

UNIVERSIDAD DE CASTILLA LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

AgricultON

Gestión de Sistemas de Información
Curso 2020/2021

Juan Manuel Porrero Almansa
Alonso Díaz Sobrino
Desiderio Almansa Porrero



Contenido

- 1. INTRODUCCIÓN.....3**
 - 1.1. PROPUESTA DE PROYECTO.....3
 - 1.2. DESCRIPCIÓN Y JUSTIFICACIÓN DEL TRABAJO.....3
 - 1.3. HERRAMIENTAS UTILIZADAS.3
- 2. FASE I: ANÁLISIS.5**
 - 2.1. REQUISITOS FUNCIONALES.5
 - 2.2. REQUISITOS DE LA INTERFAZ DE USUARIO.5
- 3. FASE II: DISEÑO.6**
 - 3.1. PRESENTACIÓN: PROTOTIPO DE LA INTERFAZ DE USUARIO.6
 - 3.2. DOMINIO.....9
 - 3.3. PERSISTENCIA.11
- 4. FASE III: IMPLEMENTACIÓN.....13**
 - 4.1. INICIO DE SESIÓN.13
 - 4.2. REGISTRO DE NUEVO USUARIO.13
 - 4.3. BARRA DE NAVEGACIÓN: PERFIL.14
 - 4.4. BARRA DE NAVEGACIÓN: PRODUCTOS.....14
 - 4.5. BARRA DE NAVEGACIÓN: CAMPOS.15
 - 4.6. VENTANA DE MÁS INFORMACIÓN Y PESTAÑA CERRAR SESIÓN.16
- 5. BIBLIOGRAFÍA.17**

1. Introducción

Las nuevas tecnologías han facilitado el desarrollo de aplicaciones móviles para cualquier sector, también para el de la producción agrícola. De esta manera, se presta un apoyo técnico importante dentro de los campos de cultivo. Estas aplicaciones para agricultura facilitan que todos los trabajadores del sector (desde agricultores a asesores, agrónomos, institutos de investigación o incluso agencias del gobierno) sean mucho más productivos, eficientes y también rentables en cuanto a la producción agrícola. También, favorecen que el trabajo sea todavía más profesional y eficiente.

1.1. Propuesta de Proyecto

La agricultura es el pilar de la economía en muchos países del mundo, entre ellos España. Factores como el cambio climático, el crecimiento de la población y la seguridad alimentaria impulsan a la industria agroalimentaria a buscar enfoques innovadores para mejorar el rendimiento de los cultivos. La Inteligencia Artificial emerge como parte esencial de la evolución tecnológica de la agroindustria.

En este sentido, nuestro equipo ha pensado en realizar un prototipo de aplicación con la que poder llevar control de los campos de cultivos de una empresa (ver rentabilidad, ubicación, cultivos que podemos encontrar en él, noticias, etc).

1.2. Descripción y justificación del trabajo.

En el trabajo teórico de esta asignatura hemos investigado sobre la Inteligencia Artificial aplicada a la Agricultura. Una de las cosas que tuvimos que investigar fueron los Sistemas Expertos. Estos sistemas de IA logran resolver problemas tal y como lo haría un especialista en un determinado ámbito.

Como ya dijimos en nuestro trabajo, la ayuda de un especialista agrícola no está siempre disponible cuando el granjero lo necesita. Por esto hemos pensado que, para nuestro proyecto, sería interesante intentar resolver este problema.

La combinación de Sistemas Expertos con la Agricultura constituye el pilar principal del origen de nuestro proyecto de prácticas. De esta manera, hemos intentado desarrollar una aproximación a un Sistema Experto en una sencilla aplicación para Android, que podremos utilizar en cualquier parte, estemos o no en mitad del campo. Decimos que es un Sistema Experto porque almacena información sobre cultivos (descripción, cuidados, etc), que el agricultor podrá consultar en cualquier momento.

1.3. Herramientas utilizadas.

ENTORNO DE DESARROLLO

La aplicación está implementada en Android, haciendo uso de **Android Studio** y el lenguaje de programación **Java**. Lo hemos decidido así porque nos ha gustado la idea de programar en esta plataforma, ya que es muy versátil y, en un futuro, nos puede servir de gran ayuda esta experiencia.

