



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL DOCENTE

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Daniel Pareja Valle

Tutor: Juan Vicente Capella Hernández

2017/2018

Resumen

En este trabajo final de grado (TFG) se implementa una aplicación para uso docente en la plataforma Android, se pretende facilitar la comunicación en los centros educativos entre profesorado, alumnado y padres de los alumnos.

La funcionalidad consiste en que, mediante esta app, el profesor pueda subir contenido al servidor que se podrá descargar desde esta misma, también ofrece la opción de poder chatear tanto con padres y el alumno, además de que los usuarios podrán obtener información del centro.

Para el desarrollo de esta aplicación se ha utilizado el IDE oficial de Android, llamado Android Studio, se ha creado un servidor apache local para el almacenamiento de la base de datos y uso de los servicios desde la aplicación Android.

He contado con la colaboración de la dirección del colegio público Sant Joan Evangelista de Massamagrell para saber las necesidades que podría cubrir.

Palabras clave: App, dispositivos móviles, aplicación interactiva, Docencia, Android

Abstract

In this Final Degree Project, a teaching application is implemented on the Android platform in order to facilitate the communication in schools between teachers, students and their parents.

The functionality is that, through this application, the teacher can upload material to the server that can be downloaded from it. It also offers the option of being able to chat with both the parents and the student, as well as the fact that users can obtain information from the center.

For the development of this application, it has been used the official IDE of Android, called Android Studio, a local 'apache' server has been created for the storage of the database and the use of the Android application services.

For this purpose, I have had the collaboration of the public-school Sant Joan Evangelista de Massamagrell management to know the needs that could be covered.

Keywords : App, mobile devices, interactive application, Teaching, Android

Tabla de contenidos

| | | |
|-------|-------------------------------------------------|----|
| 1. | Introducción | 7 |
| 1.1 | Objetivos | 8 |
| 2. | Android..... | 9 |
| 2.1 | ¿Qué es Android? | 9 |
| 2.2 | Historia | 9 |
| 2.3 | Arquitectura | 10 |
| 2.2.1 | Aplicaciones..... | 10 |
| 2.2.2 | Framework de Aplicaciones | 10 |
| 2.2.3 | Librerías..... | 10 |
| 2.2.4 | Entorno de ejecución de Android..... | 11 |
| 2.2.5 | Núcleo Linux | 11 |
| 2.4 | Arquitectura..... | 11 |
| 3. | Firestore | 14 |
| 4. | XAMPP | 14 |
| 5. | Estado del Arte | 15 |
| 5.1 | Sistemas operativos para móviles..... | 15 |
| 5.2 | Aplicaciones relacionadas con la docencia | 16 |
| 5.2.1 | ClassDojo:..... | 16 |
| 5.2.2 | TokAppSchool: | 17 |
| 5.2.3 | Trello: | 17 |
| 6. | Análisis de requisitos | 17 |
| 6.1 | Usuarios | 18 |
| 6.2 | Requisitos funcionales | 18 |
| 6.3 | Requisitos no funcionales | 22 |
| 6.4 | Diagrama de casos de uso | 23 |
| 6.5 | Mockups software..... | 26 |
| 6.6 | Diagrama de clases | 31 |
| 7. | Implementación | 34 |
| 7.1 | Arquitectura de aplicación..... | 34 |
| 7.1.1 | Arquitectura hardware | 34 |

| | | |
|--------|-------------------------------------|----|
| 7.1.2 | Arquitectura Software | 34 |
| 7.2 | Base de Datos..... | 35 |
| 7.3 | Servicios PHP..... | 37 |
| 7.3.1 | Crear Tabla: | 38 |
| 7.3.2 | Enviar Mensaje:..... | 38 |
| 7.3.3 | Enviar Notificación:..... | 38 |
| 7.4 | Aplicación Android | 39 |
| 7.4.1 | Login..... | 40 |
| 7.4.2 | Registro..... | 41 |
| 7.4.3 | Aplimenu | 42 |
| 7.4.4 | Ubicación..... | 42 |
| 7.4.5 | Contacto..... | 42 |
| 7.4.6 | Noticias..... | 43 |
| 7.4.7 | Comedor | 43 |
| 7.4.8 | Recursos | 43 |
| 7.4.9 | Gestionchat..... | 43 |
| 7.4.10 | Listachat | 43 |
| 7.4.11 | Listauser | 43 |
| 7.4.12 | Mensajería | 43 |
| 7.4.13 | Asignatura | 45 |
| 7.4.14 | Listarcarp..... | 46 |
| 7.4.15 | Docuementos..... | 46 |
| 7.4.16 | Listatab | 46 |
| 7.4.17 | Permisos | 46 |
| 7.4.18 | Mensajeriatab..... | 46 |
| 6. | Presupuesto | 47 |
| 7. | Conclusiones | 48 |
| 7.1 | Valoración Personal..... | 48 |
| 7.2 | Futuras ampliaciones o mejoras..... | 48 |
| 8. | Bibliografía | 49 |

Tabla de Figuras

| | |
|--------------------------------------------------------|----|
| <i>Figura 1 : Arquitectura Android</i> | 10 |
| <i>Figura 2 : Control XMAPP</i> | 15 |
| <i>Figura 3 : Sistemas Operativos Cuota de Mercado</i> | 15 |
| <i>Figura 4 : Android lidera el mercado</i> | 16 |
| <i>Figura 5 : ClassDojo</i> | 16 |
| <i>Figura 6 : TokAppSchool</i> | 17 |
| <i>Figura 7 : Trello</i> | 17 |
| <i>Figura 8: Casos de uso</i> | 25 |
| <i>Figura 9 : Inicio MU</i> | 26 |
| <i>Figura 10 : Registro MU</i> | 26 |
| <i>Figura 11 : Menú MU</i> | 27 |
| <i>Figura 12 : Mock Ups MU</i> | 27 |
| <i>Figura 13: Asignaturas MU</i> | 28 |
| <i>Figura 145: Documentos MU 2</i> | 28 |
| <i>Figura 156: Tablón MU 1</i> | 29 |
| <i>Figura 167: Tablón MU 2</i> | 29 |
| <i>Figura 17 : Chat MU 1</i> | 30 |
| <i>Figura 18 : Chat MU 2</i> | 30 |
| <i>Figura 19 : Chat MU 3</i> | 30 |
| <i>Figura 20: Diagrama de clases</i> | 33 |
| <i>Figura 21 :Arquitectura</i> | 34 |
| <i>Figura 22 :Relaciones Base de Datos</i> | 37 |
| <i>Figura 23: Login</i> | 41 |
| <i>Figura 24: Registro</i> | 41 |
| <i>Figura 25: Menú</i> | 42 |
| <i>Figura 26:layoutMensaje</i> | 44 |
| <i>Figura 27: CardViewTablon</i> | 47 |

1. Introducción

Las TIC actualmente tienen una gran importancia en la educación, facilitan la obtención de información y la comunicación entre los diferentes niveles educativos.

Ayudan a lograr un acceso global a la educación y mejoran la calidad y la igualdad de los contenidos, por ello cualquier centro como servicio público debe integrar las TIC para alfabetización digital, material didáctico, fuentes de información ...

En la actualidad, muchos maestros quieren contar con recursos informáticos y con internet para su docencia ya que facilitan la realización de las tareas de investigación de los alumnos y personal docente, posibilitan la realización de trabajos o proyectos en común al alumnado y que docentes puedan colaborar con otros docentes.

En estos últimos años el uso de dispositivos móviles ha aumentado, según un estudio presentado por Cisco, en el año 2020 el 70 % de la población mundial tendrá un dispositivo móvil, existirán alrededor de 5500 millones de usuarios móviles.

El uso del PC ha sido superado por los dispositivos móviles, por eso muchas compañías están invirtiendo en el desarrollo de 'apps' para poder competir en el mercado.

El sistema operativo más usado en el mundo es Android, la plataforma de Google ha superado a Windows. Según datos de "StatCounter" Android ha alcanzado la cifra de un 37,93% de usuarios, mientras que la plataforma de Microsoft se queda con un 37,91%.

En este contexto he decidido desarrollar una aplicación para uso docente, para hacer más cómoda la comunicación entre profesorado y alumnado mediante el uso de un sistema de almacenamiento, un tablón para colgar notificaciones y el uso de chat entre los usuarios de la misma.

Las aplicaciones desarrolladas hasta la fecha la gran mayoría son de pago y no engloban todas estas características, además que Conselleria, en este caso ofrece una web bastante limitada para estos colegios públicos, pero no ofrece ninguna app gratuita para los usuarios de estos centros.

Es la principal motivación para el desarrollo de este proyecto.

1.1 Objetivos

El objetivo del proyecto es realizar una aplicación móvil en la plataforma Android para que se pueda distribuir mediante la Play Store para su instalación. Aunque cada centro que quiera usar la aplicación deberá hacer un registro previo para poder solicitar los datos y personalizar algunas funcionalidades.

Para poder llevar a cabo este objetivo se deben alcanzar estas metas:

- El usuario podrá obtener información del centro, datos como ubicación, calendario, menú del comedor...
- Cada usuario estará identificado con uno de estos tres roles: padre, alumno y profesor, esto permitirá la asignación de permisos para subir o descargar contenido de la aplicación, así como la modificación del contenido del tablón y el uso del chat.
- Notificar los cambios realizados a los grupos de usuarios interesados.
- Seguimiento de los usuarios alumnos por parte de los padres.
- Facilitar la comunicación entre profesor y padres.

Para ello debemos familiarizarnos con el entorno de desarrollo utilizado Android y PHP.

2. Android

2.1 ¿Qué es Android?

Android es un sistema operativo basado en el núcleo Linux, está diseñado para dispositivos móviles con pantalla táctil, relojes inteligentes, televisores y automóviles.

2.2 Historia

En el año 2003 Android fue desarrollado por la empresa Android Inc. La empresa se centraba en el desarrollo de software para móviles, esta empresa fue respaldada por Google económicamente.

En 2005 Google compró Android, el 5 noviembre 2007 fue presentado junto la fundación del **Open Handset Alliance** (un consorcio de compañías de hardware, software y telecomunicaciones) liderada por Google. Al mismo tiempo liberó la mayoría de código fuente bajo la licencia Apache, libre y de código abierto, para que se pudiese implementar en cualquier dispositivo. Se daba como garantía que está basado en el sistema operativo Linux.

Ese mismo año se lanzó la primera versión de Android SDK (*Software Development Kit*), son un conjunto de herramientas de desarrollo, depurador, bibliotecas, simulador, documentación, etc. Todas las aplicaciones Android se desarrollan en lenguaje java con este kit

De hecho, se lanzó antes el sistema operativo que el primer móvil donde colocarlo. La primera vez que lo vimos funcionando fue en octubre de 2008 en un *HTC Dream*. Este móvil creado en USA contaba con la primera versión Android 1.0

Es un móvil deslizable con teclado QWERTY, sensible al tacto, cámara de 3 megapíxeles, ranura microSD, navegador de internet y Email.

Durante todos estos años Android ha ido evolucionando hasta llegar a ser el sistema operativo más utilizado.

2.3 Arquitectura

Android contiene una pila de software agrupado en niveles siendo este sistema operativo, middleware y aplicaciones básicas para el usuario. En la Figura 1 podemos observar estas agrupaciones, así como sus módulos principales.

