

Gestión del comedor escolar

Nombre Estudiante: Rubén Peñalver Martínez

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: Albert Mata Guerra

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

Fecha de Entrega: 03/01/2018



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Gestión del comedor escolar</i>
Nombre de instalación:	<i>Perfil Escolar</i>
Nombre del autor:	<i>Rubén Peñalver Martínez</i>
Nombre del consultor/a:	Albert Mata Guerra
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2017
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Comedor, colegio y alumno.</i>

Resumen del Trabajo:

Se va a realizar una aplicación para gestionar el servicio de comedor de los centros escolares. La idea es orientar este trabajo a los centros educativos de primaria, ya que el alumnado se encuentra entre los 6 y 12 años, siendo ésta una edad difícil y de gran preocupación en el ámbito alimenticio.

Por lo tanto, se quiere implicar a las dos partes (padres y centros), en un sistema de gestión de comidas. Por un lado, diariamente los centros ofrecerán diferentes alternativas de menús, por supuesto considerados saludables. Por otro lado, los padres votarán qué menú es el preferido para que su hijo se sienta cómodo en el centro.

Otra parte del sistema será valorar la calidad del servicio de comedor por parte de los padres. Al final de cada día, se abrirá una franja horaria, donde los padres podrán valorar el menú de ese día.

El desarrollo completo de esta aplicación se ha abordado en cuatro fases:

1. Análisis y elección del tema.
2. Metodología y especificación.
3. Especificación de los casos de usos, prototipado y elección de la arquitectura.
4. Implementación.

La aplicación va a ser desarrollada para la plataforma Android, ya que se considera que

se llegará a la mayor parte de los usuarios. Teniendo en cuenta, que es una aplicación familiar y algún miembro de la familia (madre o padre), puede tener un smartphone que cumpla estas condiciones.

Abstract:

We are going to create an application to manage the school cafeteria service at all schools. The main aim is to guide this project to primary schools mainly, because children are between 6 and 12 years old and this is a difficult age for good nutritious habits and families are worried.

That is why we would like to involve both, families and schools in a food management application system. On one hand, schools will offer every day different meals and menus options, obviously all of them will be healthy. On the other hand, parents will vote what is the best menu option for his or her child so that his or her child can feel comfortable at school.

Another part of the system will value the quality of the dining service, parents will decide this part. At the end of each day, a schedule will be opened, where parents could vote and value the menu of that day.

The complete development of this application has been dealt with in four phases:

1. Analysis and choice of issue.
2. Methodology and specification.
3. Specification of the cases of uses, prototyped and architecture decision.
4. Implementation.

The application will be developed for Android platform, because we consider that we will reach at most of the application users. We have also considered that, this is a familiar application and any of the members of the family (mother or father) can easily have a smartphone that can fulfill the requirements.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo	2
1.3 Especificaciones funcionales	3
1.4 Enfoque y método seguido	6
1.5 Planificación del Trabajo	9
1.6 Revisión de la planificación	16
1.7 Viabilidad económica	17
1.8 Breve resumen de productos obtenidos.....	18
1.9 Breve descripción de los otros capítulos de la memoria	19
2. Usuarios y contextos de uso	21
2.1 Usuarios principales	21
2.2 Contextos de uso: centro escolar.....	21
2.3 Contextos de uso: alumno	23
2.4 Accesibilidad.....	24
2.5 Escenarios.....	25
3. Diseño conceptual	29
3.1 Árbol de navegación.....	29
3.2 Análisis comparativo	31
4. Prototipado.....	35
5. Definición de los casos de uso	37
5.1 Diagrama de casos de uso.....	37
5.2 Casos de uso	39
6. Diseño de la arquitectura	45
6.1 Diagrama de clases.....	45
6.2 Estructura de base de datos.....	46
6.3 Revisión del modelo de base de datos y diagrama de clases	47

6.4 Arquitectura MVC.....	52
7. Implementación.....	53
7.1 Configuración gradle	53
7.2 Configuración manifest	54
7.3 Firebase	55
7.3.1 ¿Por qué Firebase?.....	55
7.3.2 Firebase en la aplicación	58
7.3.3 Opciones.....	61
7.4 Decisiones generales	62
8. Información y pruebas.....	66
9. Vías futuras	68
10. Conclusiones	69
11. Glosario.....	71
12. Bibliografía.....	73
13. Anexos	75
13. 1. Anexo 1. Manual de usuario	75
13. 2. Anexo 2. Manual de instalación	82

Lista de figuras

Ilustración 1. Comparativa de los sistemas operativos

Ilustración 2. Home Aceproject

Ilustración 3. Desglose de tareas PEC 1

Ilustración 4. Diagrama de Gantt de las tareas PEC 1

Ilustración 5. Detalle tarea en el diagrama de Gantt

Ilustración 6. Desglose tareas PEC 2 y 3

Ilustración 7. Diagrama de Gantt desarrollo completo

Ilustración 8. Resumen detallado de las tareas de PEC 2

Ilustración 9. Resumen detallado de las tareas de PEC 3

Ilustración 10. Resumen detallado de las tareas de la entrega final

Ilustración 11. Resumen detallado de las tareas de la entrega final – Calendario

Ilustración 12. Grafo de los commits en el repositorio de GitLab

Ilustración 13. Plan Aceproject

Ilustración 14. Plan Firebase

Ilustración 15. Plan GitLab

Ilustración 16. Árbol de navegación

Ilustración 17. Interfaz Serunion Educa 1

Ilustración 18. Interfaz Serunion Educa 2

Ilustración 19. Interfaz Serunion Educa 3

Ilustración 20. Interfaz Comedor Escolar

Ilustración 21. Pantallas login y perfil de alumno

Ilustración 22. Pantallas menú del centro escolar y registro de comida

Ilustración 23. Pantallas calendario y listado de alumnos

Ilustración 24. Pantallas votación y valoración de un menú

Ilustración 25. Diagrama de casos de uso del centro escolar

Ilustración 26. Diagrama de casos de uso del alumno

Ilustración 27. Diagrama de clases

Ilustración 28. Diagrama de base de datos

Ilustración 29. Diagrama de clases revisado

Ilustración 30. Diagrama arquitectura MVC

Ilustración 31. Configuración Gradle

Ilustración 32. Configuración de la aplicación Manifest.xml

Ilustración 33. Tabla de precios de Google Cloud

Ilustración 34. Tabla de precios de Azure

Ilustración 35. Tabla de precios de AWS

Ilustración 36. Tabla de precios de Firebase

Ilustración 37. Proveedores de autenticación Firebase

Ilustración 38. Estadística de usuarios y fallos

Ilustración 39. Control de usuarios registrados

Ilustración 40. Tamaño aplicación apk

Ilustración 41. Información aplicación instalada

Ilustración 42. Información Emulador

Ilustración 43. Información dispositivo real

1. Introducción

1.1 Contexto y justificación del Trabajo

En la actualidad, existe una gran cantidad de enfermedades. Es por ello, que en nuestra vida nos encontramos con numerosos problemas de salud, ya sea porque las sufrimos nosotros mismos o porque las sufren los que nos rodean.

En ocasiones, las enfermedades parecen ser que te eligen a ti, como pueden ser las de nacimiento. En estos casos, parece inevitable que una persona deje de convivir con ella durante el resto de su vida, ¿lo podemos llamar destino?

El problema, llega cuando somos nosotros los que ayudamos a coger otro tipo de enfermedades, las cuales son totalmente evitables y, por lo tanto, somos los culpables directos. Una de estas enfermedades y en la que nos vamos a centrar para el desarrollo de nuestra aplicación, es la **obesidad infantil**. Es uno de los mayores problemas actuales de salud, donde parece que las personas hacemos pocos esfuerzos para no llegar a ciertos niveles de riesgo y hacemos caso omiso a las recomendaciones de los expertos.

Además, esta enfermedad es claramente conductora a otro tipo de enfermedades que se pueden catalogar como más peligrosas. Un estudio reciente, indica que la obesidad puede traer consigo hasta 11 tipos diferentes de cáncer [1].

Existen casos donde la culpa no es de la persona que padece la enfermedad. Nos referimos a los niños, donde parece que ellos indirectamente sufren la irresponsabilidad de los padres. Como podemos leer en un artículo, en Madrid, alrededor de un 40% de los niños entre 3 y 12 años padecen problemas de obesidad o sobrepeso [2].

Nosotros vamos a intentar poner de nuestra parte, para ayudar a intentar solucionar algunos de los problemas, proporcionando un sistema de colaboración en el servicio de comedor que se ofrece en los centros escolares. Ciertos centros pueden llegar a tener conflictos con los padres, por los menús que ofrecen a sus hijos.

Para el desarrollo de esta aplicación, nos hemos centrado en los problemas analizados. Queremos hacer partícipes a los padres de los alumnos, a la elección y valoración de los distintos menús que se ofrecen diariamente.

Tras realizar una pequeña investigación sobre productos similares nos hemos encontrado con dos aplicaciones interesantes, pero que no llegan a ofrecer lo que nosotros deseamos. Las analizamos:

- **Serunion Educa**. Esta aplicación está desarrollada con unas ideas parecidas, pero tan solo llega a ofrecer los menús diarios, con respecto a lo que nosotros queremos conseguir. Es puramente informativa, proporcionando otro tipo de información, como las actividades extraescolares de los niños. Nosotros queremos centrarnos en el

comedor y en la implicación de los padres de los niños. Una vez se haya conseguido nuestro objetivo, podremos fijar nuevas metas.

- **Comedor Escolar.** Una aplicación orientada a las cafeterías escolares, por lo que diría más bien, a cafeterías universitarias. Se parece más a una carta online, con todo lo que se ofrece. Por este motivo, queda totalmente descartada como una buena referencia.

Después del análisis realizado, creemos que es una buena oportunidad para desarrollar una aplicación que resulte interesante y pueda enganchar e implicar a los padres de los niños. Los padres, podrán consultar el menú diario que ofrezca el centro. Queremos que los centros sometan a votación el menú, entre distintas alternativas que serán controladas y publicadas por el servicio de cocina. De este modo, la intención es involucrar a los centros escolares y a los padres de los alumnos, en el desarrollo de los niños mediante una dieta sana y equilibrada. Al mismo tiempo, podremos saber la calidad del servicio de comedor que ofrece cada centro, así como la satisfacción o descontento de los padres, que serán los actores principales de nuestra aplicación.

Pensamos que esta aplicación puede resultar interesante y podrá ser ampliada en un futuro con grandes mejoras.

1.2 Objetivos del Trabajo

A nivel personal, se establecerán una serie de objetivos. Algunos de ellos los intentaremos cumplir durante el transcurso del desarrollo de nuestra aplicación y otros se cumplirán una vez finalizado el mismo.

Para una mayor claridad, enumeramos todos ellos a continuación:

- 1) Capacidad para el desarrollo de una aplicación.
- 2) Profundizar en la plataforma Android e ir mejorando los conocimientos sobre su API.
- 3) Intentar llegar al máximo de nuestros usuarios potenciales.
- 4) Mejorar el funcionamiento del servicio de comedor de los centros escolares.
- 5) Satisfacer a padres/madres/tutores de los alumnos, llevando un mayor control sobre la alimentación del alumno.
- 6) Crear una conexión entre el centro escolar y los responsables de los alumnos, a través de, el servicio del comedor.
- 7) Provocar la participación e implicación de madres/padres/tutores en mantener un servicio de comedor saludable y de calidad.
- 8) Investigar y ahondar en las herramientas ofrecidas por Google, como puede ser en nuestro caso Firebase.

- 9) Facilitar el registro al usuario mediante las credenciales de su cuenta de Google.
- 10) Autonomía para estudiar la API de Google para la extraer las credenciales del usuario.
- 11) Realizar un análisis real, con su correspondiente planificación detallada, que sirva para valorar los tiempos y aprovechar los datos para planificaciones futuras.
- 12) Conocer una herramienta que facilite la gestión de proyectos y hacer uso de ella durante el desarrollo de la aplicación.
- 13) Aprender a seleccionar para cada proyecto cada tipo de herramienta, producto, frameworks... entre todas las alternativas que ofrece el mercado, provocando mejoras y no una ralentización del mismo.
- 14) Mejorar en el uso de herramientas de control de versiones. Tengo experiencia en trabajar con SVN, por lo que, comenzar con GIT es un nuevo reto.
- 15) Añadir nuevos conocimientos e intentar crear una unión entre todo lo aprendido en este máster.
- 16) Trabajar con alguna herramienta de prototipado, comprobando a nivel personal si merece la pena.
- 17) Comprensión y realización de un modelo de datos real y coherente.
- 18) Traspasar ese modelo de datos a un esquema de base de datos, intentando que sea lo más eficiente posible abarcando todo lo que necesitamos.
- 19) Mejorar la capacidad sobre la toma de decisiones.
- 20) Por último, como objetivo principal, esperar un buen resultado generando una satisfacción personal y aprobado final de este máster.

1.3 Especificaciones funcionales

Vamos a exponer una serie de requerimientos funcionales y no funcionales, para el desarrollo de la aplicación, los cuales definirán brevemente lo que esperamos obtener en el producto final.

- Existirán 2 roles claramente identificados en la aplicación, “Centro escolar” y “Alumno”.
- El centro escolar deberá darse de alta, mediante un formulario de registro, donde se solicitarán los distintos datos del centro.

- Por otro lado, el alumno (padre/madre/tutor) deberá darse de alta, mediante un formulario de registro específico. Previamente, el centro escolar debe haber sido dado de alta.
- Alternativamente, el alumno (padre/madre/tutor), podrá registrarse o autenticarse mediante el servicio de autenticación proporcionado por Firebase [3]. Estamos interesados, especialmente, en el acceso con las credenciales de una cuenta de Google.
- El formulario de registro, de la persona responsable del alumno, deberá ofrecer los campos de una forma coherente con los posibles datos que podrán ser extraídos del usuario de Google. De forma que, mediante cualquier forma de registro, todos los campos queden completados.
- Una vez que se haya registrado un responsable, se solicitarán los datos del alumno para crear una relación con el mismo.
- La pantalla principal de la aplicación será una pantalla de login, para poder autenticar al usuario y poder hacer uso de la misma.
- Se proporcionará un sistema de recuperación de contraseña, en caso de olvido. Esta funcionalidad se ofrecerá desde la página principal de login.
- El centro escolar, tendrá un apartado, donde podrá visualizar un listado de todos los alumnos que han sido registrados.
- El centro escolar, podrá ver los datos completos del alumno, seleccionándolo sobre el listado.
- El centro publicará para cada día 2 menús completos, que posteriormente serán sometidos a votación.
- En el registro de los menús se hará uso de subida de imágenes, a través de la galería del smartphone.
- El elemento principal de la aplicación será un “calendario” para poder hacer la mayoría de las acciones sobre cada uno de los días. Se mostrarán los últimos y siguientes días.
- Se ofrecerá funcionalidad anterior y siguiente, en la pantalla “calendario”, mostrando días anteriores o siguientes.
- Los días tendrán diferentes estados:
 - Cerrado: No se han publicado los menús para poder ser votados.
 - Votación: Se puede acceder al día para votar el menú deseado.
 - Disponible: La votación ha finalizado y se puede ver el menú escogido del día.

- Valoración: Se podrá realizar una valoración del menú del día anterior.
- Los responsables de los alumnos podrán votar entre los distintos menús del día, para seleccionar el menú deseado para su hijo.
- Los responsables de los alumnos podrán valorar el menú del día. Se abrirá una franja horaria en la que se podrá realizar una valoración.
- Control del horario, en el que se cambiará de estado un día para poder realizar las distintas operaciones sobre el mismo.
- Los datos serán almacenados en una base de datos externa.
- Para el almacenamiento de los datos, se hará uso de nuevo de la plataforma de Firebase [4]. La base de datos es de tipo noSQL, por lo que los datos se almacenarán en formato JSON.

Por otro lado, la interfaz que se diseñará para soportar la funcionalidad deseada será la siguiente:

- Pantalla para el registro del centro escolar: Desde esta pantalla se dará de alta un centro escolar, teniendo que introducir todos los datos que lo caracteriza.
- Pantalla para el registro de usuario: Aquí, un usuario se dará de alta en la aplicación. Se entiende que el usuario deberá ser padre/madre/tutor de un alumno del centro. Los datos por cumplimentar deberán ser los mismos que serán extraídos de un usuario con cuenta en Google.
- Pantalla de login: Desde esta pantalla se podrán realizar las siguientes acciones:
 - Un usuario con cuenta en Google podrá ser dado de alta utilizando las mismas credenciales. Del mismo modo, se procederá a la autenticación cuando el usuario ya se encuentre en la base de datos.
 - Se podrá acceder a la aplicación tanto con las credenciales de la aplicación, como con las credenciales de la cuenta Google.
 - Se podrá recuperar la contraseña en caso de olvido. Se enviará un correo electrónico al usuario.
- Pantalla para el registro de un alumno: Un usuario que ya ha sido registrado, tendrá que dar de alta, obligatoriamente, al alumno por el que se ha registrado. En los datos personales, deberán introducir una foto del mismo. La foto deberá hacerse en ese mismo instante, ya que saltará la cámara al pinchar en el cuadro de la foto.

- Pantalla para el listado de alumnos: Una funcionalidad para el centro escolar. Tendrá un apartado desde el cual, podrá visualizar cada uno de los usuarios que han sido dados de alta en la aplicación.
- Pantalla para el detalle de un alumno: El centro escolar, podrá acceder al perfil de un alumno desde el listado.
- Pantalla para el registro de los menús: El centro escolar registrará aquí cada uno de los menús que posteriormente serán publicados. Es obligatorio introducir una foto del menú que se va a ofrecer, se abrirá la galería del smartphone. Desde aquí, se indicará los datos del menú y en qué día se quiere registrar.
- Pantalla calendario: Se considera la pantalla principal de la aplicación, ya que desde aquí se podrá acceder al núcleo de nuestro objetivo. Mostrará los días de la semana actual, teniendo cada uno de ellos el estado que le corresponda.
- Pantalla de selección de menú: Se llegará desde la pantalla calendario, si el estado del día seleccionado permite votar menú (estado votación).
- Pantalla para la valoración de un menú: Se accederá a través del calendario, si el estado del día seleccionado lo permite (estado valoración). Se podrá valorar el menú del día, el menú que mayor número de votos haya obtenido.

1.4 Enfoque y método seguido

Como ya pudimos ver, en el análisis que hicimos en el punto 1.1, nos dimos cuenta de que no existe ninguna aplicación que se acerque a lo que nosotros deseamos. Éste, es el principal motivo por lo que hemos optado por el desarrollo de una nueva aplicación.

Se considera, que los padres siempre están preocupados por los hijos. Por este motivo, vemos interesante el desarrollo de esta aplicación. Sobre todo, cuando los niños son pequeños, la alimentación es una de las cosas que más preocupa a los padres. Con esta aplicación, se intentará llevar un control del alumno en el centro escolar.

Como siempre, el objetivo es llegar a la mayor cantidad de usuarios. Por ello, hemos optado por hacer el nuevo desarrollo en la plataforma de Android. Como podemos ver en la siguiente imagen, estudios recientes [5] indican que el sistema operativo Android es el más utilizado en el mercado, con una cuota aproximada del 40%. Incluso, lo importante es que sigue creciendo. Actualmente, ha llegado a superar a Windows, con la temible pregunta de ¿qué será de los ordenadores?

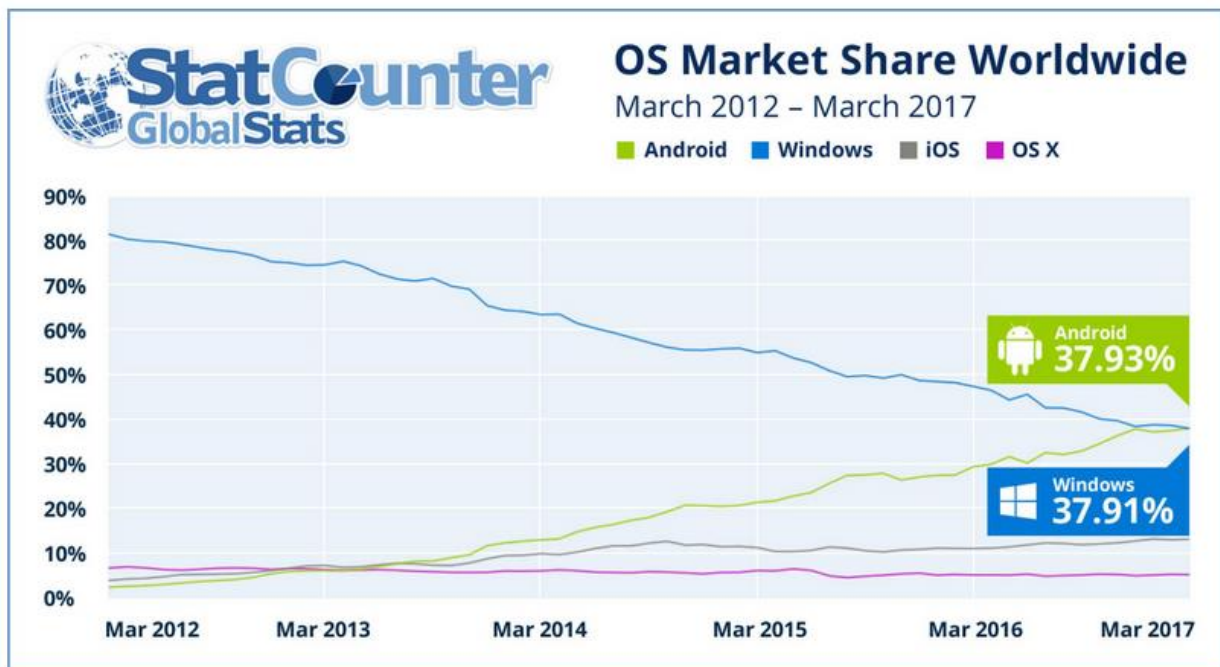


Ilustración 1. Comparativa de los sistemas operativos

Se podría haber optado por un desarrollo multiplataforma, en vez de un desarrollo nativo para la plataforma de Android. La primera justificación, es que se considera que, en un porcentaje muy alto, en la mayoría de las viviendas hay un smartphone con sistema operativo Android. Por otro lado, se tiene la intención de seguir ampliando la funcionalidad de la aplicación y uno nunca sabe si necesitará funcionalidad que solo te permita un desarrollo nativo. Esta última justificación es más por prevención y posibles vías futuras.

Se considera que una buena planificación del trabajo ayuda a tener un mejor desarrollo durante el proyecto. También, este tipo de productos proporciona una mejor apreciación sobre la cantidad de tareas que hemos de realizar y el estado en el que se encuentra cada una de ellas. Por ello, se ha realizado una breve búsqueda sobre diferentes productos de software para una gestión adecuada del proyecto, entre los que podemos destacar se encuentran Zoho, Wrike, Aceproject y Teamwork.

Tras realizar un breve análisis sobre lo que puede aportar cada uno de estos productos, entre las distintas alternativas mencionadas anteriormente, nos hemos decantado por Aceproject. Los motivos principales que han hecho que se tome esta decisión son los siguientes:

- Una interfaz agradable e intuitiva.
- Se centra en lo más importante de un desarrollo, proyecto y tareas.
- Crear tareas, estimarlas, acotarlas mediante fechas y relacionarlas.
- Generar un diagrama de Gantt de todas las tareas.

- Controlar el tiempo que le dedicamos a cada una de ellas mediante un cronómetro que se proporciona.
- Controlar el estado de cada una de las tareas.
- Es gratuito para el uso que le vamos a dar.

Como metodología de trabajo, se hará uso de un sistema de control de versiones, en este caso Git. Por lo que, se irá guardando el proyecto en un repositorio remoto que nos ofrece de forma gratuita la plataforma de GitLab. Ayudará a llevar un control exhaustivo de las modificaciones que se van realizando durante todo el desarrollo.

Es recomendable usar un sistema de control de versiones, ya que, si trabajas en equipo, también queda grabado qué usuario ha realizado la modificación y sobre qué ficheros. En este caso, el proyecto será desarrollado por una sola persona, pero la idea es ir guardando sobre la rama principal cada vez que hemos ampliado con una funcionalidad nueva y el proyecto es estable. Llevar este control, nos permite poder volver atrás, a una versión anterior de un fichero en el caso de que haya existido algún problema o simplemente necesitemos información que haya sido borrada en versiones posteriores.

Por otro lado, debido a que el producto va a ser desarrollado para el sistema operativo Android, el entorno de desarrollo que vamos a utilizar es Android Studio, que es el entorno oficial para esta plataforma. Este entorno, proporciona un emulador, en el que podremos simular nuestra aplicación e ir realizando las pruebas pertinentes. Android Studio se ha terminado afianzando como IDE oficial, debido a la gran evolución que ha tenido esta plataforma y el interés en desarrollar un IDE exclusivamente para ella. Esto ocurrió hace pocos años, ya que hasta entonces el otro IDE por excelencia para el desarrollo era Eclipse.

