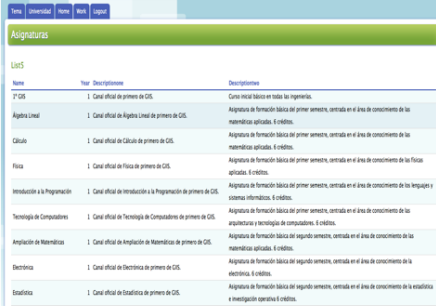
Se muestra todos los grados almacenados para una universidad concreta



Seleccionamos un grado para mostrar todas las asignaturas de un grado concreto.



Se muestra todos las asignaturas de un grado concreto



En la parte inferior de la pantalla podemos añadir una nueva asignatura para una grado concreto.




Figura 4.7: Interfaz aplicación web (Gestión asignaturas)

Mapa de navegación del administrador

En este apartado se pretende mostrar las funcionalidades de manera gráfica del administrador en la pagina web.

Esta página web únicamente tiene funcionalidades para los administradores, es decir, un usuario normal no puede acceder a esta página. En la siguiente figura se muestra cuales son los pasos en la navegación a través de la página web.

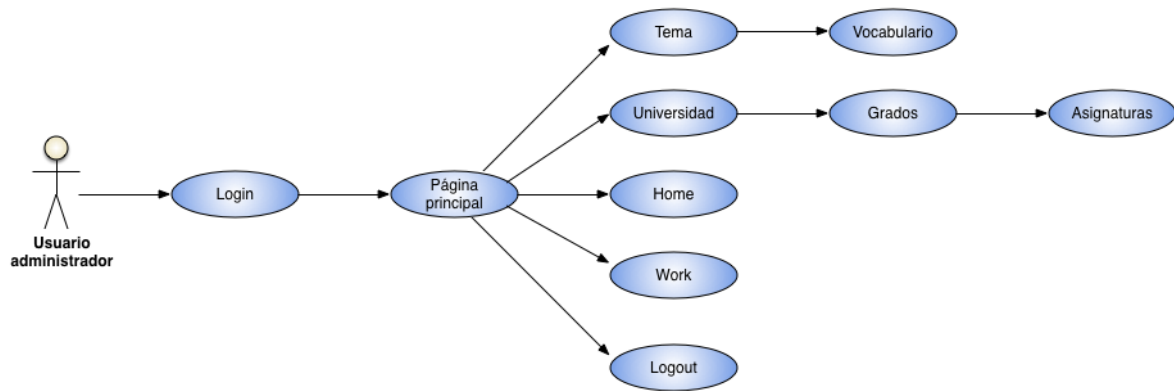


Figura 4.8: Funcionalidades aplicación web

5-Implementación

Una vez visto el diseño de nuestro proyecto, se va a proceder a describir de forma detallada las tareas realizadas durante el desarrollo de nuestra pagina web y el desarrollo de los servicios web ofrecidos para ser consumidos desde los dispositivos móviles.

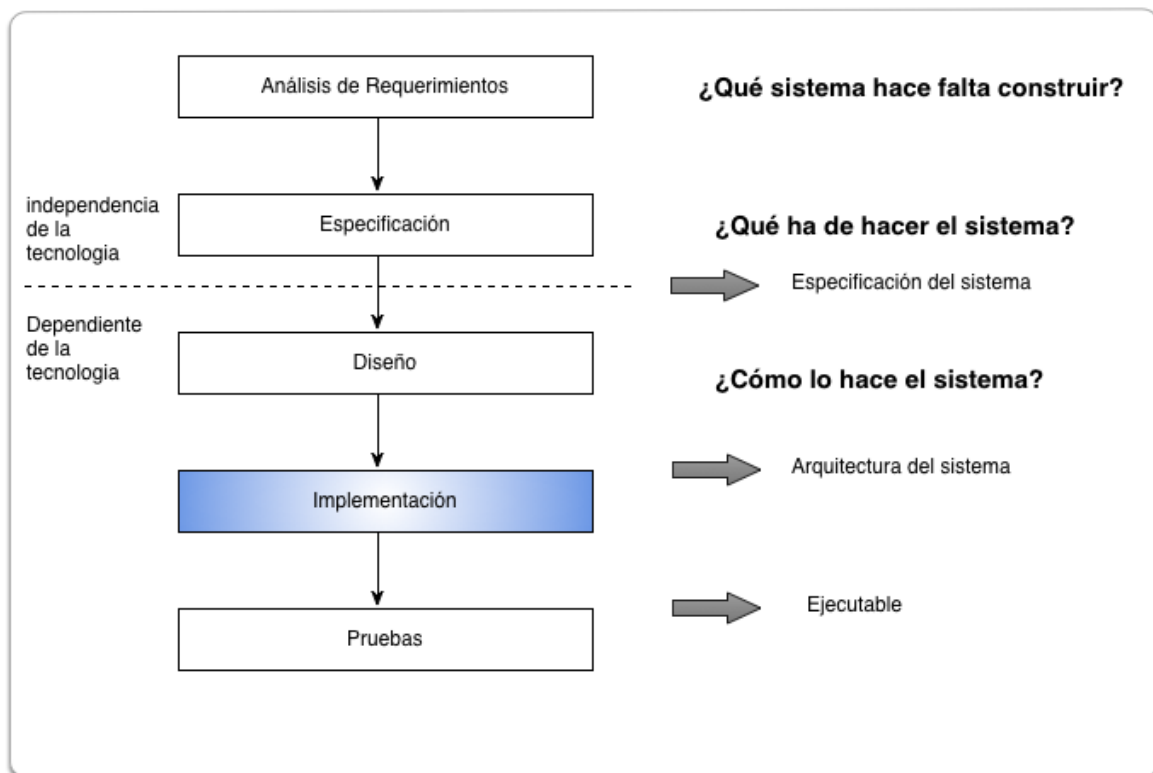


Figura 5: Fase de implementación

En esta sección se explica como se ha llevado a cabo cada funcionalidad de nuestro proyecto y como se relaciona la interfaz definida en el apartado anterior y la implementación. Todas las funcionalidades se explicaran detalladamente que objetivo tiene, que necesita, que proceso lleva a cabo y que salida devuelve. En primer lugar se empezará a explicar los servicios web implementados en java y finalmente se definirá como se ha realizado la aplicación web con webratio.

5.1 Web Service

En este apartado se explicará todo el proceso para desarrollar los servicios web necesarios para la comunicación entre los dispositivos móviles. En primer lugar se explicará que requisitos hemos utilizado para desarrollar estos servicios, que herramientas se han utilizado y como se configuran, como se importa nuestro proyecto y finalmente se explicará en detalle cada función desarrollada.

5.1.1 Requisitos necesarios

Para poder desarrollar un web service con estas características es necesario tener una máquina con java instalado. Para instalar java basta con acceder al siguiente enlace y descargarnos el JDK <http://www.oracle.com/technetwork/es/java/javase/downloads/jdk7-downloads-1880260.html>.

Una vez descargado e instalado utilizaremos la herramienta de desarrollo eclipse para poder definir estos servicios web. Es necesario instalar la versión “Eclipse IDE for Java EE Developers” disponible en el siguiente enlace <http://www.eclipse.org/downloads/>

Eclipse Luna (4.4) Release for Mac OS X (Cocoa)

Mac OS X users please note: Eclipse requires Mac OS X 10.5 (Leopard) or greater.

Eclipse Standard 4.4 205 MB
Downloaded 97,819 Times [Other Downloads](#)

Standard Eclipse package suited for Java and plug-in development plus adding new plugins; already includes Git, Marketplace Client, source code and...

Mac OS X 32 Bit
Mac OS X 64 Bit

Package Solutions Filter Packages

Eclipse IDE for Java EE Developers 257 MB
Downloaded 83,825 Times

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

Mac OS X 32 Bit
Mac OS X 64 Bit

Eclipse IDE for C/C++ Developers 164 MB
Downloaded 23,675 Times

An IDE for C/C++ developers with Mylyn integration.

Mac OS X 32 Bit
Mac OS X 64 Bit

JRebel for Eclipse IDE

See Java Code Changes Instantly. Save Time. Reduce Stress. Finish Projects Faster!

Promoted
Download

Eclipse IDE for Java Developers 154 MB
Downloaded 22,499 Times

The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

Mac OS X 32 Bit
Mac OS X 64 Bit

Figura 5.1: Página web eclipse descargar

Es necesario que cada desarrollador descargue la versión correspondiente con su máquina que en este caso se utiliza Mac OS X 10.9.1.

Una vez descargado la versión correspondiente podemos encontrar un ejecutable dentro de la carpeta anteriormente descargada que será el encargado de ejecutar nuestra herramienta. Una vez ejecutada nos mostrará la siguiente interfaz.

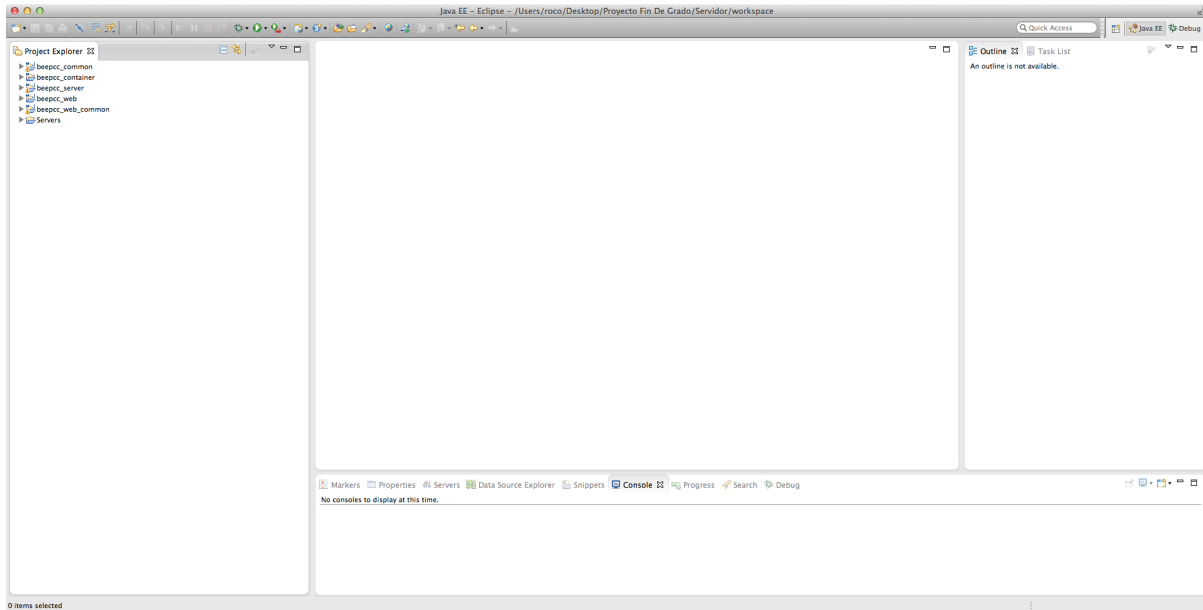


Figura 5.2: Interfaz eclipse

Una vez que tengamos nuestra herramienta funcionando correctamente necesitamos importar nuestro proyecto y para ello accedemos a File->Import->Existing Maven Projects y seleccionamos los proyectos adjuntos para los servicios web.

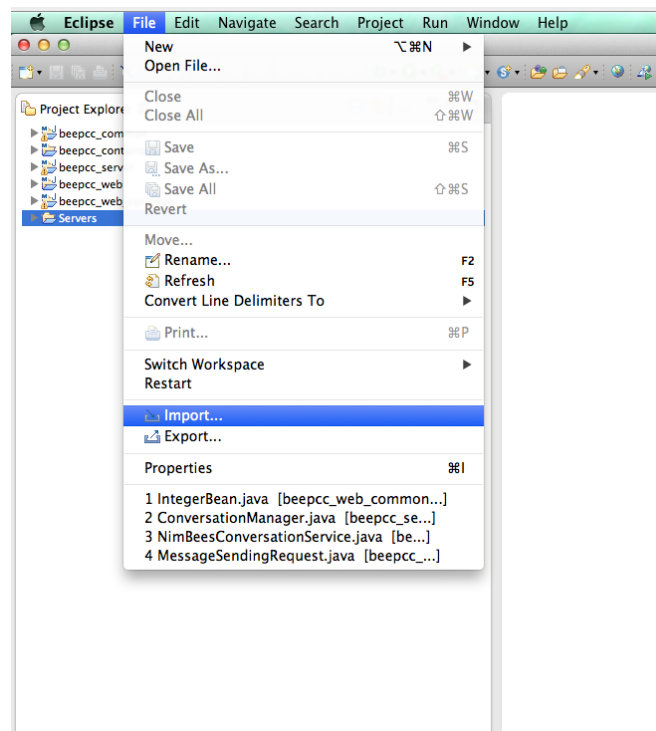


Figura 5.3: Import eclipse

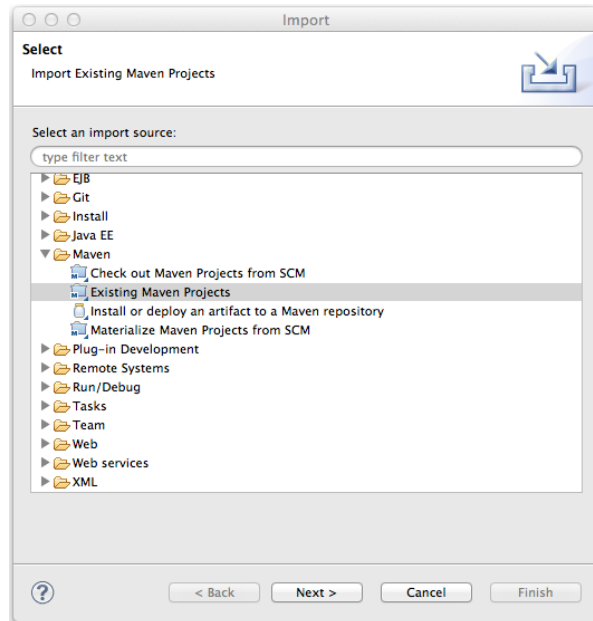


Figura 5.4: Maven project

Una vez importado los proyectos es necesario que maven descargue todas las librerías necesarios para poder compilar y ejecutar nuestro proyecto.

Para ello en primer lugar limpiamos todos los proyectos de archivos generados automáticamente. Para este paso es necesario acceder con el botón derecho en cada proyecto Run as -> Maven clean.