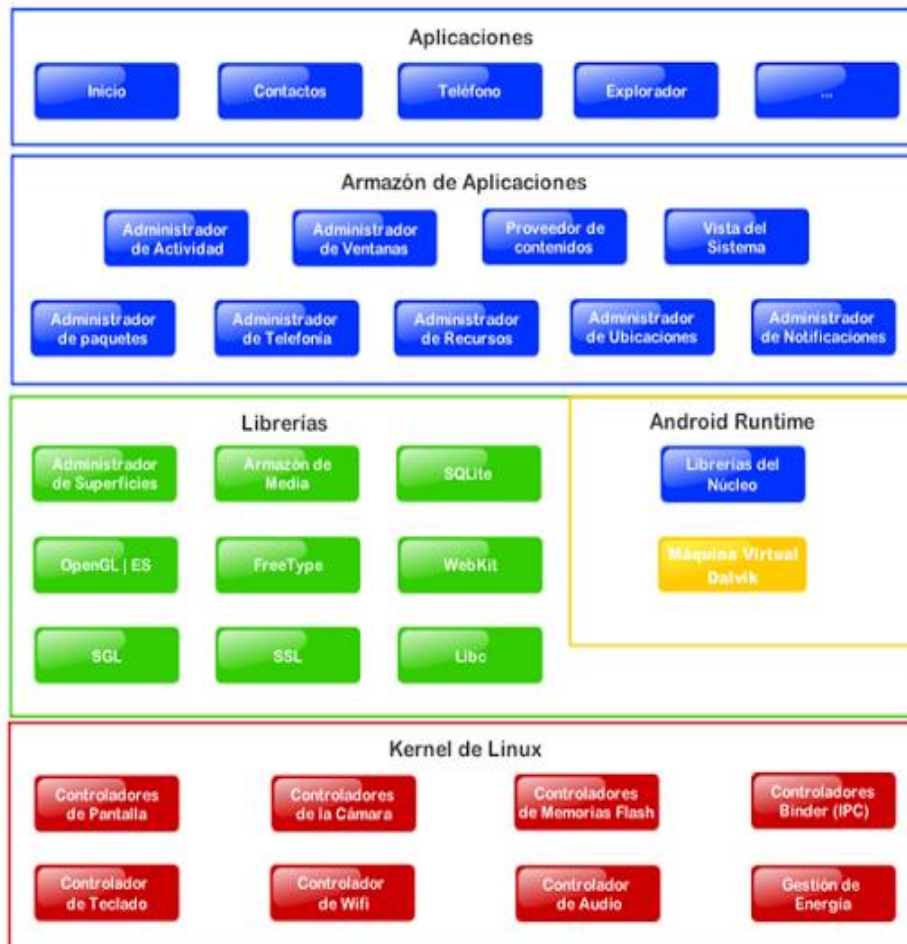


Figura 1 : Arquitectura Android

2.2.1 Aplicaciones

En este nivel se incluyen todas las aplicaciones, las que añade el usuario y las que vienen por defecto con el sistema operativo, están en el primer nivel por que utilizan todo lo posterior.

2.2.2 Framework de Aplicaciones

Conjunto de herramientas de desarrollo, todas las aplicaciones utilizan las mismas API y el mismo Framework.

2.2.3 Librerías

Estas librerías se han creado mediante C/C++ son junto con el núcleo los pilares que constituyen Android.

2.2.4 Entorno de ejecución de Android

En este nivel se sitúan las *Core Libraries*, librerías con clases java y la máquina virtual *Dalvik*.

Dalvik es un intérprete que permite ejecutar aplicaciones programadas en Java, requiere poca memoria y ejecuta varias instancias de la máquina simultáneamente gracias al apoyo del núcleo en la gestión de hilos memoria y procesos, la gran diferencia con una máquina virtual de java es que utiliza los registros como unidad primaria de almacenamiento de datos en lugar de la pila, no se basa en el *bytecode* de java, sino que utiliza archivos “.dex” por tanto convierte los archivos “.class” en “.dex”

2.2.5 Núcleo Linux

El núcleo de Android utiliza Linux v 2.6 y se encarga de la seguridad. Gestión de memoria, gestión de procesos, modelo de controladores y pila de red y actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

2.4 Arquitectura

- Android **1.0** Nivel de **API 1** (septiembre 2008):
No tubo uso comercial ya que fue la primera versión.
- Android **1.1** Nivel de **API 2** (febrero 2009):
Se solucionaron errores de la anterior versión, es la opción a escoger si queremos desarrollar una aplicación para todos los dispositivos, casi no existen usuarios con esta versión.
- Android **1.5** Nivel de **API 3** (abril 2009) **Cupcake**:
Teclado en pantalla con predicción de texto, grabación avanzada de audio y video. Widgets y Live Folders.
- Android **1.6** Nivel de **API 4** (septiembre 2009) **Donut**:
Búsqueda avanzada, incorpora *gestures* y síntesis de texto de voz, trabaja con diferentes densidades de pantalla, nuevo atributo *XML onClick*.
- Android **2.0** Nivel de **API 5** (octubre 2009) **Éclair**:
La mayoría de los fabricantes pasaron directamente de la versión 1.6 a la 2.1.API para manejar bluetooth 2.1. Optimizado la velocidad del hardware, nueva interfaz del navegador y soporte para HTML5.
- Android **2.1** Nivel de **API 7** (enero 2010):
Permite introducir un campo de texto dictando sin necesidad de usar el teclado. Fondos de pantalla animados, información sobre la señal de la red actual que posea el dispositivo.
- Android **2.2** Nivel de **API 8** (mayo 2010) **Froyo**:
Incremento de velocidad CPU, instalar aplicaciones en un medio de almacenamiento externo, aplicaciones se actualizan de forma automática cuando aparece una nueva versión, se mejora la



- conectividad (se puede dar acceso a internet a otros dispositivos) se añade el soporte a Wi-Fi IEEE 802.11n y notificaciones *push*.
- Android **2.3** Nivel de **API 9** (diciembre 2010) **Gingerbread:**
Soporta mayores tamaños de pantalla y resoluciones, mejora de la funcionalidad de “cortar, copiar y pegar” y teclado en pantalla con capacidad multitáctil.
 - Android **3.0** Nivel de **API 11** (febrero 2011) **Honeycomb:**
Se lanza la versión 3.0 optimizada para dispositivos con pantallas grandes. Se incluyen los ‘*fragments*’, con los que diseñamos diferentes elementos de la interfaz de usuario, las teclas físicas son reemplazadas por teclas en pantalla. Primera versión que soporta procesadores multinúcleo. Android 3.0 es compatible con las aplicaciones creadas en las versiones anteriores.
 - Android **3.1** Nivel de **API 12** (mayo 2011):
Manejo de dispositivos conectados por USB (tanto host como dispositivo). Protocolo de transferencia de fotos y video y de tiempo real
 - Android **3.2** Nivel de **API 13** (julio 2011):
Optimizaciones para distintos tipos de tableta. Sincronización multimedia desde SD.
 - Android **4.0** Nivel de **API 14** (octubre 2011) **Ice Cream Sandwich:**
Se unifican las versiones anteriores en una sola compatible con cualquier dispositivo, reconocimiento facial, mejora de reconocimiento por voz. Se mejora la API para comunicaciones por NFC y la integración con redes sociales.
 - Android **4.0.3** Nivel de **API 15** (diciembre 2011):
Mejoras en *APIs* de redes sociales, calendario, revisor ortográfico, texto a voz y bases de datos entre otros.
 - Android **4.1** Nivel de **API 16** (julio 2012) **Jelly Bean:**
Fluidez de interfaz de usuario, aumento de la velocidad del procesador al tocar la pantalla, se mejoran notificaciones, los widgets de escritorio pueden ajustar su tamaño y hacerse sitio de forma automática al situarlos en el escritorio. Mejoras en Google *Search* y soportes para usuarios internacionales.
 - Android **4.2** Nivel de **API 17** (noviembre 2012):
Cuentas de usuario en el mismo dispositivo solo disponible en tabletas, los widgets de escritorio pueden aparecer en la pantalla de bloqueo.
 - Android **4.3** Nivel de **API 18** (julio 2013):
Mejoras en múltiples áreas entre ellas perfiles restringidos, se da soporte a bluetooth Low Energy, permite a los dispositivos comunicarse con los periféricos con bajo consumo de energía, se da soporte a OpenGL ES 3.0

- Android **4.4** Nivel de **API 19** (octubre 2013) **KitKat**:
Con esta versión se pretende que Android esté disponible en una gama aún más amplia de dispositivos, se mejora el rendimiento en los dispositivos y la interfaz de usuario.
- Android **5.0** Nivel de **API 21** (noviembre 2014) **Lollipop**:
Nuevas plataformas como *Android Wear*, *Android TV* y *Android Auto*, el modo ahorro de batería se activa por defecto. Incorpora soporte nativo para OpenGL ES 3.1.
- Android **5.1** Nivel de **API 22** (marzo 2015):
Se añade soporte para varias tarjetas SIM, para acceder a esta API la aplicación ha de estar firmada con un certificado que coincide con el que tiene el usuario en la tarjeta UICC.
- Android **6.0** Nivel de **API 23** (octubre 2015) **Mashmallow**:
Mucha más protección a la privacidad de los usuarios, se realiza copia de seguridad automática de todos los datos de las aplicaciones. Integra el asistente por voz *Now on Tap*. Se puede utilizar parte de un dispositivo de almacenamiento externo, para que sea usado como almacenamiento interno. Podemos fragmentar formatear y encriptar una tarjeta SD para ser usada como memoria interna, montar lápices USB de forma nativa. Mejoras de posicionamiento utilizando redes Wifi y dispositivos bluetooth.
- Android **7.0** Nivel de **API 24** (julio 2016) **Nougat**:
Se pueden abrir varias aplicaciones al mismo tiempo en la pantalla, las notificaciones han sido rediseñadas para un uso más ágil. Más opciones para personalizar el estilo de los mensajes. Se incorpora la compilación *Just in Time*, donde no se compila hasta que el código va a ser ejecutado, el gestor de batería *Doze* se activa poco tiempo después de apagarse la pantalla, ahorrando batería, se ha añadido a la API para gráficos 3D, *Vulkan*, como alternativa a OpenGL.
- Android **7.1** Nivel de **API 25** (diciembre 2016):
Accesos directos a aplicaciones, desde el icono de aplicación con pulsación prolongada, aparecen varias opciones que podremos seleccionar. Posibilidad de instar imágenes desde teclado.

Se debe elegir la versión que más convenga para crear una aplicación en nuestro caso he elegido el *SDK Version API 25*, Android 7.1.1, *Build Tools Version 25.0.2*, debido a que tiene un mejor sistema de notificaciones y el dispositivo móvil para la fase de depuración lo soporta.

3. Firebase

Firebase es una plataforma de Google para el desarrollo de aplicaciones móviles y aplicaciones Web. Esta plataforma ofrece una gran cantidad de servicios para el desarrollo de aplicaciones:

- **Firestore Analytics**, proporciona una visión profunda sobre el uso de la aplicación por parte de los usuarios.
- **Firestore Cloud Messaging**, plataforma para mensajes y notificaciones para las plataformas Android, IOS y aplicaciones web.
- **Firestore Auth**, puede autenticar los usuarios utilizando únicamente código del lado del cliente, la autenticación de los usuarios con email y contraseña que se almacenara en *Firebase* y la autenticación con las plataformas *Facebook, Twiter, Google y Github*.
- **Firestore Database**, proporciona una base de datos en tiempo real y back-end, este servicio proporciona una API que permite sincronizar la información y almacenarla en *Firebase*
- **Firestore Storage**, servicio para cargas y descargas de archivos para aplicaciones *Firebase*.

En este proyecto se ha usado el servicio *Firebase Cloud Messaging* para los mensajes y las notificaciones del chat y el tablón.

4. XAMPP

Para la implementación del registro de usuarios y los servicios de mensajes en un principio se pensó hacerlo con una base de datos integrada en la aplicación y el uso de SQLITE, pero se quedaba corta, así que se optó por el uso WAMP que permite el uso de un servidor Apache, gestión de Base de datos MySQL y los intérpretes para lenguajes de script PHP.

EL nombre es un acrónimo:

- **W**, referencia a los sistemas operativos en el que vamos a trabajar, en nuestro caso Windows.
- **A**, servidor HTTP Apache, servidor web de código abierto para distintas plataformas.
- **M**, *MariaDB* sistema de gestión de base de datos derivado de MySQL.
- **P**, hace referencia al lenguaje de programación PHP.

Para la gestión de la base de datos se ha usado la herramienta PhpMyAdmin, se han creado servicios en PHP para las consultas necesarias y el servidor Apache en local para poder simular la aplicación.

| Modules | | | | |
|---------|--------|----------------|---------|------------------------|
| Service | Module | PID(s) | Port(s) | Actions |
| | Apache | 10788 10308 | 80, 443 | Stop Admin Config Logs |
| | MySQL | 3168 | 3306 | Stop Admin Config Logs |

Figura 2 : Control XMAPP

5. Estado del Arte

En este apartado hablaremos de los sistemas Operativos de los dispositivos móviles y compararemos algunos de ellos, además mostraremos algunas de las aplicaciones dedicadas a la parte de docencia semejantes a este proyecto.