También hemos creído que sería más óptimo que pueda usarse desde Smartphones o Tablets, ya que si nos encontramos en un determinado campo de cultivo y necesitamos la información bastaría con sacar el teléfono y abrir la aplicación, sin necesidad de un ordenador.

REPOSITORIO

Para tener un control de las versiones de esta práctica y también para que todos los componentes del grupo pudieran trabajar a la vez, hemos usado un repositorio de código privado en GitHub. Podrás encontrarlo en el siguiente enlace:

<https://github.com/juanma1501/AgricultON>

BASE DE DATOS

Para la parte de persistencia de nuestra aplicación, hemos usado **SQLite**, una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos SQL auto-contenido, sin servidor y sin configuración. Nos hemos decantado por SQLite porque es muy cómoda para los dispositivos móviles, al ser rápida y simple.

Para la proporción de mecanismos básicos para relacionar la aplicación y la información, hemos utilizado el conector SQLiteOpenHelper.

LIBRERIAS DE TERCEROS

Hemos usado dos librerías externas para complementar el desarrollo de nuestra aplicación. Estas son:

- JustifiedTextView. Se trata de una librería que permite justificar texto en una aplicación Android.
- HelloChart. Se trata de una librería que permite implementar gráficos estadísticos.

2. FASE I: ANÁLISIS.

En primer lugar, hay que tener en cuenta que los usuarios finales de la aplicación será un grupo totalmente heterogéneo, por lo que debe ser una aplicación sencilla y la usabilidad de la interfaz debe ser lo más clara y fácil posible. En cuanto a las funciones de lectura o introducción de datos que realizará, serán administradas por la propia aplicación, por lo que la mayor parte de las funcionalidades tendrán que acceder a la base de datos para resolverse.

2.1. Requisitos funcionales.

En la siguiente lista se dan los requisitos funcionales que ha de cumplir la aplicación:

- Registrar usuario existente
- Crear un nuevo usuario. En caso de que no se disponga ya de un registro.
- Mostrar información relevante al usuario.
- Mostrar información relevante a los cultivos/productos que el usuario esté cultivando.
- Crear nuevos perfiles de cultivos/productos.
- Ver localización de los campos con los cultivos que poseen.

2.2. Requisitos de la interfaz de usuario.

En la siguiente lista se dan los requisitos de la interfaz de usuario que ha de cumplir la aplicación:

- A la hora de autenticarse, será necesario un Nombre de Usuario y una Contraseña.
- En caso de que el usuario introduzca mal sus credenciales o no exista, se mostrará un mensaje de error.
- Para la creación de los nuevos usuarios, será necesaria una ventana o formulario en el que se soliciten los datos necesarios.
- Será necesario dividir la aplicación en distintas ventanas, para separar cada funcionalidad. En una tendremos el Perfil del usuario, en otra los Productos, y en otra los Campos.
- Para añadir productos será necesario un formulario para introducir todos sus datos.
- Será necesario un botón para cerrar la sesión o salir de la aplicación.
- De forma adicional también consideramos necesaria una ventana en la que se informe al usuario sobre quienes somos, que es la aplicación y los derechos de autor.

3. FASE II: DISEÑO.

Para desarrollar nuestra aplicación hemos tenido en cuenta las diferentes partes que forman el sistema: la capa de presentación, la capa de datos y la capa de lógica.

Comenzamos nuestro proyecto diseñando la interfaz de usuario, después seguimos modelando la parte lógica y finalizamos añadiendo la parte de persistencia de datos.

3.1. Presentación: Prototipo de la Interfaz de Usuario.

Tras haber identificado los requisitos principales de la aplicación, nos centramos en la elaboración de los bocetos de esta, preocupándonos más por el posicionamiento y diseño de las ventanas, botones y formularios que la controlan, basándonos en los conceptos vistos en las asignaturas de Interacción Persona-Ordenador I e Interacción Persona-Ordenador II.

Para ello, hemos usado la aplicación Balsamiq Mockups, que incluye todas las interfaces del sistema (sin funcionalidad). A continuación, se mostrarán dichos bocetos que pueden haber sufrido o no cambios con respecto al producto final.

