

Documentación

Winedroid



Realizado por Christian Román Sánchez

Índice

1.Presentación del proyecto.....	2
Explicación resumida.....	2
Estudio de mercado.....	3
Valor de producto.....	3
2.Planificación de tareas y estimación de costes.....	4
Planificación de tareas.....	4
Estimación de costes.....	7
Herramientas usadas.....	8
Gestión de Riesgos.....	9
3.Análisis de la solución.....	11
Análisis de la solución: Requisitos.....	11
Análisis de escenarios: Casos de uso.....	12
4.Diseño de la solución.....	16
Diseño del interfaz del usuario.....	16
Diagrama de clases.....	29
Diagrama de Persistencia.....	30
Arquitectura del sistema.....	34
5.Implementación de la solución.....	35
Justificación tecnológica.....	36
Aspectos esenciales de la implementación.....	42
6.Testeo y pruebas de la solución.....	47
Plan de pruebas.....	47
Solución a problemas encontrados.....	59
7.Lanzamiento y puesta en marcha.....	60
Aspectos relevantes del despliegue.....	60
Manual de uso.....	66
8.Valoración y conclusiones.....	73
9.Bibliografía y recursos utilizados.....	74

1. Presentacion del proyecto

Vamos a implementar una aplicación de vinoteca la cual llamaremos Winedroid, con esta aplicación queremos atraer al público adulto para que pueda realizar sus listas con los mejores vinos que ha ido encontrando, dejando así su valoración y su comentario además de compartir a nivel social su experiencia con estos vinos, a continuación, vamos a presentar las características que posee la aplicación que vamos a desarrollar:

- La aplicación tendrá una línea de tiempo, donde veremos los vinos mejores valorados en lista, estos vinos se ordenarán de mejor valorado a peor valorado y se mostraran los 10 mejores vinos valorados, cuando pulsemos sobre alguno de estos vinos, se nos abrirá la ficha con la información del vino, en el cual podremos valorar, comentar o incluso ver su denominación de origen, podremos también ver su bodega geo localizada.
- Los usuarios tendrán acceso a un buscador, mediante el cual podrán buscar un vino mediante su nombre o su tipo de uva, además mediante ese buscador podrá buscar a un usuario para agregarlo y ver que vinos va valorando o un hashtag con el cual visualizara los vinos que tengan esos hashtags.
- Los usuarios tendrán acceso a su perfil donde podrán visualizar su foto, nombre de usuario y una breve descripción sobre él. Dispondrá además de un botón para editar su perfil a placer y un botón que generará un código QR, el cual, podrá ser utilizado por otros usuarios para escanearle y agregarle como amigo.
- Los usuarios podrán escanear un código QR para agregar a un usuario o podrán escanear el código de barras de un vino para visualizar su ficha o en caso de que esta no exista para añadir su ficha.
- La identificación se realizará mediante Google y Twitter además de implementar un sistema propio.
- En lo referente al servidor se almacenará remotamente todos los datos del usuario, usaremos Firebase en este aspecto, y utilizaremos sus métodos para acceder a ella.

Estudio de Mercado

Vamos a realizar un estudio de mercado que nos indicara si él hay mucha competencia en el mercado al que está orientado nuestra aplicación.

Entorno a la competencia hemos podido comprobar que no hay muchas aplicaciones de vinotecas que ofrezcan lo que ofrecemos a los usuarios, sin embargo, el mercado ya está hecho y como siempre unirnos a el será un camino duro.

El público al que vamos dirigido seria público adulto, de más de 18 años, gente que pudiera disfrutar del vino y utilizar nuestra aplicación para comentar y valorar sus vinos.

- **Análisis Dafo**

Debilidades: La mayor dificultad que tenemos es que no abarcamos un público muy amplio y que la mayoría de la gente adulta no está experimentada a utilizar un teléfono móvil y sus aplicaciones por lo tanto puede sernos un poco difícil adaptarnos a esa circunstancia, aparte es nuestra primera app por lo tanto es nuestro primer contacto con este mundo

Amenazas: Nuestra principal amenaza es que no podamos hacernos un hueco en el mercado ya que como hemos dicho en el punto anterior la población adulta no tiende a usar las aplicaciones por lo tanto eso puede llevarnos a no encajar

Fortalezas: Una de nuestras fortalezas es la facilidad para entender la aplicación un aspecto fundamental a la hora de realizar aplicaciones además de que el mercado va aumentando poco a poco.

Oportunidades: Esta aplicación puede ser una oportunidad para dar un salto, ya que no existen muchas aplicaciones de este estilo y al realizarla podemos dar a conocer el mundo del vino lo cual nos llevaría a que muchas bodegas apoyaran nuestra iniciativa, lo cual a largo plazo atraería más usuarios a nuestra aplicación.

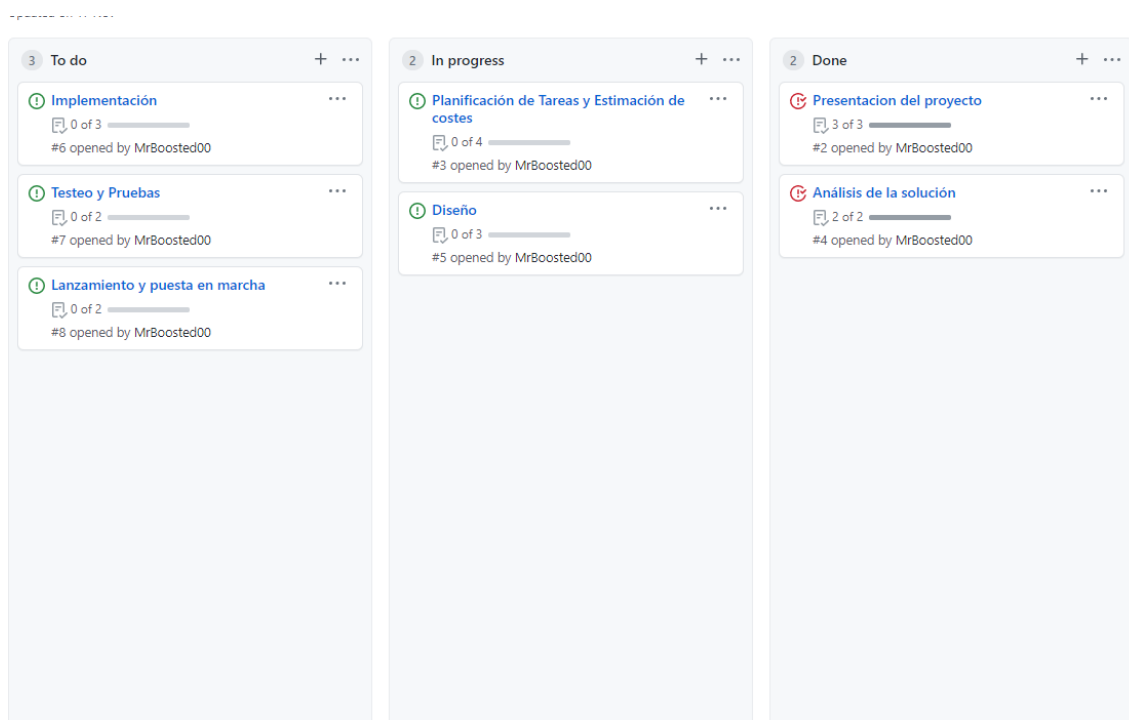
Valor del producto

Un diseño más fácil y todo lo explicado anteriormente es el valor de nuestro producto, no hay ninguna app que sea exactamente igual a la nuestra lo cual puede ser un valor añadido.

2. Planificación de tareas y estimación de costes

Planificación de tareas


Para la planificación de tareas hemos utilizado un tablero Kanban que nos ofrece GitHub para poder organizar mejor el proyecto. Dicho tablero nos ofrece la posibilidad de dividir las tareas en subtareas que podemos marcar con un checkbox además que podemos dividir las en To Do, In Progress y Done para saber por qué tarea vamos y que tareas tenemos acabadas.




Como podemos observar tenemos un tablero en el cual hemos añadido issues, las issues tendrán checkbox para marcar el número de actividades que contiene cada una


Issues a terminar para el día 21 de octubre de 2020:

Presentacion del proyecto #2

 Closed MrBoosted00 opened this issue on 20 Oct · 0 comments

 MrBoosted00 commented on 20 Oct · edited ▾ Owner 😊 ...

- ☒ Explicación resumida
- ☒ Estudio de mercado
- ☒ Valor de producto


 MrBoosted00 commented on 21 Oct · edited ▾ Owner 😊 ...


- ☒ Requisitos de la aplicacion
- ☒ Analisis de escenarios

Previsto por Terminar: 21/10/2020

Issues a terminar el 28 de octubre:

Planificación de Tareas y Estimación de costes #3

 Open MrBoosted00 opened this issue on 21 Oct · 0 comments

 MrBoosted00 commented on 21 Oct · edited ▾ Owner 😊 ...


- ☐ Planificación y organización de tareas
- ☐ Estimación de costes y recursos: hardware, software y humanos
- ☐ Herramientas usadas
- ☐ Gestión de riesgos

Previsto por terminar el 27/10/2020
Como muy tarde 28/10/2020

Issues a terminar el 11 de noviembre:

Diseño #5

 Open MrBoosted00 opened this issue on 21 Oct · 0 comments

 MrBoosted00 commented on 21 Oct · edited ▾ Owner 😊 ...

- ☐ Diseño de la interfaz de usuario
- ☐ Diseño de la persistencia de la información
- ☐ Diseño de la arquitectura del sistema

Previsto por terminar: 10/11/2020
Plazo maximo: 11/11/2020

Issues a terminar el 30 de noviembre:

Implementación #6

Open

MrBoosted00 opened this issue on 21 Oct · 0 comments



MrBoosted00 commented on 21 Oct · edited

Owner



- ☐ Justificación tecnológica
- ☐ Aspectos esenciales de la implementación
- ☐ Desarrollo de la funcionalidad indicada por el tutor/a

Previsto por terminar: 30/11/2020

Issues a terminar el 8 de diciembre:

Testeo y Pruebas #7

Open

MrBoosted00 opened this issue on 21 Oct · 0 comments



MrBoosted00 commented on 21 Oct · edited

Owner



- ☐ Plan de pruebas (unitarias, integración, sistema y usuarios)
- ☐ Solución a problemas encontrados

Previsto por terminar: 8/12/2020

Issues a terminar el 15 de diciembre:

Lanzamiento y puesta en marcha #8

Open

MrBoosted00 opened this issue on 21 Oct · 0 comments



MrBoosted00 commented on 21 Oct · edited

Owner



- ☐ Aspectos relevantes del despliegue y puesta en marcha del sistema
- ☐ Manual de uso

Previsto por terminar: 15/12/2020

Estimación de costes

La estimación de costes es un proceso que consiste en desarrollar una aproximación a los recursos monetarios necesarios para completar las actividades del proyecto. Se trata de una predicción basada en la información que tenemos disponible en un momento dado.

Para este proyecto necesitaremos licencias ofimáticas para realizar los documentos necesarios para la aplicación, utilizaremos las licencias de Office 365, compraremos una licencia anual para utilizarla lo cual serán 9,80€ al mes al ser el proyecto 4 meses saldría por un total de 39,2€.

Necesitaremos un equipo que se adapte al proyecto el cual ya tenemos escogidos, como se utilizaran para futuros proyectos se aplicara un porcentaje del precio total al cliente el precio total seria unos 1200€ por cada ordenador, al tratarse de 1 ordenadores serian 1200€ al cliente le cobraríamos un 33% del precio total lo cual serian 400€.

La cuota de autónomo se pagaría un porcentaje ya que se dividirá en la totalidad de los distintos proyectos, lo cual serian un 93,50€

Para el proyecto dispondremos de un trabajador Christian Román Sánchez que trabajara durante 130h en la realización del proyecto y se le pagara a 22€ cada hora lo cual sería un total de 2860€

<u>Sevicio</u>	<u>Precio</u>
Office 365	39,2€
Equipo	400€
Cuota de Autonomo	93,5€
Sueldo	2860€
Total Costes sin IVA:	3392,7€
Total Costes con IVA(21%):	4512€

Herramientas usadas

En este apartado vamos a explicar las herramientas que tenemos pensado utilizar a la hora de realizar la aplicación para escribir código utilizaremos Intel IJ, ya que usaremos Kotlin native, por lo tanto nuestra aplicación ira escrita en Kotlin, Intel IJ se trata de un Ide que nos sirve para programar una gran cantidad de código, entre ellos el que deseamos desarrollar, además de ser un ide con muchas características y el que creemos perfecto para esta tarea, para las tareas ofimática vamos a usar Office 365, ya que con el tendremos acceso a Word, Excel y todos estos programas que nos ayudaran a la hora de realizar la documentación, la guía del usuario...

Para probar la aplicación usaremos un emulador de Android que será el emulador que nos trae Intel IJ y para controlar todo el código fuente además de controlar también las tareas a realizar usaremos GitHub.

Utilizaremos Google Firebase para la base de datos y para la autenticación del usuario, aunque se trate de un sistema relativamente reciente tiene una base bastante amplia de gente que lo desarrolla.

Gestión de riesgos

El trabajo que realizamos se realiza en una oficina por lo tanto no conlleva riesgos laborales tan graves como en otros sectores, sin embargo, hay riesgos los cuales deben ser mitigados.

Los principales riesgos laborales que experimentan nuestra profesión son:

– Fatiga visual o muscular

Pueden aparecer molestias musculares por el uso de pantallas además de que el espacio debe contar la luz adecuada, en lo referente a la fatiga muscular, pueden producirse por posturas incorrectas a la hora de sentarse o por una mala ubicación del equipo.



Para solucionar este riesgo laboral es necesario seleccionar adecuadamente el equipamiento, como pueden ser sillas, mesas y los distintos equipos informáticos (pantallas, teclados...)

– Golpes o caídas

El lugar de trabajo debe mantenerse limpio y ordenado, dejando **libres de obstáculos las zonas de paso**. El material de trabajo se debe almacenar en estanterías y armarios. Si no seguimos todas estas indicaciones, puede suponer en que el personal se caiga o pueda recibir un golpe por material mal colocado



– Contacto eléctrico

Los trabajadores están en contacto constante de material eléctrico por lo tanto deben respetar las normas de seguridad básicas en el uso de los equipos eléctricos y **revisar el estado de cada equipo** antes de su uso

– Carga mental

El estrés en el trabajo o la desmotivación pueden ser un riesgo laboral y, por lo tanto, afectan a la salud de los trabajadores. Por ello, es importante realizar tareas variadas, **realizar paradas periódicas para prevenir la fatiga**, seguir hábitos de vida saludable y realizar ejercicio de forma habitual.



Además de los riesgos laborales también existen los riesgos específicos de proyectos de software, los cuales debemos tener en cuenta a la hora de realizar uno, para controlar que hacer y cómo actuar frente a la pérdida de información, al retraso del proyecto o frente a una disputa, estos riesgos

Frente a la falta de organización del equipo, usaremos los tableros de GitHub para controlar que tareas debemos realizar y terminar antes de empezar otras

- Pérdida de código fuente: Tendremos un servidor en el que guardaremos todo el trabajo que vayamos haciendo al día además de disponer de copias de seguridad en la nube.

- Mala calidad del producto: Se pospondrá el lanzamiento hasta poder ofrecer la calidad que deseamos para el producto

- Retraso en una entrega, trabajaremos en esa entrega hasta acabarla y entregarla y se descontará del precio ofertado al cliente.

- Poca retroalimentación con el cliente: Mandaremos actualizaciones al cliente con los últimos cambios realizados para que opine acerca de la aplicación y de cómo vamos trabajándola.