5.1 Sistemas operativos para móviles

Estos sistemas operativos son mucho más simples y orientados a conectividad inalámbrica. Según el servicio de estadísticas *NetMarketShare*, la cuota de mercado de sistemas operativos móviles en 2017 es el siguiente:

- **Android** 68,54 %
- **iOS** 25,78 %
- **Windows Phone** 1,95 %
- **Otros** 3,73 %

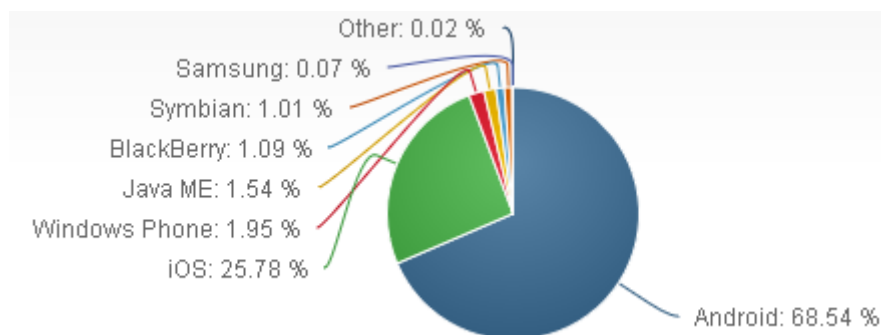


Figura 3 : Sistemas Operativos Cuota de Mercado

Se observa que Android supera con diferencia a los demás sistemas operativos para dispositivos móviles, gran parte de este éxito se debe a que Android es un sistema de código abierto, mientras que IOS y Windows *Phone* no lo son.

En cuanto al mercado global de sistemas operativos, como he citado anteriormente Android es el sistema operativo más usado del mundo, como se muestra en la figura 3 ha ido aumentando hasta alcanzar a Windows.

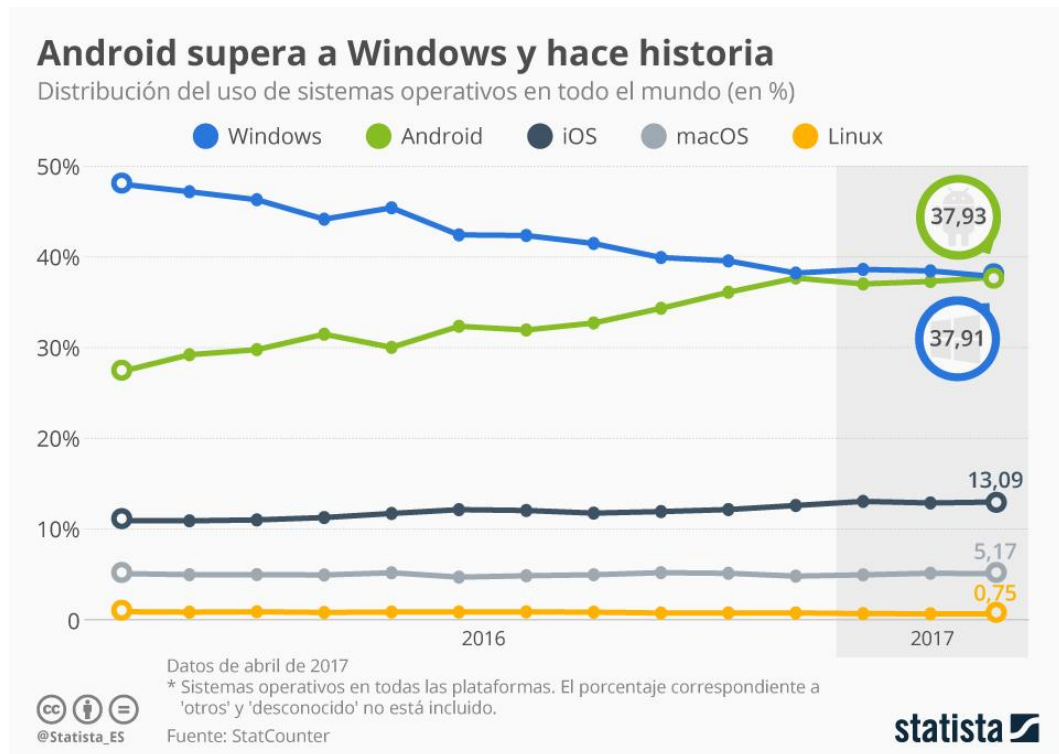


Figura 4 : Android lidera el mercado

5.2 Aplicaciones relacionadas con la docencia

En el ámbito de la docencia existen muchas aplicaciones para dispositivos móviles que intentan facilitar la comunicación de los alumnos, profesores y padres. Estas son algunas de las Apps que intentan mejorar la comunicación entre padres, alumnos y profesores.

5.2.1 ClassDojo:



ClassDojo

Esta aplicación ayuda a los profesores a mejorar las conductas en sus aulas, los profesores comparten con padres y administración datos sobre comportamientos.

Figura 5 : ClassDojo

Los padres pueden revisar el progreso del alumno en la escuela. Generación de informes y fácil gestión.

ClassDojo está centrada en un alumno que necesita más seguimiento paterno, en edades tempranas, hay comunicación entre profesores, padres y dirección pero el uso de los alumnos está centrado en añadir fotos y videos en sus carpetas, tiene bastantes problemas con el acceso, el seguimiento de los alumnos solo se puede ver semanalmente, no permite el cambio de contraseña, fallos con la visualización de los mensajes y el idioma en el que está definida es el inglés, por tanto si un padre no entiende inglés no podrá interactuar con la aplicación.

5.2.2 TokAppSchool:



Figura 6 : TokAppSchool

Aplicación de mensajería educativa, pretende agilizar la comunicación entre centros, profesores, alumnos y padres.

Y comunicación interna del centro, en tiempo real, permite enviar mensajes a padres y alumnos sin que sea visible su número de móvil, para formalizar reuniones, autorizar salidas, teniendo la posibilidad de solicitar respuesta o no a sus comunicados.

5.2.3 Trello:



Figura 7 : Trello

Es un gestor de tareas que mejora la productividad de cualquiera profesional, tareas, informes, reuniones con padres, fechas de exámenes.

Permite establecer alertas e interactuar con otros miembros de tu equipo para evitar que se te pasen citas de interés. Esta más orientada a los maestros, pero permite la interacción con los demás usuarios.

Existen más aplicaciones similares que ayudan a la comunicación entre usuarios padres y alumnos, pero la aplicación que se desarrolla en este proyecto permite personalizar las funcionalidades dependiendo las necesidades de cada centro e integra poder compartir documentos, chat en tiempo real y el tablón donde los profesores podrán anunciar comunicados.

6. Análisis de requisitos

En este apartado hablaremos de los requisitos necesarios que se han tenido en cuenta para desarrollar la aplicación.

Esta fase es la más importante para poder desarrollar el software ya que se realiza un estudio y captura de requisitos necesarios, mediante reuniones con nuestro cliente para cubrir todas las necesidades que tenga nuestro cliente.

En este caso ha habido reuniones con la dirección del centro público Sant Joan Evangelista de Massamagrell. En las que se especificó los roles de usuario que se necesitarían, la creación de una contraseña física para el control de usuarios, y la personalización de la interfaz

Una vez recogida todas las necesidades se ha procedido al estudio de las tecnologías para poder implementar la aplicación.

6.1 Usuarios

En toda aplicación software se va a necesitar un usuario administrador que tendrá acceso a todas las funcionalidades del programa.

Tendremos 3 roles de usuario, alumnos, profesores y padres.

El administrador tiene acceso a todas las funcionalidades de la aplicación.

Dependiendo el rol que tenga cada usuario tendrán unos permisos u otros.

Los profesores son los que pueden iniciar cualquier tablón con cualquier curso.

En cuestión de mensajes, podrán modificar y comentar e incluso podrán marca una opción para que sus mensajes no puedan ser respondidos solo leídos.

Los padres pueden iniciar chats con los profesores de sus alumnos, así como ver todas las notificaciones que reciben sus hijos, acceso directo con todos los profesores del curso.

Los alumnos pueden iniciar chats personales con el profesor correspondiente a su curso y tienen acceso al contenido que les deja ver el profesor y las notificaciones que se envíen entre padres y profesores.

6.2 Requisitos funcionales

Función descrita como conjunto de entradas, comportamientos y salidas.

Registro de usuario

- **Entradas:** Se rellena un formulario donde se crea un nuevo usuario.
- **Comportamiento:** en esta parte de debe comprobar que los campos introducidos son correctos, que no existe ese usuario en nuestra base de datos, que no se dejen campos obligatorios en blanco.
- **Salidas:** Si todo es correcto el usuario se almacena en la base de datos, en caso de error se le informara con una notificación de los campos que no son correcto y no se introduce en la base de datos.

Login de usuario

- **Entradas:** Al usuario se le pide el DNI como identificador y el password que se creó anteriormente.
- **Comportamiento:** Realizamos una consulta a la base de datos para comprobar los datos metidos por el usuario.
- **Salidas:** si son correctos podrá ingresar a nuestra aplicación, si falla alguno de los datos se le comunica mediante una notificación en la aplicación y le dice cuál es el campo que ha fallado.

Consultar Mapa

- **Entradas:** el usuario accede mediante el menú de la aplicación.
- **Comportamiento:** Se accede a la librería de Google maps .
- **Salidas:** Se muestra mediante Google maps la ubicación del centro y como llegar a él.

Consultar Comedor

- **Entradas:** el usuario accede mediante el menú de la aplicación.
- **Comportamiento:** Se accede al documento pdf de la web del colegio con una consulta a la web.
- **Salidas:** Se visualiza el menú del comedor y tiene la opción de descargarlo.

Consultar Centro

- **Entradas:** el usuario accede mediante el menú de la aplicación.
- **Comportamiento:** Se accede a la página web del centro y se carga en la aplicación.
- **Salidas:** Se muestra la página web del centro.

Añadir a mi lista

- **Entradas:** los usuarios se les pide el curso correspondiente y el usuario.
- **Comportamiento:** Se comprueba en la base de datos si es correcto.
- **Salidas:** Si es correcto se les añade los permisos correspondientes la primera vez que inician, luego no es necesario.

Consultar tablón

- **Entradas:** los usuarios acceden desde el menú de recursos.
- **Comportamiento:** Se comprueba en la base de datos sus permisos como usuario.
- **Salidas:** Se les muestra los tablonos a los que tiene acceso y puede visualizar.

Crear conversación

- **Entradas:** el usuario selecciona a otro usuario para crear la conversación.
- **Comportamiento:** el usuario crea una nueva tabla en la base de datos.
- **Salidas:** el usuario accede a la sala de chat con la persona seleccionada.

Consultar conversación

- **Entradas:** el usuario selecciona el chat que tiene ya creado.

- **Comportamiento:** Se carga todo el contenido de esa sala de chat de la base de datos.
- **Salidas:** el usuario visualizo todo el contenido de esa conversación.

Enviar mensaje

- **Entradas:** el usuario introduce el mensaje y presiona en el botón enviar.
- **Comportamiento:** se introduce el mensaje en la base de datos y se recarga la pantalla se envía notificación al usuario receptor.
- **Salidas:** el usuario visualiza su mensaje en la pantalla.

Subir documento

- **Entradas:** el usuario adjunta documento a la carpeta seleccionada.
- **Comportamiento:** Se almacena el documento en el servidor y se crea una referencia en la base de datos.
- **Salidas:** el usuario observa en la aplicación su documento almacenado en la carpeta correspondiente.

Dar permisos de mensaje tablón

- **Entradas:** El usuario cuando crea su entrada puede seleccionar los usuarios que lo pueden visualizar y si pueden responder a ese mensaje.
- **Comportamiento:** se añade a la base de datos los permisos seleccionados por el usuario.
- **Salidas:** se le comunica al usuario que los permisos han sido creados correctamente.

Descargar documento

- **Entradas:** el usuario selecciona el documento la opción de descarga.
- **Comportamiento:** se descarga el archivo desde el servidor.
- **Salidas:** El usuario recibe el documento en su dispositivo.

