

“ SommelAI ”

**CONVOCATORIA:** 26/06/2024

**ALUMNO/A:** Luciano Iván Blanco

**TUTOR/A:** José Antonio Lara Sánchez

**CICLO:** Desarrollo de aplicaciones multiplataforma

**INDICE**

1. [ABSTRACT](#_bookmark0) 3
2. [INTRODUCCIÓN](#_bookmark1) 5
   1. [Justificación y contexto](#_bookmark2) 5
   2. [Planteamiento del problema](#_bookmark3) 5
   3. [Objetivos del trabajo](#_bookmark4) 6
3. [MOTIVACION](#_bookmark5) 7
   1. [Marco teórico del trabajo](#_bookmark6) 8
4. [TECNOLOGÍAS EMPLEADAS](#_bookmark8) 11
   1. [Software necesario](#_bookmark9) 11
5. [DESARROLLO DEL TRABAJO](#_bookmark11) 15
   1. Estructura del proyecto 15
   2. Gestión de base de datos 17
   3. Gestión de usuarios 18
   4. Adaptación a otros idiomas 19
   5. Funcionalidades de la aplicación 20

5.5.1 Gestión de usuarios 20

5.5.1.1 Inicio y cierre de sesión 20

5.5.1.2 Registro de usuario 21

5.5.2 Gestión de la aplicación 21

5.5.2.1 Mi cuenta 21

5.5.2.2 Editar perfil 22

5.5.2.3 Inicio (Búsqueda) 22

5.5.2.3.1 Búsqueda por tipo/país 23

5.5.2.3.2 Búsqueda por texto – Detalles del vino 23

5.5.2.4 Escaneo de vinos 24

5.5.2.5 Historial de vinos escaneados – Detalles del vino 24

55.2.6 Chatbot con la IA 25

1. [CONCLUSIONES](#_bookmark12) 27
2. [REFERENCIAS 2](#_bookmark13)9
   1. [Índice de citas e índice de imágenes 2](#_bookmark16)9
3. [ANEXOS 2](#_bookmark17)9

# ABSTRACT

El presente Proyecto de Fin de Curso (PFC) para el ciclo de Desarrollo de Aplicaciones Multiplataforma (DAM) se centra en el diseño y desarrollo de “SommelAI”, una aplicación móvil para dispositivos Android destinada a entusiastas del vino y usuarios interesados en ampliar su conocimiento sobre vinos y sus maridajes.

La aplicación permite a los usuarios escanear etiquetas de vinos mediante la cámara de sus dispositivos móviles. A través de una IA, obtenemos la información de la imagen, la clasificamos y proporcionamos instantáneamente información detallada sobre el vino escaneado, incluyendo sus características, origen, notas de cata y recomendaciones personalizadas de alimentos para maridar.

Este proyecto aborda la problemática de la accesibilidad y disponibilidad de información específica y confiable sobre vinos para consumidores y aficionados, facilitando una herramienta educativa y de consulta rápida que enriquece la experiencia culinaria y vinícola del usuario, haciendo uso de tecnologías de última generación. También cuenta con un asistente personalizado basado en IA, al cual se le pueden hacer diversas consultas relacionadas con vinos mediante un “chatbot” que incorpora la aplicación.

La investigación subyacente examina las tecnologías actuales en el reconocimiento de imágenes y procesamiento de datos para implementar una solución eficaz que integra una API de información sobre vinos, ofreciendo una interfaz de usuario intuitiva y una experiencia de usuario fluida. A través del desarrollo de “SommelAI”, este trabajo demuestra la viabilidad de aplicar tecnologías avanzadas en el ámbito de la enología digital, promoviendo un nuevo enfoque en la interacción entre consumidores y el mundo del vino.

La conclusión principal del proyecto resalta el potencial de las aplicaciones móviles especializadas en transformar cómo los consumidores descubren, aprenden y se relacionan con los vinos, proponiendo un modelo para futuras innovaciones en la industria vinícola y la tecnología de la información.

# INTRODUCCIÓN

En esta primera parte se detallan los aspectos más generales del proyecto, de forma que permita al lector comprender la motivación que existe detrás de este trabajo. Por ello, comprende la definición de los requisitos que este trabajo abarca, las necesidades que se pretenden cubrir y la planificación temporal del proyecto.

## Justificación y contexto

En la actualidad, la digitalización ha transformado numerosos aspectos de nuestras vidas, incluyendo cómo consumimos información y tomamos decisiones de compra. Sin embargo, a pesar de la abundancia de información disponible en línea, los consumidores y aficionados al vino a menudo enfrentan dificultades para entender y seleccionar vinos, lo que limita su capacidad para experimentar y disfrutar plenamente de la riqueza y diversidad del mundo vinícola. “SommelAI” surge como una solución a esta problemática, ofreciendo una herramienta interactiva y accesible que permite a los usuarios obtener información confiable y específica sobre vinos de manera instantánea.

