

ZAYTA

— APP —

RECONOCIMIENTO Y MAPEO
DE VARIEDADES DE ACEITUNAS

POR LUIS EDUARDO ÁLVAREZ

Índice

1. Introducción.....	3
2. Antecedentes y estado del mercado	3
3. Desarrollo	4
Arquitectura de desarrollo	4
Estructura funcional del sistema	5
Estructura del software del proyecto	6
Diseño del sistema	7
Principales casos de uso	9
Diseño de la base de datos.....	15
Pruebas	17
4. Documentación.....	22
Manual de instalación y configuración.....	22
Manual de usuario.	24
5. Presupuesto	33
6. Mejoras propuestas	33
7. Conclusiones	34
8. Bibliografía y referencias	34

1. Introducción

Título del proyecto: Zayta.

Resumen: Este proyecto tiene como objetivo proveer al usuario final de una herramienta que sea capaz de identificar distintas variedades de aceitunas, a través de la cámara o galería de un dispositivo Android. Además, de almacenar los datos extraídos durante el proceso de reconocimiento para realizar un mapa en el que observar las distintas variedades reconocidas y guardadas. Por último, el usuario también podrá consultar los datos de cada reconocimiento a partir de una lista con todos los registros que haya guardado.

Como nombre del proyecto se ha elegido el origen etimológico de la palabra aceite, que significa "jugo de aceituna".

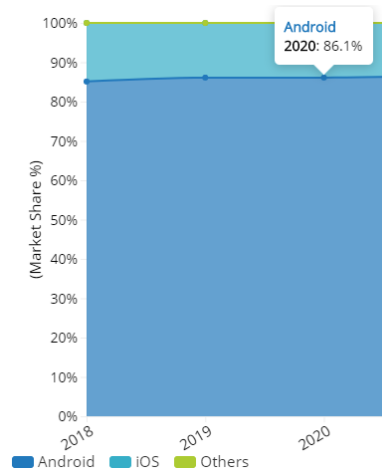
Palabras clave: Reconocimiento – Variedad – Aceituna

2. Antecedentes y estado del mercado

Con el paso de los años, el refinamiento de los procesos de calidad en un producto final ha ido creciendo exponencialmente. Además de la demanda, por parte de sanidad, de conocer la trazabilidad de ese producto. En este caso, el aceite y en consecuencia la aceituna. Es en ese punto, donde esta herramienta puede solventar un problema no resuelto hasta ahora, que es saber de qué variedades de aceitunas está compuesto el aceite que producen y como consecuencia mejorar el proceso de trazabilidad. También, el uso de esta aplicación puede satisfacer la necesidad de conocer las variedades de aceitunas a nivel particular y en un ámbito meramente informativo.

La plataforma elegida para el desarrollo y distribución de esta aplicación son los dispositivos que dispongan del sistema operativo Android, ya que la mayoría de los usuarios en el mercado disponen de ello.

<https://www.idc.com/promo/smartphone-market-share/os>



3. Desarrollo

Arquitectura de desarrollo

Set de datos:

Nota introductoria:

En principio, iba a recolectar las fotos de las distintas variedades fotografiando, dado el confinamiento por el **COVID19** en el que me hallo, he optado por recolectar las imágenes de internet. Existen relativamente pocas fotos de las distintas variedades en internet para establecer el set de datos para entrenar la red convolucional. Entonces, he optado por recoger vídeos en los que se vea de manera adecuada y recortar frame a frame.

El set de datos utilizado para etiquetar las distintas variedades se ha desarrollado con las siguientes tecnologías:



Para recortar los frames de los vídeos y extraer las imágenes de ellos, se ha utilizado la línea de comandos y [FFmpeg](#).

Se han utilizado **DOS Batch Scripts** para estructurar los ficheros de manera adecuada para su uso.

El modelo de datos resultante, que utilizará la aplicación, se ha extraído a través del entrenamiento realizado con [Machine Learning Kit for Firebase](#).

Aplicación:



El lenguaje de programación en el que se ha desarrollado la aplicación es Java 8.

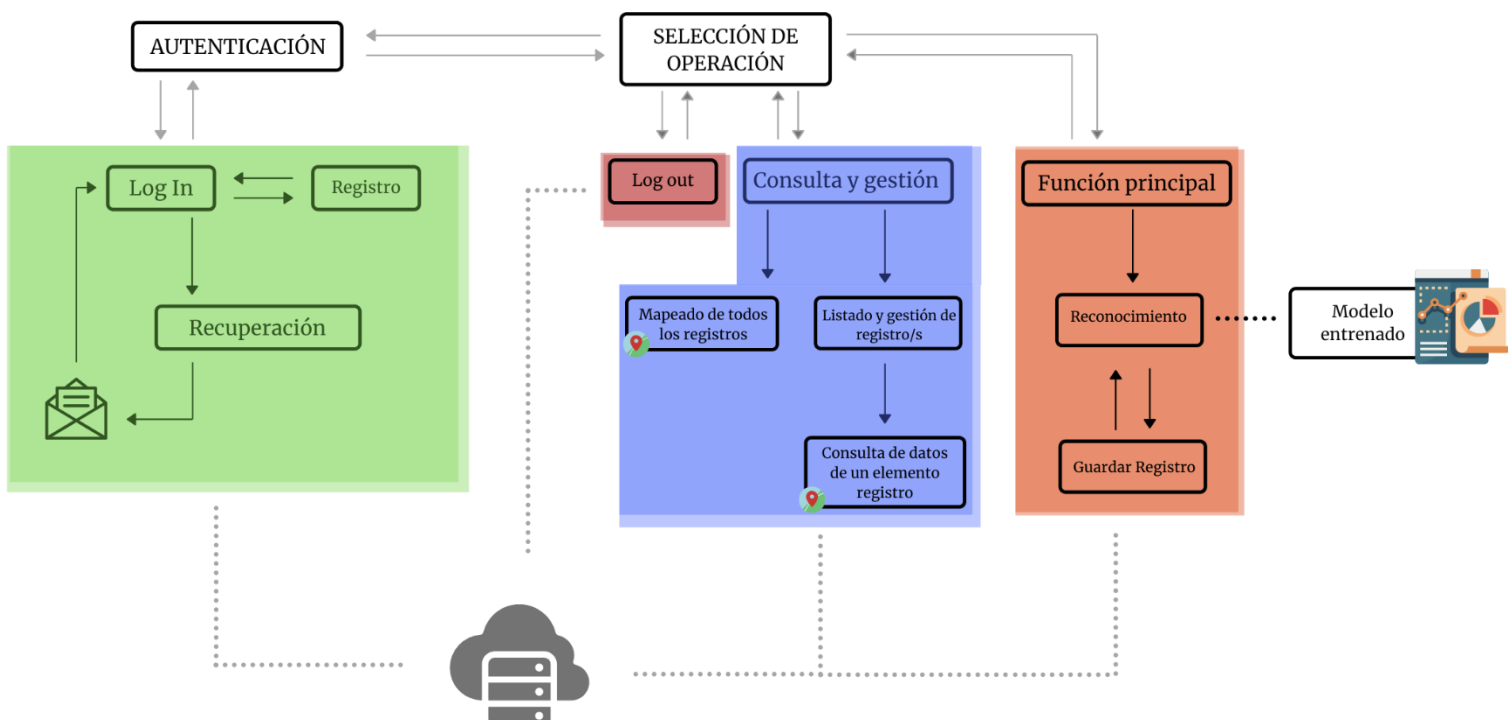
Como entorno de desarrollo integrado se ha utilizado Android Studio, ya que es de los IDEs más completos para desarrollar una aplicación **Android**.

Para realizar la autenticación y el restablecimiento de contraseñas vía email se ha utilizado Firebase Authentication.

Como base de datos para el almacenamiento, gestión y sincronización de los registros he elegido Firestore Realtime Database.

El software utilizado para el control de versiones ha sido GitKraken.

Estructura funcional del sistema



La estructura del sistema se compone de 3 módulos principales, todos ellos tienen una comunicación directa con la base de datos y están comunicados mediante un menú (Selección de operación):

Autenticación: Formado por 3 actividades encargadas de gestionar el registro, restablecimiento de contraseña y la sesión de los usuarios.

Consulta y gestión: Formado por 3 actividades cuyo objetivo es mostrar y gestionar los registros de cada usuario.

Función principal: Compuesto por una sola actividad que realiza el reconocimiento de una imagen entrante, vía cámara o galería, la etiqueta como variedad del fruto y ofrece la opción de registrarlo en la base de datos. Además, muestra una gráfica a modo de detalle.

Estructura del software del proyecto

Estructura del modelo de datos del reconocimiento. Estructura de la aplicación.

```

|—aceitunas
|   |—cornicabra
|       cornicabra1.jpg
|       cornicabra1.png
|       cornicabra10.jpg
|       ...
|   |—gordal
|       gordal1.jpg
|       gordal1.png
|       gordal10.png
|       ...
|   |—hojiblanca
|       hojiblanca1.jpg
|       hojiblanca1.png
|       hojiblanca10.jpg
|       ...
|   |—picual
|       picual1.jpg
|       picual1.png
|       picual10.jpg
|       ...