Añadir entrada de tablón

- **Entradas:** El usuario añade una entrada al tablón.
- **Comportamiento:** Se guarda la entrada en la base de datos.
- **Salidas:** El usuario puede visualizar los permisos para añadir a su entrada recién creada.

Eliminar entrada de tablón

- **Entradas:** El usuario le da a una entrada suya a la opción eliminar.
- **Comportamiento:** Se comprueba en la base de datos si es correcto y se elimina ese contenido de la base de datos.

- **Salidas:** Si es correcto ningún usuario podrá ver esa entrada ya que habrá sido eliminada.

Listar profesores

- **Entradas:** los usuarios seleccionan lista de profesores del curso correspondiente.
- **Comportamiento:** Se realiza una consulta a la base de datos y recibimos todos los profesores que corresponde a ese curso.
- **Salidas:** el usuario visualiza el nombre de los profesores del curso correspondiente.

Listar alumnos

- **Entradas:** los usuarios seleccionan lista de alumnos del curso correspondiente
- **Comportamiento:** Se realiza una consulta a la base de datos y recibimos todos los alumnos que corresponde a ese curso.
- **Salidas** el usuario visualiza el nombre de los usuarios del curso correspondiente.

Notificar que se ha leído tablón

- **Entradas:** el usuario marca la opción para que se le notifique.
- **Comportamiento:** Cuando el usuario receptor entra debe marcar la opción de leído del mensaje.
- **Salidas:** Si es correcto el usuario que envía el mensaje recibe una notificación de que se ha leído por el usuario correspondiente.

Notificar que se ha enviado

- **Entradas:** el usuario envía un mensaje a un usuario que no está conectado.
- **Comportamiento:** se crea una notificación y se envía al usuario que recibe el mensaje.
- **Salidas:** el usuario recibe una notificación como que se le ha enviado el mensaje.

Notificar que se ha creado

- **Entradas:** el usuario crea un chat o un tablón.
- **Comportamiento:** Se crea una notificación y se envía a los usuarios correspondiente.
- **Salidas:** los usuarios correspondientes reciben una notificación.

Validar Código

- **Entradas:** Para poder registrarse en la aplicación es necesario la introducción de código.
- **Comportamiento:** Se comprueba que el código es correcto con una consulta a la base de datos.
- **Salidas:** Si es correcto se les añade los permisos correspondientes y se crea el usuario.

Crear Carpeta

- **Entradas:** los usuarios pulsan el botón crear carpeta.
- **Comportamiento:** se almacena esa carpeta en el servidor.
- **Salidas:** los usuarios pueden visualizar la carpeta creado.

Borrar Carpeta

- **Entradas:** los usuarios que han creado la carpeta le dan a la opción de borrar.
- **Comportamiento:** se elimina la carpeta mediante una consulta a la base de datos.
- **Salidas:** la carpeta es eliminada de la aplicación.

Borrar documento

- **Entradas:** el dueño del documento le da a la opción de borrar.
- **Comportamiento:** se elimina el documento del servidor mediante una consulta.
- **Salidas:** el documento es eliminado de la aplicación.

Consultar Chat

- **Entradas:** el usuario le da a un chat de la lista de chats.
- **Comportamiento:** se carga los datos del chat desde la base de datos
- **Salidas:** el usuario accede a la sala de chat.

6.3 Requisitos no funcionales

Conectividad: El dispositivo de tener acceso a internet para comunicarse con el servidor y acceso a GPS con localización válida para que se muestre el recorrido hasta el centro.

Seguridad: el software dispone de un sistema de autenticación, se están tratando datos personales por tanto debe haber una conexión segura a la base de datos y se deben realizar copias de seguridad del servidor, también a cada centro se le agregara una clave para activar a cada usuario.

Persistencia: el servidor debe tener los mecanismos necesarios para evitar caídas de conexión o duplicar su contenido en otro servidor (activo/pasivo).

Facilidad de uso: Debe ser una interfaz cómoda para el usuario y se les debe dar facilidades para entender la aplicación, así como un manual de uso.

Escalabilidad: el sistema debe adaptarse a las necesidades de cada centro y no debe tener problemas en agregar funcionalidades nuevas, además de poder aumentar la capacidad del servidor y la base de datos.

Integración: La aplicación solo funcionara en los sistemas de Android que sean posteriores a la versión de su compilación.

Accesibilidad: La aplicación debe seguir los patrones de accesibilidad de Google.

Soporte: el sistema debe ser actualizado y se deben atender los problemas que tengan los usuarios con su uso.

Robustez de datos: el sistema debe garantizar que no hay usuarios duplicados y debe controlar la entrada correcta de datos, se debe de respetar la ley de protección de datos en todo momento.

6.4 Diagrama de casos de uso

Un diagrama de casos de uso pretende representar de forma gráfica las interacciones que podrá realizar cada tipo de rol en el aplicativo, así como las interconexiones entre los diferentes escenarios que ofrece.

Para ello usaremos UML, es un lenguaje de modelado de sistemas de software, para describir métodos o procesos, se utiliza para definir un sistema.

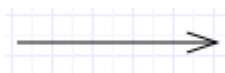
Este sistema define una notación gráfica para representar casos de uso llamado modelo de casos de uso.

Componentes primarios:

- **Casos de uso:** Operaciones o tareas específicas que realiza el sistema tras una orden externa, puede ser invocado desde otro caso de uso o a petición de un actor.
- **Actores:** Alguien o algo que interactúa con el sistema, pero es externo al sistema, Representa un papel, no a un usuario individual del sistema.
- **Sistema:** es un diagrama que contiene el nombre del sistema y los símbolos de los casos de uso.

Relaciones entre casos de uso:

- **Relación de Asociación:** indica la invocación de un caso de uso.



- **Relación de extensión (<<extend>>):** añade acciones que pueden ser opcionales, al comportamiento de un caso de uso general.
- **Relación de inclusión (<<include>>):** incluye el comportamiento completo de un caso de uso general.



- **Relación de generalización:** se especifican los pasos extra que se añaden al caso de uso general, para representar una funcionalidad diferente.



Para esta aplicación el diagrama de casos de usos está representado en la figura:8

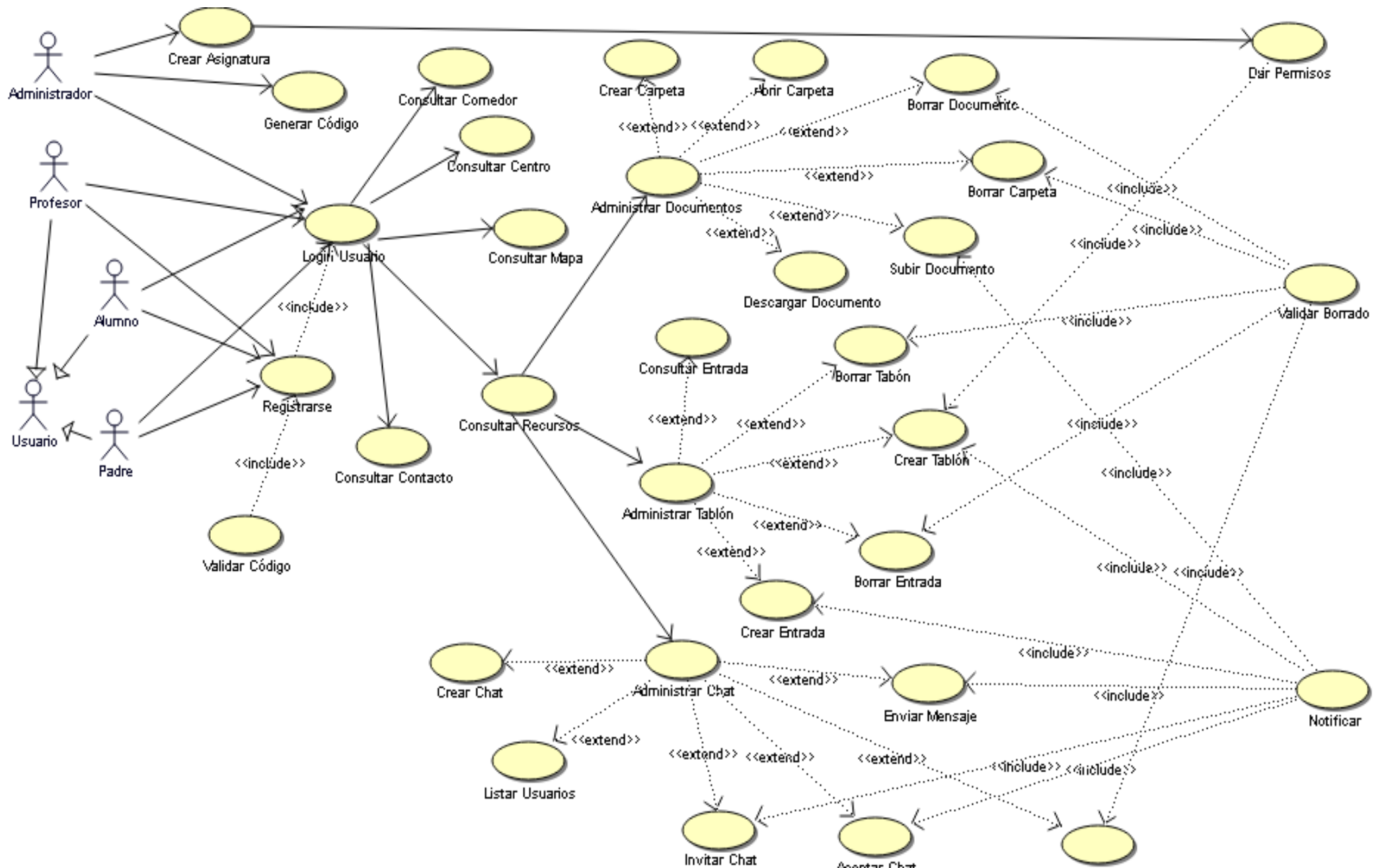


Figura 8: Casos de uso

6.5 Mockups software

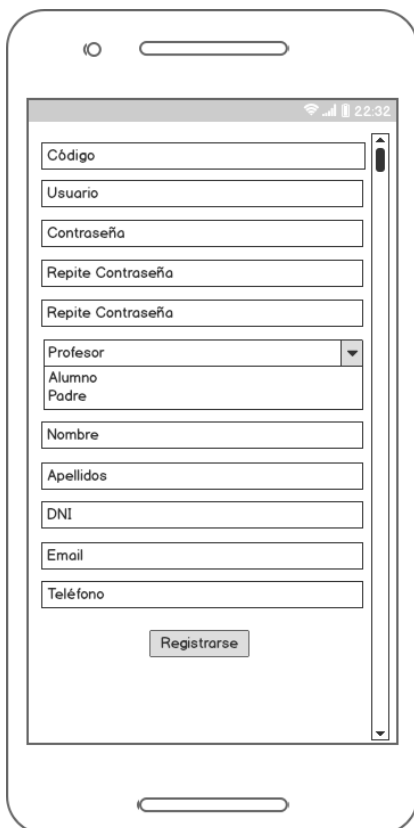
Maqueta de la visión de un usuario de la aplicación



Pantalla de Inicio

La interfaz de esta pantalla incluye dos campos editables, DNI y contraseña para que el usuario pueda acceder a la aplicación, en la parte de abajo dos botones, uno para validar los campos introducidos y el otro para poder ir al formulario de registro.

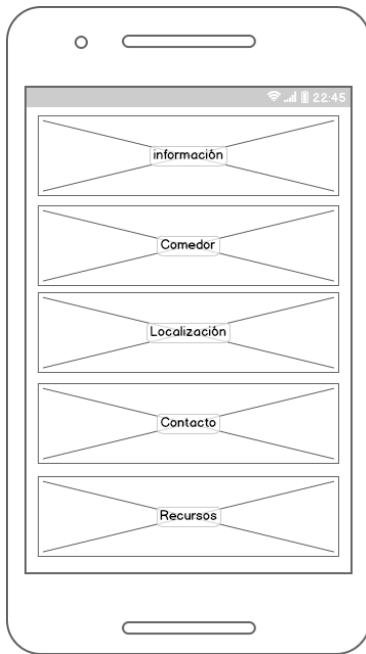
Figura 9 : Inicio MU



Registro

En esta pantalla el usuario debe rellenar un formulario necesario para su registro, el código debe dárselo el administrador, una vez registrado el usuario deberá *loguearse*.