## Planteamiento del problema

El objetivo principal de este proyecto es desarrollar una aplicación móvil que utilice tecnologías avanzadas de reconocimiento de imágenes e inteligencia artificial para proporcionar información detallada y personalizada sobre vinos a partir del escaneo de etiquetas. Los objetivos específicos incluyen:

* Desarrollar una interfaz de usuario intuitiva y atractiva que permita a los usuarios interactuar fácilmente con la aplicación y acceder a la información de manera eficiente.
* Implementar algoritmos de reconocimiento de imágenes que permitan identificar automáticamente los vinos a partir de las etiquetas escaneadas.
* Integrar una API externa de información sobre vinos para recuperar datos precisos y actualizados sobre los vinos escaneados.
* Proporcionar recomendaciones personalizadas de maridaje basadas en el historial y las preferencias del usuario.
* Garantizar una experiencia de usuario fluida y educativa, promoviendo una cultura de apreciación y conocimiento vinícola.
* Brindar al usuario un “Sommelier” virtual personalizado basado en una IA para que pueda abordar y resolver mediante un “chatbot” cualquier duda que tenga respecto a este campo.

A través del desarrollo de “SommelAI”, este proyecto busca demostrar la viabilidad de aplicar tecnologías avanzadas en el ámbito de la enología digital, promoviendo un nuevo enfoque en la interacción entre consumidores y el mundo del vino. La aplicación no solo facilitará el acceso a información detallada y confiable sobre vinos, sino que también enriquecerá la experiencia de consumo, permitiendo a los usuarios descubrir, aprender y disfrutar del vino de manera más informada y placentera.

## Objetivos del trabajo

Una vez planteado el problema, se detallarán los objetivos que se pretenden alcanzar en el trabajo de investigación.

# MOTIVACION

La elección de desarrollar “SommelAI”, una aplicación Android para el escaneo de etiquetas de vinos y códigos de barras con recomendaciones de maridaje, como mi Proyecto de Fin de Curso (PFC) para el ciclo de Desarrollo de Aplicaciones Multiplataforma (DAM), surge de una confluencia de intereses personales y la identificación de una necesidad no satisfecha en el mercado. Mi fascinación por la tecnología y el deseo de aplicar mis habilidades en un proyecto que combine innovación técnica con una pasión por la enología y la gastronomía han sido determinantes en esta elección.

La idea de SommelAI nace de observar cómo, en numerosas ocasiones, consumidores y aficionados al vino enfrentan dificultades al tratar de entender y seleccionar vinos, limitando su capacidad para experimentar y disfrutar plenamente de la riqueza y diversidad que ofrece el mundo vinícola. La motivación detrás de este proyecto también se fundamenta en la creciente tendencia de digitalización en todos los aspectos de nuestras vidas, incluyendo cómo consumimos información y tomamos decisiones de compra. A pesar de la abundancia de información disponible en línea, existe una falta de herramientas interactivas y accesibles que permitan a los usuarios obtener información confiable y específica sobre vinos de manera instantánea.

La posibilidad de contribuir a la solución de este problema mediante el desarrollo de una aplicación práctica, educativa y de fácil uso, capaz de mejorar significativamente la experiencia de elección y disfrute del vino, representa una oportunidad apasionante

y retadora. Este proyecto no solo me permite aplicar y ampliar mis conocimientos técnicos en áreas como el desarrollo de aplicaciones móviles, la programación en Kotlin o Java, y la integración con APIs y tecnologías de reconocimiento de imágenes, sino que también me brinda la posibilidad de explorar el interesante cruce entre tecnología, cultura vinícola y gastronomía.

Además, el uso de inteligencia artificial (IA) en la aplicación para analizar y clasificar las etiquetas de vino, así como para proporcionar recomendaciones personalizadas de maridaje y un asistente virtual, añade una capa de innovación que potencia la utilidad y efectividad de la herramienta. La visión de empoderar a los usuarios a través de una herramienta que desmitifique el proceso de selección de vinos y promueva una cultura de apreciación y conocimiento vinícola es lo que me motiva profundamente a llevar adelante este proyecto.

## Marco teórico del trabajo

**Arquitectura y Diseño**

“SommelAI” se estructura en torno a una arquitectura de tres capas: la capa de presentación (UI/UX), la capa de lógica de negocio y la capa de datos. Esta estructura facilita la gestión de la interacción del usuario, el procesamiento de datos y el almacenamiento de información desde una base de datos y una API externa de información sobre vinos.