```

```

ActividadInformacionRegistro.java
ActividadMenuInicio.java
ActividadMisRegistros.java
ActividadSplashScreen.java
Reconocimiento.java

—asynctasks
    CuentaAtras.java

—fragmentos
    |—mapas
    |   FragmentMapsInformacionRegistro.java
    |   FragmentMapsTodosLosRegistros.java
    |—registros
    |   MyRegistroRecyclerViewAdapter.java
    |   OnMiRegistroInteractionListener.java
    |   RegistroFragment.java
—login
    ActividadHasOlvidadoPass.java
    ActividadLogin.java
    ActividadRegistro.java

—objetos
    Registro.java
    Usuario.java

—tools
    Tools.java

```

En la carpeta **assets** se encuentra el modelo de datos entrenado que utilizaremos en el proceso de reconocimiento.

```
└─model
    dict.txt
    manifest.json
    model.tflite
```

Diseño del sistema

Patrón de diseño.

En primer lugar, iba a realizar el proyecto con el patrón **MVVM**, pero a falta de experiencia y tiempo con ese patrón, he decidido estructurar el proyecto con el patrón de arquitectura **MVC: Modelo-Vista-Controlador**.

El **MVC** está formado por 3 componentes:

- 1. Modelo:** Este componente está compuesto por las entidades que representan el modelo de dominio de la aplicación. Es el primer componente tomado en cuenta, ya que en él se definen los atributos que utiliza la aplicación. En este caso, el modelo está ubicado en la carpeta '**objetos**' y contiene las clases **Registro** y **Usuario**.
- 2. Vista:** Información y funcionalidades que se muestran de manera gráfica. Define la estructura que saldrá en pantalla. Está formado por los ficheros .xml almacenados en la carpeta res/layout/.
- 3. Controlador:** Se encarga de realizar la comunicación entre el modelo y la vista. Tiene toda la lógica de control y parte de la lógica de visualización. En el caso de Zayta, el controlador lo forman las actividades y fragmentos.

User Interface & User Experience:

A nivel **visual**, se ha adoptado un estilo de diseño con tendencia ascendente este año, el estilo **neomórfico** (*neumorphism*). Es la combinación de la palabra Nuevo y el término Eskeumorfismo, acuñado por Steve Jobs para describir el aspecto "tradicional" que tenían sus interfaces en las primeras versiones de Iphone. Tiene como objetivo fidelizar la visualización de los elementos de la interfaz de usuario en comparación con la vida real. Está caracterizado por la acentuación de sombras, relieves y contrastes sin dejar de lado la usabilidad.

En cuanto a **usabilidad**, he tenido en cuenta los **10 principios heurísticos de Nielsen**

Formularios:

En total, existen **3 formularios** en la aplicación. Las **medidas adoptadas** para verificar la entrada de datos son las siguientes:
La regla común para realizar la comprobación de campos, denotar un fallo e informar de este mismo al usuario, en cualquiera de las 3 actividades, es pulsar el botón de acción correspondiente a la actividad en uso. Ya sea 'Log In', 'Registrarte' o 'Enviar'.
Si existe algún campo que denominemos como no válido, se notificará al usuario en un panel flotante, en la parte inferior de la pantalla, durante unos segundos.
Para que los campos de las actividades sean válidos deben cumplir ciertos requisitos, explicados a continuación:

Log In:

- **Email:** Debe tener una longitud mínima de 5 caracteres en los que deben incluirse una '@' y un '.'
- **Contraseña:** Debe no estar vacío y tener como mínimo una longitud de 8 caracteres.

Registro:

- **Email:** Es un campo en común con la actividad "Log In" y "Recuperación de contraseña", por lo tanto, debe cumplir los mismos requisitos.
- **Contraseña:** Es un campo en común con la actividad Log In, por lo tanto, debe cumplir los mismos requisitos.
- **Repetir Contraseña:** Este campo será válido cuando el campo contraseña sea correcto y contenga la misma información que este.

Recuperación de contraseña:

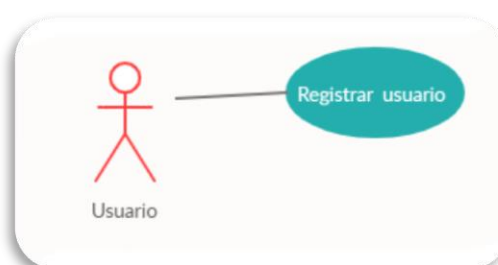
- **Email:** Es un campo en común con la actividad "Log In" y "Registro" por lo tanto, debe cumplir los mismos requisitos.

Principales casos de uso

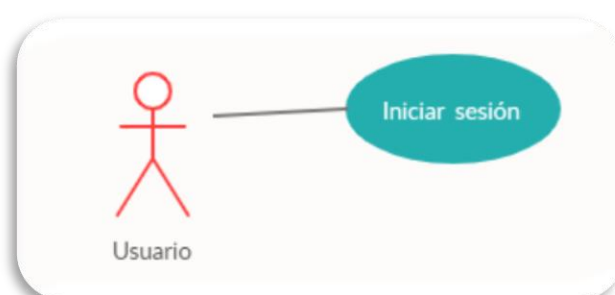
En los casos de uso se presentan 2 actores: Administrador y usuario. Cabe destacar que pueden existir varios usuarios por cada cuenta.

- **Administrador:** Encargado de gestionar el correcto funcionamiento de la aplicación con la potestad de gestionar por completo la cuenta de cualquier usuario.
- **Usuario:** Cualquier persona que utilice la aplicación, incluido el administrador.

Nombre	CU01 – Registrar usuario
Actores implicados	Usuario.
Propósito	Registrar un usuario en la aplicación.
Resumen	El usuario introduce los datos, el sistema los valida y registra un usuario.
Curso normal de los eventos	<ol style="list-style-type: none">1. El usuario introduce el mail, su contraseña y repite está en un segundo campo.2. El sistema valida los datos.3. El sistema comprueba que no esté registrado y almacena los datos del usuario.4. El sistema inicia sesión automáticamente y lanza la actividad MenúInicio.
Curso alternativo de los eventos	<p>Alternativa 1:</p> <ol style="list-style-type: none">1. El usuario introduce datos erróneos.2. El sistema detecta la invalidez de dichos datos.3. El sistema lanza un mensaje de error. <p>Alternativa 2:</p> <ol style="list-style-type: none">1. El usuario introduce datos válidos.2. El sistema valida los datos.3. El sistema comprueba que ya está registrado.4. El sistema muestra un mensaje de error.



Nombre	CU02 - Iniciar sesión
Actores implicados	Usuario.
Propósito	Iniciar la sesión de un usuario.
Resumen	El usuario introduce los datos, el sistema los valida e inicia la sesión de un usuario.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El sistema comprueba que no exista una sesión abierta. 2. El usuario introduce el mail y contraseña. 3. El sistema valida que los datos son correctos. 4. El sistema comprueba que el usuario esté registrado y que esos datos coinciden con los registros. 5. El sistema lanza la actividad MenúInicio.
Curso alternativo de los eventos	<p>Alternativa 1:</p> <ol style="list-style-type: none"> 1. El sistema comprueba que existe una sesión abierta. 2. El sistema lanza la actividad MenúInicio. <p>Alternativa 2:</p> <ol style="list-style-type: none"> 1. El usuario introduce datos no válidos. 2. El sistema detecta la invalidez de dichos datos. 3. El sistema muestra un mensaje de error. <p>Alternativa 3:</p> <ol style="list-style-type: none"> 1. El usuario introduce datos válidos. 2. El sistema valida que los datos son correctos. 3. El sistema comprueba que el usuario no está registrado. 4. El sistema muestra un mensaje de error.



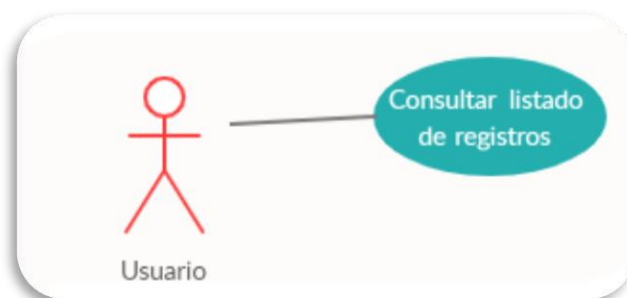
Nombre	CU03 – Reconocer una variedad.
Actores implicados	Usuario.
Propósito	Reconocer una variedad de aceituna.
Resumen	El usuario toma o rescata una foto del dispositivo y el sistema reconoce la variedad que hay en dicha foto.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El usuario selecciona una imagen. 2. El sistema procesa dicha imagen. 3. El sistema muestra las estadísticas obtenidas.
Curso alternativo de los eventos	<ol style="list-style-type: none"> 1. El usuario selecciona una imagen. 2. El usuario cancela el proceso. 3. Fin del caso de uso.