3.Análisis de la solución

Requisitos

-Requisitos funcionales:

- +RF1 Debe tener una línea de tiempo donde ver los vinos valorados
- +RF2 Tendremos un mapa donde tener geo localizadas las bodegas con sus vinos
- +RF3 Usaremos al cámara para subir una foto o haremos uso de la galería
- +RF4 Tendremos un sistema que escaneara el código de barras del vino, lo cual nos llevara a su ficha si esta creado para poner nuestra valoración, o nos llevara a una ficha para añadirlo en caso de que no este
- +RF5 Podremos agregar usuarios mediante código QR o buscando su nombre
- +RF6 El sistema de creación de cuentas podrá ser propio y deberá añadir los sistemas de autorización de Google y Twitter
- +RF7 Almacenaremos los datos remotamente en un servidor en una base de datos no relacional
- +RF8 Haremos uso de una API REST para acceder al almacenamiento

-Requisitos no funcionales:

- +RNF1 Compatibilidad- la aplicación debe ser compatible en cualquier dispositivo móvil que utilicemos.
- +RNF2 Utilizaremos Kotlin como lenguaje para la aplicación móvil, haremos uso de Kotlin Native para que pueda ser compatible con IOS
- +RNF3 El uso de la API REST lo realizaremos mediante SpringBoot
- +RNF4 Utilizaremos Firebase para la autorización

-Requisitos de información:

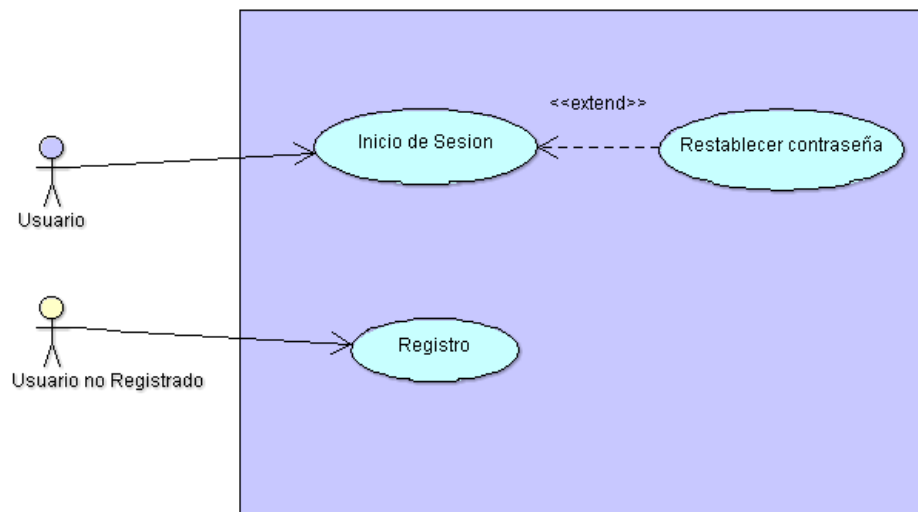
- +RI1 De los vinos guardaremos su código de barras, su nombre y su bodega geo localizada
- +RI2 De los usuarios guardaremos su nombre de usuario, su nombre y su fecha de nacimiento, además de la localización en la cual añadió el vino en cuestión.

Diagramas de casos de uso

El actor que hemos encontrado en esta aplicación sería el Usuario y el Usuario no registrado, estos tomarían relación con los siguientes casos de uso:

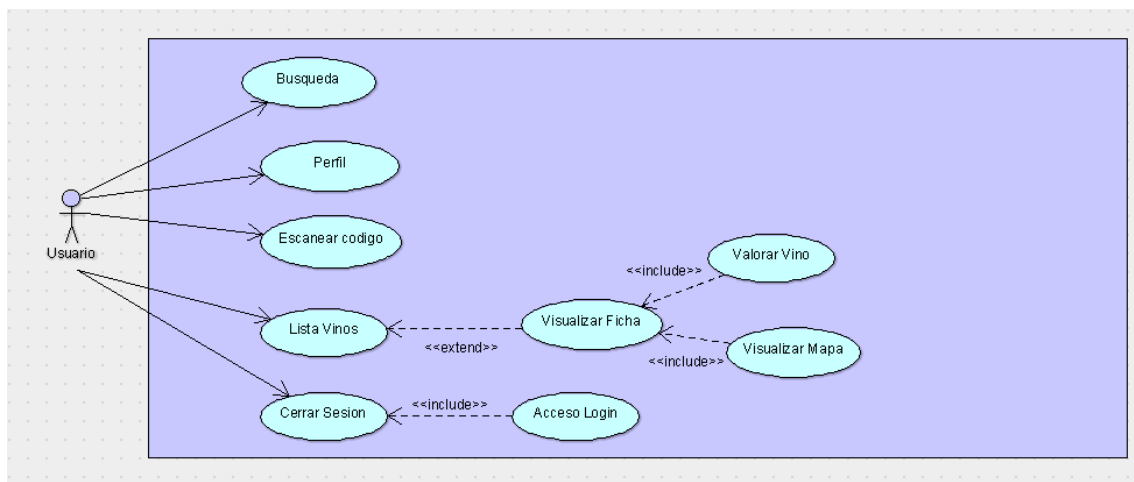
Caso de uso: Inicio de la aplicación

Cuando el usuario inicie la aplicación podrá registrarse en caso de ser Usuario no registrado o Iniciar Sesión en caso de estar registrado



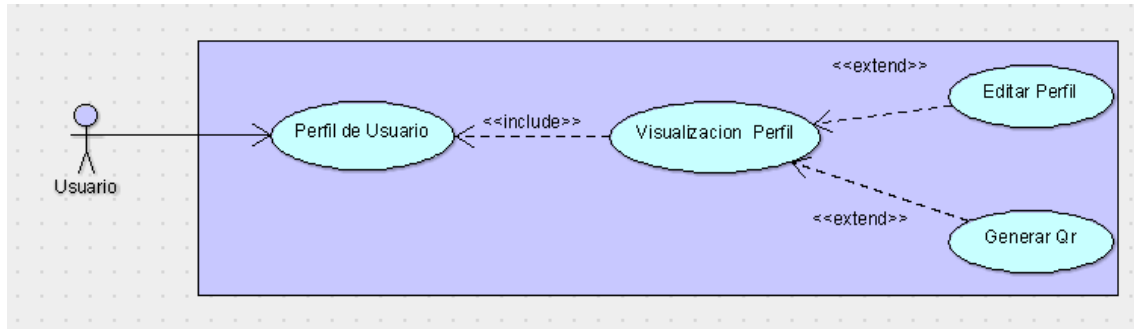
Caso de uso: Pantalla Principal

En la pantalla principal de la aplicación encontraremos las diferentes opciones de esta además de tener también la lista con los vinos mejores valorados



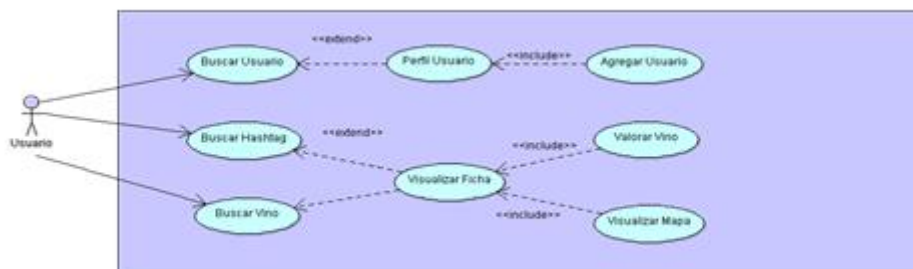
Caso de uso: Perfil Usuario

Cuando el usuario acceda a su perfil le aparecerá su perfil con su fotografía y su descripción, y tendrá la posibilidad de editar su perfil o de generar un código QR para su perfil



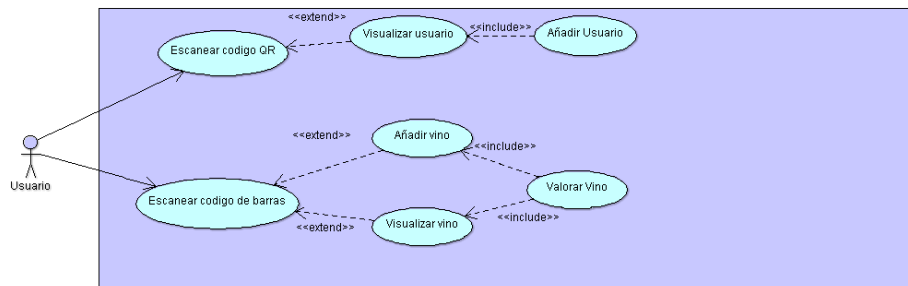
Caso de uso: Búsqueda del usuario

Este caso se dará cuando el usuario quiera Buscar, podrá buscar por hashtag, por vino o por usuario.



Caso de uso: Escaneo de código

A la hora de escanear un código el usuario podrá escanear un código QR para ver el perfil de otro usuario y agregarlo si así lo desea o podrá consultar un vino que haya escaneado mediante el escaneo de su código de barras, en caso de que este vino no exista le aparecerá una ventana para añadir el vino



4.Diseño de la solución

Diseño de la interfaz del usuario

En este documento mostraremos un prototipo de las pantallas que constara nuestra aplicación con su explicación de cada una y sus funciones además de cual enlaza con cual. Para ello utilizaremos capturas que hemos ido recopilando a lo largo de la creación del prototipo, así será más claro a la hora de programarlo y a la hora de que el cliente vea su producto final.

La paleta de colores seria el color blanco, color negro y el siguiente color:



Que se trata del color rojo vino.

Como logo de la aplicación usaremos el siguiente, que ya ha sido previamente aprobado por el cliente.



La fuente que utilizaremos será Roboto de la cual dejamos una muestra a continuación

Roboto
SUNGLASSES
Self-driving robot ice cream truck
Fudgesicles only 25¢
ICE CREAM
Marshmallows & almonds
#9876543210
Music around the block
Summer heat rising up from the boardwalk

Splash Screen



La primera pantalla que vería el usuario se trataría de un Splash Screen, a continuación le aparecería la pantalla de login.

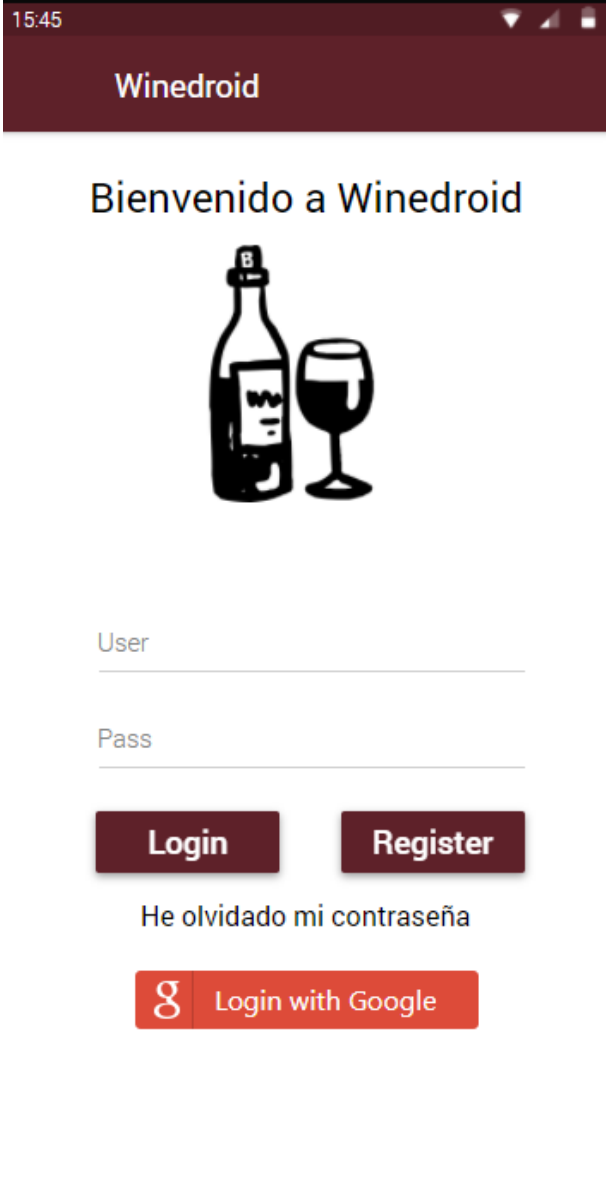


Se trata de una pantalla bastante simple que contendría el logo de la aplicación además del nombre de esta, la pantalla aparecía solamente al iniciar la aplicación.

Winedroid



Pantalla de Inicio de Sesión



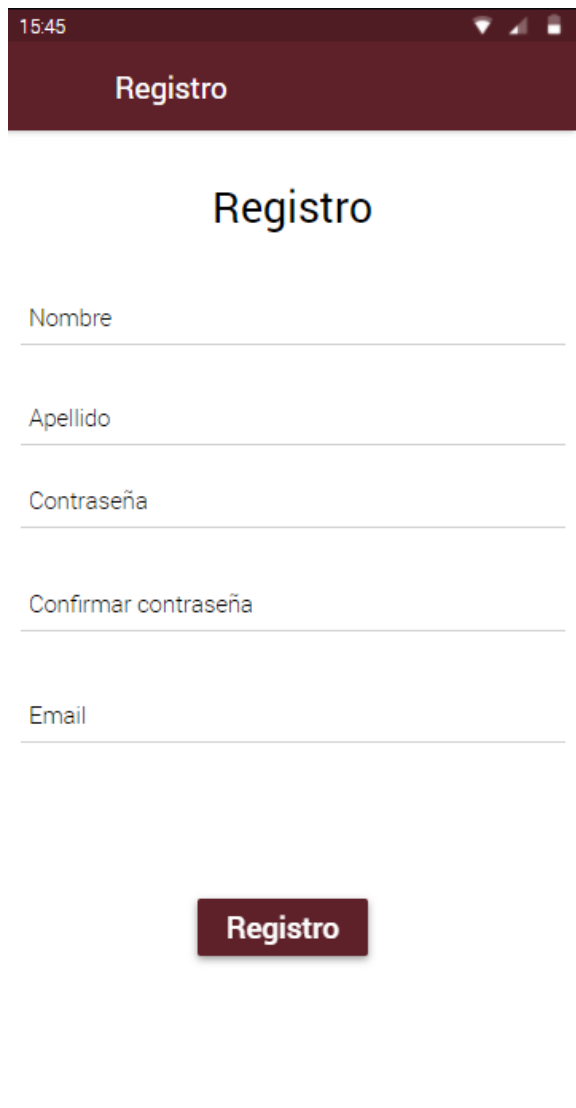
The screenshot shows a mobile application interface for 'Winedroid'. At the top, a dark red header bar contains the time '15:45' on the left and status icons on the right. Below the header, the word 'Winedroid' is centered in white. The main content area has a light gray background. It starts with the text 'Bienvenido a Winedroid' in a bold, black font. Below this is a black line-art illustration of a wine bottle and a glass. Further down are two input fields: 'User' and 'Pass', each with a light gray border and a small 'x' icon on the right. Below the 'Pass' field are two dark red buttons with white text: 'Login' and 'Register'. Under these buttons is a link that says 'He olvidado mi contraseña'. At the bottom is a red button with the Google 'G' logo and the text 'Login with Google'. A thin black horizontal line is at the very bottom of the screen.

Se trata de la pantalla de inicio de sesión que usaría el usuario para acceder a la aplicación, el usuario facilitaría su usuario y su contraseña y si los datos son correctos el usuario accedería a la pantalla de inicio de la aplicación.

El usuario podría acceder por medio de google o de twitter también y en caso de no querer usar ninguna de estas cuentas, podría pulsar en Register, lo cual le llevaría a una pantalla de Registro que veremos a continuación.

En caso de que el usuario haya olvidado su contraseña podrá pulsar en He olvidado mi contraseña para acceder a la pantalla de recuperación de contraseña.

Pantalla de Registro



15:45

Registro

Registro

Nombre

Apellido

Contraseña

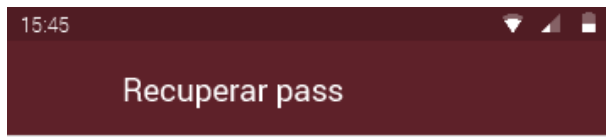
Confirmar contraseña

Email

Registro

Esta pantalla aparecería al pulsar el botón de registro de la pantalla de login, en esta pantalla el usuario introducirá sus datos además de su email el cual se vinculara a su cuenta para así poder recuperarla en caso de pérdida de contraseña, después de introducir sus datos el usuario pulsara en el botón de Registro y creara así su cuenta en la aplicación con una imagen por defecto que el usuario podrá cambiar en su perfil, lo cual lo llevara a la pantalla de inicio de sesión para iniciar sesión en la aplicación.

Pantalla Recuperación contraseña



Esta pantalla sería la de recuperación de la cuenta en la cual el usuario escribiría su correo y enviaría un código.