1. **Capa de Presentación (UI/UX):**

Diseñada para ser intuitiva y atractiva, permite a los usuarios interactuar fácilmente con la aplicación, escanear etiquetas de vinos y navegar a través de la información y recomendaciones proporcionadas.

1. **Capa de Lógica de Negocio:**

Aquí se implementan los algoritmos de reconocimiento de imágenes, la integración de la IA y también se procesan las consultas a la API de información sobre vinos. Esta capa analiza los datos escaneados y recupera información relevante para el usuario.

1. **Capa de Datos:**

Gestiona la comunicación con la base de datos interna de la aplicación y la API externa, almacenando historiales de búsqueda, preferencias del usuario y caché de información para mejorar la eficiencia y personalización de la experiencia del usuario.

**Funcionalidades Principales y Flujos de Usuario**

1. **Inicio y Registro:** Al abrir la aplicación por primera vez, los usuarios pueden registrarse o iniciar sesión para personalizar su experiencia, guardar sus historiales y preferencias.
2. **Pantalla Principal:** Después del inicio de sesión, se presenta la pantalla principal con opciones para escanear un vino, acceder a los vinos almacenados en la base de datos mediante búsqueda por texto o aplicando diferentes tipos de filtro, acceder al historial de vinos escaneados, utilizar el “chatbot” y ajustar configuraciones de perfil.
3. **Escaneo de etiquetas:** Utilizando la cámara del dispositivo, el usuario puede escanear la etiqueta de un vino. La aplicación procesa esta imagen, extrayendo el texto de la misma e identificando el vino mediante IA y recuperando información de su base de datos o mediante una API externa.
4. **Detalle del Vino:** Tras el escaneo, se muestra una pantalla detallada del vino con información sobre su origen, variedad de uva, notas de cata, y recomendaciones de maridaje. Además, los usuarios pueden leer y dejar valoraciones y comentarios.
5. **Gestión del Perfil:** Los usuarios pueden gestionar y modificar datos de su cuenta.
6. **Vinos favoritos guardados:** Una sección dedicada a almacenar los vinos favoritos de los usuarios, donde podrán calificarlos y guardarlos para tenerlos siempre a mano.

# TECNOLOGÍAS EMPLEADAS

Una vez definidos tanto los objetivos del sistema como todo el conjunto de especificaciones de diseño, tanto centrado en el usuario como en técnico y de arquitectura, se realiza la implementación de la aplicación propiamente dicha. Para ello, se toman como base normas básicas de diseño como es asegurar el máximo desacoplamiento y reutilización de componentes, así como la utilización de estándares tanto de desarrollo como en cuanto a las herramientas y APIs y herramientas externas utilizadas. Además, también se ha tratado de orientar el desarrollo a la implementación de tareas asíncronas en lo posible, para no empeorar la experiencia de usuario. A continuación, se detallan tanto las herramientas utilizadas como los aspectos más técnicos relativos al desarrollo de la aplicación y sus funcionalidades implementadas.

**Tecnologías y Herramientas Empleadas**

* **Reconocimiento de Imágenes y búsqueda con IA:** Uso de bibliotecas avanzadas de procesamiento de imágenes para el escaneo eficiente de etiquetas.
* **API de Información sobre Vinos:** Integración con una API externa que proporciona acceso a una amplia base de datos de vinos, notas de cata, y recomendaciones de maridajes.
* **Bases de Datos y Almacenamiento en la Nube:** Para almacenar perfiles de usuarios, historiales de escaneo, y caché de información para agilizar el acceso y mejorar la personalización.

## Software necesario

* **Android SDK:** es el kit de desarrollo oficial de Android y su descarga viene integrada con la del IDE Android Studio. A la hora de elegir la versión del SDK para que la aplicación fuera compatible, se ha optado por establecer como versión mínima la versión 24 (Android Nougat 7.0) y la versión de compilación en la versión 34 (Android 14). Con esto, tal como se puede observar en la siguiente imagen (Xakata, 2024), se asegura una cuota de dispositivos Android de alrededor del 95%:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Figura 1 Distribución de la cuota de dispositivos Android.**