Nombre	CU04 – Guardar un registro.
Actores implicados	Usuario.
Propósito	Guardar un nuevo registro.
Resumen	El usuario después de reconocer una variedad guarda un registro.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El usuario guarda el registro. 2. El sistema procesa los datos del registro. 3. El sistema almacena los datos. 4. El sistema reinicia la actividad de reconocimiento.
Curso alternativo de los eventos	



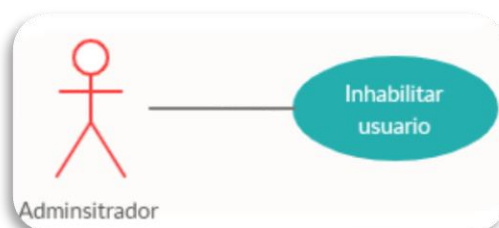
Nombre	CU05 – Consultar listado de registros
Actores implicados	Usuario.
Propósito	Consultar todos los registros guardados.
Resumen	El usuario consulta una lista donde aparecen todos los registros que ha guardado.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El usuario realiza la petición. 2. El sistema procesa la petición. El sistema muestra una lista con todos los registros guardados.
Curso alternativo de los eventos	



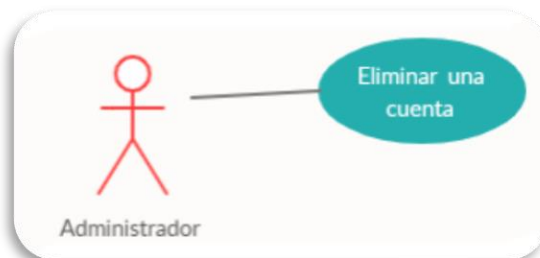
Nombre	CU06 – Restablecer contraseña.
Actores implicados	Usuario.
Propósito	Cambiar la contraseña de un usuario.
Resumen	El usuario introduce su email, el sistema valida los datos y envía un enlace para restablecer la contraseña.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El usuario introduce los datos. 2. El sistema valida los datos. 3. El sistema comprueba que el usuario esté registrado. 4. El sistema envía un mensaje con un enlace al email del usuario. 5. El usuario cambia la contraseña.
Curso alternativo de los eventos	<p>Alternativa 1:</p> <ol style="list-style-type: none"> 1. El usuario introduce datos no válidos. 2. El sistema detecta la invalidez de dichos datos. 3. El sistema muestra un mensaje de error. <p>Alternativa 2:</p> <ol style="list-style-type: none"> 1. El usuario introduce los datos. 2. El sistema valida los datos. 3. El sistema comprueba que no está registrado. 4. El sistema muestra un mensaje de error.



Nombre	CU07 – Inhabilitar usuario
Actores implicados	Administrador.
Propósito	Inhabilitar la cuenta de un usuario
Resumen	El administrador accede a la plataforma de administración e inhabilita la cuenta de un usuario.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El administrador selecciona inhabilitar la cuenta de un usuario. 2. El sistema registra la cuenta como inhabilitada.
Curso alternativo de los eventos	



Nombre	CU08 – Eliminar una cuenta
Actores implicados	Administrador.
Propósito	Borrar la cuenta de un usuario y sus registros
Resumen	El administrador accede a la plataforma de administración y elimina la cuenta de un usuario.
Curso normal de los eventos	<ol style="list-style-type: none"> 1. El administrador selecciona eliminar la cuenta de un usuario. 2. El sistema registra la cuenta como eliminada. 3. El sistema elimina los datos de dicho usuario.
Curso alternativo de los eventos	



Diseño de la base de datos

El sistema gestor de base de datos se ha implementado con 2 módulos comunicados que proporciona Firebase. Estos son:

Firestore Authentication:

Se encarga de gestionar el proceso de autenticación de usuarios, así como de almacenar y gestionar sus datos: Identificador, proveedor de correo, fecha de creación, inicio de sesión y UID de usuario.

La gestión de contraseñas es segura ya que su hash está encriptado con el algoritmo SCRYPT.



The screenshot shows the Firebase Authentication console. At the top, there is a search bar with the placeholder text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario'. To the right of the search bar is a blue button labeled 'Añadir usuario', a refresh icon, and a menu icon. Below the search bar is a table with the following columns: 'Identificador', 'Proveedores', 'Fecha de creación', 'Inicio de sesión', and 'UID de usuario ↑'. The table contains one row of data for a user with the email 'luisalvarez7696@gmail.co...'. The 'Proveedores' column shows an email icon. The 'Fecha de creación' and 'Inicio de sesión' columns both show the date '18 may. 2020'. The 'UID de usuario' column shows the long alphanumeric string 'InxQyGx1o6bpouplfjRIE74KRx1'.

Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
luisalvarez7696@gmail.co...		18 may. 2020	18 may. 2020	InxQyGx1o6bpouplfjRIE74KRx1

Firestore RealTime DataBase:

Encargado de gestionar los cambios de los datos ofrecidos por la aplicación en tiempo real.

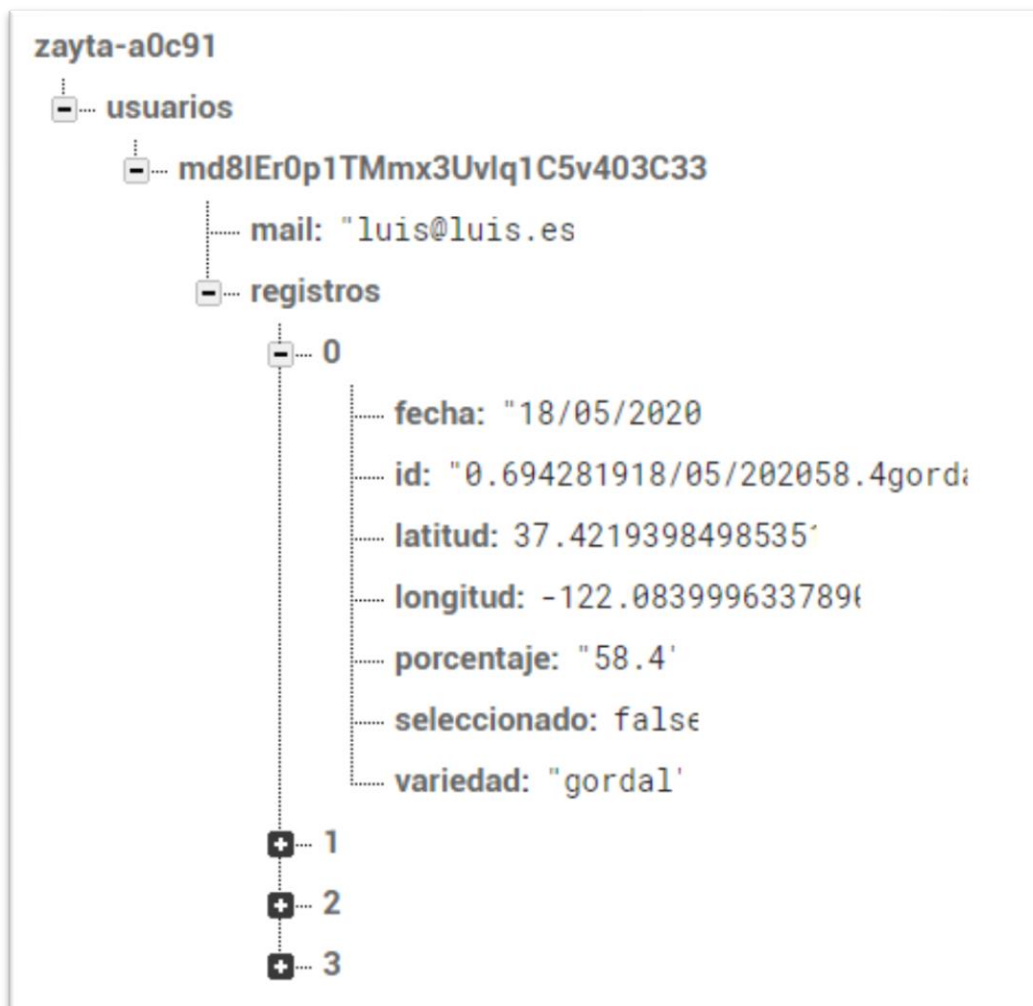
La base de datos reside en una colección NoSQL, que contiene varios objetos anidados.

Su estructura está compuesta por:

Usuarios: Contiene una lista de objetos usuario referenciados desde Firebase Authentication por su UID.

Usuario: Alberga el mail del usuario y una lista de registros.

Registro: Almacena la fecha en la que se realizó el registro, un id único, latitud, longitud, el porcentaje de confianza del reconocimiento, si está seleccionado como elemento borrado y la variedad con la que ha sido etiquetada.