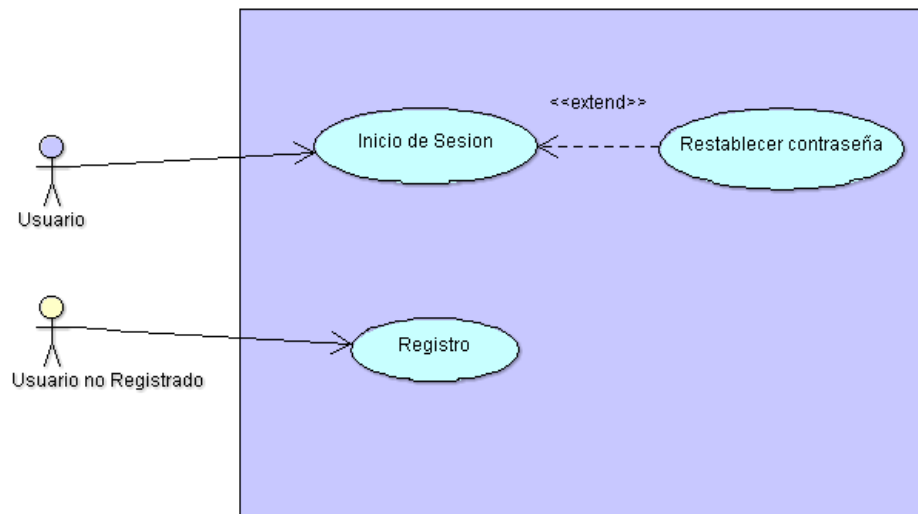
Introduce el correo enlazado a la cuenta
para recuperar tu contraseña



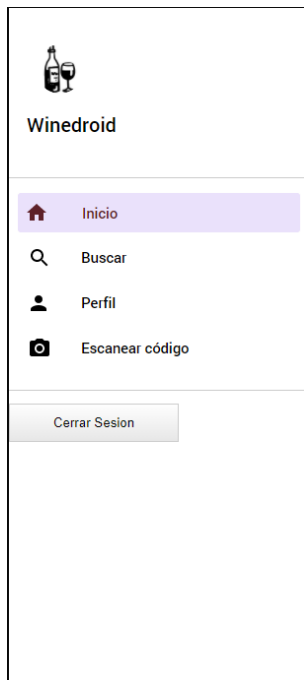
Enviar



Las pantallas anteriores hacen referencia al caso de uso de **inicio de la aplicación**, en la cual el usuario tendrá la posibilidad de acceder a la aplicación iniciando sesión o registrándose

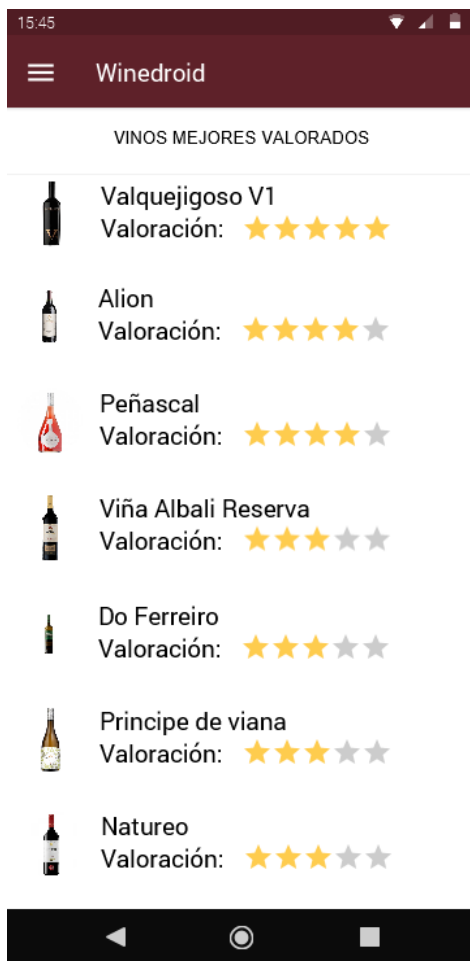


Menú de navegación



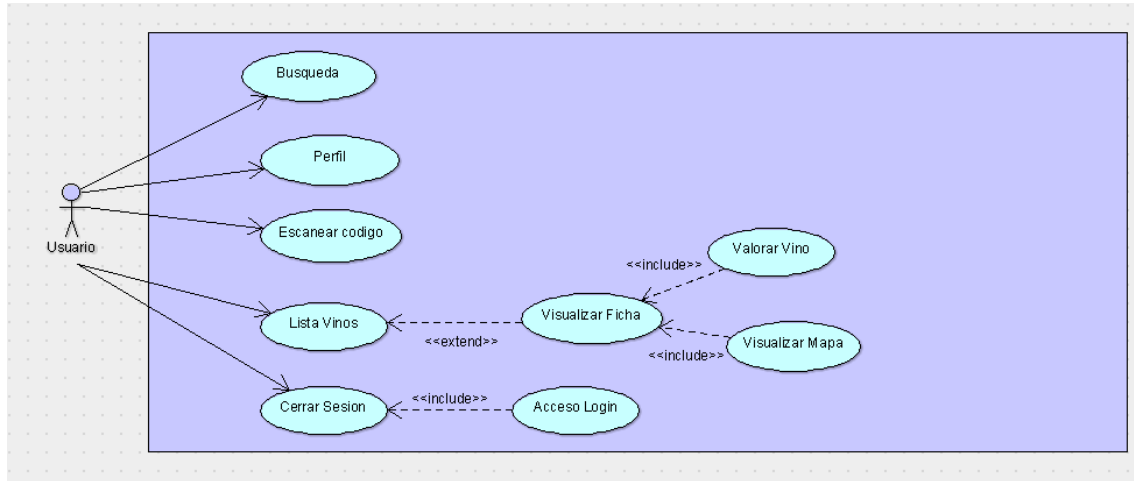
Es el menú de navegación de la aplicación el cual nos llevara a las diferentes pantallas de esta

Pantalla de Inicio de la aplicación



Se trata de la pantalla inicial de la aplicación que contendrá un listado de los vinos mejor valorados de la aplicación a los cuales el usuario podrá acceder para ver más información y comentarios sobre estos.

Las pantallas anteriores están relacionadas con el caso de uso **Pantalla de inicio** el menú de navegación nos mostrará las distintas opciones como podemos ver en el diagrama y la ficha de los vinos nos permitirá visualizar la ficha



Pantalla de búsqueda



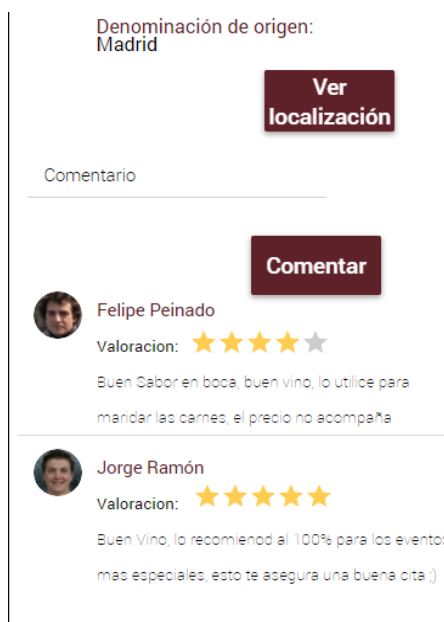
La pantalla de búsqueda de la aplicación en la cual podrás buscar por nombre de vino, tipo de vino y usuario, te mostrara a continuación los resultados abajo con un pequeño resumen y podrá pulsarle para acceder a la información de este.

Ficha de vino

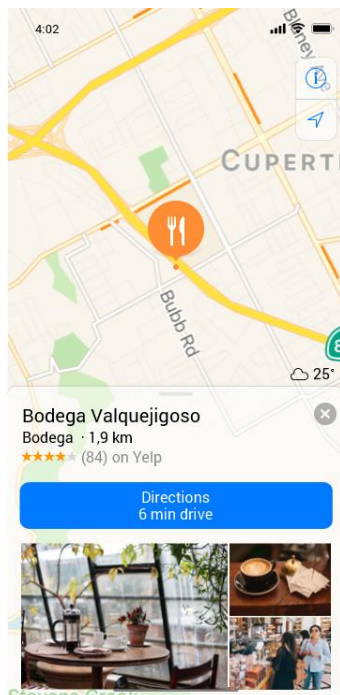


Esta pantalla nos muestra la ficha del vino, con su fotografía y una descripción de este, además nos permite valorarlo y ver su denominación de origen, además de poder comentar y ver los comentarios de este.

Tendrá además un botón con el que podremos ver la localización de su bodega en un mapa.

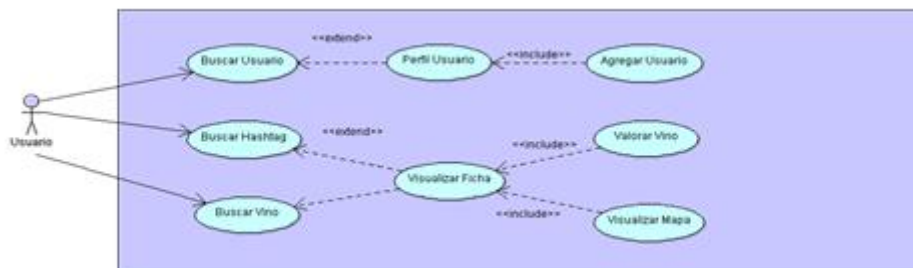


Mapa



Se trata de un mapa en el que nos mostrara donde se encuentra la bodega del vino que hemos seleccionado

Las pantallas anteriores hacen referencia al caso de uso de pantalla de búsqueda el cual nos muestra la unión de las distintas pantallas anteriores con la pantalla de búsqueda, con la cual podemos visualizar la ficha del vino, valorarla y visualizar su mapa



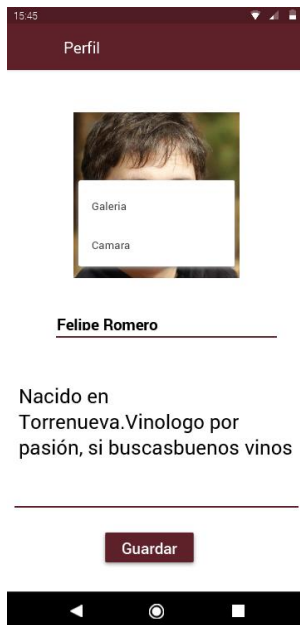


Perfil

Se trata de una pantalla que nos mostrara el perfil del usuario con su nombre y la descripción de este.

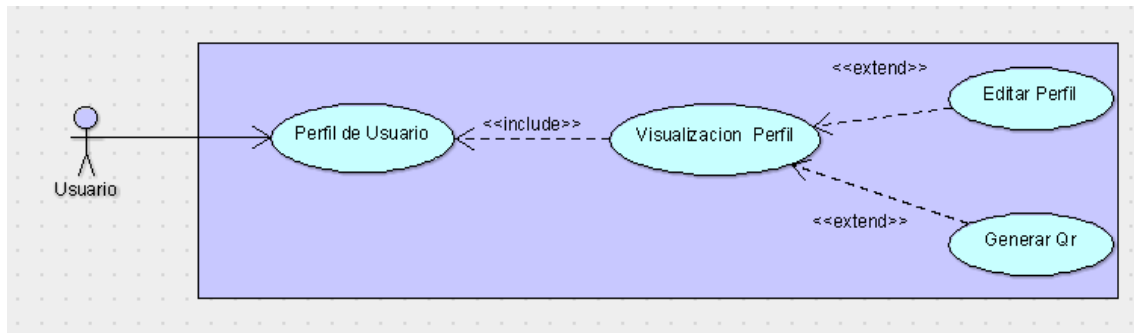
Tiene un botón con el cual podrá generar un código QR para poder ser agregado y en el menú de opciones tendrá un botón de editar el perfil.

Modificar Perfil



Esta pantalla seria la pantalla de modificación de perfil, donde el usuario podrá cambiar su nombre, su fotografía y su biografía, al pulsar en la fotografía te aparecerá un menú flotante en el cual te preguntara de donde deseas seleccionar la imagen, al pulsar en guardar se guardaran los cambios del perfil.

El diagrama que hace referencia al caso de uso anterior es el de la pantalla de perfil el cual nos muestra las distintas opciones que tendremos acceso



Código QR



Esta pantalla nos mostrara el código QR que utilizara el usuario para ser agregado, para volver a la pantalla anterior pulsara atrás.



Ficha de creación de vino

Esta sería la ficha que se nos mostraría al escanear un vino y que este fuera inexistente dentro de la aplicación, nos saldrá una pantalla donde rellenaremos los datos y escribiremos nuestro comentario, a continuación al pulsar en el check nos saldrá un mensaje de si deseamos o no guardar el vino.

Las anteriores pantallas harían referencia al siguiente caso de uso:

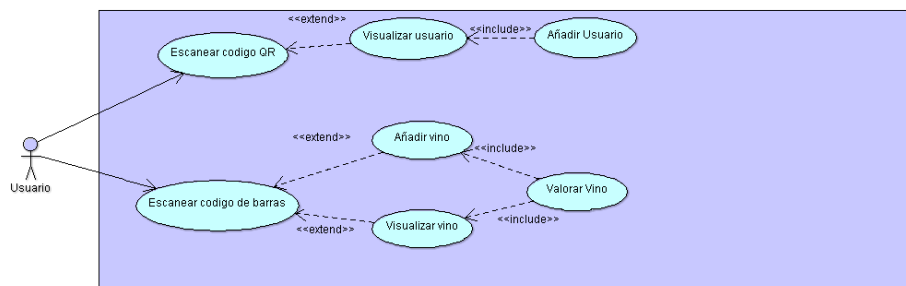
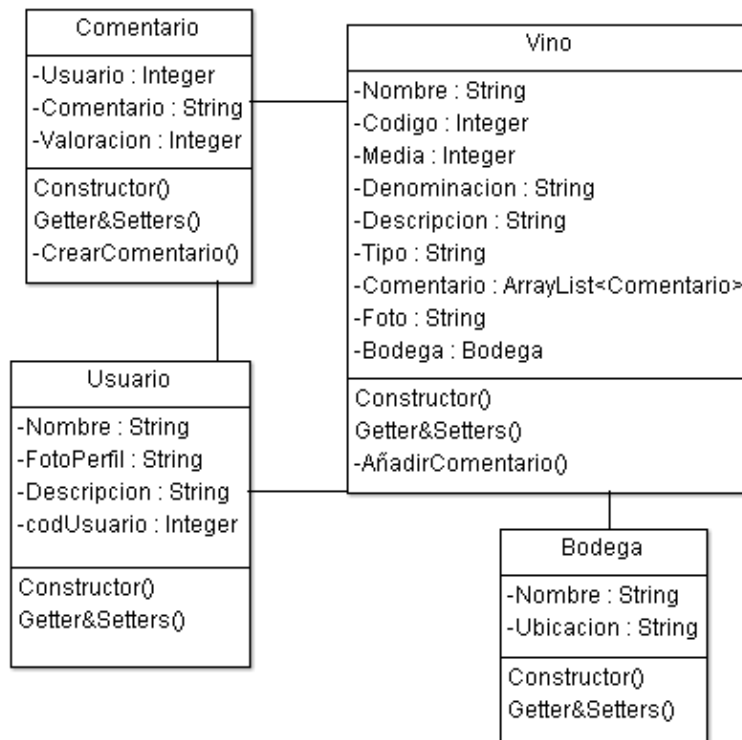


Diagrama de clases

A continuación, vamos a ver las clases de las que va a disponer la aplicación con sus métodos y sus atributos aparte de las uniones entre ellas.



Como podemos comprobar hemos creado la clase vino la cual contendrá una pila de comentarios para visualizar el comentario de cada usuario, el usuario tiene vínculo con la clase vino y con la clase comentario, ya que el usuario puede visualizar un vino y escribir un comentario en él. En el caso de las bodegas, guardaremos su nombre y su ubicación, la ubicación para poder acceder mediante el mapa y el nombre para poder encontrarlas en la base de datos. La clase comentario contendrá el comentario del usuario y su valoración. La clase Usuario guardará dentro de ella el nombre de este, su contraseña, el email y el codUsuario, el cual nos servirá para encontrarlo en la base de datos.

Diagrama de persistencia

Vamos a ver las ventajas y desventajas de una base de datos SQL y una base de datos NoSQL

Ventajas y desventajas de Base de datos SQL

Ventajas:

- **Madurez:** Al ser más antiguo tiene una gran variedad de información para poder realizar cualquier tipo de desarrollo o extracción de información, lo cual nos ayuda a desarrollar más rápidamente.
- **Atomicidad:** Esto quiere decir que cualquier operación realizada en la base de datos, garantiza que si a la mitad de cualquier operación de base de datos, surge algún tipo de problema, la operación no se completa.
- **Estándares bien definidos:** Por ejemplo, la creación de tablas, el insertar, eliminar y actualizar información, consultas, se escriben bajo la misma sintaxis, basados en el estándar de SQL.
- **Sencillez en la escritura:** Su principal aceptación, es su sencillez de escritura ya que se asemeja mucho al lenguaje humano.

Desventajas:

- **Crecimiento:** Cuando estas bases de datos tienden a crecer demasiado en el almacenamiento y el mantenimiento es sumamente difícil y costoso, suelen presentar fallas en tiempo de respuesta.
- **Cambios en la estructura:** Realizar cambios puede requerir mucho tiempo del sistema y unos conocimientos avanzados, ya que en ocasiones a la hora de realizar cambios debemos cambiar la estructura.
- **Complejidad en la instalación:** Puede ser complicado de instalar a la hora de que depende del sistema operativo, y no garantizan un buen funcionamiento si no se cumplen los requisitos mínimos.