* **Android Studio:** se trata del IDE oficial de desarrollo para Android. Se caracteriza por su facilidad de uso y por la gran cantidad de documentación que hay disponible en relación a su utilización. Además, como ya se ha comentado, permite la descarga de diferentes versiones del SDK de Android para realizar el desarrollo.
* **Kotlin**: es el lenguaje elegido para el desarrollo. Kotlin es un lenguaje de programación de código abierto, estáticamente tipado, que admite tanto la programación orientada a objetos como la funcional. Desarrollado por JetBrains y gestionado por la Kotlin Foundation, Kotlin está diseñado para ser interoperable con Java, lo que permite a los desarrolladores utilizar bibliotecas de Java y Kotlin en el mismo proyecto. Es el lenguaje preferido por Google para el desarrollo de aplicaciones Android, ofreciendo una sintaxis más concisa y características avanzadas que mejoran la productividad y la seguridad del código.
* **GitHub**: Es una plataforma de desarrollo colaborativo de software basada en la nube que permite a los desarrolladores alojar y revisar código, gestionar proyectos y construir software junto con otros desarrolladores. GitHub proporciona control de versiones mediante Git, lo que facilita la colaboración y el seguimiento de cambios en el código. Además, ofrece herramientas para la integración continua, la gestión de incidencias y la documentación, lo que mejora la eficiencia y la organización del desarrollo de software.
* **Gemini**: Es una API desarrollada por Google DeepMind que permite el acceso a modelos multimodales capaces de razonar a través de texto, imágenes, código y audio. La API de Gemini es utilizada para desarrollar aplicaciones que requieren capacidades avanzadas de procesamiento y análisis de datos. Proporciona una serie de herramientas y SDKs oficiales en varios lenguajes de programación, lo que facilita su integración en diferentes proyectos. Gemini es especialmente útil para tareas que requieren inteligencia artificial avanzada, como el reconocimiento de imágenes y el procesamiento de lenguaje natural.

**Librerías empleadas:**

* **Firebase**: tal como se ha comentado, se utiliza esta herramienta para proporcionar funcionalidades a la aplicación y disminuir el tiempo y recursos invertidos de otro modo en su desarrollo. Es por ello que se utilizan Firebase Real Database para la gestión de la base de datos y Firebase Authentication para la creación de usuarios.
* **ML Kit:** Utilizado para el reconocimiento de texto en las etiquetas de vino, permitiendo a la aplicación extraer y procesar información relevante de manera eficiente.
* **Glide**: Utilizado para el manejo y carga eficiente de imágenes dentro de la aplicación, mejorando la experiencia del usuario al mostrar imágenes de alta calidad sin afectar el rendimiento.
* **Retrofit**: Utilizado para realizar solicitudes HTTP y manejar la comunicación con APIs externas, facilitando la integración de servicios web en la aplicación.
* **TensorFlow**: Utilizado para implementar modelos de aprendizaje automático dentro de la aplicación, permitiendo el procesamiento avanzado de datos y la mejora de funcionalidades basadas en IA.
* **Gemini**: Utilizado para acceder a modelos multimodales avanzados que permiten el razonamiento a través de texto e imágenes, mejorando las capacidades de procesamiento y análisis de datos de la aplicación. Es la IA utilizada para el reconocimiento de imágenes y la búsqueda y obtención de datos relacionados con el vino, así como la IA que está detrás del “chatbot”.
* **Material Design:** es un sistema de diseño open source que permite a los desarrolladores implementar soluciones digitales de gran calidad visual.

Proporciona tanto recomendaciones de desarrollo como estándares a seguir y componentes listos para usar. En este caso, se han utilizado los componentes que proporciona para formularios (botones, cuadros de texto), para las barras de navegación y los estilos que permite que sean aplicados a otro tipo de componentes. También se ha utilizado para obtener alguno de los iconos mostrados en la aplicación.

# DESARROLLO DEL TRABAJO

En este apartado y sus subsecciones se detallan los aspectos más relevantes relacionados con el desarrollo de la aplicación y que van desde explicar la estructura elegida para la disposición interna del código del proyecto como a especificar el funcionamiento de diferentes bloques funcionales pasando por el detalle de los componentes principales entre los usados para construir la interfaz gráfica de acuerdo con el prototipo realizado en fases anteriores.

## Estructura del proyecto

Para el desarrollo de la aplicación se ha utilizado la estructura por defecto que provee Android Studio en la creación de un proyecto Android y se ha personalizado para el uso concreto de esta aplicación una estructura más clara mediante el uso de paquetes más específicos, siguiendo tanto el árbol de navegación definido como los casos de uso del sistema.

Siguiendo el patrón MVVM que se seleccionó como el más adecuado en este caso, se ha estructurado el proyecto de la siguiente forma:

|  |  |
| --- | --- |
| **Figura 2 Estructura de la aplicación 1.**    **Figura 3: Estructura de la aplicación 2.**    **Figura 4: Estructura de la aplicación 3.** | * **/manifests:** es el directorio que contiene el xml que define los activities que contiene la aplicación, cuál actúa de principal, los servicios que utilizan y, además, los permisos y features que el sistema necesita. * **/data:** Es el directorio que contiene todos los ficheros que interactúan con los servicios (Firebase, Network, DataStore). * **/firebase:** Es el directorio donde manejamos los servicios de Firebase. * **/network:** Directorio donde manejamos las peticiones http y lo relacionado a las mismas. * **/storage:** Directorio donde manejamos el almacenamiento en local usando DataStore * **/ui:** Es el directorio que contiene todos los ficheros relacionados con la lógica, este se divide en subcarpetas acorde a la estructura del menú de la app. * **/adapters:** aquí se ubican las clases que interactúan entre las vistas y los modelos para utilizar los datos que estos últimos contienen convertidos a un formato que las vistas puedan utilizar para mostrarlos. * **/drawable:** aparecen todos aquellos archivos utilizados en las diferentes plantillas visuales como pueden ser los gradientes, selectores, iconos en formato vector, etc. * **/font:** aquí se localiza las fuentes personalizadas utilizada en los estilos del sistema. * **/layout:** en este directorio se ubican todos los xml que conforman las vistas de la aplicación desde las activities a los fragments, pasando por los layouts propios de la toolbar, las cards o las filas de las diferentes listas que se muestran en las diferentes secciones. * **/menu:** es el espacio dedicado para los layouts que definen la estructura del menú lateral y de la barra inferior que se visualizan en el sistema. * **/mipmap:** dentro de este conjunto de carpetas se encuentran los recursos gráficos necesarios para la aplicación ordenados por sus dimensiones dependiendo del tamaño del dispositivo utilizado. * **/navigation:** dentro están los xml relacionados con la navegación de nuestra app. * **/values:** aquí se recogen todos aquellos xml que definen parámetros generales de la aplicación como son los literales, los estilos o los colores |

## Gestión de base de datos

## Como se ha comentado anteriormente, para la gestión de la base de datos se ha decidido utilizar Firebase Realtime Database. El primer paso para la utilización de esta funcionalidad es la creación del proyecto de la aplicación en Firebase, la selección de un nombre de base de datos, la configuración de reglas de acceso y la descarga de la información identificativa. Una vez incorporado este fichero al proyecto y añadido el módulo a utilizar en Gradle, ya se puede comenzar a utilizar este gestor de base de datos.

## Firebase Realtime Database genera y gestiona bases de datos de tipo NoSQL y esto hace que haya parte de la información que no se pueda trasladar a un diagrama de base de datos jerárquico, al menos de un modo realmente representativo. Por eso en el apartado de Diseño de la arquitectura se resumió esta representación a las principales tablas. A continuación, se aporta un ejemplo del árbol que resulta de la configuración de un usuario, el escaneo de un producto y la adición de este producto a la lista de favoritos

## Interfaz de usuario gráfica, Texto, Aplicación Descripción generada automáticamente

**Figura 5: Estructura de la BBDD 1.**

## Interfaz de usuario gráfica, Texto, Aplicación Descripción generada automáticamente

**Figura 6: Estructura de la BBDD 2.**

## Gestión de usuarios

## Al igual que con el módulo anterior, para realizar la gestión de usuarios con Firebase Authentication es suficiente con la creación del proyecto y la utilización de los snippets de código generados a tal efecto y que se pueden encontrar en su documentación.

## Captura de pantalla de un celular Descripción generada automáticamenteEn este caso, la autenticación de los usuarios se ha decidido que se realice mediante el método de email y contraseña, sin realizar ningún tipo de comprobación de si el correo es real o no, porque tampoco es algo que sea objeto de la funcionalidad de este sistema, ya que no se realizan comunicaciones por correo electrónico en ningún caso.

**Figura 7: Pestaña de “Authentication” de Firebase.**

## Tampoco es necesaria para esta aplicación la utilización de roles de usuario puesto que, tal como se definió durante la fase de establecimiento de objetivos y requerimientos, todos los usuarios tendrán disponibles las mismas funcionalidades y serán de la misma categoría.

## Además, esta gestión de usuarios se utiliza de forma muy relacionada con la gestión de base de datos anteriormente comentada ya que las reglas de acceso que se definen en la base de datos dependen en gran medida de que el usuario este autenticado con Firebase para poder leer y escribir los datos que correspondan. En caso de que el usuario no se autentique o pierda la sesión, no tendrá acceso de escritura a la información de la base de datos.

## Firebase también proporciona una consola para poder administrar los usuarios creados y poder actualizarlos, borrarlos o cambiar los proveedores de autenticación que se utilizan. De igual modo, se ha añadido un mecanismo en la aplicación para poder borrar la cuenta del usuario de forma completa, eliminando también la información relativa a ese usuario de la base de datos

## Interfaz de usuario gráfica, Aplicación Descripción generada automáticamente

**Figura 8: Pestaña de “Authentication” de Firebase 2.**

## Adaptación a otros idiomas

## Como primera aproximación a un sistema multi-idioma, se ha implementado en esta aplicación la posibilidad de ver su contenido tanto en español como en inglés por medio de los ficheros string.xml para cada uno de los idiomas. La aplicación se muestra con el idioma por defecto en el que el usuario tenga configurado su dispositivo.