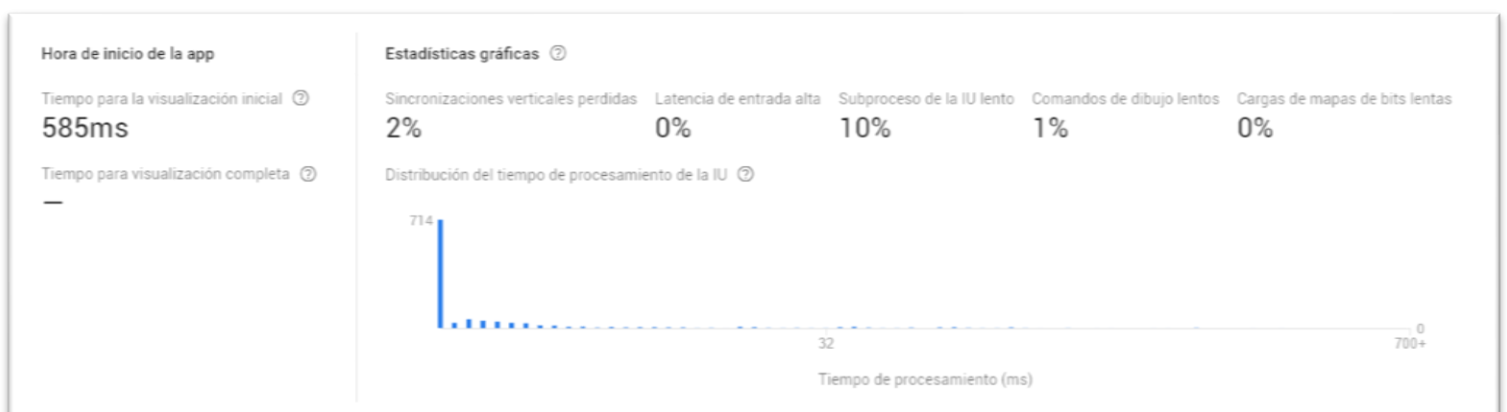
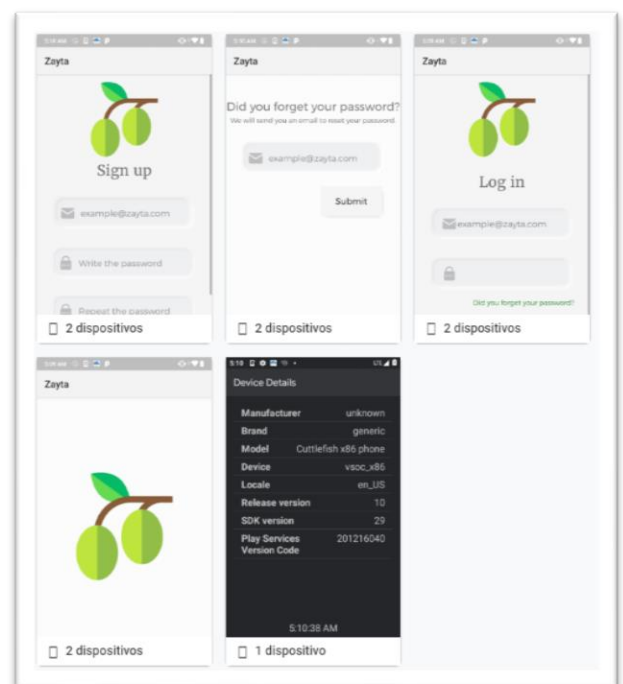
Pruebas

En cuanto a rendimiento se refiere, para sacar el mayor potencial posible a las herramientas que nos otorga Firebase, he decidido utilizar la herramienta: **Firestore Test Lab**.

Esta es una infraestructura de pruebas basada en la nube, tiene una interfaz sencilla y permite realizar pruebas con distintas configuraciones en distintos dispositivos móviles.

He realizado las pruebas con la configuración **Robo** que ofrece esta infraestructura, en una multitud de dispositivos con distintos niveles de API y resolución.

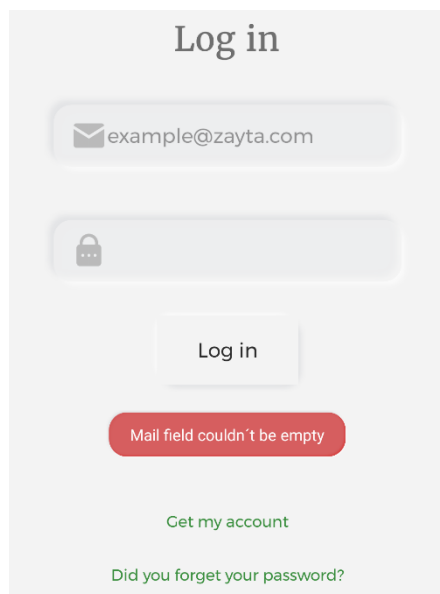
En conclusión, la respuesta por parte entre dispositivos antiguos y modernos dista bastante para la visualización inicial, pero desde una perspectiva general, la aplicación funciona de manera óptima.



Por otra parte, he realizado distintas pruebas para buscar posibles errores que cometa el usuario final:

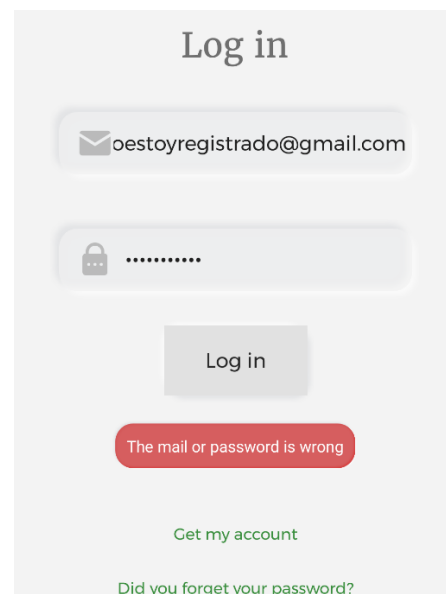
Actividad Log In:

El sistema lanza un mensaje de error si no existe conexión a internet, alguno de los campos está vacío, son incorrectos, si el mail no está registrado en la aplicación o si el mail no coincide con la contraseña.



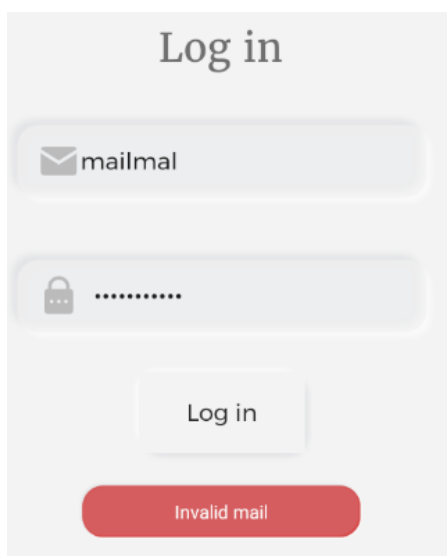
The screenshot shows a 'Log in' form with two input fields. The first field, labeled with an envelope icon, contains the email 'example@zayta.com'. The second field, labeled with a lock icon, is empty. Below the fields is a 'Log in' button. A red error message at the bottom states 'Mail field couldn't be empty'. Below the error message are two links: 'Get my account' and 'Did you forget your password?'.

Campos vacíos



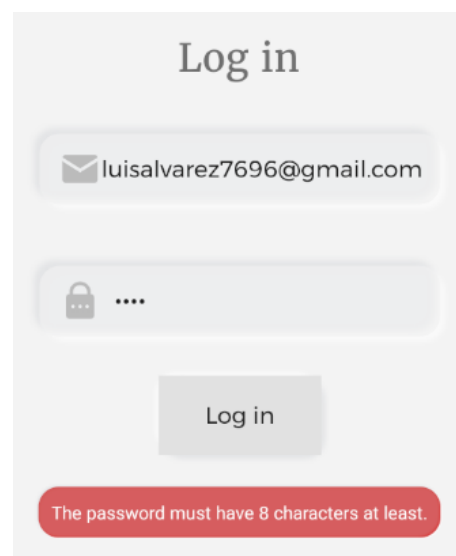
The screenshot shows a 'Log in' form with two input fields. The first field, labeled with an envelope icon, contains the email 'oestoyregistrado@gmail.com'. The second field, labeled with a lock icon, contains a password represented by dots. Below the fields is a 'Log in' button. A red error message at the bottom states 'The mail or password is wrong'. Below the error message are two links: 'Get my account' and 'Did you forget your password?'.