VENTANA DE INICIO DE SESIÓN

Para que cada usuario tenga sus datos bien protegidos, hemos pensado que era necesario diseñar una ventana de autenticación como paso previo al uso de las funcionalidades de la aplicación. De esta forma, se podrá guardar los productos que el usuario registre y otra información.

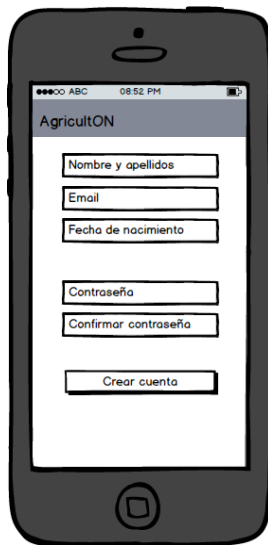
Para la autenticación, la app dispone de un campo para introducir el email, otro para la contraseña, y un botón “Iniciar sesión”. También cuenta con un botón “Registrarse” para poder registrar nuevos usuarios siempre que sea necesario. Al pulsarlo nos llevará a una nueva ventana para hacerlo.

Una vez el usuario se haya autenticado, podrá navegar por el sistema gracias a una barra, que se encuentra en la parte inferior de la pantalla. Esta barra consta de tres botones que nos servirán para desplazarnos por las distintas ventanas: perfil, productos y campos. Hemos decidido situar la barra de navegación ahí debido a que es mucho más fácil para el usuario llegar con el dedo pulgar hasta esa zona de la pantalla si está sujetando el dispositivo con una mano.



Ilustración 3.1: Inicio de sesión.

VENTANA DE REGISTRO



Como hemos mencionado en el apartado anterior, la aplicación dispone de un botón para registrar nuevos usuarios. Al pulsarlo nos aparecerá la ventana de registro, que se trata de una sencilla ventana que consta de varios campos en los que introduciremos los siguientes datos respectivamente: Nombre y apellidos, Email, Fecha de nacimiento, Contraseña y Confirmación de contraseña. Cuando el usuario halla introducido todos sus datos, deberá pulsar el botón que se encuentra en la parte inferior de la pantalla, “Crear cuenta”, para que la operación se realice.

Ilustración 3.2: Ventana de registro.

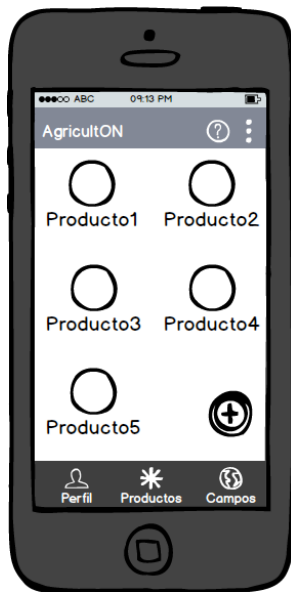
VENTANA PERFIL

Al pulsar el botón “Perfil” de la barra de navegación, nos llevará a esta ventana, en la que encontraremos información sobre el usuario: Email, Fecha de nacimiento y Hora de ultima conexión. En la parte superior podremos encontrar el nombre y la foto de perfil del usuario, y en la parte inferior podemos ver un gráfico con información sobre los pasos dados.



Ilustración 3.3: Ventana de Perfil de Usuario.

VENTANA PRODUCTOS



Cuando pulsamos el botón “Productos” en la barra de navegación, ingresaremos en la ventana de productos. Esta ventana es muy simple, únicamente consta de un listado de los productos que tenemos añadidos, y un botón en la parte inferior izquierda con un símbolo de “+”, que nos servirá para añadir nuevos productos.

Al pulsar el botón de añadir nuevos productos, aparecerá un formulario en el que podremos introducir la información relevante a este. Consta de tres campos, uno para insertar una imagen del producto, otro para el nombre y otro para escribir su descripción y cuidados. Una vez se introduzcan los datos, se podrá guardar el producto pulsando el botón de “Guardar”, o por el contrario cancelar la operación pulsando el botón “Cancelar”.

Por otro lado, estando en la ventana de productos, si pulsamos cualquiera de los productos nos llevara a otra ventana en la que aparecerá la información relevante al producto seleccionado.