Figura 10 : Registro MU



Menú

Cada caja es un botón que tendrá asociada una imagen, a esta pantalla se accederá una vez confirmado el *login*, accederemos a esta sección donde podremos elegir la funcionalidad que queramos acceder.

Figura 11 : Menú MU

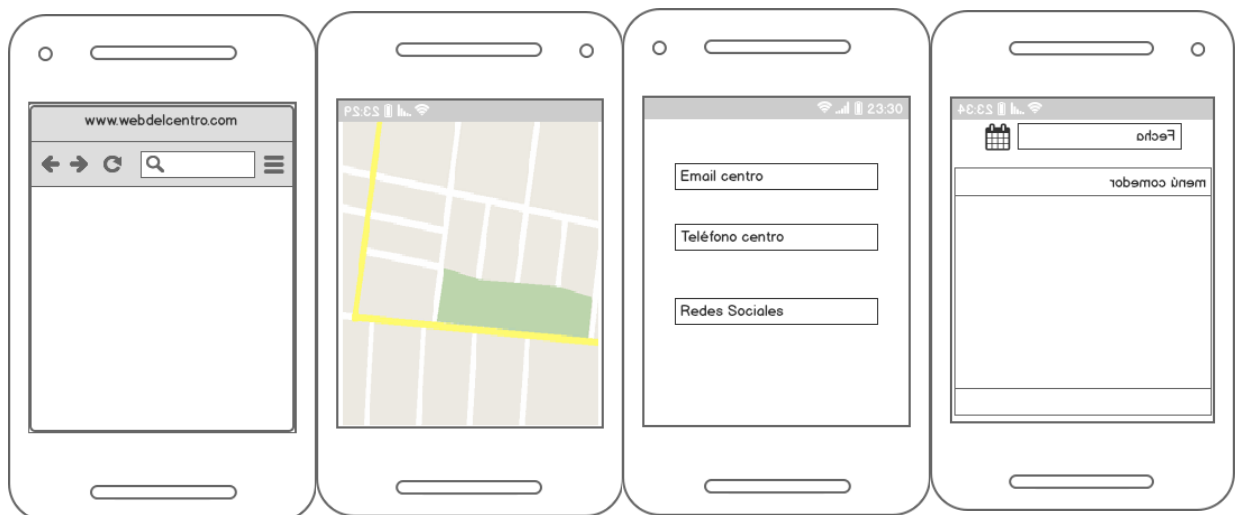


Figura 12 : Mock Ups MU

Información

Esta interfaz mostraremos la web del centro que corresponda.

Localización

Con esta opción accederemos a Google maps donde podremos ver el recorrido que debemos de seguir para llegar al centro.

Contacto

En esta sección el usuario podrá visualizar la forma de comunicarse con el centro educativo.

Comedor

El usuario podrá visualizar el menú de comedor de este centro diciendo el día, semana o mes.

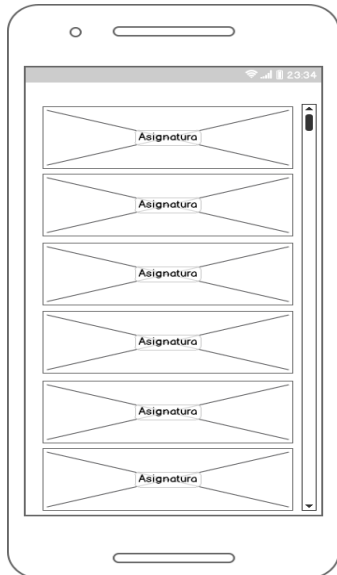


Figura 13: Asignaturas
MU

Recursos

En la sección de recursos el usuario puede elegir entre tres Documentos Tablón y chat.

Si el usuario selecciona uno de estos tres, deberá elegir el nombre de la asignatura correspondiente para poder entrar.

Documentos

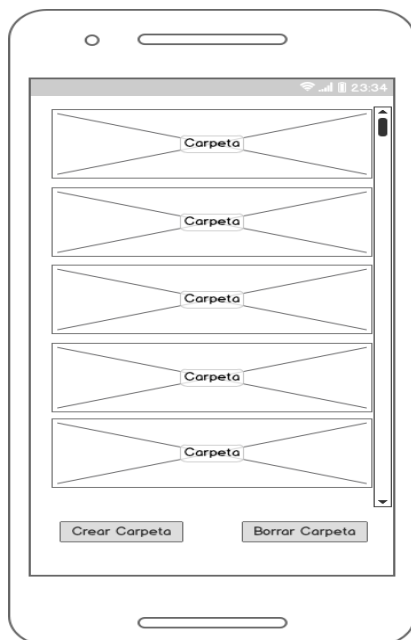


Figura 14 : Documentos MU 1

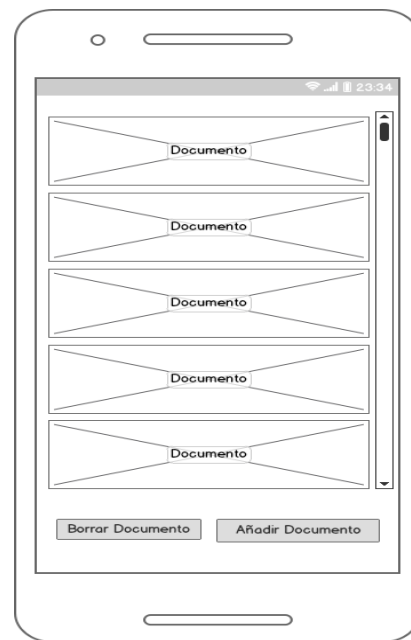


Figura 145: Documentos MU 2

El usuario puede abrir una carpeta seleccionándola, también hay dos botones con los que puede crear una carpeta indicándole su nombre y puede borrar carpetas que el haya creado, el botón de borrar accederá a una lista de sus carpetas creadas como se muestra en la figura 14. Una vez dentro de la carpeta el usuario tendrá una lista de documentos que podrá descargar seleccionándolos, y también tenemos las opciones de borrar y añadir documento que serán idénticas a las de las carpetas.

Tablón

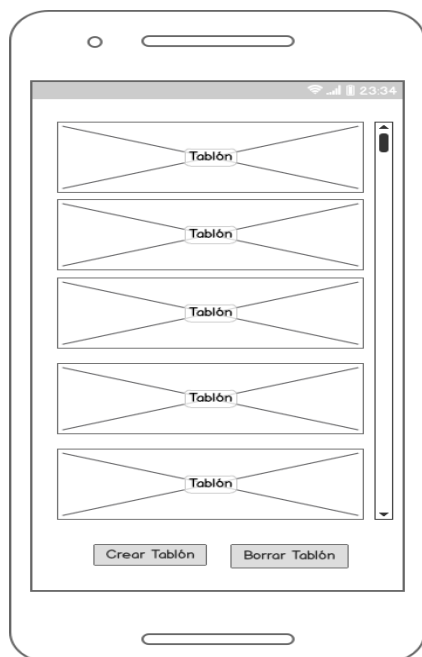


Figura 156: Tablón MU 1

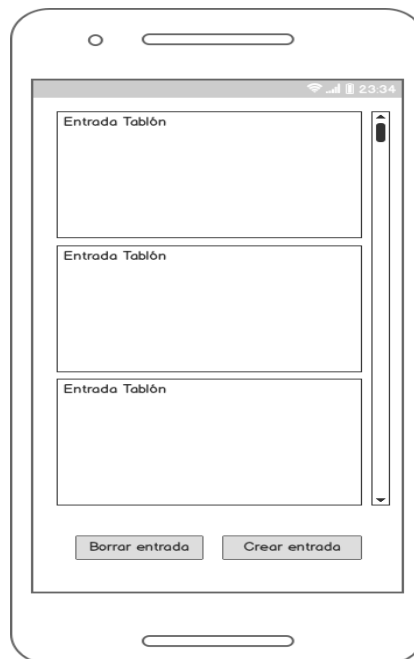


Figura 167: Tablón MU 2

Una vez elegida la asignatura el usuario puede entrar a un tablón ya creado, crear tablón poniendo un nombre de tablón, también puede Borrar tablones, cuando selecciona esta opción le saldrá una lista de los tablonos que ha creado él, una vez entra en un tablón podrá leer las entradas que se encuentren en él y tiene las opciones de crear entrada, donde colgara su entrada al final del tablón.

Podrá borrar entrada donde se le indicará una lista de sus tablonos para que seleccione el que desea eliminar.

Chat

El usuario al entrar en esta opción visualizará dos botones que serán Lista Chats y lista Usuarios

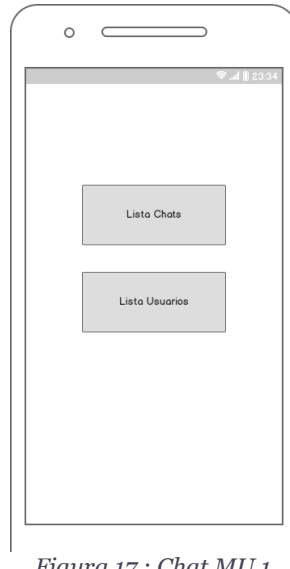


Figura 17 : Chat MU 1

Si el usuario selecciona lista de usuarios se le mostrara los usuarios que corresponda a esa asignatura. Si selecciona cualquiera de estos usuarios, se le enviara una petición para poder crear el nuevo chat entre ellos.

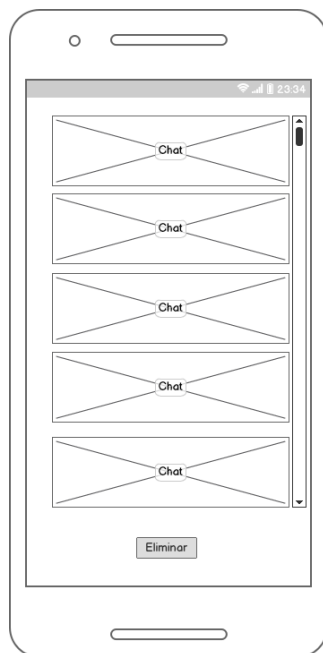


Figura 18 : Chat MU 2

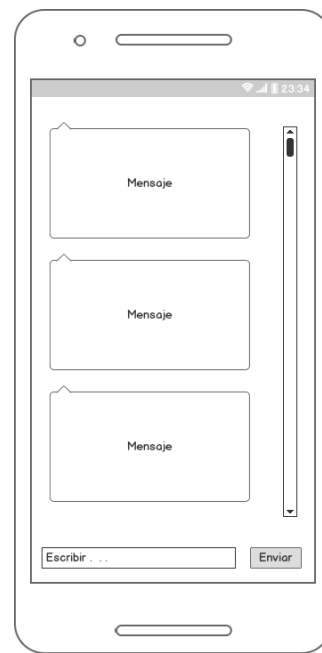


Figura 19 : Chat MU 3

La figura 19 corresponde a seleccionar Lista Chats, dispone de un botón eliminar, cuando se pulse aparecerá la lista de chats y se debe seleccionar uno para poder eliminarlo. Si en vez de pulsar el botón eliminar selecciona se selecciona cualquier chat de ellos entrará en el chat con el otro usuario, se accederá a la figura 20.

Una vez entrado en el chat con el otro usuario se visualizará la figura 20.

Se podrán observar todos los mensajes enviados y recibidos, además en la parte inferior podrás escribir un mensaje al otro usuario y enviárselo pulsando el botón enviar.

6.6 Diagrama de clases

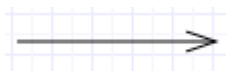
En este apartado se mostrarán las clases del sistema, sus atributos, operaciones y las relaciones entre los objetos de la aplicación.

Las clases están compuestas en tres partes:

- Sección superior: Nombre de la clase.
- Sección Media: Atributos de la clase.
- Sección inferior: Operaciones de la clase (métodos).

Relaciones entre clases a nivel de instancia:

- **Asociación:** Permite asociar objetos que colaboran entre sí, el tiempo de vida de un objeto no depende del otro.



- **Herencia:** la subclase hereda métodos y atributos especificados por una superclase, por lo tanto, la subclase poseerá las características de la superclase más los suyos propios.