Insertión de un mail no registrado
Contraseña incorrecta



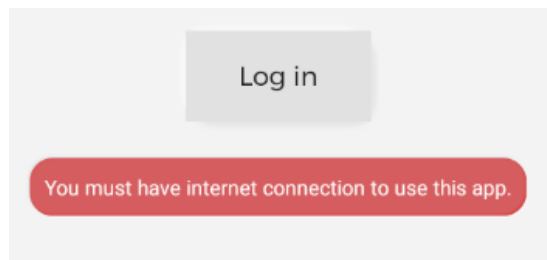
The screenshot shows a 'Log in' form with two input fields. The first field, labeled with an envelope icon, contains the email 'mailmal'. The second field, labeled with a lock icon, contains a password represented by dots. Below the fields is a 'Log in' button. A red error message at the bottom states 'Invalid mail'. Below the error message are two links: 'Get my account' and 'Did you forget your password?'.

Mail no válido



The screenshot shows a 'Log in' form with two input fields. The first field, labeled with an envelope icon, contains the email 'luisalvarez7696@gmail.com'. The second field, labeled with a lock icon, contains a password represented by dots. Below the fields is a 'Log in' button. A red error message at the bottom states 'The password must have 8 characters at least.'. Below the error message are two links: 'Get my account' and 'Did you forget your password?'.

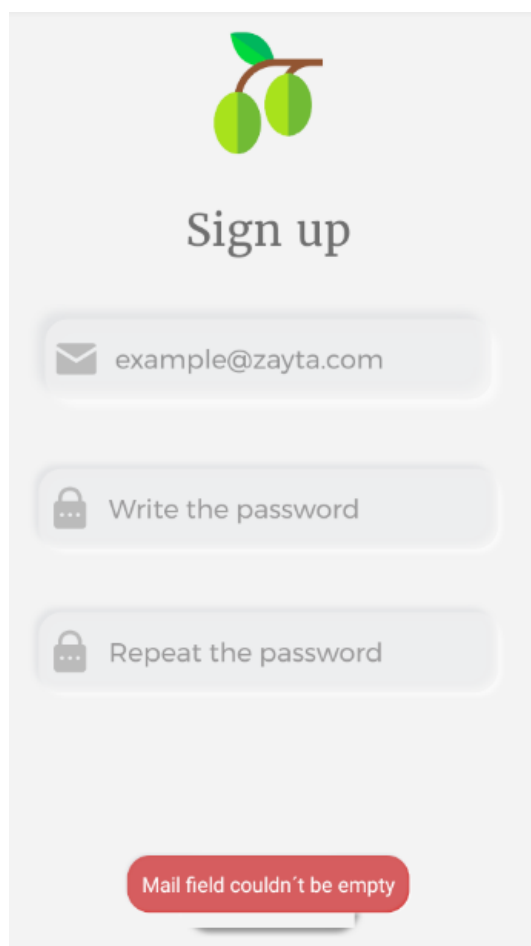
Contraseña no válida



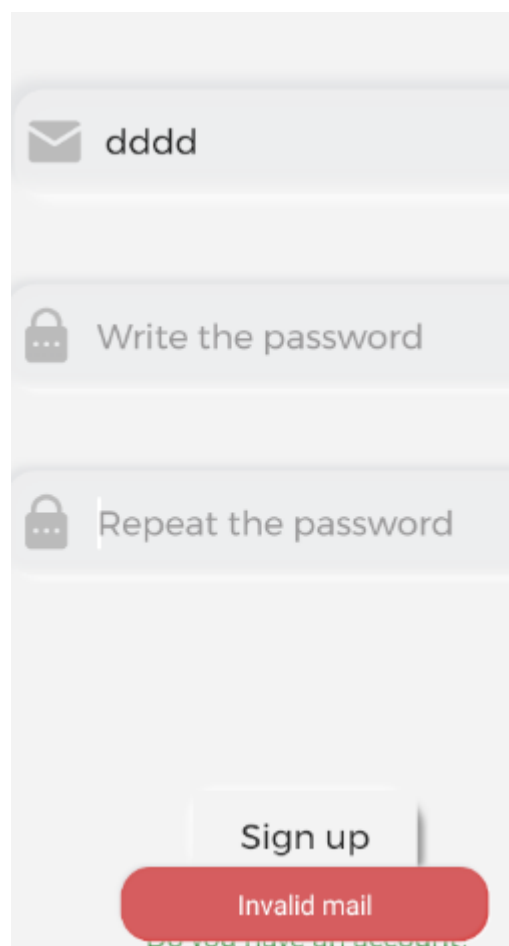
Sin conexión a internet

Actividad Registro de usuario:

El sistema lanzará un mensaje de error si no se dispone de conexión a internet, el mail a registrar ya dispone de cuenta en la aplicación, alguno de los campos está vacío, si el mail o contraseña no son válidos o si la repetición de la contraseña no coincide con su anterior campo





Campos vacíos




Mail no válido

Sign up

 luisalvarez7696@gmail.com






[Sign up](#)


You must have internet connection to use this app.


[Do you have an account?](#)

Sin conexión a internet

Sign up

 luisalvarez7696@gmail.com



 Repeat the password


[Sign up](#)


The password must have 8 characters at least.


[Do you have an account?](#)

Contraseña no válida

Sign up

 luisalvarez7696@gmail.com



 ..


[Sign up](#)


Passwords doesn't match.


[Do you have an account?](#)

Las contraseñas no coinciden

Sign up

 luisalvarez7696@gmail.com





[Sign up](#)

¡Oops! We couldn't sign up you. You may already have an account ...


[Do you have an account?](#)

Cuenta ya registrada

Actividad restablecer contraseña:

En esta actividad se lanzar un mensaje de error si: No existe conexión a internet, el usuario no está registrado o el mail no es válido.

Did you forget your password?
We will send you an email to reset your password.

 example@zayta.com


Submit

You must have internet connection to use this app.

Log in

No existe conexión a internet

Did you forget your password?
We will send you an email to reset your password.

 example@zayta.com


Submit

Mail field couldn't be empty

Log in

Mail vacío

Did you forget your password?
We will send you an email to reset your password.

 aaaa


Submit

Invalid mail

Log in

Mail no válido

Did you forget your password?
We will send you an email to reset your password.

 luisalvare@gmail.com

Submit

Your mail seems not to be sign up...

Log in

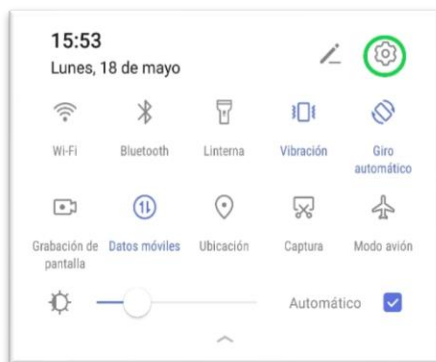
Mail no registrado en la base de datos

4. Documentación

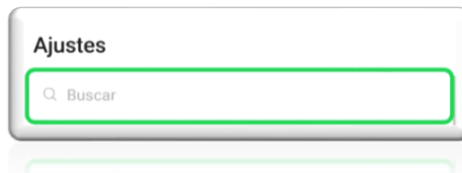
Manual de instalación y configuración.

Para instalar la aplicación solo deberemos tener activada la instalación de aplicaciones externas desde el navegador o aplicación desde la que lo instalemos.

Ejemplo: Si nos descargamos la aplicación desde un enlace con Google Chrome:

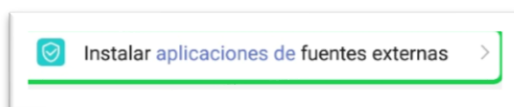


1. Ajustes

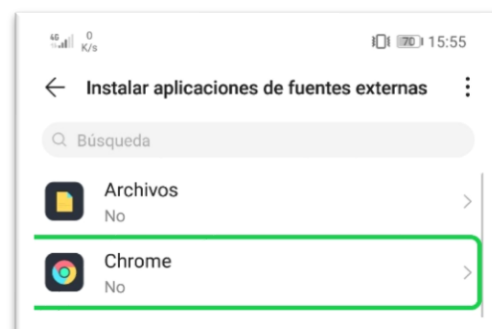


2. Barra de búsqueda: Escribir

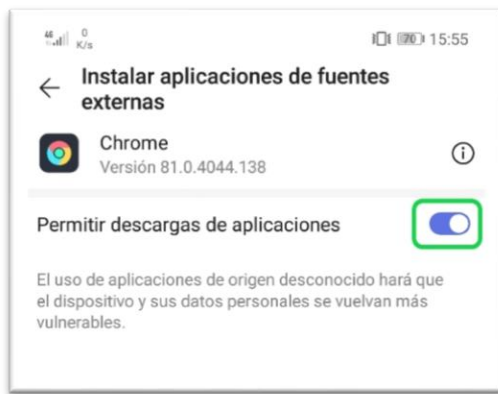
"Instalar aplicaciones de fuentes externas"