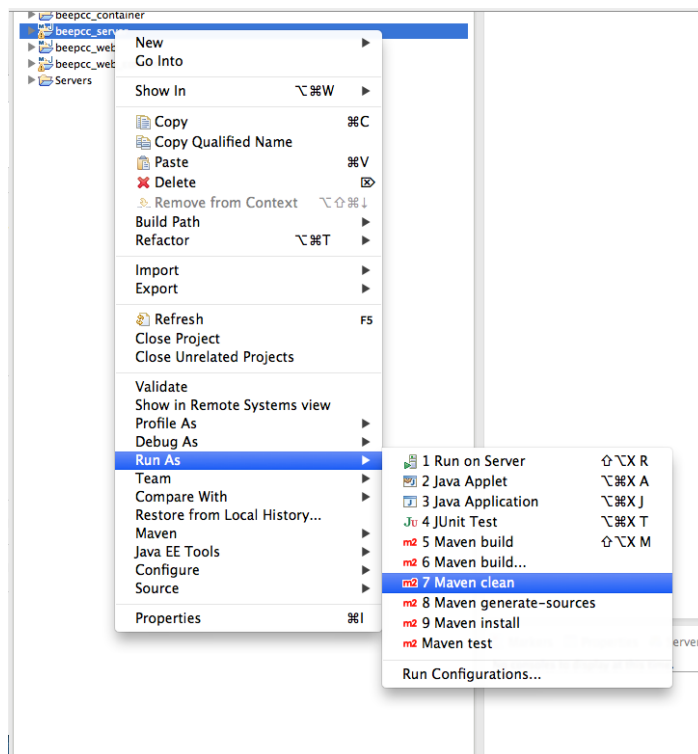


Figura 5.5: Maven clean

Una vez limpiado todos nuestro proyectos pasamos a compilarlos para que maven descargue todos los plugin necesarios para este proyecto. Esta es una de las características principales de maven que facilita todo este proceso bastante costoso de buscar manualmente los plugin que sean de utilidad para el proyecto.

Para este paso, al igual que antes, con el botón derecho Run As->Maven Build

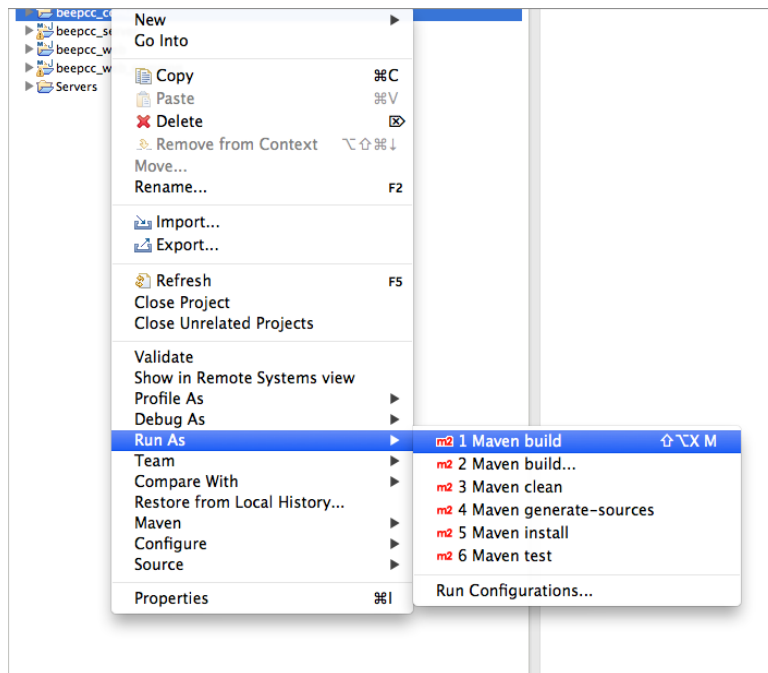


Figura 5.5: Maven build

Una vez realizado este paso ya estaría disponible nuestro proyecto para continuar el desarrollo.

Estos servicios web utilizan una base de datos que en este caso he utilizado Mysql debido a que es el gestor de base de datos más utilizado a lo largo de la carrera.

Este gestor se puede descargar de manera gratuita desde la pagina oficial <http://dev.mysql.com/downloads/mysql/>.

Para facilitar la gestión de nuestra base de datos recomiendo descargar Mysql Workbench que es una herramienta gráfica que permite interactuar con nuestra base de datos de manera sencilla evitando el uso de la consola.

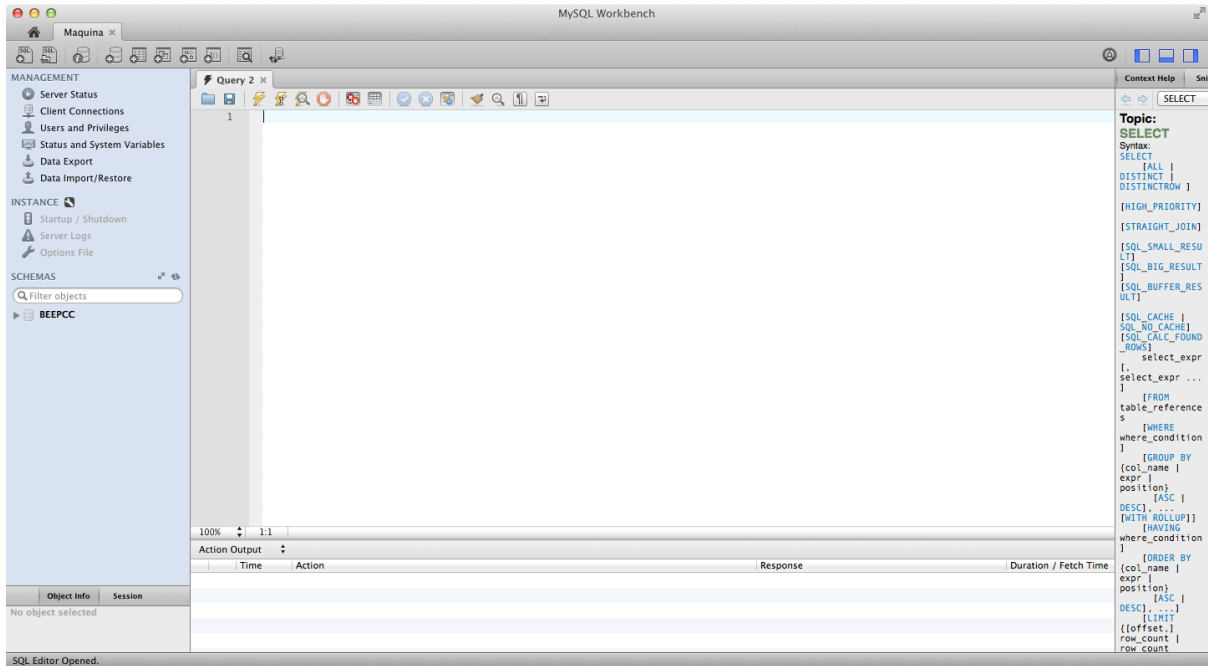


Figura 5.6: WorkBench

Una vez finalizado este paso ya tendríamos todas las herramientas para poder desarrollar nuestro proyecto a falta de explicar la instalación de nuestro contenedor web que se explicará al finalizar nuestro desarrollo.

5.1.2 Desarrollo de las funcionalidades principales

Registro de un usuario

El registro de un usuario, es una funcionalidad que nos permite registrar a un usuario en nuestro sistema. Para poder utilizar todas las funcionalidades de nuestro proyecto es necesario que el usuario esté registrado.

Para el registro está definida una función en el dispositivo móvil que le permite al usuario introducir sus datos. Estos datos son enviados al servidor para que los almacene y le devuelva un identificador al usuario.

Una vez que se han almacenado el usuario en nuestro sistema, el dispositivo realiza una petición para que le devuelva el servidor todas aquellas universidades que estén almacenadas en nuestra base de datos para que el usuario pueda elegir en que universidad se encuentra.

Una vez que el usuario a seleccionado una universidad el dispositivo realiza una petición indicando cual es la universidad seleccionada para que el servidor le devuelva todos aquellos grados que están registrados para esa universidad.

Una vez seleccionado el grado, el dispositivo realiza una petición al servidor para que le devuelva todas las asignaturas de ese grado. Una vez devueltos los almacena en la base de datos del dispositivo para permitir al usuario suscribirse en cualquier canal relacionado con dicha asignatura.

Se puede observar todo el desarrollo en la siguiente figura.

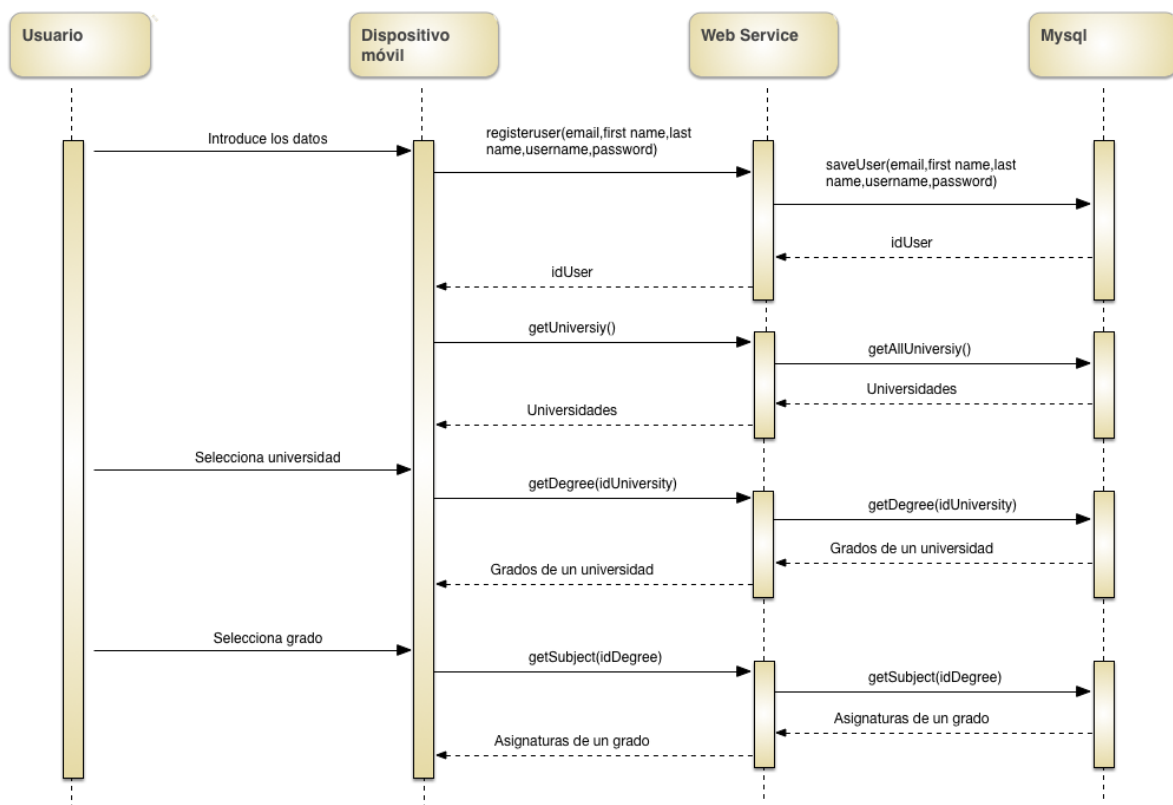


Figura 5.7: Diagrama de secuencia registro de un usuario

Subscribirse a un canal existente

Subscribirse a un canal existente, es una funcionalidad que le permite a un usuario suscribirse a canales relacionados con una asignatura, un grado, una universidad...

Para realizar este proceso es necesario que el dispositivo tenga almacenado el identificador del canal para poderlo transmitir al servidor y este registrarlo.

Algunos canales tienen cierta jerarquía, es decir, que son subcanales de algún canal padre por ejemplo en el ámbito académico un canal de la asignatura de Desarrollo de Programas es un subcanal del canal de Grado en Ingeniería de Software y a su vez de la Universidad de Extremadura. Hay que destacar que si un usuario se suscribe en un canal hijo se debe registrar en todos los canales padres.

Para este proceso cuando un usuario selecciona un canal que desea registrarse, el dispositivo envía el identificador del canal. El servidor cuando obtiene esta petición le suscribe y comprueba si ese canal existe y si es dependiente de algún canal padre.

En el caso de que el canal depende de un canal padre este le devuelve al dispositivo el canal del padre y automáticamente este se registra en el nuevo canal y así sucesivamente hasta que no dependan de más padres.

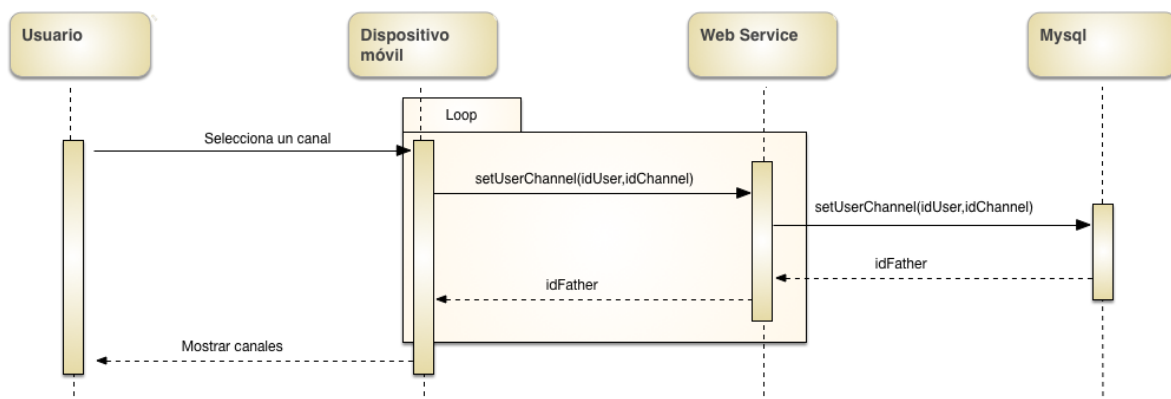


Figura 5.8: Diagrama de secuencia suscribirse a un canal existente

Crear Canal Temático

Crear un canal temático, es una funcionalidad que permite al usuario crear un canal con todos aquellos usuarios que tengas ciertos conocimientos sobre un tema concreto. Este canal se crea totalmente anónimo, es decir, un usuario no sabe con quien se está comunicando para evitar conflictos.

El objetivo de esta funcionalidad es proporcionar a los usuarios un nuevo servicio que les permita obtener ciertos conocimientos de otros usuarios con el beneficio de ser una comunicación en tiempo real.

El usuario cuando desea crear un canal con estas características le indica al dispositivo móvil el tema deseado para crear el canal y el dispositivo móvil se lo indica al servidor, que se encarga de buscar los usuarios más recomendados para dicho tema y crea el canal automáticamente.

El proceso que se lleva a cabo es el siguiente.