## Funcionalidades de la aplicación

## En este apartado se detalla cada una de las ventanas (activity o fragment) de la aplicación desde el punto de vista técnico comentando los datos más relevantes de su implementación. Para mostrar las pantallas en esta sección se utilizan las capturas de pantalla realizadas con el dispositivo Samsung Galaxy A52 con Android 14.

## Gestión de usuarios

## Inicio y cierre de sesión

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto  Descripción generada automáticamente **Figura 9: Ventana de “Login”.** | “LoginFragment”: La activity MainActivity se encarga de gestionar la autentificación y creación de usuarios. En ella se encuentra el LoginFragment el cual realiza la autenticación de un usuario, siempre que haya sido creado previamente. Los objetos TextInputEditText son los que provee Material Design y, además, se han añadido validaciones para garantizar que el texto que se introduce en el campo Email sea una dirección de correo electrónico y que la contraseña tenga seis o más caracteres.  Si todo esto se cumple, el botón Iniciar sesión se habilita y se realiza la llamada a Firebase Authentication para realizar la autenticación en el servicio con el método signInWithEmailAndPassword y también se realiza la carga en memoria de la información de base de datos del usuario. Se informa al usuario del resultado de esta llamada mediante un toast tanto en el caso exitoso como en el erróneo. Como también se ha comentado anteriormente, es en esta activity en la que se inicia en la carga en segundo plano de los catálogos de la API de los vinos para su almacenamiento en cache.  Para cerrar sesión, se debe utilizar la opción a tal efecto del menú del perfil del usuario, que tras cerrar la sesión con el método signOut de Firebase, redirige de nuevo a esta pantalla. |

## Registro de usuario

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente **Figura 10: Ventana de “Registro”.** | **“SignUpFragment”:**  En el caso del SignUpFragment, su objetivo es la creación de un nuevo usuario. Al igual que en el caso anterior, los TextInputEditText también son de Material Design y tienen asociadas validaciones similares a las anteriores: tanto el nombre de usuario como la contraseña deben ser mayores de seis caracteres, el contenido del campo email debe ser una dirección de correo electrónico y el contenido de los dos campos referentes a la contraseña debe coincidir. Si las validaciones no se cumplen, se mostrará un mensaje debajo del campo en cuestión para informar de esta situación.  Si se cumplen las validaciones, el botón "Crear usuario" se habilita para la creación del usuario en Firebase mediante el método createUserWithEmailAndPassword, y simultáneamente la creación de este en DataStore. Además, también se crea un nodo en Firebase Realtime Database para guardar la información del usuario. |

## Gestión de la aplicación

## Mi cuenta

|  |  |
| --- | --- |
| Imagen que contiene Interfaz de usuario gráfica  Descripción generada automáticamente **Figura 11: Ventana de “Perfil”.** | **“ProfileFragment”:**  A este fragment se accede utilizando la opción del navbar. Esta pantalla se corresponde con la pantalla de solo lectura detallada anteriormente para la configuración del perfil. Este fragmento, al igual que el resto, se muestra autocontenido dentro de la actividad HomeActivity para tener acceso a la barra inferior del menú.  Muestra los datos principales del perfil del usuario, como su foto de perfil, nombre de usuario y correo electrónico. En este fragmento no se permite la edición de ninguno de estos aspectos; simplemente se da acceso mediante el botón de “Editar perfil” a la pantalla de “Edición de perfil”. También se da la opción de eliminar la cuenta, al pulsar el botón "Borrar cuenta", se muestra un AlertDialog para que el usuario confirme si está seguro de querer borrar la cuenta. Si acepta, se elimina el usuario en Firebase con el método delete y toda la información relacionada de la base de datos, así como también todos los datos de DataStore asociados al mismo. |

## Editar perfil

|  |  |
| --- | --- |
| **Figura 12: Ventana de “EditPerfil”.** | **“EditProfileFragment”:**  En este fragmento podremos editar algunos datos del usuario, tales como el nombre de usuario, nombre y apellido, contraseña y foto de perfil.  Para modificar la foto del usuario, basta con pulsar en la imagen y se nos abrirá un dialogo de solicitud de permisos en caso de no tenerlos habilitados.  Al realizar la actualización de alguno de estos campos, la misma se aplicará tanto en Firebase como en DataStore. |

