

# Trabajo de Fin de Grado

Grado en Ingeniería Informática

---

## Desarrollo de aplicaciones multiplataforma en el sector agrícola ecológico de Canarias

*Development of multiplatform applications in the organic  
farming sector of the Canary Islands*

Ruymán Rodríguez Martín

---

La Laguna, 4 de julio de 2020

Dña. **Jezabel Miriam Molina Gil**, con N.I.F. 78.507.682-B profesor Ayudante Doctor adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

Dña. **Yanira González González**, con N.I.F. 78.555.933-P Doctora en Informática, contratada como Técnico de Grado Medio en el Servicio de Sistemas de Información del Complejo Hospitalario Universitario de Canarias, como cotutora.

## **C E R T I F I C A (N)**

Que la presente memoria titulada:

*“Desarrollo de aplicaciones multiplataforma en el sector agrícola ecológico de Canarias”*

ha sido realizada bajo su dirección por D. **Ruymán Rodríguez Martín**,  
con N.I.F. 78.559.460-Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de julio de 2020

# Agradecimientos

A Jezabel Miriam Molina Gil y a Yanira González González por proporcionarme un TFG, asesorarme durante todo el proceso y por creer en mis capacidades.

A Iballa González González, productora del sector agrícola ecológico de Canarias, por toda la ayuda e información que me ha proporcionado para poder llevar a cabo este proyecto.

Finalmente, a mi familia, por apoyarme siempre durante los años de estudio del grado.

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

## Resumen

*La agricultura ecológica es un sector que se encuentra al alza, debido a la concienciación que está adquiriendo la sociedad sobre el cambio climático. Este tipo de agricultura respeta el medio ambiente y obtiene productos de la mejor calidad. Es importante, proporcionar todas las herramientas posibles a este tipo de producciones, para que se realicen a nivel local y así no tener que importarlos, pues la propia importación no favorece al medio ambiente.*

*Por ello, el objetivo de este proyecto es facilitar lo máximo posible el trabajo administrativo que se realiza a pie de campo. Se ha desarrollado una herramienta que permite a los productores registrar sus labores diarias desde el campo o desde casa. Y, para acelerar las certificaciones que requieren estas producciones, se permite el acceso de los inspectores al cuaderno de campo mediante usuarios con un rol específico, que les permitirá acceder a los datos necesarios en modo sólo lectura.*

*La aplicación está disponible tanto en versión web como para dispositivos móviles y tabletas. Se ha apostado por crear una interfaz intuitiva, elegante y amigable, pero con toda la potencia que requiere.*

*Para el desarrollo del proyecto se han utilizado las tecnologías más actuales como son React.js, React Native, Firestore o Spring Boot.*

**Palabras clave:** Agricultura ecológica, cuaderno de campo, Canarias, React, React Native, Spring Boot

## Abstract

*The organic farming is a growing sector due to the conscientization that the society is taking because of climate change. This kind of farming respects the environment and it obtains better quality products. It is important to provide all the possible tools to this kind of productions. Carrying out them locally, does not imply import them, since the import itself does not favor the environment.*

*For this reason, the objective of this project is to facilitate as much as possible the administrative work carried out at the field level. A tool, that allows producers to record their daily work from the field or from home, has been developed. It allows them to keep the field notebook updated. Besides, in order to accelerating the certifications that these productions require, inspectors have been allowed to access to the field notebook through specific users' roles. This will provide them access to the necessary data, in read-only mode.*

*The app is available both in web version, smartphones and tablets. It has been opted for an intuitive, friendly and elegant interface but with the necessary power.*

*For the development of this project it has been used the most current technologies as React.js, React Native, Firestore or Spring Boot.*

**Keywords:** organic farming, field notebook, Canary Islands, React, React Native, Spring Boot

# Índice general

<b>Capítulo 1</b>	<b>Introducción.....</b>	<b>1</b>
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Estructura del documento.....	2
<b>Capítulo 2</b>	<b>Estado del arte.....</b>	<b>4</b>
2.1	Tipos de agricultura.....	4
2.1.1	Convencional.....	5
2.1.2	Ecológica.....	5
2.1.3	Integrada.....	5
2.2	Aplicaciones existentes.....	5
2.2.1	Agroptima.....	5
2.2.2	Toolagro.....	6
2.2.3	Agricolum.....	6
2.2.4	Conclusión del estudio de las aplicaciones existentes.....	6
<b>Capítulo 3</b>	<b>Preliminares.....</b>	<b>7</b>
3.1	Vocabulario.....	7
3.2	Definición.....	7
3.3	Funcionamiento de la aplicación.....	9
3.4	Distribución.....	9
<b>Capítulo 4</b>	<b>Tecnologías utilizadas.....</b>	<b>11</b>
4.1	Base de datos.....	11
4.2	Backend.....	12

4.3	Frontend web.....	14
4.4	Frontend para dispositivos móviles y tabletas.....	15
4.5	Otros.....	16
<b>Capítulo 5</b>	<b>Base de datos.....</b>	<b>17</b>
5.1	Entidades y atributos.....	17
5.2	Relaciones entre entidades.....	22
5.3	Restricciones.....	24
5.4	Actualización y eliminación de datos.....	24
5.5	Diagrama entidad-relación.....	25
5.6	Diagrama del modelo lógico relacional.....	26
5.7	Obtención de credenciales.....	27
<b>Capítulo 6</b>	<b>Backend.....</b>	<b>28</b>
6.1	Control de integridad de los datos.....	28
6.2	Seguridad.....	30
6.2.1	Tokens de autenticación.....	30
6.2.2	Control de acceso a recursos.....	30
6.2.3	Control de fallos de inicio de sesión.....	30
<b>Capítulo 7</b>	<b>Interfaz web y móvil.....</b>	<b>32</b>
7.1	Selección de colores.....	32
7.2	Diseño.....	33
7.2.1	Formulario de inicio de sesión.....	33
7.2.2	Menú lateral.....	33
7.2.3	Formularios.....	34
7.2.4	Listas.....	36
7.2.5	Tarjetas.....	38
7.2.6	Listas de tarjetas.....	38
7.2.7	Campo de búsqueda.....	39
7.2.8	Filtro de elementos por su estado.....	40
7.2.9	Formularios especiales.....	40
7.3	Funcionalidad.....	42

7.4 Exportación de datos.....	43
<b>Capítulo 8 Presupuesto.....</b>	<b>45</b>
<b>Capítulo 9 Conclusiones y líneas futuras.....</b>	<b>46</b>
<b>Capítulo 10 Summary and Conclusions.....</b>	<b>47</b>

# Índice de figuras

Figura 2.1: Fragmento de cuaderno de campo del Gobierno de Canarias.....	4
Figura 2.2: Agroptima.....	5
Figura 2.3: Toolagro.....	6
Figura 2.4: Agricolum.....	6
Figura 3.1: Estructura del sistema.....	9
Figura 4.1: MySQL Workbench.....	12
Figura 4.2: Spring Tool Suite 4 for Eclipse.....	14
Figura 4.3: Atom.....	15
Figura 5.1: Modelo entidad-relación.....	25
Figura 5.2: Modelo lógico relacional.....	26
Figura 5.3: Obtención de credenciales.....	27
Figura 6.1: Código de control de restricciones de los cultivos.....	29
Figura 6.2: Código de inicio de sesión y control de fallos de autenticación .....	31
Figura 7.1: Selección de colores con Color Tool de Material Design.....	32
Figura 7.2: Formulario de inicio de sesión (móvil).....	33
Figura 7.3: Formulario de inicio de sesión (web).....	33
Figura 7.4: Menú lateral (móvil).....	34
Figura 7.5: Menú lateral (web).....	34
Figura 7.6: Formulario (móvil).....	34
Figura 7.7: Formulario (web).....	34
Figura 7.8: Formulario de fertilizantes (React Native - móvil).....	35

Figura 7.9: Lista 1 (móvil).....	36
Figura 7.10: Lista 2 (móvil).....	36
Figura 7.11: Lista (web).....	37
Figura 7.12: Lista de fertilizantes (React - web).....	37
Figura 7.13: Tarjeta (móvil).....	38
Figura 7.14: Tarjeta (web).....	38
Figura 7.15: Lista de tarjetas (móvil).....	39
Figura 7.16: Lista de tarjetas (web).....	39
Figura 7.17: Campo de búsqueda (móvil).....	39
Figura 7.18: Campo de búsqueda (web).....	39
Figura 7.19: Filtro (web).....	40
Figura 7.20: Filtro (móvil).....	40
Figura 7.21: Formulario documento 1 (móvil).....	40
Figura 7.22: Formulario documento 2 (móvil).....	40
Figura 7.23: Formulario documento (web).....	41
Figura 7.24: Formulario producción 2 (móvil).....	41
Figura 7.25: Formulario producción 1 (móvil).....	41
Figura 7.26: Formulario producción 1 (web).....	42
Figura 7.27: Formulario producción 2 (web).....	42
Figura 7.28: Exportar a Excel (móvil).....	44
Figura 7.29: Exportar a Excel (web).....	44
Figura 7.30: Documento de Excel resultante de la exportación.....	44

# Índice de tablas

Tabla 3.1: Funcionalidades de la aplicación.....	8
Tabla 8.1: Presupuesto del proyecto.....	45

# Capítulo 1

## Introducción

### 1.1 Motivación

La producción ecológica agrícola está aumentando debido a la demanda de los consumidores concienciados con el medio ambiente. Uno de los factores más relevantes de este tipo de agricultura es que es sostenible. La agricultura ecológica genera productos de la mejor calidad, debido a que se respeta la estacionalidad y distribución de las especies vegetales a producir.

El cuaderno de campo permite cumplir con la normativa europea, nacional y autonómica que requieren el registro de todas las operaciones que se realicen en los cultivos. Por ello, crearemos un cuaderno de campo adaptado a la agricultura ecológica en Canarias, que nos permita registrar todos los datos necesarios para cumplir con la normativa que regula dicho tipo de producción agrícola. Será una aplicación web para trabajar desde dispositivos de escritorio, y una aplicación móvil multiplataforma para trabajar desde el propio campo de cultivo.

En la actualidad, existen múltiples herramientas, tanto web como para dispositivos móviles, que permiten la gestión de un cuaderno de campo. Entre esas herramientas nos encontramos con Agroptima, Toolagro y Agricolum. Las dos primeras no son simplemente un cuaderno de campo, las podríamos considerar como ERP (aplicación de planificación de recursos empresariales). Por esta cuestión, están fuera del ámbito del proyecto, que solo desea la gestión del cuaderno de campo. En cuanto a Agricolum, si que es un cuaderno de campo, pero pensada para la agricultura convencional. Ningún software de los existentes hasta el momento se adaptaba a las necesidades de Canarias. Están pensados para las grandes extensiones de las zonas continentales y no para pequeños terrenos, como los que se cosechan en las islas.

Por tanto, ninguna de las aplicaciones analizadas cumple con las necesidades de los cuadernos de campo para la agricultura ecológica.

## 1.2 Objetivos

El objetivo de este proyecto, es el desarrollo de una aplicación que cubra las necesidades reales de los productores de la agricultura ecológica de Canarias, en los referido al cuaderno de campo. La aplicación permitirá que los inspectores que deben certificar las producciones puedan acceder a los datos, para comprobar que se cumplen los requisitos que exige la legislación actual.

La aplicación contará con una versión web, para trabajar desde dispositivos de escritorio, y que también se adapta perfectamente a las tabletas. Y con una versión, para dispositivos móviles y tabletas, para trabajar desde cualquier lugar. La interfaz debe ser intuitiva y fácil de utilizar, para reducir los tiempos de aprendizaje de utilización.

Debido a que se debe mantener el registro de toda la información durante varios años, la mayoría de los datos contarán con la posibilidad de ser activados o desactivados, por defecto, estarán activados. Esto permitirá que se muestren o no los datos, para proporcionar una interfaz más limpia, sólo con los datos que el usuario utiliza.

Finalmente, el nombre de la aplicación será higopico. La razón de elegir este nombre es sencilla, en Canarias existe una planta que se puede encontrar en barrancos y fincas desatendidas. Estás plantas, sin que nadie las cuide, dan todos los años sus frutos. Dicha planta es la tunera, y su fruto es el “higo pico”. No existe un producto más ecológico que este en Canarias.

### 1.2.1 Aplicación web

La versión web, permitirá a los usuarios trabajar desde dispositivos de escritorio y tabletas. Se mostrará por pantalla la mayor cantidad de información posible.

### 1.2.2 Aplicación móvil

La aplicación móvil, que funcionará tanto en teléfonos inteligentes como en tabletas, permitirá al productor trabajar desde cualquier lugar. Esta versión, mostrará menos información que la versión web, en lo que a las listas de datos se refiere, debido a las limitaciones de las dimensiones de la pantalla de dichos dispositivos.

## 1.3 Estructura del documento

El documento está estructurado siguiendo el siguiente esquema:

- Capítulo 1: Introducción

Se indican las motivaciones y objetivos del proyecto.