Antes de mostrar la estimación de las tareas, vamos a dar unos pequeños apuntes sobre la forma de trabajar, con la que se intentará tener una constancia para conseguir el objetivo final. A continuación, mostramos una relación de los días que se tiene pensado trabajar en el proyecto y qué cantidad de horas vamos a dedicar en cada uno de ellos:

- Martes: 4 horas.
- Jueves: 4 horas.
- Viernes: 4 horas.
- Sábado: 6 horas.
- Domingo: 3 horas.

La dedicación de horas a la semana suma un total de 21 horas, que daría una media de 3 horas por días. El proyecto se debe realizar en unos 100 días y se le tiene que dedicar un mínimo de 300 horas, por lo tanto, la media diaria es unas 3 horas. Para mantener esa media semanal, se ha dividido las horas entre los días que se va a poder trabajar sobre el proyecto.

1.5 Planificación del Trabajo

A continuación, vamos a explicar cómo hemos dividido el proyecto en tareas y podremos ver cómo ha quedado tal división, mediante Aceproject [6]. Comenzamos con la página principal, donde podemos ver la información del total de tareas. Observamos que hemos desglosado el proyecto en un total de 37 tareas. Debido a que hemos creado relaciones entre las tareas, nos avisa de que tenemos 1 para comenzar pronto y 7 disponibles.

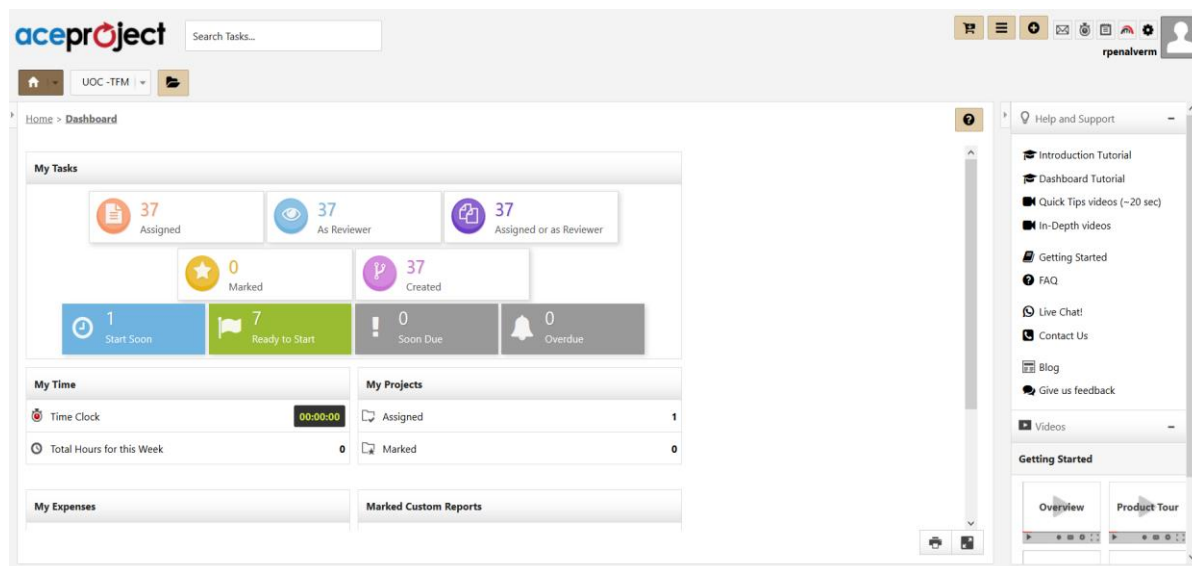


Ilustración 2. Home Aceproject

Por la dificultad, en mostrar una imagen en la que podamos apreciar los detalles de cada una de las tareas, junto al diagrama de Gantt, explicaremos la primera entrega por separado.

PEC 2

Teniendo en cuenta los días y las horas que vamos a emplear en el proyecto, se le ha puesto a cada tarea una fecha de inicio y otra fecha de fin, en base a las horas que le vamos a dedicar a cada una de ellas. La fecha de inicio de la primera tarea de cada entrega y la fecha de fin de la última tarea de cada entrega, coinciden con las fechas de inicio y fin de cada una de las entregas por separado. En la siguiente imagen, podemos ver el desglose de cada una de las tareas que componen la primera entrega.

Task #	Summary	Start Date	End Date	Estimated Hours	Duration
1	PEC 2 - Usuarios y contextos de uso	10/12/2017	10/15/2017	15.00	4 days
2	PEC 2 - Diseño conceptual	10/16/2017	10/17/2017	4.00	2 days
3	PEC 2 - Prototipado	10/18/2017	10/24/2017	20.00	7 days
4	PEC 2 - Evaluación	10/25/2017	10/26/2017	4.00	2 days
5	PEC 2 - Evaluación-Definición de los casos de uso	10/27/2017	10/28/2017	10.00	2 days
6	PEC 2 - Evaluación-Diseño de la arquitectura	10/29/2017	11/1/2017	10.00	4 days

Ilustración 3. Desglose de tareas PEC 2

Posteriormente, podemos observar el diagrama de Gantt que se ha generado sobre las tareas de la primera entrega. Vemos, la relación que hay entre cada una de ellas. En Aceproject, el diagrama aparece a continuación de las tareas, pudiendo identificar cada línea del diagrama con su línea correspondiente de su tarea desglosada. Aquí, lo hemos tenido que mostrar por separado.

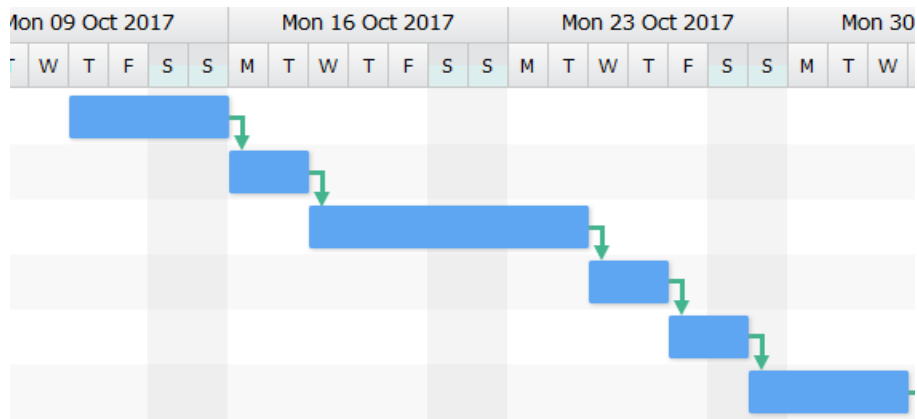


Ilustración 4. Diagrama de Gantt de las tareas PEC 2

También, en el diagrama de Gantt, podemos ver el estado de nuestra tarea. Si pasamos el ratón por encima de alguna de ellas, obtenemos información sobre la misma. Podemos ver un ejemplo sobre una de las tareas de la primera entrega en la siguiente imagen.

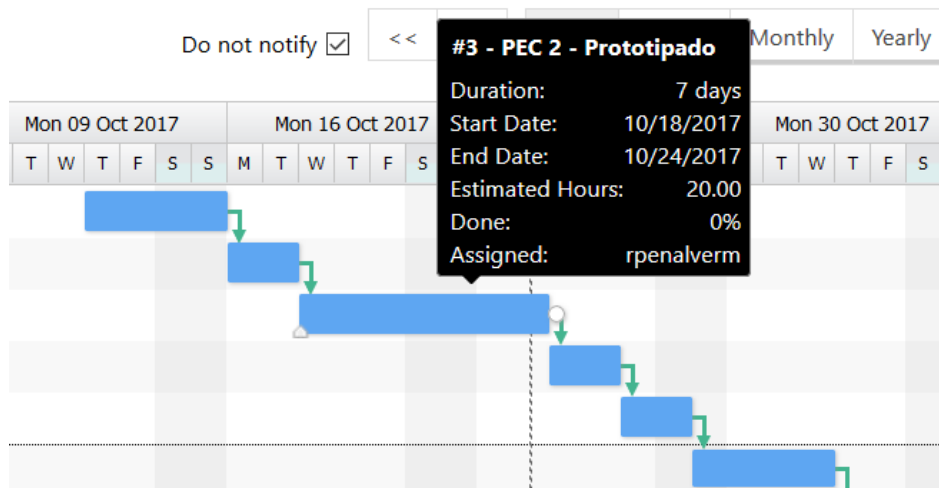


Ilustración 5. Detalle tarea en el diagrama de Gantt

PEC 3 y PEC 4 (entrega final)

Task #	Summary	Start Date	End Date	Estimated Hours	Duration
7	PEC 3 - Formación y documentación Firebase	11/2/2017	12/13/2017	10.00	42 days
8	PEC 3 - Pruebas	11/19/2017	12/13/2017	16.00	25 days
9	PEC 3 - Darse de alta - Centro escolar	11/2/2017	11/2/2017	4.00	1 day
10	PEC 3 - Pantalla registro centro escolar	11/2/2017	11/3/2017	2.00	2 days
11	PEC 3 - Darse de alta - Responsable del alumno	11/4/2017	11/4/2017	4.00	1 day
12	PEC 3 - Pantalla de registro de usuario	11/4/2017	11/4/2017	2.00	1 day
13	PEC 3 - Alta y autenticación con Firebase	11/5/2017	11/7/2017	8.00	3 days
14	PEC 3 - Login	11/8/2017	11/10/2017	6.00	3 days
15	PEC 3 - Recuperación de contraseña	11/11/2017	11/11/2017	6.00	1 day
16	PEC 3 - Pantalla de login	11/5/2017	11/11/2017	3.00	7 days
17	PEC 3 - Registro alumno	11/12/2017	11/14/2017	5.00	3 days
18	PEC 3 - Pantalla de registro de alumno	11/12/2017	11/16/2017	3.00	5 days
19	PEC 3 - Listado de alumnos para el centro escolar	11/17/2017	11/17/2017	5.00	1 day
20	PEC 3 - Pantalla de listado de alumnos	11/17/2017	11/18/2017	6.00	2 days
21	PEC 3 - Detalle del alumno	11/19/2017	11/19/2017	2.00	1 day
22	PEC 3 - Pantalla de detalle de alumno	11/19/2017	11/19/2017	2.00	1 day
23	PEC 3 - Registro de los menús	11/20/2017	11/21/2017	4.00	2 days
24	PEC 3 - Pantalla de registro de menús	11/20/2017	11/23/2017	4.00	4 days
25	PEC 3 - Calendario	11/24/2017	11/24/2017	4.00	1 day
26	PEC 3 - Función anterior y siguiente del calendario	11/25/2017	11/25/2017	4.00	1 day
27	PEC 3 - Pantalla calendario	11/24/2017	11/26/2017	6.00	3 days
28	PEC 3 - Votación	11/27/2017	11/28/2017	3.00	2 days
29	PEC 3 - Control de horario	11/27/2017	11/28/2017	4.00	2 days
30	PEC 3 - Pantalla de selección de menús	11/27/2017	12/1/2017	4.00	5 days
31	PEC 3 - Pantalla detalle comida	12/2/2017	12/2/2017	2.00	1 day
32	PEC 3 - Valoración	12/3/2017	12/4/2017	4.00	2 days
33	PEC 3 - Pantalla valoración	12/3/2017	12/5/2017	3.00	3 days
34	PEC 4 - Memoria del TFM	12/14/2017	12/23/2017	35.00	10 days
35	PEC 4 - Anexo 1: Manual de usuario	12/24/2017	12/29/2017	15.00	6 days
36	PEC 4 - Anexo 2: Manual de instalación	12/30/2017	12/30/2017	5.00	1 day
37	PEC 4 - Vídeo de presentación	12/31/2017	1/3/2018	8.00	4 days

Ilustración 6. Desglose tareas PEC 3 y 4

Diagrama de Gantt completo

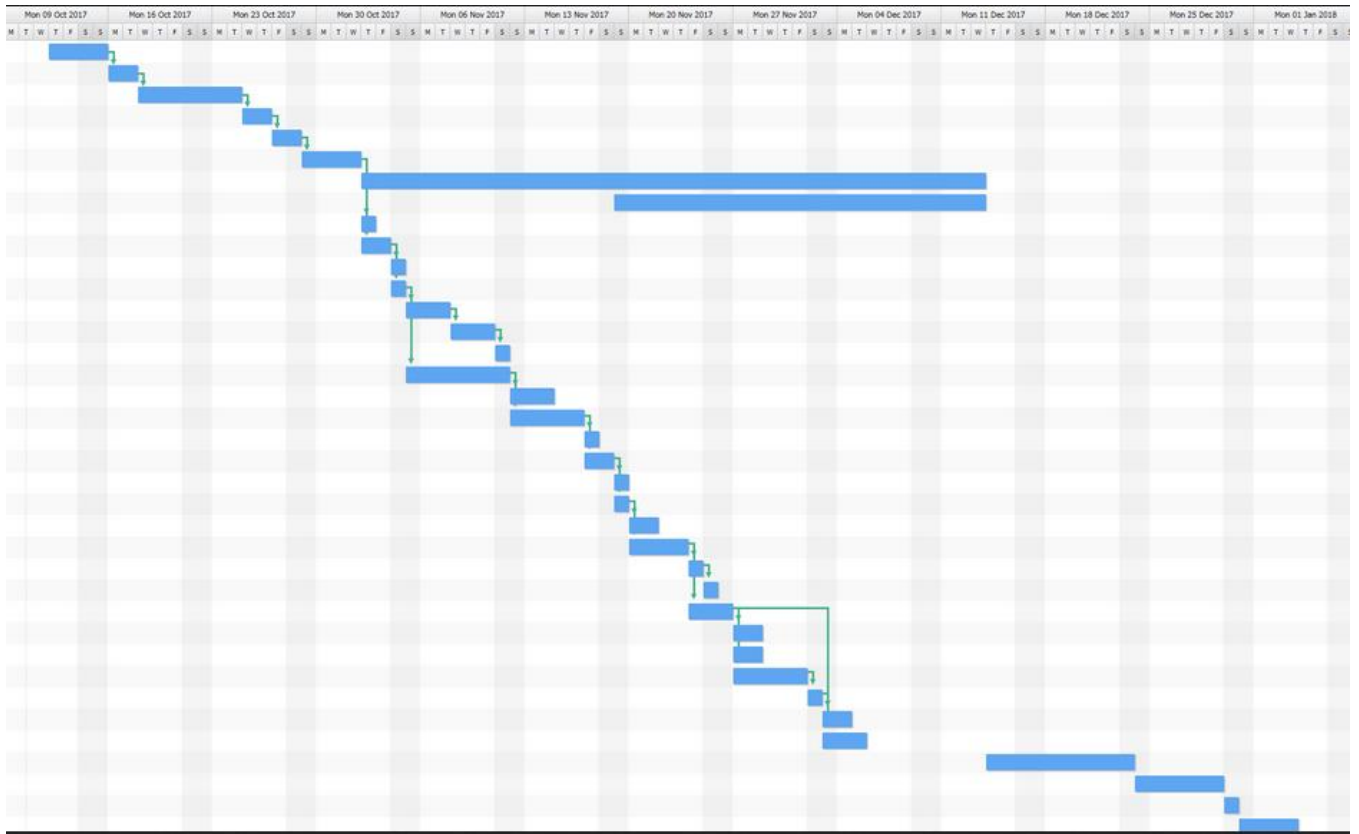


Ilustración 7. Diagrama de Gantt desarrollo completo

RESUMEN DETALLADO DE LAS TAREAS DE LA PEC 2

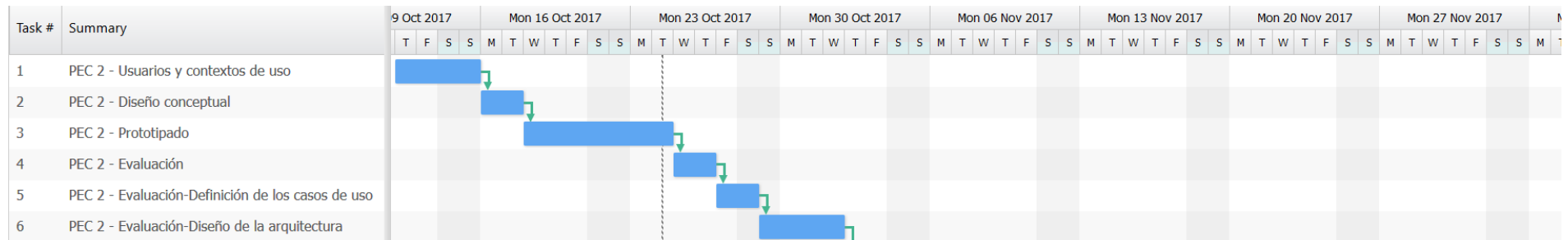


Ilustración 8. Resumen detallado de las tareas de PEC 2

RESUMEN DETALLADO DE LAS TAREAS DE LA PEC 3

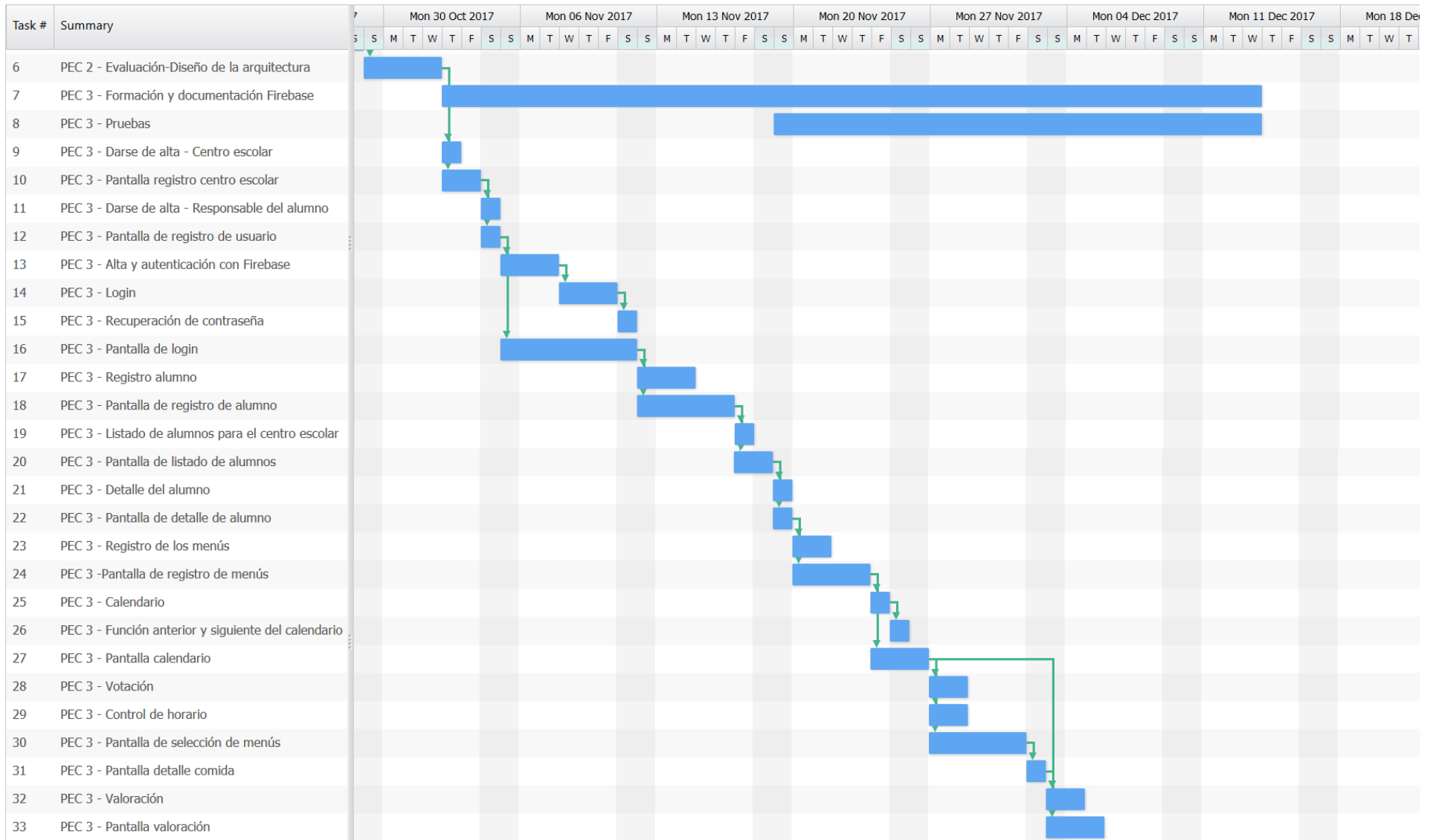


Ilustración 9. Resumen detallado de las tareas de PEC 3

RESUMEN DETALLADO DE LAS TAREAS DE LA PEC 4 (entrega final)

Task #	Summary	Mon 11 Dec 2017					Mon 18 Dec 2017					Mon 25 Dec 2017					Mon 01 Jan 2018					Mon 08 Jan 2018					Mon 15 Jan 2018					Mon 22 Jan 2018					Mon 29 Jan 20																																																						
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S																																		
34	PEC 4 - Memoria del TFM																																																																																										
35	PEC 4 - Anexo 1: Manual de usuario																																																																																										
36	PEC 4 - Anexo 2: Manual de instalación																																																																																										
37	PEC 4 - Vídeo de presentación																																																																																										

Ilustración 10 Resumen detallado de las tareas de la entrega final

◀◀
◀
December
▼
2017
▶
▶▶

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
50	10	11	12	13	14 ■ #1 UOC -TFM ■ #34 PEC 4 - Memoria del TFM	15	16
51	17	18	19	20	21	22	23 ■ #1 UOC -TFM ■ #34 PEC 4 - Memoria del TFM
52	24 ■ #1 UOC -TFM ■ #35 PEC 4 - Anexo 1: Manual de usuario	25	26	27	28	29 ■ #1 UOC -TFM ■ #35 PEC 4 - Anexo 1: Manual de usuario	30 ■ #1 UOC -TFM ■ #36 PEC 4 - Anexo 2: Manual de instalación
1	31 ■ #1 UOC -TFM ■ #37 PEC 4 - Vídeo de presentación						

Ilustración 11 Resumen detallado de las tareas de la entrega final - Calendario

1.6 Revisión de la planificación

Como dijimos anteriormente, la idea es trabajar con una herramienta de control de versiones, en este caso GIT. Para corroborar el uso de dicha herramienta y de que se ha seguido la planificación y en el orden indicado de las tareas, mostramos una captura del grafo generado mediante la plataforma GitLab. Los nodos de este grafo representan cada una de las subidas que se hicieron durante el desarrollo de la aplicación. Siempre hemos trabajado en la rama principal (master), por lo que solo veremos la ramificación vertical sobre la misma. Lo vemos en la siguiente imagen:

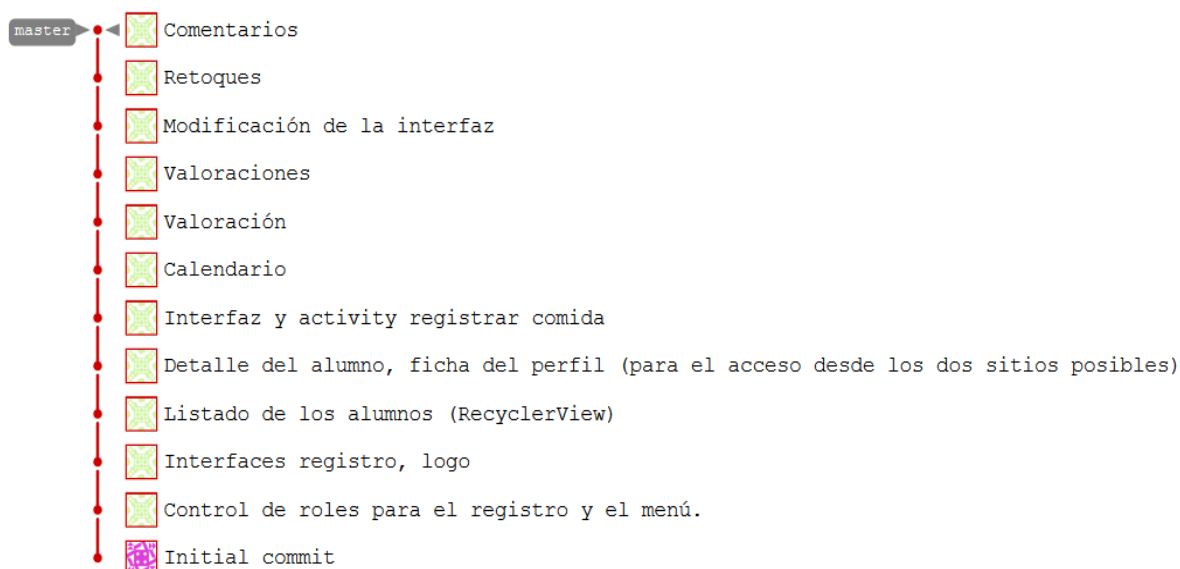


Ilustración 12 Grafo de los commits en el repositorio de GitLab

La planificación ha seguido su curso con normalidad, aproximadamente hasta finales de la PEC 3. Las tareas se han ido realizando poco a poco, intentando continuar con el plan establecido en el análisis de la planificación, pero el trabajo se ha ido acumulando según avanzábamos hacia las últimas tareas.