## Inicio (Búsqueda)

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente **Figura 13: Ventana de “SearchFragment”.** | “SearchFragment”: Este es el fragment al cual se llega luego de iniciar sesión exitosamente. En el mismo tenemos diferentes opciones de búsqueda.  Contamos con un campo de texto para poder buscar los vinos de nuestra API. Estos, al iniciar la app, se guardan en la memoria local para así minimizar los tiempos de carga y realizar una búsqueda instantánea. También tenemos botones para realizar la búsqueda de todos los vinos clasificándolos por tipos. Asimismo, podemos filtrar por países.  Al pulsar cualquiera de estos botones, se navega a un fragmento con una RecyclerView, la cual nos lista todos nuestros vinos con una información reducida.  En el margen inferior derecho, contamos con un Floating button, el cual al ser pulsado nos llevará al fragmento "ScannerCameraFragment", donde podremos tomar una fotografía del vino o cargar una foto del mismo almacenada previamente. |

## Búsqueda por tipo/país

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Texto  Descripción generada automáticamente **Figura 14: Ventana de “WineList”.** | “WineListFragment”: Pantalla que lista los vinos por tipo/país luego de pulsar su respectivo botón en el "SearchFragment".  Nos muestra una breve descripción de los vinos, en la cual podemos encontrar su calificación, nombre del vino, bodega, año y zona de origen.  Al pulsar en alguno de los vinos, se nos abre una pantalla con los detalles del vino proporcionados por la API y, como complemento, una breve descripción del vino proporcionada por la IA. |

## Búsqueda por texto – Detalles del vino

|  |  |
| --- | --- |
| Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  **Figura 15: Ventana de “ WineListFragment” y de “WineResponseDetail”.** | “SearchNameFragment” & “WineResponseDetailFragment”:En la primera imagen podemos observar el funcionamiento de la búsqueda por texto. La segunda imagen corresponde a la vista del ítem al ser pulsado, esta estructura es la misma tanto para la búsqueda por “tipo/país” como para la búsqueda por texto.Aquí podemos encontrar información más detallada del vino.El último campo de la vista del ítem corresponde a la descripción generada por IA. |

## Escaneo de vinos

|  |  |
| --- | --- |
| **Figura 16: Ventana de “ScannerCameraFragment”.** | “ScannerCameraFragment”:El acceso a este fragmento es mediante el Floating button del "SearchFragment".Este fragmento contiene una SurfaceView para mostrar el enfoque de la cámara. Podemos escanear el vino deseado enfocándolo y pulsando el botón de la cámara. También contamos con dos botones en el margen superior: uno para activar el flash y otro para cerrar la cámara y volver al fragmento anterior. Además, tenemos la opción de abrir la galería y seleccionar una imagen capturada previamente para su posterior análisis.Al seleccionar una imagen o tomar una fotografía, se procede a analizar la imagen con una IA para primero extraer el texto y luego clasificarlo, obteniendo así solo los datos necesarios para su posterior búsqueda en la base de datos. Si tenemos una coincidencia en la búsqueda, la redirección se hace hacia el "WineListFragment", mostrando una o más coincidencias. Si el vino no está en nuestra base de datos, se redirige al usuario hacia el "FeedFragment", donde se mostrará la información extraída de la etiqueta más un complemento que nos brinda la IA.La navegación hacia cualquiera de estos fragmentos se hace a través de LiveData. |

## Historial de vinos escaneados – Detalles del vino

|  |  |
| --- | --- |
| **Figura 17: Ventana de “FeedFragment” y “WineDetailFragment”.** | **“FeedFragment” & “WineDetailFragment”** El **FeedFragment** es el fragmento al cual navegamos desde el "ScannerCameraFragment" en caso de no encontrar el vino escaneado en nuestra base de datos. Este fragmento muestra el historial de vinos escaneados por el usuario, el cual se almacena en Firebase.  Aquí tenemos una información reducida de cada vino, pero si pulsamos en uno de ellos podremos ver su detalle, así como una descripción proporcionada por IA. También tenemos la opción de calificarlo y eliminarlo de nuestro historial. |

## Chatbot con la IA

|  |  |
| --- | --- |
| **Figura 18: Ventana de “Chatbot 1”.**  **Figura 19: Ventana de “Chatbot 2”.** | **“SommelFragment”** A este fragmento accedemos desde el navbar.  Es un “asistente virtual” que tenemos, el cual en realidad es una IA a la cual se le establece como parámetro que sea un profesional de los vinos para resolver todas nuestras dudas. Este asistente virtual está diseñado para proporcionar recomendaciones personalizadas, responder preguntas sobre maridajes, sugerir vinos basados en preferencias personales y ofrecer información detallada sobre diferentes tipos de vinos.  De aquí nace el concepto de "Sommel-AI", integrando las características de un "Sommelier" con las capacidades y posibilidades que nos brinda una IA. Además, puede acceder a una vasta base de datos de vinos y utilizar algoritmos avanzados para analizar y comparar diferentes opciones, asegurando que las sugerencias sean precisas y relevantes.  La integración de la IA permite una experiencia interactiva y educativa, haciendo que el proceso de selección y disfrute del vino sea más accesible y agradable para todos los usuarios. |