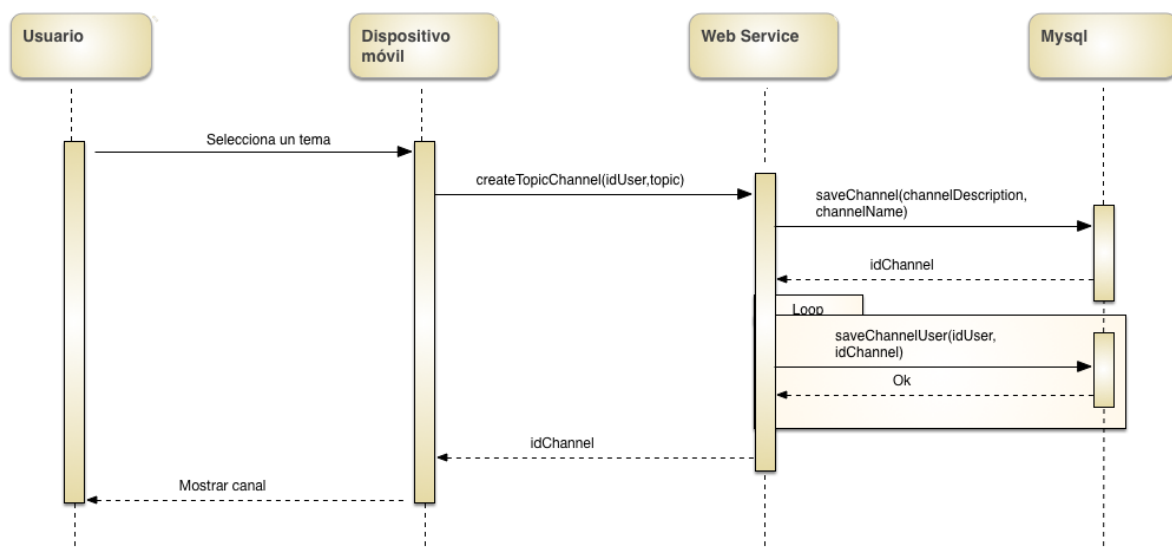


Figura 5.9: Diagrama de secuencia crear canal temático

Para entender mejor el proceso que se lleva a cabo en el servidor es necesario mostrar la siguiente figura.

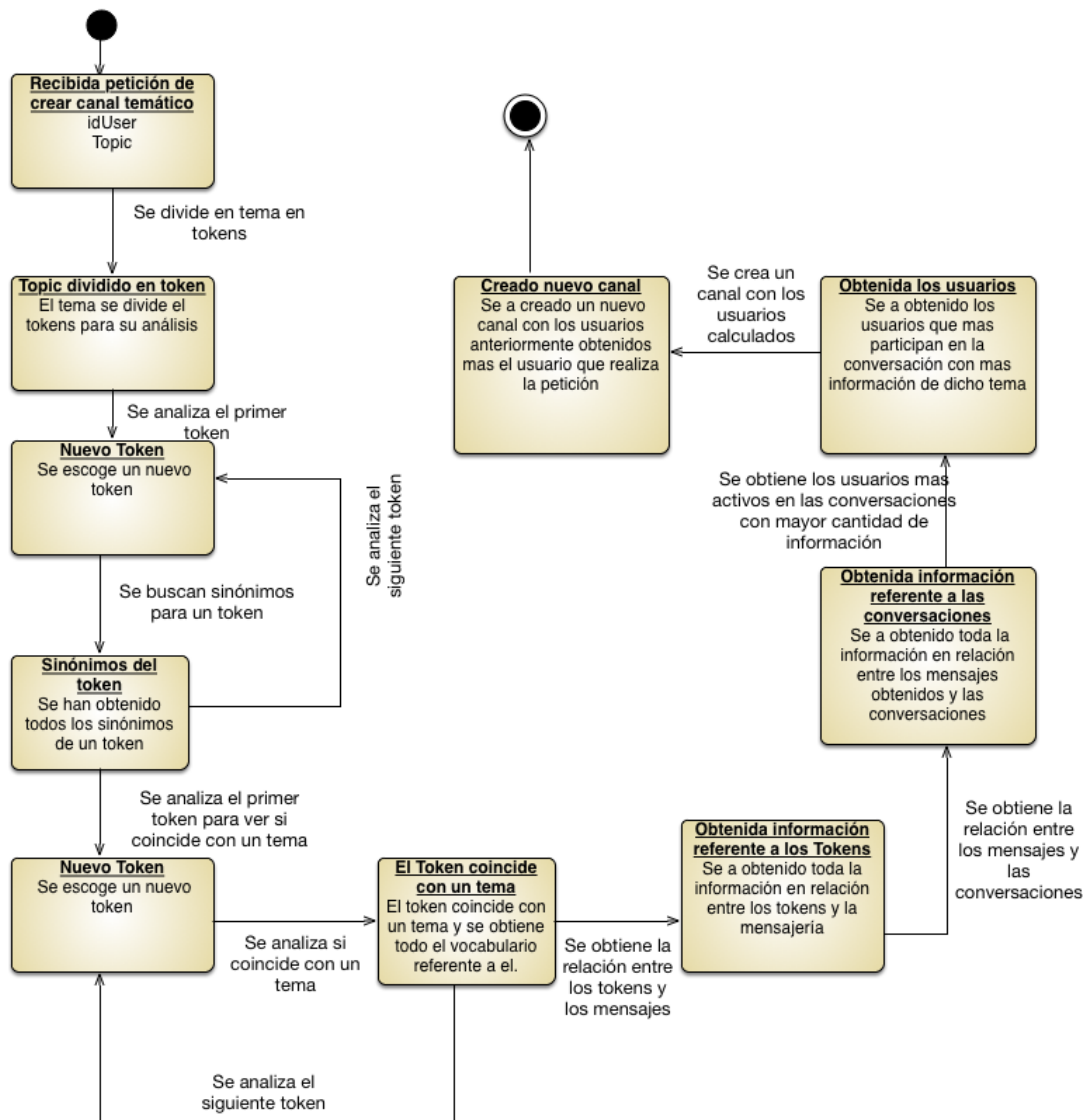


Figura 5.10: Diagrama de estado crear canal temático

Como se muestra en la figura anterior cuando un usuario desde la aplicación móvil desea crear un canal, con todos aquellos usuarios que tengan ciertos conocimientos de un tema concreto, este le realiza una petición al servidor con el identificador del usuario y un tema concreto. El servidor será el encargado de realizar esta búsqueda para ahorrar trabajo a los dispositivos móviles debido a que actualmente hay muchas limitaciones en las tarifas de datos de los dispositivos móviles y encargar de esta función al dispositivo saturaría la red debido a la gran cantidad de peticiones que deberían hacerse a todos los usuarios.

Cuando la petición llega al servidor este analiza el tema suministrado. En primer lugar divide el tema en diferentes token, que es una cadena de caracteres que tienen cierto significado coherente orientado al vocabulario español.

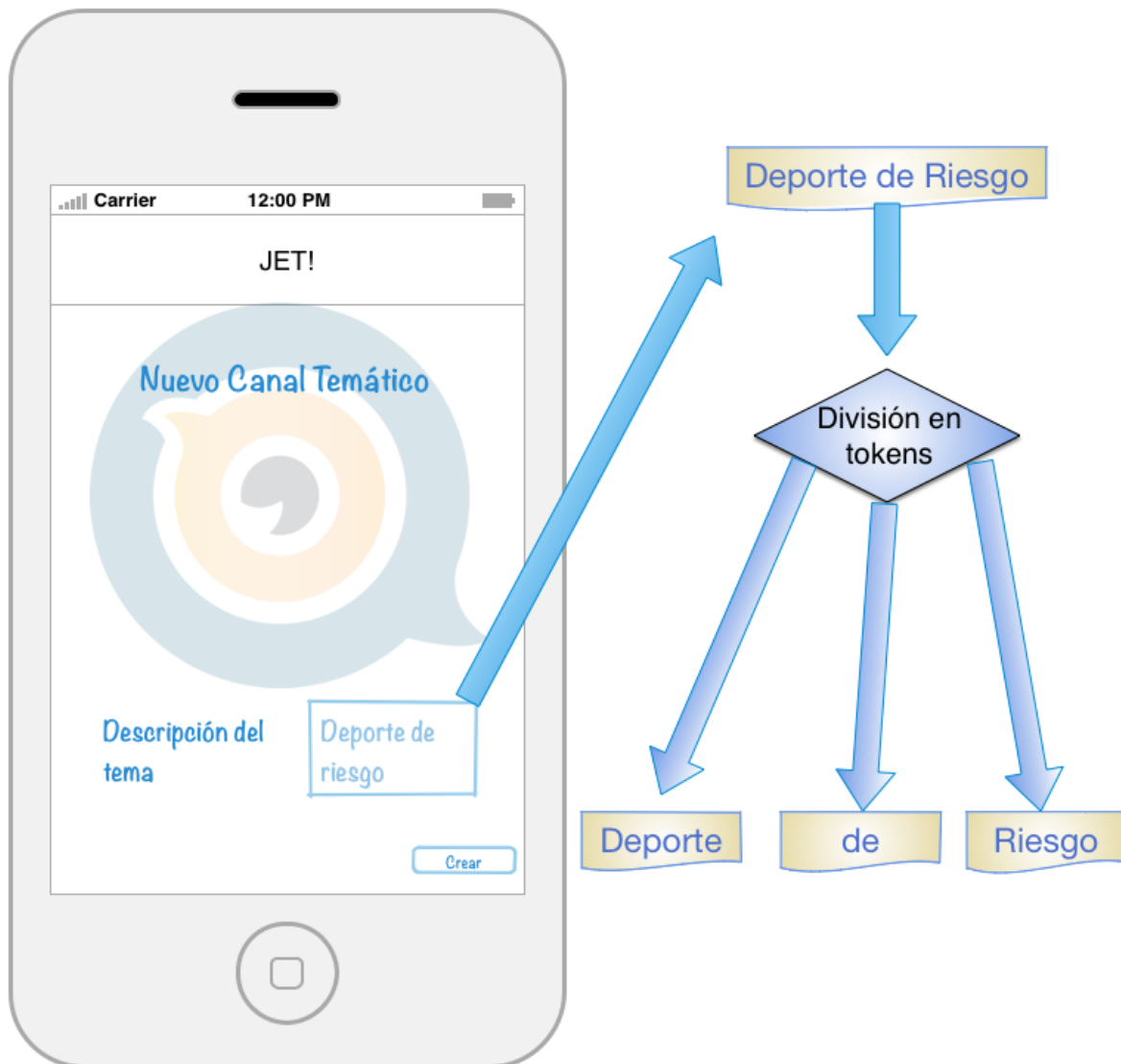


Figura 5.11: Representación división en token

Una vez dividido el tema en tokens el siguiente paso a llevar a cabo es comprobar si los tokens tienen significado propio y para ello tenemos definido un diccionario con la mayoría de las palabras en español exceptuando todo tipo de pronombres, preposiciones... que no dan ningún significado temático.



Figura 5.12: Comprobación diccionario

Como se puede comprobar los token “Deporte” y “Riesgo” si se encuentran en nuestro vocabulario, es decir, que por si solo tiene significado, en cambio el determinante “de” no cumple este requisito por lo que es desechado.

El siguiente paso es comprobar si coincide los tokens resultantes con un tema genérico de los que tenemos almacenados. En el caso de que se cumpla esta coincidencia desglosamos el tema en el vocabulario relacionado, permitiendo una búsqueda más concreta para ofrecer un servicio de mayor calidad al usuario.

Por ejemplo, el token Deporte puede ser el padre de una jerarquía en la que sus hijos son los diversos token que en su conjunto describen dicho tema. Algunos de estos token pueden ser Balonmano, Futbol, Gimnasia...

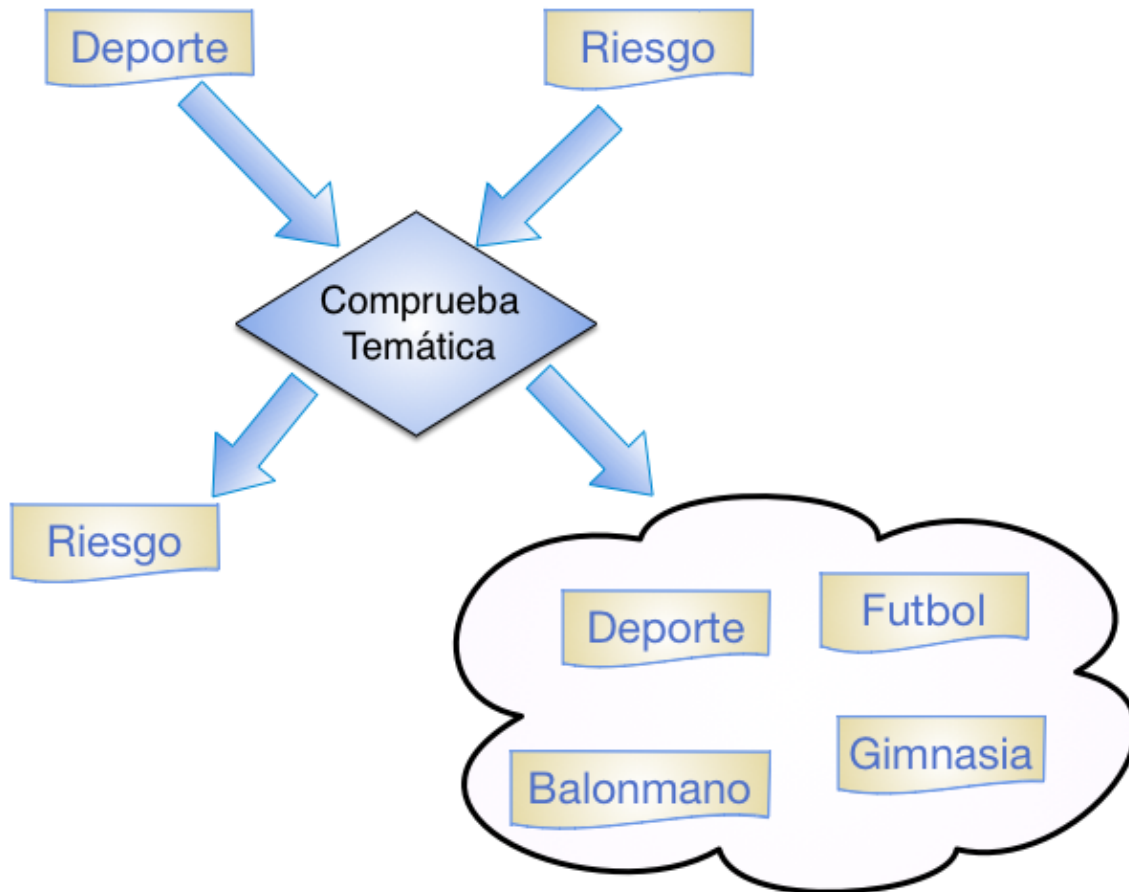


Figura 5.13: Comprobación temática

Una vez desglosados los posibles tokens temáticos pasamos a buscar si estos tokens tienen algún tipo de sinónimo. Esta búsqueda es necesaria debido a que no todos los usuarios hablan de la misma manera y un usuario puede hacer referencia a un token que tiene el mismo significado que otro token, pero totalmente diferentes en su escritura. Si no realizamos esta búsqueda un usuario que sea muy experto en un tema puede no compartir sus conocimientos simplemente porque escribe un concepto de diferente manera que el resto.