- Capítulo 2: Estado del arte  
Estudio de la situación actual de los cuadernos de campos para la agricultura ecológica y análisis de las distintas aplicaciones que existen en el mercado.
- Capítulo 3: Preliminares  
Explicación del vocabulario utilizado. Definición, funcionamiento y distribución de la aplicación.
- Capítulo 4: Tecnologías utilizadas  
En este capítulo se indican todas la herramientas y librerías utilizadas durante el desarrollo de la aplicación.
- Capítulo 5: Base de datos  
Explicación de la estructura, relaciones y restricciones de la base de datos. Se incluyen indicaciones de como obtener las credenciales de acceso en Firebase.
- Capítulo 6: Backend  
Definición de todas las partes que componen el backend. Se explican las medidas de seguridad implementadas.
- Capítulo 7: Interfaz web y móvil  
En este capítulo se pueden observar las distintas partes de la interfaz de usuario, tanto para web como para móviles.
- Capítulo 8: Presupuesto  
Estimación de coste que supone el desarrollo de este proyecto.
- Capítulo 9: Conclusiones y líneas futuras
- Capítulo 10: Conclusions and future lines

# Capítulo 2

## Estado del arte

La legislación europea, nacional y autonómica exige a los productores agrarios el registro de muchas de la operaciones que realizan en sus campos, y para facilitar la labor de registrar todos estos datos, existen los cuadernos de campo o de explotación (ver figura 2.1). Estos datos varían en función del tipo de agricultura que se realice en la explotación [9].



CUADERNO DE EXPLOTACION



---

**HOJA DE PRÁCTICAS DE CULTIVO** D.G.

IDF. NÚMERO DE IDENTIFICACIÓN DEL TERRENO	Práctica de cultivo			
	Cultivo	PRÁCTICA / LABOR REALIZADA (1)	Fecha de realización	Observaciones

Figura 2.1: Fragmento de cuaderno de campo del Gobierno de Canarias

A continuación veremos los tipos de agricultura y analizaremos algunas de las aplicaciones de cuadernos de campo digital que actualmente existen.

### 2.1 Tipos de agricultura

Una de las principales diferencias, a nivel legislativo, es que las denominaciones ecológica e integrada están reservadas y requieren del cumplimiento estricto de normativas que se deben acreditar con los certificados correspondientes [4][14].

### 2.1.1 Convencional

Este tipo se utiliza para obtener la mayor cantidad de producto posible. Se utilizan los agroquímicos que sean necesarios, para maximizar la producción y minimizar el espacio utilizado.

### 2.1.2 Ecológica

Respetar las características ecológicas del lugar y del suelo, también la estacionalidad y distribución de las especies vegetales a producir. Sólo se permite el uso de productos naturales para fertilizar la tierra. Con esta práctica se obtienen productos de la máxima calidad y de manera sostenible.

### 2.1.3 Integrada

Está a medio camino entre la convencional y la ecológica. Se permite el uso de agroquímicos, pero de manera restringida y minimizando su uso, permitiendo sólo algunas sustancias, las cuales dependen del tipo de cultivo. Se obtienen productos de buena calidad y de manera sostenible.

## 2.2 Aplicaciones existentes

Las explotaciones agrarias en España son un sector importante y por ello existen muchas empresas tecnológicas trabajando en herramientas para dicho sector.

### 2.2.1 Agroptima



Figura 2.2: Agroptima

Una de las principales herramientas es Agroptima, que va más allá de lo que es un cuaderno de campo, pues añade muchas funcionalidades extras como: una base de datos de fitosanitarios y semillas, control de stock, control de costes e ingresos, etc. Estas funcionalidades sobrecargan la interfaz de usuario con muchos elementos que no son necesarios, impidiendo a los usuarios, que solo quieren un cuaderno de campo, utilizar la aplicación de manera sencilla y eficaz [5].

Además, Agroptima está pensada para la agricultura convencional y las grandes extensiones de la España peninsular. Por ello, no es una herramienta que se adapte a las necesidades que quiere cubrir este proyecto.

### 2.2.2 Toolagro



Figura 2.3: Toolagro

Es una aplicación de planificación de recursos empresariales (ERP) orientado a las empresas del sector agrícola, desde la producción hasta la comercialización del producto. Incluye un cuaderno de campo multiusuario que puede ser gestionado mediante aplicación web y móvil [11].

Al igual que Agroptima, estamos ante una aplicación que sobrepasa las prestaciones que queremos para un cuaderno de campo para los productores ecológicos de Canarias.

### 2.2.3 Agricolum

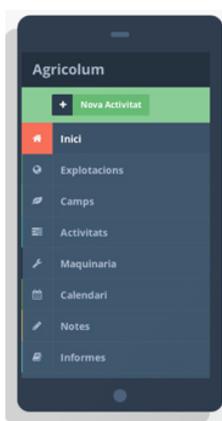


Figura 2.4: Agricolum

Agricolum si es un cuaderno de campo digital [3]. Tienen disponibles varios pack, en función de las necesidades del productor. Disponen de un pack gratuito que permite tener dos explotaciones con un máximo de 20 actividades por mes [2].

El problema de esta aplicación es que está pensada para la agricultura convencional.

### 2.2.4 Conclusión del estudio de las aplicaciones existentes

Ninguna de las aplicaciones vistas cumplen con las necesidades de un productor de la agricultura ecológica en lo relacionado al cuaderno de campo. Estas aplicaciones no están diseñadas para este grupo de agricultores. Tampoco cumplen las necesidades de los productores canarios, pues las extensiones de los cultivos son mucho más pequeñas y las herramientas utilizadas difieren de las usadas en la España peninsular.

# Capítulo 3

## Preliminares

### 3.1 Vocabulario

En el proyecto se utilizarán diversos términos, que no tienen el significado normal, por ello serán definidos ahora:

- **Productor:** persona o empresa que genera productos de agricultura ecológica.
- **Plantación:** extensión de tierra donde tiene lugar la producción.
- **Parcela:** extensión de tierra que forman las plantaciones. Estas tienen un número de polígono y de parcela que viene dada por la división horizontal del terreno realizada por las autoridades competentes. Pueden ser invernaderos.
- **Recintos:** Divisiones realizadas a las parcelas por el productor, para tener una mejor organización y control de todos los que se produce en esa parte del terreno. Un recinto puede contener distintos cultivos.
- **Producción:** Cultivos plantados en los recintos, en los cuales se realizarán las tareas diarias de los agricultores: abonado, tratamientos y otras labores.

### 3.2 Definición

El proyecto, como fue definido en la introducción, es un cuaderno de campo donde los productores registran todas las actividades realizadas en sus plantaciones.

El cuaderno de campo será revisado por las entidades certificadoras, para otorgarles o no el sello de productor ecológico. Por tanto, como existirán distintos tipos de usuarios que accederán a los datos, se ha creado un sistema de roles. Los usuarios sólo podrán tener uno siguientes roles:

- **Administrador:** posee acceso para la creación, modificación y eliminación de los usuarios. No tiene acceso a ninguna otra información de la base de datos.
- **Productor:** es el encargado de introducir todos los datos. Puede crear, modificar y eliminar cualquier dato, a excepción de los datos de usuarios, a los que no tiene

acceso.

- **Inspector:** sólo tiene acceso de lectura a la información de las producciones.

La aplicación estará desarrollada por capas, por los que contará con una base de datos en Firebase (Firestore) de Google, pero que debido a que se utilizará su API para Java, no habrá acceso a las Function de Firestore.

El backend realizará el control de acceso de los usuario mediante tokens de autenticación con un vencimiento y cifrado a definir en la configuración a la hora de realizar el despliegue. También autorizará o denegará el acceso de los usuario a los datos dependiendo de su rol dentro de la aplicación. Debido a que no se tiene acceso a las Functions de Firebase, se desarrollarán en backend el control de restricciones y de integridad de los datos.

Por último, tanto la aplicación web como la aplicación para dispositivos móviles tendrán las mismas funcionalidades, las cuales se detallan en la siguiente tabla, indicando que rol es necesario para tener acceso a ellas:

<b>Funcionalidad</b>	<b>Rol</b>
Iniciar y cerrar sesión	Todos
Consultar información del usuario actual	Todos
Modificar contraseña del usuario actual	Todos
Consultar información del productor	Inspector y productor
Modificar información del productor	Productor
Consultar cultivos, documentos, fertilizantes, fitosanitarios, labores, municipios, parcelas, plantaciones, producciones, proveedores, provincias, semillas y recintos	Inspector y productor
Crear, modificar y eliminar cultivos, documentos, fertilizantes, fitosanitarios, labores, municipios, parcelas, plantaciones, producciones, proveedores, provincias, semillas y recintos	Productor
Exportar información a Microsoft Excel	Inspector y productor
Consultar, crear, modificar y eliminar usuarios	Administrador

**Tabla 3.1:** Funcionalidades de la aplicación

### 3.3 Funcionamiento de la aplicación

Las plantaciones se dividen en parcelas, que a su vez se dividen en recintos. Dentro de los recintos se realiza la producción de distintos cultivos. En las plantaciones se puede indicar la procedencia del agua de riego. Las parcelas tendrán que indicar a qué municipio pertenecen, y opcionalmente, se indicará el polígono al que pertenece y su número de parcela.

Una producción se realiza en una campaña específica y puede utilizar una semilla determinada. Se podrán indicar las fechas de siembra y de recolección. Los trabajadores de la productora realizarán abonados, tratamientos y otras labores a estas producciones. Estas operaciones quedarán registradas con la fecha y un campo de observaciones opcional.

Para poder utilizar semillas y fitosanitarios en las producciones, se debe conocer su procedencia, para poder certificar que es ecológico. Para ello, se indicará el documento que identifica dicha procedencia. En estos documentos debe constar el proveedor al que pertenece.

### 3.4 Distribución

La comunicación entre las diferentes capas que forman el sistema la podemos ver en la Figura 3.1, donde se observan 3 componentes: el servidor de base de datos, el servidor de aplicaciones y los terminales de los usuarios (ordenadores, tabletas y teléfonos inteligentes).

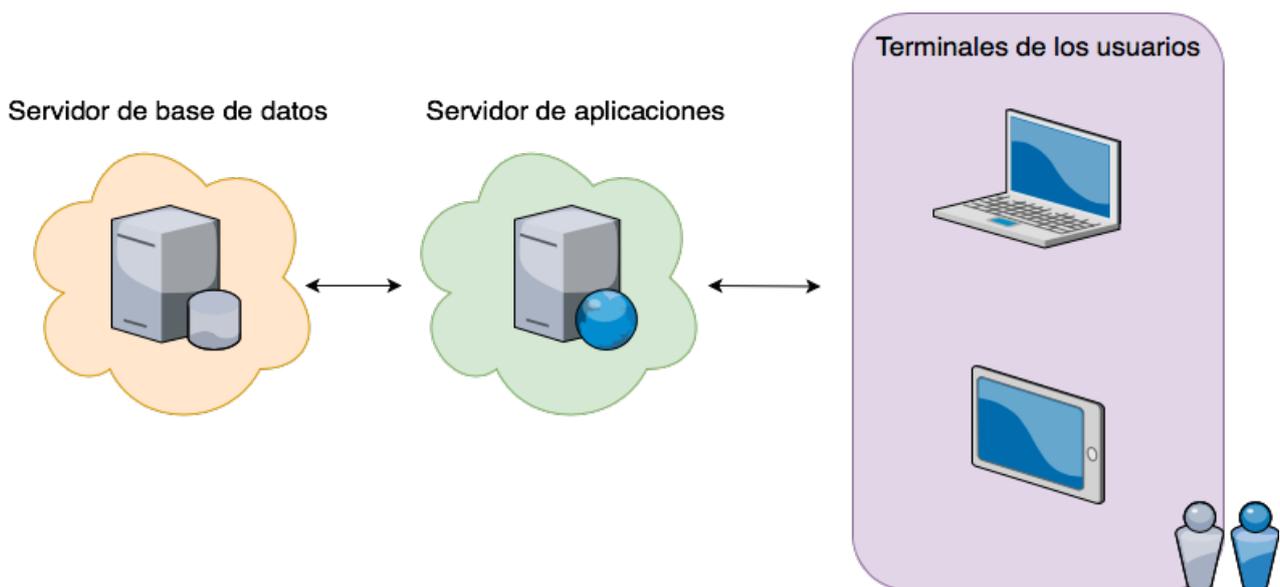


Figura 3.1: Estructura del sistema

El servidor de la base de datos sólo es accesible por el servidor de aplicaciones, debido a que la parte backend de nuestra aplicación, que se ejecuta en el servidor de aplicaciones, es el encargado de controlar todas las peticiones que realiza el usuario, y si fuese necesario, realizar las consultas a la base de datos que sean oportunas. Esto añade seguridad a nuestros datos, pues el usuario final no tiene acceso a la base de datos de manera directa.

# Capítulo 4

## Tecnologías utilizadas

Las tecnologías utilizadas dependen de cada fase del desarrollo de la aplicación, por ello, se expondrán en diferentes apartados. Dichos apartados son:

1. Base de datos
2. Backend
3. Frontend web
4. Frontend para dispositivos móviles y tabletas
5. Otros

### 4.1 Base de datos

- **Draw.io:**

Aplicación web que permite diseñar diferentes tipos de diagramas. En este caso, ha sido utilizada para el diseño de el modelo entidad-relación de la base de datos.

- **MySQL Workbench:**

Herramienta unificada para desarrolladores de bases de datos. Esta herramienta está desarrollada por ORACLE. Permite diseñar, desarrollar, administrar y migrar bases de datos. Solamente funciona con base de datos MySQL. En cambio, para el diseño del modelo lógico relacional, sirve para diferentes gestores de bases de datos de tipo SQL, para ese motivo ha sido utilizada.