Ventajas y desventajas base de datos NoSQL

Ventajas

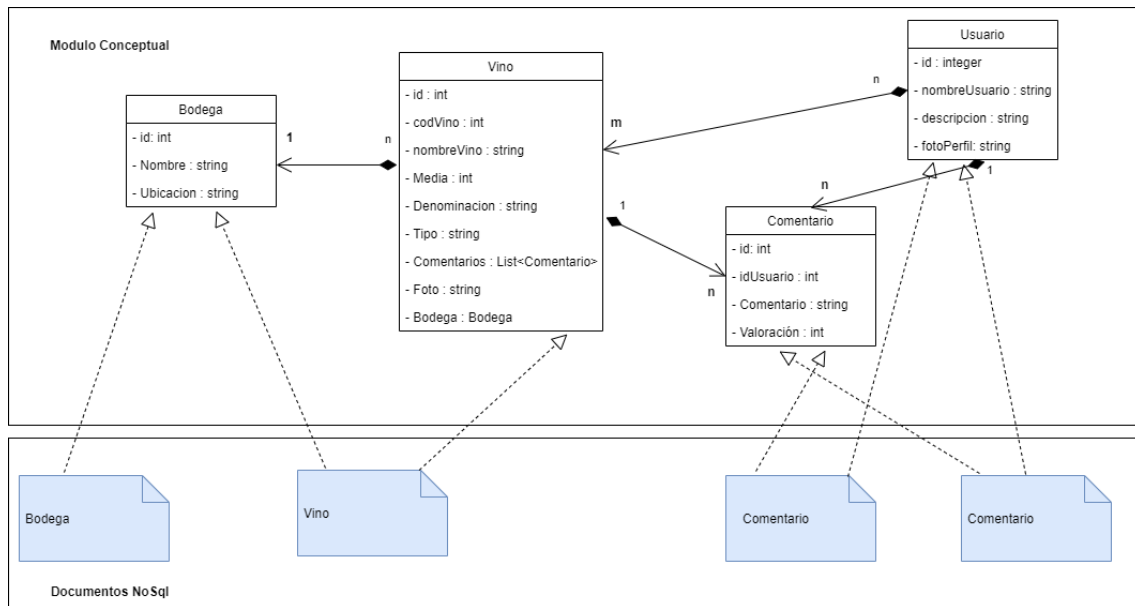
- **Versatilidad:** La principal ventaja por la cual esta nueva tecnología difiere de las demás soluciones de bases de datos es la versatilidad que ofrece a crecimientos o cambios sobre la forma como almacena la información, si es necesario agregar un nuevo campo sobre una “colección”, simplemente se agregan sobre el documento y el sistema sigue operando sin agregar configuraciones extras.
- **Crecimiento Horizontal:** Soportan una escalabilidad descentralizada, es decir, soportan estructuras distribuidas, si durante la operación se ve que el rendimiento del servidor baja, se instalan nuevos nodos operativos para que balanceen la carga de trabajo, a esto le llaman crecimiento horizontal.
- **Disponibilidad de Recursos:** No se requieren servidores con una gran cantidad de recursos disponibles para operar, pueden empezar a operar con bajos recursos e ir creciendo dependiendo de las necesidades del proyecto.
- **Optimización:** Tienen un algoritmo interno para reescribir las consultas escritas por los usuarios, esto con el fin de no sobrecargar el rendimiento de los servidores y mantener un nivel óptimo en las operaciones.

Desventajas

- **Atomicidad:** No todas las bases de datos contienen la característica de la atomicidad en la información, por lo tanto, puede ser diferente en cada uno de los nodos replicas que se puedan configurar en la arquitectura de base de datos.
- **Documentación del Software:** Dado que NoSQL, es relativamente nuevo, las operaciones pueden ser limitadas y se requiera de conocimientos avanzados con el uso de la herramienta.
- **Estándares en el lenguaje:** No se tiene un estándar definido entre los diferentes motores que ofrecen este servicio por lo tanto cada motor puede tener un estándar propio.
- **Herramientas GUI(Graphical User Interface):** Las herramientas que ofrecen para la administración de estas herramientas, no disponen de interfaz gráfica lo que requiere conocimiento amplio de las instrucciones a utilizar para su mantenimiento.

Para nuestra aplicación hemos decidido utilizar un sistema NoSQL ya que nos brinda la posibilidad de aumentar la base de datos y no tener que empezar con una base de datos excesivamente grande, sino ir desarrollándola poco a poco conforme la aplicación se va dando a conocer, además que nos brinda de una velocidad mayor permitiéndonos realizar operaciones en tiempo real si fuera necesario.

A continuación, vamos a mostrar el diagrama de persistencia el cual consta de dos modelos el modelo conceptual que describe las clases que se manejan a nivel aplicación y el modelo de documentos NoSQL que se crearan indicando los datos de las clases a almacenar, las flechas nos indican lo que vamos a guardar de cada clase, cuando apuntan a dos clases indica que están relacionadas entre ellas y que guardara los datos de una para hacer referencia a la otra



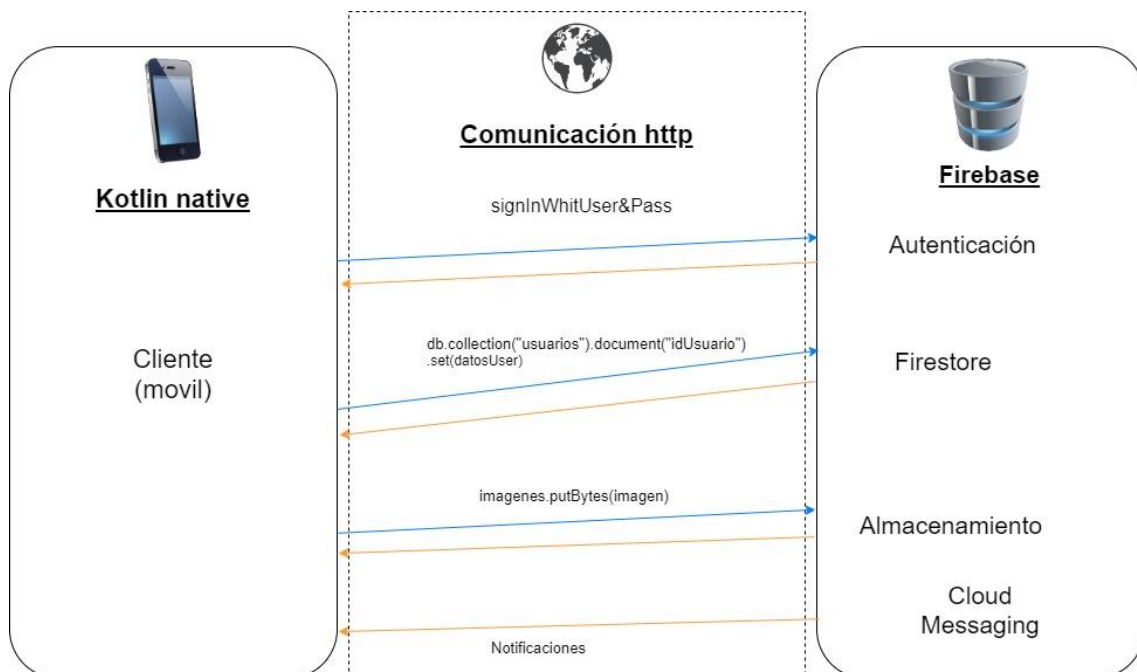
Arquitectura del sistema

La arquitectura del sistema es la parte en la que definiremos la unión entre nuestra aplicación y nuestra base de datos.

En nuestro caso vamos a utilizar una arquitectura basada en servicios la cual aporta flexibilidad, consiguiendo que se reduzcan los costes de la integración, mediante la automatización de las herramientas necesarias. Esta arquitectura SOA puede servir de apoyo a nuestra aplicación ayudándonos a agilizar los procesos, facilitando la adaptación al cambio y habilitando la posibilidad de implementar nuevas estrategias.

Es una arquitectura:

- Flexible, nos permite la reutilización.
- Versátil, hace posible que los servicios puedan consumirse en aplicaciones.
- Que aumenta la eficiencia de procesos, reduce los costes de mantenimiento y simplifica el diseño.



Esta será la arquitectura de nuestro sistema, tendremos una aplicación cliente que será de móvil y un servidor con el que se comunicará.

La comunicación se realizará mediante comunicación HTTP, es decir, utilizando el protocolo HTTP el cual nos permite realizar una petición de datos y recursos, con el fin de intercambiar datos entre la base y la

aplicación del cliente. Utilizaremos Firebase, por lo tanto, esta comunicación se realizará con los métodos de la librería de Firebase.

Firebase se trata de una plataforma para el desarrollo de aplicaciones móviles y web desarrollada por Google, se ubica en la nube y posee distintos servicios que veremos a continuación.

El cliente se realizará con la tecnología de Kotlin Native, estará escrito en lenguaje Kotlin. La tecnología de Kotlin Native nos permite crear aplicaciones nativas que funcionan tanto en Android como en IOS

Servicios Firebase:

Los servicios que contiene Firebase son los siguientes:

-Autenticación: Realizar la autenticación el usuario, el cliente le pasará un usuario y el método en cuestión, le devolverá si el usuario existe para esa contraseña, permitiendo así, acceder a la aplicación.

-Storage: Es la base de datos donde guardaremos los ficheros dentro de Firebase para luego acceder a ellos.

-Firestore: Es la base de datos en tiempo real de Firebase en la cual almacenaremos los datos del usuario en el momento que se actualicen.

-Cloud Messaging: es una solución de mensajería multiplataforma que nos permite enviar mensajes, mensajes que utilizaremos para las notificaciones de nuestra aplicación cliente.

5.Implementación de la solución

Análisis tecnológico

En este apartado explicaremos las tecnologías existentes que podríamos haber utilizado a la hora de realizar la aplicación y porque hemos elegido la tecnología que tenemos pensado utilizar actualmente, explicaremos las ventajas y desventajas que nos supone.

Para la aplicación móvil debemos usar una tecnología que nos sirva tanto para Android como para iOS, por lo tanto, no podremos ni usar Java ni usar Swift ya que tendríamos que crear dos aplicaciones, una para Android y otra para iOS lo cual haría que no fueran exactamente la misma, Llegados a este punto tenemos una lista de aplicaciones que podemos utilizar:

Kotlin Native

esta tecnología nos permite crear nuestra aplicación en código Kotlin compilándola a archivos binarios. Actualmente ofrece soporte a iOS, MacOS , Android y Windows entre muchos otros.



Ventajas

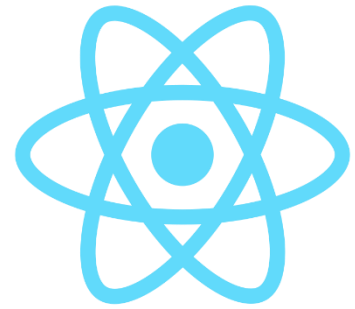
- Nos ofrece todas las características de un lenguaje de estilo orientado a objetos con un enfoque funcional
- La codificación en Kotlin es muy corta y directa. Mismo trabajo con menos líneas de código que en Java
- Está respaldado por JetBrains, Google y Spring Framework, definitivamente está aquí para quedarse, crecer y evolucionar

Desventajas

- Coincidencia de patrones débil
- Tamaño extra de tiempo de ejecución
- Comunidad de soporte pequeña

React Native

Se trata de un framework de JavaScript que permite crear aplicaciones nativas para IOS y para Android, esta basado en la librería de JavaScript React. Una de sus ventajas es que las aplicaciones tienen un aspecto profesional y su rendimiento es considerable, pero es necesario conocer cada plataforma a fondo y son propensas a tener una infinidad de bugs.



Ventajas

- Desarrollo rápido, es decir su tiempo de desarrollo es muy corto, ya que ofrece muchos componentes que pueden acelerar el proceso de desarrollo.
- Nos ofrece la posibilidad de desarrollar código tanto en IOS como en Android
- Es un framework muy utilizado por lo que se pueden encontrar una gran cantidad de fotos de desarrolladores para solucionar problemas.

Desventajas

- Aunque es muy utilizado siguen en estado beta por lo tanto pueden encontrarse problemas de compatibilidad con los demás paquetes de JavaScript
- Algunas de sus características y módulos necesitan que el desarrollador conozca Native y alguna de sus funcionalidades.

Flutter

Es un framework de código abierto que ha desarrollado google para la creación de aplicaciones nativas, de forma que realizas un único proyecto para todos los sistemas operativos sin embargo se trata de un framework muy joven y que aún es experimental por lo tanto muchas librerías están muy limitadas.



Ventajas

- Al realizar cambios en el código, se pueden ver los efectos inmediatamente, sin necesidad de compilar
- No es necesario construir un código por separado para las dos plataformas ya que genera un código base que sirve para IOS y para Android.

Desventajas

- Es necesario aprender el lenguaje de programación Dart
- Se trata de un framework joven que no tiene una gran comunidad detrás, por lo que no hay mucho soporte a la hora de encontrar un problema
- Las librerías a las que podemos acceder están muy limitadas.

En este caso utilizaremos Kotlin Native, porque a pese a sus desventajas se trata de un lenguaje que se asemeja a Java y al tratarse este de un lenguaje no será necesario invertir mucho tiempo en aprender otro lenguaje, además de que me ayudare a que el código sea más corto y mejor entendible, a eso hay que sumarle que cada vez se le va sumando más comunidad y que viene para quedarse.

A la hora de pensar en el servidor tenemos también varias posibilidades entre las que caben destacar:

SpringBoot

Permite que compilemos aplicaciones web como un archivo .jar que puede ejecutarse como una aplicación java normal lo cual es muy útil a la hora de realizar arquitecturas de micro servicios.



Ventajas

- Reduce el tiempo de desarrollo y aumenta la eficiencia a la hora de trabajar en equipo
- Nos proporciona un servidor HTTP embebido como Tomcat o Jetty que para desarrollar y testear la aplicación es muy útil
- Tiene una gran variedad de plugins que le permite trabajar con la mayoría de bases de datos tanto SQL como NoSQL

Desventajas

- No nos proporciona un sistema claro para poder realizar cambios en tiempo real en la base de datos.
- Debes entender el sistema Spring que usa por debajo SpringBoot para la modificación de algunas opciones avanzadas.

Firestore

Se trata de una plataforma creada por google cuya principal función es desarrollar y facilitar la creación de apps, dicha plataforma se encuentra en la nube y contiene diversas funciones para que el desarrollador pueda adaptarlas a sus necesidades.



Ventajas

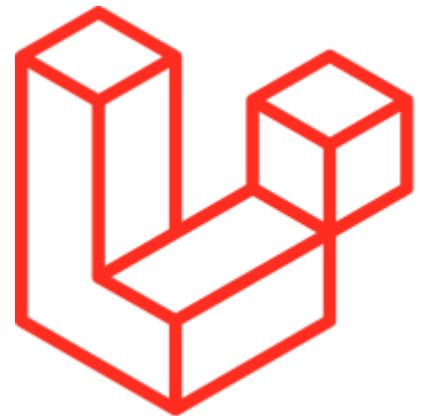
- Recomendable para aplicaciones que necesiten compartir en tiempo real
- Sus funcionalidades se complementan muy bien y se gestionan de forma sencilla desde un único panel
- Su soporte es gratuito vía email
- Sus inicios son gratuitos, pero permite adaptarse a las necesidades de la aplicación mediante planes de pago.
- Proporciona seguridad para todos los usuarios con los certificados SSL
- Permite a los desarrolladores centrarse menos en el backend para poder centrarse en otros aspectos.

Desventajas

- Posee restricciones con la versión gratuita a la hora de realizar un testing de la apk, solo 15 dispositivos por día de los cuales 5 serán dispositivos físicos y los demás serán virtualizados
- Ofrece muchas funciones con la versión gratuita, pero para tener acceso a todas las funcionalidades debemos pagar uno de los paquetes que nos ofrece.

Laravel

Es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.



Ventajas

- Desarrollo rápido: permite empezar desde una plantilla completamente funcional y obtener las características específicas para la aplicación.
- Al proporcionar una estructura de programación intuitiva para su aplicación y guiarlo hacia las mejores prácticas y patrones de desarrollo de software, un marco PHP lo guía hacia la escritura de código que se acople débilmente y se repita lo menos posible. Lo cual hace más fácil de leer, probar, mantener y desarrollar.
- Tiene una comunidad activa de desarrolladores y usuarios.

Desventajas

- Posee una ejecución lenta ya que al ser un framework php siempre agrega complejidad y sobrecarga en forma de clases y bibliotecas cargadas antes de que el código sea llamado.
- Requiere de tiempo para aprender y dominar, al tratarse de un marco de PHP es necesario entenderlo y entender sus patrones de diseño detrás de él.

En el caso del servidor he pensado en utilizar Firebase, gracias a su base de datos en tiempo real, al sistema de autenticación y a su almacenamiento en la nube, además de ser un sistema con una gran comunidad y que nos ofrece una extensa variedad de apps multiplataforma con Apis integradas a SDK individuales como puede ser Android e iOS.