Con esta función podemos realizar una búsqueda no por el tema en sí, sino por el significado que tiene ese tema. Esto garantiza que da igual como el usuario describa su tema, se encontrará la mejor solución para que el usuario tenga acceso a la mejor información.

Por ejemplo, la palabra Riesgo puede tener sinónimos como Peligroso, Aventura...

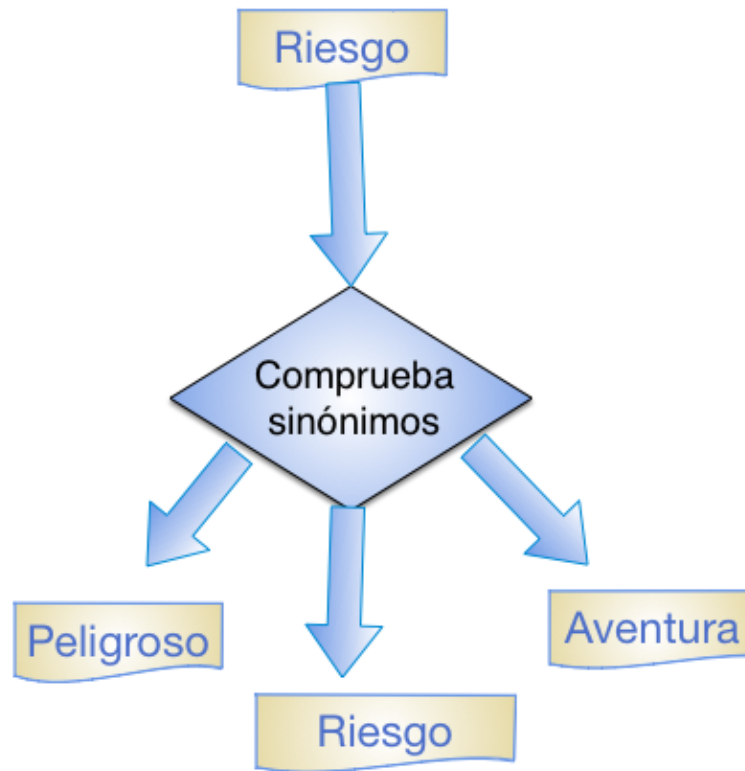


Figura 5.14: Comprobación sinónimos

Una vez encontrado todos los posibles sinónimos, es cuando empezamos a buscar que usuarios son los que más conocimientos tienen de este tema. La primera opción es analizar todos los mensajes, pero esto sería inmanejable debido a la gran cantidad que habría almacenado en la base de datos.

Para esto surge la necesidad de abstraer un poco la solución y en vez de analizar la temática de cada mensaje analizaremos la temática de cada conversación permitiéndonos averiguar en que conversaciones se habla más de dicho tema. Estas conversaciones son analizadas mientras los usuarios se comunican a través de la conversación, es decir, cuando se desea crear un canal temático y necesitamos analizar las conversaciones, ya tenemos toda la información necesaria para evitar un nuevo análisis.

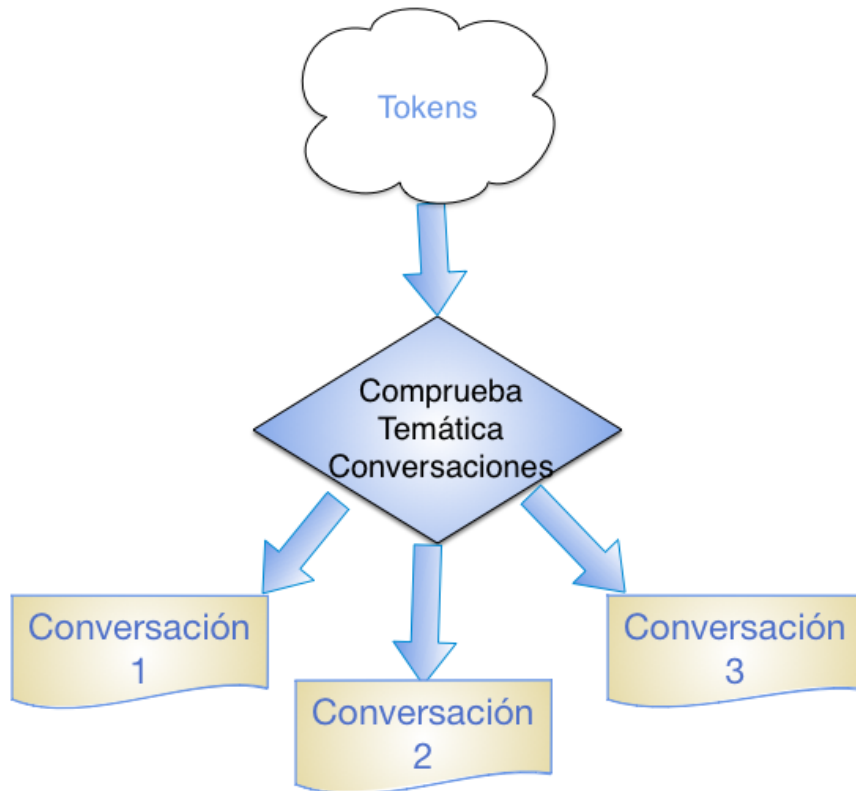


Figura 5.15: Comprobación conversaciones

Una vez obtenido todas las conversaciones que más hablan de dicho tema se pasa a obtener quienes son los usuarios que más se comunican en dichas conversaciones debido a que en la mayoría de las conversaciones hay un pequeño grupo de usuarios que son los que más escriben y suponemos que son los que más información tienen.

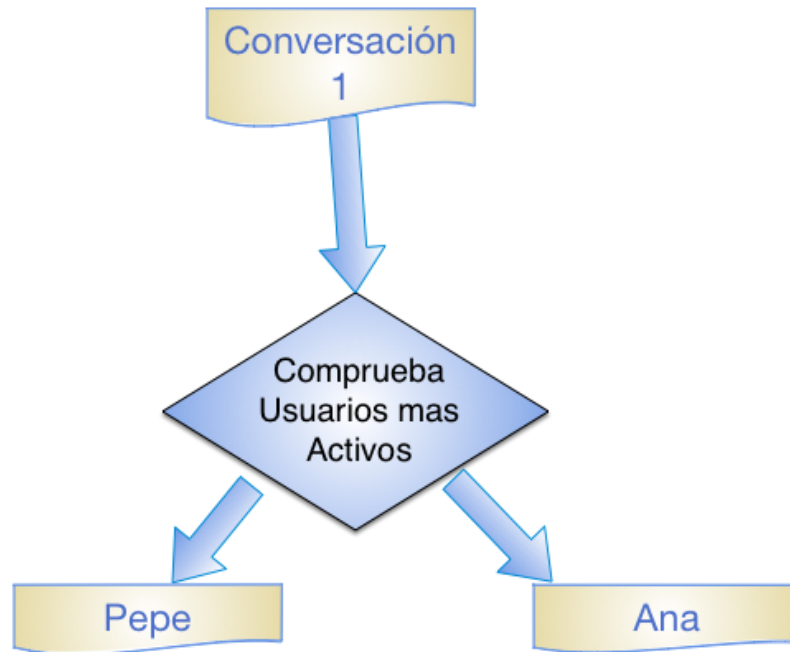


Figura 5.16: Comprobación usuarios

Una vez recuperado un conjunto de usuarios, el análisis de los mensajes se convierte en un proceso más eficiente debido a que tenemos una búsqueda más concreta con información de la conversación y del usuario.

Una vez finalizado el análisis de la información transmitida por dichos usuarios en las conversaciones anteriormente calculadas se procede a seleccionar los usuarios definitivos.



Figura 5.17: Ranking Usuarios

Crear Canal Localizado

Crear un canal localizado, es una funcionalidad que permite crear un nuevo canal con aquellos usuarios que cumplan un requisito orientado a ciertas posiciones localizadas. Estos requisitos pueden ser aquellos usuarios que vivan cerca del hogar de un usuario, o el lugar donde almuerza, o el lugar donde trabaja o directamente cerca de un punto concreto. El usuario es el encargado de seleccionar que tipo de canal quiere crear y se lo comunica al servidor que será el encargado de analizar esta información junto con la información del propio usuario, en el caso de tener que crear un canal por cercanía, y buscar la relación con el resto de usuarios.

Una vez analizada esta información y encontrado los usuarios que cumplen este requisito se crea el nuevo canal donde posteriormente cada usuario podrá crear diferentes conversaciones dentro del propio canal.

El objetivo de esta función es poderse comunicar con aquellos usuarios que puedan ofrecer cierta información referente a un posición dada.

Tenemos una opción que es crear un canal cumpliendo un tipo (Cerca de mi hogar,Cerca de mi trabajo,Cerca de mi lugar de almuerzo).

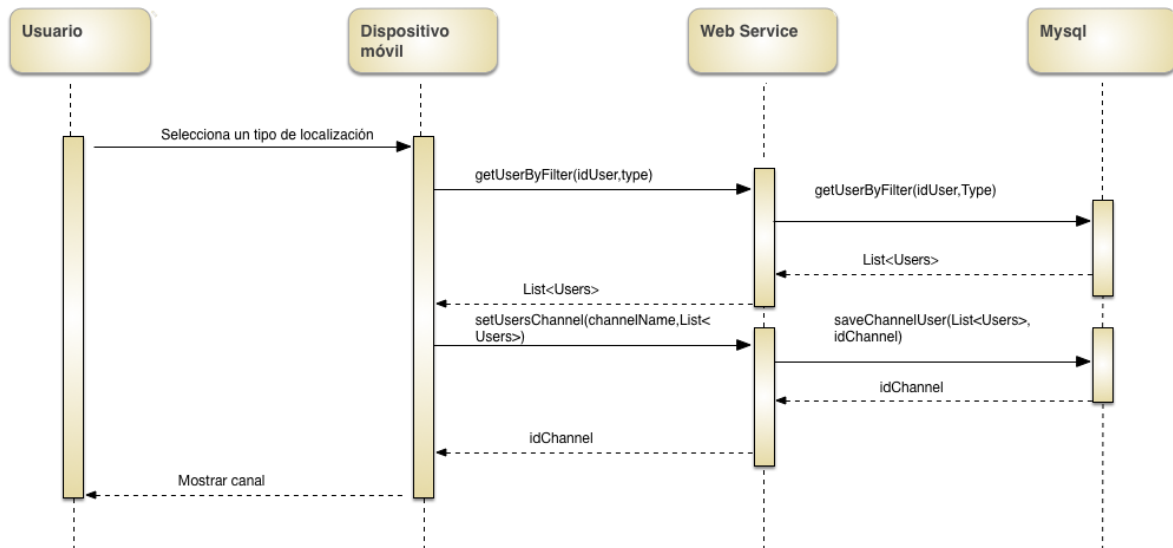


Figura 5.18: Diagrama de secuencia Crear canal localizado por coordenadas

Y otra opción es crear un canal cumpliendo la cercanía de unas coordenadas concretas.

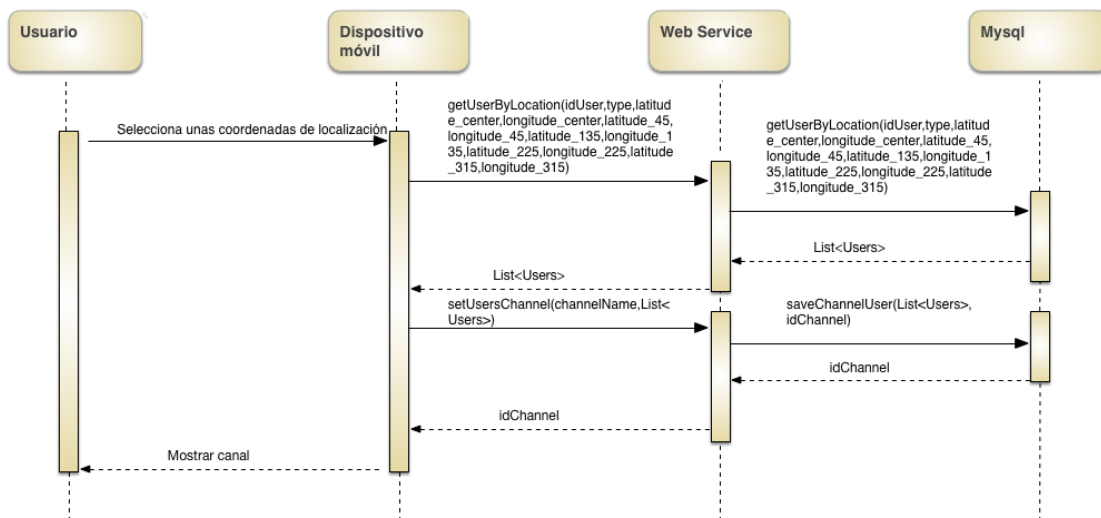


Figura 5.19: Diagrama de secuencia Crear canal localizado por tipo

Crear Conversación

Crear conversación, es una funcionalidad que le permite al usuario crear una nueva conversación dentro de un canal concreto. Estas conversación se utilizan para comunicarse entre los diferentes usuarios de un canal. Cuando un usuario crea una nueva conversación esta le aparece a todos los participantes del canal para que posteriormente puedan comunicarse a través de mensajes dentro de la conversación.

