



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



## **TFG del Grado en Ingeniería Informática**

**APP móvil para la gestión de una ganadería**



Presentado por Elena Recio Pérez en Universidad de  
Burgos —9 de julio de 2024

Tutor: José Ignacio Santos Martín





UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. José Ignacio Santos Martín, profesor del departamento de Ingeniería de Organización.

Expone:

Que el alumno D. Elena Recio Pérez, con DNI 72902846E, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG. Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 9 de julio de 2024

Vº. Bº. del Tutor:

## **Resumen**

La parte administrativa de una ganadería es la parte más tediosa del negocio, ya que se requieren muchos registros por cada uno de los animales de la explotación, así como información a la orden del día de todo lo que ocurre en ella.

Para ello, en este proyecto se propone el desarrollo de una aplicación móvil para que los ganaderos puedan registrar y consultar datos en cualquier momento del día sin depender de los horarios de las administraciones públicas, para así agilizar el proceso de registro de animales una vez que se acude a determinada administración.

La aplicación desarrollada lleva integrada una base de datos para poder guardar en nuestro dispositivo todos aquellos datos que queramos que estén visibles en la aplicación.

La principal idea es que en un futuro la podamos lanzar a GooglePlay para que todas aquellas personas dedicadas a este sector puedan hacer uso de ella.

## **Descriptores**

Ganadería, sprint, animal, Android, gestión de animales.

## **Abstract**

The administrative part of a livestock farm is the most tedious part of the business, as many records are required for each of the animals on the farm, as well as day-to-day information on everything that happens on the farm.

To this end, this project proposes the development of a mobile application so that livestock farmers can register and consult data at any time of the day without depending on the opening hours of the public administrations, in order to speed up the process of registering animals once they go to a certain administration.

The developed application has an integrated database to be able to save in our device all the data that we want to be visible in the application.

The main idea is that in the future we will be able to launch it on GooglePlay so that all those dedicated to this sector can make use of it.

## **Keywords**

Livestock, sprint, animal, Android, animal management.

# Índice general

---

<b>1. Introducción .....</b>	<b>1</b>
1.1. Estructura de la memoria.....	2
1.2. Documentos adjuntos .....	3
1.3. Enlaces adjuntos .....	3
<b>2. Objetivos del proyecto .....</b>	<b>4</b>
2.1. Objetivos generales .....	4
2.2. Objetivos técnicos.....	4
2.3. Objetivos personales .....	5
<b>3. Conceptos teóricos .....</b>	<b>6</b>
<b>4. Técnicas y herramientas.....</b>	<b>12</b>
4.1. Metodologías ágiles.....	12
4.2. Control de versiones.....	12
4.3. Gestión del proyecto .....	13
4.4. Entorno de desarrollo integrado .....	13
4.5. Lenguaje de programación.....	13
4.6. Edición de texto.....	14
4.7. Hosting del repositorio.....	14
4.8. Gestor de referencias .....	14
<b>5. Aspectos relevantes del desarrollo del proyecto .....</b>	<b>15</b>
5.1. Inicio del proyecto .....	15
5.2. Metodología de trabajo .....	15
5.3. Formación.....	16
5.4. Desarrollo del proyecto .....	17
<b>6. Trabajos relacionados.....</b>	<b>18</b>
<b>7. Conclusiones y líneas de trabajo futuras .....</b>	<b>20</b>
<b>Bibliografía .....</b>	<b>21</b>

# Índice de ilustraciones

---

Ilustración 1.....	6
Ilustración 2.....	7
Ilustración 3.....	8
Ilustración 4.....	9
Ilustración 5.....	10

# 1. Introducción

---

La ganadería es una actividad económica vital, especialmente en áreas donde la agricultura no puede satisfacer las necesidades alimenticias de la población. Además de ser una fuente crucial de alimentos, la ganadería contribuye al desarrollo económico y social, proporcionando empleo e ingresos a millones de personas en áreas rurales. En la actualidad el sector ganadero se enfrenta a desafíos significativos, como la deforestación, la emisión de gases de efecto invernadero y la explotación laboral, entre otros problemas sociales y medioambientales.