- **Agregación:** cuando se requiere modelar objetos que son instancias de clases definidas por el desarrollador de la aplicación.
 - **Por Valor:** el tiempo de vida del objeto incluido está condicionado por el tiempo de vida de que lo incluye (Composición).



- **Por Referencia:** el tiempo de vida del objeto incluido es independiente de que lo incluye (Agregación).



- **Dependencia o Instanciación(uso):** se usa para denotar la dependencia que tiene una clase de otra



El diagrama de clases empleado en la aplicación es el que se muestra en la figura 21.

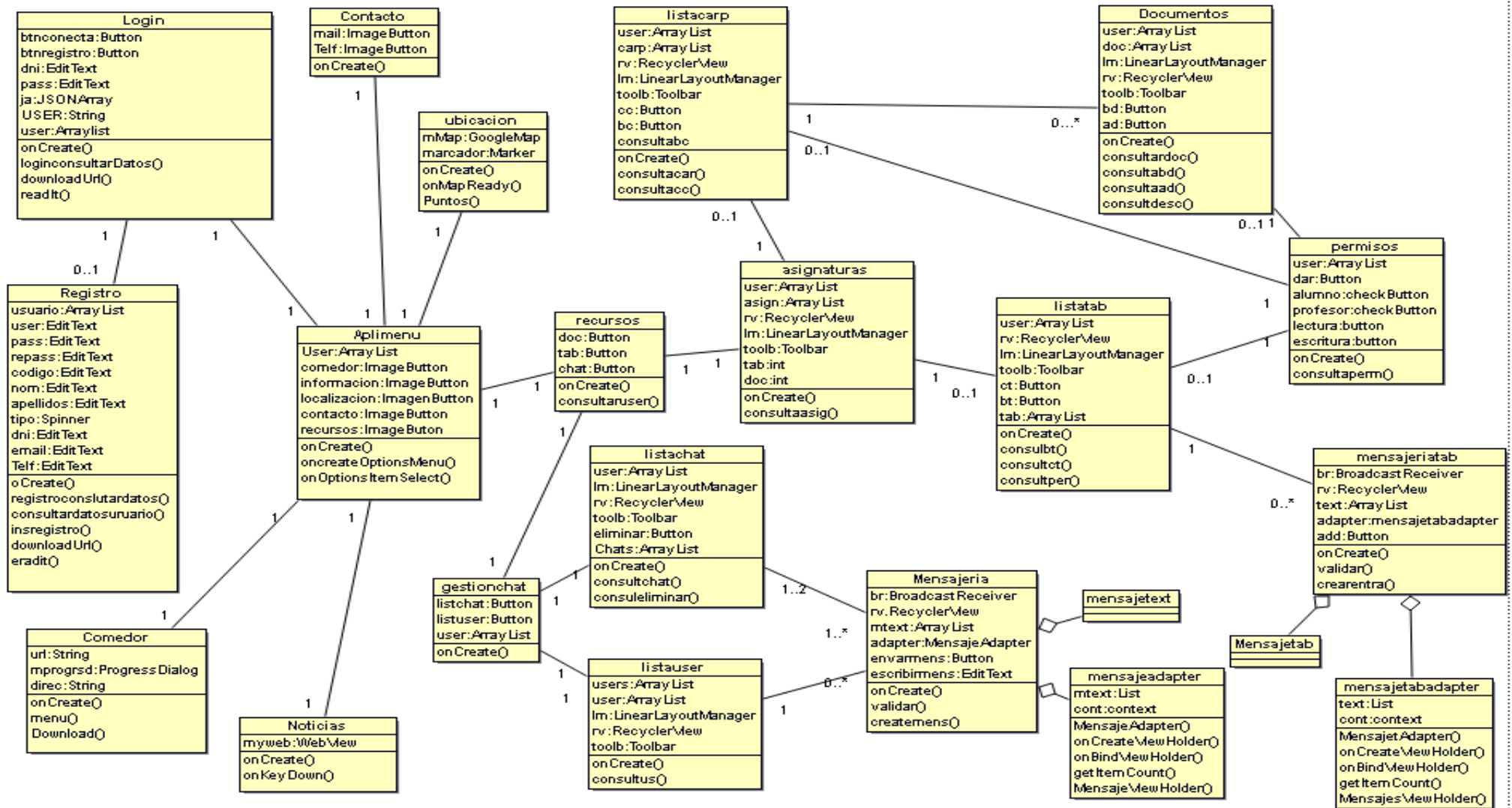


Figura 20: Diagrama de clases

7. Implementación

7.1 Arquitectura de aplicación

7.1.1 Arquitectura hardware

Hardware que se ha utilizado para desarrollar esta aplicación y sus características:

- PC portátil: PC Lenovo con sistema operativo Windows 10, con procesador Intel(R) Core(TM) i7-4702MQ CPU 2.20 GHz, RAM 6.00 GB
- Teléfono móvil: SAMSUNG Galaxy S6 32Gigas de almacenamiento, Procesador Octa-Core, velocidad de CPU 2.1GHz, Versión de Android 7.0

El PC se ha utilizado para el desarrollo de la aplicación y la emulación de un dispositivo Android para la parte de depuración, también como almacenamiento en local del servidor apache y la base de datos.

El dispositivo móvil se ha usado para depurar la aplicación y testear todas sus funcionalidades.

7.1.2 Arquitectura Software

Visión global del funcionamiento del proyecto:



Figura 21 :Arquitectura

La aplicación pide los datos al usuario desde el dispositivo móvil, este dispositivo realiza una llamada a los servicios creados en PHP con los que accede a la Base de Datos, Los archivos y la base de datos están almacenados en el servidor.

También hacemos uso de servicios Google y usamos *firebase* para la creación de notificaciones, con la ayuda de los servicios en PHP conseguimos el token de cada dispositivo para poder identificarnos y dirigir estas notificaciones, este token no se almacena en la base de datos ya que cambia cada vez que inicia la aplicación, también recogemos la hora del servidor y el acceso a la base de datos.

7.2 Base de Datos

Este proyecto utiliza una base de datos relacional MySQL.

La elección de este tipo de base de datos es por las ventajas que esto conlleva:

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece a la normalización por ser más comprensible y aplicable.

La migración a otra máquina no tiene mucha dificultad, es compatible con la mayoría de sistemas operativos y su acceso y su conexión es mejor que el *SQLite* que proporciona Android.

Para la creación de esta base de datos se ha usado la herramienta que proporciona AMPP que es *phpmyAdmin 4.7.0*.

Las tablas utilizadas son las siguientes:

Codigos: Aquí almacenaremos el token del dispositivo con el que se conecte el usuario, cada vez que inicie la aplicación.

Curso: Esta tabla almacena el nombre de los cursos, se usa para poder listar en la aplicación los cursos a los que pertenece cada usuario.

Datos: En ella guardaremos los nombres de las carpetas y los nombres de los documentos, también identificaremos el usuario y a la asignatura que pertenece también podremos ver Los archivos se almacenarán en el servidor.

Permiso: Se almacena los tipos de permisos que hay e un identificador.

Relacionusuariocurso: esta tabla la utilizamos para ver la relación rápidamente entre el usuario y sus cursos.

Claves: En esta tabla el administrador del sistema ingresara los valores correspondientes a los códigos que el genera.

En la aplicación el usuario debe introducir su código para registrarse, si no se encuentra en esta tabla ese código, el usuario no puede registrarse.

El administrador almacenara el curso que le corresponde al usuario.

Tipo_usuario: en esta tabla se almacenan los roles de los usuarios.

Asignatura: se almacenan el nombre de la asignatura y la relación con el centro y el curso al que pertenecen.

Centros: El administrador almacenar todos los centros que tengan la aplicación con los correspondientes campos.

Relacionusuariopermiso: Relaciona los permisos que tiene el usuario con el usuario.

Usuario: aquí almacenaremos todos los datos necesarios para identificar a un usuario

Tablon: en esta tabla nos podemos encontrar la asignatura en la que se encuentra el usuario, los permisos que tiene el usuario, la hora y fecha en la que se ha creado el tablón.

Chat: En esta tabla encontramos los identificadores tanto de emisor como de receptor, la variable hora que se encuentra la fecha y hora de la creación de la sala del chat, una variable *String* identificadora *nomtab*

Para finalizar, procedemos a explicar la creación de tablas dinámicas, que será donde se almacenaran los mensajes tanto del apartado del chat como del tablón.

La creación de tablonos se realizará usando la variable nombre y la variable hora de la tabla **tablon**.

Usando los servicios PHP crearemos una tabla por cada tablón donde tendremos el id del tablón, usuario, variable entrada corresponde al texto que introduce el usuario en el tablón y hora. En este caso el id será la clave principal.

Mientras que, en el caso de la mensajería del chat, usaremos la variable *nomtab* de la tabla chat, la variable hora y el mensaje que será un campo *String* con los mensajes de los usuarios. Se creará una tabla por cada chat creado.

La toma de esta decisión es debido a que a la hora de crear y recibir los mensajes tienes muchas menos restricciones ya que cada conversación corresponderá a una tabla, con esta opción también controlamos a la hora

de eliminar el chat que la conversación del usuario que no lo ha eliminado se borre, si el usuario volviera a crear chat con la otra persona no saldría los mensajes anteriores en su dispositivo ya que se actualizaría la variable hora. mientras que el otro usuario que no elimino el chat conserva todos los mensajes. Esto se podrá gracias a los id del emisor y el receptor.

Esquema relacional

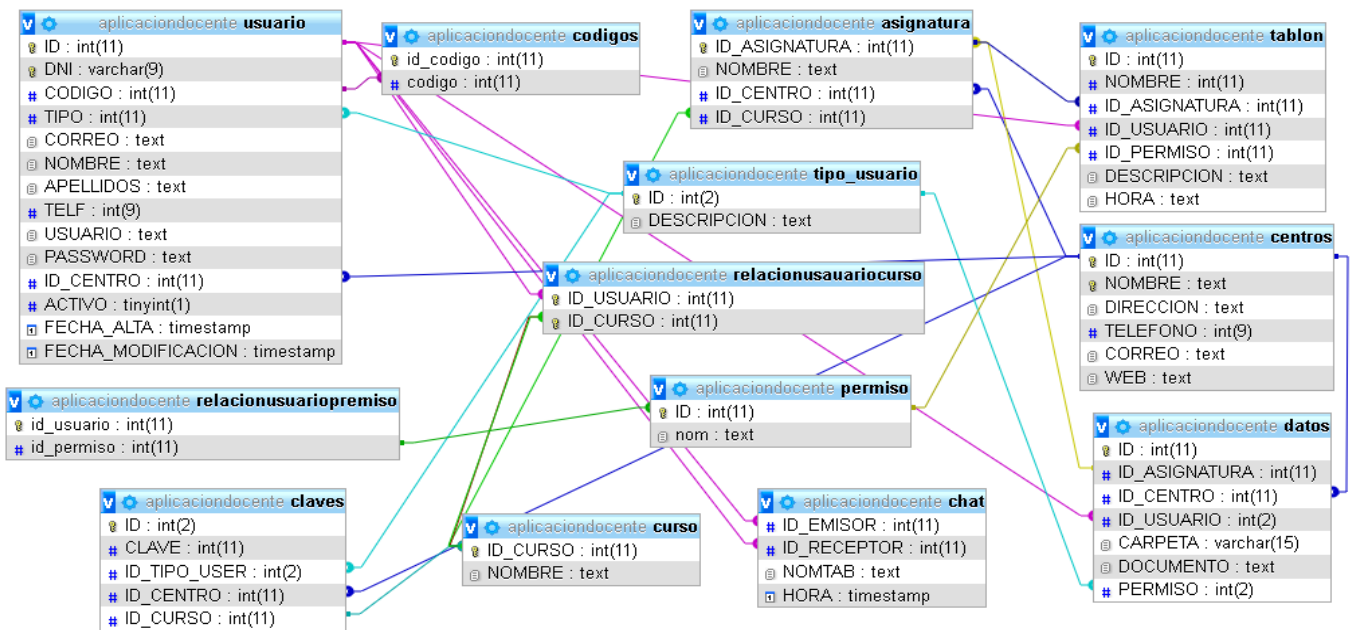


Figura 22 :Relaciones Base de Datos

7.3 Servicios PHP

Estos servicios se han realizado para comunicar la base de datos con la aplicación, la base de datos se encuentra en el servidor, que esta creado en localhost.