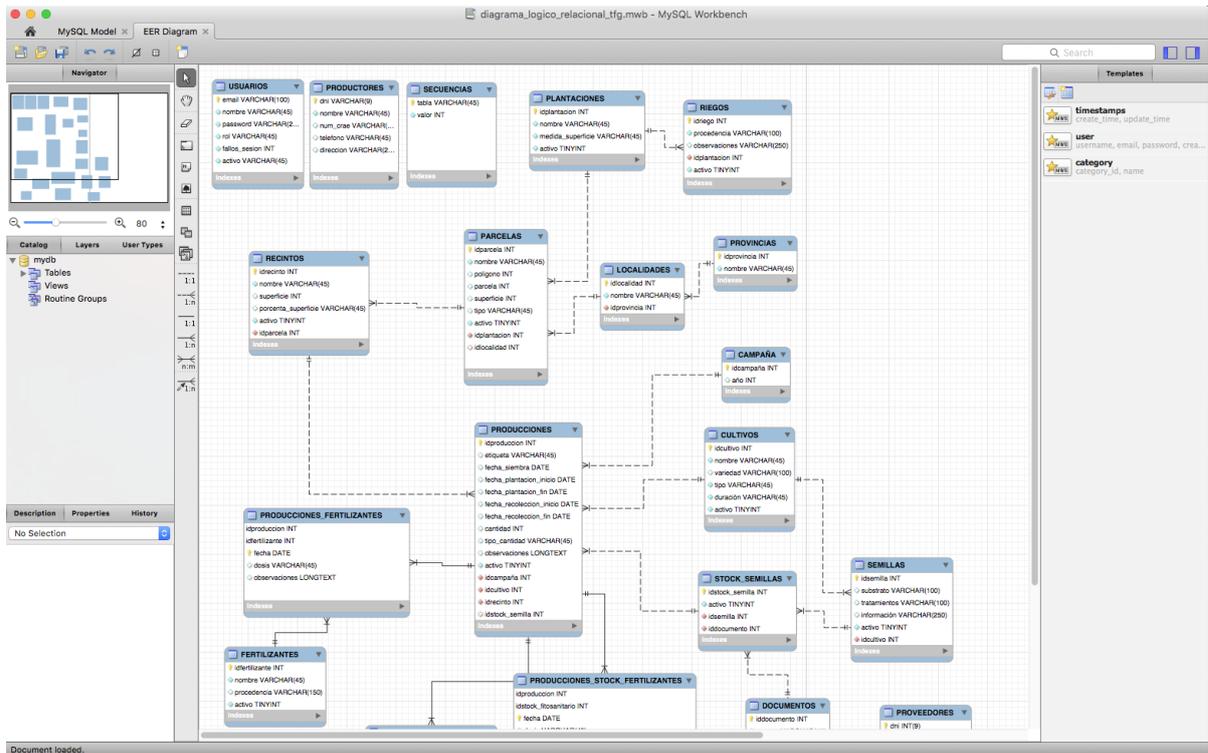


Figura 4.1: MySQL Workbench

- **Cloud Firestore:**

Base de datos noSQL en la nube, que pertenece a al grupo de herramientas de Firebase de Google. Esta base de datos se actualiza en tiempo real, por tanto, si es utilizada por varios usuarios, las inserciones, modificaciones y eliminaciones de los datos son visibles casi automáticamente. Cuenta con una versión gratuita, que es la utilizada en el proyecto. Esta versión está limitada a 1 Gbyte de espacio de almacenamiento y, un límite de escrituras y lecturas diarias.

Google proporciona la API para diferentes lenguajes de programación. La versión para java, no puede hacer uso de las Cloud Functions, que le dan mayor potencia a la base de datos.

## 4.2 Backend

- **Java:**

Uno de los lenguajes de programación más populares. Java permite realizar desarrollos para múltiples dispositivos, pues es independiente de la plataforma.

- **Maven:**

Herramienta de software para crear y gestionar proyectos de Java. Se configura en el proyecto utilizando un fichero XML (pom.xml). Maven se encarga de descargar e

instalar todas dependencias indicadas en el fichero de configuración. También permite automatizar tareas, como por ejemplo, crear el ejecutable de la aplicación.

- **Spring:**

Conjunto de frameworks para el desarrollo de aplicaciones, para cualquier plataforma. Entre otras muchas características incluye utilidades para el desarrollo web, servlet, multihilo, acceso a datos, REST [15].

- **Spring Boot:**

Herramienta que facilita la integración Spring en el desarrollo de aplicaciones. Esta herramienta se encarga de autoconfigurar todos los aspectos de nuestra aplicación, sin casi definir nada. Resuelve todas las dependencias, utilizando, en este caso, Maven [16].

- **Spring WebFlux:**

Forma parte de la extensas utilidades de Spring. Provee a la aplicación del paradigma de programación reactiva, con lo que se acelerarán las peticiones REST [17].

- **Spring Cloud GCP Firestore Module:**

Módulo de Spring para realizar las comunicaciones con las base de datos en Cloud FireStore.

- **Librería de JSON Web Tokens para Java:**

Genera tokens cifrados de autenticación de usuarios utilizando los distintos algoritmos de cifrado que incorpora [6].

- **Apache POI:**

API para Java que permite la creación de documentos de Microsoft Office. En el proyecto será utilizada para la generación de un documentos de Excel [13].

- **Spring Tool Suite 4 for Eclipse:**

IDE para el desarrollo en java de aplicaciones utilizando Spring Boot. Realmente se trata de una modificación de la IDE Eclipse adaptada a Spring Boot.

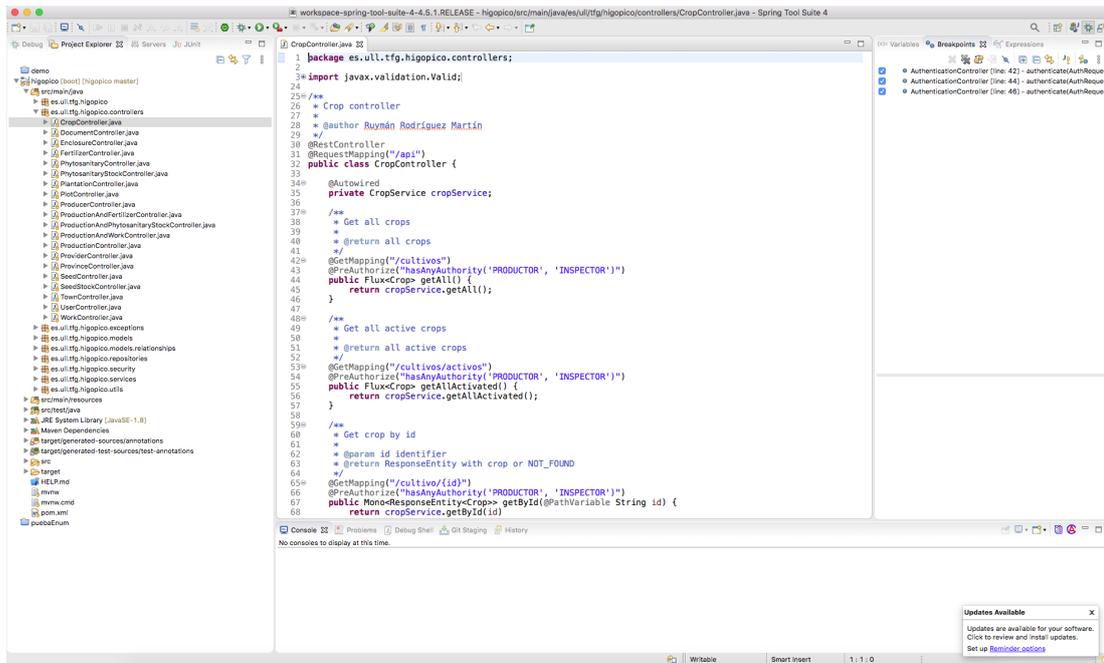


Figura 4.2: Spring Tool Suite 4 for Eclipse

## 4.3 Frontend web

- **JavaScript:**

El lenguaje de programación más utilizado para el desarrollo de páginas web. Se podría afirmar que no existe página web que no lo utilice. Es un lenguaje interpretado, por lo que no requiere ser compilado.

- **Node.js:**

Entorno de ejecución multiplataforma para la capa del servidor, basado en JavaScript.

- **npm:**

Gestor de paquete por defecto para Node.js

- **React:**

Biblioteca de JavaScript creada para el desarrollo de interfaces de usuario. Es declarativo y está basado en componentes. Estos componentes compondrán las interfaces de usuario. Es un proyecto de código abierto desarrollado y mantenido por Facebook [7].

- **JSX:**

Extensión de la sintaxis de JavaScript utilizado por React, para el desarrollo de elementos de la interfaz.

- **Atom:**

Editor de texto para el desarrollo de aplicaciones. Permite la instalación de paquetes que le añaden funcionalidades extra. Tiene paquetes para desarrollo en JavaScript y React.js.

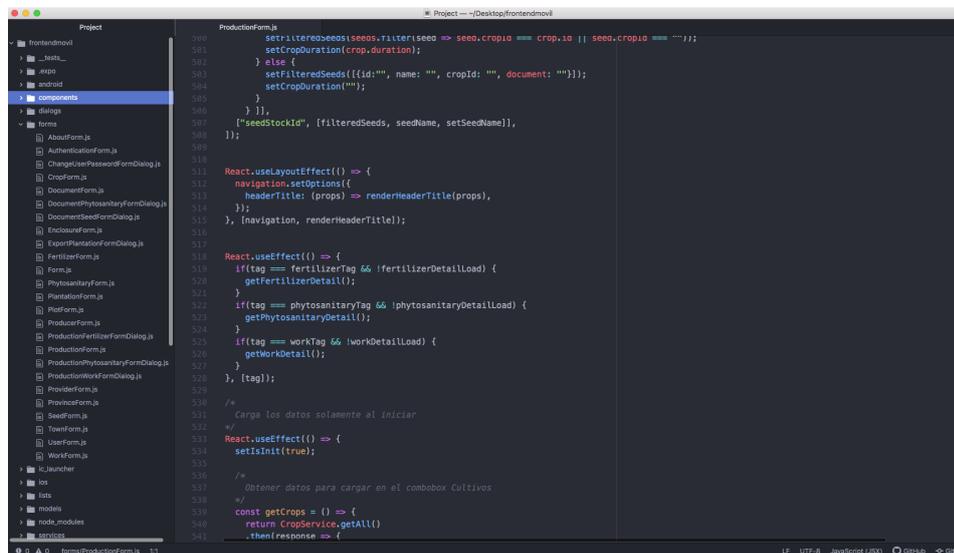


Figura 4.3: Atom

- **Material-UI:**

Paquete de componentes para React.js que utilizan la normativa de diseño Material Design. Se instala mediante el gestor de paquetes npm [12].

- **Color Tool de Material Design:**

Herramienta online en la que indicamos los colores que utilizará nuestra aplicación y nos muestra como deben ser utilizados estos colores en los distintos elementos de la interfaz.

## 4.4 Frontend para dispositivos móviles y tabletas

En el desarrollo para dispositivos móviles y tabletas se usan muchas de las tecnologías utilizadas para el desarrollo web: JavaScript, node.js, npm, JSX, Atom y Color Tool de Material Design. A parte de esas, se utilizan:

- **React Native:**

Variante de React.js pensada para la creación de interfaces de usuario nativas de distintos sistemas operativos. Entre ellos se incluyen Android, iOS y Microsoft Windows [8].

- **Expo:**

Plataforma de código abierto, para la creación de aplicaciones nativas utilizando React Native. Esta provista de un utilidad de gran ayuda para el desarrollo, una aplicación para los distintos dispositivos móviles, que nos permite probar la aplicación en desarrollo sin tener que generar el ejecutable e instalarlo [1].

## 4.5 Otros

- **Github:**

Plataforma de desarrollo colaborativo en la nube, para alojar proyectos. Utiliza la herramienta de control de versiones Git.

# Capítulo 5

## Base de datos

### 5.1 Entidades y atributos

La base de datos esta compuestas por todas las entidades necesarias que se identificaron y están relacionadas entre sí. Estas entidades normalmente estarán representadas en la base de datos por una tabla.

Debido a que en las empresas los datos son muy valiosos y no se deben perder, se ha añadido el atributo 'activo' a la mayoría de las entidades. Este atributo, que por defecto toma valor verdadero, permite ocultar y que no sea accesible el elemento, pues ya no está disponible o ya no se utiliza. Con ello mantendremos una interfaz de usuario limpia, sin información innecesaria. Todas las entidades poseen el atributo id, que los identifica inequívocamente.

A continuación se muestran las entidades creadas, ordenadas por orden alfabético:

- **Cultivo:**

Representa un cultivo.

Atributos:

- Nombre: nombre del cultivo. No puede ser nulo.
- Variedad: variedad del cultivo.
- Tipo: a seleccionar entre aromática, cereal, fruta y hortaliza. No puede ser nulo.
- Duración: a seleccionar entre anual, bienal y perenne. No puede ser nulo.
- Activo

- **Documento:**

Representa a un documento de tipo albarán o factura.