La gestión de las explotaciones ganaderas es una actividad compleja y en ocasiones poco tecnificada. Muchos ganaderos aún llevan un control manual y en papel de aspectos vitales como nacimientos, ventas, muertes, alimentación y salud de los animales, lo que puede ser complicado y tedioso, especialmente cuando se tiene un gran número de animales a cargo.

Aunque existen soluciones tecnológicas para mejorar esta gestión, éstas suelen ser costosas y difíciles de usar para aquellos menos familiarizados con la tecnología. Por ello, es fundamental desarrollar herramientas accesibles y eficientes que faciliten la administración y control de las actividades ganaderas. El objetivo de este proyecto es desarrollar una aplicación de gestión de explotaciones bovinas que permita a los ganaderos llevar un control detallado y simplificado de sus animales, abarcando desde los registros de nacimientos y ventas hasta la alimentación y los cuidados médicos.

Con el avance de las nuevas tecnologías, se espera que la ganadería evolucione hacia formas más sostenibles y responsables, manteniendo su papel crucial en la alimentación mundial y el desarrollo económico de las comunidades rurales. Al adoptar prácticas más innovadoras y sostenibles, la ganadería puede continuar siendo una fuente significativa de empleo e ingresos, contribuyendo al equilibrio de los ecosistemas y al bienestar social y económico de las poblaciones rurales en todo el mundo.



## 1.1. Estructura de la memoria

La presente memoria contiene la siguiente estructura e información:

- **Introducción:** pequeña introducción al tema principal abordado para adentrarnos en el mundo de la ganadería. Estructura de la memoria y de los productos entregados.
- **Objetivos del proyecto:** se detallan los objetivos principales del proyecto.
- **Conceptos teóricos:** pequeñas explicaciones de cada uno de los conceptos teóricos que se han ido aplicando a lo largo del proyecto.
- **Técnicas y herramientas:** explicación y listado de las técnicas que se han utilizado para el desarrollo del proyecto, así como las herramientas utilizadas para aplicarlas.
- **Aspectos relevantes del desarrollo:** visión completa y detallada del desarrollo del proyecto, donde se abarcan los aspectos más interesantes y relevantes como los de la fase de análisis, diseño o implementación, además del diseño, ciclo de vida del proyecto y la propia implementación.
- **Trabajos relacionados:** pequeño resumen de los trabajos relacionados que se han encontrado para este proyecto.
- **Conclusiones y líneas de trabajo futuras:** conclusiones acerca del desarrollo y posibles planes de futuro para el proyecto

Aparte de este documento, se proporciona otro documento con todos los anexos correspondientes, que contiene:

- **Plan de proyecto software:** se detalla la planificación temporal llevada a cabo y el estudio de la viabilidad, tanto económica como legal.
- **Especificación de requisitos:** se detallan los requisitos funcionales y no funcionales del proyecto, los objetivos generales y los casos de uso.
- **Especificación de diseño:** se documenta la fase del diseño, la parte de software, el diseño de los datos y el diseño arquitectónico.
- **Documentación técnica de programación:** se muestra y explica la estructura de directorios, el manual del programador, la compilación y ejecución de la aplicación, así como las pruebas del sistema.
- **Documentación de usuario:** se detallan los requisitos de los usuarios, la instalación del proyecto para que puedan hacer uso de ella y el manual de usuario para conocer el funcionamiento.
- **Anexo de sostenibilización curricular:** reflexión personal acerca de la sostenibilidad de nuestro proyecto.