Cada clase PHP creada es un método que ofrece un servicio en el cual se llamará a la base de datos para realizar inserciones o consultas dependiendo de las necesidades. No obstante, hay que tener en cuenta el fichero de las conexiones que no ofrece un servicio hacía el exterior, pero si hacía las propias clases de esta capa.

En los siguientes apartados procedemos a exponer los diversos servicios que nos ofrecen la creación de tablas dinámicas en el apartado del chat.



7.3.1 Crear Tabla:

```

1. public static function CreateTable($NameTable) {
2.     try{
3.         $consultar = "CREATE TABLE $NameTable (
4.             id VARCHAR (50) PRIMARY KEY,
5.             mensaje VARCHAR(500) NOT NULL,
6.             usuario VARCHAR(10) NOT NULL,
7.             hora_del_mensaje VARCHAR(50) NOT NULL )";
8.         $respuesta = Database::getInstance()->getDb()-
>prepare($consultar);
9.         $respuesta -> execute(array());
10.            return 200;
11.        }catch(PDOException $e){
12.            return -1;
13.        }
14.    }

```

Con este método creamos las tablas en la base de datos cada vez que creamos un chat nuevo.

7.3.2 Enviar Mensaje:

```

1. public static function EnviarMensaje($TableName, $id, $mensaje,$
hora_del_mensaje){
2.     try{
3.         $consultar = "INSERT INTO $TableName (id,mensaje
, hora_del_mensaje) VALUES(?,?,?)";
4.         $respuesta = Database::getInstance()->getDb()-
>prepare($consultar);
5.         $respuesta-
>execute(array($id,$mensaje,$hora_del_mensaje));
6.         return 200;
7.     }catch(PDOException $e){
8.         return -1;
9.     }
10. }

```

En este método introducimos los valores en la nueva tabla creada.

7.3.3 Enviar Notificación:

```

1. public static function EnviarNotificacion($Mensaje,$hora
,$token,$emisor_del_mensaje){
2.     ignore_user_abort();
3.     ob_start();
4.
5.     $url = 'https://fcm.googleapis.com/fcm/send';
6.
7.     $fields = array('to' => $token,
8.     'data' => array('mensaje' => $Mensaje,'hora' => $hor
a,'cabecera' => $emisor_del_mensaje.' te envio un nuevo
mensaje','cuerpo' => $Mensaje));
9.
10.    define('GOOGLE_API_KEY', 'AIzaSyAPIMgYOW6P1Bzs8
92ZJW4Vtahc-i7LXzY');
11.
12.    $headers = array(
13.        'Authorization:key='.GOOGLE_API_KEY,

```



```

14.         'Content-Type: application/json'
15.     );
16.
17.     $ch = curl_init();
18.     curl_setopt($ch, CURLOPT_URL, $url);
19.     curl_setopt($ch, CURLOPT_POST, true);
20.     curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
21.     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false)
22. ;
23.         curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));
24.
25.     $result = curl_exec($ch);
26.     if($result === false)
27.         die('Curl failed ' . curl_error());
28.     curl_close($ch);
29.     return $result;
30. }

```

En este método hacemos una llamada al Protocolo HTTP de *Firebase Cloud Messaging*, para poder enviar una notificación al usuario si ha recibido algún mensaje y no está en el chat o tablón que la ha recibido.

7.4 Aplicación Android

En esta parte hablaremos de la implementación de la APK Android.

Para esta implementación se ha necesitado que el usuario acepte los permisos que se necesitaban para que funcionaran todas las Necesidades. Se ha establecido que todas actividades tengan siempre una posición vertical.

Permisos

```

<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />

```

CALL_PHONE: Permite que una aplicación inicie una llamada telefónica sin pasar por la interfaz de usuario del marcador.

INTERNET: Permite que las aplicaciones abran conexiones de red.

ACCESS_NETWORK_STATE: Permite que las aplicaciones accedan a información sobre redes.

READ.PHONE_STATE: Permite el acceso de solo lectura al estado del teléfono, incluido el número de teléfono del dispositivo, la información de la red actual.

WRITE: EXTERNAL_STORAGE: Permite a la aplicación acceder al sistema de almacenamiento externo del dispositivo.

READ_GSERVICES: permite a una aplicación utilizar parte de Google *services framework*.

ACCESS_FINE_LOCATION: Permite que una aplicación acceda a la ubicación aproximada.

Breve explicación de cada una de las clases.

7.4.1 Login

Actividad principal de la aplicación, en esta actividad se permite acceder con el DNI y la contraseña o si no estás registrado avanza a la actividad de registrarse. Si ya estamos registrados y completamos los campos, al pinchar en el botón acceder se ejecuta el siguiente *execute*:

```
new
MainActivity.ConsultarDatos().execute("http://192.168.1.104/aplicacion
docente/consulta.php?dni=" +
dni.getText().toString()+"&password="+pass.getText().toString());
```

Donde *ConsultaDatos* es una *AsyncTask<String,void,String>* (clase asíncrona), que lo que hace es comprobar que los datos son correctos con la consulta *url*, si da error la respuesta, se muestra un mensaje al usuario y se le indica el fallo, si los datos son correcto la respuesta a esta consulta la metemos en un Array JSON.

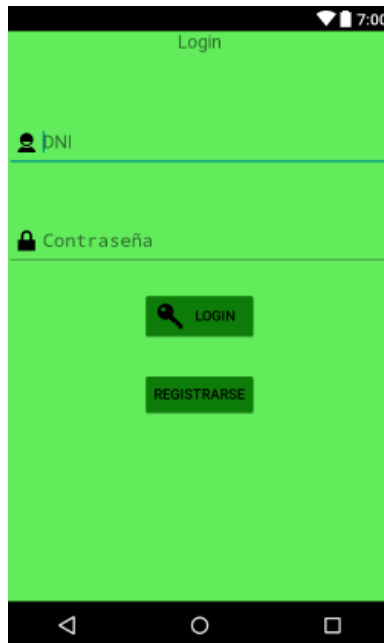


Figura 23: Login

7.4.2 Registro

En esta actividad el usuario rellena un formulario, y cuando le da a registrarse él se vuelve a ejecutar otro *execute* con los parámetros necesarios para registrar, si todo es correcto se vuelve a la actividad *Login*, si no lo es se muestra el error por pantalla.

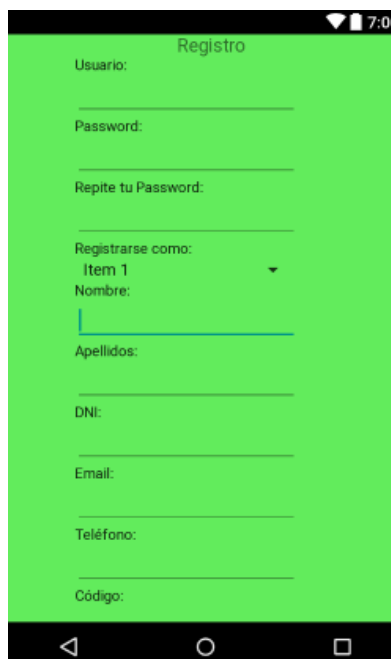


Figura 24: Registro

7.4.3 Aplimenu

En esta actividad se muestra una interfaz con los *imagen buttons* correspondientes y cada uno de ellos te lleva una funcionalidad de la aplicación, este menú se hizo según la reunión que se tuvo con el cliente. Desde el puedes acceder a 5 clases.



Figura 25: Menú

7.4.4 Ubicación

Para el desarrollo de esta actividad hacemos uso de las librerías correspondiente para el acceso a *Google maps*, le añadimos las coordenadas del centro y movemos la cámara para centra la ubicación.

```
final LatLng punto1 = new LatLng(39.5697838,-0.3286616);
mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

mMap.addMarker(new
MarkerOptions().position(punto1).title("Colegio").icon(BitmapDescripto
rFactory.fromResource(R.drawable.marcadorcole)).snippet("CEIP San
Juan Evangelista").alpha(0.8f));
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(punto1,15));
```

7.4.5 Contacto

El usuario dispone de dos botones, con uno accedes al correo y cargamos la dirección de correo del centro. El otro botón es para poder realizar una llamada al centro, sale de la aplicación y llama al centro.

```
llamada.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
```

```
Intent intent = new Intent(Intent.ACTION_CALL);
intent.setData(Uri.parse("tel:666-666-666"));
```

7.4.6 Noticias

En esta actividad cargamos un *WebView* con la dirección del centro.

```
myWebView.setWebViewClient(new WebViewClient());
myWebView.loadUrl("http://mestreacasa.gva.es/web/sanjuanevangelista/1");
```

7.4.7 Comedor

El usuario accede a un *pdf* que se descarga de un enlace de la web que va cambiando mensualmente en el caso del cliente. Y lo muestra en la aplicación

7.4.8 Recursos

En esta actividad se visualizan 3 botones, cada uno de ellos lleva a una actividad, asignaturas, gestiona chat.

7.4.9 Gestionchat

Con esta actividad podemos acceder a dos botones *listachat* *listauser*, también tenemos todos los datos de nuestro usuario guardados en un Array

7.4.10 Listachat

Cargamos los chats que se han creado en nuestro usuario desde la base de datos con *execute* y cada una de las salas sale como un botón al que se puede acceder.

7.4.11 Listauser

Cargamos la lista de usuarios que pertenecen a nuestro curso y aun no tenemos chat con ellos con una consulta, y aparecerán en la interfaz como botones. Si se pincha en alguno de ellos se creará un chat y accederemos a la actividad de *Mensajería*.

7.4.12 Mensajería

Esta clase tiene más importancia, ya que es la sala de chat. Para el desarrollo de esta la interfaz de esta actividad se han creados imágenes '9.png', mapas de bits de tamaño modificable para la creación de estas imágenes *Android studio*, debes acceder al *bat draw9.patch* que se suele encontrar en esta dirección:

```
C:\Users\Usuario\AppData\Local\Android\sdk\tools
```

Con esta aplicación cargas la imagen con el fondo transparente y puedes seleccionar la parte de la imagen que sea modificable. Crear una imagen de emisor de mensaje y otra de receptor para poder diferenciar los mensajes de

diferentes usuarios. Se han creado dos interfaces un *layout_Card_view Mensajes* y un *layout* principal de mensajes:

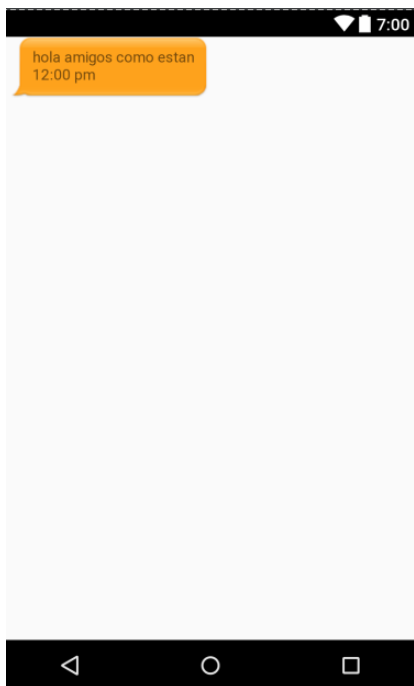


Figura 24: LayoutCardviewmensajes

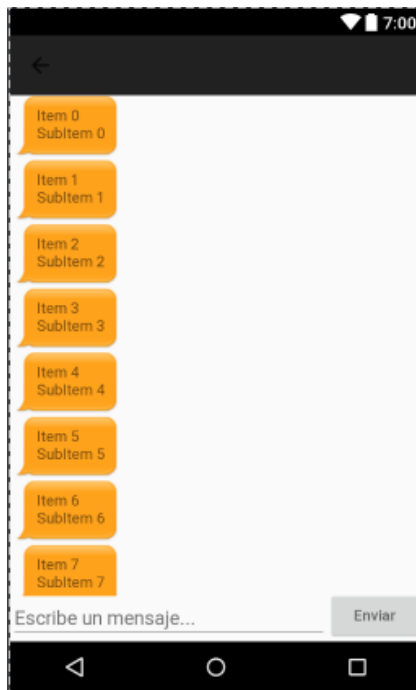


Figura 26: layoutMensaje

Para cambiar las imágenes de los mensajes se crea la clase *MensajesAdapter*

Utilizamos esta parte de código para cambiar las imágenes de los mensajes

```

if (mtext.get(position).getTipoMensaje() == 1) { //EMISOR
    holder.mensajeBG.setBackgroundResource(R.drawable.in_message_bg);
    rl.addRule(RelativeLayout.ALIGN_PARENT_LEFT, 0);
    rl.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
    llMensaje.gravity = Gravity.RIGHT;
    llHora.gravity = Gravity.RIGHT;
    fl.gravity = Gravity.RIGHT;
    holder.TvMensaje.setGravity(Gravity.RIGHT);
} else if (mtext.get(position).getTipoMensaje() == 2) { //RECEPTOR
    holder.mensajeBG.setBackgroundResource(R.drawable.out_message_bg);
    rl.addRule(0, RelativeLayout.ALIGN_PARENT_RIGHT);
    rl.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
    llMensaje.gravity = Gravity.LEFT;
    llHora.gravity = Gravity.LEFT;
    fl.gravity = Gravity.LEFT;
    holder.TvMensaje.setGravity(Gravity.LEFT);
}
    
```

finalmente, cada vez que enviamos debemos de enviar una notificación para que el otro usuario reciba el aviso de que se le ha escrito, para ello tenemos las clases de servicios de *FireBaseId* y *FireBaseServiceMensaje*.