Aspectos esenciales de la implementación

En este punto vamos a mostrar los aspectos esenciales a la hora de implementar. Añadiremos además imágenes con el código y una explicación. Lo que queremos conseguir en esta parte de la documentación es visualizar la parte importante del código y explicar los cambios que se realizarían en la base de datos o en la aplicación consiguiendo así un mejor entendimiento de esta.

loginUser():

```
//Iniciar sesion con email y pass
private fun loginUser() {
    //Obtenemos usuario y contraseña
    email = txtEmail.text.toString()
    password = txtPass.text.toString()
    //Verificamos que los campos no este vacios
    if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password)) {
        //Iniciamos sesión con el método signIn y enviamos usuario y contraseña
        mAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(this) {
                //Verificamos que la tarea se ejecutó correctamente
                task ->
                if (task.isSuccessful) {
                    guardarDatosUser()
                    // Si se inició correctamente la sesión vamos a la vista Home de la aplicación
                    goHome() // Creamos nuestro método en la parte de abajo
                } else {
                    // sino le avisamos el usuario que ocurrio un problema
                    Toast.makeText(
                        context: this, text: "Ha fallado la autenticacion",
                        Toast.LENGTH_SHORT
                    ).show()
                }
            }
    } else {
        Toast.makeText(context: this, text: "Introduzca los campos", Toast.LENGTH_SHORT).show()
    }
}
```

Este sería el código para poder logear a un usuario en Firebase, lo que hace es recoger el email y la contraseña que el usuario ha introducido en las cajas de texto de la aplicación, a continuación en caso de que las cajas no se encuentren vacias, llama con la variable mAuth(Variable de FirebaseAuth) al método signInWithEmailAndPassword, al cual le pasamos el email que el usuario ha introducido y la contraseña, esto realizara una llamada en Firebase que buscara el email introducido y en caso de que exista comparara las contraseñas, si el email existe llamara al método

guardarDatosUser, el cual mostraremos más adelante y a continuación llamara al método goHome, le cual nos lleva a la main activity.

guardarDatosUser():

```
//Metodo para guardar datos en preferencias y que se inicie sesion automaticamente
private fun guardarDatosUser() {
    val prefs =
        getSharedPreferences("Datos.Sesion", Context.MODE_PRIVATE).edit()
    prefs.putString("email", email)
    prefs.apply()
}
```

Este método nos crea una variable prefs, para editar el archivo de preferencias y guardar el email del usuario logueado, de tal forma que usando el método comprobarSession():

```
private fun ComprobarSession(): Boolean {
    //Comprobamos si existe una sesion iniciada
    val prefs = getSharedPreferences("Datos.Sesion", Context.MODE_PRIVATE)
    val email = prefs.getString( key: "email", defValue: null)
    return email != null
}
```

recogeríamos de nuevo el fichero de preferencias y comprueba si ese usuario se encuentra en él, en caso contrario devuelve null.

loginGoogle():

```
private fun loginGoogle() {
    val googleConf = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken("689444492010-2uiu8teu95mdjrvah0bqhdbbocn9rhiv.apps.goog...").requestEmail().build()

    val googleClient = GoogleSignIn.getClient(this, googleConf)
    googleClient.signOut()

    startActivityResult(googleClient.signInIntent, GOOGLE_SIGN_IN)
}
```

Con esta función buscamos logear al usuario mediante el proveedor de google, utilizaremos una variable en este caso googleConf, para guardar la configuración del inicio de sesión de google y otra variable, googleClient, la cual guardara el cliente que obtenemos con el contexto de la aplicación y la configuración mencionada anteriormente, hacemos un signout para que se cierre la sesión y el usuario no la mantenga en otros lados e iniciamos la actividad de google client, utilizando una statica como request

code en este caso, `GOOGLE_SIGN_IN` que será 100, debemos hacer un override de la función **onActivityResult**

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    //Request code del inicio de google
    if (requestCode == GOOGLE_SIGN_IN && resultCode == Activity.RESULT_OK) {
        val task = GoogleSignIn.getSignedInAccountFromIntent(data)
        try {
            val account = task.getResult(ApiException::class.java)
            if (account != null) {
                val credencial = GoogleAuthProvider.getCredential(account.idToken, null)
                mAuth.signInWithCredential(credencial).addOnCompleteListener(this) { task ->
                    if (task.isSuccessful) {
                        email = account.email ?: ""
                        val user: FirebaseUser = mAuth.currentUser!!
                        val photo = user.photoUrl
                        val currentUserDb = dbReference.child(user.uid)
                        val usuario = Usuario(user.displayName, getString(R.string.descripcion_predeterminada), photo.toString())
                        currentUserDb.setValue(usuario)
                        guardarDatosUser()
                        goHome()
                    } else {
                        Toast.makeText(
                            context: this, text: "Ha fallado la autenticacion",
                            Toast.LENGTH_SHORT
                        ).show()
                    }
                }
            }
        } catch (e: ApiException) {
            // sino le avisamos el usuario que ocurrio un problema
            Toast.makeText(
                context: this, text: "hola amigo",
                Toast.LENGTH_SHORT
            ).show()
        }
    }
}
```

Debemos hacer el override para recoger el requestCode de nuestro `GOOGLE_SIGN_IN`, en caso de que sea correcto y de que exista un resultado meteremos en resultado en una variable llamada account, esa variable contendrá la cuenta del usuario, guardaremos la credencial de la cuenta con el metodo **GoogleAuthProvider.getCredencial** y le daremos el idToken de la cuenta, después con la varibale de **FirebaseAuth(mAuth)** llamaremos al metodo **signInWithCredential**, al cual le pasaremos la credencial del usuario, en caso de que no de error guardaremos el email del usuario en una variable para guardarlo luego en el archivo de preferences con el metodo **guardarDatosUser()**, recogeremos también su Nick y su imagen de perfil, para ello nos deberemos crear una variable de **FirebaseUser**, en la cual guardaremos el usuario llamando a **mAuth.currentUser**, después cogeremos su Nick con **user.displayName** que nos devuelve el nombre del usuario en google y llamaremos también a **user.photoUrl** para obtener la Url de su foto de perfil, crearemos una referencia a la base de datos con la variable global **dbReference** guardara

una referencia a la base de datos de Firebase, para así poder guardar la información del usuario que se ha logueado con google en la base de datos, después guardamos los datos en preferences y vamos a la main activity.

createAccount():

```
//Vamos a crear nuestro método para crear una nueva cuenta
private fun createNewAccount() {
    //Obtenemos los datos de nuestras cajas de texto
    firstName = txtName.text.toString()
    lastName = txtlastName.text.toString()
    email = txtEmail.text.toString()
    password = txtPassword.text.toString()
    //Verificamos que los campos estén llenos
    if (!TextUtils.isEmpty(firstName) && !TextUtils.isEmpty(lastName)
        && !TextUtils.isEmpty(email) && !TextUtils.isEmpty(password))
    ) {
        if (password.length >= 6) {
            if (txtConfPassword.text.toString().equals(password)) {
                Toast.makeText(
                    context, this, text: "Registrando Usuario",
                    Toast.LENGTH_SHORT
                ).show()
                //vamos a dar de alta el usuario con el correo y la contraseña
                auth.createUserWithEmailAndPassword(email, password)
                    .addOnCompleteListener(this) { it: Task<AuthResult>
                        if (it.isSuccessful) {
                            val user: FirebaseUser = auth.currentUser!!
                            val currentUserDb = dbReference.child(user.uid)
                            val usuario = Usuario( nickname: "$firstName $lastName", getString(R.string.descripcion_predeterminada), getString(R.string.foto_predeterminada))
                            currentUserDb.setValue(usuario)
                            //Por último nos vamos a la vista home
                            updateUserInfoAndGoHome()
                        }
                    }.addOnFailureListener { it: Exception
                        // si el registro falla se mostrara este mensaje
                        Toast.makeText(
                            context, this, text: "El email introducido ya esta siendo utilizado",
                            Toast.LENGTH_SHORT
                        ).show()
                    }
            }
        } else {
            txtPassword.text.clear()
            txtConfPassword.text.clear()
            Toast.makeText(
                context, this, text: "La contraseñas no coinciden",
                Toast.LENGTH_SHORT
            ).show()
        }
    } else {
        Toast.makeText(
            context, this, text: "La contraseña debe contener 6 o mas caracteres",
            Toast.LENGTH_SHORT
        ).show()
    }
} else {
    Toast.makeText( context, this, text: "Llene todos los campos", Toast.LENGTH_SHORT).show()
}
}
```

En esta función recogemos los datos del usuario a registrar y realizamos las comprobaciones, si esta vacío, las contraseñas no coinciden, o la contraseña tiene más de 6 caracteres si no se cumplen estas condiciones, llamaremos al método de FirebaseAuth CreateUserWithEmailAndPassword al cual le pasaremos el email y la contraseña que el usuario a introducido, utilizaremos una variable de FirebaseDatabase para obtener la referencia a la base de datos y

guardamos en Usuarios/user.uid, en esa dirección que le hemos dado vamos a guardar el usuario con set value, después llevaremos al usuario a la pantalla de login.

subirImagenAFirebase()

```
private fun subirImagenAFirebase(){
    if(photoUri == null){
        guardarUsuario(user!!.getFotoPerfil())
    }else {
        val filename = UUID.randomUUID().toString()
        val ref = FirebaseStorage.getInstance().getReference( location: "/imagenes/$filename")

        ref.putFile(photoUri!!).addOnSuccessListener { it: UploadTask.TaskSnapshot!
            ref.downloadUrl.addOnSuccessListener { it: Uri!
                guardarUsuario(it.toString())
            }
        }
    }
}
```

Este método lo usamos en editar perfil, el método se utiliza para subir una imagen a Firebase Storage metiendo la uri en la base de datos, después recogeremos la url de Firebase Storage para tener acceso a la imagen.

6. Testeo y pruebas de la solución

Plan de pruebas

Un plan de pruebas es el producto que define los objetivos de la prueba de un sistema para así establecer y coordinar la estrategia de trabajo las distintas pruebas que existen son:

- **Pruebas unitarias:** Tienen como objetivo verificar la funcionalidad de cada componente individualmente una vez ha sido programado constituyen la prueba inicial del sistema, existen dos enfoques el **enfoque estructural**, que verifica la estructura interna del componente con independencia y el **enfoque funcional** que comprueba que los componentes del sistema funcionen de forma correcta.
- **Pruebas de integración:** Verifican el correcto ensamblaje entre los distintos componentes una vez probado unitariamente de este tipo de prueba existen dos tipos **incremental**, es decir se combina el siguiente componente a probar con los demás que están comprobados, de esta forma comprobamos si el error se encuentra en la comunicación de estos, y el otro tipo es el **no incremental**, se prueba cada componente por separada y se integran todos cuando se han realizado las pruebas
- **Pruebas del sistema:** Buscan ejercitar el sistema comprobando la integración del mismo de forma global, comprobando si funciona correctamente el sistema y su comunicación con otros sistemas
- **Pruebas de implantación:** Comprueban el funcionamiento del sistema integrado en el entorno de producción.
- **Pruebas de aceptación:** Validan que el sistema cumpla con el funcionamiento esperado
- **Pruebas de regresión:** Comprueban si los cambios de un componente introducen un comportamiento no deseado o errores

A continuación, vamos a ver las distintas pruebas que tenemos pensado realizar en nuestra aplicación para comprobar el correcto funcionamiento de ella

- **Pruebas unitarias**

- + Caso de prueba 1: Para este caso de prueba realizaremos el registro del usuario. Comprobaremos que el usuario pueda registrarse sin problema, guardando sus datos y su email en la base de datos además probaremos a registrar un usuario con un email que se encuentre ya usado en la base de datos.

- + Caso de prueba 2: En este caso comprobaremos el login, si el usuario registrado accede correctamente y que al meter credenciales invalidas nos envíe un mensaje de error.

- + Caso de prueba 3: Comprobaremos el login de google, si el usuario se registra correctamente mediante ese proveedor y guardamos su información.

- + Caso de prueba 4: Realizaremos las pruebas del perfil, comprobaremos que el usuario accede correctamente a su perfil y que el perfil carga correctamente sus datos de la base de datos.

- + Caso de prueba 5: Comprobaremos que el usuario puede cambiar su perfil, añadiendo una imagen, cambiando la predeterminada y su descripción, posteriormente comprobaremos que esa imagen y los nuevos datos se han guardado en la base de datos.

- **Pruebas de seguridad**

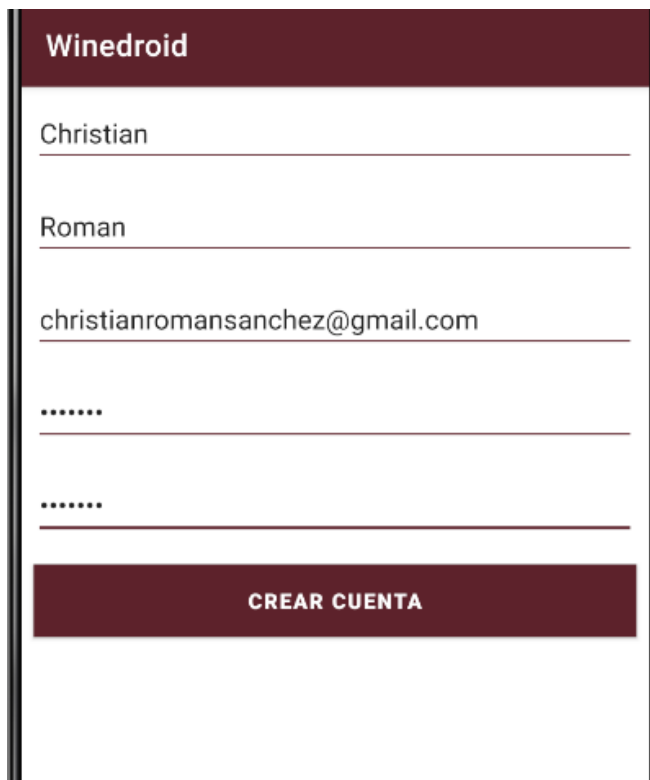
- + Caso de prueba 6: Se comprobará si un usuario no registrado puede acceder a la aplicación

A continuación, realizaremos las pruebas anteriormente citadas:

Caso de prueba 1

Registraremos un usuario introduciendo un correo valido que no existe en la base de datos

El resultado esperado es que la operación se realice sin inconvenientes



The screenshot shows a registration form titled "Winedroid". It contains five input fields: a name field with "Christian", a surname field with "Roman", an email field with "christianromansanchez@gmail.com", a password field with six dots, and a confirmation password field with six dots. Below the fields is a dark red button labeled "CREAR CUENTA".

Una vez hemos introducido los datos y hemos creado nuestra cuenta buscaremos si el mismo usuario se nos ha generado en Firebase

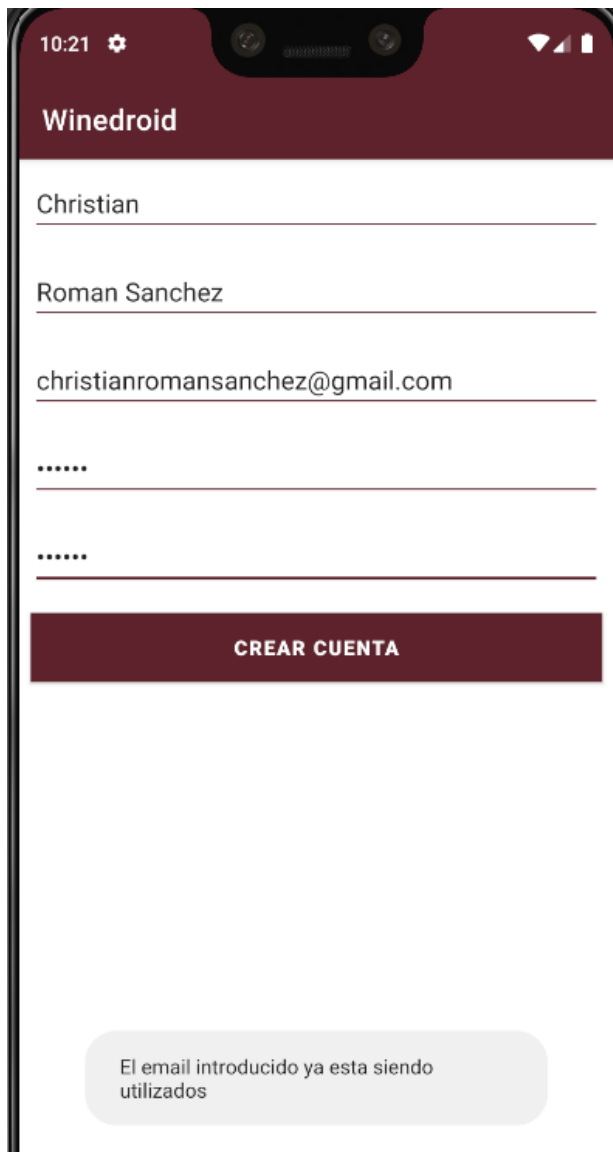
christianromansanchez@gm... 7 ene. 2021 7 ene. 2021 bWiZM0bkgQZN6yzm77ufygjFKYA3

Como hemos comprobado se nos ha guardado un usuario y le ha generado un Uid, que utilizaremos a continuación para guardar sus datos en Realtime Database, la descripción y la fotografía serán dos predeterminadas

```
bWiZM0bkgQZN6yzm77ufygjFKYA3
{
  descripcion: "Edita tu perfil para cambiar tu descripcion"
  fotoPerfil: "https://firebasestorage.googleapis.com/v0/b/win..."
  nick: "Christian Roman"
}
```

Como hemos comprobado la operación se realiza sin problemas

A continuación, vamos a meter un email repetido en la base de datos
El resultado esperado es que la aplicación no nos deje y devuelva un mensaje de error



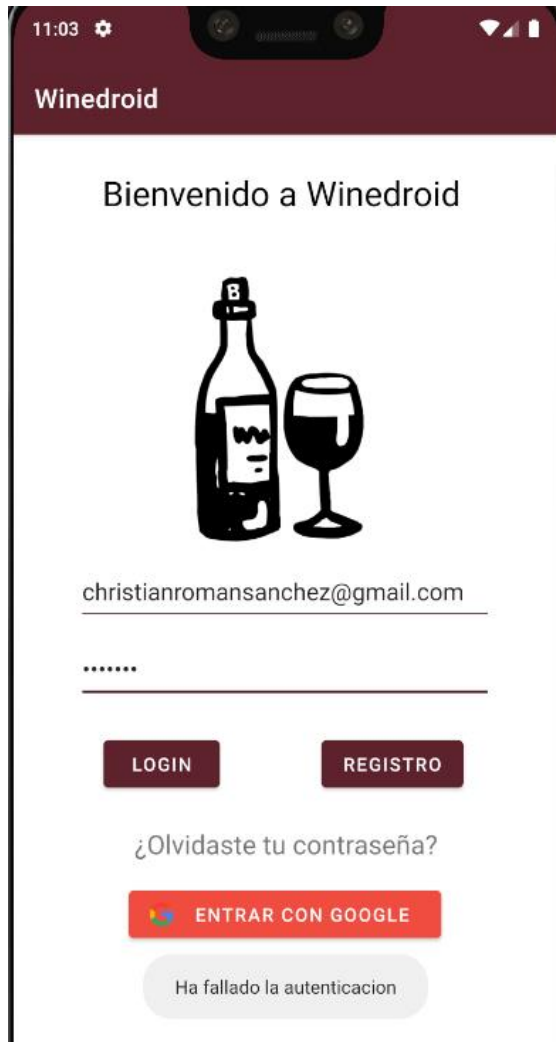
Como podemos visualizar nos devuelve un error explicándonos que el email introducido ya está siendo utilizado, por lo tanto, no podemos registrar a ese nuevo usuario, y al comprobar la base de datos podemos observar que los datos del usuario anterior no han variado.