## **1.2. Documentos adjuntos**

- Aplicación Android, Gestión Ganadera.
- Memoria del proyecto.
- Anexos del proyecto

## **1.3. Enlaces adjuntos**

- Repositorio GitHub
- Videos en sharepoint de descripción del proyecto.
- Zube

## 2. Objetivos del proyecto

---

### 2.1. Objetivos generales

El objetivo principal del proyecto es desarrollar una aplicación móvil para dispositivos Android destinada a la gestión administrativa de una ganadería. Esta aplicación permitirá a los ganaderos:

- **Gestionar registros administrativos:** facilitar la administración de altas, bajas, nacimientos y muertes de los animales.
- **Acceso a la información en cualquier momento:** proveer a los ganaderos de una herramienta que les permita consultar información relevante sobre los animales en cualquier lugar y momento
- **Almacenamiento estructurado de datos:** garantizar que todos los datos se almacenen de manera organizada y fácilmente accesible.
- **Generar vistas de resúmenes:** ofrecer resúmenes claros y concisos de la información registrada para una mejor toma de decisiones.

### 2.2. Objetivos técnicos

Para poder alcanzar los objetivos generales, se han establecido los siguientes objetivos técnicos:

- **Desarrollar un algoritmo eficiente:** implementar un algoritmo que maneje eficazmente los datos de la ganadería.
- **Arquitectura Modelo-Vista-Controlador:** implementar una arquitectura adecuada que incluya modelos, controladores y fragmentos, asegurando un código limpio y mantenible
- **Gestión de dependencias con Gradle** para la gestión de dependencias y la construcción del proyecto, facilitando la integración de librerías y herramientas
- **Control de versiones:** integrar el uso de herramientas de control de versiones como Git y GitHub para llevar un registro de los cambios y colaborar eficientemente
- **Desarrollo colaborativo en Fork** para gestionar el desarrollo colaborativo del proyecto.
- **Diseño de interfaces de usuario:** utilizar mockups para diseñar interfaces de usuario intuitivas y eficientes.
- **Metodologías ágiles:** aplicar metodologías ágiles como Scrum para gestionar el desarrollo del proyecto de manera iterativa y colaborativa.

## 2.3. Objetivos personales

A nivel personal, mis objetivos han sido:

- **Contribuir al sector ganadero:** realizar una contribución significativa al sector ganadero, continuando con la tradición familiar.
- **Aplicación de conocimientos adquiridos:** aplicar y consolidar el máximo número de conocimientos adquiridos durante la carrera.
- **Exploración de nuevas herramientas:** utilizar metodologías y herramientas novedosas que son estándar en el mercado laboral.
- **Introducción a la visión artificial:** adentrarme en el campo de la visión artificial, explorando su aplicación en la ganadería.
- **Profundización en el desarrollo de aplicaciones Android:** incrementar el conocimiento y la experiencia en el desarrollo de aplicaciones Android y aumentar el contacto con el ecosistema de Google.

Con estos objetivos, se busca no solo desarrollar una herramienta útil para la gestión ganadera, sino también crecer profesionalmente, integrando conocimientos teóricos y prácticos en un proyecto de relevancia personal y profesional.

## 3. Conceptos teóricos

---

Antes de comenzar con el proyecto, es primordial hacer un abordaje teórico de lo que vamos a usar y como se va a usar, por ello, lo primero es definir el patrón de la aplicación, ver qué tipo de base de datos se utilizara y como se van a gestionar las dependencias. Se detalla todo a continuación:

### 3.1. Modelo-Vista-Controlador (MVC)

El patrón Modelo-Vista-controlador [30] fue creado en el contexto Smalltalk para separar la interfaz gráfica del código funcional de una aplicación. Como tuvo un gran impacto, se empezó a aplicar a otros lenguajes de programación orientados a objetos como es en este caso, Java.

El principal objetivo del patrón es apoyar los modelos funcionales y mentales de la información relevante para los usuarios, permitiendo un modelo que facilite su consulta y manejo. Para construir artefactos manejables, es crucial incorporar los modelos mentales del usuario desde el inicio del diseño del sistema, ya que hacerlo después puede ser inviable.