FirestoreID:

```
public class fireBaseId extends FirebaseInstanceIdService {
    @Override
    public void onTokenRefresh() {
        super.onTokenRefresh();
    }
}
```

El servicio que crea la notificación es *FireBaseServiceMensaje*.

```
@Override
public void onMessageReceived(RemoteMessage remoteMessage) {
    super.onMessageReceived(remoteMessage);
    String type = remoteMessage.getData().get("type");
    String cabecera = remoteMessage.getData().get("cabecera");
    String cuerpo = remoteMessage.getData().get("cuerpo");
    switch (type) {
        case "mensaje":
            String mensaje = remoteMessage.getData().get("mensaje");
            String hora = remoteMessage.getData().get("hora");
            String receptor = remoteMessage.getData().get("receptor");
            String emisorPHP = remoteMessage.getData().get("emisor");
            //String emisor =
            Preferences.obtenerPreferenceString(this,
            Preferences.PREFERENCE_USUARIO_LOGIN);
            if (emisor.equals(receptor)) {
                Mensaje(mensaje, hora, emisorPHP);
                showNotification(cabecera, cuerpo);
            }
    }
}
```

A la hora de enviar mensajes para que siempre te muestre el último mensaje que recibe y actualice la interfaz sola, además de evitar que cuando escribas texto en el *editText* y sobrepases su tamaño no te tape la pantalla del chat usamos estos dos comandos:

```
setScrollbarChat();
```

```
LinearLayoutManager lm=new LinearLayoutManager(this);
lm.setStackFromEnd(true);
```

7.4.13 Asignatura

La actividad ejecuta un *execute* y carga de la base de datos todas las asignaturas que correspondientes al usuario y se carga en la interfaz aparecen cada asignatura como un botón. Con esta clase puedes invocar a dos actividades, *Listacap* y *listatab*

7.4.14 Listarcarp

En esta actividad se lanza un *execute* que carga las carpetas que corresponden a cada usuario dependiendo de sus permisos, cada carpeta que muestra la interfaz es un botón que te lleva a la actividad que te mostrara todas las carpetas que se encuentran en ella.

Otra funcionalidad es la de crear una carpeta, si accedes a ese botón accederás al menú de permisos donde tendrás la opción de a cada rol de usuario ponerle los permisos que tiene sobre esa carpeta

7.4.15 Docuementos

Se carga de la base de datos con un *execute* todos los archivos que tiene la carpeta seleccionada según los permisos que tenga el usuario, también tenemos la opción de borrar documentos y de añadir documentos, solo podrá borrar los documentos subidos por el mismo usuario, para añadir documentos también deberá dar permisos a cada usuario sobre el documento.

7.4.16 Listatab

Se recogen los tablonos de la base de datos que correspondan a la asignatura seleccionada y se cargan todos aquellos que el usuario pueda visualizar dependiendo de los permisos que se tengan, a los que podrá acceder y con ello cargara la actividad *tablon*. En esta actividad el usuario podrá borra los tablonos que haya creado él, también podrá crear tablonos y acceder a la actividad permisos donde deberá selecciona los permisos de cada usuario sobre el tablón.

7.4.17 Permisos

El usuario accederá a esta actividad una vez haya creado una carpeta, un documento o un tablón, el usuario podrá seleccionar el rol y darle los permisos sobre lo que acaba de crear.

7.4.18 Mensajeriatab

Esta actividad es muy parecida a la mensajería del chat ya que también usa el servicio de *firebase*, la diferencia más grande es que en esta actividad puedes eliminar los tablonos que has creado el usuario.

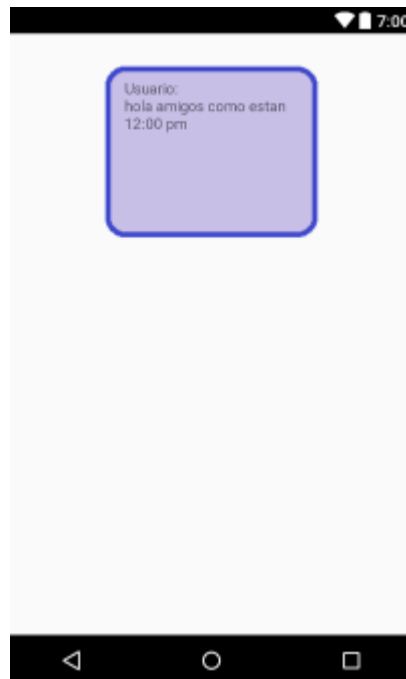


Figura 27: CardViewTablon

6. Presupuesto

Para este proyecto se ha simulado un servidor Apache en local, pero para conseguir todas las funcionalidades en todos los dispositivos se necesita un servidor donde almacenar la base de datos con el mantenimiento que esto conlleva.

Para la realización de este presupuesto configuraremos un servidor dedicado.

| Servidor | Precio |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Dell PowerEdge R200, 2GB RAM, 250GB, SATA, 1 CPU, no RAID, Intel Quad Core, X3350, 2,66/2,83GHz 64bits, firewall Hasta 10 Reglas y 5 cambios (12 meses), BACKUPS NFS 10 GB (12 MESES) | 1.009 € |

Desglosado:

- firewall Hasta 10 Reglas y 5 cambios (12 meses) = 250€
- BACKUPS NFS 10 GB (12 MESES) = 24€

Para calcular el tiempo dedicado al proyecto aproximadamente han sido unas 700 horas, teniendo en cuenta que el mayor tiempo me lo he pasado investigando y aprendiendo a realizar gran parte del proyecto, suponiendo que se cobrase 8€ la hora

El primer pago sería 6.609€ incluye el primer año de mantenimiento, y el mantenimiento del servidor 274€ al año

Sería muy básico, depende de las funcionalidades y mantenimiento y mejoras podría aumentar dicho precio.

Si el centro tuviera un servidor propio se podría alojar el servidor Apache y el precio sería 5.600€

Este presupuesto depende mucho de las dimensiones del centro y los recursos que dispongan.

7. Conclusiones

7.1 Valoración Personal.

En este TFG he podido aprender muchas cosas de la programación en Android que no sabía, he asentado conocimientos en PHP que no he visto en la carrera además de haber podido trabajar en un caso real para un colegio.

He tenido muchas reuniones con ellos, también ha habido bastantes problemas para el desarrollo de la conexión entre el servidor y la app ya que todas las aplicaciones que había desarrollado en Android antes del proyecto usaban base de datos local *MySQLite*, también he tenido problemas con el cliente, ya que hasta que tuvieron claro lo que necesitaban no pude saber las dimensiones del proyecto que ellos querían, intente contactar con Conselleria para ver qué tipos de servicios tenían en la web de su colegio , pero no colaboraron mucho , así que en una última reunión con el director del colegio David Ruiz Obón pudimos llegar a un acuerdo de las cosas que sí que tenían claras que querían.

En cuanto a Android me ha gustado todas las librerías y los servicios de los que dispone Google para la implementación de sus aplicaciones. No conocía tampoco el Protocolo HTTP de *Firebase Clud Messaging* y la verdad que tiene muchos más servicios que no he utilizado. Me he apoyado mucho en la Documentación de la API de Android.

7.2 Futuras ampliaciones o mejoras.

En cuanto a las ampliaciones y mejoras, se pueden añadir muchas más funcionalidades, el chat se puede mejorar añadiendo conversaciones entre grupos, también se pueden ampliar funcionalidades personalizadas dependiendo del cliente.

Hay más formas de hacer notificaciones que no sean utilizando **firebase**, se puede mejorar considerablemente este apartado, en cuanto a la parte de la web, una aplicación web habría mejorado considerablemente la aplicación.

8. Bibliografía

Páginas Web Consultadas año 2017

Universidad Politécnica de Valencia, Master en desarrollo de aplicaciones Android

<http://www.androidcurso.com/index.php/146>

Isabel V. Android PIT Android para principiantes

<https://www.androidpit.es/sdk-android>

Miguel Ángel Godoy Soro ,2014-2015 Diseño de una aplicación Android para la recogida de datos de tráfico y su comunicación a un servidor.

<https://riunet.upv.es/bitstream/handle/10251/53940/Memoria.pdf;sequence=2>

Universidad Politécnica de Valencia, Blog Historia de la informática

<https://histinf.blogs.upv.es/2012/12/14/android/>

Marcos Perelló Soto ,2013-2014 Desarrollo de una aplicación móvil Android para mejorar la integración de los estudiantes de intercambio en la UPV mediante el uso de herramientas útiles.

<https://riunet.upv.es/bitstream/handle/10251/48386/PERELL%C3%93%20-%20Desarrollo%20de%20una%20aplicaci%C3%B3n%20m%C3%B3vil%20android%20para%20mejorar%20la%20integraci%C3%B3n%20%20%20de%20los%20estudia....pdf?sequence=1>

Cristian López Espejo,2013-2014 Desarrollo de una aplicación móvil Android para mejorar la integración de los estudiantes de intercambio en el aspecto de movilidad mediante el uso de mapas estáticos Google Maps.

<https://riunet.upv.es/bitstream/handle/10251/48403/L%C3%93PEZ%20-%20Desarrollo%20de%20una%20aplicaci%C3%B3n%20m%C3%B3vil%20android%20para%20mejorar%20la%20integraci%C3%B3n%20%20%20de%20los%20estudiant...pdf?sequence=1>

Javier Colomer Barberá,2015 Aplicación Android para reproducción de audio en la plataforma arduino

https://riunet.upv.es/bitstream/handle/10251/57432/memoria_TFG_Javier_Colomer_Barber%C3%A1.pdf?sequence=1

Universidad Carlos III de Madrid, Software de comunicaciones

<https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>

Fecha Consulta:2017

Jose María Acuña Morgado, Blog sobre nuevas tecnologías

<http://jmacuna73.blogspot.com.es/2017/03/sistemas-operativos-moviles.html>

Netmarketshare , Operating System Market Share

<https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0>

Sergio, Alemany Ibor, 2014-2015 Desarrollo de una aplicación para la consulta de consumo eléctrico a través de dispositivos móviles

https://riunet.upv.es/bitstream/handle/10251/53857/TFG%20Sergio%20Alemany_1436217728813584524600602947244.pdf?sequence=3

Statista, Sistemas operativos

<https://es.statista.com/grafico/8787/android-es-ya-el-sistema-operativo-mas-usado-del-mundo/>

Google, Firebase

<https://firebase.google.com/>

Laura Molina, Universidad de Chile , Aplicaciones de n capas, Septiembre 2017

<https://users.dcc.uchile.cl/~psalinas/uml/modelo.html><https://laurmolina7821.wordpress.com/1-1-3-aplicaciones-de-2-3-y-n-capas/>

Android developer

<https://developer.android.com/>

Dinahosting, Servidor dedicado

<https://dinahosting.com/dedicados/configurar/maquina/4>

Video2brain , Aprende Firebase

<https://www.video2brain.com/es/cursos/aprende-firebase>