```
bWiZM0bkgQZN6yzm77ufygjFKYA3
[
  ..... descripcion: "Edita tu perfil para cambiar tu descripcion"
  ..... fotoPerfil: "https://firebasestorage.googleapis.com/v0/b/win..."
  ..... nick: "Christian Roman"
```

Caso de prueba 2

Vamos a probar con el usuario que hemos introducido anteriormente primero probaremos a logear con una contraseña incorrecta:

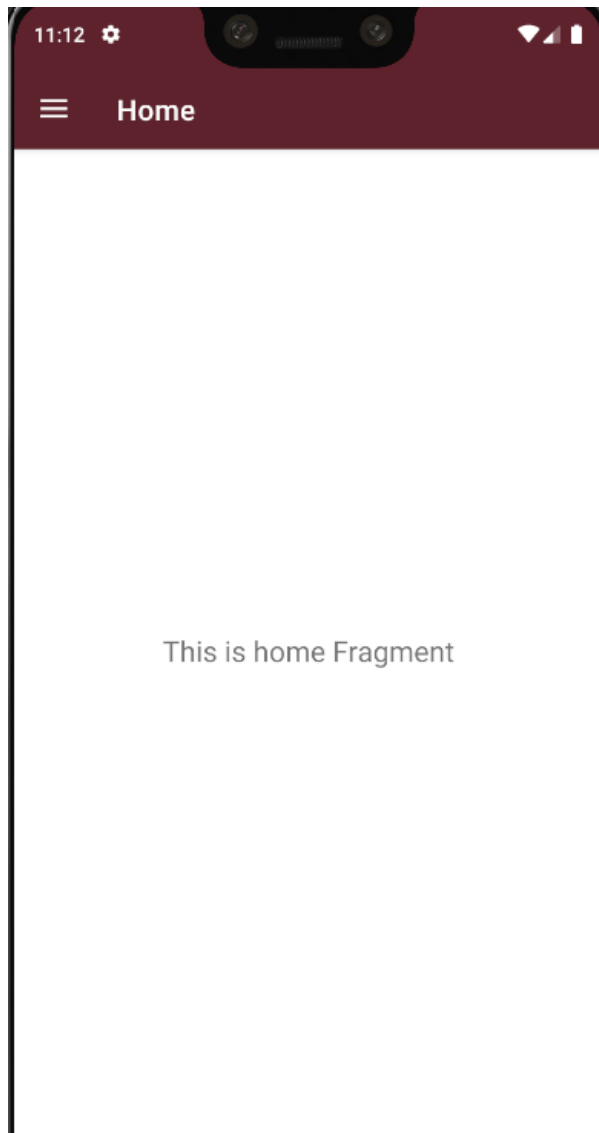
El resultado esperado es que falle y nos devuelva un mensaje de error



Como podemos ver nos devuelve un mensaje diciendo que ha fallado la autenticación.

A continuación, introduciremos la contraseña correctamente

El resultado esperado en este caso es que el usuario logue correctamente y acceda al home de la misma.



Accedemos a la pantalla principal de la aplicación por lo tanto el login se realiza correctamente


Caso de prueba 3

En la pantalla de login pulsaremos en entrar con google

El resultado esperado es que el usuario entre sin problemas y su información se guarde en Firebase

Winedroid

Bienvenido a Winedroid




Correo

Contraseña

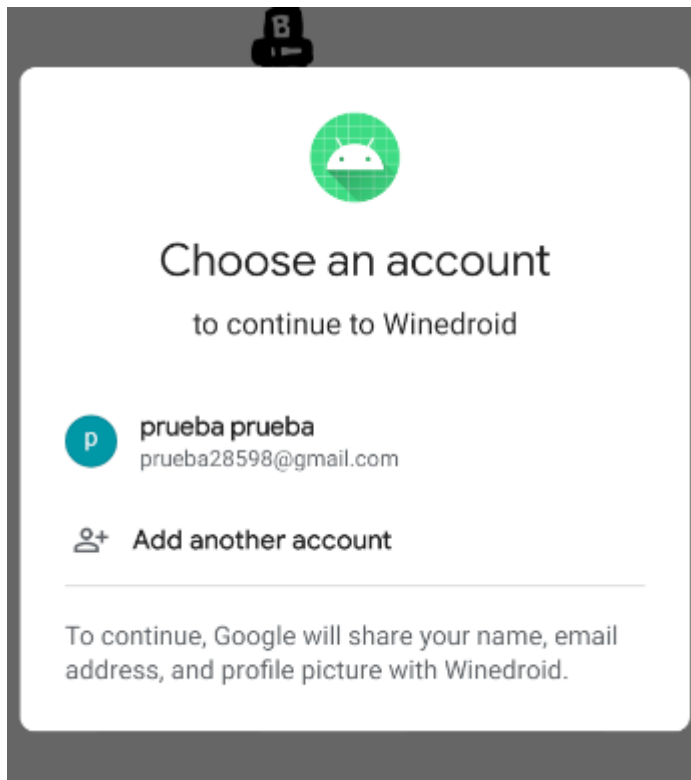
LOGIN

REGISTRO

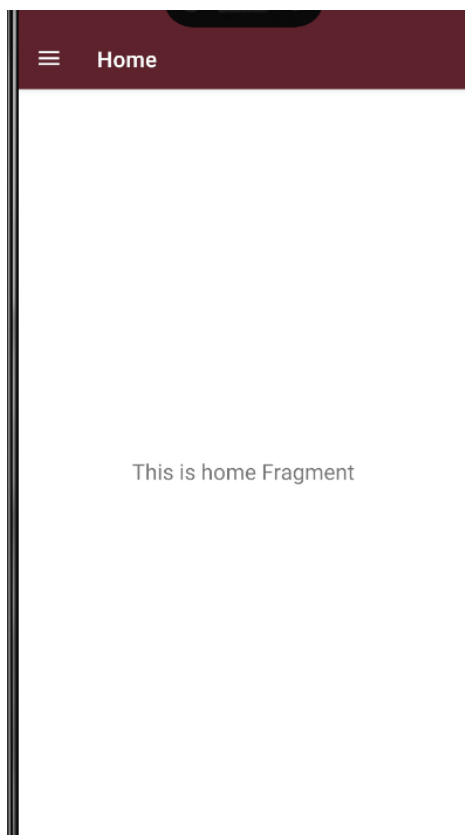
¿Olvidaste tu contraseña?

 ENTRAR CON GOOGLE

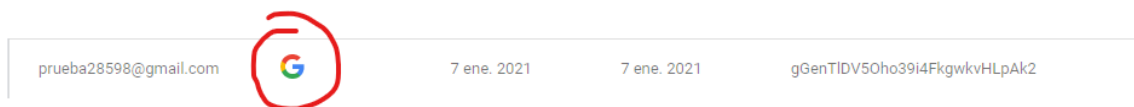
Nos saldrá el siguiente dialogo:



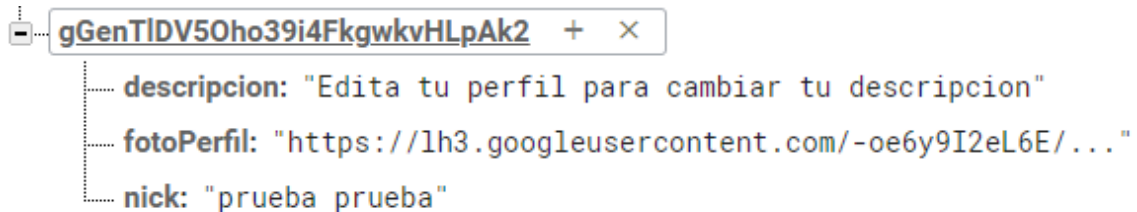
Seleccionamos la cuenta o añadimos otra y a continuación el usuario accederá a la pantalla de home



Ahora vamos a comprobar Firebase



Como podemos ver el proveedor ha sido google y la cuenta se ha creado, generando el uid del usuario y si comprobamos Realtime Database:

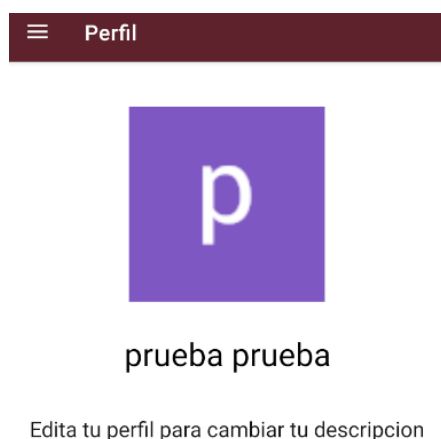


Comprobamos que hemos recogido sus datos de google obteniendo así su Nick y su foto de perfil correctamente.

Caso de prueba 4

Vamos a comprobar que el usuario accede a su perfil y le carga la información correctamente de la base de datos:

Resultado esperado: El usuario puede acceder a su perfil y le cargara los datos introducidos o recogidos de Google



Como podemos ver el usuario accede sin problemas a su perfil y visualiza su Nick, su imagen de perfil y la descripción predeterminada

GENERAR QR

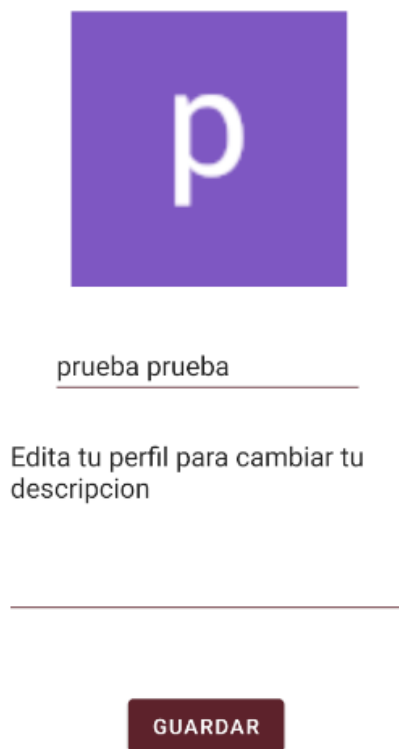
EDITAR PERFIL

Caso de prueba 5:

Comprobaremos que el usuario puede editar su perfil, cambiando su nick, su descripción y su imagen de perfil

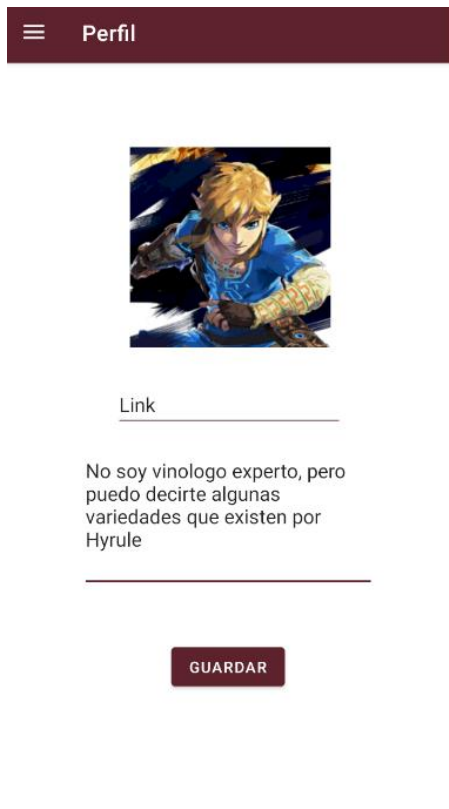
Resultado: El usuario edita su perfil con éxito y la información se guarda en Firebase

Accedemos a editar perfil desde perfil:

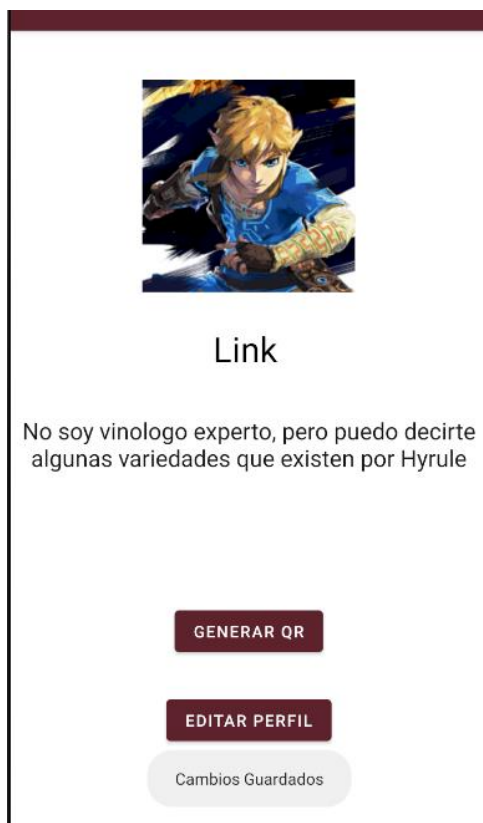


Cambiaremos los datos, del nombre y la descripción

A continuación, seleccionaremos una imagen clicando en nuestra foto de perfil



Al pulsar en guardar nos lleva a la pantalla anterior y nos muestra un mensaje de cambios guardados



Ahora comprobaremos los datos en Realtime Database de Firebase:

gGenTIDV50ho39i4FkgwkvHLpAk2

descripcion: "No soy vinologo experto, pero puedo decirte alg..."

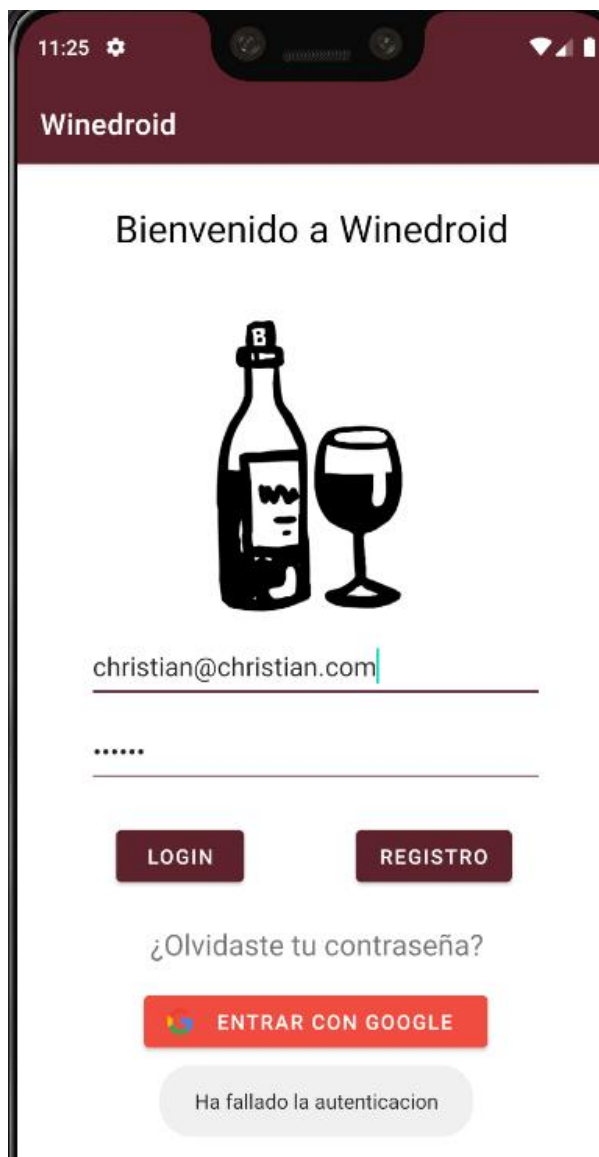
fotoPerfil: "https://firebasestorage.googleapis.com/v0/b/win..."

nick: "Link " ×

Caso de prueba 6:

Intentamos acceder a la aplicación con un usuario no registrado

Resultado: la aplicación devolverá un error de autenticación



El caso se realiza correctamente ya que devuelve el error

Solución a problemas encontrados:

La verdad es que sinceramente no me han surgido muchos problemas a la hora de programar, pues Kotlin es un lenguaje muy parecido a Java por lo que no había muchas cosas extrañas, además que tiene una comunidad activa que ayuda bastante a la hora de tener un problema, sin embargo, si he tenido algún que otro problema a la hora de programar, como puede ser el caso de la imagen, no era capaz de visualizar la imagen al tratarse de una url, ya que en Realtime Database guardamos una referencia a esa imagen con su dirección html, por lo tanto tuve que encontrar una solución, que en este caso fue utilizar la librería Picasso para recoger la imagen de una ruta HTML.

7.Lanzamiento y puesta en marcha

Aspectos relevantes del despliegue

Aquí mostraremos los aspectos más relevantes del despliegue tanto del servidor como de nuestra aplicación cliente para entender mejor a lo que nos referimos, vamos a poner una pequeña definición:

El Despliegue de software son todas las actividades que hacen que un sistema de software esté disponible para su uso.

El proceso de implementación general consiste en varias actividades interrelacionadas con posibles transiciones entre ellas. Estas actividades pueden ocurrir en el lado del desarrollador de software o en el lado del consumidor o en ambos. Debido a que cada sistema de software es único, los procesos o procedimientos precisos dentro de cada actividad difícilmente pueden definirse. Por lo tanto, la "implementación" debe interpretarse como un proceso general que debe personalizarse de acuerdo con los requisitos o características específicos.

Las actividades de despliegue son:

- **Lanzamiento:** La actividad de publicación se deriva del proceso de desarrollo completado y, a veces, se clasifica como parte del proceso de desarrollo en lugar del proceso de implementación. Incluye todas las operaciones para preparar un sistema para el compilado y la transferencia a los sistemas informáticos en los que se ejecutará en producción.
- **Instalación y activación:** Para sistemas simples, la instalación implica establecer algún tipo de comando, acceso directo, script o servicio para ejecutar el software (manual o automáticamente). Para sistemas complejos puede implicar la configuración del sistema, posiblemente haciendo preguntas al usuario final sobre su uso previsto, o preguntándoles directamente cómo les gustaría que se configuren, y / o haciendo que todos los subsistemas necesarios estén listos para usar. La activación es la actividad de iniciar el componente ejecutable de software por primera vez.
- **Desactivación:** La desactivación es la inversa de la activación y se refiere al cierre de cualquier componente que ya se esté ejecutando de un sistema.

- **Desinstalación:** La desinstalación es el inverso de la instalación. Es la eliminación de un sistema que ya no es necesario
- **Actualización:** El proceso de actualización reemplaza una versión anterior de todo o parte de un sistema de software con una versión más reciente
- **Seguimiento de versiones:** Los sistemas de seguimiento de la versión ayudan al usuario a encontrar e instalar actualizaciones a los sistemas de software.