Sus componentes, como bien indica su nombre son, un modelo de datos, una vista y un controlador, se detallan a continuación:

- Modelo: representa la parte lógica y los datos de la aplicación. Se encarga de gestionar el estado de la aplicación y de interactuar con la base de datos.
  - Un ejemplo de modelo es mi clase Animal, que representa los datos de un animal en la explotación ganadera con los mismos campos que tenemos en la base de datos. Adjunto imagen de la declaración de los campos, se omiten los constructores, getters y setters.

```
public class Animal { 18 usages  👤 Elena Recio
    private long id; 3 usages
    private String nombre; 4 usages
    private String crotal; 4 usages
    private String crotal_madre; 4 usages
    private String crotal_padre; 4 usages
    private String sexo; 4 usages
    private String fecha_nac; 4 usages
    private String raza; 4 usages
}
```

*Ilustración 1*

- Vista: es la interfaz de usuario de la aplicación. Se encarga de presentar los datos al usuario y de capturar las interacciones de este, como pueden ser las entradas de texto.
  - Un ejemplo de vista es la de los animales de mi aplicación, donde en ella se muestra un listado con los campos que tenemos de cada animal. Se adjunta captura del código para el listado:

```
<TextView
    android:id="@+id/tvCrotal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Crotal"
    android:textSize="30sp"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:paddingBottom="4dp" />

<TextView
    android:id="@+id/tvNombre"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nombre"
    android:textSize="24sp"
    android:textStyle="bold"
    android:textColor="@color/black"
    android:paddingBottom="4dp" />

<TextView
    android:id="@+id/tvCrotalMadre"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
```

*Ilustración 2*

- Controlador: actúa a modo de intermediario entre el modelo y la vista. Gestiona toda la lógica de la aplicación, es decir, procesa las entradas del usuario recibidas desde la vista y actualiza el modelo en consecuencia. Por lo que también puede actualizar la vista para reflejar los cambios en el modelo.
  - Continuando con los animales, un ejemplo de controlador sería el siguiente:

```
public class AnimalesControlador { 7 usages  ▲ Elena Recio
    private BD bd; 5 usages
    private String TABLA_ANIMALES = "t_animales"; 4 usages
    public AnimalesControlador () { 2 usages  ▲ Elena Recio
    }

    public void inicializarBD(Context context) { bd = new BD(context); }

    public int eliminarAnimal(Animal animal) { no usages  ▲ Elena Recio
        SQLiteDatabase base = bd.getWritableDatabase();
        String[] argumentos = {String.valueOf(animal.getId())};
        return base.delete(TABLA_ANIMALES, whereClause: "id= ?", argumentos);
    }

    public long nuevoAnimal(Animal animal) { 1 usage  ▲ Elena Recio
        SQLiteDatabase base = bd.getWritableDatabase();
        ContentValues insertar = new ContentValues();
        insertar.put("nombre", animal.getNombre());
        insertar.put("crotal", animal.getCrotal());
        insertar.put("crotal_madre", animal.getCrotal_madre());
        insertar.put("crotal_padre", animal.getCrotal_padre());
        insertar.put("sexo", animal.getSexo());
        insertar.put("fecha_nac", animal.getFecha_nac());
        insertar.put("raza", animal.getRaza());

        long result = base.insert(TABLA_ANIMALES, nullColumnHack: null, insertar);
        if (result == -1) {
```

*Ilustración 3*

Lo que hace es inicializar la base de datos con la que estamos conectados y con los datos del modelo se gestionan desde el controlador todas las operaciones de añadir un animal, eliminar, editar etc.

A continuación, se adjunta un gráfico que resume a la perfección lo que es este patrón:

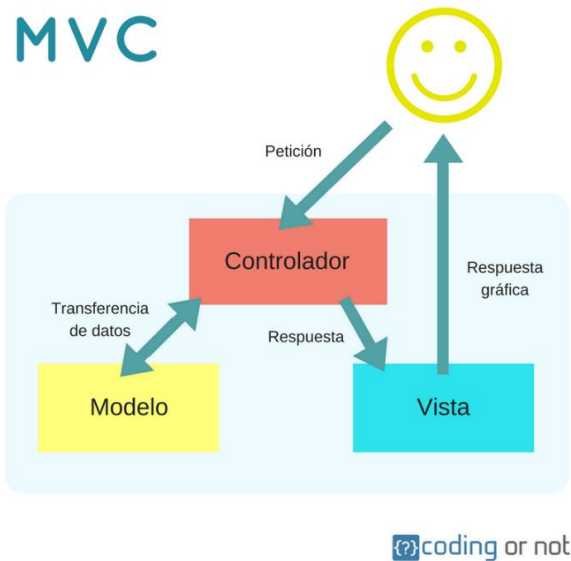


Ilustración 4 [30]