Unos de los principales problemas ha sido la forma en la que se estaban guardando los datos en la plataforma de Firebase. En un principio, se intentó trasladar la idea de un modelo de datos relacional a una base de datos no relacional. Según se avanzaba, se daba un paso hacia delante y dos hacia detrás. Darse cuenta de que el modelo de datos no era el idóneo para el trabajo que se estaba realizando y volver a pensar un nuevo modelo que se integrara correctamente con la forma de guardar los datos en formato JSON conllevó en una pérdida de tiempo considerable.

Por otro lado, pequeños detalles que no se tienen en cuenta acaban provocando dolor de cabeza. Esta pequeña funcionalidad, dentro de bonitas acciones que se pensaron en un primer momento... como validaciones de datos, tamaño de las imágenes, tiempos de carga, retroceso sobre las pantallas, cierres de sesión, número de lecturas o escrituras, un buen diseño de la interfaz... algunas de ellas se tienen que sacrificar para poder realizar una aplicación decente en un tiempo decente.

Al llegar a la última semana, para cumplir los objetivos, se dedica unas 10 horas por día. Esto se traduce a que en los últimos 7 días se trabaja unas 70 horas en el proyecto. Aun así, la memoria no se elabora todo lo que se merecía, dejando gran parte para la entrega final. Según la planificación realizada en una semana se deben de dedicar 21 horas al proyecto, por lo que aproximadamente se dedica 50 horas extras a la PEC 3.

Posteriormente, al trabajo realizado, se considera que se vuelve a la normalidad y al rumbo normal de la planificación.

1.7 Viabilidad económica

El coste final de un proyecto tiene muchas variables, como pueden ser el caché de la empresa, el precio por hora de un analista, programador, diseñador, hasta las diferentes tecnologías que se vayan a emplear en el desarrollo. En este caso, hablaremos sobre los costes fijos de las tecnologías que vamos a utilizar durante el ciclo de vida de la aplicación, que va desde la planificación hasta la publicación del mismo.

Comenzamos con la herramienta principal de desarrollo, Android Studio v3.0. Es la herramienta por excelencia para trabajar con aplicaciones enfocadas al sistema operativo Android. El uso de ésta no conlleva ningún coste, ofreciendo continuas actualizaciones gratuitas del producto completo.

Ilustración 13 Plan Aceproject



Por otro lado, tenemos Aceproject, que nos ayuda a la planificación y gestión de las tareas. Una herramienta muy buena para la gestión de proyectos, como ya hemos podido observar en el apartado de la planificación. En la imagen que encontramos a la izquierda, observamos que es de uso gratuito, ya que en este caso solo vamos a planificar un proyecto, en el cual va a trabajar una sola persona y evidentemente no llegaremos a superar el límite de tamaño establecido. Idóneo para este trabajo.

Siguiendo con otra de las herramientas que nos proporciona seguridad en el transcurso de la programación de la aplicación, nos encontramos con GitLab. Una plataforma que permite que almacenemos el proyecto y cada uno de los cambios que hacemos diariamente mediante Git, el software de control de versiones. Se ofrecen diferentes planes, pero para el uso que se quiere, perfectamente podemos adquirir el plan gratuito y trabajar con garantías. Lo podemos ver en la ilustración 15.

Por último, el recurso principal para el desarrollo de la aplicación es Firebase. Una plataforma que nos ofrece gran cantidad de productos, los cuales nos ayudan y nos dan

facilidades en la construcción del aplicativo final. En la ilustración 14, observamos el plan que hemos escogido para poder hacer uso de esta plataforma. Se ha hecho uso de dos de sus productos, Authentication y Realtime Database, sobre los que cumplimos con las condiciones para no tener ningún recargo.

Productos	Plan Spark Límites generosos para aficionados Sin cargo
Incluido sin cargo Authentication (except Phone Auth), Analytics, Predictions, App Indexing, Dynamic Links, Invites, Remote Config, Cloud Messaging (FCM), Performance Monitoring, Crash Reporting, and Crashlytics.	✓ Incluidos
Realtime Database Conexiones simultáneas ? GB almacenados GB descargados Multiple databases per project	100 1GB 10 GB/mes ✗

Ilustración 14 Plan Firebase

Early Adopter
\$0 per user monthly
Unlimited repositories
2,000 CI pipeline minutes/month
Basic support
Multiple issue boards
Burndown charts
Push rules
Related Issues
Service Desk
Current plan

Ilustración 15 Plan GitLab

Para concluir, quiero decir que se va a desarrollar la aplicación a coste cero. Sin embargo, si se quisiera publicar en la tienda de Google, habría que pagar una tasa de 25 USD por tener una cuenta de desarrollador. Si se establece un precio a la aplicación por su compra, el beneficio sería de un 70% del total, ya que el 30% de los beneficios van directamente para Google.

1.8 Breve resumen de productos obtenidos

Al finalizar el desarrollo de nuestra aplicación, tendremos los siguientes productos:

- Fichero .zip con todo lo realizado durante el desarrollo, incluye todo el código fuente de la aplicación.
- Fichero del esquema de base de datos noSQL en formato JSON.
- Aplicación empaquetada en formato “apk” para poder ser instalada.
- Memoria del trabajo.

- Anexo 1: Manual de usuario.
- Anexo 2: Manual de instalación.
- Vídeo de presentación.

1.9 Breve descripción de los otros capítulos de la memoria

En este apartado, vamos a comentar brevemente lo que podemos esperar en esta memoria. Explicaremos sin entrar en detalle, cada uno de los apartados que componen nuestro documento. Vemos cada uno de ellos a continuación:

Usuarios y contextos de uso. Da a conocer las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso. Se centra en las necesidades de los usuarios, para poder implementar la funcionalidad correcta y poder satisfacer cada uno de sus deseos de uso. Si analizamos cada uno de los problemas que pueden tener, nuestro producto tendrá mejores garantías y expectativas de obtener un buen resultado.

Diseño conceptual. Una vez que se ha obtenido los requisitos de los usuarios y sus ideas principales de uso, se elabora cada uno de los posibles escenarios en los que un usuario de nuestra aplicación se pueda encontrar. Así, podremos determinar si vamos en el camino correcto. Nuestra aplicación va a ser desarrollada para la plataforma de Android, por lo tanto, nuestros escenarios estarán enfocados al uso del Smartphone.

Prototipado. Un prototipo es una representación de la aplicación que permite mostrar decisiones de diseño y que éstas sean evaluadas antes de desarrollar el producto final. Su versatilidad hace que sea sencillo y económico introducir modificaciones en el diseño e iterar incorporando mejoras fruto de la discusión con los miembros del equipo o de los resultados obtenidos en la evaluación. Intenta plasmar cada uno de los flujos de interacción que se han podido ver en los escenarios planteados en el apartado anterior. Este apartado, se inicia con un análisis de las alternativas que tenemos para el desarrollo de un buen prototipo.

Definición de los casos de uso. Define formalmente cada uno de los casos de uso. Tras el prototipado, visualizaremos mejor la funcionalidad de nuestra aplicación. Para ello, muestra gráficamente cada uno de los casos de uso, mediante diagramas UML. En cada uno de estos diagramas, quedarán identificados los actores o condiciones que pueden alterar el flujo de interacción.

Diseño de la arquitectura. El objetivo de este apartado es el de definir la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, el modelo de datos que se utilizará para gestionar los diferentes procesos y la estructura de la API que servirá para realizar las peticiones enviados al servidor desde un cliente. Para ello, muestra gráficamente tanto el esquema de base de datos, como el modelo de datos de nuestra aplicación, mediante diagramas UML. También, se puede ver un diagrama orientativo para reflejar el modelo de arquitectura que se ha construido y seguido durante todo el desarrollo.

Por último, analiza las distintas alternativas que se ha encontrado para hacer uso de un modelo de transacción de datos cliente-servidor.

Implementación. Detalla algunos de los detalles importantes implicados en el desarrollo de la aplicación. Instrucciones del código fuente o aspectos de diseño se comentarán para poder entender cómo se ha construido el producto final.

Información y pruebas. Breve resumen de los diferentes tamaños de la aplicación cuando ha sido descargada e instalada en un dispositivo. Se mencionarán las pruebas que han sido realizadas para verificar el comportamiento de la aplicación.

Vías futuras. Este apartado identifica las posibles vías futuras que podría tener nuestro proyecto, así como las distintas ideas y posibilidades de mejora de nuestro producto.

Conclusiones. Expone un pequeño análisis sobre nuestra experiencia a lo largo del desarrollo de la aplicación y se proporciona una opinión, basada en las distintas facilidades o dificultades encontradas.

Glosario. Definición de los términos y acrónimos más relevantes que nos podemos encontrar en la memoria.

Bibliografía. Referencias bibliográficas que nos podemos encontrar durante la lectura de esta memoria.

Anexo 1. Manual de usuario. Manual de usuario que indica paso a paso la funcionalidad de la aplicación. Cada interacción posible será detallada cuidadosamente.

Anexo 2. Manual de instalación. Ofrece un manual de instalación, obteniendo el binario mediante otro tipo de descarga en nuestro Smartphone, que no sea desde Play Store. También, indicaremos como simular nuestra aplicación en Android Studio, sin necesidad de tener que instalarla en nuestro dispositivo.

2. Usuarios y contextos de uso

2.1 Usuarios principales

Tras el análisis realizado en el primer punto, hemos llegado a la conclusión de que van a existir dos roles en nuestra aplicación, que podemos identificar como el centro escolar y la persona responsable del alumno (padre/madre/tutor), que por simplicidad los llamaremos a partir de ahora **centro escolar** y **alumno**.

Ahora, nuestra labor es relacionar estos dos tipos de roles, con los diferentes usuarios reales a los que les va a corresponder tal etiqueta. Se considera que el rol correspondiente al centro escolar llevará asociado consigo un cargo administrativo en el centro o un cargo dentro del personal laboral del servicio de comedor. Alzando la vista hacia este tipo de trabajo, basándonos en una experiencia personal, se considera que debemos de orientar tal funcionalidad hacia personas que se encuentren entre los 30 y 50 años aproximadamente.

Por otro lado, los usuarios potenciales asociados al rol alumno, serán precisamente los padres o tutores de un alumno. Para poder orientarnos, hemos analizado los posibles usuarios mediante dos perspectivas diferente. Como primer punto, podemos decir que por normal general y según un estudio que fue publicado el año pasado [7], actualmente, la media de edad de la maternidad se encuentra aproximadamente en los 30 años. Como segundo punto, también tenemos que tener presente que, la educación primaria se realiza entre los 6 y 12 años (Ministerio de educación [8]), que suelen ser los alumnos que hacen uso del comedor y queremos orientar esta aplicación. Por lo tanto, basándonos en los dos puntos, podemos focalizar este rol en las personas que tengan entre 30 y 50 años.

Una vez que hemos detectado cada uno de nuestros usuarios principales, podemos corroborar que toda nuestra aplicación estará orientada a usuarios que se encuentren entre los 30 y 50 años casi con total seguridad.

2.2 Contextos de uso: centro escolar

Cada día existe un control más exhaustivo sobre la alimentación de los niños pequeños. Los centros escolares, quieren que la información llegue a los padres de estos niños. Una posible solución es el desarrollo de una aplicación en la que se pueda informar a los padres sobre los menús que ofrecen a sus hijos y darle la oportunidad de participar ellos mismos. A la vez, controlar cuál es la alimentación que gusta más o menos, pasando por un mínimo control de calidad.

Por esta demanda en el mercado, consideramos que, para satisfacer la necesidad de los centros escolares, en nuestra aplicación, se deben cumplir los siguientes requisitos:

- 1) Libre registro de un centro escolar.

- 2) Solicitar únicamente los datos necesarios, para no molestar al cliente.
- 3) Previamente al registro de los datos, la cuenta de correo electrónico deberá ser activada.
- 4) Sistema de verificación de cuenta, mediante un correo que se envía al usuario.
- 5) Ofrecer sistema de recuperación de contraseña.
- 6) La recuperación de contraseña se produce vía correo electrónico. La contraseña no se puede almacenar en claro en la base de datos. En el cuerpo del correo se introducirá un enlace, donde el usuario deberá pinchar para cambiar su contraseña.
- 7) Un menú, sencillo e intuitivo que a simple vista se pueda comprender la funcionalidad que se ofrece.
- 8) Registrar los distintos menús.
- 9) En cada registro de menú se indicará foto, el día que será publicado y los datos de la comida.
- 10) La foto del menú será cargada entre las que se encuentran disponibles en el dispositivo móvil o accesibles desde el mismo, como puede ser la galería de fotos o Google Drive.
- 11) Acceder a un listado completo de todos los alumnos que se encuentran en el comedor.
- 12) Visualizar la ficha de un alumno.
- 13) A la ficha del alumno se accede desde el listado de alumnos.
- 14) Ver el calendario.
- 15) Desde el calendario, acceder a los menús de cada día.
- 16) Los días del calendario tienen estado.
- 17) Los estados del calendario indicarán la funcionalidad permitida.
- 18) Los estados del calendario son:
 - a. CERRADO: No hay dos menús disponibles
 - b. VOTACIÓN: Hay dos menús disponibles y es un día posterior al actual.
 - c. DISPONIBLE: Hay dos menús disponibles y es el día actual.
 - d. VALORACIÓN: Hay dos menús disponibles y el día ya se ha pasado.

- 19) El centro escolar independientemente de cuál sea el estado, no podrá realizar ninguna acción sobre el día seleccionado.

2.3 Contextos de uso: alumno

En muchos centros escolares se mantiene el servicio de comedor, algunos por facilitar el horario a las familias y otros por tener horario lectivo por la tarde. A los padres les invade la preocupación de cómo se alimentan sus hijos cada día. Llevar un control y un seguimiento adecuado del tipo de alimentación que proporciona el centro, ayuda a una estabilidad alimenticia para los niños y a transmitir tranquilidad a los padres.

Por ello, se va a incluir una serie de necesidades, que se consideran prioritarias en el desarrollo de esta aplicación. Vamos a intentar contentar a cada uno de esos padres que piensan que deben conocer cuál es el estado del servicio de comedor del centro escolar y que puedan ejercer sus medidas de presión hacia el mismo para que sus hijos puedan llevar una dieta equilibrada y constante a lo largo de todo el curso. Los requisitos que incluiremos en nuestro producto serán los siguientes:

- 20) Libre registro de cualquier usuario.
- 21) El usuario deberá cumplimentar los datos solicitados.
- 22) Previamente a completar el registro, el usuario deberá verificar su cuenta.
- 23) Sistema de verificación de cuenta mediante un correo electrónico que se le envía al usuario.
- 24) Una alternativa al registro natural será mediante las credenciales de una cuenta de Google.
- 25) La verificación de cuenta es automática cuando se realiza el registro mediante una cuenta de Google.
- 26) El usuario indicará a qué centro pertenece.
- 27) Deberá rellenar el perfil correspondiente del alumno que sea responsable.
- 28) Deberá insertar una foto del alumno en el registro. Para realizar dicha foto se activará la cámara de fotos del dispositivo al pinchar en el lugar indicado de la pantalla. Con la intención de que sea una foto reciente.
- 29) Para acceder al menú será obligatorio registrar al alumno.
- 30) El usuario podrá visitar su propio perfil, pudiendo supervisar los datos proporcionados al centro escolar.
- 31) Tendrá acceso a un calendario, donde se publicarán los menús.

- 32) Podrá votar diariamente cada uno de los menús.
- 33) Valoración del menú diario.
- 34) La funcionalidad estará restringida al estado actual de cada día que se visualice en el calendario.
- 35) Los estados del calendario son:
 - a. CERRADO: No hay dos menús disponibles
 - b. VOTACIÓN: Hay dos menús disponibles y es un día posterior al actual. El alumno podrá votar uno de los dos menús.
 - c. DISPONIBLE: Hay dos menús disponibles y es el día actual. El usuario podrá ver el menú seleccionado, ya que aparece destacado.
 - d. VALORACIÓN: Hay dos menús disponibles y el día ya se ha pasado. El usuario podrá realizar una valoración, indicando la puntuación que a su juicio merece el menú de ese día.

2.4 Accesibilidad

Los usuarios no quieren perder el tiempo en el aprendizaje, por lo que se ha intentado que la carga cognitiva sea la menor posible. Esto, ha influido en la elección sobre la pantalla del calendario, que veremos más adelante en el apartado de prototipado, una de las pantallas más importantes de nuestra aplicación. A través de esta pantalla se votarán y valorarán los menús. Se han pensado dos alternativas, las cuales valoramos a continuación:

Alternativa 1: Los días se muestran en una lista. Cada ítem de la lista se divide en tres partes, el día al que pertenece, el estado en modo texto en el que se encuentra (cerrado, votación, valoración y disponible). Por último, un color identificativo por estado. Mostraremos 5 ítems que identificarán cada uno de los días de la semana actual. Permitiremos la navegación hacia la semana anterior y posterior.

Alternativa 2: Tipo calendario. El calendario mostrará el mes actual completo y se podrá navegar entre los meses anteriores y posteriores. Cada día será identificado con un color, el cual dependerá del estado en el que se encuentre (cerrado, votación, valoración y disponible). Debajo del calendario, mostraremos una leyenda con los colores y sus significados.

Nosotros nos hemos decantado por la primera alternativa, aunque se considera que la segunda alternativa es más potente. Pero, debido a que gran cantidad de aplicaciones han fracasado por tener un nivel alto de complejidad, nos centraremos en “hacerle la vida más fácil al usuario”. La elección sobre la primera alternativa ha sido basada en los siguientes criterios:

- Menos carga cognitiva, debido a que el usuario no tiene por qué empezar a identificar el color con el estado en el que se encuentra y mucho menos aún, tener que recordar cada uno de ellos cada vez que esté haciendo uso de la aplicación.

- Mayor sencillez. Se identifican un menor número de días a la vez. El estado se puede identificar de una manera clara y concisa.
- Mejor dinamismo, ya que los componentes tendrán un mayor tamaño y facilita interactuar con la pantalla en movimiento mediante una mayor legibilidad de la misma.
- Tamaño de los botones. Cada ítem de la lista es un botón que ocupa todo el ancho de la pantalla, en comparación con el calendario que tienes que “acertar” en clicar el día correcto.
- Mayor brevedad en los mensajes y los enlaces son descriptivos. Lo importante de cada día es el estado en el que se encuentra, para conocer el usuario que función tiene disponible. En nuestra elección se conoce con una palabra clara y grande. En la otra pantalla, no existe mensaje, creando confusión y un mayor trabajo visual, debido a que tiene que ir mirando el color del día e identificarlo en la leyenda.

Para el diseño de la aplicación, trataremos de mantener una estructura homogénea en cada una de las pantallas. Se considera que no se deben dar cambios bruscos en el diseño, por lo que una estructura similar en cada una de las pantallas hace más agradable el uso de la aplicación.

Conociendo que las figuras identifican más que los colores, en la aplicación se ha considerado el uso de iconos para facilitar el aprendizaje. Por ejemplo, en la pantalla de login, aparte de tener un placeholder para el nombre de usuario y la contraseña, acompañamos con el icono de una persona y con el icono de un candado. Estos detalles, ayudan al usuario a identificar en una mayor brevedad de tiempo la problemática de nuestra aplicación.

Por último, en los menús de cada uno de los roles que podemos identificar en nuestra aplicación, cada botón se compone de un texto explicativo y un icono relacionado con la funcionalidad que se ofrece pinchando en el mismo.

2.5 Escenarios

A continuación, vamos a describir los posibles escenarios en los que se pueden encontrar nuestros usuarios a la hora de hacer uso de nuestra aplicación.

Escenario 1



Rol: Centro escolar

Nombre: Ana.

Edad: 44 años.

Profesión: Cocinera.

Descripción de la persona: Ana es cocinera desde hace 20 años. Comenzó como ayudante y ahora es dueña de su propia empresa cocina. Tiene 2 hijos, uno de 4 años y otro de 6 años. Vive en San Javier y va andando a su puesto de trabajo. Su pasión por los niños le ha llevado a trabajar en un centro escolar, encargándose del menú diario para ellos e intentando conservar una dieta saludable, como es la dieta mediterránea. Tiene un smartphone con sistema operativo Android, donde pasa el tiempo muerto navegando por internet en busca de las distintas recetas y novedosos ingredientes para ofrecer delicias a los niños. Tiene tarifa de datos contratada y su conexión es 4G.

Descripción del escenario: Son las 15:00 de un sábado cualquiera. Está disfrutando de una comida en familia en un restaurante del pueblo. Su hijo mayor se está comiendo con gusto el plato principal, por lo tanto, ella se interesa por el menú. Saca su smartphone y decide que para un día de la siguiente semana es uno de los menús a escoger. Lo registra en la aplicación, aprovechando que ha podido echarle una foto a uno de los platos. Introduce los datos importantes como son los ingredientes o el tipo de menú (carne, pescado, pasta).

Escenario 2



Rol: Centro escolar

Nombre: Pepe.

Edad: 32 años.

Profesión: Administrativo.

Descripción de la persona: Pepe trabaja como administrativo en uno de los colegios de pueblo. Tiene que ir al trabajo en coche, porque vive en las afueras del mismo. Tiene un hijo de 8 años, al que lleva consigo al colegio cada día aprovechando que van al mismo. Es el encargado de la administración de muchas secciones del colegio, entre ellas el servicio de comedor que ofrecen. Tienen a una empresa externa contratada, que es la encargada de enviar por correo cada semana los menús que van a ofrecer. Hace uso de un smartphone con sistema operativo Android, que está al servicio del colegio. No tiene tarifa de datos contratada, pero hace uso de la conexión wifi de la que dispone el colegio.

Descripción del escenario: Es un lunes por la mañana, se encuentra en su puesto de trabajo como cada día. Le llega un correo al móvil, a la cuenta del colegio, en el que acaba de recibir los menús de la semana siguiente. Abre el correo y se descarga todas las fotos que ha recibido con el listado de menús. Registra cada uno de los menús, con los datos que le han sido proporcionados.

Escenario 3



Rol: Alumno

Nombre: Alex.

Edad: 50 años.

Profesión: Albañil.

Descripción de la persona: Alex es albañil toda su vida, no tiene estudios y desde hace 5 años es separado. Tiene 1 hija de 10 años. Vive en San Javier, pero su trabajo lo tiene alejado de casa durante todo el día. La semana que le toca estar con su hija, tiene una gran preocupación por la alimentación de su hija y va al comedor escolar. Tiene un smartphone de última generación, con conexión de datos 4G y el que casi no sabe utilizar. Necesita que las aplicaciones que utiliza sean intuitivas y de fácil aprendizaje.

Descripción del escenario: Son las 17:00 de un jueves cualquiera. Acaba de llegar de trabajar y ha ido a por su hija al colegio, después de que comiera allí y tuviera una actividad extraescolar. Al llegar a la casa, la hija le pide algo de comer. Él preocupado, se introduce en la aplicación y en pocos clics ve qué ha comido su hija. Ahora que sabe que esa comida no le ha gustado, puede darle una buena merienda.

Escenario 4



Rol: Alumno

Nombre: Alicia.

Edad: 36 años.

Profesión: Abogada.

Descripción de la persona: Alicia trabaja en un despacho de abogados en Barcelona. Tiene que ir al trabajo en coche, porque vive a 10 km del mismo. Tiene un hijo de 6 años, es primeriza y lo lleva al colegio cada día antes de comenzar su jornada laboral. Ni su marido ni ella pueden ir a recoger al niño a mediodía, por lo que se tiene que quedar en el comedor. Está histérica por el cuidado de su hijo, no come casi nada en casa y quiere saber qué le dan de comer en el colegio. También, quiere participar en la elección de la alimentación de su crío, ya que le da mucha importancia a que su niño mantenga una dieta alimenticia lo más sana

posible. Tiene un móvil con sistema operativo IOS, pero su marido tiene un smartphone con sistema operativo Android y con su tarifa de datos correspondiente.

Descripción del escenario: Es martes por la noche y acaba de llegar del trabajo. Coge como cada día el móvil a su marido y tras preguntar a su hijo, mira el menú diario para valorarlo. A su vez, comprueba la disponibilidad de los menús posteriores, para seleccionar cada uno de los que ella considera más saludables.