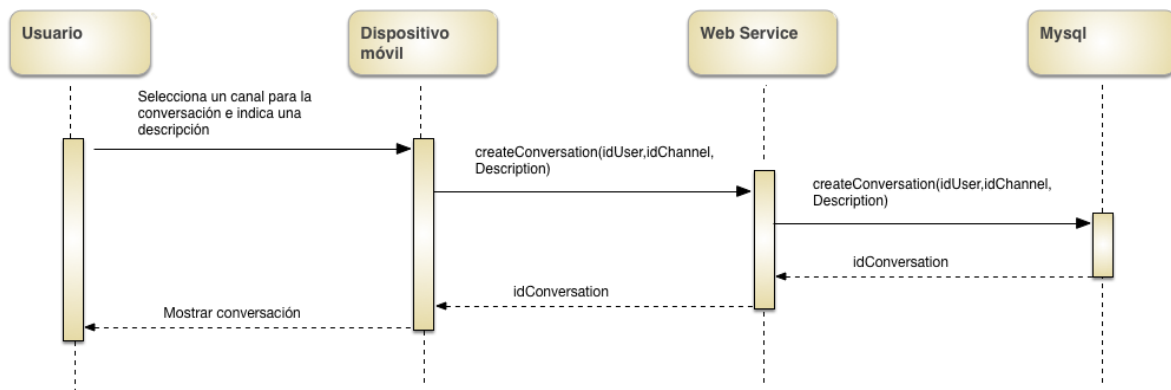


Figura 5.20: Diagrama de secuencia Crear conversación

Enviar Mensaje

Enviar mensaje, es una funcionalidad que le permite al usuario transmitir información a todos los participantes de una conversación. Estos mensajes son enviados al servidor para que este se ponga en contacto con la plataforma Nimbees y pueda ser transmitido a todos los usuarios registrados. Hay que destacar que el dispositivo móvil es el encargado de analizar los mensajes entrantes y comprobar si le pertenece o debe desecharlo. Estos mensajes al pasar por la parte servidora son analizados para todos los servicios que necesitan un análisis temático de dichos mensajes.

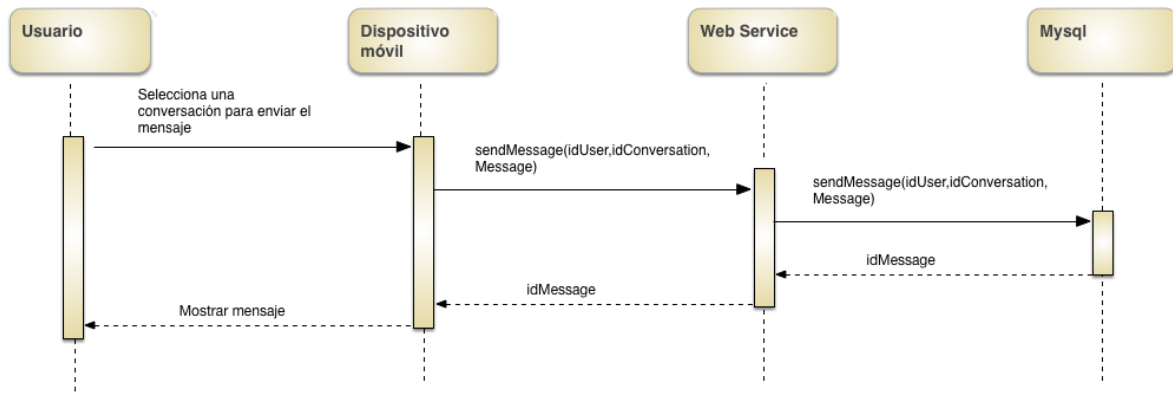


Figura 5.21: Diagrama de secuencia Enviar mensaje

En la siguiente figura se muestra con mayor detalle el proceso que se lleva a cabo en el servidor cuando llega un mensaje nuevo para ser enviado.

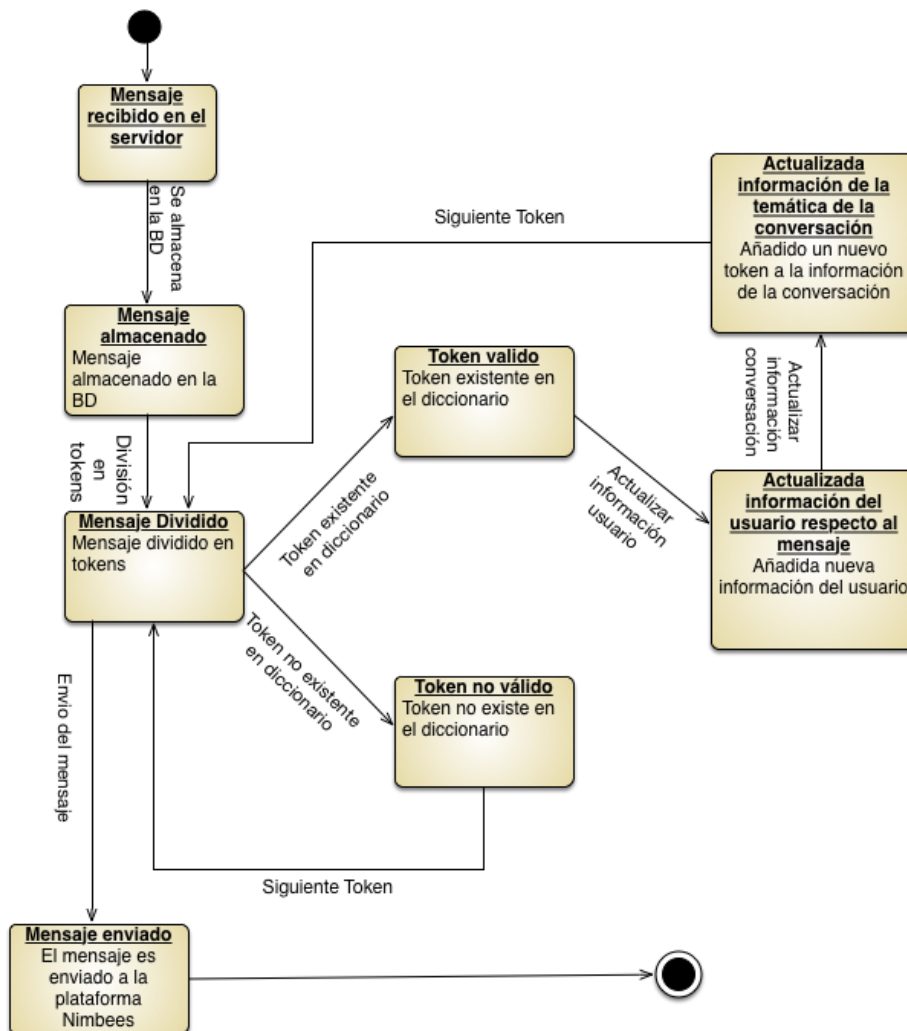


Figura 5.22: Diagrama de estado Enviar mensaje

Ver información (Usuario más influyente)

Esta funcionalidad permite al usuario ver quien es el que más mensajes le ha enviado. Este dato es puramente informativo con el objetivo de intentar hacer más entretenido el uso de la aplicación. Este dato en un futuro puede servir para realizar sugerencias de amistades orientadas al uso de la aplicación.

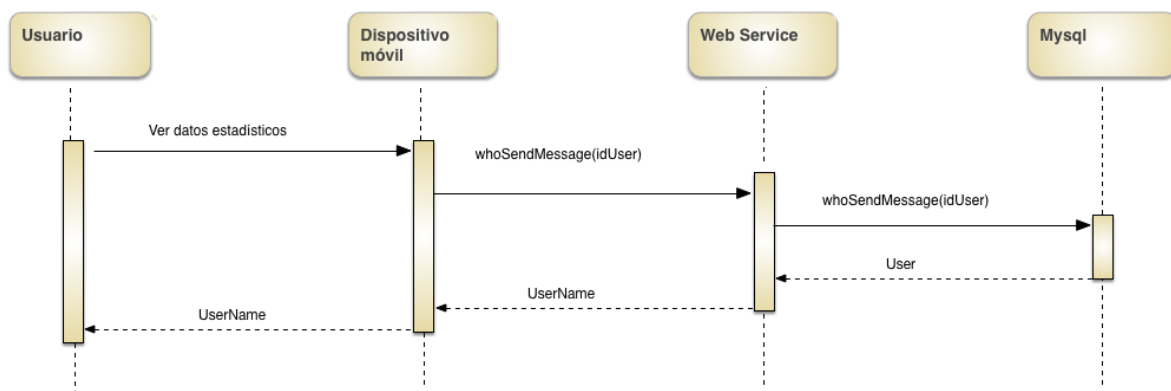


Figura 5.23: Diagrama de secuencia Ver información(Usuario más influyente)

Ver información (Usuario más influido)

Esta funcionalidad le permite al usuario ver quien es el usuario que más mensajes a recibido de su parte. Este dato es puramente informativo con el objetivo de intentar hacer más entretenido el uso de la aplicación. Este dato en un futuro puede servir para realizar sugerencias de amistades orientadas al uso de la aplicación.

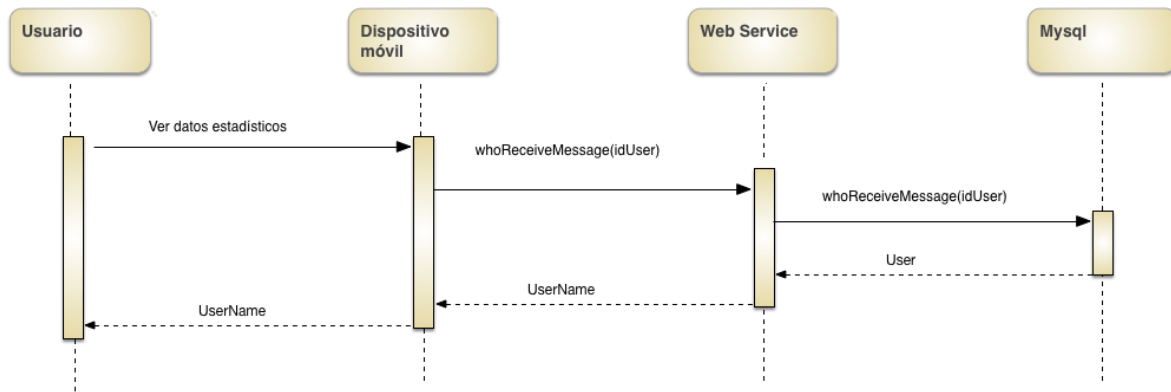


Figura 5.24: Diagrama de secuencia Ver información(Usuario más influido)

Ver información(Intervalo de tiempo que más interactúa)

Esta funcionalidad le permite al usuario saber en que intervalo de tiempo envía más mensajes a través de la aplicación. Este dato es puramente informativo con el objetivo de intentar hacer más entretenido el uso de la aplicación. Este dato en un futuro puede servir para realizar sugerencias de amistades orientadas al uso de la aplicación.

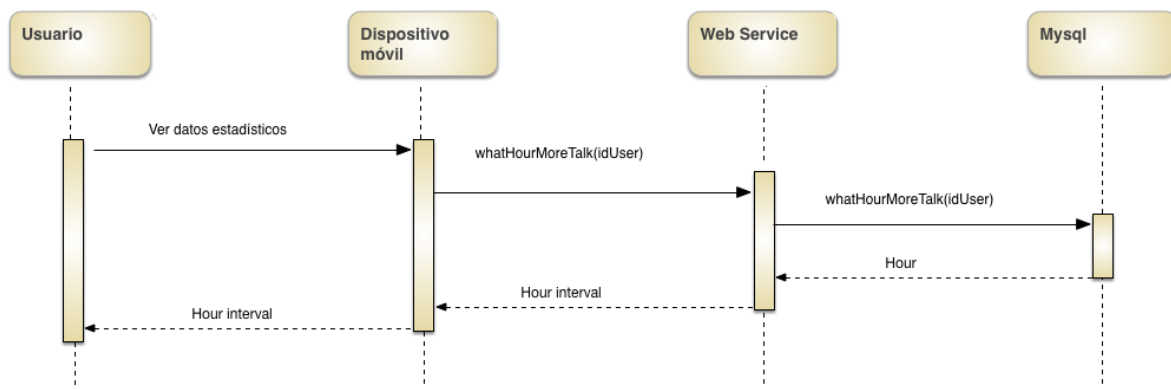


Figura 5.25: Diagrama de secuencia Ver información(Intervalo de tiempo que más interactúa)

Ver información (Hora que se duerme)

Esta funcionalidad le permite al usuario saber cual es el intervalo de hora que suele dejar de utilizar la aplicación, es decir, una aproximación de cuando el usuario suele dormirse.

Este dato es puramente informativo con el objetivo de intentar hacer más entretenido el uso de la aplicación. Este dato en un futuro puede servir para realizar sugerencias de amistades orientadas al uso de la aplicación.

El proceso que lleva a cabo es un análisis del registro de los mensajes enviados por dicho usuario.

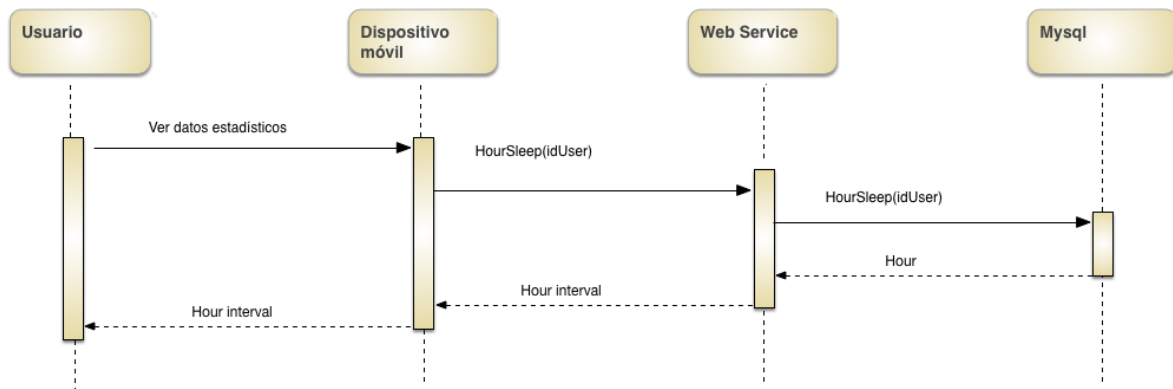


Figura 5.26: Diagrama de secuencia Ver información (Hora que se duerme)

Ver información (Hora que se despierta)

Esta funcionalidad le permite al usuario saber cual es el intervalo de hora que suele empezar a utilizar la aplicación, es decir, una aproximación de cuando el usuario suele despertarse.

Este dato es puramente informativo con el objetivo de intentar hacer más entretenido el uso de la aplicación. Este dato en un futuro puede servir para realizar sugerencias de amistades orientadas al uso de la aplicación.

El proceso que lleva a cabo es un análisis del registro de los mensajes enviados por dicho usuario.