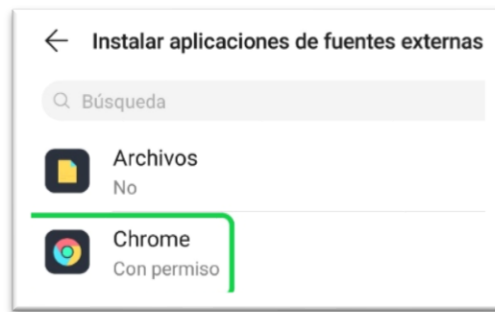
3. Seleccionar opción



4. Seleccionar Chrome

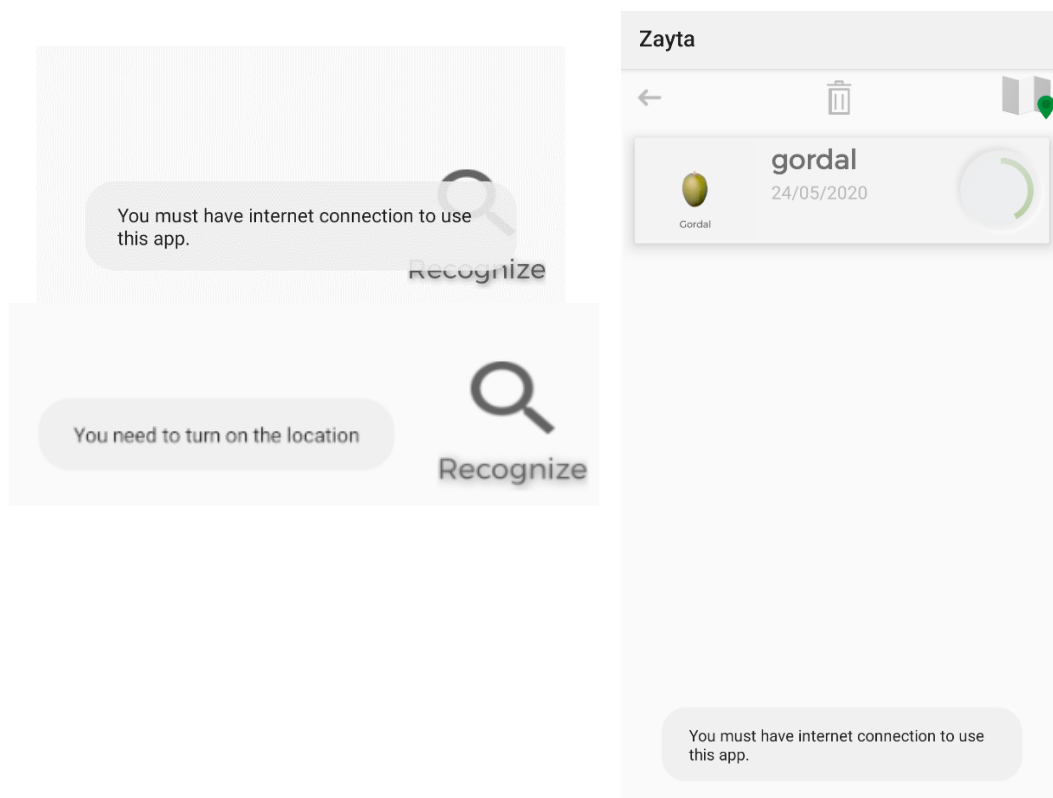


5.Activar switch



6.Resultado Final

En el resto de las actividades he tenido en cuenta la conexión a internet del usuario y gps, ya que, al realizar un reconocimiento, guardar o visualizar un registro, la aplicación lo requiere. Entonces, si el usuario no tiene activo alguno de ambos, la aplicación lo notifica.



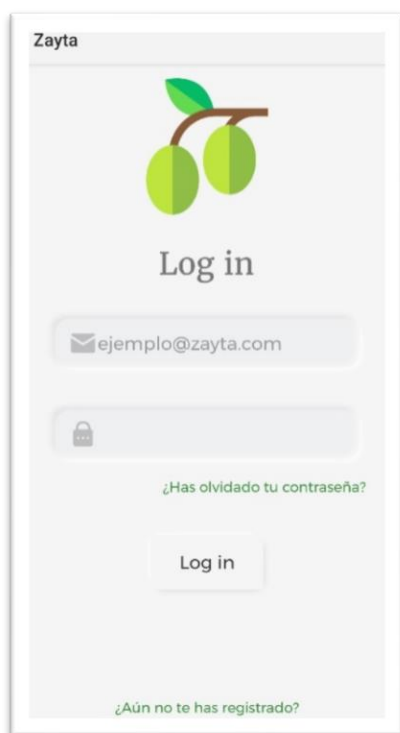
Al lanzar la aplicación veremos la actividad **Splash Screen**, durante unos segundos. Acto seguido, si el usuario no ha iniciado sesión con anterioridad, la aplicación requerirá el mail y contraseña.

Para utilizar la aplicación se necesita conexión a internet y ubicación del usuario, si no los tiene activados, la aplicación los requerirá.

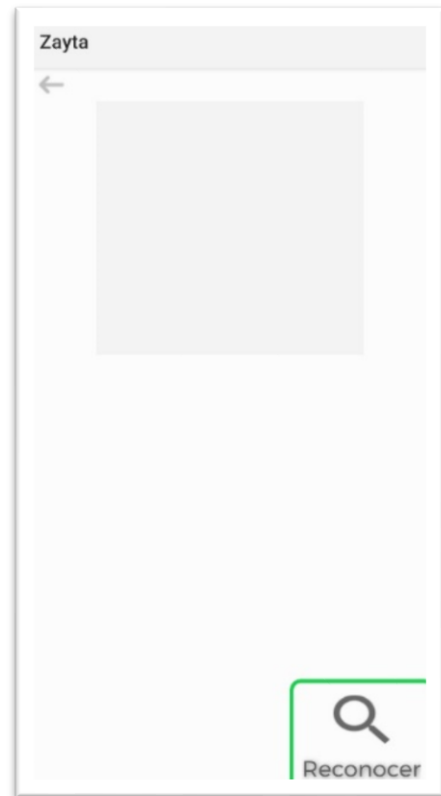


Log In:

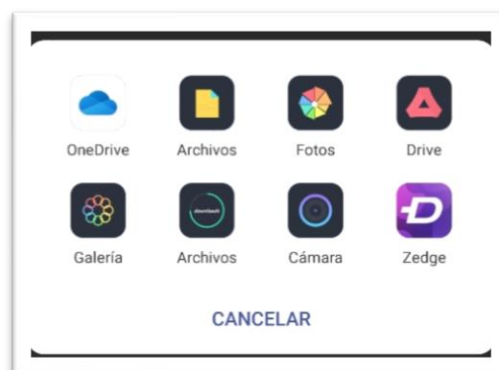
Introducir mail y contraseña y pulsar el botón de **Log In**.



Reconocer una variedad: En el menú principal, seleccionar la opción de Reconocimiento. Dentro de esta actividad encontraremos un botón con forma de lupa.

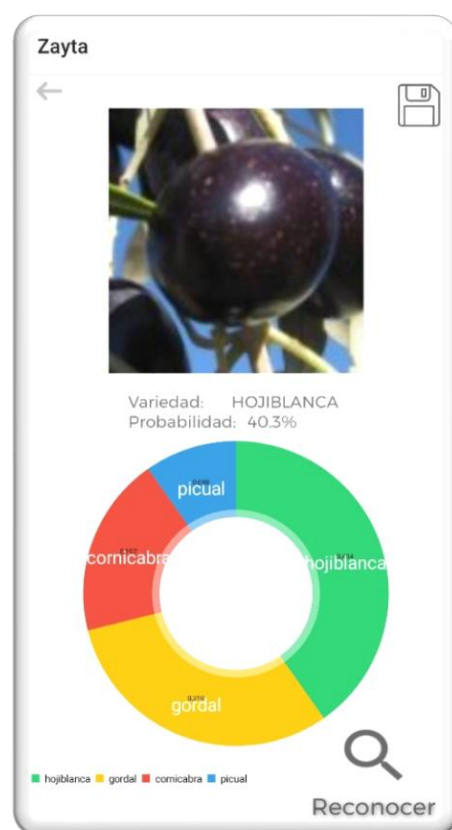
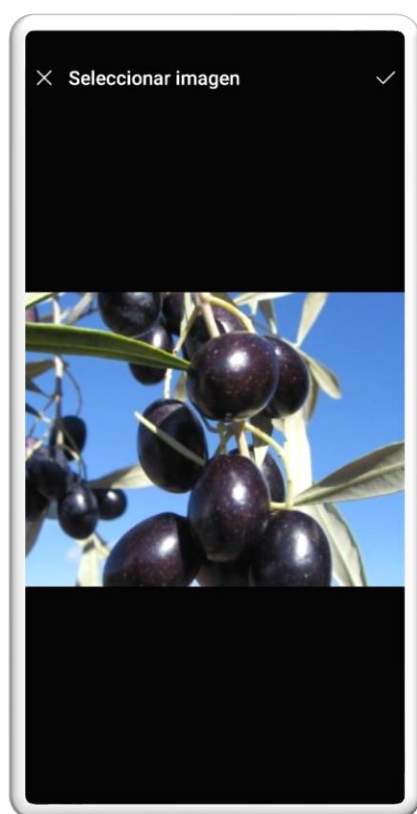


Al seleccionar la opción Reconocer, comenzará el proceso de selección de imagen. Este puede ser realizado por cualquiera de los medios de selección de imagen de los que disponga el dispositivo. Desde la cámara hasta la galería.