3. Diseño conceptual

3.1 Árbol de navegación

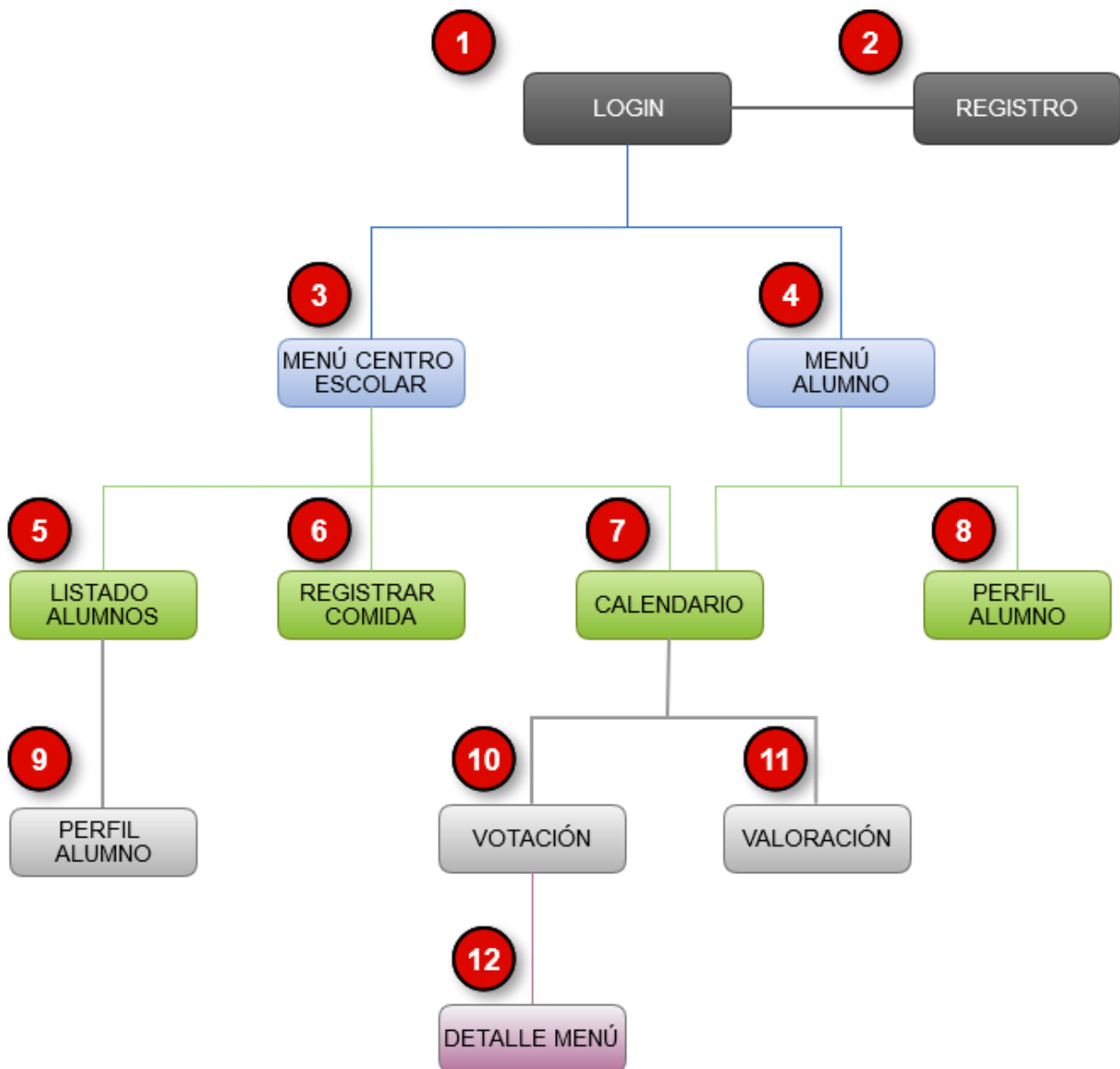


Ilustración 16 Árbol de navegación

Se ofrece una breve descripción de cada una de las pantallas:

1. Es la pantalla de acceso a nuestra app. Para ello se pedirá al usuario una serie de credenciales o acceso mediante la cuenta de Google.
2. Si el usuario no se encuentra registrado, tendrá la posibilidad de registrarse accediendo desde la pantalla de login. En esta pantalla también identificaremos los dos posibles roles que podrá tener un usuario (centro escolar o alumno).
3. Menú principal para el centro escolar, desde donde podrán acceder al registro de una comida o a la visualización del listado de alumno. También podrán ver cuál es el estado del calendario, tal como lo verán los responsables de los alumnos.
4. Menú principal para el alumno, desde donde podrán acceder al perfil del alumno para completar el registro. Posteriormente, tendrán acceso al calendario para comenzar a hacer uso de la misma.
5. El centro escolar podrá visualizar un listado de los alumnos que están haciendo uso del servicio de comedor. Desde el mismo, podrán acceder al perfil de un alumno.
6. Aquí el centro, podrá registrar cada uno de los menús que quieran ofrecen al alumnado.
7. Esta pantalla ofrecerá un calendario. Es la misma pantalla para el centro escolar y el alumno. Se podrán observar los distintos menús que se están ofreciendo ese día, si ya ha sido publicado la información correspondiente. Cada día podrá tener diferentes estados, según se encuentre la información disponible.
8. Se registrarán todos los datos del alumno.
9. Ofrece el perfil de un alumno, podría ser la misma pantalla que la del punto anterior. En este caso, el centro no puede modificar datos del alumno.
10. Pantalla a la que se accederá mediante el calendario, pinchando en un día en concreto. Esto será posible si aún no ha pasado el plazo de votación. Se mostrarán las distintas opciones, con el número de votos que lleva cada una de ellas en ese momento.
11. Se accederá de igual forma que la pantalla anterior. Esta pantalla se mostrará cuando el plazo de votación haya finalizado. Se mostrará el menú del día y se abrirá una posibilidad de valorarlo a lo largo del día.
12. Visualizaremos los detalles de cada menú, una fotografía del mismo, así como los ingredientes principales.

Revisión: Finalmente, la pantalla número 12 no ha sido implementada. Los datos que detallan un menú pueden ser ofrecidos con claridad en las pantallas de votación y valoración, por lo que intentamos no saturar al usuario con excesivas pantallas.

3.2 Análisis comparativo

En la introducción a este proyecto se realiza una pequeña investigación sobre productos similares que se encuentran actualmente en el mercado. Ahora, se va a analizar más detenidamente las opciones que ya comentamos. Redactamos una crítica constructiva, que nos ayudará a no cometer los mismos errores o extraer detalles que sean útiles en nuestro desarrollo.

App Serunion Educa

En la información de esta app se cita textualmente:

“Es la app que Serunión pone a disposición a padres y centros educativos para facilitar toda la información diaria y personalizada de los niños en el colegio, tales como los menús que se sirven, las actividades realizadas en el comedor y el tiempo de patio, y todos los servicios que han contratado. En definitiva, es un amplio canal de comunicación diario entre padres y colegio.”

Me gustaría comentar, que en este texto se dice que *“es un amplio canal diario entre padres y colegio, pero como veremos a continuación es una aplicación”*. Este caso la comunicación solo van en una dirección (centro educativo → padres), sin embargo, lo que nosotros pretendemos es dotar nuestra aplicación de una funcionalidad que se traduzca en una comunicación en ambos sentidos (centro educativo ↔ padres).

Por otro lado, vamos a comentar el diseño de su interfaz, la cual podemos ver a continuación:

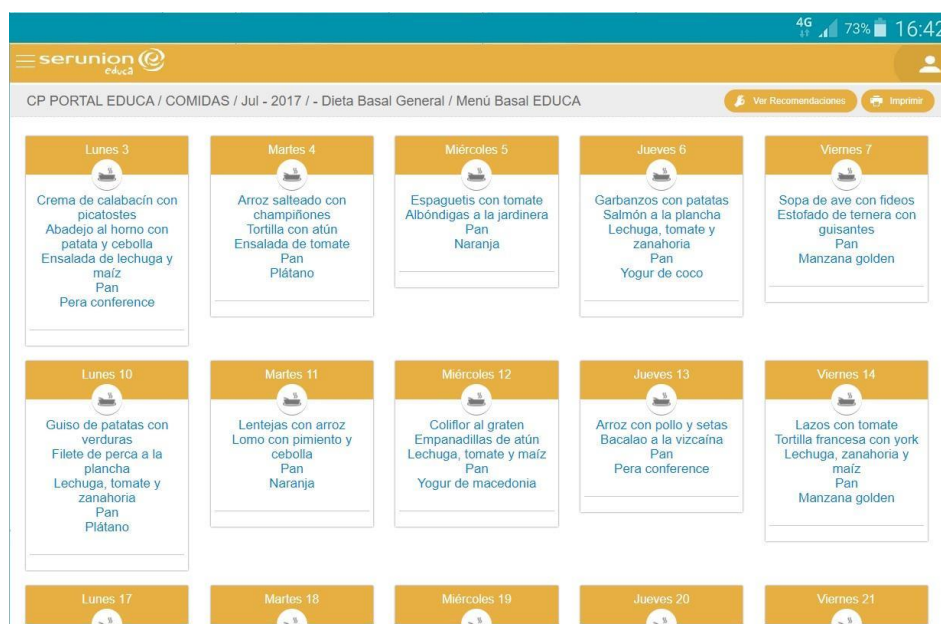


Ilustración 17 Interfaz Serunion Educa 1

Se puede observar en la imagen anterior que, en cuanto a diseño se ofrecen una gran cantidad de ítems. Esto puede provocar en el usuario cierta confusión o tener que detenerse a buscar exactamente el día que quiere encontrar. Por otro lado, cuanto mayor es la cantidad de ítems que queremos ofrecer por pantalla, se disminuye el tamaño del contenido y de la letra, otro punto que puede crear dificultad a cierto tipo de usuarios.

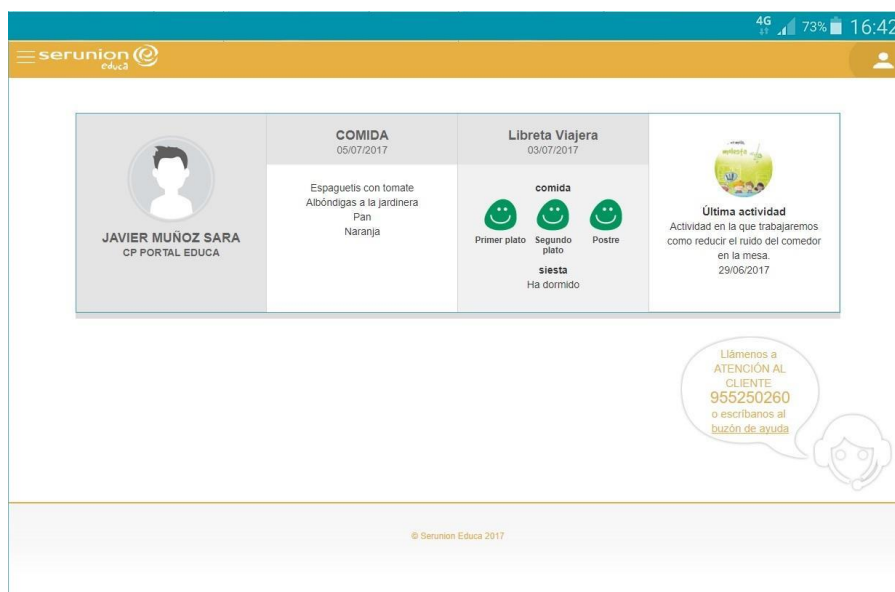
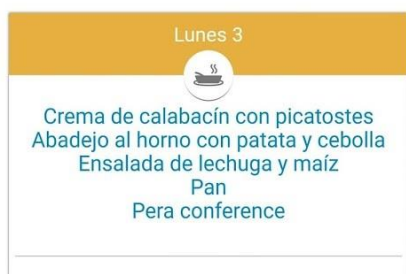
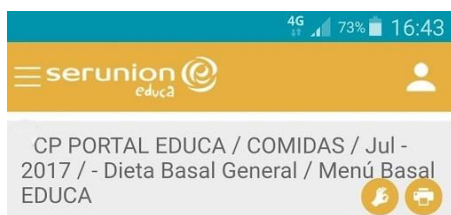


Ilustración 18 Interfaz Serunion Educa 2

Ilustración 19 Interfaz Serunion Educa 3



En la imagen anterior y en la imagen a nuestra izquierda, todo lo que se muestra es a título informativo. A simple vista, no se ve que se ofrezca ningún tipo de funcionalidad extra para que los padres puedan interactuar con el centro escolar.

En la imagen que tenemos a la izquierda, es lo que estábamos comentando anteriormente. En otra pantalla han restringido el número de ítems. Consideremos que crear diferentes vías de acceso hacia una misma información puede estar bien, pero puede llegar a confundir al usuario.

También, ofrece la miga de pan, que es más propio de una aplicación web. Desde nuestro punto de vista, creemos que un usuario debe saber en cada momento dónde se encuentra, haciéndole interactuar fácilmente, sin llegar a tener que realizar tantos pasos para llegar a la información que se desea.

Por último, en la imagen anterior mezcla todo tipo de información. Se podría haber organizado por pestañas en una misma ventana, para que el usuario se centre en la información que está interesado.

En nuestro caso, hay que comprender que la edad que queremos abarcar oscila entre los 30 y 50 años. No tiene la misma capacidad cognitiva una persona de 30 años, que una persona de 50 años. Por otro lado, tampoco tienen la misma capacidad de comprensión, la misma calidad de visión, la misma velocidad de reacción, entre otras cosas. Por lo tanto, nuestra idea es ofrecer las mismas garantías a todo tipo de personas que se encuentre en este abanico. Para ello, ofreceremos pocos ítems por pantalla, con un tamaño de letra adecuado y que la facilidad de aprendizaje lleve al usuario a seguir haciendo uso de nuestra aplicación.

App Serunion Comedor Escolar

En esta aplicación hay poco que comentar, ya que es una aplicación meramente informativa. Es como una guía de cafeterías, con sus respectivos horarios y precios. Tan solo nos podemos fijar en la interfaz que han diseñado. Como podremos ver a continuación, el contraste de colores entre el texto y el fondo que se ha escogido hace que se tenga que forzar la vista para la comprensión. Esto implica que, no se tenga una lectura agradable y el usuario no se sienta cómodo.



Ilustración 20 Interfaz Comedor Escolar

Podemos comprobar que, en la primera y segunda imagen, el fondo no ayuda a distinguir los elementos de los que disponemos. Tampoco la lectura de la segunda imagen, con ese tipo de fuente y esos colores, se puede realizar de una forma agradable. Una buena solución hubiera sido difuminar la imagen de fondo, hasta llegar a un punto que resulte legible lo que queremos ofrecer.

Por último, en la última imagen, la calidad del texto es mucho mayor, pero podemos observar como hay un espacio entre el título y el contenido que luego conlleva a una mala lectura del final de la carta. Son detalles que tendremos en cuenta en el diseño de nuestra interfaz. Como ya hemos dicho muchas veces, una aplicación puede resultar ser bastante buena, pero acabará siendo desechada si el usuario no se encuentra comfortable haciendo uso de la misma.

4. Prototipado

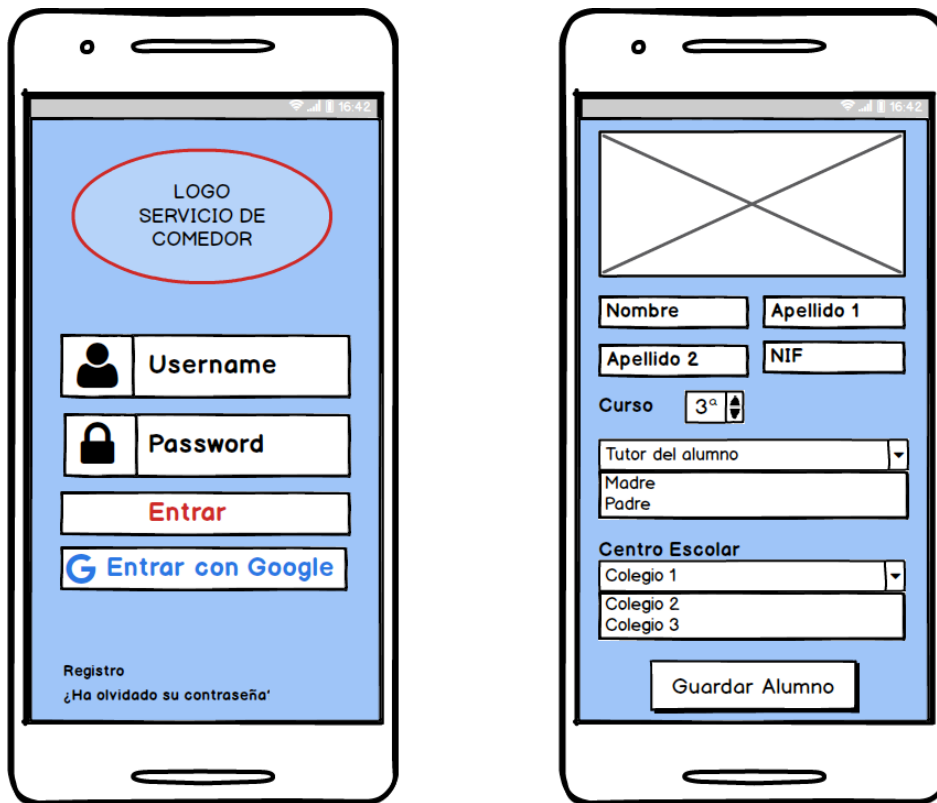


Ilustración 21 Pantallas login y perfil de alumno



Ilustración 22 Pantallas menú del centro escolar y registro de comida

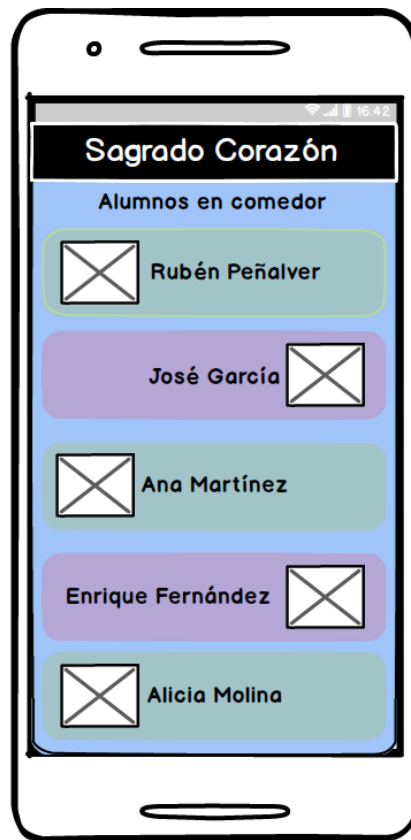
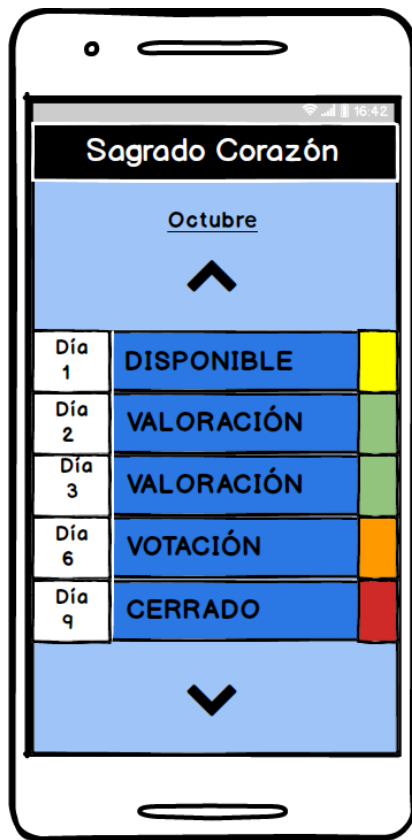


Ilustración 23 Pantallas calendario y listado de alumnos

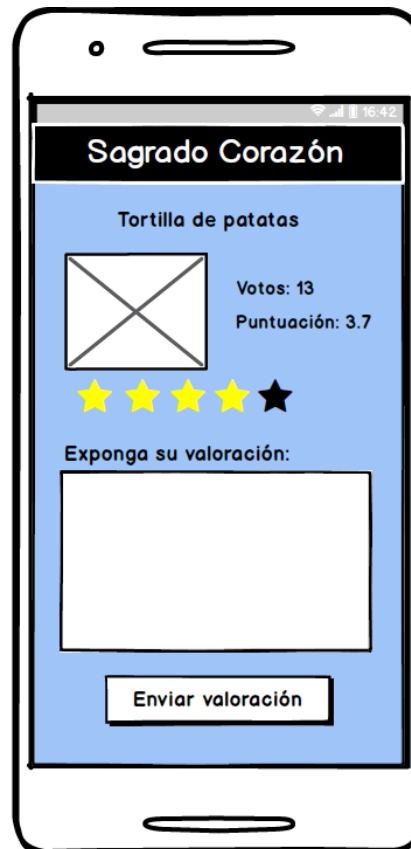
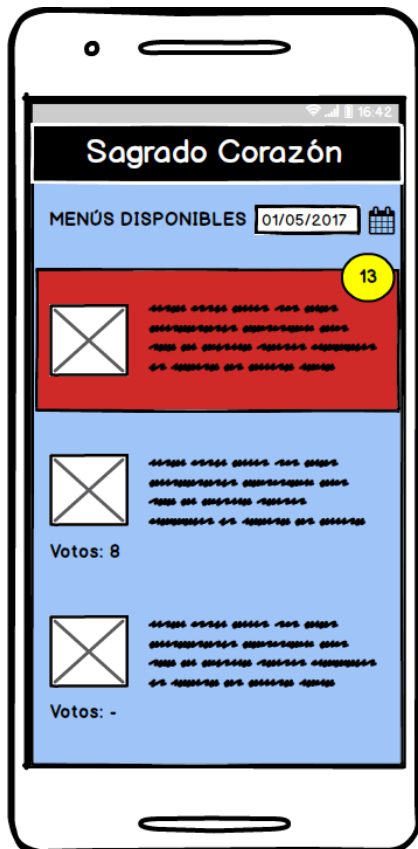