Ilustración 3.4: Ventana de Productos.

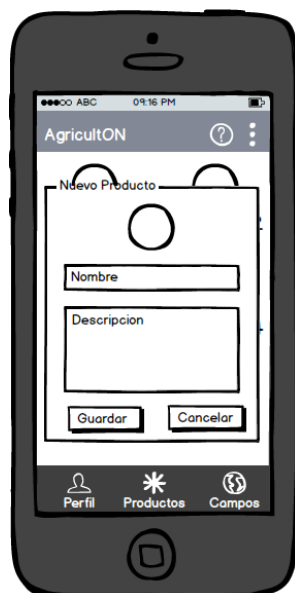


Ilustración 3.5: Formulario para nuevos Productos/Cultivos.

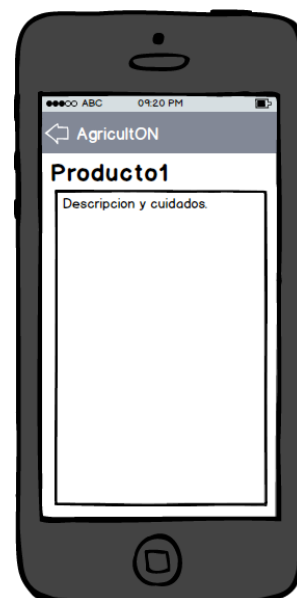
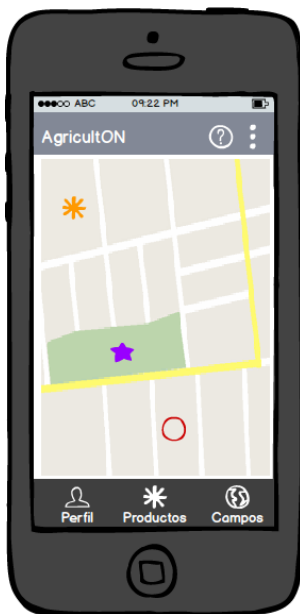


Ilustración 3.6: Ventana de información del Producto seleccionado.

VENTANA CAMPOS



Cuando pulsamos el botón “Campos” de la barra de navegación, iremos a la ventana de campos. En esta ventana se nos mostrará un mapa de la zona, y en este aparecerán señalados los campos que contienen cultivos con el icono del producto.

Ilustración 3.7: Ventana de Campos.

3.2. Dominio.

En cuanto a la lógica de negocio, el usuario es representado por la clase **Usuario** con las propiedades necesarias para realizar la autenticación, “idUser” y “contrasena”. Además, cuenta con otros atributos como “nombre”, “apellidos”, “fechaNacimiento”, “email” y “foto”, que nos darán la información necesaria sobre quien es el usuario logeado. También tiene un atributo “ultimoAcceso” que nos proporcionará la fecha y hora del ultimo acceso a la aplicación. Y por último un Array “listaPasos” en el que iremos registrando los pasos que va realizando el usuario.

Respecto a la clase **Cultivo**, se trata de la clase que representa los productos/cultivos que el usuario inserta en la app. Consta de tres atributos “nombre” que hace referencia al nombre del cultivo, “descripcion” que guardará la información relevante a sus cuidados, características, etc., y “foto” que almacenará la imagen del cultivo.

Por último, la clase **Paso** hace referencia a los pasos que el usuario ha dado en un determinado día. Para ello consta de tres atributos: “idUser” que nos dirá que usuario ha dado los pasos, “pasos” que almacena la cantidad de pasos dados y “fehca” que nos indica el día que los dio.