Al seleccionar la imagen, comenzará el proceso de recorte.
Después de seleccionar la franja donde aparece una aceituna y pulsar en “**Cortar**” comenzará el proceso de reconocimiento y etiquetado de la imagen.

Los resultados serán mostrados junto con la imagen recortada.



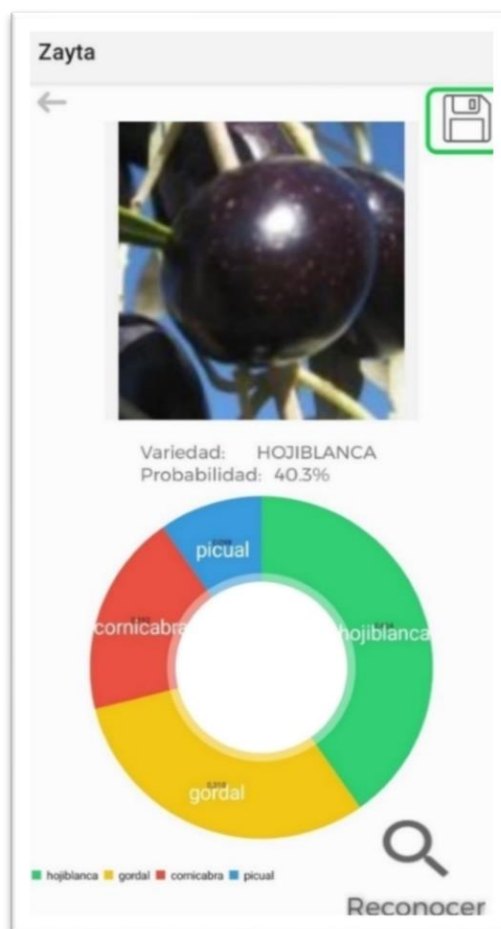
Guardar un registro:

Una vez hayamos realizado el reconocimiento de una variedad, aparecerá el botón "**Guardar**" en la parte superior derecha de la pantalla. El usuario pulsará ese botón y se producirá el guardado del registro.

Tras completar el proceso de guardado, aparecerá un mensaje de retroalimentación y la actividad Reconocimiento volverá a su estado inicial.

Si no se desea guardar el registro, simplemente ha de pulsar de nuevo el botón "**Reconocer**".

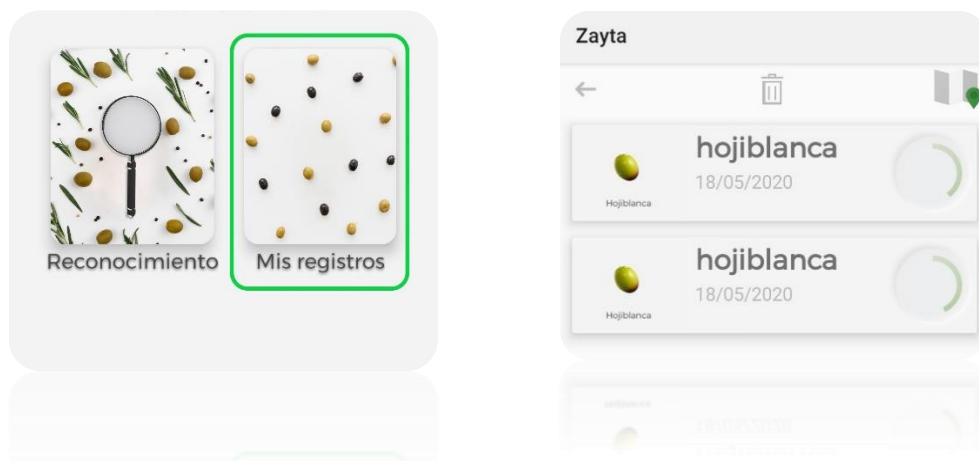
Este registro, aparecerá posteriormente en la actividad **Mis Registros**, para poder consultarlo con una vista previa desde dicha actividad o desde una vista individual al seleccionarlo.



Consultar la lista de registros:

Desde el menú principal el usuario seleccionará la opción **Mis Registros**.

Al lanzarse la aplicación el usuario podrá ver la lista de todos los registros que haya guardado.



Consultar mapa de todos los registros:

En el menú principal seleccionar **Mis Registros**.

Al lanzar la actividad en la parte superior derecha de la pantalla se divisará un botón en forma de mapa.

Al pulsarlo se lanzará la actividad correspondiente. Y mostrará todas las variedades de aceitunas registradas.

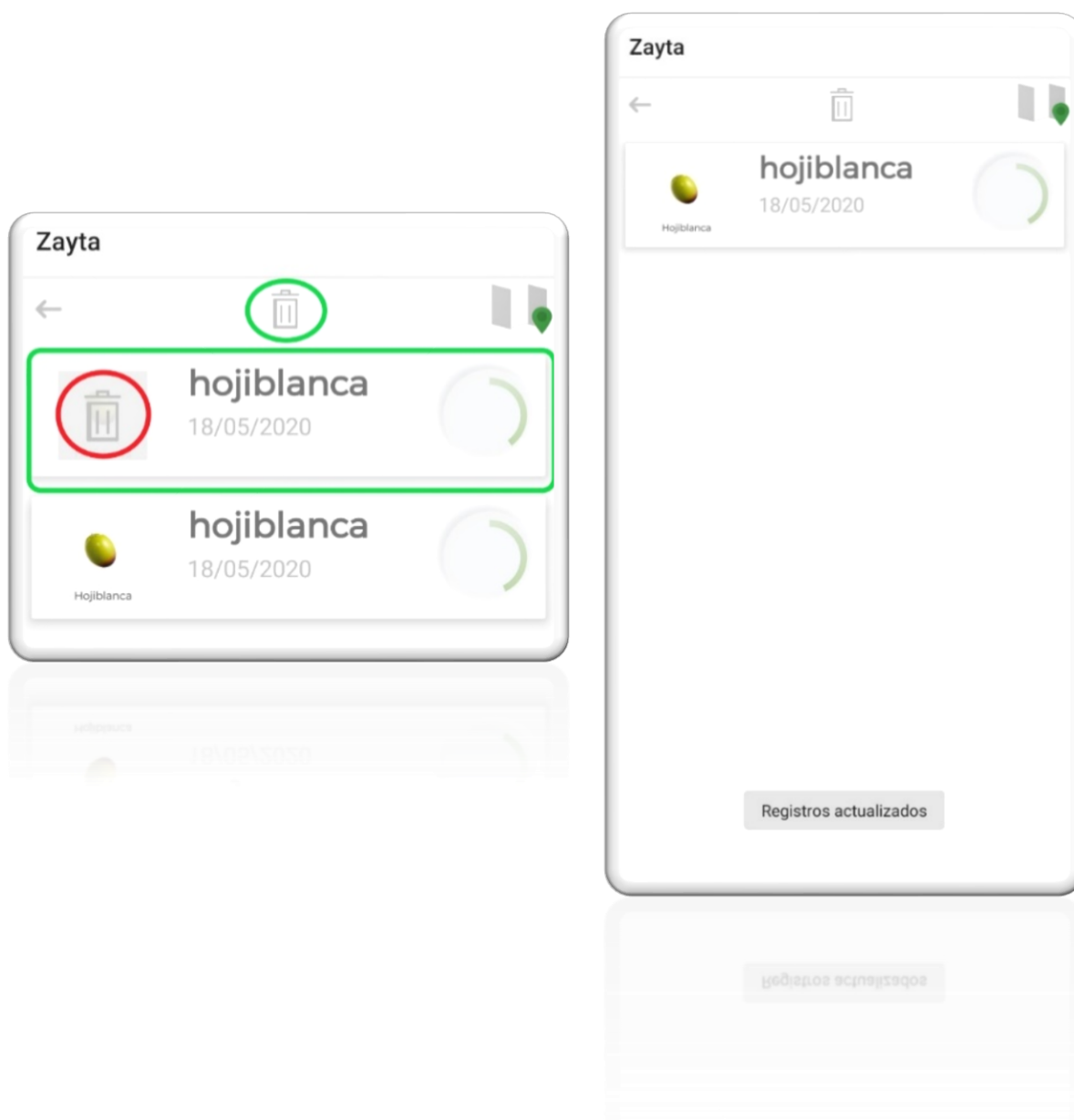
Al seleccionar una aceituna en esta actividad aparecerá un mensaje indicando su variedad.



Borrar uno o varios registros:

Desde el Menú Principal seleccionamos la opción Mis Registros. Esta vez, para seleccionar los elementos a borrar, mantendremos pulsado dicho elemento hasta que este muestre un icono con una papelera.