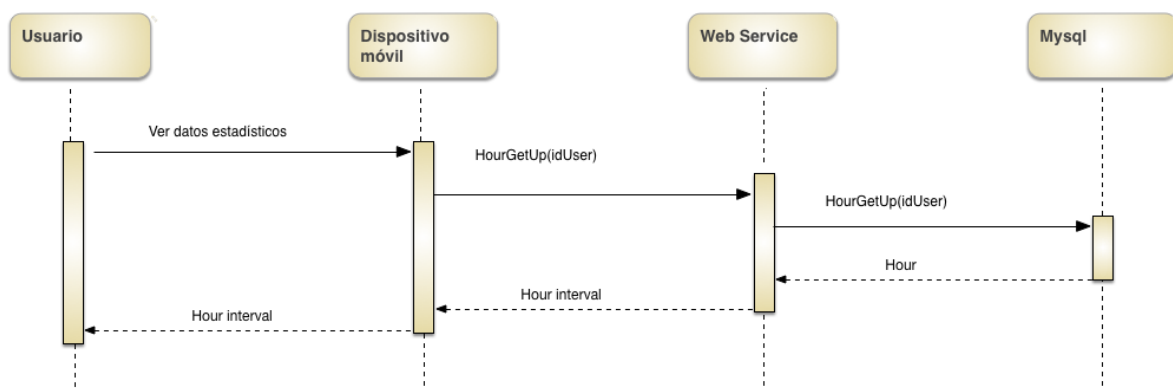


Figura 5.27: Diagrama de secuencia Ver información(Hora que se despierta)

Ver información (Tema más hablado por el usuario)

Esta funcionalidad le permite al usuario saber cual es el tema que más habla por la aplicación, es decir, cual es el temas de mayor interés para dicho usuario. Este dato es puramente informativo con el objetivo de intentar hacer más entretenido el uso de la aplicación. Este dato en un futuro puede servir para realizar sugerencias de amistades orientadas al uso de la aplicación.

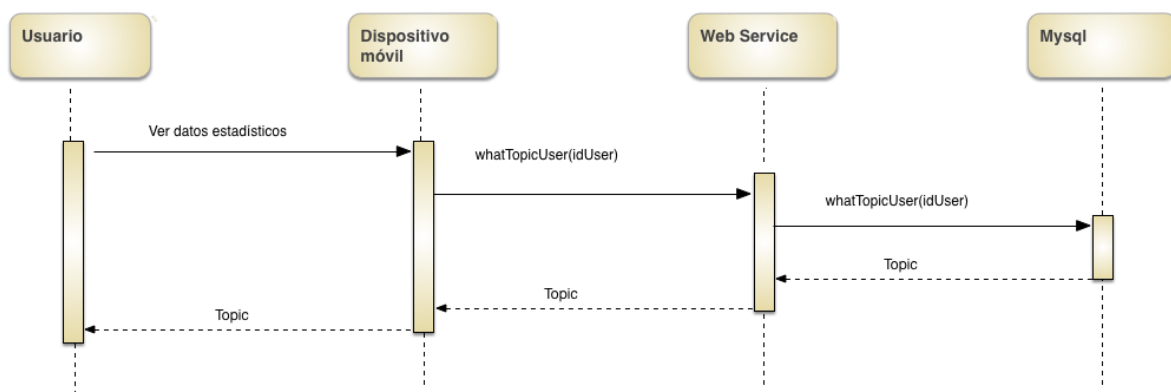


Figura 5.28: Diagrama de secuencia Ver información(Tema más hablado por el usuario)

5.1.3 Despliegue

Una vez finalizado el desarrollo de los servicios web es necesarios proceder al despliegue de nuestra aplicación.

Para desplegar estos servicios web se utilizará un servidor alojado en la Politécnica de Caceres con sistema operativo Ubuntu 12.04 con la intención de estar disponible 24h y con una máquina que soporte la carga de trabajo que necesita una aplicación de estas características en la cual se enviaran mensajes constantemente y a lo largo del día.

Este despliegue se realizará en un contenedor como es Tomcat eligiendo una versión reciente como es la 7.

En este servidor también se alojará la base de datos debido a que es la máquina con mayor potencia que tenemos en comparación con la Raspberry Pi que contendrá la aplicación web para el administrador.

Para una configuración inicial es necesario tener instalado Mysql y Tomcat. Hay que destacar que es necesario descargarnos el conector entre java y mysql para poder tener una interfaz entre estas dos tecnologías. Este conector se puede descargar gratuitamente desde la página oficial de mysql <http://dev.mysql.com/downloads/connector/j/3.0.html>.

Una vez descargado es necesario que lo incluyamos en la carpeta “Directorio Tomcat/lib/”. Para desplegar cualquier aplicación web es necesario exportar el proyecto en formato .war, que será el archivo que debemos incluir a Tomcat en la carpeta “webapps” dentro del directorio de Tomcat para que este lo pueda desplegar automáticamente cuando arranquemos dicho contenedor.

Una vez realizado todos estos pasos para la configuración inicial para arrancarlo es necesario ejecutar el “startup.sh” que contiene todo el procedimiento para poder arrancar nuestro contenedor.

Al tener una comunicación segura en base a un certificado es necesario modificar el archivo de configuración para poder atender a las peticiones que provienen por el puerto seguro.

5.2 Aplicación Web

En este apartado se explicará los pasos necesarios para desarrollar nuestra aplicación web que será la encargada de proporcionar todos los servicios necesarios para que el administrador pueda gestionar los datos fundamentales de nuestra aplicación.

Esta aplicación web, como es lógico, comparte la misma base de datos que los servicios web para poder acceder a los datos totalmente actualizados en tiempo real.

Esta aplicación web para repartir la carga de trabajo y tener bien diferenciado la parte administradora de la parte de los usuarios se va a desplegar en un servidor diferente como es la raspberry.

Los pasos que se van a llevar a cabo para explicar este desarrollo es en primer lugar definir los requisitos para poder desarrollar la aplicación web, seguido se explicará como se a desarrollado y finalmente como se ha desplegado en el servidor.

5.2.1 Requisitos necesarios

Como bien se ha explicado en apartados anteriores la aplicación web se desarrollará con la herramienta de Webratio, que permite desarrollar paginas web a través de modelos.

Esta metodología de desarrollo disminuye considerablemente el tiempo de implementación cuando se necesita una página web donde la lógica de negocio sea simple, en este caso solo se encarga de administrar la base de datos de nuestra aplicación.

Para poder utilizar esta herramienta es necesario comprar una licencia o tener licencia de estudiando, como es en este caso, que permite varias funciones aunque no todas las que proporciona la licencia principal.

Una vez conseguida la licencia se puede descargar directamente la herramienta desde la página oficial <http://www.webratio.com/portal/content/en/pricing>

Para la aplicación web no es necesario que instalemos una base de datos debido a que esta herramienta utilizará la base de datos del servidor del Web Service.

Una vez instalada esta herramienta se puede observar la siguiente interfaz para poder desarrollar nuestro proyecto.

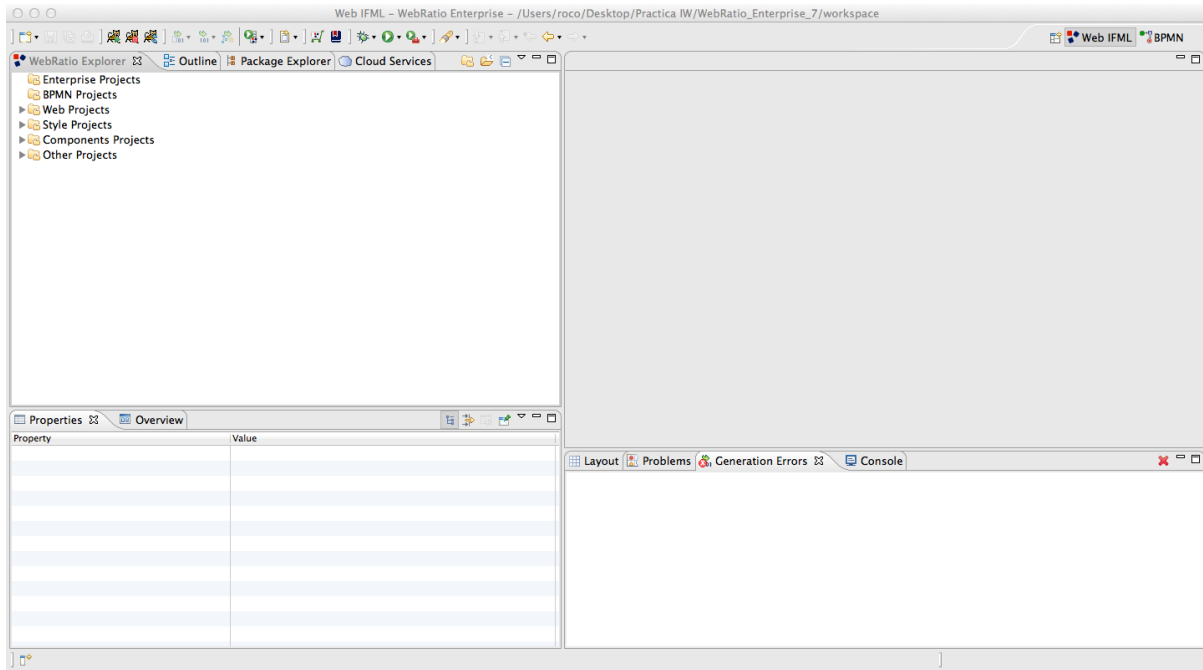


Figura 5.29: Interfaz WebRatio

Para importar nuestro proyecto basta con ir a File->Import->WebRatio Project e incluir el proyecto Admin adjunto a esta documentación.

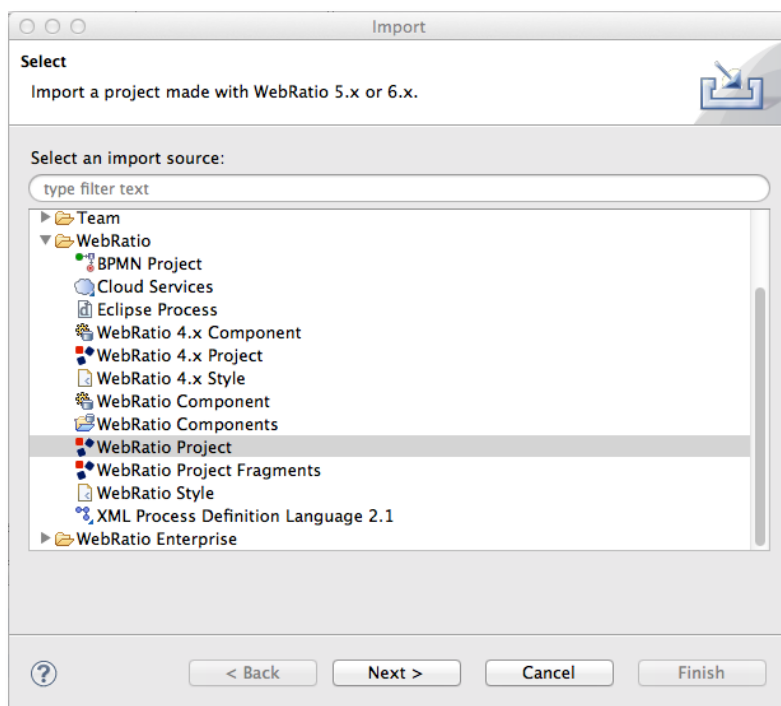


Figura 5.30: Import Project WebRatio

Este proyecto contiene un estilo propio de carácter simple que facilite al administrador encontrar de manera rápida y eficiente todas las funcionalidades que esta aplicación web proporciona.

Para incluir este estilo es necesario ir a File->Import-> WebRatio Style y seleccionar el estilo adjunto a esta documentación.

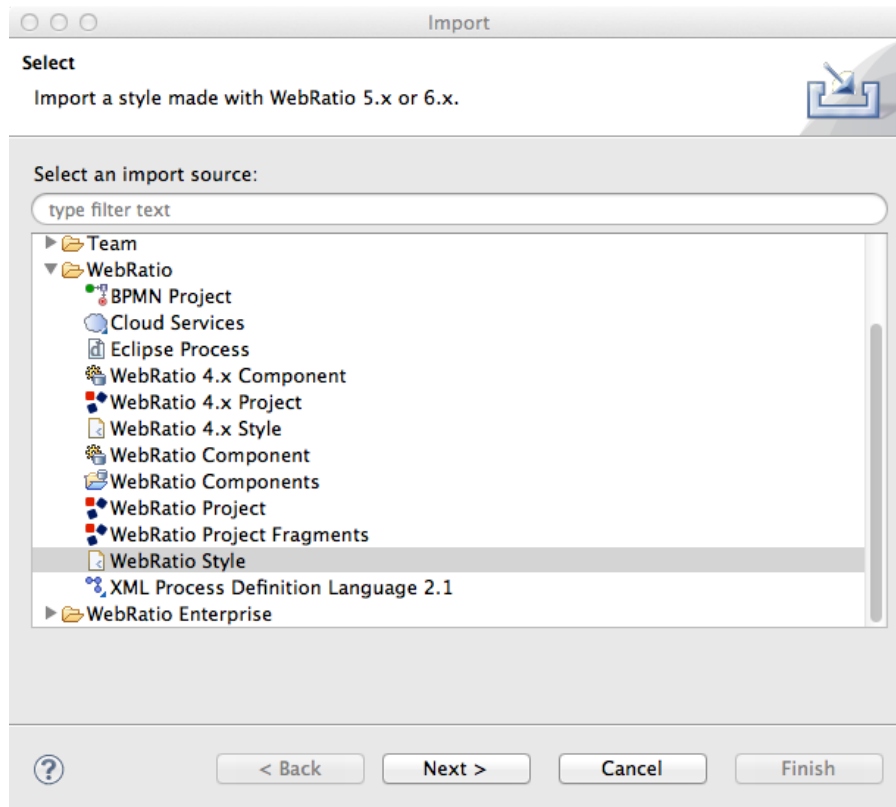


Figura 5.31: Import Style Webratio

Finalizado este proceso ya tendríamos nuestro proyecto disponible para poder desarrollar nuevas funcionalidades.

5.2.2 Desarrollo de las funcionalidades principales

Login

Esta funcionalidad permite al usuario loguearse en nuestra pagina web. Los datos de los usuarios deben insertarse directamente en la base de datos debido a que vamos a tener únicamente a los usuarios administradores.

La estructura de la base de datos es la siguiente.

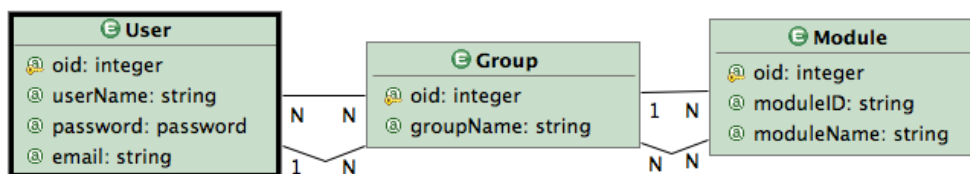


Figura 5.32: Modelo de datos Login

- **Module:** Esta tabla es la encargada de administrar las zonas privadas de nuestra aplicación por ejemplo en este caso únicamente vamos a tener una zona privada que es a la que accede el administrador pero si deseamos tenemos tener otra zona, por ejemplo para los usuarios registrados tendríamos diferenciada las dos zonas en dicha tabla.
- **Group:** Esta tabla se utiliza para diferenciar los distintos roles posibles para un usuario que tiene relación con las paginas accesible para dicho grupo.
- **User:** Esta tabla es donde se encuentran los usuarios registrados en esta aplicación web. Hay que destacar que en la aplicación el nombre referente es User pero en la base de datos es Admin para poder diferenciar los usuarios de la aplicación web y los usuarios de los dispositivos móviles.