Atributos:

- Número: número del documento. No puede ser nulo.
- Fecha: fecha de emisión del documento. No puede ser nulo.
- Tipo: a seleccionar entre albarán o factura. No puede ser nulo.
- Activo
- **Fertilizante:**  
Representa un fertilizante.  
Atributos:
  - Nombre: nombre del fertilizante. No puede ser nulo. Es único.
  - Origen: procedencia del fertilizante.
  - Activo
- **Fitosanitario:**  
Representa un fitosanitario.  
Atributos:
  - Nombre: nombre del fitosanitario. No puede ser nulo. Es único.
  - Activo
- **Labor:**  
Representa un trabajo realizado por el agricultor.  
Atributos:
  - Nombre: nombre de la labor. No puede ser nulo. Es único.
  - Activo
- **Municipio:**  
Representa un municipio.  
Atributos:
  - Nombre: nombre del municipio. No puede ser nulo. Es único.
- **Parcela:**  
Representa una parcela, puede estar al aire libre o ser un invernadero.  
Atributos:

- Nombre: nombre de la parcela. No puede ser nulo.
- Polígono: número de polígono de la división horizontal del territorio donde se encuentra la parcela.
- Parcela: número de parcela del polígono.
- Superficie: dimensión de la parcela.
- Tipo: a seleccionar entre aire libre e invernadero. No puede ser nulo.
- Activo

- **Plantación:**

Representa una plantación.

Atributos:

- Nombre: nombre de la plantación. No puede ser nulo. Es único.
- Medida: unidad de medida de la superficie utilizada en esta plantación. A seleccionar entre m<sup>2</sup> y ha. No puede ser nulo.
- Riego\_Origen: indica la procedencia del agua de riego.
- Riego\_Observaciones: observaciones del agua de riego.
- Activo

- **Producción:**

Representa una producción.

Atributos:

- Nombre: nombre de la plantación. No puede ser nulo.
- Cantidad: cantidad plantada.
- Tipo\_Cantidad: a seleccionar entre superficie o plantas.
- Fecha\_Siembra: fecha de siembra de cultivos de tipo perenne.
- Fecha\_Plantación: atributo compuesto. Indica la fecha de inicio y fin de la siembra de cultivos no perennes.
- Fecha\_Recolección: atributo compuesto. Indica la fecha de inicio y fin de la recolección.
- Año: año de la campaña en la que se inició la producción. No puede ser nulo.
- Observaciones: observaciones sobre la producción.

- Activo

- **Productor:**

Representa al productor.

Atributos:

- Nombre: nombre de la persona física o empresa. No puede ser nulo. Es único.
- Dirección: dirección del productor.
- Teléfono: teléfonos del productor.
- Dni: número del documento de identificación del productor (DNI, CIF, ...). No puede ser nulo. Es único.
- Num\_Crae: número de registro de operadores de producción ecológica de Canarias. No puede ser nulo. Es único.

- **Proveedor:**

Representa un proveedor.

Atributos:

- Nombre: nombre del proveedor. No puede ser nulo. Es único.
- Dirección: dirección del proveedor.
- Teléfono: teléfonos del proveedor.
- Dni: número del documento de identificación del proveedor (DNI, CIF, ...). No puede ser nulo. Es único.
- Activo

- **Provincia:**

Representa una provincia.

Atributos:

- Nombre: nombre de la provincia. No puede ser nulo. Es único.

- **Recinto:**

Representa un recinto

Atributos:

- Nombre: nombre del recinto. No puede ser nulo.

- Superficie: dimensión del recinto.
  - Porcentaje\_Superficie: porcentaje de la superficie de la parcela que ocupa. Es un campo calculado.
  - Activo
- **Secuencias:**  
Representa la secuencia actual de un identificador de una entidad.  
Atributos:
    - Tabla: nombre de la tabla a la que pertenece la secuencia. Es la clave primaria.
    - Valor: último valor del identificador insertado en la tabla.
- **Semilla:**  
Representa una semilla.  
Atributos:
    - Nombre: nombre de la semilla. No puede ser nulo. Es único.
    - Sustrato: sustrato de la semilla.
    - Tratamientos: tratamientos dados a la semilla.
    - Información: otra información sobre la semilla.
    - Activo
- **Stock de fitosanitarios:**  
Representa un fitosanitario que se encuentra disponible para ser aplicado en una producción.  
Atributos:
    - Lote: lote de procedencia del fitosanitario en stock.
    - Activo
- **Stock de semillas:**  
Representa una semilla que se encuentra disponible para ser utilizada en una producción.  
Atributos:
    - Activo

- **Usuario:**

Representa un usuario de la aplicación.

Atributos:

- Nombre: nombre del usuario. No puede ser nulo. Es único.
- Email: correo electrónico del usuario. No puede ser nulo. Es único.
- Contraseña: contraseña del usuario. No puede ser nulo. Tamaño mínimo de 8 caracteres.
- Fallos\_Sesión: contador de intentos fallidos de inicio de sesión.
- Rol: rol del usuario dentro de la aplicación. A seleccionar entre administrador, inspector o productor. No puede ser nulo.
- Activo: aparte de mostrar/ocultar al usuario, si está indicado como falso, el usuario no podrá iniciar sesión.

## 5.2 Relaciones entre entidades

La base de datos, en este caso, es relacional, existe una relación entre casi todos los datos que se introducen. Por ello se definen las siguientes relaciones, para que los datos tengan coherencia:

- **Cultivo – Producción:**

Una producción sólo puede ser de un cultivo en concreto. Un cultivo puede estar en distintas producciones.

- **Cultivo – Semilla:**

Una semilla sólo puede ser de un cultivo. Un cultivo puede tener muchas semillas.

- **Documento – Proveedor:**

Un documento sólo puede pertenecer a un productor, pero un productor puede emitir muchos documentos.

- **Documento – Stock\_Fitosanitarios:**

Un documento puede tener muchos stock\_fitosanitarios. Un stock\_fitosanitario sólo puede pertenecer a un documento.

- **Documento – Stock\_Semillas:**

Un documento puede tener muchos stock\_semillas. Un stock\_semillas sólo puede

pertenecer a un documento.

- **Fertilizante – Producción:**

Un fertilizante puede ser aplicado muchas veces en una producción. En una producción pueden ser aplicados múltiples fertilizantes.

- **Fitosanitario – Stock\_Fitosanitarios:**

Un fitosanitario puede estar en muchos stock\_fitosanitarios. Un stock\_fitosanitarios sólo puede tener un fitosanitario asignado.

- **Labor – Producción:**

Una labor puede ser realizada muchas veces en una producción. En una producción pueden ser realizadas múltiples labores diferentes.

- **Municipio – Parcela:**

Un municipio puede tener muchas parcelas, pero una parcela sólo puede pertenecer a un municipio.

- **Municipio – Provincia:**

Un municipio sólo puede pertenecer a una provincia. Una provincia está compuesta de muchos municipios.

- **Parcela – Plantación:**

Una parcela sólo puede pertenecer a una plantación. Las plantaciones pueden tener muchas parcelas.

- **Parcela – Recinto:**

Una parcela puede tener muchos recintos, pero un recinto sólo puede pertenecer a una parcela.

- **Producción – Recinto:**

Una producción se realiza en un recinto en concreto. En un recinto pueden existir muchas producciones.

- **Producción – Stock\_Fitosanitarios:**

A una producción se le puede aplicar muchos fitosanitarios en stock. Un fitosanitario en stock puede ser aplicado a múltiples producciones.

- **Producción – Stock\_Semillas:**

Una producción puede tener asignada una semilla en stock. Una semilla en stock puede ser utilizada en muchas producciones.

- **Semilla – Stock\_Semillas:**

Una semilla puede estar en varios stock de semillas, pero un stock de semillas sólo tiene asignada una semilla.

## 5.3 Restricciones

Adicionalmente, se definen unas restricciones que no se han indicado en los apartados anteriores. Estas se deben cumplir siempre, para garantizar la integridad de los datos:

- En los atributos en los que indicamos que se debe seleccionar un valor dado, pues tienen un dominio, se debe comprobar que el valor del atributo está entre esos valores.
- No puede existir más de una combinación de nombre y variedad en los cultivos.
- No pueden existir dos documentos con el mismo número, fecha, proveedor y tipo.
- No puede existir el mismo nombre de parcela en la misma plantación.
- No puede existir el mismo nombre de recinto en la misma parcela.
- No puede existir más de un Stock\_Fitosanitarios que haga referencia al mismo fitosanitario y mismo documento.
- No puede existir más de un Stock\_Semillas que haga referencia a la misma semilla y mismo documento.

## 5.4 Actualización y eliminación de datos

La actualización de datos que tienen relación con otros, se pondrá realizar en cascada. Debido a que Cloud Firestore no es una base de datos SQL, no se indicará nada, pero se debe tener en cuenta para futuras adaptaciones.

Las eliminaciones de datos, se controlan desde el backend, puesto que Cloud Firestore no da acceso a las Cloud Functions desde java, no estarán permitidas si el dato que se desea eliminar es referenciado por otro. Para poder eliminarlo, se deberá quitar la referencia manualmente y luego eliminar el dato.

## 5.5 Diagrama entidad-relación

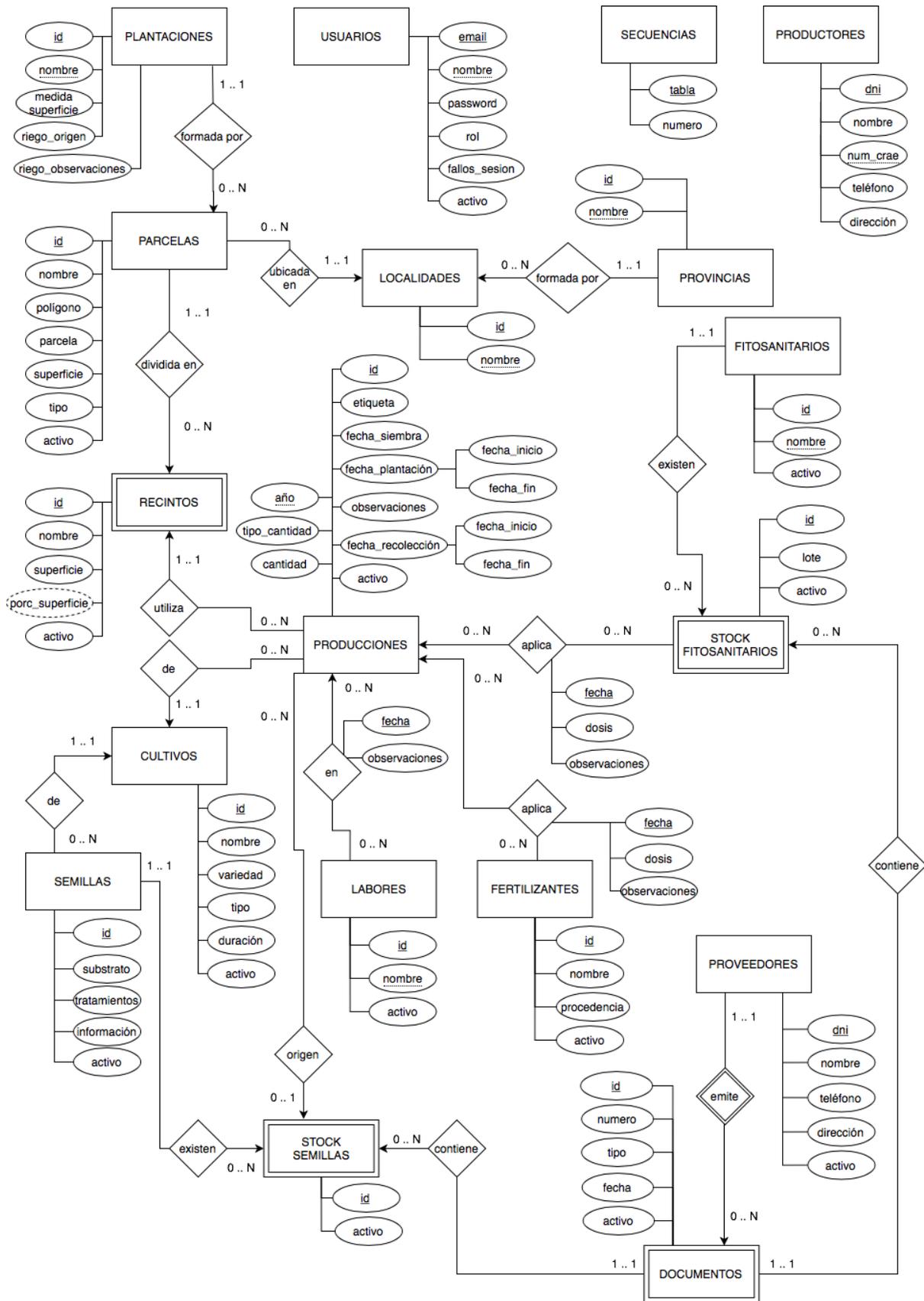


Figura 5.1: Modelo entidad-relación

## 5.6 Diagrama del modelo lógico relacional

En este diagrama, se aprecian todas las tablas y sus atributos. También se puede observar la existencia de tablas que son creadas para permitir las relaciones entre distintas entidades.