A continuación, se muestra el diagrama de clases:

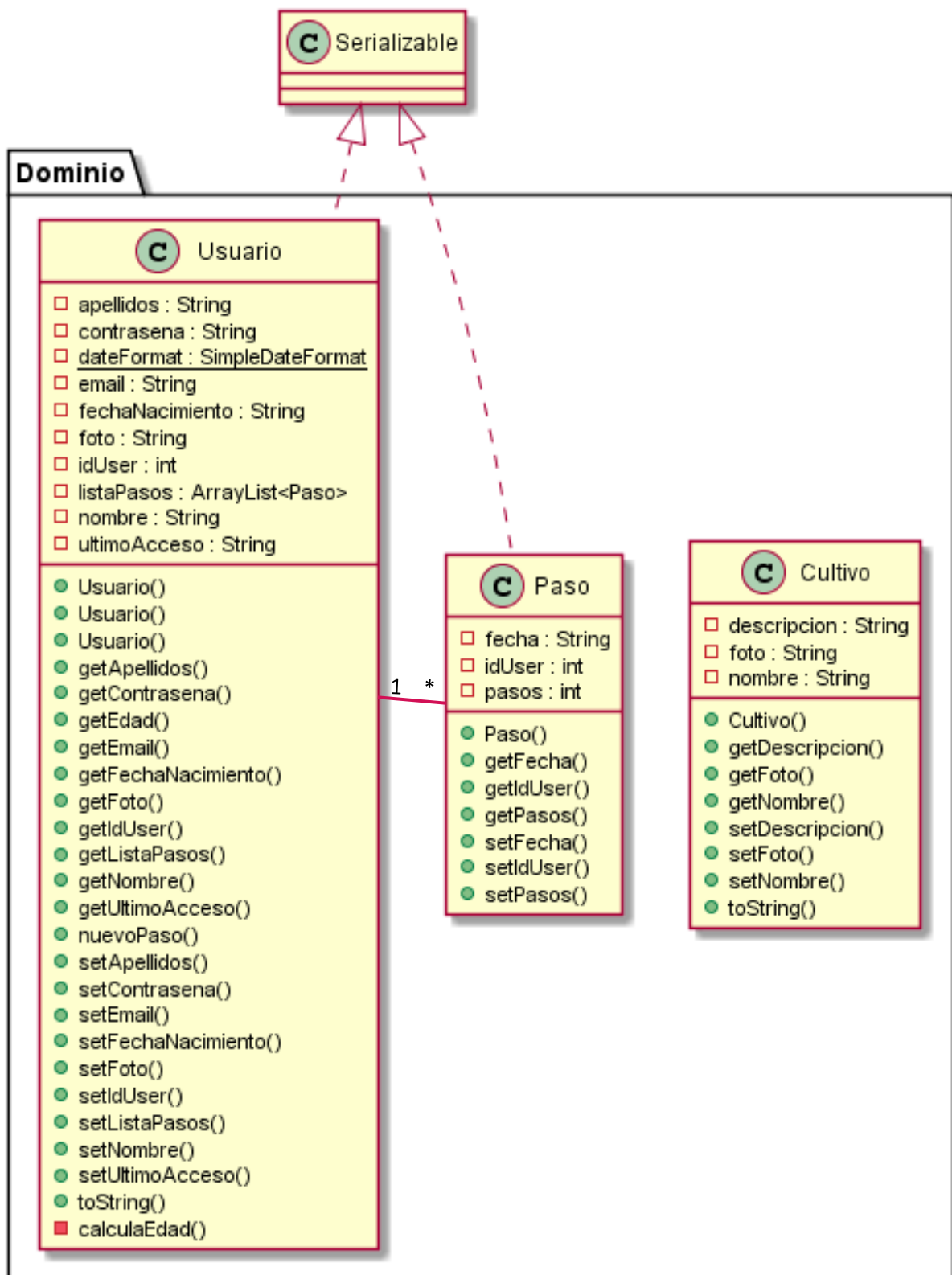


Ilustración 3.8: Diagrama de clases de Dominio.

3.3. Persistencia.

Entre todos los sistemas de bases de datos open source (SQLServer, MySQL, Oracle, etc), hay uno en particular que se adapta muy bien a las aplicaciones móviles, el **SQLite**. Este sistema de base de datos solo requiere un fichero para almacenar los datos, ya que su lógica de funcionamiento se implementa por la plataforma que desee interactuar con los datos.

En el paquete de Persistencia podemos encontrar dos clases, **ConectorDB** y **AdminSQLiteOpenHelper**. Con la clase ConectorDB creamos la conexión con la base de datos y también leemos los datos de cada tabla y los vamos almacenando. Con respecto a la clase AdminSQLiteOpenHelper, la usamos para administrar las tablas, crearlas, destruirlas, para almacenar los datos predeterminados de los cultivos, etc.