Ilustración 24 Pantallas votación y valoración de un menú

5. Definición de los casos de uso

5.1 Diagrama de casos de uso

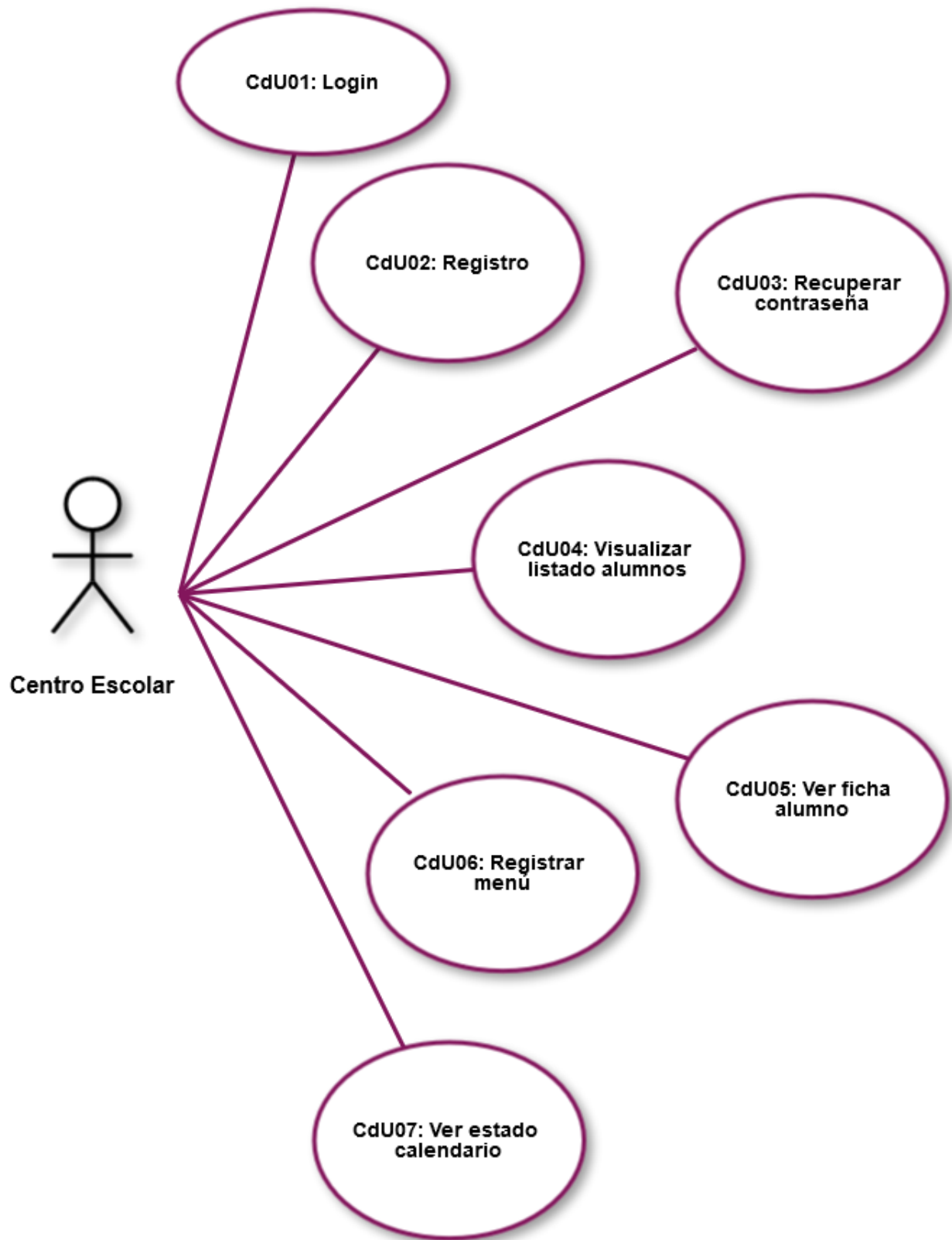


Ilustración 25 Diagrama de casos de uso del centro escolar

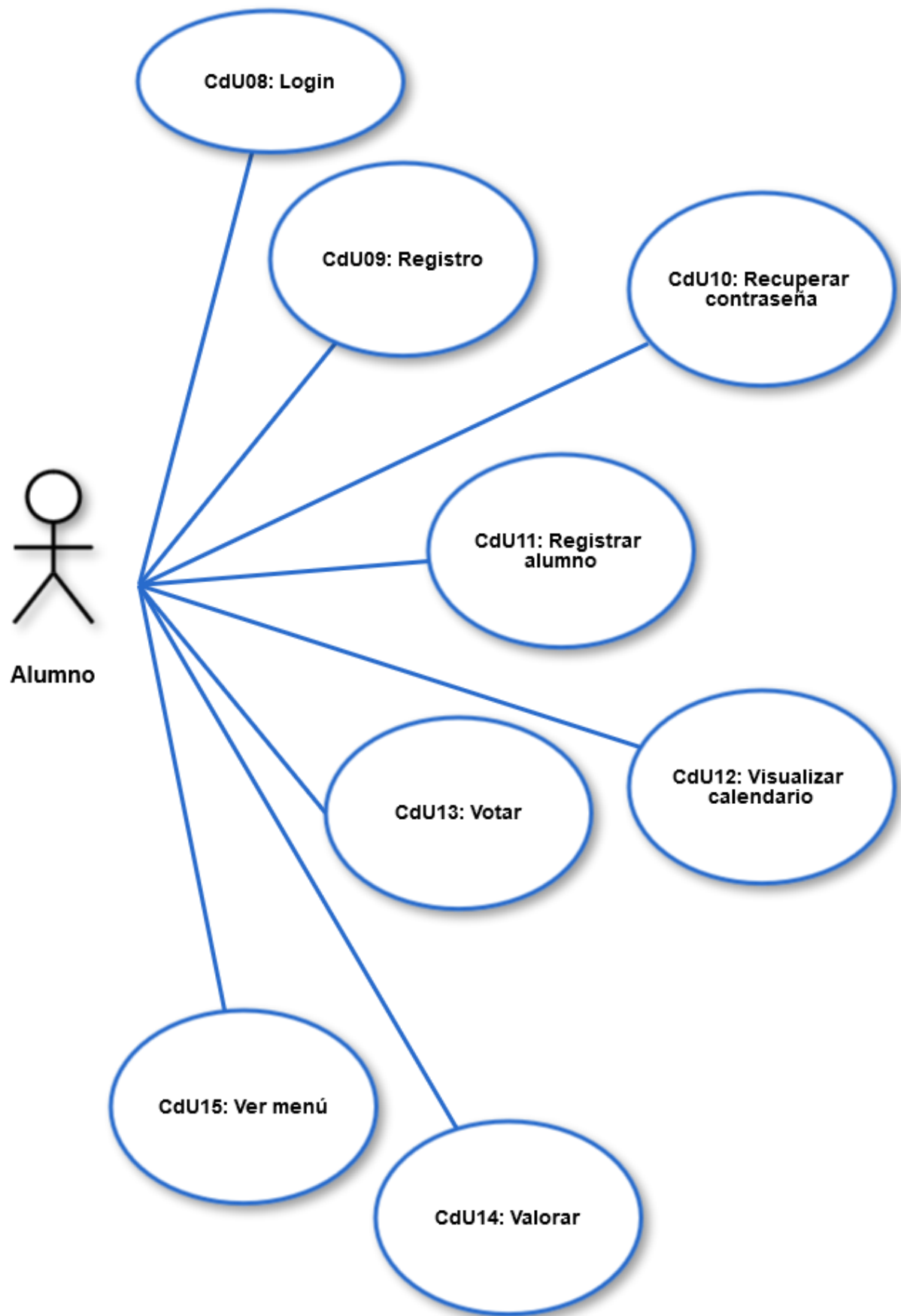


Ilustración 26 Diagrama de casos de uso del alumno

5.2 Casos de uso

Caso de Uso	CdU01 – Login
Descripción	El centro de escolar accede a la aplicación.
Actores	Centro escolar
Asunciones	El centro escolar se encuentra registrado.
Pasos	1.A. Introducir credenciales 2.A. Pulsar el botón de login
Variaciones	1.A. Pulsar el botón de entrar con Google. 2.S. El sistema extrae las credenciales de la cuenta de Google y las compara con las que se tienen registradas.
Cuestiones	

Caso de Uso	CdU02 – Registro
Descripción	El centro escolar procede a registrarse.
Actores	Centro escolar
Asunciones	El centro escolar tiene cuenta de Google.
Pasos	1.A. Acceder a la pantalla de registro mediante el enlace de la pantalla de login. 2.A. El centro escolar rellena todos los datos solicitados. 3.A. Pulsar el botón para finalizar el registro. 4.S. El sistema almacena los datos del nuevo centro.
Variaciones	1.A. Pulsar en el botón de entrar con Google. 1.S.1. El sistema extraerá los datos de su cuenta de Google y procederá a almacenarlas en nuestra base de datos.
Cuestiones	Conocer qué datos vamos a extraer de la cuenta de Google.

Caso de Uso	CdU03 – Recuperar contraseña
Descripción	El centro escolar se ha olvidado de la contraseña y quiere recuperarla.
Actores	Centro escolar
Asunciones	El centro se encuentra registrado.
Pasos	<p>1.A. Rellena el nombre de usuario en la pantalla de login.</p> <p>2.A. Pincha en el enlace de recordar contraseña de la pantalla de login.</p> <p>3.S. El sistema envía un correo a la cuenta proporcionada con las credenciales.</p>
Variaciones	
Cuestiones	

Caso de Uso	CdU04 – Visualizar listado alumnos
Descripción	Listar a todos los alumnos que se encuentran registrados.
Actores	Centro escolar
Asunciones	Hay alumnos registrados y el centro se encuentra en el menú.
Pasos	<p>1.A. Clicar en el elemento de menú listado de alumnos.</p> <p>2.S. El sistema mostrará en una nueva pantalla todo el listado de alumnos que se encuentren registrados.</p>
Variaciones	2.S. No hay alumnos registrados y el sistema mostrará un mensaje.
Cuestiones	

Caso de Uso	CdU05 – Ver ficha alumno
Descripción	El centro escolar quiere ver la ficha de un alumno.
Actores	Centro escolar
Asunciones	Hay alumnos registrados.
Pasos	<p>1.A. En el listado pinchar sobre uno de los alumnos que se muestran.</p> <p>2.S. El sistema cargará los datos de ese alumno y los mostrará en una nueva pantalla.</p>
Variaciones	
Cuestiones	

Caso de Uso	CdU06 – Registrar menú
Descripción	El centro escolar quiere registrar un menú en el sistema.
Actores	Centro escolar.
Asunciones	El día seleccionado se encuentra en estado CERRADO.
Pasos	<p>1.A. Pinchar sobre la opción de registrar menú existente en el menú.</p> <p>2.A. Rellenar los datos que se solicitan.</p> <p>3.A. Adjuntar una foto del menú.</p> <p>4.A. Finalizar registro</p> <p>5.S. El sistema almacenará los datos del nuevo menú.</p> <p>6.S. El sistema actualiza el estado del día correspondiente que se ofrece en el calendario.</p>
Variaciones	
Cuestiones	

Caso de Uso	CdU07 – Ver estado calendario
Descripción	Acceder al calendario para ver cuál es el estado de cada día.
Actores	Centro escolar
Asunciones	
Pasos	<p>1.A. Entrar mediante la opción correspondiente del menú.</p> <p>2.S. El sistema cargará cada uno de los días y sus correspondientes estados.</p>
Variaciones	
Cuestiones	

Caso de Uso	CdU08 – Login
Descripción	El alumno accede a la aplicación.
Actores	Alumno
Asunciones	El alumno se encuentra registrado.
Pasos	<p>1.A. Introducir credenciales</p> <p>2.A. Pulsar el botón de login</p>

	3.S. El sistema comprueba las credenciales.
Variaciones	1.A. Pulsar el botón de entrar con Google. 2.S. El sistema extrae las credenciales de la cuenta de Google y las compara con las que se encuentran registradas.
Cuestiones	

Caso de Uso	CdU09 – Registro
Descripción	El alumno procede a registrarse.
Actores	Alumno
Asunciones	El alumno tiene cuenta de Google.
Pasos	1.A. Acceder a la pantalla de registro mediante el enlace de la pantalla de login. 2.A. El alumno rellena todos los datos solicitados. 3.A. Pulsar el botón para finalizar el registro. 4.S. El sistema registra al nuevo usuario en el sistema.
Variaciones	1.A. Pulsar en el botón de entrar con Google. 1.S.1. El sistema extraerá los datos de su cuenta de Google y procederá a almacenarlas en nuestra base de datos.
Cuestiones	Conocer qué datos vamos a extraer de la cuenta de Google.

Caso de Uso	CdU10 – Recuperar contraseña
Descripción	El alumno se ha olvidado de la contraseña y quiere recuperarla.
Actores	Alumno
Asunciones	El alumno se encuentra registrado.
Pasos	1.A. Rellena el nombre de usuario en la pantalla de login. 2.A. Pincha en el enlace de recordar contraseña de la pantalla de login. 3.S. El sistema envía un correo a la cuenta proporcionada con las credenciales.
Variaciones	
Cuestiones	

Caso de Uso	CdU11 – Registrar alumno
Descripción	Asociar un registro con rol de alumno a un alumno existente de la escuela.
Actores	Alumno
Asunciones	El alumno se encuentra registrado.
Pasos	<p>1.A. Pinchar en la opción correspondiente de menú.</p> <p>2.A. Rellenar los datos que se solicitan en la ficha de un alumno.</p> <p>3.A. Realizar una foto sobre el alumno y adjuntarla para el registro.</p> <p>4.A. Finalizar registro.</p> <p>5.S. El sistema asocia la ficha del alumno al usuario que se encuentra logueado.</p>
Variaciones	
Cuestiones	

Caso de Uso	CdU12 – Visualizar calendario
Descripción	Acceder al calendario para ver cuál es el estado de cada día.
Actores	Alumno
Asunciones	
Pasos	<p>1.A. Entrar mediante la opción correspondiente del menú.</p> <p>2.S. El sistema cargará cada uno de los días y sus correspondientes estados.</p>
Variaciones	
Cuestiones	

Caso de Uso	CdU13 – Votar
Descripción	El alumno votará sobre un menú sobre las alternativas que se ofrecen.
Actores	Alumno
Asunciones	Los menús se encuentran registrados y el estado del día es VOTACIÓN.
Pasos	<p>1.A. Pichar sobre el día que queremos ver.</p> <p>2.A. Seleccionamos uno de los dos menús para votar sobre el mismo.</p> <p>3.S. El sistema aumentará el contador de votos sobre ese menú.</p>

	4.S. El sistema registrará que esa persona ya ha votado ese día.
Variaciones	3.S. La persona ya ha votado y no se admitirá su votación.
Cuestiones	

Caso de Uso	CdU14 – Valorar
Descripción	El alumno quiere valorar el menú.
Actores	Alumno
Asunciones	El día se encuentra en estado VALORACIÓN.
Pasos	<p>1.A. Pinchar sobre el día que queremos realizar la valoración.</p> <p>2.A. Rellenar los datos para dar nuestra opinión sobre el menú.</p> <p>3.A. Enviar valoración.</p> <p>4.S. El sistema registrará la valoración enviada de ese menú.</p> <p>5.S. El sistema registrará que esa persona ya ha realizado la valoración ese día.</p>
Variaciones	4.S. El sistema no admitirá la valoración si la persona ya la ha realizado previamente.
Cuestiones	

Caso de Uso	CdU15 – Ver menú
Descripción	El alumno quiere conocer los detalles de un menú.
Actores	Alumno
Asunciones	El día se encuentra en estado DISPONIBLE o VOTACIÓN.
Pasos	<p>1.A. Entrar en uno de los días.</p> <p>2.A. Pinchamos sobre uno de los menús.</p> <p>3.S. El sistema cargará los datos del menú para ofrecerlos en una nueva pantalla.</p>
Variaciones	
Cuestiones	

6. Diseño de la arquitectura

6.1 Diagrama de clases

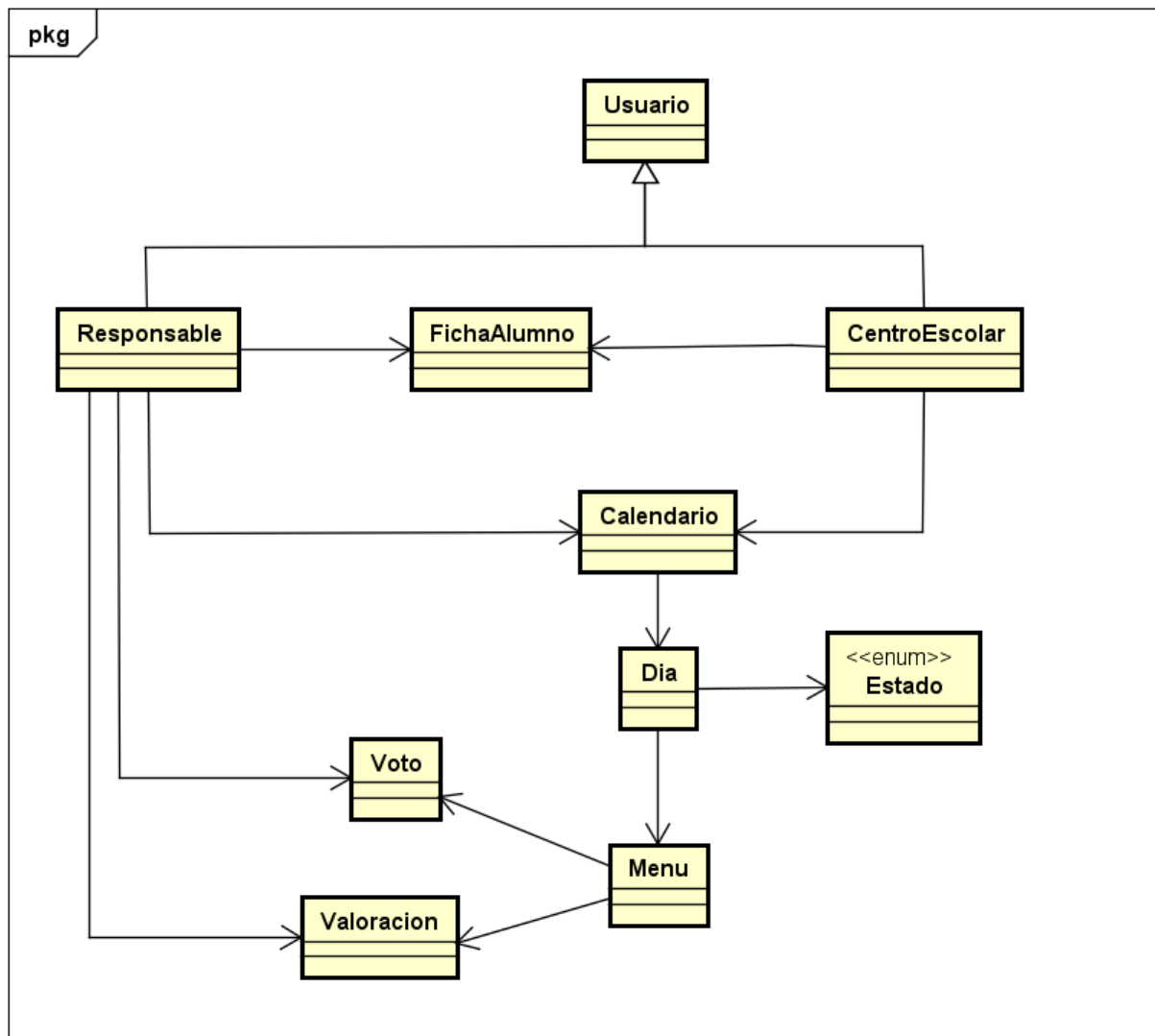


Ilustración 27 Diagrama de clases

- Usuario contiene los datos comunes en el registro.
- Responsable y CentroEscolar son especificación de usuario.
- Estado será un enumerado con valores DISPONIBLE, VOTACIÓN, VALORACIÓN y CERRADO.
- Calendario ejercerá como clase intermedia, para cargar los días que queremos mostrar en esta clase. No se va a persistir.

6.2 Estructura de base de datos

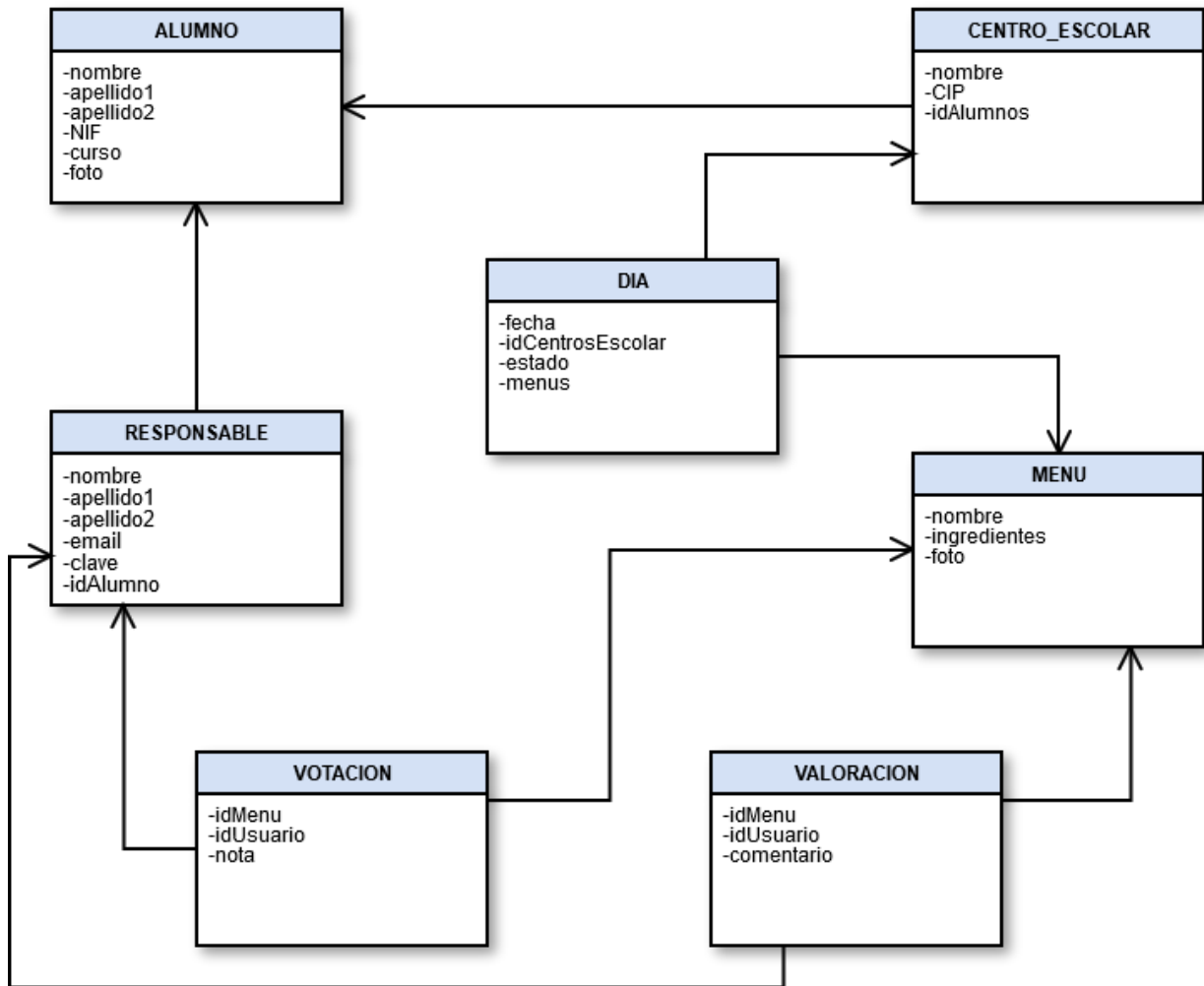


Ilustración 28 Diagrama de base de datos

El esquema se ha realizado pensando en una base de datos no relacional, por lo que tendremos los siguiente:

- Formato de los datos: JSON.
- Producto para trabajar como servidor y cliente: Firebase.

6.3 Revisión del modelo de base de datos y diagrama de clases

Comentamos anteriormente que hubo que darle un giro a la forma de estructurar los datos en Firebase, a modo de intentar el menor número de lecturas y escrituras para agilizar el flujo de la aplicación.

Hay que destacar, que es muy importante la estructura resultante, ya que cuando a un usuario se le dan permisos de lectura o escritura sobre uno de los nodos, se le está dando también sobre todos los subniveles del mismo. Cambiar el modelo es muy costoso, por lo que antes de avanzar conviene tener las ideas claras sobre cómo queremos almacenar los datos.

Otro punto importante, es conocer el funcionamiento de Firebase. Cuando queremos recuperar un dato de esta plataforma, debemos referenciar el nodo que queremos recuperar. Esto quiere decir que, cuando recuperamos un nodo estamos recuperando todos los datos que se encuentra en todos los subniveles del mismo. Se recomienda tener pocos niveles, ya que hará que la aplicación sea más ligera.

Ahora, se explica el nuevo diagrama y para ello damos un color diferente a cada uno de los elementos que se consideran importantes en el nuevo esquema:

- **Negro:** Son las diferentes “tablas” que tenemos en nuestra base de datos. Son los nodos principales, desde los cuales accederemos al resto de datos.
- **Rojo:** Son nuestros identificadores. En este caso se ha hecho uso del correo electrónico, ya que no puede haber dos correos electrónicos iguales en la aplicación. Firebase no permite signos de puntuación en las claves de los nodos, por lo que al correo le hemos quitado tales signos.
- **Morado:** Identifican a cada una de nuestras propiedades. Estas propiedades, suelen constituir una clase de nuestro modelo de clases.
- **Verde:** Son los valores de nuestras propiedades.

Explicamos cada uno de los nodos principales, para conocer rápidamente la estructura final:

- **Alumnos.** Cuando queremos recuperar todos los alumnos de un centro, tan solo tenemos que crear una referencia hacia uno de los nodos clave de segundo nivel. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (centro escolar).
 - Nivel 3: Clave 2 (alumno).
 - Nivel 4: Propiedades y valores.

- **Centros escolares.** Cuando queremos recuperar todos los centros escolares, referenciamos a un nodo de segundo nivel. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (centro escolar).
 - Nivel 3: Propiedades y valores.

- **Menús.** Almacena cada uno de los menús que se registran en la aplicación en el centro y día correspondiente. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (centro escolar).
 - Nivel 3: Clave 2 (día).
 - Nivel 4: Clave 3 (mes).
 - Nivel 5: Clave 4 (año).
 - Nivel 6: Clave 5 (número del menú).
 - Nivel 7: Propiedades y valores.

- **Puntuaciones.** Persistimos cada uno de los usuarios que han valorado un menú, en el día correspondiente. Asegurando que no puedan volver a valorar. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (centro escolar).
 - Nivel 3: Clave 2 (día).
 - Nivel 4: Clave 3 (mes).
 - Nivel 5: Clave 4 (año).
 - Nivel 6: Propiedad (usuario) y valor (true = valora).

- **Usuario Centro.** Establecemos una relación entre los usuarios y el centro escolar. Cuando queremos comprobar a qué centro pertenece un usuario consultamos esta tabla. En la forma antigua, deberíamos haber consultado el centro, qué alumnos pertenecen a este centro y posteriormente, buscar al alumno. Provocaba demasiadas consultadas, ahora consultamos directamente a qué centro y luego directamente al alumno. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (usuario).
 - Nivel 2: Propiedad (centro escolar).

- **Usuario Rol.** Establece una relación entre el usuario y rol que tiene asignado. Consultando directamente el usuario, obtenemos el rol del mismo. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (usuario).
 - Nivel 2: Propiedad (rol).
- **Votos.** Persistimos cada uno de los usuarios que han votado un menú, en el día correspondiente. Asegurando que no puedan volver a votar. Tiene la siguiente estructura:
 - Nivel 1: Nombre de la tabla.
 - Nivel 2: Clave 1 (centro escolar).
 - Nivel 3: Clave 2 (día).
 - Nivel 4: Clave 3 (mes).
 - Nivel 5: Clave 4 (año).
 - Nivel 6: Propiedad (usuario) y valor (true = vota).