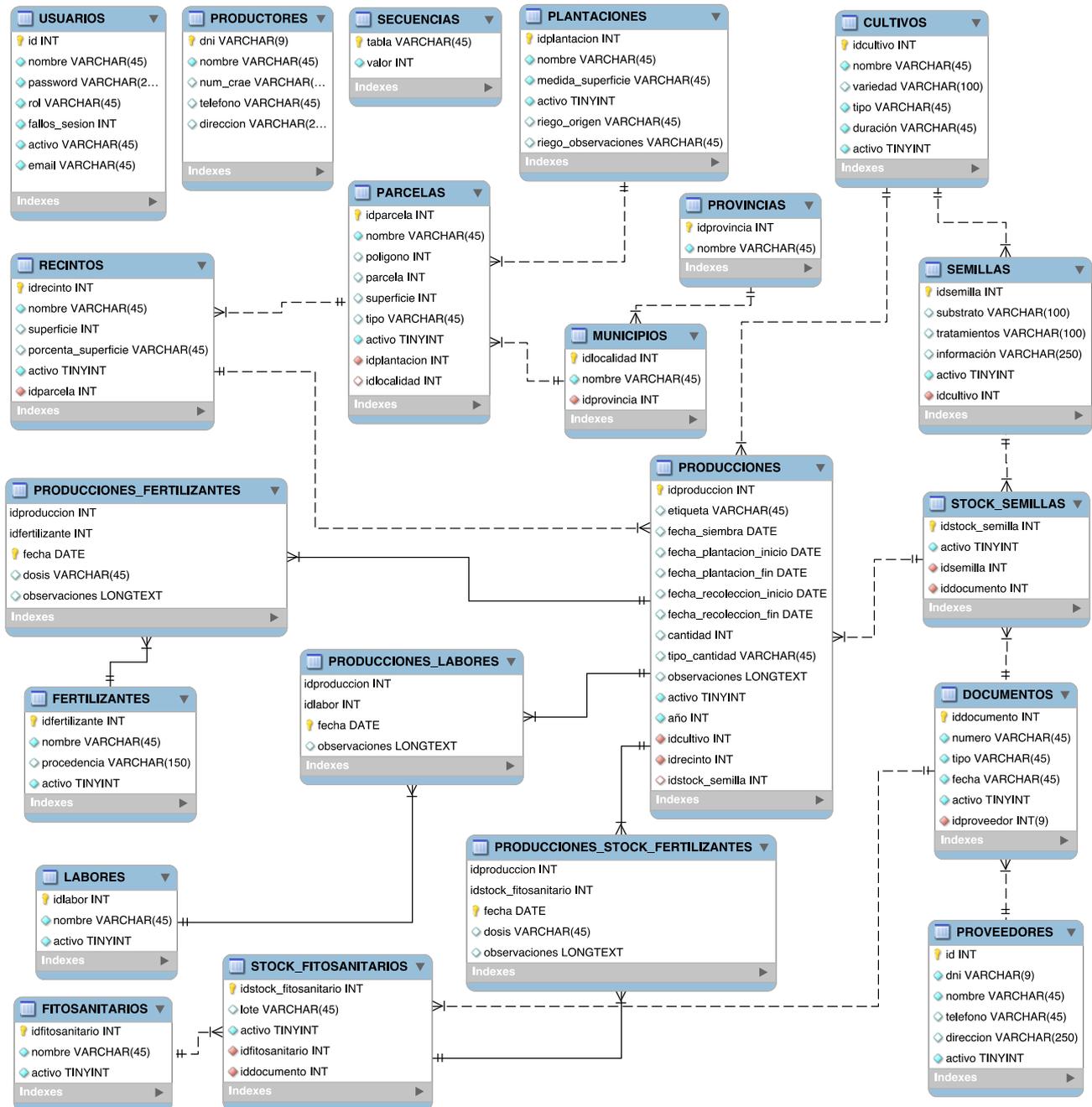


Figura 5.2: Modelo lógico relacional

## 5.7 Obtención de credenciales

La obtención de las credenciales para el acceso a la base de datos desde el backend de la aplicación es muy importante, sin ellas, no se podrá tener acceso a los datos.

Para obtenerlas hay que crear el proyecto dentro de Firebase. Dentro de este proyecto se generará la base de datos de la aplicación. Una vez se haya finalizado el proceso de creación, hay que dirigirse al apartado cuentas de servicio (ver figura 5.3), dentro de las configuraciones del proyecto. Por último, se indica la plataforma para la que se va a generar y se hace click en el botón generar nueva clave privada. Se descargará un fichero, ese fichero son las credenciales. Este fichero debe estar siempre a buen recaudo, pues cualquier persona que tenga acceso a él tendrá acceso a los datos como administrador.

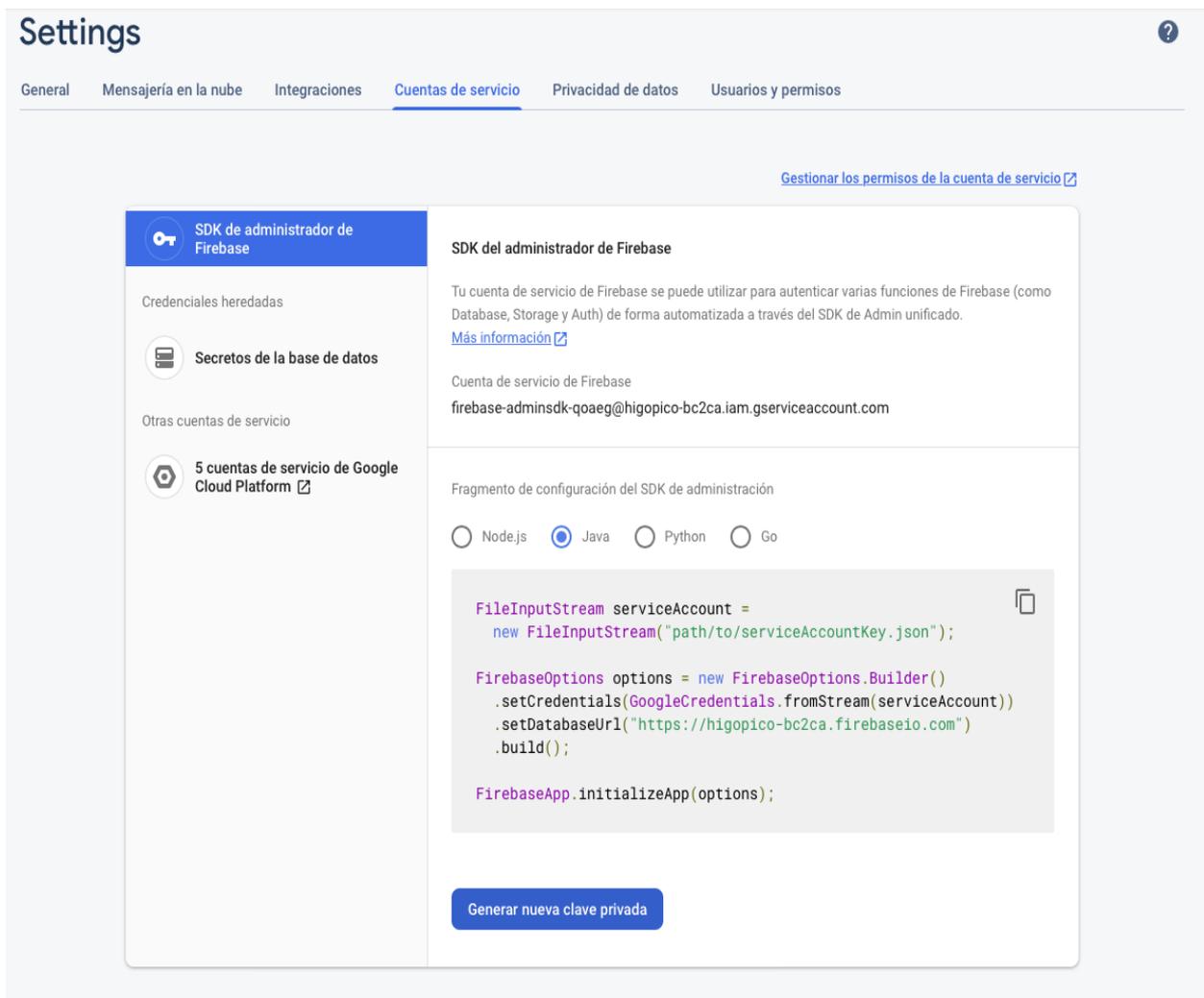


Figura 5.3: Obtención de credenciales

# Capítulo 6

## Backend

En esta capa de la aplicación, se han desarrollado todos los controladores necesarios para proveer a la interfaz web de una API funcional y segura. Sin tener en cuenta esto, que es el propósito general de un backend, se ha tenido que desarrollar todo el control de integridad de los datos, cosa que normalmente realiza el gestor de bases de datos de manera transparente.

### 6.1 Control de integridad de los datos

El control de integridad se realiza justo antes de cada inserción, actualización o eliminación. Este control lo realizan los servicios dentro del backend (ver figura 6.1), y en caso de encontrar un problema que impidan llevar a cabo la operación indicada por el usuario, devuelve un error, que dependerá del tipo de restricción que se haya incumplido.

Se han desarrollado los siguientes tipos de restricciones:

- Único: no puede existir más de un elemento en la misma tabla con el mismo valor
- Clave: comprueba que no exista un elemento con la misma clave a la hora de insertar, pues si existiera, Firestore actualiza el elemento, cosa que no ocurre en bases de datos noSQL, que devuelven un error.
- Superclave: comprueba que la combinación de atributos que forman la superclave son únicos dentro de la tabla.
- Restricciones al eliminar: impide eliminar un elemento si existen elementos de otras tablas que hacen referencia a él.

```

/**
 * Returns signal if restrictions to create are satisfied. If not, Error
 *
 * @param crop new crop
 * @return void signal
 */
private Mono<Void> checkConstraintsCreating(Crop crop) {
    final Mono<Boolean> existId = existId(crop.getId());
    final Mono<Boolean> existNameAndVariety = existNameAndVariety(crop.getName(), crop.getVariety());

    return Mono.zip(existId, existNameAndVariety)
        .flatMap(tuple -> {
            final boolean eId = tuple.getT1();
            final boolean eNameAndVariety = tuple.getT2();

            if(eId) {
                // Error, no se cumple la restricción de id UNICO
                throw new GenerateIdException(
                    String.format(MSG_ERROR_ID_EXIST, crop.getId())
                );
            }
            if(eNameAndVariety) {
                // Error, no se cumple la restricción de combinación de nombre y variedad UNICO
                throw new ConstraintViolationException(
                    String.format(MSG_ERROR_NAME_AND_VARIETY_EXIST, crop.getName() + " - " + crop.getVariety())
                );
            }
            // Se cumplen las restricciones
            return Mono.empty();
        });
}

/**
 * Returns signal if restrictions to update are satisfied. If not, Error
 *
 * @param crop updated crop
 * @return void signal
 */
private Mono<Void> checkConstraintsUpdating(Crop crop, Crop old) {
    return validNameAndVariety(crop.getName(), crop.getVariety(), old.getName(), old.getVariety())
        .flatMap(r -> {
            if (!r) {
                // Error, no se cumple la restricción de combinación de nombre y variedad UNICO
                throw new ConstraintViolationException(
                    String.format(MSG_ERROR_NAME_AND_VARIETY_EXIST, crop.getName() + " - " + crop.getVariety())
                );
            }
            // Se cumplen las restricciones
            return Mono.empty();
        });
}

/**
 * Returns signal if restrictions to delete are satisfied. If not, Error
 *
 * @param id identifier
 * @return void signal
 */
private Mono<Void> checkConstraintsDeleting(String id) {
    final Mono<Boolean> existSeed = seedService.existByCropId(id);
    final Mono<Boolean> existProduction = productionService.existByCropId(id);

    return Mono.zip(existSeed, existProduction)
        .flatMap(tuple -> {
            final boolean eSeed = tuple.getT1();
            final boolean eProduction = tuple.getT2();

            if(eSeed) {
                // Error, existen semillas asociadas a este cultivo
                throw new ConstraintViolationException(MSG_ERROR_ON_DELETE_EXIST_SEEDS);
            }
            if(eProduction) {
                // Error, existen producciones asociadas a este cultivo
                throw new ConstraintViolationException(MSG_ERROR_ON_DELETE_EXIST_PRODUCTIONS);
            }
            // Se cumplen las restricciones
            return Mono.empty();
        });
}
}

```

Figura 6.1: Código de control de restricciones de los cultivos

## 6.2 Seguridad

La seguridad es algo muy importante y sobre todo en una aplicación que estará disponible en la nube. Para poder impedir accesos no autorizados a los datos, se han implementado distintos sistemas de seguridad.

### 6.2.1 Tokens de autenticación

Utilizando la tecnología de JSON Web Tokens se genera un token codificado con un el algoritmo HS512. Esto sucede cuando el usuario a realizado un inicio de sesión correcto. Este tokens se envía como respuesta a la solicitud de autenticación. Con ello conseguimos que la contraseña de usuario esté en todas las llamadas, ahora sólo estará en una, la de la autenticación.

La clave para realizar el cifrado del token y la duración de los tokens se determina en el fichero de configuración del backend a la hora de realizar el despliegue. Hay que tener en cuenta que este fichero no puede ser accesible por cualquier persona, pues comprometería la seguridad de la aplicación.

### 6.2.2 Control de acceso a recursos

Sólo se permite el acceso sin estar identificado a la ruta para el inicio de sesión. Para el resto de recursos de la API, en la cabecera de la solicitud debe tener un token válido. A parte, dependiendo del rol del usuario, se limita el acceso a las funciones de la API. Si se intenta acceder a un recurso para el que no se tienen los privilegios suficientes se denegará la petición.