- **Adaptación:** La actividad de adaptación también es un proceso para modificar un sistema de software que se ha instalado anteriormente.

Entendiendo los puntos anteriores podemos visualizar cual sería el despliegue de nuestra aplicación, la cual debemos entender como dos partes, el backend (nuestra aplicación de servidor a la cual realizamos las peticiones de datos en este caso Firebase) y nuestra aplicación cliente(que sería la aplicación móvil que estamos desarrollando), en la parte del backend al seleccionar como servidor Firebase, todas las configuraciones que debemos realizar, las realiza automáticamente, por lo tanto para empezar solo debemos acceder a la consola de Firebase y pulsar en añadir proyecto, a continuación seguiremos los pasos:

× Crear un proyecto(paso 1 de 3)

Empieza por ponerle un nombre al proyecto[?]

Introduce el nombre de tu proyecto

Add Firebase to one of your existing Google Cloud Platform projects
 [Learn more](#)

 My Project

Continuar


Introducimos el nombre al proyecto


× Crear un proyecto(paso 2 de 3)


Google Analytics para tu proyecto de Firebase


Google Analytics es una solución de analíticas gratuita e ilimitada que permite crear informes y realizar tareas de segmentación, entre otras acciones, en Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions y Cloud Functions.


Google Analytics permite usar:


 A/B Testing ⓘ

 Segmentación de usuarios en productos de Firebase ⓘ

 Predicción del comportamiento de usuarios ⓘ

 Usuarios sin fallos ⓘ

 Activadores de Cloud Functions basados en eventos ⓘ

 Informes gratuitos ilimitados ⓘ

☒ Habilitar Google Analytics en este proyecto
Opción recomendada

[Anterior](#)


Continuar

Decidimos si habilitar analytics o no

× Crear un proyecto(paso 3 de 3)

Configurar Google Analytics

Selecciona o crea una cuenta de Google Analytics ⓘ

 Default Account for Firebase ▼

Crear automáticamente una propiedad en esta cuenta ✎

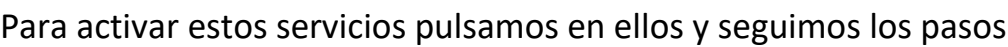
Al crear el proyecto, también se creará una propiedad de Google Analytics en la cuenta de este servicio que hayas elegido, que se asociará a tu proyecto de Firebase. De esta forma, se habilitará el flujo de datos entre los productos. Los datos que se exportan de tu propiedad de Google Analytics a Firebase están sujetos a los términos del servicio de Firebase, mientras que los datos que se importan de Firebase a Google Analytics se rigen por los términos del servicio de Google Analytics. [Más información](#)

[Anterior](#)

Crear proyecto

Y damos clic en crear proyecto y ya tendríamos nuestro proyecto creado
A continuación, añadiremos una aplicación siguiendo los pasos que te dan:

y le daremos los servicios que queramos seleccionándolos en la barra de la izquierda



Para la parte del cliente debemos poner la aplicación a disposición de los usuarios mediante las tiendas de aplicaciones, utilizaríamos PlayStore para teléfonos Android y AppStore para teléfonos IOS.

Para poder subir nuestra app a PlayStore debemos seguir los siguientes pasos:

- Crear la apk de nuestra aplicación
- Debemos tener una cuenta de desarrollador, es decir, realizar un pago de 25 dólares para disponer de una cuenta de desarrollador con la que podremos acceder a Google Play y publicar nuestras aplicaciones.
- Accederemos a la consola de desarrolladores de Google play, registraremos y pulsaremos en crear nueva app, donde crearemos el perfil de la nueva app.
- Elegiremos el idioma predeterminado para el resto de detalles a introducir, texto, imágenes etc. y escribiremos el nombre de la aplicación, el cual debe ser distinto al nombre de la apk y podemos traducirlo a varios idiomas
- A continuación, podremos aportar información más completa sobre la aplicación, incluyendo un título breve y una descripción completa. Toda esta información se mostrará a los usuarios en Google play
- Subiremos el icono de la aplicación y capturas de pantalla de está mostrando su uso, es necesario un mínimo de dos y hasta un total de 8 por categoría (teléfono, Tablet, Android TV o wear OS) añadiremos además la categoría de la aplicación y podremos añadir etiquetas
- Subiremos el archivo de la aplicación creando una nueva versión, en el apartado de Versiones de aplicación podemos crear una nueva versión de producción o una beta, alfa, prueba interna o pre registro, para publicarla como versión estable hacemos clic en gestionar en el apartado de producción.
- Subimos el archivo apk o el Android App Bundle (formato AAB). Subimos el archivo a la consola de desarrolladores de Google Play
- La consola extraerá algunos detalles como la versión, sin embargo, la lista de cambios deberemos rellenarla nosotros pudiendo incluir tantas traducciones como queramos

- Los últimos pasos serian seleccionar si queremos añadirla de pago o no, deberemos pasar la clasificación de contenido, el contenido y audiencia objetivo y por ultimo después de pasar todos estos apartados podremos enviar la aplicación a google para su revisión, si todo va bien la aplicación será visible en las próximas 24 horas

Para poder subir nuestra aplicación a la App Store de Apple debemos:

- Registrarnos como particular o como empresa en el programa de desarrolladores de Apple introduciendo nuestra tarjeta de crédito.
- Crear los certificados Apple para la aplicación después de iniciar sesión en Apple. Estos se generan a través del programa Xcode o de tu cuenta de desarrollador de Apple.
- Registrar un dispositivo móvil de Apple en tu cuenta de desarrollador.
- Generar un id. Otro de los requisitos para publicar una app en la Apple Store es el app ID
- Crear un perfil de aprovisionamiento
- Publicar la aplicación a través de iTunnes Connect, para ello daremos clic en myapps y luego a añadir. Rellenamos los datos y la creamos
- Diseñaremos la página de la descripción de la app donde debemos cargar también el archivo build que generamos con Xcode, definiremos además su icono
- Esperamos a que la aplicación sea revisada y a los dos días la aplicación pasara a estar disponible

Manual de usuario

La idea de un manual de usuario es dar asistencia a las personas que utilizar el sistema por lo general está redactado por un escritor técnico la mayoría de los manuales de usuario contienen tanto una guía escrita como imágenes asociadas a esta. En el caso de aplicaciones informáticas es habitual incluir capturas de pantalla de como el programa

Login de la aplicación



Bienvenido a Winedroid



User

Pass

Login