Para poder desarrollar el login es necesario tener una pagina que sea publica para que cualquier usuario pueda acceder. Como la implementación con esta herramienta es con modelos en la siguiente figura se muestra el referente al login.

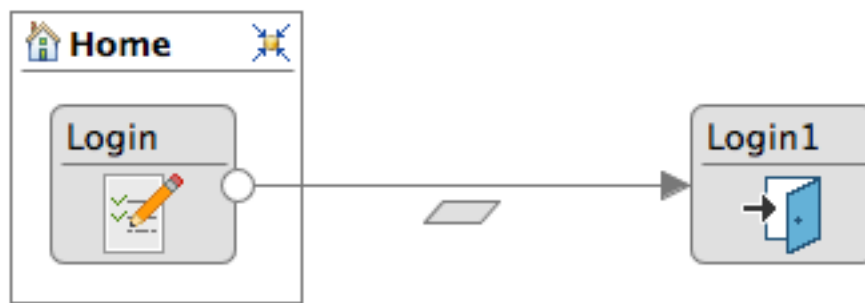


Figura 5.33: Modelo login

Insertar un nuevo tema

Esta función le permite al administrador crear un nuevo tema para mejorar nuestra búsqueda temática. Cuanto más concreto sea el tema, más concreta será la búsqueda y así proporcionar un mejor servicio al usuario. Recordamos que cuando un usuario busca por un token concreto y este es un tema utiliza todo el vocabulario de dicho tema para obtener mejores resultados. Esta información se almacena en una tabla compartida entre el Web Service y la aplicación web.

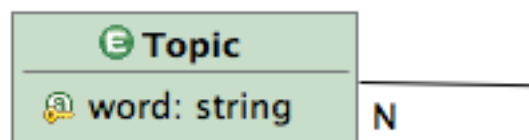


Figura 5.34: Modelo de datos Insertar un nuevo tema

Esta estructura controla que no exista temas con un mismo nombre pero puede existir vocabulario que pertenezca a varios temas.

En la pantalla para gestionar los temas nos muestra una lista completa con todos los temas almacenados en la base de datos y en la parte inferior un campo de texto que nos permitirá añadir el nuevo tema.

El modelo para este servicio es el siguiente.

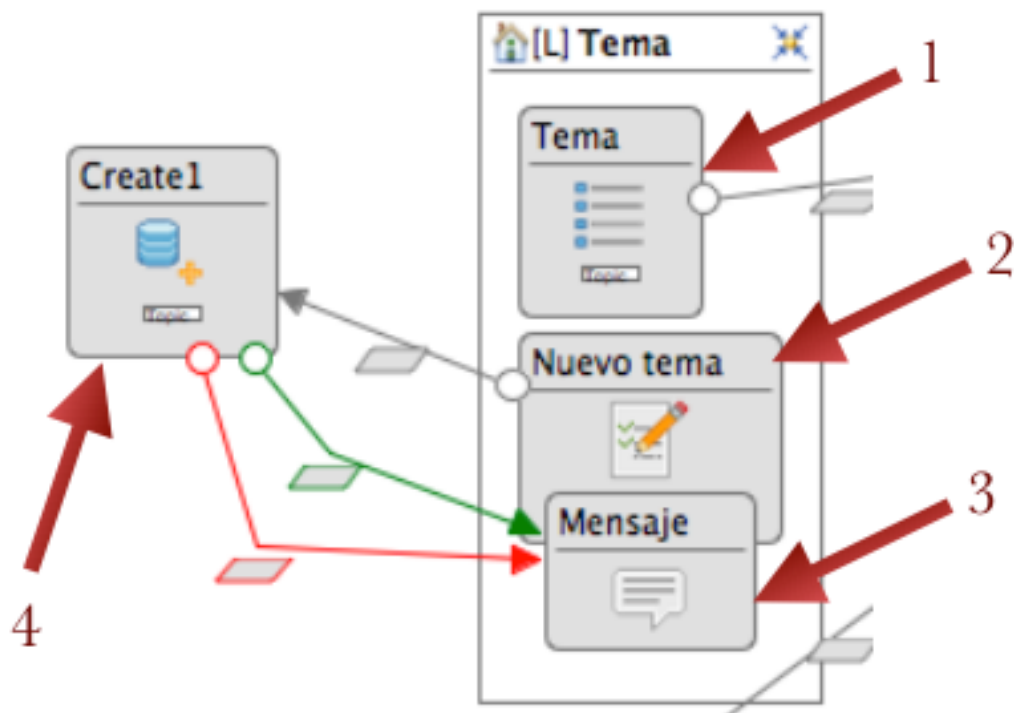


Figura 5.35: Modelo insertar un nuevo tema

1- Este componente se encarga de mostrar todos los temas almacenados en nuestra base de datos.

2-Este componente se encarga de mostrarnos un campo de texto que nos permita añadir un nuevo tema.

3-Este componente se encarga de mostrarnos un mensaje informativo sobre el procedimiento de almacenar un tema en nuestra base de datos.

4-Este componente se encarga de almacenar en la base de datos el tema introducido.

Insertar nuevo vocabulario

Esta función se encarga de proporcionar al administrador un servicio que le permite insertar nuevas palabras a un tema concreto. Cuanto más vocabulario tengamos en nuestra base de datos, más detallada será la búsqueda de los usuarios que hablen de un tema concreto.

Este trabajo suele estar orientado a un experto en el vocabulario español, debido a la gran cantidad de relaciones existentes en nuestro lenguaje.

El modelo para esta función es muy parecido al modelo de la función anteriormente definida.

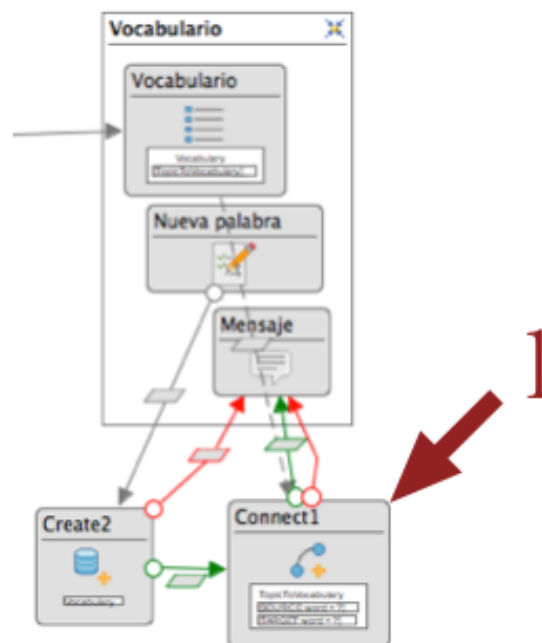


Figura 5.36: Modelo Insertar una nueva palabra

1- Este componente se encarga de crear una relación entre dos tablas por ejemplo en este caso para insertar a una nueva palabra al vocabulario necesita tener una relación con algún tema como mínimo y este componente es el encargado de crear dicha relación.

Crear una nueva Universidad

Esta función permite al administrador añadir una nueva universidad a nuestra base de datos.

Cuando un usuario se registra le aparece el listado de todas las universidades almacenadas y selecciona una de ellas, esta información puede varias o pueden aparecer nuevas universidades y esta función esta orientada a facilitar esta gestión.

El modelo para esta función es la siguiente.

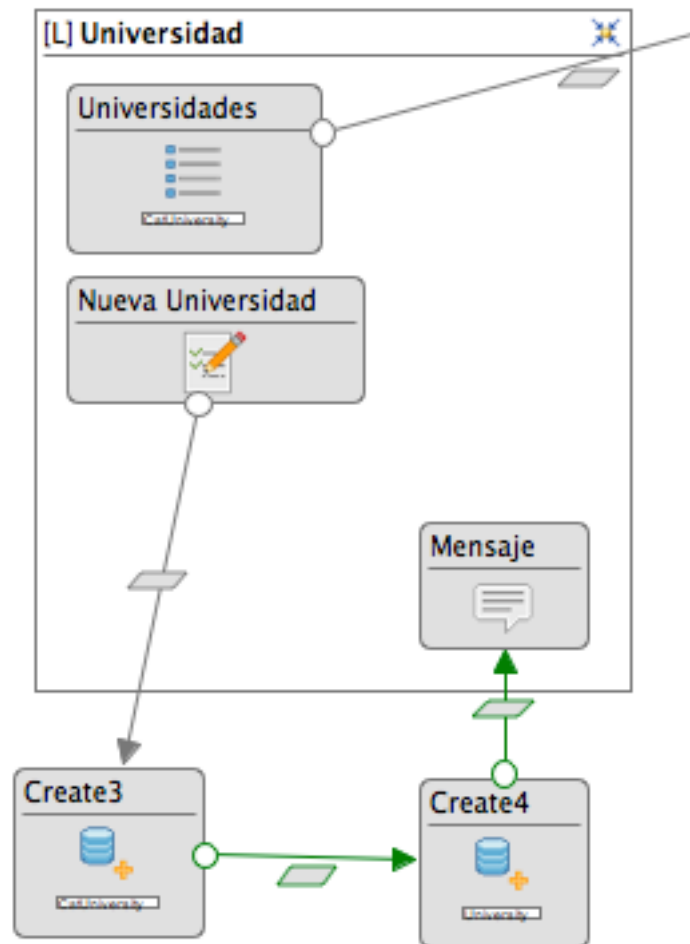


Figura 5.37: Modelo crear una nueva universidad

Crear un nuevo Grado

Esta función permite al administrador añadir un nuevo grado en nuestra base de datos en relación a una universidad dada. El modelo de esta función es el siguiente.

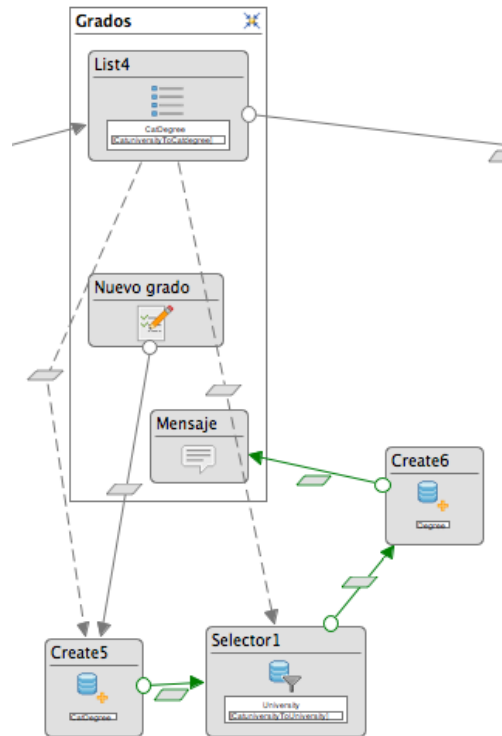


Figura 5.38: Modelo crear un nuevo grado.

Crear una nueva asignatura

Esta función permite al administrador añadir una nueva asignatura a un grado específico.

Toda asignatura esta compuesta por un nombre, un año y dos descripciones que permitirán al usuario dar la suficiente información para suscribirse a dicha asignatura.

El modelo de esta función es la siguiente.

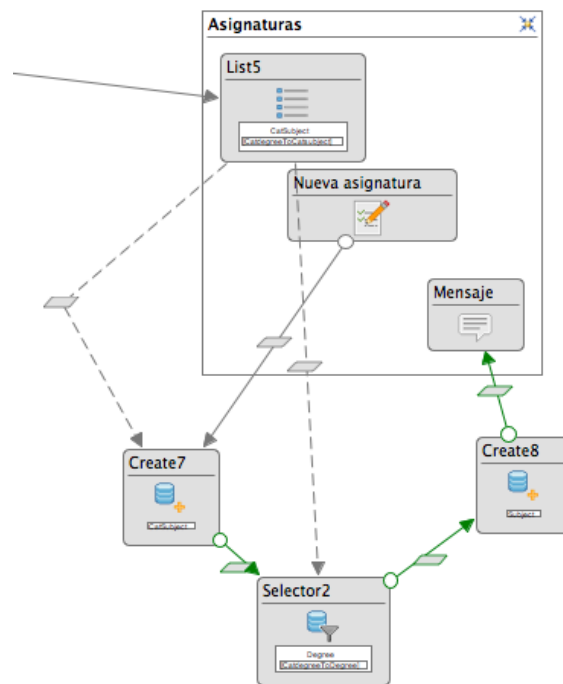


Figura 5.39: Modelo crear una nueva asignatura

Ver mapa estadístico (Home)

Esta función permite al administrador observar en que sectores del mapa donde viven los usuarios de esta aplicación. Estos datos pueden ser muy útiles para numerosas empresas que estén orientadas a ofrecer servicios a los estudiantes. Esta información se obtiene de la localización calculada en los dispositivos móviles.

El modelo de esta función es el siguiente.

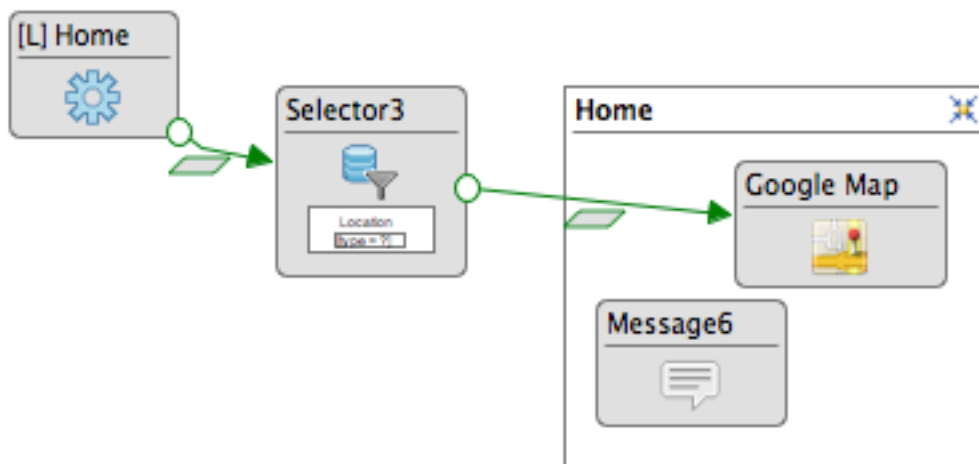


Figura 5.40: Modelo ver mapa Home
Ver mapa estadístico (Work)

Esta función permite al administrador observar en que sectores del mapa donde trabajan los usuarios de esta aplicación. Estos datos pueden ser muy útiles para numerosas empresas que estén orientadas a ofrecer servicios a los estudiantes. Esta información se obtiene de la localización calculada en los dispositivos móviles.