Esquema JSON en Firebase:

```
{
  "alumnos" : {
    "jcarrion@acom" : {
      "pemartinez@gmailcom" : {
        "apellido1Alumno" : "Pérez",
        "apellido2Alumno" : "Rox",
        "correoResponsable" : "ruben.pemartinez@gmail.com",
        "dniAlumno" : "88820657G",
        "dniResponsable" : "86510705F",
        "fotoAlumnoEncodedB64" : "iVBORw0KGgoAAAANSUh...",
        "nombreAlumno" : "Ana",
        "parentesco" : "Madre"
      }
    }
  },
  "centrosEscolares" : {
    "scorazon@gmailcom" : {
      "cip" : "232568076C",
      "correo" : "scorazon@gmail.com",
      "nombreCentro" : "Sagrado Corazon"
    }
  },
  "menus" : {
    "scorazon@gmailcom" : {
      "12" : {
        "12" : {
          "2017" : {
            "Primer Menú" : {
              "fechaPublicacion" : "12/12/2017",
              "foto" : "/9j/4AAQSkZJRgABAQEgAQAALAA..."
            }
          }
        }
      }
    }
  }
}
```


Por último, mostrar el diagrama de clases resultante, una vez que se han realizado todas las modificaciones pertinentes. Hay que destacar, que las relaciones no se han establecido en las clases del modelo, sino que se ha estructurado la base de datos de Firebase simulando estas relaciones. Además, si evitamos tener las relaciones identificadas en nuestras clases y las tenemos calculadas, evitaremos recuperar una gran cantidad de datos innecesarios. Ya que, si recuperamos una clase, estamos recuperando los datos que van dentro de la misma, por lo que si tiene clases relacionadas también cargaremos sus datos.

De igual modo, la clase Dia no ha sido implementada, ya que no se ha tenido que realizar otro tipo de operación con ella. En cada situación que hemos hecho uso de la fecha ha sido un dato calculado. La fecha forma parte de las claves del esquema JSON para acceder al nodo concreto que se desee.

Concluimos diciendo, que el cambio que se ha dado de una estructura a otra es bastante grande. La costumbre de trabajar con bases de datos relacionales conlleva a querer estructurar del mismo modo los datos en otro tipo de bases de datos, en este caso noSQL. Finalmente, se cree que se ha conseguido una buena estructura y organización de la información.

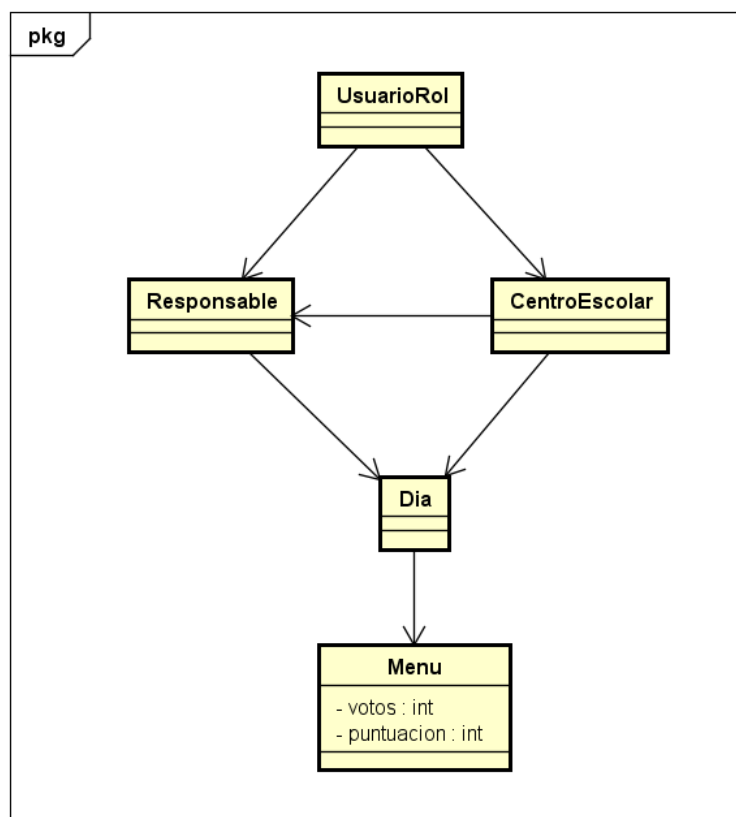


Ilustración 29 Diagrama de clases revisado

6.4 Arquitectura MVC

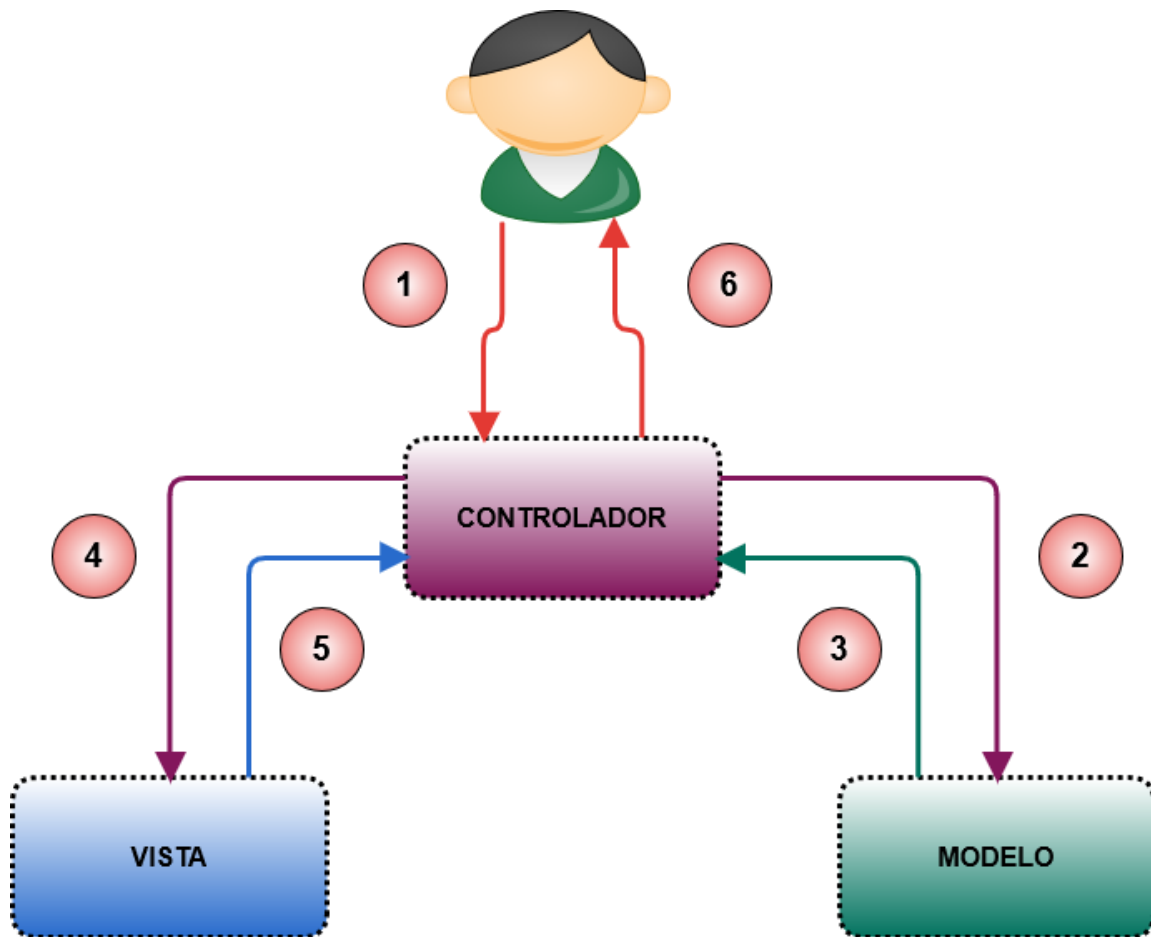


Ilustración 30 Diagrama arquitectura MVC

Detalle por pasos:

1. El usuario envía una petición a la aplicación, al iniciar la aplicación o al interactuar con la misma.
2. El controlador, en nuestro caso cualquier *activity*, solicita al modelo los datos que necesita.
3. El modelo le devuelve los datos solicitados al controlador.
4. El controlador solicita a la vista la pantalla solicitada, en nuestro caso los *layout.xml*.
5. La vista seleccionada se le devuelve al controlador.
6. El controlador le devuelve al usuario la vista, una vez que ha cargado todos los datos que ha cargado del modelo.

7. Implementación

7.1 Configuración gradle

Comenzamos con una breve explicación de la configuración del proyecto. Para ello se aporta la imagen siguiente, que hace referencia a uno de los ficheros importantes:

```
android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "ruben.uoc.comedorescolar"
        minSdkVersion 21
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    compile 'com.google.firebase:firebase-auth:11.6.2'
    compile 'com.google.firebase:firebase-database:11.6.2'
    compile 'com.google.android.gms:play-services-auth:11.6.2'
    compile 'com.android.support:cardview-v7:26.1.0'
    compile 'com.android.support:recyclerview-v7:26.1.0'
}

apply plugin: 'com.google.gms.google-services'
```

Ilustración 31 Configuración Gradle

La idea ha sido trabajar con una API reciente, en este caso con la API26. Como podemos ver en la configuración, por la implementación realizada, la versión mínima soportada es la API 21. Elegir una API u otra, se traduce en una guerra entre hacer uso de las novedades que ofrece Android, con la cantidad de usuarios a los que queremos llegar. Se considera, que esta tecnología crece deprisa, igual que la compra de nuevos terminales, por lo que no se le ha dado importancia al nivel de la API utilizada. De igual forma, mi principal meta u objetivo del máster es y así ha sido, aprender y conocer las novedades que ofrece el mercado.

Como segunda puntualización, se puede observar en el apartado de declaración de dependencias las librerías que hemos tenido que incorporar al proyecto, las cuales nos han proporcionado otros recursos que no vienen incorporados por defecto en el SDK de Android.

Entre ellas podemos destacar, lo que ya mencionamos en puntos anteriores, el uso de Firebase que proporciona el sistema de autenticación y el proceso de almacenamiento de datos. Por otro lado, en aspectos de diseño, hay que destacar el uso de RecyclerView y CardView, dos componentes que nos han permitido realizar un buen diseño en el listado de alumnos. Para la lista se ha utilizado un RecyclerView y para cada uno de los ítems de la lista un CardView. Se ha tenido que diseñar dos CardView diferentes, para poder asignar diferentes estilos y layouts a los elementos pares o impares de la lista.

Por supuesto, he de mencionar que para hacer uso de Firebase se debe de disponer de los servicios de Google Play.

7.2 Configuración manifest

Mostramos la configuración completa en la imagen siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ruben.uoc.comedorescolar">

    <uses-feature
        android:name="android.hardware.camera"
        android:required="false" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Comedor Escolar"
        android:supportsRtl="true"
        android:screenOrientation="portrait"
        android:theme="@style/AppTheme">
        <activity android:name=".activities.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".activities.RegistroActivity" />
        <activity android:name=".activities.MenuActivity" />
        <activity android:name=".activities.ListadoAlumnosActivity" />
        <activity android:name=".activities.AlumnoPerfilActivity" />
        <activity android:name=".activities.RegistroMenuActivity" />
        <activity android:name=".activities.CalendarioActivity" />
        <activity android:name=".activities.DiaCalendarioActivity" />
        <activity android:name=".activities.ValoracionActivity" />
    </application>
</manifest>
```

Ilustración 32 Configuración de la aplicación manifest.xml

En el fichero manifest.xml tenemos la configuración de la aplicación. Aquí, indicamos los recursos hardware y software que podríamos necesitar para el buen funcionamiento de la aplicación, por ejemplo, en nuestro caso la cámara de fotos.

La aplicación se compone de un total de 9 actividades y se indica cuál es la actividad principal, lanzándose así al iniciar la aplicación. También establecemos que será ejecutada en modo vertical, por lo que si un usuario gira su dispositivo móvil se cargará el mismo layout, ya que no diferenciamos entre las distintas orientaciones de pantalla.

Se ha diseñado una imagen personalizada para el logo, la cual será el icono de la aplicación una vez que haya sido instalada en un dispositivo. Se proporciona en todos los tamaños solicitados, para que se adapte a las distintas resoluciones de pantalla. En este mismo fichero, indicamos cuál es el icono que debemos y dónde lo debe de localizar cuando necesite cargarlo.

7.3 Firebase

7.3.1 ¿Por qué Firebase?

Para el funcionamiento correcto de la aplicación, la mayoría de los datos tienen que ser compartidos en red, por lo tanto, se necesita una parte de servidor donde podamos almacenar cada uno de los datos. Posteriormente, esos datos también tienen que poder ser recuperados.

El método principal para persistir los datos en los dispositivos Android es mediante el motor de base de datos *SQLite*. Android proporciona las herramientas necesarias para la creación y la gestión. Una base de datos embebida en el dispositivo que podemos atacar mediante consultas *SQL*. En este caso los datos serían almacenados en local, por lo que por esta razón esta opción ha tenido que ser descartada.

Por otro lado, Android ofrece la clase *SharedPreferences* para almacenar las preferencias del usuario. Se utiliza normalmente para almacenar parámetros de configuración o las credenciales del usuario una vez que ya las introdujo por primera vez, donde los datos se almacenan en pares clave-valor. No se utiliza para almacenar cantidades grandes de datos y también los datos se quedan registrados localmente, por lo tanto, esta opción también ha quedado descartada. Sin embargo, aunque no es el caso, se podía haber utilizado este recurso para almacenar las credenciales del usuario, para aquellos usuarios que no hacen uso del login mediante la cuenta de Google y tienen que introducir siempre el usuario y la contraseña.

Una vez descartado lo que Android nos ofrece, se pensó en el desarrollo de un servicio web *REST* implementado en *Java*. Posteriormente, este servicio web sería atacado desde la aplicación Android mediante la librería *Retrofit*, ya que es un buen cliente *HTTP* para Android y *Java*. Los datos serían devueltos en formato *XML* o *JSON* y serían tratados de forma parecida a como se ha hecho finalmente con *Firebase*, asociándose a una de las clases del modelo de datos. Se realizó una búsqueda de un servidor Tomcat para la publicación de este servicio web. No quedé satisfecho debido a que está muy limitada la oferta de servicios gratuitos que ofrezcan este servicio. Además, quedaría de nuestra cuenta administrar este servidor. Al final esta opción también quedó descartada, personalmente es la que más me gusta, pero el

tiempo limitado para el desarrollo de este proyecto y el interés en centrar totalmente el desarrollo en una aplicación Android, fueron motivos suficientes para buscar otras soluciones.

Después de este análisis, centramos la búsqueda en los diferentes productos que permiten almacenar datos en la nube, entre los que destacamos *Google Cloud*, *Azure*, *Heroku*, *AWS* y *Firebase*. Algunos de estos productos los analizamos brevemente a continuación:

- Google Cloud.** Google Cloud SQL es un servicio totalmente administrado que facilita la configuración, el mantenimiento y la administración de las bases de datos MySQL relacionales en la nube. Su segunda generación de Cloud SQL ofrece altos niveles de rendimiento, escalabilidad y comodidad. Como se aloja en Google Cloud Platform, proporciona una infraestructura de base de datos para aplicaciones que se ejecutan en cualquier lugar [10]. La tabla de precios por el uso de este producto es la siguiente:

ELEMENTO	PLAN POR USO	PLAN POR PAQUETE
Instancia de Cloud SQL <small>de 125 MB a 16 GB de RAM</small>	De 0,025 a 2,31 USD por hora	De 0,0150 a 1,464 USD por hora ¹
Almacenamiento	0,024 USD por GB al mes	El precio del paquete incluye de 500 MB a 10 GB por instancia ²
Red	<ul style="list-style-type: none"> Entrada: gratis Salida: 0,12 USD por GB 	<ul style="list-style-type: none"> Entrada: gratis Salida: 0,12 USD por GB
Dirección IPv4 de la instancia	0,01 USD por hora de inactividad	0,01 USD por hora de inactividad

Ilustración 33 Tabla de precios de Google Cloud

- Azure.** Azure Database for MySQL proporciona un servicio de base de datos administrado para el desarrollo y la implementación de aplicaciones que permite configurar una base de datos MySQL en cuestión de minutos y escalarla sobre la marcha, todo ello en la nube que goza de su confianza. Con modelos de precios inclusivos, obtiene toda la funcionalidad que desea (como alta disponibilidad, seguridad y recuperación) de forma integrada, sin costo adicional [11]. Este producto no se ofrece de forma gratuita, salvo que sea el primer registro y se ofrezcan todas las opciones por un tiempo limitado (30 días). A continuación, se ofrece la tabla de precios:

Básico

For throughput up to 100 compute units with variable IOPS.

UNIDADES DE PROCESO	PRECIO DE VISTA PREVIA
50	€0,018/hora
100	€0,035/hora

STORAGE	OPERACIONES DE E/S POR SEGUNDO APROVISIONADAS INCLUIDAS	PRECIO DE VISTA PREVIA
Primeros 50 GB	–	Incluido
Cada 125 GB adicionales (hasta 1 TB)*	–	€0,061/GB/mes

Ilustración 34 Tabla de precios de Azure

- **AWS.** Amazon DynamoDB es un servicio de base de datos NoSQL y flexible para todas las aplicaciones que requieren latencias constantes y de meros milisegundos a cualquier escala. Se trata de una base de datos totalmente administrada en la nube, compatible con modelos de almacenamiento de datos de valor de clave y de documentos. Su modelo de datos flexible, desempeño fiable y escalado automático de la capacidad de desempeño lo hacen ideal para móviles, web, juegos, tecnología publicitaria, IoT y mucho más [12]. Amazon DynamoDB Accelerator (DAX) es una memoria caché totalmente administrada y de gran disponibilidad que puede reducir los tiempos de respuesta de Amazon DynamoDB de milisegundos a microsegundos, incluso con millones de solicitudes por segundo. Este servicio es gratuito mientras no se sobrepasen los límites establecidos. Lo vemos:

The image shows two panels from the AWS DynamoDB pricing page. The left panel, titled 'BASE DE DATOS DynamoDB', highlights '25 GB de almacenamiento' and describes it as a 'Base de datos NoSQL rápida y flexible con una escalabilidad perfecta'. It includes a link to 'Obtenga más información sobre DynamoDB'. The right panel, also titled 'DynamoDB', lists '25 GB de almacenamiento', '25 unidades de capacidad de escritura', and '25 unidades de capacidad de lectura'. It states 'Suficiente para manejar hasta 200 millones de solicitudes al mes.' and features a 'PRUEBE DYNAMODB DE FORMA GRATUITA' button. Below the button, it notes 'No vence al final del periodo de 12 meses de la capa gratuita de AWS. ¿Cuál es la diferencia entre ofertas que vencen y ofertas que no vencen? VER LAS PREGUNTAS FRECUENTES'.

Ilustración 35 Tabla de precios de AWS

- **Firebase.** Firebase Realtime Database es una base de datos NoSQL alojada en la nube que te permite almacenar y sincronizar datos entre tus usuarios en tiempo real. La sincronización en tiempo real permite que los usuarios accedan a sus datos desde cualquier dispositivo, web o móvil, y los ayuda a trabajar en conjunto. Realtime Database se incluye en los SDK para dispositivos móviles y web, de manera que puedas crear apps sin la necesidad de usar servidores. También puedes ejecutar un código de back-end que responda a los eventos que se activan con tu base de datos a través de Cloud Function para Firebase. Cuando los usuarios se desconectan, los SDK de Realtime Database usan la caché local del dispositivo para publicar y almacenar cambios. Cuando el dispositivo se conecta, los datos locales se sincronizan de manera automática [4]. Este servicio se ofrece de forma gratuita, mientras no se sobrepasen los límites establecidos por la plataforma. Podemos ver el plan básico del producto en siguiente imagen:

Productos	Plan Spark Límites generosos para aficionados Sin cargo
Incluido sin cargo Authentication (except Phone Auth), Analytics, Predictions, App Indexing, Dynamic Links, Invites, Remote Config, Cloud Messaging (FCM), Performance Monitoring, Crash Reporting, and Crashlytics.	✓ Incluidos
Realtime Database Conexiones simultáneas ? GB almacenados GB descargados Multiple databases per project	100 1GB 10 GB/mes ✗

Ilustración 36 Tabla de precios de Firebase

Una vez realizado este pequeño análisis, hemos descartado directamente las opciones de Google Cloud y Azure, porque no ofrecen versión gratuita. Estas dos plataformas, te ofrecen todos sus productos abiertos por un tiempo limitado de prueba o directamente con una cartera previamente cargada, de forma gratuita, hasta que el uso de los productos la dejen a cero.

Dejando como posibles soluciones para nuestro proyecto a las plataformas AWS y Firebase, nos hemos decantado por Firebase. Este producto ha sido utilizado en varias asignaturas del máster, por lo que la curva de aprendizaje para el uso de otros productos de la misma plataforma se considera menor. Además, visto otros productos de la misma plataforma, se entiende que serán de provecho para el desarrollo de la aplicación. En conjunto, se ha pensado que se le va a dar un mejor rendimiento a esta plataforma.

Debido a que su uso no conlleva ningún coste, se incluyó en el apartado 1.7 de viabilidad económica, con el resto de opciones seleccionadas para el desarrollo de la aplicación. En el siguiente apartado se explicará detalladamente cómo lo hemos incorporado a nuestro proyecto.

7.3.2 Firebase en la aplicación

Esta plataforma se elige porque nos proporciona los recursos que necesitamos. Evidentemente, para conocer dicha plataforma ha sido necesario emplear un tiempo considerable en las lecturas de la documentación de sus productos. Además, nos encontramos con un problema a la hora de almacenar los datos, ya que el diseño de la estructura principal estaba orientado a una base de datos relacional. Por lo tanto, se ha tenido que repasar, comprender y profundizar en una forma correcta de persistir los datos en formato *JSON* en una base de datos *noSQL* y rediseñar el esquema de base de datos, para intentar realizar el mínimo de lecturas o escrituras hacia la base de datos, de manera que pueda ser lo más eficiente posible.

Para poder interactuar con el producto Realtime Database y realizar escrituras o lecturas debemos de hacer uso de la instancia de base de datos “FirebaseDatabase” y crear referencias a nuestros datos con “DatabaseReference”. Esto lo conseguimos de la siguiente forma:

```
private FirebaseDatabase database;
private DatabaseReference myRef;

// Instancia de BBDD Firebase.
database = FirebaseDatabase.getInstance();

// Referencia a los recursos de BBDD.
myRef = database.getReference("usuarioRol/" + user.getEmail().replace(".", ""));

// Se lee de base de datos (una sola vez) y no permanece en escucha.
myRef.addListenerForSingleValueEvent(new ValueEventListener() {
```

Nos beneficiamos del sistema de autenticación que nos proporciona esta plataforma, el cual es mediante correo electrónico y contraseña. También, aprovechamos que permite realizar una autenticación con terceros, por lo que hemos abierto la alternativa al usuario del registro/autenticación en nuestra aplicación mediante una cuenta de Google.

La autenticación a través de la cuenta de Google se produce gracias al protocolo de autorización OAuth2. Permite a terceros (nuestra aplicación) acceder a los contenidos o datos de un usuario en otro proveedor (Google), sin que se tenga que conocer las credenciales del mismo. Previamente, el usuario debe haber dado permiso para ello, por lo que se produce una negociación entre el cliente y el servidor de recursos, para la generación de un token que irá en el cuerpo de cada petición. De este modo, se genera la confianza entre las partes implicadas en la transferencia de datos.

A continuación, se muestra un ejemplo de los pasos a seguir para establecer la autenticación con una cuenta de Google, de forma muy resumida. Estos pasos han sido adecuados a nuestra aplicación de modo y orden correcto.