Register

He olvidado mi contraseña

 Login with Google

Introducimos el usuario

Introducimos la contraseña

Con este botón accedemos al registro de la aplicación

En caso de olvidar la contraseña accederemos aquí

Entramos a la aplicación usando un correo de Google

Si el usuario y la contraseña son correctos accederemos a la aplicación

Registro de la aplicación

The diagram shows a mobile application registration screen. At the top, a dark red header bar contains the time '15:45' and status icons. Below the header, the title 'Registro' is centered. The form consists of five input fields: 'Nombre', 'Apellido', 'Contraseña', 'Confirmar contraseña', and 'Email'. To the right of the form, five text boxes provide instructions for each field, connected by blue arrows. The first four instructions are in white boxes, while the last one is in a purple box. At the bottom of the form is a dark red button labeled 'Registro'. A blue arrow points from a text box describing the button's function to this button.

15:45

Registro

Nombre

Apellido

Contraseña

Confirmar contraseña

Email

Registro

Introduce tu nombre

Introduce tu apellido

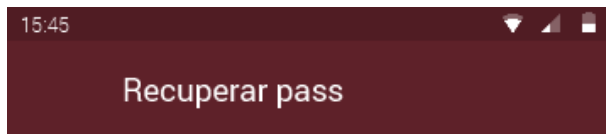
Introduce tu contraseña

Confirma tu contraseña

Introduce el email que se vinculara a la cuenta

Al pulsar el botón registrara su usuario con el email y la contraseña y guardara los datos en la base de datos para que pueda acceder en cualquier momento

Registro de la aplicación



Introduce el correo enlazado a la cuenta
para recuperar tu contraseña

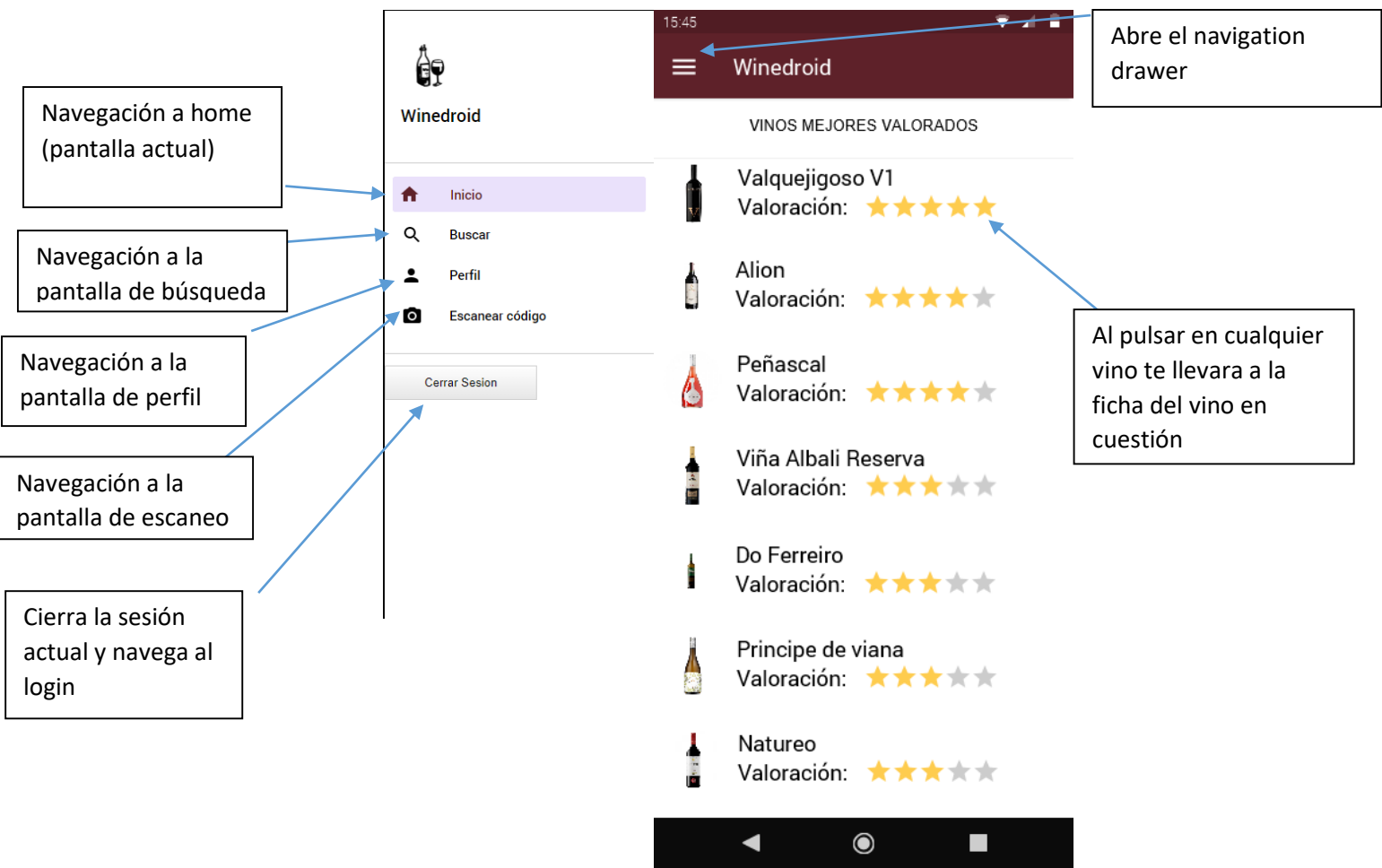
Introduces tu email

Enviar

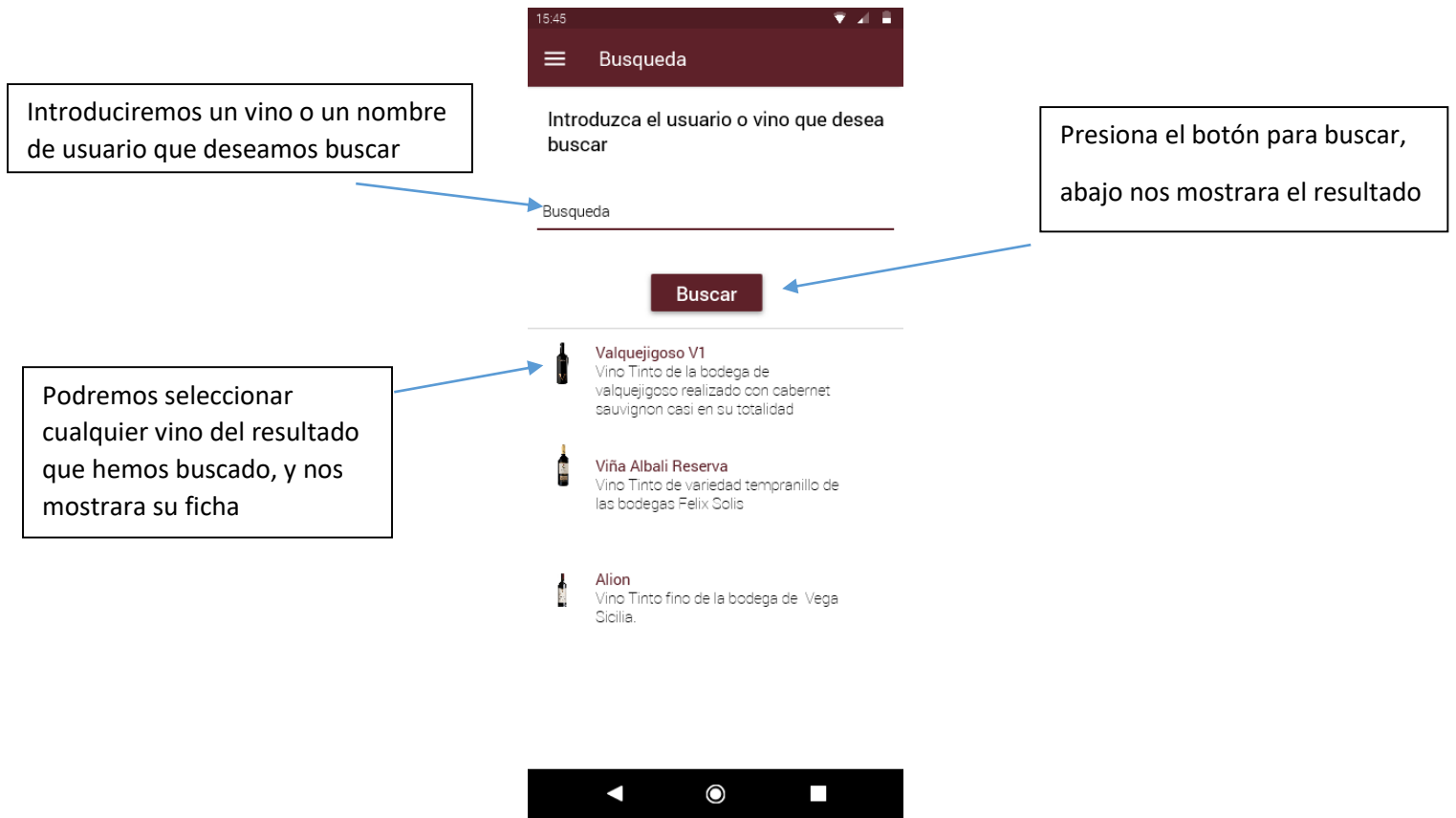
Se enviará un correo a tu
email con el que podrás
recuperar tu contraseña



Navigation drawer y Home de la aplicación



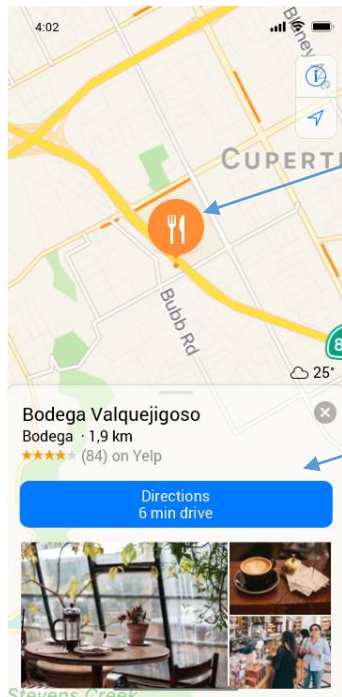
Pantalla de búsqueda



Ficha de vino



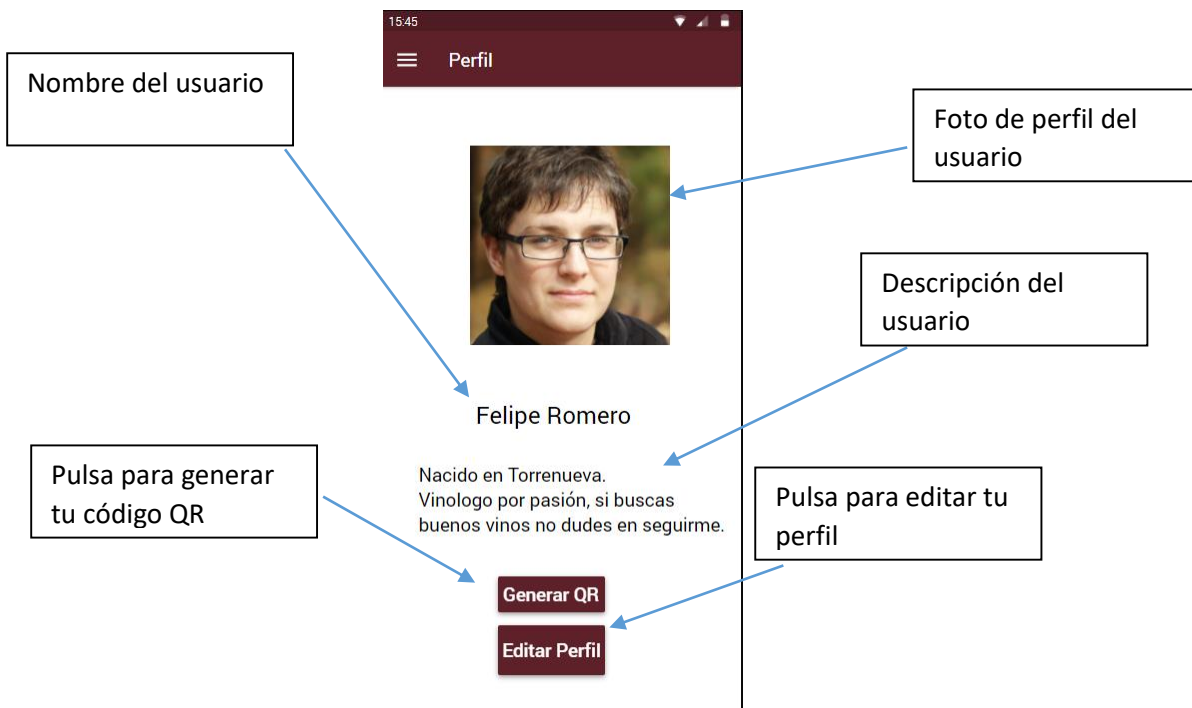
Mapa



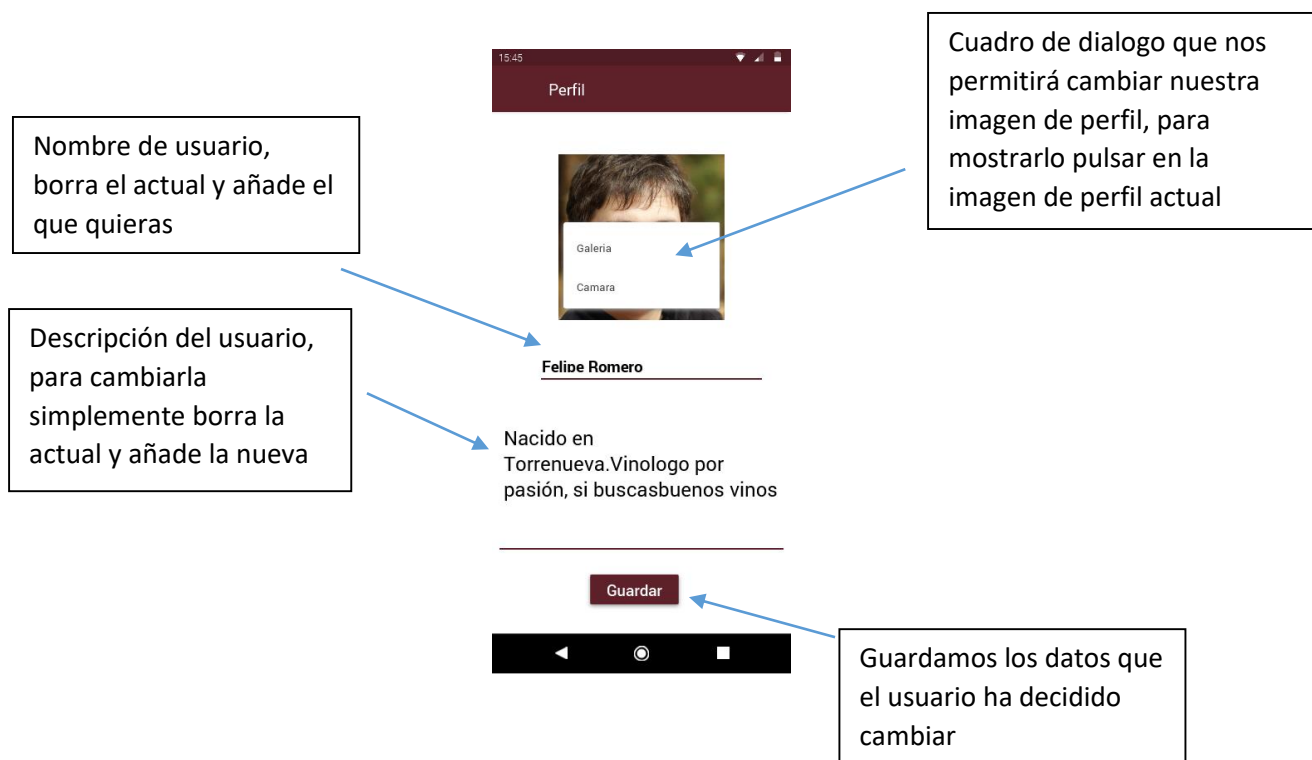
Punto donde se encuentra la bodega.

Pulsa para que te dirija a la bodega.

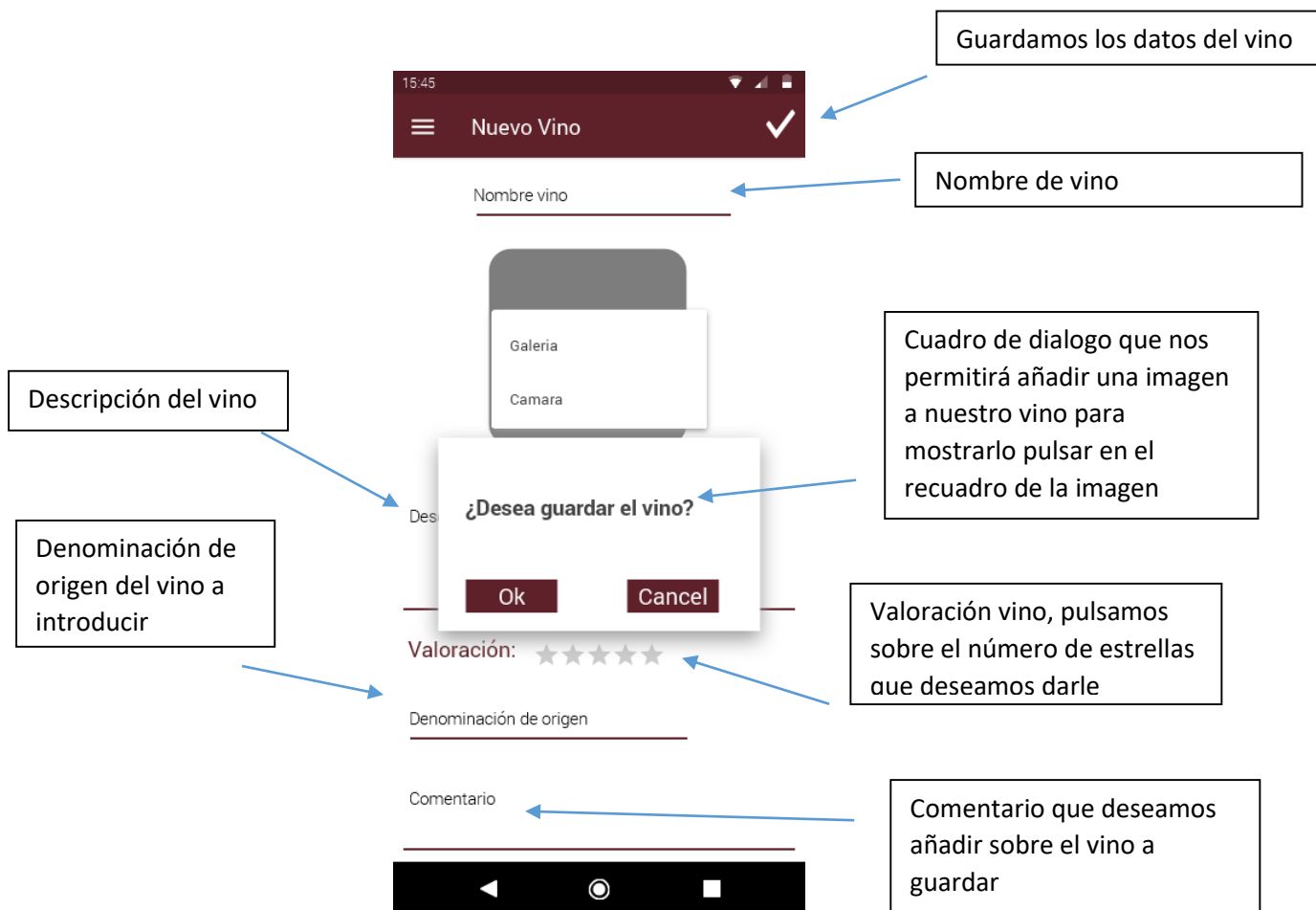
Perfil



Editar Perfil



Crear ficha vino



8. Valoración y conclusiones

Cuando vi el enunciado que me entregaron me pareció un proyecto demasiado complicado, era realizar una aplicación de 0 con cosas que no había visto nunca, con otras que llevaba sin ver desde hace tiempo y todo ello aprendiendo un lenguaje nuevo, la verdad es que lo veía prácticamente imposible, me veía que no iba a llegar al momento si quiera de entregar esta documentación.

Empezamos escribiendo documentación, lo cual de por si se me hace pesado y no se me da especialmente bien, lo primero fue sencillo teníamos que pensar que queríamos que realizara la aplicación, teníamos unos requisitos que debíamos cumplir, pero todo lo demás era libre, en eso me ha ayudado bastante tener entregas cada cierto tiempo y que el tutor contestase a mis dudas.

En este trabajo he tocado todos los palos de nuestro grado he realizado cosas de EIE por ejemplo cuando tuvimos que analizar a que target iba dirigido nuestra aplicación, o FOL de primero, al tener que ver los riesgos laborales del mundo de la informática y como prevenirlos. He utilizado entornos de desarrollo ya que he visto miles de tecnología y he buscado la que mejor se adaptaba a la hora de realizar mi proyecto además que he realizado diagramas de casos de uso que aprendí a hacerlos en esa asignatura y diagramas de clase, algo que había olvidado pero que ahora sin duda alguna comprendo mejor que antes a parte que comprendo mejor su utilidad en un proyecto de informática, pues es el esqueleto de la aplicación, es el esquema inicial en el que basaremos la aplicación.

Para finalizar he visto la importancia y las diferencias con una base de datos noSQL, las cuales actualmente había estado evitando, pero que sin embargo he descubierto que son muy útiles a la hora de realizar un proyecto ya que van creciendo según la necesidad, Firebase que era una tecnología que evitaba porque me parecía complicada al final me ha resultado relativamente sencilla de manejar y Kotlin un lenguaje que siempre he querido aprender al fin lo he podido aprender y comprender lo cual me puede abrir más puertas a lo largo de mi vida laboral.

En conclusión, saco que el trabajo a pesar de ser frustrante algunas veces me ha resultado realmente útil por todo lo que he aprendido que se que me ayudara de aquí en adelante, ya que entenderé mejor los diagramas que me pasen o entenderé mejor porque usamos una tecnología en vez de otra y esos detalles que no me habría dado cuenta antes de realizar este proyecto.

9. Bibliografía y recursos utilizados

<https://www.xatakandroid.com/aplicaciones-android/como-crear-publicar-app-google-play>

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

https://es.wikipedia.org/wiki/Despliegue_de_software#Actividades_de_despliegue

<http://www.pmoinformatica.com/2016/01/elaborar-plan-pruebas-software.html>

<https://manuel.cillero.es/doc/metodologia/metrica-3/procesos-principales/asi/actividad-10/#:~:text=El%20plan%20de%20pruebas%20es,de%20las%20actividades%20de%20prueba.>

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/publicar-una-app-en-la-apple-store-crear-una-app/>

<https://blog.mailrelay.com/es/2018/09/06/estudio-de-mercado>

<https://www.infoautonomos.com/plan-de-negocio/analisis-dafo/>

<https://www.unir.net/ingenieria/revista/riesgos-laborales-informatica/>

https://dis.um.es/~barzana/Informatica/IAGP/IAGP_riesgos.html#:~:text=Los%20riesgos%20del%20proyecto%20identifican,software%20que%20hay%20que%20producir.

https://es.wikipedia.org/wiki/Requisito_funcional

https://es.wikipedia.org/wiki/Requisito_no_funcional

<https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/>

<https://blog.bi-geek.com/introduccion-a-kotlin/>

<https://www.synergyweb.es/blog/laravel-desarrollo-medida/>

<https://slashmobility.com/blog/2019/06/pros-y-contras-de-flutter/#:~:text=Desventajas%20de%20Flutter,-Dart%20necesario%3A%20para&text=Est%C3%A1%20enfocado%20solo%20a%20m%C3%B3vil,solo%20est%C3%A1%20enfocada%20para%20m%C3%B3vil.&text=Librer%C3%ADas%20limitadas%3A%20las%20bibliotecas%20a,est%C3%A1n%20muy%20limitadas%20en%20Flutter>

<https://clouddistrict.com/que-es-react-native/>

<https://es.wikipedia.org/wiki/Firebase>

<https://es.quora.com/Cu%C3%A1les-son-las-ventajas-de-Spring-Boot-sobre-Node-js-para-un-servicio-web-RESTful>

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

<https://stackoverflow.com/>