### 6.2.3 Control de fallos de inicio de sesión

Los ataques de fuerza bruta son muy comunes en internet, pues existen listas de contraseñas robadas que son utilizadas para realizar dichos ataques. Para evitarlos, se contarán los intentos erróneos de inicio de sesión, cuando se llegue a 5, el usuario será desactivado, esto impedirá que pueda iniciar sesión, pues sus peticiones serán siempre ignoradas. Si se realizan varios intentos, sin llegar al máximo, y por último se introduce la contraseña correcta, permitiendo el acceso, el contador se pone a 0.

Cuando un usuario ha sido bloqueado, la única persona que puede desbloquearla es el administrador. En el caso de que el usuario comprometido sea el administrador, este también se bloqueará, pero para poder desbloquearlo, se tendrá que realizar accediendo directamente a la base de datos.

```

1 package es.ull.tfg.higopico.security;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.http.HttpStatus;
5 import org.springframework.http.ResponseEntity;
6 import org.springframework.security.crypto.password.PasswordEncoder;
7 import org.springframework.web.bind.annotation.RequestBody;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RequestMethod;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import es.ull.tfg.higopico.services.UserService;
13 import reactor.core.publisher.Mono;
14
15 /**
16  * Users authentication controller
17  *
18  * @author Ruymán Rodríguez Martín
19  */
20 @RestController
21 @RequestMapping("/api")
22 public class AuthenticationController {
23
24     private static int MAX_LOGIN_FAILURES = 5; // Número de fallos para desactivar la cuenta
25
26     @Autowired
27     private JwtUtil jwtUtil;
28     @Autowired
29     private PasswordEncoder passwordEncoder;
30     @Autowired
31     private UserService userService;
32
33
34     /**
35      * Authenticate user
36      *
37      * @param ar authentication request
38      * @return authentication response or error
39      */
40     @RequestMapping(value = "/auth", method = RequestMethod.POST)
41     public Mono<ResponseEntity<?>> authenticate(@RequestBody AuthRequest ar) {
42         return userService.getActivatedByUsername(ar.getUsername())
43             .map((user) -> {
44                 if (passwordEncoder.matches(ar.getPassword(), user.getPassword())) {
45                     // Autenticado correctamente
46                     if (user.getLoginFailures() > 0) {
47                         // Contador de fallos a cero
48                         userService.resetSessionFailures(user).subscribe();
49                     }
50                     // Enviamos respuesta con token de identificación
51                     return ResponseEntity.ok(
52                         new AuthResponse(
53                             jwtUtil.generateToken(new AppUser(user)),
54                             user.getId(),
55                             user.getRole().toString()
56                         )
57                     );
58                 } else {
59                     if (user.getLoginFailures() > MAX_LOGIN_FAILURES - 1) {
60                         // Sobrepasado el máximo de fallos, bloqueamos el usuario
61                         userService.deactivate(user).subscribe();
62                     } else {
63                         // Incrementamos errores de login
64                         userService.increaseSessionFailures(user).subscribe();
65                     }
66                 }
67                 return ResponseEntity.status(HttpStatus.UNAUTHORIZED).build();
68             }
69         })
70         .defaultIfEmpty(ResponseEntity.status(HttpStatus.UNAUTHORIZED).build());
71     }

```

Figura 6.2: Código de inicio de sesión y control de fallos de autenticación

# Capítulo 7

## Interfaz web y móvil

El diseño de la interfaz está basado en las directrices de Material Design [10]. En la versión móvil el diseño cumple correctamente, pero en la versión web no se cumple estrictamente con todos los aspectos de dichas directrices.

### 7.1 Selección de colores

El primer paso para poder diseñar la aplicación es la selección de los colores que se van a utilizar en ella, para ello se utilizó la herramienta Color Tool de Material Design. Se seleccionó como color principal un verde y, para el color secundario, un púrpura (ver figura 7.1).



Figura 7.1: Selección de colores con Color Tool de Material Design

## 7.2 Diseño

En este apartado se mostrará el diseño de los principales elementos que componen la interfaz del usuario.

### 7.2.1 Formulario de inicio de sesión

El formulario de inicio de sesión, es la primera pantalla que le aparecerá al usuario. En la versión web se debe introducir el nombre de usuario y la contraseña. En la versión móvil, también hay que insertar la dirección del servidor de aplicaciones. Una vez que se haya iniciado sesión correctamente la aplicación memorizará la dirección y no la volverá a pedir mas.



Figura 7.2: Formulario de inicio de sesión (móvil)

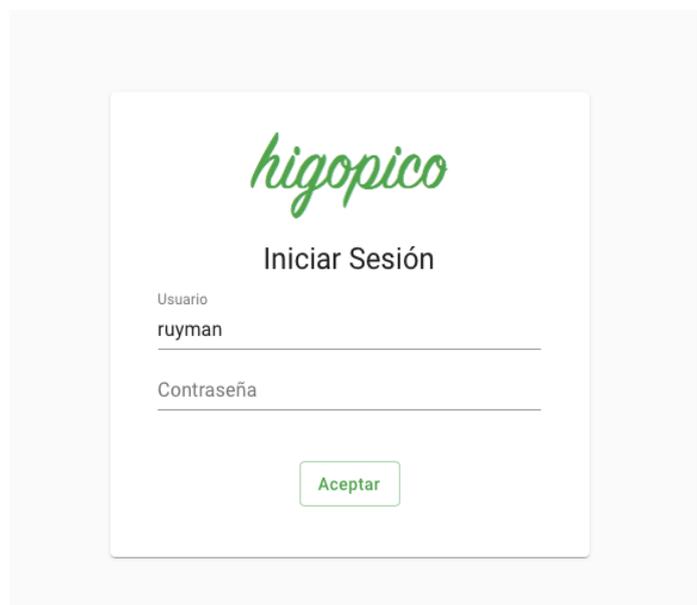


Figura 7.3: Formulario de inicio de sesión (web)

### 7.2.2 Menú lateral

El menú lateral es desplegable en las dos versiones de la aplicación. En la versión web, se oculta parcialmente, mostrando solamente los iconos de los elementos del menú. En la versión móvil, se debe realizar un gesto de deslizar el marco izquierdo hacia la derecha para que aparezca.



Figura 7.4: Menú lateral (móvil)

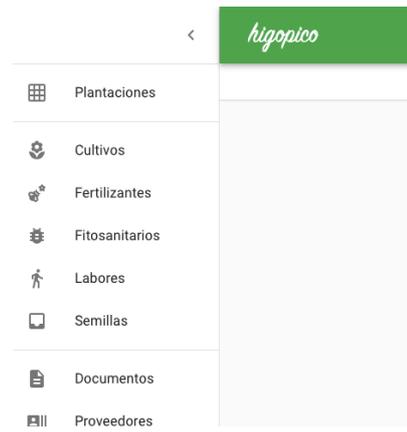


Figura 7.5: Menú lateral (web)

En la versión móvil se tiene acceso a los datos del usuario y al botón de cerrar sesión desde este menú, pero en la versión web, el usuario aparece en la barra de título de la aplicación, al lado derecho, donde un desplegable permite elegir entre ir al formulario de los datos del usuario o cerrar sesión.

### 7.2.3 Formularios

La mayoría de los formularios tienen el mismo estilo, esto es así, porque se ha generado un formulario padre del que heredan el resto de formularios. Con ello, se consigue que el aspecto sea siempre el mismo y en caso de necesitar modificar cualquier elemento, sólo se tiene que modificar el formulario padre.

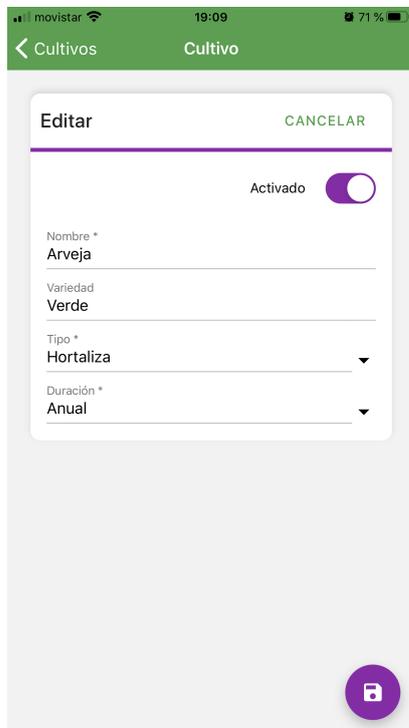


Figura 7.6: Formulario (móvil)

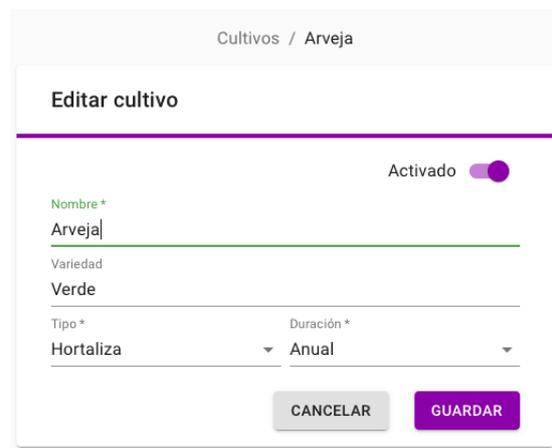


Figura 7.7: Formulario (web)

La barra de color púrpura que aparece debajo del título del encabezado se trata de una barra de progreso. Cuando se está realizando alguna consulta, la barra de progreso cambia a modo indeterminado, que la muestra cargando continuamente. Cuando termina el proceso, la barra vuelve a su estado normal, una línea de separación. Está pensado, para que el usuario sepa que la aplicación está realizando alguna operación.

El formulario dispone de un interruptor, para activar o desactivar los elementos. Este es de tipo opcional, pues no todos los formularios hijos lo requieren.

En la versión móvil, el botón de guardar, es un botón flotante, que desaparece si se cancela la edición. En el formulario web, se ocultan los botones si se está editando.

```
3 import Fertilizer from '../models/Fertilizer';
4 import FertilizerService from '../services/FertilizerService';
5 import Form from './Form';
6
7 // Elementos del formulario
8 const fields = [
9   {
10    name: 'name', label: 'Nombre', input:'textfield',
11    autoCapitalize: "sentences", validateOnBlur: true, required: true
12  },
13  {name: 'origin', label: 'Origen', input:'textfield'},
14  {name: 'active'},
15 ];
16
17 /*
18  Validación del formulario
19 */
20 const validateField = (field, value) => {
21   const error = {};
22   const requiredFields = [
23     'name',
24   ];
25   // Comprobar que exista un valor asignado
26   if (requiredFields.indexOf(field) > -1) {
27     if (!value) {
28       error[field] = 'Requerido';
29     }
30   }
31   return error;
32 }
33
34 export default function FertilizerForm({route, navigation, logout}) {
35   const { id, readOnly } = route.params; // Parámetros
36
37   return (
38     <Form
39       id={id}
40       title="fertilizante"
41       model={Fertilizer}
42       parents={[ ["Fertilizantes", "/fertilizantes"], ]}
43       fieldName="name"
44       fields={fields}
45       validateField={validateField}
46       get={FertilizerService.get}
47       create={FertilizerService.create}
48       update={FertilizerService.update}
49       activate={FertilizerService.changeActive}
50       readOnly={readOnly}
51       navigation={navigation}
52       logout={logout}
53     />
54   );
55 }
```

Figura 7.8: Formulario de fertilizantes (React Native - móvil)

## 7.2.4 Listas

En las listas, como ocurre en los formularios, existe una lista padre de las que el resto de listas heredan.

En la versión móvil, en las listas si se desplaza el elemento hacia la derecha, aparecerá el botón para desactivarlo. En cambio si lo se desplaza a la izquierda, aparecerá la opción de eliminarlo. Al final de la lista existe un botón flotante para insertar nuevos elementos.

La versión web, debido a que las pantallas son mayores que las de los dispositivos móviles, las listas son tablas de datos, que muestran muchas más información al usuario. También están disponibles los botones para activar o eliminar el elemento.

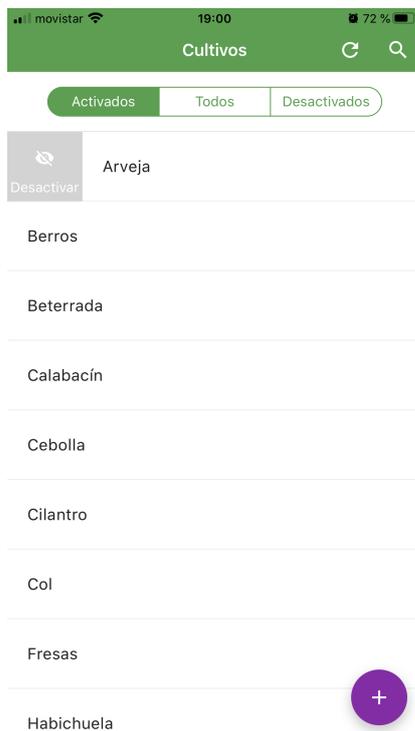


Figura 7.9: Lista 1 (móvil)

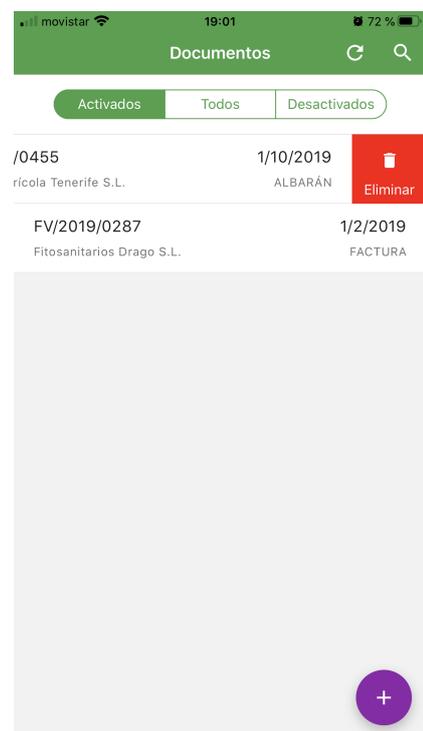


Figura 7.10: Lista 2 (móvil)

En la versión web, la listas paginan los resultados, mostrando nada más que la cantidad de elementos seleccionados por página. Para ello, tienen en la parte inferior todos los controles necesarios para navegar por las páginas y para la selección de la cantidad de elementos a mostrar.