El modelo de esta función es el siguiente.

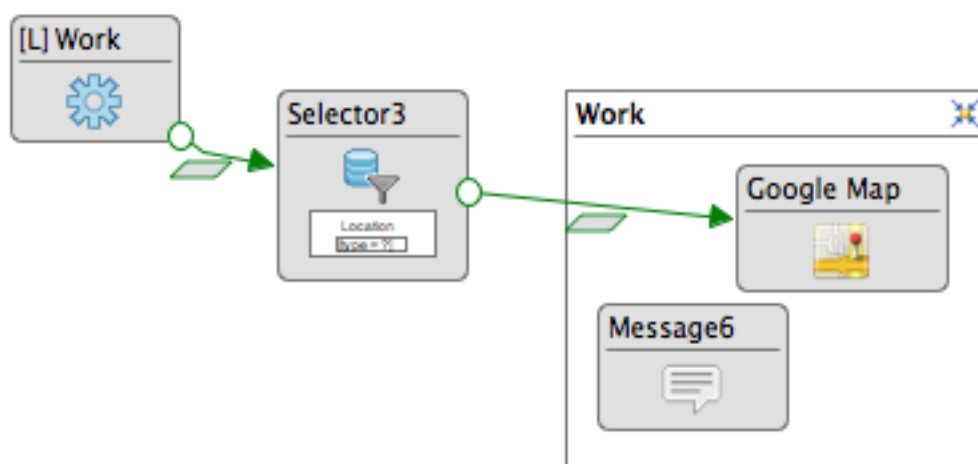


Figura 5.41: Modelo ver mapa Work

5.2.3 Despliegue

Es despliegue de la aplicación web es idéntico que el de los servicios web con la diferencia de que se encuentra en un servidor diferente y se debe configurar para que este pueda acceder a la base de datos de manera remota.

El objetivo de esta división, en diferentes servidores, es debido a la carga de trabajo que debe soportar el servidor del Web Service al tener que procesar numerosas peticiones de envío de mensajes. Si la aplicación web para la administración y el Webservice crecieran con muchas nuevas funcionalidades el servidor no podría ofrecer unos servicios eficientes.

6- Pruebas

En la fase de pruebas, llevada a cabo conjuntamente con el desarrollo, es una fase fundamental porque permite encontrar todos los errores posibles en un proyecto y conseguir una mayor calidad de nuestro software.

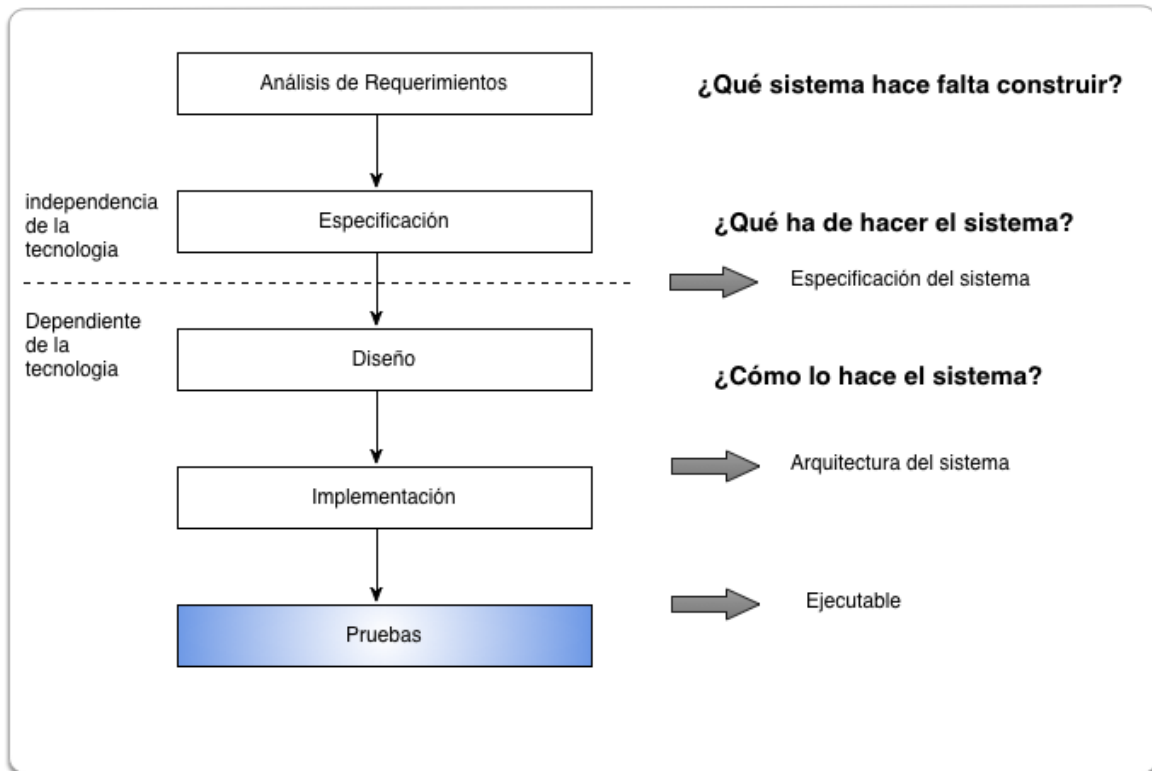


Figura 6: Fase pruebas

Para definir las pruebas es necesario diferenciar dos tipos:

- **Caja negra:** Son un tipo de pruebas que sirven para analizar un elemento en función de las entradas que tiene y las salidas que debe dar, sin tener en cuenta su funcionamiento interno.
- **Caja blanca:** Son un tipo de pruebas que se aplican sobre las funciones internas de un módulo.

Las pruebas llevadas a cabo son las siguientes:

- **Pruebas unitarias:** Son aquellas pruebas que se ejecutan por cada modulo y se pretende encontrar todos los errores posibles de dicho módulo.
- **Pruebas del sistema:** Son aquellas pruebas que permite encontrar todos los errores posibles en un caso de uso.
- **Pruebas de carga:** Son aquellas pruebas que permite comprobar si el servidor soporta numerosos usuarios consumiendo servicios a la vez.

- Pruebas de integridad de datos y base de datos: Permite asegurar que los métodos de gestión de la base de datos funcionan adecuadamente sin ocasionar corrupción de datos.
- Prueba de Seguridad y Control de acceso: Verifica que aquellos servicios privados solo puedan ser accedido por los usuarios registrados en la aplicación.
- Pruebas del ciclo de negocio: Asegura que el sistema funciona de acuerdo con el modelo de negocios, es decir, se ejecutan cada caso de uso llevando a cabo un flujo utilizando datos válidos y no válidos.
- Pruebas de funcionalidad: Se comprueba que cumplan todos los requisitos funcionales previstos en el análisis inicial del proyecto.
- Pruebas de campo: Se analiza el funcionamiento corriendo el sistema en un ambiente real.

Con la ejecución de todas estas pruebas hemos comprobado todos los posibles errores que pueda tener la aplicación, garantizando una calidad óptima.

7- Ampliaciones Y Mejoras

A continuación se procede a describir todas aquellas ampliaciones y mejoras que se pueden realizar sobre **JET!** convirtiéndola en una aplicación mucho más competitiva.

7.1 Ampliaciones

7.1.1 Aplicación web

- Permitir a un usuario comunicarse a través de la página web permitiendo competir no solo en los dispositivos móviles sino también en las aplicaciones para PC.
- Añadir nuevos servicios, como estadísticas visuales para los usuarios en función del uso de la aplicación.
- Compartir medallas con el dispositivo móvil.

7.1.2 Web Service

- Mejorar la búsqueda temática para obtener una solución de mayor calidad.
- Controlar la creación de canales abusivos.
- Definir nuevos canales enlazando la temática con la geolocalización.
- Crear servicios de sugerencia de amistad.
- Permitir transmitir archivos multimedia.
- Añadir servicios publicitarios

7.2 Mejoras

7.2.1 Aplicación web

- Emplear un tipo de desarrollo orientado a lógica de negocios complejas.
- Mejoras genéricas de usabilidad.
- Mejoras de diseño orientadas a la aplicación.
- Definir una comunicación segura entre el usuario y la aplicación.
- Contratar un dominio.

7.2.2 Web Service

- Convertir el almacenamiento en una base de datos NoSql como BigTable.
- Mejorar la eficiencia.
- Utilizar algún servidor con variabilidad de recursos dependiendo la carga.

8- Conclusión

El desarrollo de la plataforma de comunicación JET! y su análisis temático ha permitido tener en nuestras manos una aplicación totalmente funcional, que es el resultado de numerosas horas de desarrollo y de búsqueda de información. Esta plataforma está totalmente accesible para poder ser ampliada y mejorada, con el objetivo de conseguir una de las mejores aplicaciones para dispositivos móviles orientados a la comunicación y en especial a las necesidades de los usuarios. Este proyecto me ha enseñado a trabajar en equipo teniendo claro los objetivos de cada desarrollador, permitiendo un proyecto final constituido por los granos de arena aportados por cada uno.

Esta aplicación ofrece a los usuarios una alternativa a los grupos de mil temas creados en las aplicaciones actuales y ofrecerles un espacio único para centrarse en lo que de verdad necesitan y con el componente del análisis de la temática esta aplicación llega más allá, obteniendo información no solo de las personas cercanas sino de los mayores expertos de un tema.

Esta aplicación ayudará a todos los estudiantes permitiendo buscar información de manera rápida y sin la necesidad de estar delante de Google durante horas para que luego se encuentre información totalmente anticuada.

Con el desarrollo de esta aplicación no solo ha mejorado mis conocimientos sobre nuevas tecnologías sino me ha permitido ver el esfuerzo que se necesita para tener una plataforma totalmente funcional que garantice las bases de una aplicación móvil como JET!

Para poder definir estas bases ha sido necesario obtener ciertos conocimientos sobre nuevas tecnologías, como Nimbees, que nos ha permitido una comunicación eficiente y sencilla y conocimientos de como se puede desarrollar un Web Service que se encargue de toda la base de una aplicación móvil y el control de los datos necesarios para su correcto funcionamiento. Muchos desarrolladores consideran que el desarrollo de un Web Service es la parte fea una aplicación pero a lo largo del desarrollo de mi proyecto me he dado cuenta que la mayor responsabilidad se encuentra en la parte servidora aunque no siempre se pueda demostrar.

Para finalizar me gustaría aclarar que actualmente, en el mundo software, existe una idea de que solo las grandes empresas y los grandes proyectos son los que triunfan pero es un gran

error, actualmente todos provienen de pequeñas ideas que nadie consideraban como un hecho especial pero moldeando poco a poco se convertiría en una necesidad para la sociedad. En el mundo de la informática es imposible saber a ciencia cierta que idea destacará al día siguiente y cual fracasará, vivimos en una sociedad donde las tecnologías han conseguido una gran relevancia y no se ha podido llegar aquí sin que muchos proyectos hayan fracasado, por lo que nunca hay que darse por vencido.

9- Agradecimientos

Para finalizar este documento quiero agradecer aquellas personas que me han apoyado para poder terminar mi carrera y en este caso el proyecto de fin de grado. Principalmente dar las gracias a mi familia que son los que han estado desde el principio para cosas buenas y para las malas y siempre pensando en mi bienestar y regalándome el mejor regalo, mi futuro.

También quiero dar las gracias a todos mis amigos que son los que me ofrecen tan grandes momentos con ellos y principalmente a mis compañeros de piso que siempre han estado ahí cuando más los necesitaba.

Cuando se consigue un sueño tan grande como es una carrera los méritos no solo son propios si no que pertenecen también a todos aquellos profesores que nos han ofrecido sus conocimientos durante estos años permitiéndonos mejorar como futuros ingenieros y principalmente como personas aprendiendo los valores del esfuerzo y la recompensa que de él se obtiene y a trabajar en equipo con sus pro y sus contras y la típica frase más usada por los profesores de “Esto los hacéis en 30 minutos” y luego suele durar toda una noche pero finalmente se consiguieron todos los objetivos pero esta adquisición de conocimientos nunca acabará esto ha sido un simple proyecto de los muchos que nos encontraremos a lo largo de nuestra vida como Ingenieros Informáticos.

También me gustaría agradecer a todos los compañeros de Gloin que han estado ayudándome en todo lo que necesitaba y resolviéndome las dudas en tiempo record :)

Finalmente agradecer al resto de compañeros de la carrera que ha sido un gran grupo aunque no fuéramos muy participativos en clase creo que hemos aprendido mucho unos de otros.

Gracias por vuestra ayuda y apoyo que me han permitido cumplir un sueño y sin vuestra ayuda hubiera sido un camino muy complicado.

10- Bibliografía

En este apartado se mencionan las fuentes y referencias básicas utilizadas para poder elaborar nuestro proyecto. Es muy complicado recopilar todas las fuentes utilizadas debido a la gran cantidad de tecnologías analizadas y distintas formas de conseguir ofrecer un servicio de calidad.

- *Metodologías de desarrollo software* http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software
- *Metodologías ágiles* <http://www.proyectosagiles.org/que-es-scrum>
- *Comparación sistemas operativos móviles* http://www.pcactual.com/articulo/laboratorio/comparativas/13412/ponemos_nota_los_sistemas_operativos_moviles_mas_populares.html
- *Oauth* <http://blog.s21sec.com/2012/06/oauth-open-authorization-protocol.html>
- *Patrones de diseño web service* http://www.upch.edu.pe/dui/userdocs/lib/exe/fetch.php/blog/20100317_linuxweek_soaroaphp.pdf
- *Que es y como se hacen los tesauros* <http://orion.lcg.ufrj.br/Dr.Dobbs/books/book5/chap09.htm>
- *Construcción tesauo* <http://crpit.com/confpapers/CRPITV74Yang.pdf>
- *Más información sobre Tesauo* <http://es.wikipedia.org/wiki/Tesauo>
- *Más información sobre Tesauo* <http://eprints.rclis.org/12079/1/ad0907.pdf>

- *Maven* <http://maven.apache.org>
- *Mysql* <http://www.mysql.com>
- *Apache Tomcat* <http://tomcat.apache.org>
- *Certificados de seguridad* http://es.wikipedia.org/wiki/Transport_Layer_Security