```
private FirebaseAuth mAuth;
private FirebaseUser user;

private static final int RC_SIGN_IN = 9001;
private GoogleSignInClient mGoogleSignInClient;
private GoogleSignInAccount cuentaGoogle;

// Instancia de Firebase.
mAuth = FirebaseAuth.getInstance();

// Configuración del login con Google
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestEmail()
    .build();
```

```

mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

// Iniciamos la actividad de Google para obtener la cuenta del usuario.
Intent signInIntent = mGoogleSignInClient.getSignInIntent();
startActivityForResult(signInIntent, RC_SIGN_IN);

// Acceso con las credenciales
AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(),
null);
mAuth.signInWithCredential(credential).addOnCompleteListener(this,
new OnCompleteListener<AuthResult>() {
...

```

Garantiza que el registro se realice mediante una cuenta de correo correcta, ya que se ha implementado la necesidad de que la cuenta sea verificada para poder hacer uso de la aplicación. De este modo, se evita tener datos guardados o cuentas falsas que acaban molestando y ocupando una serie de espacio que finalmente queda obsoleto.

Para la verificación, inmediatamente después de que un usuario se registre en la aplicación, se le envía un correo con un enlace que permite verificar su cuenta. Una vez que la cuenta ha sido verificada, el usuario podrá entrar libremente y hacer uso de ella. Este sistema, también nos asegura que no haya dos cuentas de correo idénticas registradas, gracias al producto “Authentication” de Firebase. En nuestro caso, hemos aprovechado esta situación, para usar las cuentas de correo como identificadores de los usuarios y los centros escolares.

Por último, el sistema de recuperación de contraseña también se ha implementado con los recursos que nos ofrece esta plataforma. De modo que cuando un usuario no recuerda una contraseña, mediante su correo electrónico podrá modificarla siguiendo los siguientes pasos:

1. Clic en “¿Ha olvidado su contraseña?”, opción que se encuentra en la parte inferior de pantalla principal.
2. El usuario recibe un correo electrónico en la cuenta que ha sido introducida en esa misma pantalla. Como requisito previo, la cuenta debe estar registrada en la aplicación.
3. Mediante el enlace que se encuentra en el correo electrónico, se proporciona una ventana para el cambio de contraseña. No se almacena la contraseña en claro en la base de datos, ya que siempre hay que intentar preservar el derecho de la privacidad de los usuarios. Así que, con este sistema garantizamos que el usuario es quien dice ser.
4. La contraseña es cambiada y confirmada, mediante el servicio REST de Firebase.
5. El usuario puede volver a entrar con sus nuevas credenciales.

Como hemos podido observar esta plataforma ofrece una gran variedad de productos. En muchos proyectos, la elección de una tecnología u otra puede ser determinante en los tiempos de desarrollo. Una buena planificación y estudio de lo que se necesita, se quiere y lo que ofrece el mercado es muy importante antes de comenzar con la implementación.

7.3.3 Opciones

Firestore ofrece distintos modos de autenticación, entre los que se han elegido la cuenta de Google y correo electrónico con contraseña. Se toma esta decisión debido a que, a mi parecer, son las opciones más formales para una aplicación. Sin embargo, existen distintos proveedores que podrían ser añadidos en cualquier momento y Firestore proporciona el sistema de autenticación. Además de la implementación correspondiente por cada proveedor, hay que habilitarlos en la plataforma. Lo vemos en la siguiente imagen:








Proveedor	Estado
 Correo electrónico/contraseña	Habilitada
 Teléfono	Inhabilitado
 Google	Habilitada
 Facebook	Inhabilitado
 Twitter	Inhabilitado
 GitHub	Inhabilitado
 Anónimo	Inhabilitado

Ilustración 37 Proveedores de autenticación Firestore

También, ofrece un control de los usuarios y de los fallos. Podemos ver en la siguiente imagen, los usuarios activos diarios y mensuales, como el porcentaje de fallos que se han producido.

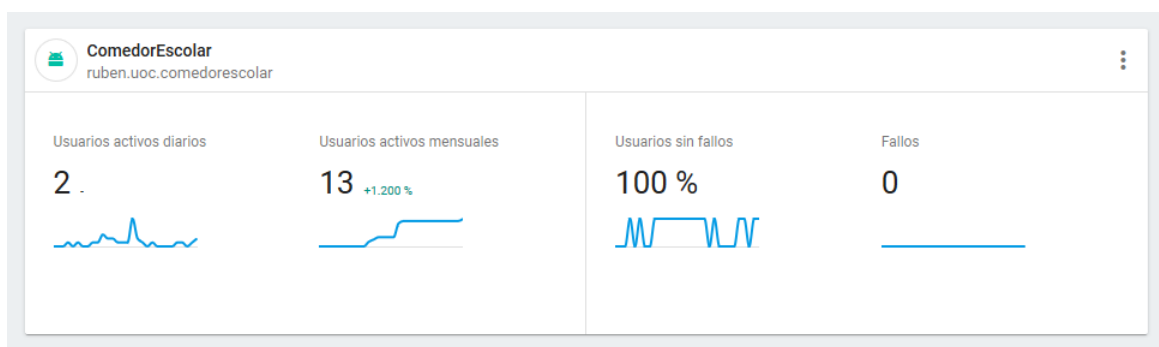


Ilustración 38 Estadística de usuarios y fallos

Por otro lado, ofrece la posibilidad de añadir usuarios manualmente, sin necesidad de hacer el registro por la aplicación. Igualmente, indica cada uno de los usuarios registrados y con qué proveedor lo han hecho. Por motivos de privacidad, en ningún momento se puede obtener la contraseña de un usuario. En la imagen siguiente, observamos que existen tres usuarios registros, dos de ellos mediante cuenta de Google y uno mediante correo electrónico y contraseña.



Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
[redacted]@gmail.com		21 nov. 2017	27 dic. 2017	KvemHLjhZ9SeaDva6Pof7Wse6st1
[redacted]@hotmail.com		26 dic. 2017	26 dic. 2017	nxWpl06VSudAuffyWg88GlcvnM3P2
[redacted]@gmail.com		26 dic. 2017	26 dic. 2017	ohKQFE4IUEORcJiKIL25KB96BH2

Filas por página: 50 1-3 de 3

Ilustración 39 Control de usuarios registrados

7.4 Decisiones generales

Por motivos de diseño y una mejor visualización, los botones en la aplicación son de tipo *TextView*. De este modo se ha podido realizar una mejor personalización y mejora visual de un “botón”. Para ello, también se ha hecho uso de un recurso de tipo *shape* rectangular incorporado a la carpeta *drawable* en formato XML, para dibujar los bordes a cada uno de los botones. Aquí, indicamos el borde, el tamaño del borde y el color del background del botón.

Para dar una mayor homogeneidad a la aplicación, se ha diseñado un fondo propio y se ha incorporado en cada uno de los layouts.

Se ha utilizado un tema general sin *Action Bar*, ya que se considera que para nuestra aplicación no ha sido necesario mostrar una barra de título o menú de opciones disponibles para el usuario. Nos hemos beneficiado de un poco más de espacio, por lo que en cada una de las pantallas hemos establecido un título a nuestro gusto incorporado en la misma pantalla. En un futuro se podría añadir en la aplicación, para integrar alguna opción de menú, como puede ser la de cerrar sesión. De modo que estés en la pantalla que estés, se le brinde la oportunidad al usuario de cerrar sesión y salir de la aplicación. Al mismo tiempo, se puede meditar lo de incorporar el título a la propia barra, diferenciándolo así de la propia pantalla.

En el registro de los alumnos se ha hecho uso de la cámara fotográfica del dispositivo. Se obliga al usuario a hacer una fotografía para poder avanzar en el registro. Una vez que la fotografía se ha realizado correctamente, recuperamos la foto codificada en su versión miniatura como *Bitmap* de los datos extras mediante la clave “data” en el método *onActivityResult()*. La cámara se consigue que sea activada mediante el siguiente código:

```
private void hacerFofa() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(takePictureIntent, REQUEST_CODE_CAMERA);
    } else {
        Toast.makeText(this, "La cámara no está disponible.",
            Toast.LENGTH_LONG).show();
    }
}
```

Sin embargo, para variar, en el registro de los menús se debe recuperar una foto mediante uno de los proveedores de almacenamiento del dispositivo, como la Galería o Google Drive. Esto se consigue mediante las siguientes instrucciones:

```
private void seleccionarFoto() {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("image/*");
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    startActivityForResult(intent, REQUEST_IMAGE);
}
```

Para el listado de los alumnos se ha personalizado un componente *RecyclerView* de modo que podamos jugar con cada uno de los elementos de la lista. Se han creado dos layouts distintos, para darle una vista diferente a los elementos de la lista, dependiendo de si son elementos pares o impares. Esta asociación la conseguimos realizar en el método *onCreateViewHolder()* del *RecyclerView*. Lo vemos a continuación:

```
@Override
public AlumnoViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View v;
    switch (viewType) {
        case EVEN:
            v =
                LayoutInflater.from(parent.getContext()).inflate(R.layout.card_view_alumno_p
                    ar, parent, false);
            break;
        default:
            v =
                LayoutInflater.from(parent.getContext()).inflate(R.layout.card_view_alumno_i
                    mpar, parent, false);
            break;
    }

    return new AlumnoViewHolder(v);
}
```

En el listado de centros escolares, que podemos encontrar en el registro de un alumno, hemos utilizado un componente *Spinner* con un adaptador personalizado. De este modo, se ha conseguido darle un aspecto personalizado a los elementos de la lista cuando son desplegados o cuando han sido seleccionados. Personalizar el elemento seleccionado se consigue de la siguiente manera:

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    TextView label = new TextView(context);
    label.setTextColor(Color.WHITE);
}
```

```
label.setText(centrosEscolares.get(position).getNombreCentro());  
return label;  
}
```

Y personalizar los elementos desplegables de esta otra:

```
@Override  
public View getDropDownView(int position, View convertView, ViewGroup  
parent) {  
    TextView label = new TextView(context);  
    label.setTextColor(Color.BLUE);  
    label.setText(centrosEscolares.get(position).getNombreCentro());  
    return label;  
}
```

En las vistas que se ha diferenciado funcionalidad dependiendo de un rol u otro, de modo que han sido ocultados ciertos componentes gráficos, se ha hecho mediante el valor *GONE*, de modo que no ocupen un lugar en nuestra vista. De modo que, los componentes que tienen que ser recuperados cuando los necesitamos porque así el rol lo requiere, entonces establecemos el estado de visibilidad del componente a *VISIBLE*, ocupando un lugar en nuestro layout. Por defecto, ocultamos la visibilidad de los componentes directamente en el componte dentro del fichero XML del layout, mediante la instrucción siguiente:

```
android:visibility="gone"
```

Sin embargo, la visibilidad la restablecemos de forma programática en tiempo de ejecución. Esto se hace mediante la instrucción siguiente:

```
findViewById(R.id.contenedor_registro_centro).setVisibility(View.VISIBLE);
```

El patrón MVC queda claramente reflejado en la estructura del proyecto. Las clases del modelo de datos han sido guardadas en una carpeta denominada *model*. Las vistas son los layouts, que se encuentran almacenados en el directorio *res/layout*. Por último, nuestras actividades son nuestros controladores que son los encargados de cargar o almacenar cada uno de los datos sobre la base de datos y enviar o recibir las acciones de la vista.

Al trabajar con las imágenes y una base de datos que almacena los datos en formato *JSON*, ha sido necesario transformar cada una de ellas a un tipo de dato *String* en formato *Base64*. En caso contrario, cuando las hemos tenido que recuperar, las hemos tenido que decodificar para transformarlas en elementos *Bitmap* y poder asignarla al *ImageView* correspondiente.

La codificación de la foto a base 64 para poder persistirla en base datos se realiza de la siguiente manera:

```
ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();  
fotoAlumno.compress(Bitmap.CompressFormat.PNG, 100, byteArrayOutputStream);  
byte[] fotoAlumnoByteArray = byteArrayOutputStream.toByteArray();  
String fotoAlumnoEncodedB64 = Base64.encodeToString(fotoAlumnoByteArray,  
Base64.DEFAULT);
```


Vemos que las instrucciones hablan por sí solas. Generamos un stream de salida, comprimimos la foto del alumno que es de tipo `Bitmap` y codificamos los bytes resultantes a `Base64`. Hay que destacar que la foto se mantiene en la calidad inicial, se puede bajar la calidad de la misma según se quiera y según cuál sea el objetivo de la foto.

El paso inverso, recuperar la foto desde base de datos y asociarla a un componente `ImageView` se consigue así:

```
String fotoB64 = alumno.getFotoAlumnoEncodedB64();
byte[] fotoAlumnoByteArray = Base64.decode(fotoB64, Base64.DEFAULT);
Bitmap fotoAlumnoBitmap = BitmapFactory.decodeByteArray(fotoAlumnoByteArray,
0, fotoAlumnoByteArray.length);

ImageView imagenAlumno = findViewById(R.id.perfil_foto_alumno);
imagenAlumno.setImageBitmap(fotoAlumnoBitmap);
```

`Alumno`, corresponde a una clase del modelo que tiene una propiedad de tipo `String` en la que se recupera la foto codificada en `Base64`. Posteriormente, se decodifica para recuperar los bytes de la foto y transformarla a una imagen de tipo `Bitmap`. Por último, buscamos el componente `ImageView` y establecemos la foto que acabará mostrándose por pantalla.

Otro aspecto importante que hay que destacar es que para trabajar con la base de datos de `Firestore`, almacenando y recuperando objetos, éstos deben implementar la interfaz `Serializable`. Esto es necesario para que `Java` pueda convertir dicho objeto en una ristra de bytes y pueda viajar a través de la red. Igualmente, para que posteriormente pueda ser recuperado. Esta interfaz no contiene métodos, por lo que lo único que hay que hacer es simplemente implementarla.

Para la transferencia de objetos `Java` en la plataforma de `Firestore`, es obligatorio que la clase que contiene los datos tenga un constructor por defecto. Esta clase, debe tener los métodos correspondientes de obtención de datos de carácter público, para que así se asignen las propiedades. Cuando se almacena un objeto, se asigna automáticamente a las ubicaciones secundarias de forma anidada.

8. Información y pruebas

Una vez terminada la aplicación y descargada en un dispositivo móvil, comprobamos que el espacio total que ocupa es de 2,66 MB en disco:

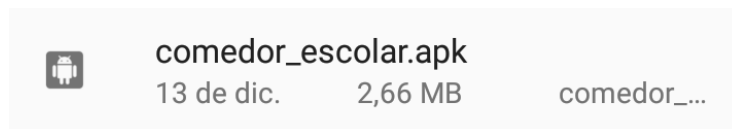


Ilustración 40 Tamaño aplicación apk

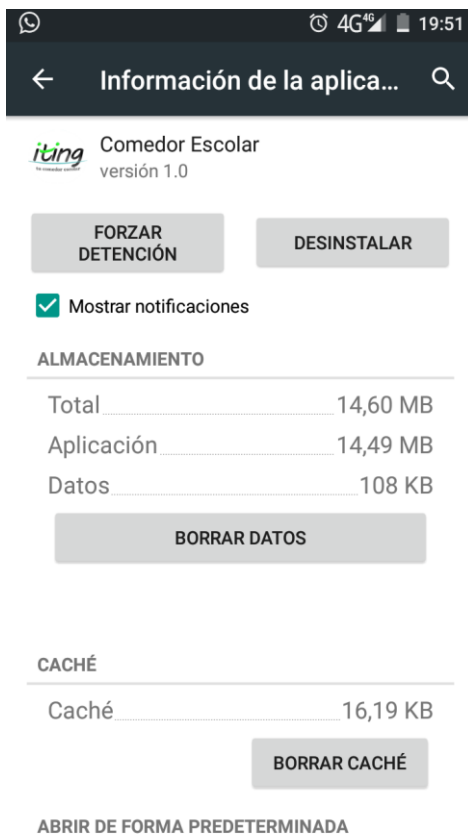


Ilustración 41 Información aplicación instalada

Debido a que el tiempo es limitado para el desarrollo del proyecto, se tomó la decisión de no realizar tests de pruebas mediante software de pruebas, entre los que se encuentran habitualmente *JUnit*, *Espresso* o *Mockito*. Aunque es recomendable realizar este tipo de pruebas, ya que garantizan testear muy bien una aplicación, incluso su interfaz.

Sin embargo, para intentar realizar una aplicación de calidad se han realizado las pruebas en el simulador de Android y un dispositivo real. Los diferentes errores, se han corregido a base de pruebas y error. Los detalles de estos dispositivos los mostramos a continuación:

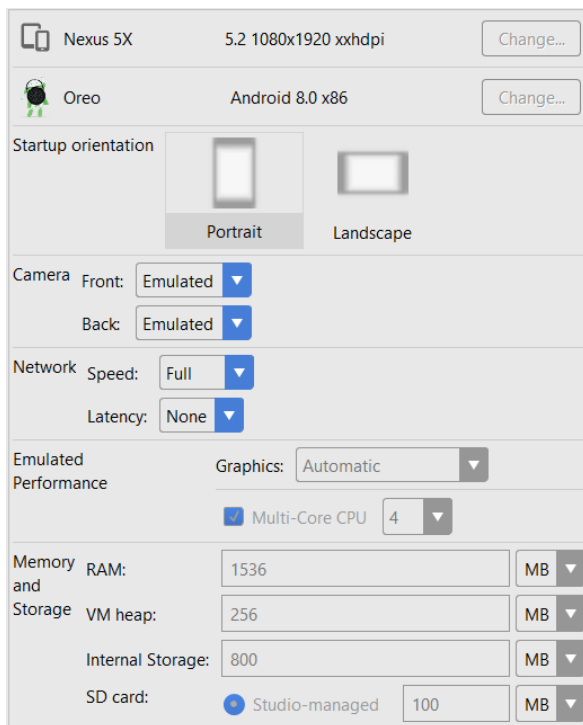


Ilustración 42 Información Emulador

Tipo de dispositivo: Emulador

Nombre versión: Oreo

Nivel de API: 26

Versión del sistema operativo: 8.0

Tipo de dispositivo: Real

Nombre versión: Lollipop

Nivel de API: 21

Versión del sistema operativo: 5.1

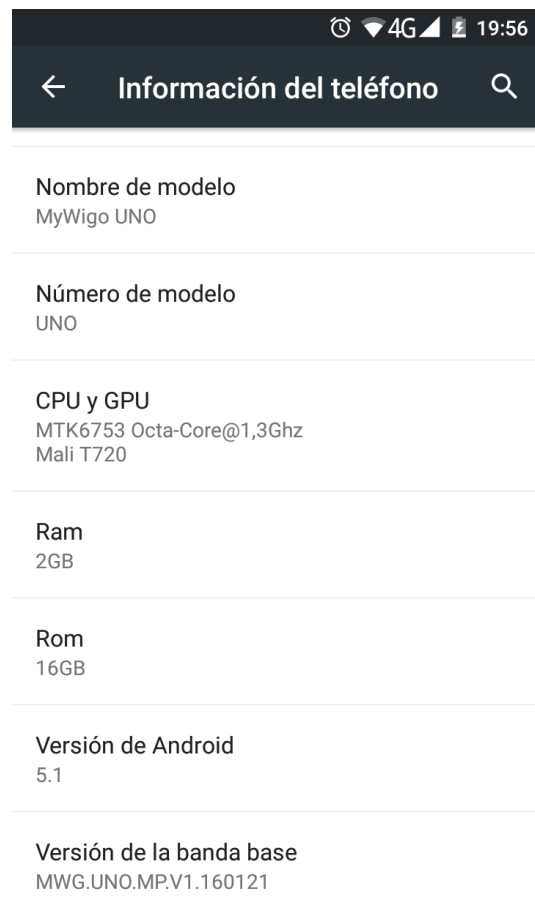


Ilustración 43 Información dispositivo real

9. Vías futuras

A continuación, vamos a exponer las posibles vías futuras que pueden realizarse sobre este proyecto.

- Se debe mejorar el control de las validaciones de los datos. Restringiendo longitudes de campos o verificando alguno de los datos que se introducen, como por ejemplo DNI.
- Introducir un control de alergias. Cada uno de los ingredientes de los menús deberán pasar un control con cada uno de los alumnos que tengan alergia a un alimento.
- Sistema de alertas. Cuando uno de los ingredientes coincide con un alumno que tiene problemas alimenticios, se deberá notificar al responsable del mismo automáticamente o de forma manual.
- Se pueden añadir más tipos de autenticación, como Facebook. Cuantas más facilidades se le dé al usuario, menos posibilidad de rechazo generar la aplicación.
- Realizar un análisis de ventajas e inconvenientes sobre los diferentes tipos de bases de datos, entre las que se encuentran las bases de datos relacionales o bases de datos noSQL. Optimizando así cada una de las consultas o datos y haciendo un análisis de tiempos de carga.
- Debido a los dos roles claramente diferenciados, se podría separar en dos aplicaciones diferentes, para su futura evolución por separado. De modo que cada una de ellas quede más ligera para el usuario, los tiempos de carga serán menores, el flujo entre pantallas se mejorará. Así, también será más fácil para el desarrollador identificar la funcionalidad correspondiente a cada uno de ellos.
- Implementar un control de pago sobre el servicio de comedor. Se le puede facilitar al usuario a pagar el comedor mediante la aplicación. De igual modo, se le notificará cuando su suscripción esté a punto de caducar. Se la cuota mensual se carga de forma automática se deberá notificar al usuario que el pago ha sido recibido.
- Añadir Crash Reporting, un producto más de los que ofrece la plataforma de Firebase. Reporta los fallos de los usuarios, así podremos detectarlos con facilidad y corregirlos lo antes posible. Tendremos estadísticas que serán de gran ayuda.

10. Conclusiones

El desarrollo de un proyecto, como hemos podido comprobar, no es sentarse y comenzar a teclear código sin parar. Cuando se quiere obtener un resultado de calidad hay que analizar detenidamente cuál es el objetivo y cómo se llega hasta él. La forma de afrontar un proyecto es clave para diferenciar cada uno de los pasos que se da.

Adquirimos una responsabilidad cuando nos embarcamos y comprometemos en el desarrollo de un proyecto. Cada uno de los pasos son totalmente necesarios, ya que antes de comenzar con la implementación de la aplicación, se corrigen muchos de los errores que se puede llegar a cometer.

Un punto importante que la gente en muchas ocasiones se olvida de realizar es el estudio de mercado sobre la idea que ha surgido. No tiene sentido el inicio de un proyecto, si el mercado te ofrece una gran variedad de productos similares, porque en un alto porcentaje uno está destinado al fracaso. En este caso, ya puede desarrollarse un proyecto de gran calidad, que se noten las diferencias y, sobre todo, lo más importante es la forma de vender tu producto. Por lo tanto, hemos aprendido que este análisis previo nos ayuda a conocer qué ofrece el mercado y cuáles son las debilidades y fortalezas de lo que posteriormente será la competencia.

Cuando finalmente se acepta el desafío de desarrollar la idea, hay otros aspectos importantes que pueden influir a corto, medio y largo plazo. Hay varias preguntas que uno debe hacerse antes de comenzar... ¿cuál es el entorno tecnológico que queremos? ¿cuál es el presupuesto que tenemos? ¿qué exigencias tenemos la obligación de cumplir? Uno aprende a que hay que realizar un estudio sobre los sistemas actuales y cuál es el crecimiento que sigue cada uno de ellos. Gracias a este estudio, se toma la decisión de realizar el desarrollo para la plataforma Android, ya que posee una cuota alta del mercado e incluso se prevé que siga al alza.

Por otro lado, posteriormente a la decisión sobre la elección de la plataforma hacia la que enfocar nuestro producto, se realiza la viabilidad económica. Hay que estudiar que recursos tenemos o debemos de adquirir para que el desarrollo se realice sin ningún contratiempo y en caso contrario, subsanar correctamente y en el menor tiempo posible. Se realiza el análisis de las diferentes plataformas antes de tomar la decisión de trabajar con Firebase. Cada una de ellas ofrece distintos productos, por lo que influye directamente en mucha de la funcionalidad que se tiene por objetivo. Dependiendo del producto elegido, ganaremos tiempo en el desarrollo de cierta funcionalidad y si nos equivocamos, causaremos el efecto contrario. Es de vital importancia saber elegir las tecnologías con las que se quiere trabajar y tener claro hasta dónde podemos llegar con cada una de ellas.

Precisamente, se tuvo un problema al trabajar con la plataforma de Firebase. Aunque se tenía claro que esta plataforma trabaja con una base de datos noSQL, se cometió el error de querer trasladar la idea de almacenar los datos como si fuera una base de datos relacional. Esto derivaba en una gran cantidad de lecturas o escrituras innecesarias, del mismo modo que

la estructura de datos no acababa de ser coherente. Se tuvo que profundizar sobre la idea de trabajar con una base de datos no relacional y trabajar con un esquema de datos JSON. Una vez solventado este problema, todo siguió su curso con normalidad. Debo decir, que esto provocó una pérdida de horas innecesaria, que se podría haber empleado para mejorar la aplicación.

En la mayoría de los proyectos suelen surgir problemas. Por lo que uno debe planificar el desarrollo teniendo en cuenta que este tipo de cosas puede suceder. Cuando se trabaja y se cometen errores, pero además tienes unos plazos que se deben cumplir, se debe buscar tiempo extra para asegurar entregar lo prometido. Considero que hubo un problema al ser ambicioso con la funcionalidad y ajustar las horas de la planificación exactamente sin contemplar retrasos por errores. Personalmente, trabajo y estudio el máster, por lo que se tuvo que enderezar, solicitando una semana de vacaciones para dedicarla a tiempo completo al desarrollo de la aplicación. Los errores tienen repercusión, cuando no se han contemplado. Realmente cuando una empresa se compromete con un cliente a cumplir unos plazos, si por errores o una mala planificación se debe de emplear más horas de las planificadas, claramente estas horas no reportan ningún beneficio a la empresa porque el cliente no se va a hacer responsable de ellas. Por lo tanto, se ha aprendido la lección de que cuando uno realiza una planificación, tiene que equilibrar el tiempo que tiene con los objetivos que hay que cumplir.

He acabado muy contento del trabajo realizado, evidentemente todo se puede mejorar, pero hay que ser consciente del tiempo que se tiene. Por ser crítico conmigo mismo, considero que se podría haber empleado otros componentes para mejorar la aplicación, pero también esto provoca dedicar un tiempo a la documentación de Android, un tiempo que no se tiene, pero me quedo con la satisfacción de ser capaz de hacerlo, de tener la autonomía de comprender e implementar otros recursos que no han sido impartidos en el máster.