Nuestra base de datos contiene tres tablas, una para Usuarios, otra para Cultivos y una última para Pasos.

```
public void onCreate(SQLiteDatabase BD) {
    try {

        BD.execSQL("create table Usuarios (
            idUser INTEGER PRIMARY KEY AUTOINCREMENT,
            email text unique,
            contrasena text,
            nombre text,
            apellido text,
            fecha_nacimiento text,
            ultimoAcceso text,
            foto text)");

        BD.execSQL("create table Cultivos (
            idCultivo INTEGER PRIMARY KEY AUTOINCREMENT,
            nombre text unique,
            descripcion text,
            foto text)");

        BD.execSQL("create table Pasos (
            idPaso INTEGER PRIMARY KEY AUTOINCREMENT,
            idUser integer,
            pasos integer,
            fecha text)");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Ilustración 3.9: Fragmento de código de creación de las tablas.

IMPLEMENTACION EN ANDROID

Para implementar una nueva base de datos SQLite en Android hemos seguido los siguientes pasos:

1. Crear una clase que extienda SQLiteOpenHelper. En nuestra aplicación esta clase corresponde con **AdminSQLiteOpenHelper**.

2. Escribir en ella, gracias al método **onCreate()** que ejecutará los comandos necesarios para crear las tablas en la base de datos (Ilustración 3.9).
3. También es necesario un método que se ejecute cada vez que cambiamos la base de datos para hacer que los datos cambien de la anterior a la nueva versión, para ello usamos el método **onUpgrade()** (Ilustración 3.10).

```
public void onUpgrade(SQLiteDatabase BD, int oldVersion, int newVersion) {
    try {
        /*Se elimina la versión anterior de la table*/
        BD.execSQL("DROP TABLE IF EXISTS Usuarios");
        BD.execSQL("DROP TABLE IF EXISTS Cultivos");
        BD.execSQL("DROP TABLE IF EXISTS Pasos");

        /*Se crea la nueva versión de la table*/
        BD.execSQL("create table Usuarios (idUser INTEGER PRIMARY KEY
            AUTOINCREMENT, email text unique, contrasena text, nombre text,
            apellido text, fecha_nacimiento text, ultimoAcceso text, foto text)");

        BD.execSQL("create table Cultivos (idCultivo INTEGER PRIMARY KEY
            AUTOINCREMENT, nombre text unique, descripcion text, foto text)");

        BD.execSQL("create table Pasos (idPaso INTEGER PRIMARY KEY
            AUTOINCREMENT, idUser integer, pasos integer, fecha text)");