# CONCLUSIONES

"SommelAI" ha sido un paso significativo hacia la innovación en el mundo del vino, proporcionando a los aficionados una herramienta interactiva y educativa para profundizar en su conocimiento sobre vinos y sus posibles maridajes. Utilizando tecnologías de punta como el reconocimiento de imágenes y la inteligencia artificial, "SommelAI" ha revolucionado la forma en que los consumidores acceden a información confiable y detallada sobre vinos, mejorando así su experiencia culinaria y enológica.

La aplicación no solo facilita el acceso a información precisa y actualizada mediante la integración de una API de información sobre vinos, sino que también personaliza la experiencia del usuario a través de un asistente virtual basado en IA. Este asistente adapta sus recomendaciones a las preferencias individuales y al historial de interacciones, lo que demuestra un avance notable en cómo la tecnología puede mejorar la interacción entre consumidores y vinos.

**Sugerencias para futuros trabajos:**

1. Ampliación de la base de datos: Sería beneficioso expandir la base de datos de vinos para incluir más variedades y regiones. Esto enriquecería las recomendaciones y aumentaría la precisión de la información proporcionada.
2. Optimización de la IA: Continuar mejorando los algoritmos de inteligencia artificial para aumentar la precisión en el reconocimiento de etiquetas y en la personalización de las recomendaciones. Esto podría incluir adaptaciones basadas en aprendizaje automático para ajustar las sugerencias a contextos específicos como eventos o estaciones del año.
3. Desarrollo multilingüe: Crear versiones de la aplicación en varios idiomas para alcanzar a un público más amplio y aumentar su accesibilidad y adopción a nivel mundial.
4. Integración con e-commerce: Habilitar la opción de comprar vinos directamente a través de la aplicación, integrando plataformas de comercio electrónico que faciliten la adquisición de los vinos recomendados.
5. Retroalimentación y estudios de usuario: Implementar un sistema de retroalimentación continua para que los usuarios puedan enviar sus impresiones y sugerencias sobre la aplicación. Realizar estudios de usuario para evaluar la usabilidad y satisfacción, permitiendo así realizar ajustes basados en datos reales para mejorar la experiencia de usuario.

# REFERENCIAS

## Índice de citas e índice de imágenes

[**Figura 1 Distribución de la cuota de dispositivos Android.** 10](#_Toc169616021)

[**Figura 2 Estructura de la aplicación 1.** 12](#_Toc169616022)

[**Figura 3: Estructura de la aplicación 2.** 13](#_Toc169616023)

[**Figura 4: Estructura de la aplicación 3.** 13](#_Toc169616024)

[**Figura 5: Estructura de la BBDD 1.** 14](#_Toc169616025)

[**Figura 6: Estructura de la BBDD 2.** 15](#_Toc169616026)

[**Figura 7: Pestaña de “Authentication” de Firebase.** 15](#_Toc169616027)

[**Figura 8: Pestaña de “Authentication” de Firebase 2.** 16](#_Toc169616028)

[**Figura 9: Ventana de “Login”.** 17](#_Toc169616029)

[**Figura 10: Ventana de “Registro”.** 18](#_Toc169616030)

[**Figura 11: Ventana de “Perfil”.** 18](#_Toc169616031)

[**Figura 12: Ventana de “EditPerfil”.** 19](#_Toc169616032)

[**Figura 13: Ventana de “SearchFragment”.** 19](#_Toc169616033)

[**Figura 14: Ventana de “WineList”.** 20](#_Toc169616034)

[**Figura 15: Ventana de “ WineListFragment” y de “WineResponseDetail”.** 20](#_Toc169616035)

[**Figura 16: Ventana de “ScannerCameraFragment”.** 21](#_Toc169616036)

[**Figura 17: Ventana de “FeedFragment” y “WineDetailFragment”.** 21](#_Toc169616037)

[**Figura 18: Ventana de “Chatbot 1”.** 22](#_Toc169616038)

[**Figura 19: Ventana de “Chatbot 2”.** 22](#_Toc169616039)

# ANEXOS

En los anexos se adjuntará todo el material citado que ocupe demasiado espacio para ser incluido en el cuerpo de texto o que pueda desviar la atención del lector.

Cualquier gráfico, estadística, línea de código, texto de referencia, entrevista o escolio técnico que no esté directamente vinculado con el desarrollo y las conclusiones del PFC pero que se quiera añadir como material de consulta es susceptible de incluirse como anexo.