También poseen la capacidad de mostrar el resultado en un orden concreto, realizando click en la etiqueta de la columna por la que se desea ordenar. Aparecerá una fecha que indica el orden, ascendente o descendente.

Cultivos
+ ↻

Activados
Todos
Desactivados

Nombre ↑	Variedad	Tipo	Duración	
Arveja	Verde	Hortaliza	Anual	🗑️ 🔍
Berros	De agua	Hortaliza	Perenne	🗑️ 🔍
Beterrada	Detroit	Hortaliza	Anual	🗑️ 🔍
Calabacín	Blanquete	Hortaliza	Anual	🗑️ 🔍
Cebolla	Blanca	Hortaliza	Anual	🗑️ 🔍
Cilantro	Común	Aromática	Perenne	🗑️ 🔍
Col	Cerrada	Hortaliza	Anual	🗑️ 🔍
Fresas	Reina de los valles	Fruta	Anual	🗑️ 🔍
Habichuela	Blanca	Hortaliza	Anual	🗑️ 🔍
Lechuga	Romana	Fruta	Anual	🗑️ 🔍

Filas por página: 10 ▾    1-10 de 15    < >

Figura 7.11: Lista (web)

```

1 import React from 'react';
2 import { useHistory } from 'react-router-dom';
3 import PropTypes from 'prop-types';
4 import FertilizerService from '../services/FertilizerService';
5 import DefaultTable from '../table/DefaultTable'
6
7 export default function FertilizerList(props) {
8   // Constantes
9   const {readOnly} = props; // Parametros
10
11   // Columnas
12   const columns = [
13     {id: 'id', label: 'Id', minWidth: 60, hidden: true},
14     {id: 'name', label: 'Nombre', minWidth: 200},
15     {id: 'origin', label: 'Origen', minWidth: 200},
16     {id: 'active', label: 'Activo', hidden: true},
17   ];
18
19   // Control de historial
20   let history = useHistory();
21
22   /*
23     Obtiene la lista de elementos (activos o todos)
24   */
25   const getData = (onlyActivated) => {
26     if(onlyActivated) {
27       return FertilizerService.getActivated();
28     } else {
29       return FertilizerService.getAll();
30     }
31   };
32
33   /*
34     Abre la edición del elemento indicado
35   */
36   const handleView = (id) => {
37     history.push("/fertilizantes/" + id);
38   }
39
40   /*
41     Creacion de un nuevo elemento
42   */
43   const handleNew = () => {
44     history.push("/fertilizantes/new")
45   }
46
47   /*
48     Elimina el elemento indicado
49   */
50   const handleDelete = (id) => {
51     return FertilizerService.delete(id);
52   }
53
54   /*
55     Activa o desactiva el elemento indicado
56   */
57   const handleActivate = (id, activate) => {
58     if (activate) {
59       return FertilizerService.activate(id);
60     } else {
61       return FertilizerService.deactivate(id);
62     }
63   }
64
65   return (
66     <DefaultTable
67       title="Fertilizantes"
68       columns={columns}
69       getData={getData}
70       onCreate={handleNew}
71       onView={handleView}
72       onDelete={handleDelete}
73       onActive={handleActivate}
74       readOnly={readOnly}
75     />
76   );
77 }
78
79 FertilizerList.propTypes = {
80   readOnly: PropTypes.bool.isRequired,
81 }

```

Figura 7.12: Lista de fertilizantes (React - web)

## 7.2.5 Tarjetas

Existen algunas de las entidades que se mostrarán como tarjetas (card), para mejorar la visibilidad y facilidad a la hora de seleccionarlas. Las entidades que utilizan este tipo son: plantaciones, parcelas, recintos y producciones.

Para poder acceder al formulario que corresponde a dicha tarjeta, se debe hacer click en editar. También están disponibles los botones para activar/desactivar y eliminar.

Las tarjetas están pensadas para que al tocarlas o hacer click, accedas al una vista de elementos que pertenecen a otra entidad. Por ejemplo, si se hace click en una tarjeta de una producción, no se accede al formulario de la producción, si no que accede a una lista con todas las parcelas de esa producción. Sólo hay una excepción, en las producciones es el único elemento donde se accede al su formulario, pues éstas no tiene elementos dependientes de ellas.



Figura 7.13: Tarjeta (móvil)

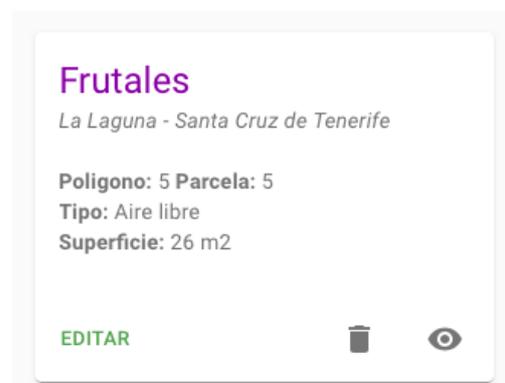


Figura 7.14: Tarjeta (web)

## 7.2.6 Listas de tarjetas

Al igual que las listas, existe una clase padre de la que heredan el resto de objetos de listas de tarjetas. Hay disponibles botones flotantes para añadir nuevos elementos y para actualizar, este último sólo en la versión web.

En la parte superior de estas listas hay una barra de navegación que permite conocer la ubicación exacta, y si se hace click sobre alguno de los elementos, lleva al usuario hacia esa ubicación.

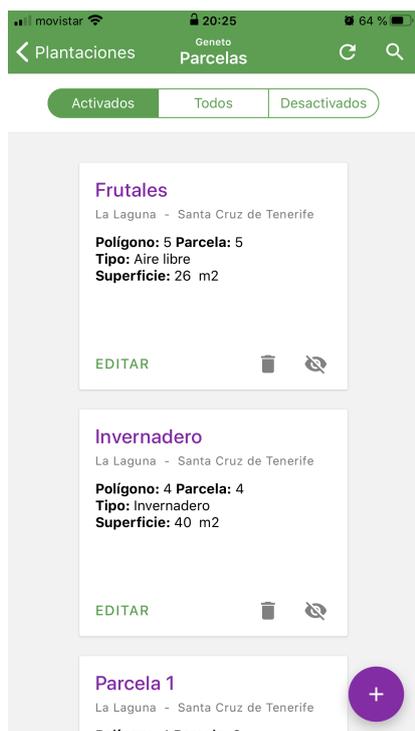


Figura 7.15: Lista de tarjetas (móvil)

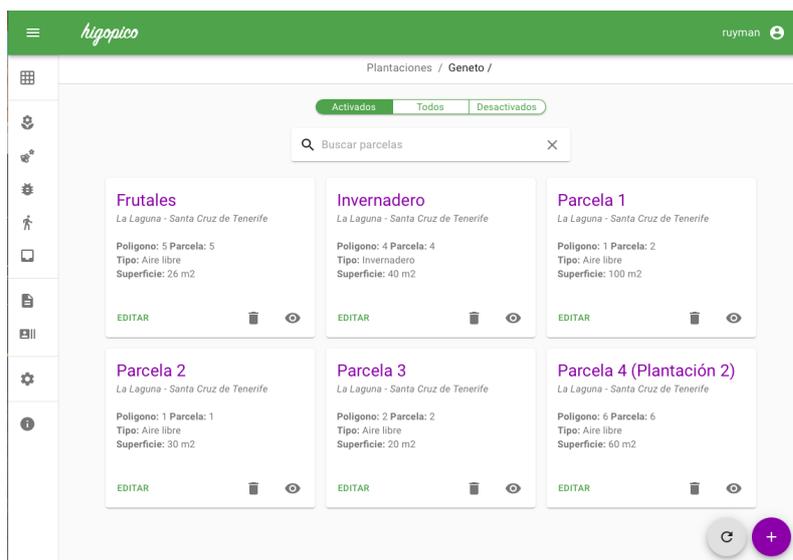


Figura 7.16: Lista de tarjetas (web)

## 7.2.7 Campo de búsqueda

En todas las listas, independientemente de si son de tarjetas o no, tenemos disponible un campo para realizar búsquedas por cualquier dato que sea visible para el usuario. En la versión para móviles, se debe hacer click en el icono de la lupa que aparece en la barra de título de la aplicación. En la versión web, el campo aparece flotando sobre las listas.



Figura 7.17: Campo de búsqueda (móvil)

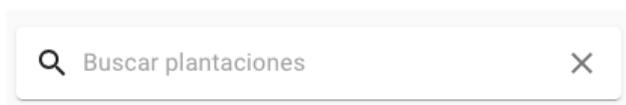


Figura 7.18: Campo de búsqueda (web)

## 7.2.8 Filtro de elementos por su estado

Las listas, tanto normales como de tarjeta, tiene en su parte superior un conjunto de botones que permiten mostrar los elementos que están activados, desactivados o todos, dependiendo de la selección que se que se realice.



Figura 7.20: Filtro (móvil)



Figura 7.19: Filtro (web)

## 7.2.9 Formularios especiales

Existen dos formularios distintos al resto. Esto formularios pertenecen a la entidad documento y la entidad producción. Son diferentes a los demás, porque dentro contienen atributos que son del tipo muchos a muchos, por tanto deben se representados como listas.

### - Documentos:

En este formulario aparecen los detalles del documento, que son fitosanitarios o semillas. En él se pueden añadir, eliminar o desactivar cada uno de estos elementos. En la versión de dispositivos móviles, aparece un menú inferior que permite desplazarse entre la información del documento y la lista de detalles del mismo.

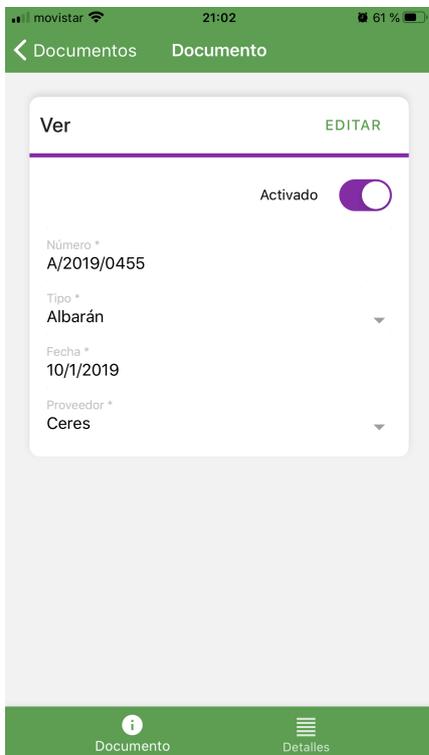


Figura 7.21: Formulario documento 1 (móvil)

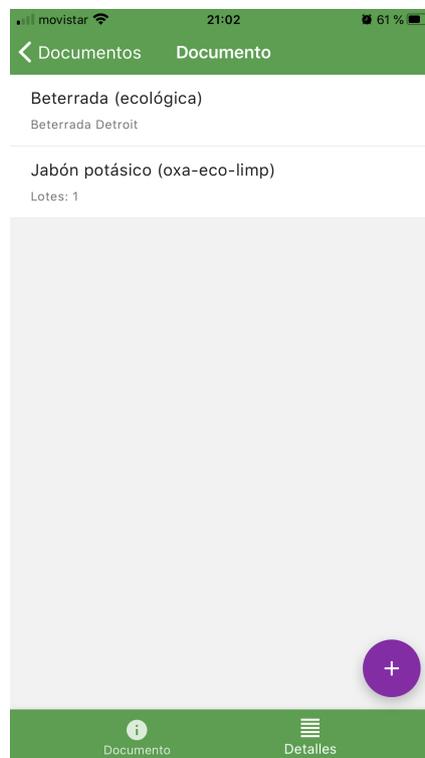


Figura 7.22: Formulario documento 2 (móvil)

Documentos / FV/2019/0287

### Editar documento

Activado

Número *	Tipo *	Fecha *
FV/2019/0287	Factura	2019-01-02

Proveedor \*

Fitosanitarios Drago S.L.