        insertarDatosPorDefecto(BD);
    } catch (SQLException e){
        e.printStackTrace();
    }
}
```

Ilustración 3.10: Fragmento de código del método onUpgrade.

4. FASE III: IMPLEMENTACIÓN.

4.1. Inicio de sesión.

The image displays two mobile app screens for the login page of the AgricultON application. Both screens feature a purple header with the 'AgricultON' logo and name. The main content area is white and contains two input fields: 'Email' and 'Contraseña'. Below these fields are two purple buttons: 'INICIAR SESIÓN' and 'REGISTRARSE'. At the bottom, there are three social media icons (Facebook, Google, and Twitter). The right screen includes an additional grey button with the text 'Rellena el email y la contraseña' located below the social media icons.

Ilustración 4.1: Implementación de la ventana de autenticación.

4.2. Registro de nuevo usuario.

The image displays two mobile app screens for the registration page of the AgricultON application. Both screens feature a purple header with a back arrow and the 'AgricultON' logo. The main content area is white and contains six input fields: 'Nombre', 'Primer Apellido', 'Correo Electrónico', 'Fecha de nacimiento', 'Contraseña', and 'Confirmar Contraseña'. Below these fields is a purple button labeled 'CREAR CUENTA'. The right screen includes an additional grey button with the text 'Rellena el nombre.' located above the 'CREAR CUENTA' button. At the bottom of both screens, there is a link that says '¿Tienes cuenta? Inicia Sesión'.

Ilustración 4.2: Implementación de la ventana de registro de nuevo usuario (dependiendo del campo que falte por rellenar, saldrá un aviso distinto).

4.3. Barra de navegación: Perfil.

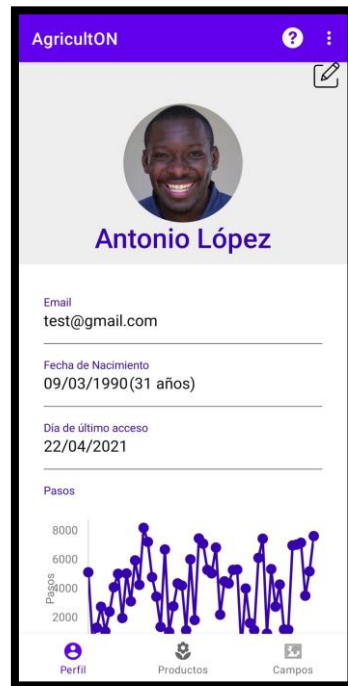


Ilustración 4.3: Implementación de la ventana de Perfil.

4.4. Barra de navegación: Productos.



Ilustración 4.4: Implementación de la ventana de Productos.



Ilustración 4.5: Implementación de la ventana del Producto seleccionado.



Ilustración 4.6: Implementación de la ventana de Registro de un Nuevo Producto.

4.5. Barra de navegación: Campos.

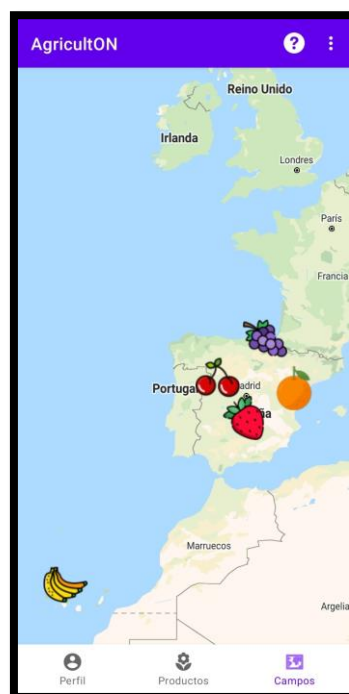


Ilustración 4.7: Implementación de la ventana de Campos.

4.6. Ventana de Más Información y Pestaña Cerrar sesión.

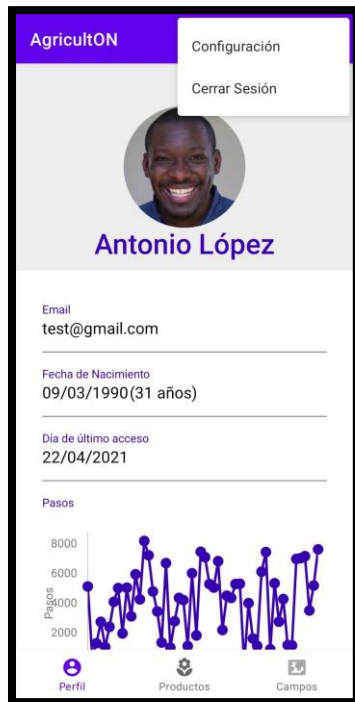


Ilustración 4.8: Implementación de la pestaña que contiene Configuración y Cerrar Sesión.



Ilustración 4.9: Implementación de la pestaña de Más Información.

5. BIBLIOGRAFÍA.

NO ONLINE.

- José Enrique Amaro Soriano. Android: Programación de dispositivos móviles a través de ejemplos. Marcombo, 2011.

ONLINE.

- Android Studio - Guía del Usuario - URL: <https://developer.android.com/studio/intro?hl=es>
- Pildorasinformáticas - Curso de Android con Android Studio - URL: <https://www.youtube.com/watch?v=pdYkmCcQFd8&list=PLU8oAlHdN5Bkn-KS1sRFISEnXXcAtAJ9P>
- codedmin. Android Line Chart – How to Draw Line Chart in Android - URL: <https://www.codingdemos.com/draw-android-line-chart/>
- codedmin. Android Pie Chart – How to Create Pie Chart in Android Studio - URL: <https://www.codingdemos.com/android-pie-chart-tutorial/>