La idea es seguir trabajando esta aplicación e ir madurando la idea de lo que se quiere. Hacerla lo más eficiente y bonita posible. Desde luego una de las cosas a mejorar es el diseño, claro está que aparte de ser programador, uno tiene que sacar las ideas de un diseñador. Se ha intentado exponer en las vías futuras, todas las vías que dejo abiertas para continuar con la aplicación.

Para concluir, quiero dar las gracias en general por el máster que ofrece esta Universidad, ya que es muy completo. He aprendido mucho sobre el desarrollo de aplicaciones móviles y las diferentes tecnologías que existen actualmente. También dar las gracias a la mayoría de los profesores, que, a pesar de ser un máster online, tienen especial interés en el aprendizaje del alumno.

11. Glosario

- **Firestore:** Producto ofrecido a través de la plataforma de Google, que proporciona distintos recursos al usuario. Nosotros vamos hacer uso del mismo para el almacenamiento de los datos de la aplicación y para el servicio de autenticación.
- **noSQL:** Clase de sistema de gestión de base de datos, que no usa SQL como lenguaje de consultas. Los datos almacenados se guardan con una estructura de clave-valor.
- **JSON:** JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. Utilizado en las bases de datos noSQL.
- **Diagrama de Gantt:** Representa un diagrama temporal que incluye cada una de las tareas de un proyecto. De modo que, podemos visualizar, entre otras cosas, fechas de inicio y fin de las tareas o relaciones entre las mismas.
- **Git:** Software de control de versiones. Permite almacenar los datos en un repositorio externo, para mayor control y seguridad.
- **IDE:** Entorno de desarrollo integrado, que ayuda y facilita al desarrollador de software el desarrollo de nuevos productos.
- **SVN:** Software de control de versiones. Es de código abierto y permite almacenar copias de los datos en un repositorio externo.
- **API:** Identifica a una interfaz de programación de aplicaciones, que ofrece un conjunto de funcionalidad agrupada, para ser utilizada por otro software.
- **Gradle.** Es una herramienta de automatización de la construcción de nuestro código. Nos ayuda a gestionar las dependencias de nuestro proyecto.
- **OAuth2.** Protocolo de autorización, que permite a terceros acceder a contenidos de propiedad de un usuario, sin manejar ni conocer las credenciales del mismo.
- **Java.** Lenguaje de programación orientado a objetos.
- **HTTP.** El protocolo de transferencia de hipertexto (Hypertext Transfer Protocol), es el protocolo de comunicaciones que permite las transferencias de información en la World Wide Web.
- **Retrofit.** Cliente REST para Android y Java.
- **XML.** Lenguaje de marcado que permite definir lenguaje de marcas. Ayuda a representar la información en la web de forma estructurada. También, usado para almacenar datos de una forma legible.

- **REST.** Tipo de arquitectura software. Cualquier interfaz entre sistemas que utilice HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.
- **Emulador.** Software que permite ejecutar programas en una plataforma diferente de aquella para la que fueron destinados.
- **Layout.** Fichero de configuración de la interfaz de Android en formato XML, donde se indica la distribución de cada uno de los elementos que componen dicha interfaz.
- **Framework.** Estructura conceptual y tecnológica de asistencia definida, normalmente con artefactos o módulos concretos de software, que pueden servir de base para la organización y desarrollo de software.
- **Login.** Término dedicado al proceso mediante el cual se controla el acceso individual a un sistema informático, mediante unas credenciales que le son asignadas a un usuario.
- **Ítem.** Cada uno de los elementos que forman parte de un conjunto de datos.
- **Prototipo.** Ejemplar que se fabrica de la interfaz de la aplicación que sirve de modelo para desarrollar uno similar. Sirve para detectar primeros errores a simple vista.
- **Placeholder.** Texto que se incluye por defecto en los campos de edición de datos, proporcionando información sobre el dato que hay que introducir.

12. Bibliografía

[1] El País (publicado el 6 de marzo de 2017). Recuperado el 3 de octubre de 2017.

http://elpais.com/elpais/2017/03/06/ciencia/1488805065_770069.html

[2] EcoDiario (publicado el 3 de octubre de 2017). Recuperado el 3 de octubre de 2017.

<http://ecodiario.economista.es/salud/noticias/8649872/10/17/La-obesidad-y-el-sobrepeso-afectan-al-40-de-los-ninos-en-Madrid-a-causa-de-la-precariedad.html>

[3] Firebase Authentication. Recuperado el día 7 de octubre de 2017.

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

[4] Firebase Realtime Database. Recuperado el día 7 de octubre de 2017.

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

[5] El País (publicado el 4 de abril de 2017). Recuperado el 8 de octubre de 2017.

https://elpais.com/tecnologia/2017/04/04/actualidad/1491296467_396232.html

[6] Aceproject. (Software de gestión de proyectos)

<https://www.aceproject.com/>

[7] 20 minutos (publicado el 13 de abril de 2017). Recuperado el 20 de octubre de 2017.

<http://www.20minutos.es/noticia/2719117/0/maternidad-tardia-retraso/espana/causas-evolucion-tendencia/>

[8] Ministerio de educación. Recuperado el 20 de octubre de 2017.

<https://www.mecd.gob.es/educacion-mecd/areas-educacion/estudiantes/educacion-primaria.html>

[9] Guía de desarrolladores Android. Recuperado el 22 de noviembre de 2017.

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

[10] Google Cloud Platform. Recuperado el 24 de noviembre de 2017.

<https://cloud.google.com/sql/?hl=es>

[11] Azure Database for MySQL. Recuperado el 24 de noviembre de 2017.

<https://azure.microsoft.com/es-es/services/mysql/>

[12] Amazon DynamoDB. Recuperado el 24 de noviembre de 2017.

<https://aws.amazon.com/es/dynamodb/>

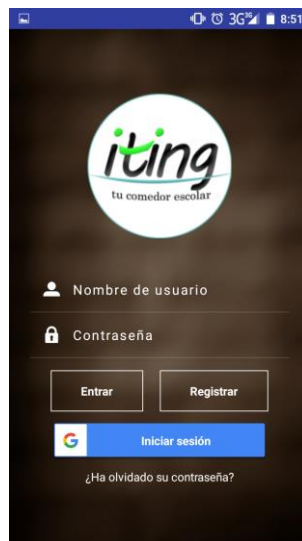
13. Anexos

13. 1. Anexo 1. Manual de usuario

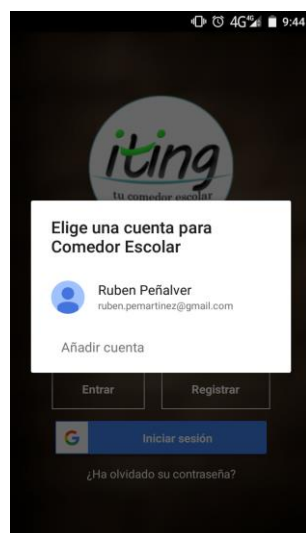
Rol: Centro Escolar

Registro: Cuenta de Google

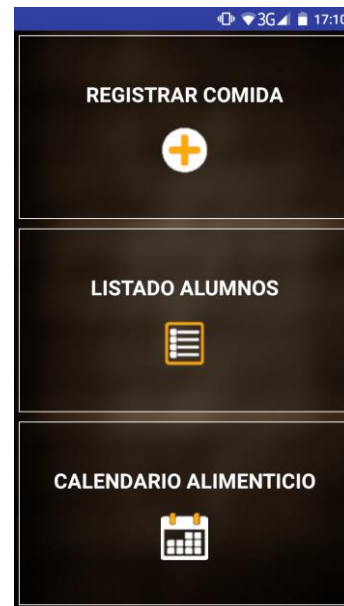
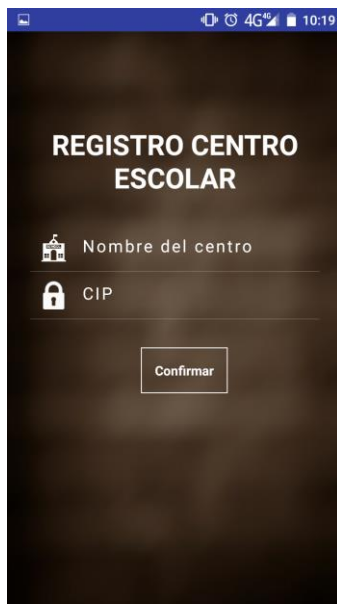
1. Para abrir la aplicación tenemos que tocar en el icono con el nombre “Comedor Escolar”. Y llegamos a la pantalla principal.



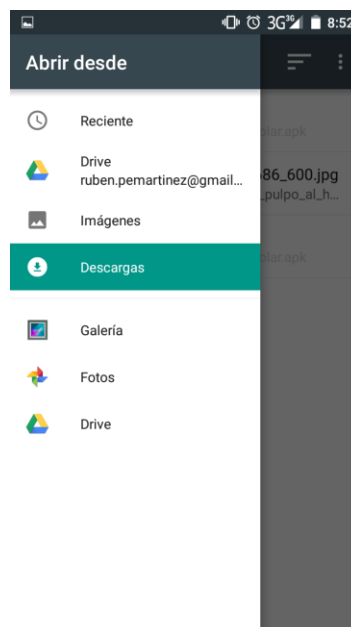
2. Existen dos formas de acceso y registro. La más sencilla es tocar sobre el botón de inicio de Google, si tenemos cuenta de Google. Seleccionamos nuestra cuenta. Continuamos con el tipo de registro que nos pertenece, centro escolar o alumno.



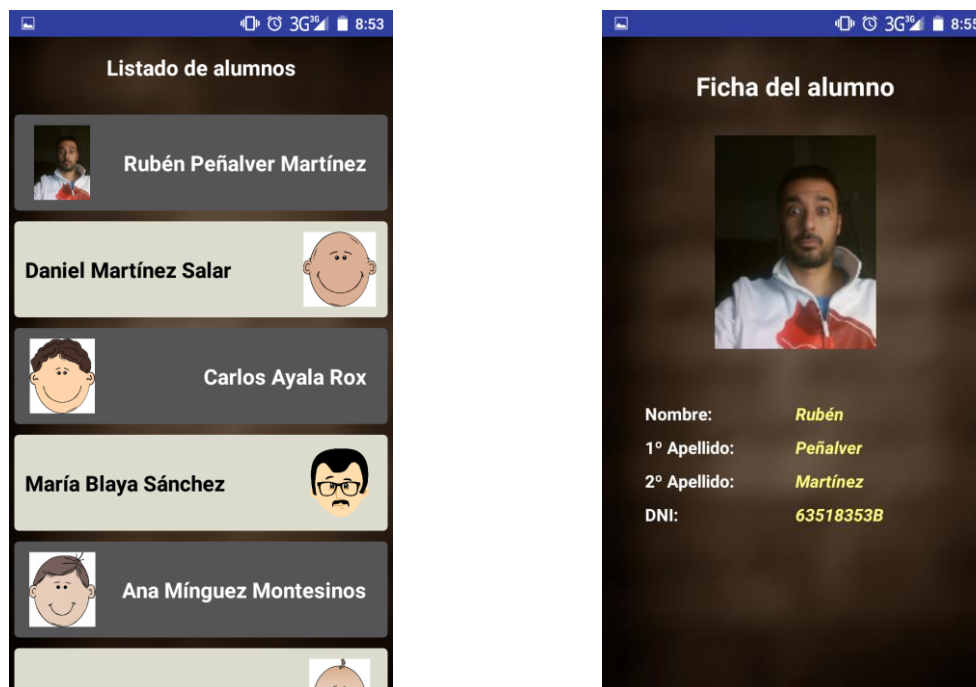
- Si seleccionamos centro escolar, hay que rellenar los siguientes datos para completar el registro. Una vez terminado el registro, debemos confirmarlo mediante el botón correspondiente. Navegamos hacia el menú, donde decidimos qué queremos hacer. Observamos toda la funcionalidad que tenemos disponible. Para acceder a alguna de ellas tocamos sobre la opción de menú que deseemos.



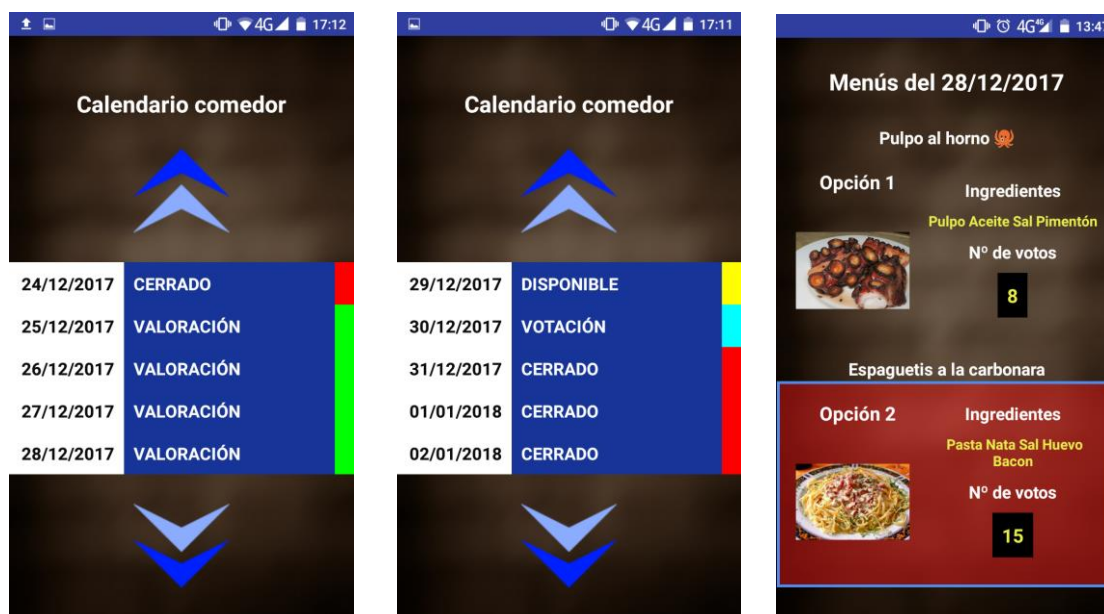
- Si elegimos la primera opción de menú "REGISTRAR COMIDA", nos encontramos con la pantalla de registro de menús. Registramos un menú y lo asociamos a una fecha. Debemos completar cada uno de los datos que se nos pide y en el formato indicado. Es obligatorio introducir al menos un ingrediente. Para introducir una foto en el registro, deberemos tocar donde se indica. Al pinchar sobre la opción de introducir una foto, se nos abrirán todas las opciones que tenemos disponible a través de nuestro dispositivo. Una vez hayamos seleccionado una foto, volvemos automáticamente a la aplicación. Finalmente, una vez tenemos los datos completados, confirmamos el registro mediante el botón correspondiente.



- Pantalla del listado de alumnos, que accedemos mediante la opción de menú “LISTADO ALUMNOS”. Veremos un listado de todos los alumnos que están registrados en nuestro centro. Podemos ver el perfil de un alumno, si pertenece a nuestro centro, para ello en el listado pincharemos sobre uno de los alumnos de la lista y se nos abrirá directamente la información del mismo.



- Opción de menú “CALENDARIO ALIMENTICIO”, que ofrece el centro escolar cada día. Los días pueden tener diferentes estados, pero cuando somos centro escolar no podemos hacer nada sobre ninguno de ellos. Lo importante para un centro escolar es que para abrir uno de los días, se debe de poner a disposición de los alumnos dos menús. Si entramos a uno de los días que no esté cerrado (en este caso VOTACIÓN), veremos qué información nos ofrece.



Rol: Alumno

Registro: Correo electrónico y contraseña

1. En este caso se ha seleccionado el registro mediante correo electrónico y contraseña. Una vez insertado los datos y pulsado el botón registrar, se envía un correo al usuario para verificar su cuenta. Si pinchamos en el enlace, que se encuentra en el cuerpo del correo, estaremos verificando la cuenta. Sin realizar este paso es imposible continuar. Cuando el registro se hace mediante una cuenta de Google, este paso se hace automáticamente.

Verifica tu dirección de correo electrónico de Comedor Escolar



noreply@comedorescolar-ec418.firebaseio.com

Hoy, 9:12
Usted ✕

Hola:

Haz clic en este enlace para verificar tu dirección de correo electrónico.

https://comedorescolar-ec418.firebaseio.com/_/auth/action?mode=verifyEmail&oobCode=qo6NEsfQZDuiN5R5qrm1UhMg3KtCdqctqAHYvG8b2rsAAAFgkhiXRg&apiKey=AlzaSyB4ilQSIDcB9hkhJDIDa_DOWYbL45Aj9jc

Si no has emitido esta solicitud, ignora este mensaje.

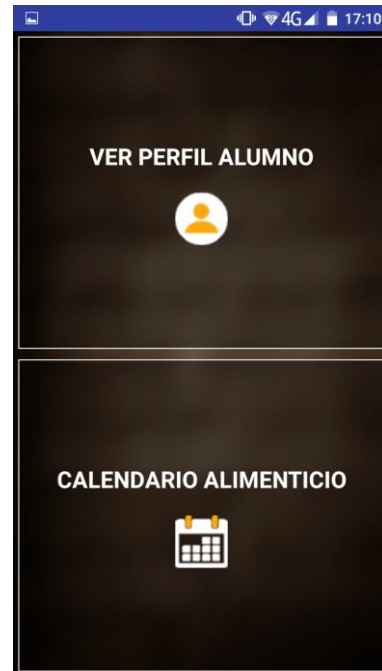
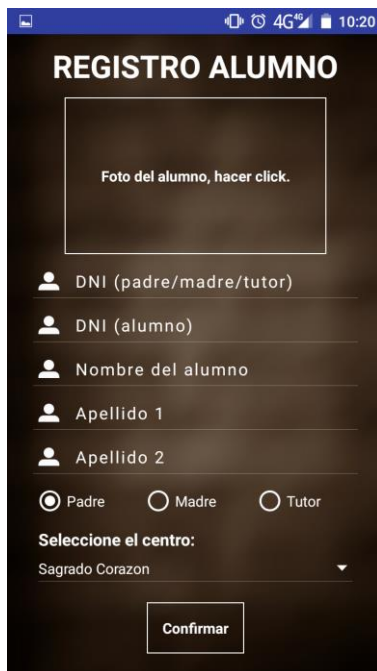
Gracias,

El equipo de Comedor Escolar

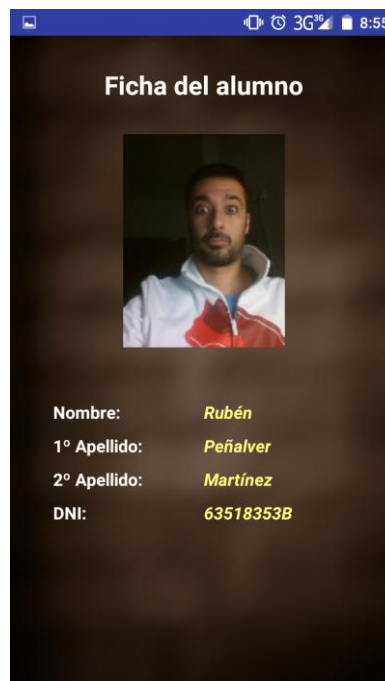
Your email has been verified

You can now sign in with your new account

2. Si el registro lo hacemos como alumno, los datos que se solicitan son los que hay que rellenar para completar el registro. Se debe de hacer una foto al alumno, para ello cuando se pincha donde se indica se activará la cámara de fotos del dispositivo de manera automática. Una vez que se ha hecho la foto y se ha aceptado, se insertará en el lugar correspondiente y se procederá a seguir con el registro. Una vez que nos hemos registrado como alumno, accedemos a la aplicación. Vemos el menú correspondiente al alumno.



3. Aquí llegamos a través de la opción de menú "VER PERFIL ALUMNO". Observamos los datos que hemos introducido en el registro.



4. Opción del menú “CALENDARIO ALIMENTICIO”. Cuando somos alumnos, si podemos ejercer operaciones sobre cada uno de los días, según el estado en el que se encuentre. Si el estado es cerrado, no podremos hacer nada, quiere decir que el centro escolar no ha publicado menús para ese día. Si el estado es disponible, vemos cuál ha sido la elección de ese día. Pero si el estado es votación, nos encontramos con el botón votar, ofreciendo la posibilidad de votar por el menú deseado. Para poder votar, nos tenemos encontrar en días posteriores al actual.

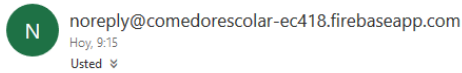


5. Continuamos en el calendario, pero el día seleccionado se encuentra en estado de valoración. Vemos que ya no está el botón votar, pero en cambio existe un botón valorar. Ahora podremos dar una puntuación al menú ganador. Se ofrece la votación resultante y la puntuación media que lleva hasta ese momento el menú. Para poder valorar, el día debe haber pasado y tener las dos comidas publicadas.



- Si el registro ha sido mediante correo y contraseña, tenemos la opción en la pantalla principal de recordar contraseña. En caso de olvido, podemos pinchar sobre la opción indicada y el sistema enviará un correo a su cuenta. En este correo habrá que pinchar en el enlace que se indica. Se abrirá una ventana, que permite introducir la nueva contraseña y una vez completados habrá que confirmar los datos. Esta ventana, informará de que el cambio se ha efectuado correctamente. Por lo que ya puede volver a entrar a la aplicación.

Cambia la contraseña de Comedor Escolar



Hola:

Haz clic en este enlace para cambiar la contraseña de Comedor Escolar de tu cuenta rubenelmorok@hotmail.com.

https://comedorescolar-ec418.firebaseio.com/_/auth/action?mode=resetPassword&oobCode=YWbyP7MlxBNnzjKXt3azlqsmqZLXwwwKtQqRSEZrCRwAAAFgkshMg&apiKey=AlzaSyB4ilQSiDcB9hkhJDIDa_DOwYbL45Aj9ic

Si no has solicitado este cambio, ignora este correo electrónico.

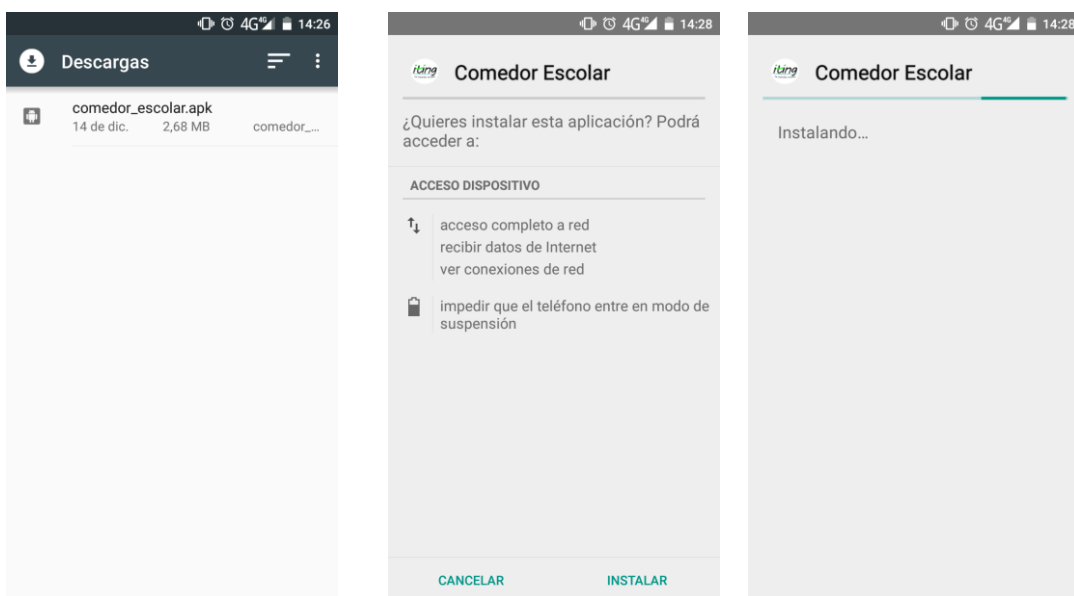
Gracias,

El equipo de Comedor Escolar

The image shows a web form titled 'Reset your password' in bold black text. Below the title, it says 'for rubenelmorok@hotmail.com'. There is a text input field labeled 'New password' with a blue underline. To the right of the input field is a blue eye icon. At the bottom right of the form is a blue button with the word 'SAVE' in white capital letters.The image shows a confirmation message box. It has the title 'Password changed' in bold black text. Below the title, it says 'You can now sign in with your new password' in a smaller black font.

13. 2. Anexo 2. Manual de instalación

- 1) Primero será necesario descomprimir el .zip que en su interior contiene la aplicación comedor_escolar.apk.
- 2) La aplicación debe trasladarse al dispositivo móvil de alguna forma. Por ejemplo:
 - a) Conectando mediante USB el dispositivo al pc.
 - b) Mediante correo electrónico.
 - c) Vía Google Drive.
- 3) Una vez en el dispositivo y lo tengamos en descargas, pincharemos sobre la aplicación. Se abrirá una ventana y tenemos que pulsar en instalar.



- 4) Ya se encuentra instalado. Podemos abrir la aplicación.