Detalles + FITOSANITARIO   + SEMILLA

Beterrada (ecológica) <small>Cultivo: Beterrada Detroit</small>	<input type="checkbox"/> <input type="checkbox"/>
Bacillus <small>Lotes: 2355, 2445</small>	<input type="checkbox"/> <input type="checkbox"/>
Jabón potásico (oxa-eco-limp) <small>Lotes:</small>	<input type="checkbox"/> <input type="checkbox"/>
Azufre micronizado <small>Lotes:</small>	<input type="checkbox"/> <input type="checkbox"/>

CANCELAR
GUARDAR

Figura 7.23: Formulario documento (web)

## - Producción:

En las producciones se realizan trabajos como son abonado, tratamientos y labores. Estas aparecen como listas dentro del formulario de la producción. Tanto en la versión web como en la móvil hubo que añadir un menú para poder navegar entre estas opciones.

Figura 7.25: Formulario producción 1 (móvil)

Figura 7.24: Formulario producción 2 (móvil)

**Producción** EDITAR

INFORMACIÓN ABONADOS TRATAMIENTOS LABORES

Activado

Nombre \* Calabazas Campaña \* 2019

Cultivo \* Calabaza Común

Semilla

Cantidad plantada 10 Medida Superficie

Observaciones

**Siembra:**

Inicio	Fin
2019-03-23	2019-04-23

**Recolección:**

Inicio	Fin
2020-06-14	2020-06-15

**Producción**

INFORMACIÓN ABONADOS TRATAMIENTOS LABORES

Azufre micronizado

fecha: 16/6/2020 dosis:

observaciones:

origen: Fitosanitarios Drago S.L. (Factura FV/2019/0287 del 2/1/2019) lotes:

Figura 7.26: Formulario producción 1 (web) Figura 7.27: Formulario producción 2 (web)

## 7.3 Funcionalidad

Las funcionalidades de la aplicación son muchas, pero dependen del rol del usuario que se conecte. Como se indicó en la introducción, existen 3 roles distintos, por lo que se explicarán las funcionalidades dependiendo de estos roles:

- **Todos:**

Es común para todos los usuarios tener acceso a su formulario de datos de usuario. Desde ese formulario pueden modificar sus contraseñas de acceso y ver sus datos de usuario.

Tienen acceso a la información sobre la aplicación “Acerca de ...”.

- **Administrador:**

Al acceder a la aplicación sólo tendrá acceso en el menú a los usuarios. Su pantalla principal es la lista de usuarios. Podrá añadir, editar, activar/desactivar o eliminar usuarios. Podrá poner los contadores de fallos de inicio de sesión de los usuarios a cero, dentro del formulario de cada usuario. También se le permite la modificación de los roles y cambiar las contraseñas de otros usuarios.

- **Inspector:**

En el menú principal solamente tendrá disponible el acceso a plantaciones y a la información del productor. Su pantalla principal es la lista de plantaciones. Desde

plantaciones puede acceder a las parcelas, recintos y producciones. A todos los lugares donde tiene acceso sólo puede leer datos, nunca podrá modificar, pues no se le permite.

Todos los campos de observaciones que existen en los distintos formularios no son visibles por este tipo de usuario, esos datos no son relevantes para ellos.

Puede realizar exportaciones de datos, pero sólo le aparecerán los datos a los que tiene acceso.

- **Productor:**

Es el usuario principal, es el encargado de introducir todos los datos, excepto los de los usuarios. A estos no tiene acceso. En el menú principal le aparecen todos los elementos, menos el de acceso a usuarios. Su pantalla principal es la lista de plantaciones.

Puede modificar la información de el productor.

Puede crear, modificar y eliminar provincias y municipios.

Puede crear, modificar, activar/desactivar y eliminar plantaciones, parcelas, recintos, producciones, cultivos, fertilizantes, fitosanitarios, labores, semillas, documentos y proveedores.

Puede crear, modificar y eliminar abonados, tratamientos y labores dentro de las producciones.

Puede realizar exportaciones de los datos.

## 7.4 Exportación de datos

La aplicación permite la exportación de datos a Excel, esta exportación se realiza desde una producción. Se debe indicar el año de la campaña a exportar, opcionalmente se puede indicar que los exporte todos. El documento de Excel resultante contendrá todos los datos de la plantación divididos de tal manera que se parezcan lo máximo posible a como se muestran en la aplicación. También contendrá hipervínculos, que asocian todos los datos, permitiendo navegar entre ellos. Un ejemplo, si estas en una producción y quieres ver la semilla que se ha utilizado, realizas click en ese elemento y se desplaza hasta la hoja y fila que corresponden.

Sólo se exportan los datos necesarios para las campañas indicadas.



Figura 7.28: Exportar a Excel (móvil)

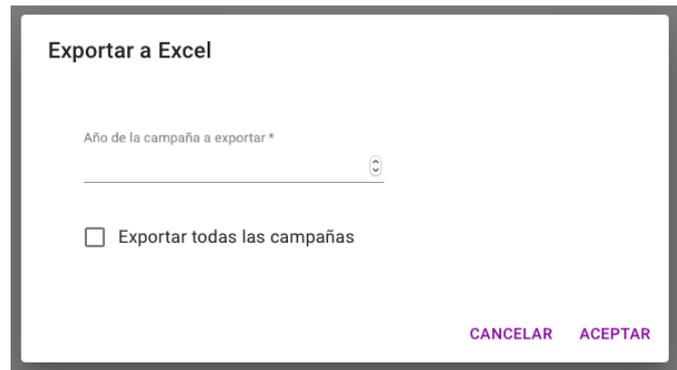


Figura 7.29: Exportar a Excel (web)

	A	B	C	D	E	F	G	H	I
1	<b>Bancales</b>								
2				Siembra		Recolección			
3	Id	Campaña	Cultivo	Inicio	Fin	Inicio	Fin	Semilla	Observaciones
4	1	2019	Barros De agua	15/11/2019					Observación
5	2	2019	Barros De agua	08/01/2019					
6									
7									
8									
9	<b>Suelo</b>								
10				Siembra		Recolección			
11	Id	Campaña	Cultivo	Inicio	Fin	Inicio	Fin	Semilla	Observaciones
12	42	2019	Barros De agua	25/02/2019					
13	45	2019	Papino Español	12/08/2019	12/08/2019				
14									
15									
16									
17	<b>Vivero</b>								
18				Siembra		Recolección			
19	Id	Campaña	Cultivo	Inicio	Fin	Inicio	Fin	Semilla	Observaciones
20	46	2019	Calabaza Común	02/03/2019	02/03/2019	02/06/2019	02/06/2019		
21	47	2019	Calabaza Común	10/04/2019	10/04/2019				
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									

Figura 7.30: Documento de Excel resultante de la exportación

# Capítulo 8

## Presupuesto

A continuación se detallarán la tareas realizadas y las horas invertidas. El precio de la hora está calculado en base a un salario de 2.000,00 € brutos mensuales más una cuota a la seguridad social de 500,00 €, que es el coste aproximado para una empresa. Finalmente el precio de una hora de trabajo calculando que un mes tiene 20 días laborales y cada día se trabajan 8 horas es de 15,66 €.

Descripción	Cantidad	Precio (€)	Total (€)
Planificación del proyecto	10	15,66	150,66
Lectura de bibliografía y documentación	40	15,66	626,40
Creación del prototipo de la aplicación	35	15,66	548,10
Desarrollo	300	15,66	4.698,00
Redacción de la memoria	30	15,66	469,80
Ordenador de sobremesa	1	629,00	629,00
Tableta HUAWEI MediaPad T5 10,1”	1	161,29	161,29
Google Firebase *	1	0	0
Amazon AWS Elastic Beanstalk *	1	0	0
		<b>TOTAL</b>	<b>7.289,19</b>

\* versión gratuita, si se superan los límites establecidos por esta se deben consultar los precios según las necesidades

**Tabla 8.1:** Presupuesto del proyecto

# Capítulo 9

## Conclusiones y líneas futuras

Se ha cumplido con los objetivos propuestos. La aplicación es completamente funcional tanto en su versión web como para dispositivos móviles. Actualmente se encuentra en uso por una productora local, que ha colaborado activamente con este proyecto, aportando todos sus conocimientos del sector e indicando cuales eran sus demandas en cuanto a aplicaciones de cuadernos de campo digital se refiere. El funcionamiento y el aspecto visual de la aplicación, tanto en la web como en los dispositivos móviles, es muy similar, lo que elimina el tener que aprender a utilizar la aplicación en las distintas plataformas.

Personalmente, he aprendido a desarrollar en nuevas tecnologías, que hoy en día están muy demandadas, y que no había utilizado hasta este proyecto. He utilizado muchas de las cosas aprendidas durante estos años en la universidad, que me han permitido desarrollar una aplicación que es completamente funcional. Y una de las cosas más importantes, he podido escuchar las necesidades de una productora local y transformarlas en la aplicación que necesitaban.

En un futuro, la aplicación puede seguir mejorando, añadiendo nuevas funcionalidades y/o modificando las existentes:

- Cambiar a una base de datos de tipo SQL, lo que aumentaría la velocidad, debido a la reducción de llamadas realizadas para el control de integridad de los datos, del cual se encargaría el gestor de bases de datos. Y, por este mismo motivo, se reduciría la carga sobre el servidor de aplicaciones.
- Añadir la exportación del cuaderno de campo de una campaña a pdf.
- Modificar la aplicación para que sea multiempresa.

# Capítulo 10

## Summary and Conclusions

All the proposed objectives have been met. The app is completely functional both in its web version and mobile devices. Currently it is in use by a local production company, that has worked closely with this project. She has providing all her knowledges of this sector and indicating what her demands were in respect of digital field notebook applications. The app's functioning and visual appearance in both web version and mobile devices is very similar, this eliminates the need for learning to use two different platforms.

I personally have learned to develop new technologies, that nowadays are very demanded, and that I have never used before. I have used what I have learned from different subjects, at university, during these years. Hence, the acquired knowledges have let me to develop a completely functional app. And one of the most important things, I have been able to hear the needs of a local producer company and to transform them into the app that they needed.

In a future, this app could be improved, adding new functionalities and/or modifying the existent ones:

- Providing a new SQL type database, this would increase the speed, due to the reduction of calls made for data integrity control, which would be handled by database manager. For that same reason, the charge over the apps server would be reduced.
- Adding the export of a campaign's field notebook to PDF.
- Modifying the app to be multi-company.

# Bibliografía

- [1] 650 Industries, Inc. "Expo." *Expo*. N.p. Web. 04 Julio 2020. <<https://expo.io>>
- [2] Agricolum S.L. "Agricolum - Precios" *agricolum.com*. N.p. Web. 04 Julio 2020. <<https://agricolum.com/precios>>
- [3] Agricolum S.L. "Cuaderno de campo agrícola | Agricolum." *agricolum.com*. N.p. Web. 04 Julio 2020. <<https://agricolum.com>>
- [4] Agroboca, S.L. "Tipos de agricultura y calidad en frutas y verduras." *agroboca.com*. N.p. Web. 04 Julio 2020. <<https://www.agroboca.com/tipos-de-agricultura>>
- [5] Agroptima, S.L. "Software y Aplicación de gestión agrícola fácil de usar Agroptima®." *Agroptima.com*. N.p. Web. 04 Julio 2020. <<https://www.agroptima.com/es/>>
- [6] Auth0 Inc. "JSON Web Tokens – jwt.io." *jwt.io*. N.p. Web. 04 Julio 2020. <<https://jwt.io>>
- [7] Facebook Inc. "React – Una biblioteca de JavaScript para construir interfaces de usuario." *reactjs.org*. N.p. Web. 04 Julio 2020. <<https://es.reactjs.org>>
- [8] Facebook Inc. "React Native · A framework for building native apps using React." *reactnative.dev*. N.p. Web. 04 Julio 2020. <<https://reactnative.dev>>
- [9] Gobierno de Canarias. "Cuaderno de campo." *gobiernodecanarias.org*. N.p. Web. 04 Julio 2020. <<https://www.gobiernodecanarias.org/agricultura/temas/cuaderno/>>
- [10] Google LLC. "Material Design." *Material Design*. N.p. Web. 04 Julio 2020. <<https://material.io>>
- [11] Keynet Sistemas Informaticos, S.L. "Software de gestión para el sector agroalimentario – ToolAgro." *Toolagro.com*. N.p. Web. 04 Julio 2020. <<https://toolagro.com/>>
- [12] Material-UI. "Material-UI: A popular React UI framework." *material-ui.com*. N.p. Web. 04 Julio 2020. <<https://material-ui.com>>

- [13] The Apache Software Foundation. "Apache POI - the Java API for Microsoft Documents." *apache.org*. N.p. Web. 04 Julio 2020. <<https://poi.apache.org>>
- [14] Tortosa, Germán. "Diferencias entre tipos de agricultura: la integrada, la agricultura convencional y la agricultura ecológica." *Compostando Ciencia Lab*. N.p. 29 Mayo 2014. Web. 04 Julio 2020. <<http://www.compostandociencia.com/2014/05/tipos-de-agricultura-html/>>
- [15] VMware, Inc. "Spring | Home." *Spring*. N.p. Web. 04 Julio 2020. <<https://spring.io>>
- [16] VMware, Inc. "Spring Boot." *Spring*. N.p. Web. 04 Julio 2020. <<https://spring.io/projects/spring-boot>>
- [17] VMware, Inc. "Web on Reactive Stack." *Spring*. N.p. Web. 04 Julio 2020. <<https://docs.spring.io/spring/docs/current/spring-framework-reference/web-reactive.html>>