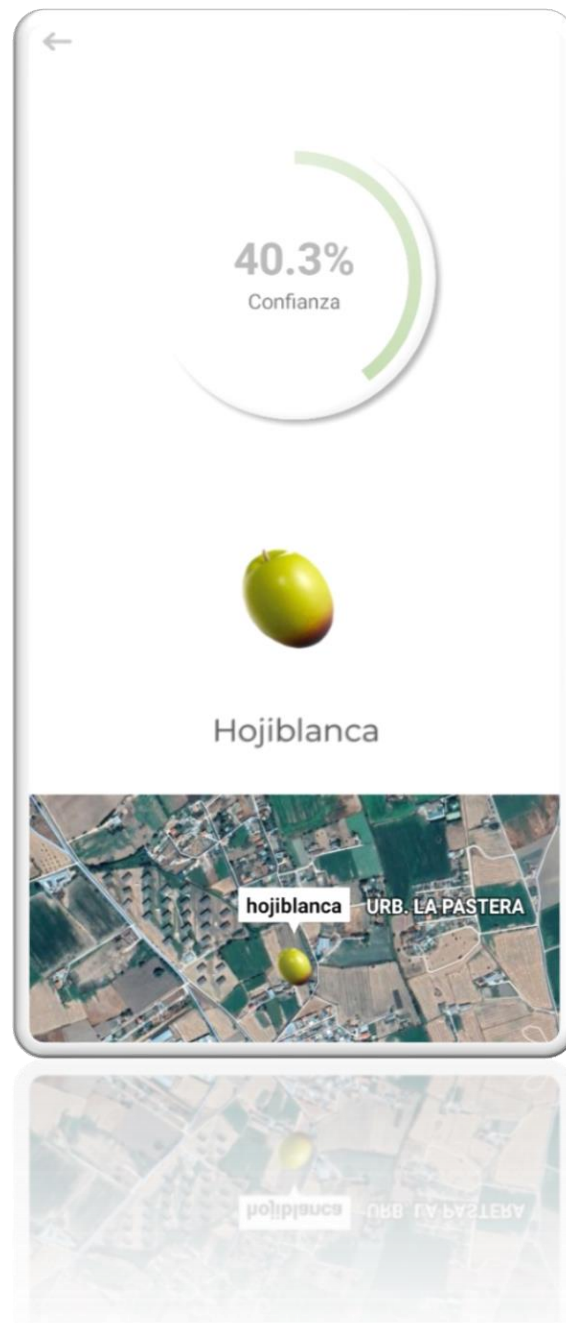
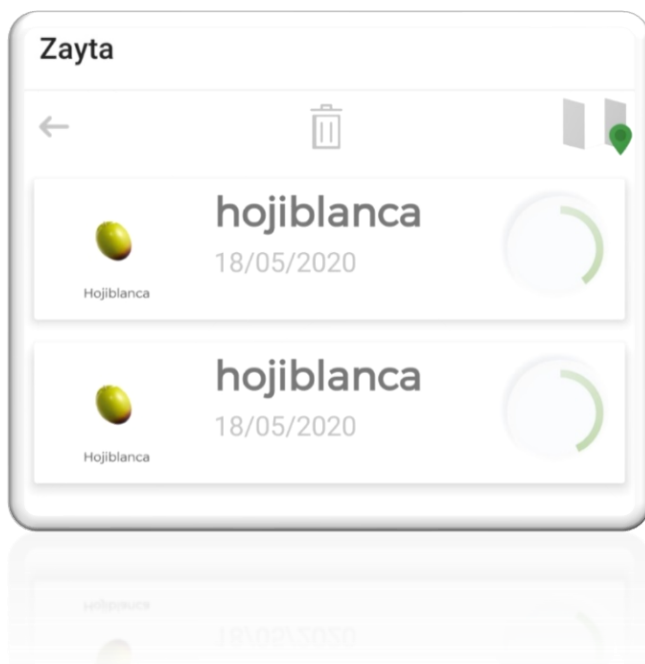
Una vez seleccionados todos los elementos a eliminar, pulsaremos la papelera situada en la parte centro-superior de la pantalla y la aplicación nos mostrará un mensaje confirmando el resultado del proceso.



Consultar un registro:

En el menu principal seleccionar **Mis Registros**.

Seleccionar un elemento Registro con una breve pulsación o toque.



Registro:

En la actividad de **Log In** seleccionar la opción de **¿Aún no te has registrado?**

Introducir mail, contraseña y repetir contraseña y pulsar el botón **Registro**.

Una vez se haya registrado el usuario, se iniciará sesión automáticamente y se lanzará el Menú de Inicio.

The image shows a mobile app screen for registration. At the top, the header 'Zayta' is visible. Below it is a logo consisting of two green circles with a brown stem. The title 'Registro' is centered. There are three input fields: the first contains the email 'raymondredd@gmail.com', the second and third contain masked passwords represented by dots. A green rectangular box highlights the 'Registro' button at the bottom. Below the button, there is a link that says '¿Ya tienes una cuenta?'.

Reestablecer contraseña:

En la actividad **Log In** seleccionar la opción:

¿Has olvidado tu contraseña?

Introducir el mail con el que el usuario se registró.

Pulsar **Enviar**.

Se mostrará por pantalla el mensaje confirmando que se ha enviado el correo.

Al terminar la operación, llegará un mensaje al correo electrónico con un enlace para restablecer la contraseña.

Acceder al enlace e introducir la nueva contraseña y seleccionar guardar.

The image shows a mobile app screen for password recovery. The header 'Zayta' is at the top. The title '¿Has olvidado tu contraseña?' is centered, followed by a subtitle '⇒ enviaremos un mail para reestablecer tu contraseña.' Below this is an input field containing the email 'ejemplo@zayta.com'. A green rectangular box highlights the 'Enviar' button at the bottom right.

Mail enviado:



Ventana mostrada en el enlace del mail:

The screenshot shows a web form titled 'Cambiar la contraseña'. Below the title, it says 'de luisalvarez7696@gmail.com'. There is a label 'Nueva contraseña' followed by a text input field. To the right of the input field is an eye icon. At the bottom right of the form is a blue button labeled 'GUARDAR'.

The screenshot shows a confirmation message. It starts with the heading 'Se ha cambiado la contraseña'. Below this, it says 'Ya puedes iniciar sesión con la nueva contraseña'.

5. Presupuesto

Consultando diversas fuentes como:

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

<https://cuantocuestaunaapp.com/>

<https://www.yeeply.com/blog/cuanto-cuesta-crear-una-app/>

<https://cotizarmiapp.com/#>

He llegado a la conclusión de que el presupuesto estimado para una aplicación bilingüe con un diseño de últimas tendencias, sistema de log in, registro y restablecimiento de contraseña integrado, que requiere de una base de datos en cloud y almacenamiento por parte de los usuarios en esa nube.

Los precios oscilan entre los 6156,36€ y 9750€, por lo que he decidido valorar la aplicación en un punto intermedio: **7953,18 €**.

Si el cliente requiere de cubrir más variedades de aceitunas en el modelo de datos, tendríamos que contratar el plan Blaze de Firebase e incrementaría el presupuesto en función de las horas de entrenamiento con ML Kit. Hasta el momento Zayta utiliza el plan Spark, el cual es gratuito.

En la realización del presupuesto se ha tenido en cuenta que la aplicación no se va a subir a la Play Store de Google, en caso contrario habría que pagar la facturación anual por tener la aplicación en esta plataforma.

6. Mejoras propuestas

Realizar informes de los registros guardados por el usuario.

Agregar más variedades de aceitunas al modelo de datos.

Reconocimiento en tiempo real.

Añadir información específica de variedades como: Peso medio, rendimiento aproximado y tamaño.

7. Conclusiones

Desarrollar esta aplicación ha sido una experiencia enriquecedora tanto a nivel personal como profesional. Este proceso me ha servido para asentar los conocimientos aprendidos estos años y ponerlos en práctica. Además, he aprendido a utilizar otras tecnologías que desconocía. Este proceso ha supuesto un reto, pero con esfuerzo e interés he conseguido desarrollarlo con éxito. Finalmente, este proyecto me ha hecho darme cuenta de que este sector está en constante renovación y que la manera de adaptarse es seguir aprendiendo.

8. Bibliografía y referencias

Libros:

Editorial	Título	Autor/es
Garceta	Entornos de Desarrollo	Alicia Ramos Martín, M. ^a Jesús Ramos Martín
Garceta	Acceso a Datos	Alicia Ramos Martín, M. ^a Jesús Ramos Martín
Garceta	Android	Iván López Montalbán, Manuel Martínez Carbonell, Juan Carlos Manrique Hernández

Referencias web:

<https://docs.oracle.com/javase/8/docs/api/>

<https://github.com/4inodev/Neomorphic-FrameLayout-Android>

<https://github.com/PhilJay/MPAndroidChart>

<https://medium.com/pildorasux/10-heuristicos-nielsen-abc9c6ad04c0>

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

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