### 3.2. View/Container

En Android, esto se refiere a los componentes de la interfaz de usuario que ayudan a organizar y gestionar los elementos de la interfaz de usuario. En este proyecto se hace mucho uso de esto, ya que hay muchas views, cada botón, cada texto, cada entrada por teclado es un view, pero el conjunto de estas, como por ejemplo los listados o formularios hacen un container. A continuación, se explica más detalladamente:

- **View**, se trata de la clase base para todos los componentes de la interfaz de usuario en Android. Representa un rectángulo en la pantalla y se utiliza para capturar eventos como los toques en pantalla, la entrada de texto etc. Algunos de los muchos ejemplos de esto son: TextView, ImageView, Button.
- **Container**, es la clase que se encarga de organizar los views detallados anteriormente, se utilizan sobre todo para organizar la composición de interfaz de usuario de la aplicación. Alguno de los muchos ejemplos de esto puede ser: LinearLayout, RelativeLayout, ConstraintLayout.

A continuación, se adjunta una imagen de código de una de las vistas del proyecto para aclararlo mejor:



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="8dp"
    android:clickable="true"
    android:focusable="true">

    <TextView
        android:id="@+id/tvCrotal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Crotal"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:paddingBottom="4dp" />

    <TextView
        android:id="@+id/tvNombre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nombre"
        android:textSize="24sp"
        android:textStyle="bold"

```

*Ilustración 5*

Como se ve, tenemos los TextView, que nos darán la información del campo deseado, y en la primera línea un LinearLayout que nos agrupará estos TextView en una línea, con la que posteriormente se hará una lista.

## SQLite DatabaseOpenHelper

SQLite es una base de datos relacional integrada que es ligera y eficiente, lo que la hace ideal para aplicaciones móviles. En Android SQLiteOpenHelper es una clase que facilita la creación y gestión de bases de datos de este tipo.

Algunas de sus características son:

- Creación y actualización de la base de datos: proporciona métodos para crear y actualizar la base de datos, manejando automáticamente las versiones.
- Gestion de conexiones, simplifica las conexiones a las bases de datos.

## Gradle

Es una herramienta de automatización de construcción que se utiliza en el desarrollo de aplicaciones Android. Es una parte fundamental de Android Studio. Gradle nos facilita la construcción y la gestión de dependencias y la configuración de múltiples variantes de compilación de una aplicación. Algunas de las características más importantes:

- Depuración colaborativa, permite compartir los resultados de la compilación con el equipo.
- Construcción incremental y automatizada, automatiza y valida el proceso de compilación si algo ha cambiado, incluyendo la verificación del código, la compilación de archivos fuente, la generación de archivos APK y la ejecución de las pruebas.
- Dependencias transitivas, las descarga y las administra automáticamente.
- Compara builds, resalta rápidamente las diferencias entre compilaciones de múltiples proyectos.

[31]

## 4. Técnicas y herramientas

---

### 4.1. Metodologías ágiles

Las metodologías ágiles son enfoques para la gestión y desarrollo de proyectos que promueven la colaboración, flexibilidad y entrega continua de valor. En este proyecto, se ha utilizado la metodología ágil Scrum [5], que es un marco de trabajo que divide el desarrollo en ciclos cortos llamados sprints, normalmente de dos a cuatro semanas, durante los cuales se desarrollan funcionalidades incrementales del producto. Al final de cada Sprint, se revisa y ajusta el trabajo realizado. Esto permite adaptarse rápidamente a los cambios y asegurar que el producto final cumpla con las expectativas del usuario.

Los sprints son periodos de tiempo durante los cuales el equipo de desarrollo trabaja en una lista de tareas específicas. Cada Sprint comienza con una planificación detallada y culmina con una revisión y retrospectiva, lo que facilita la mejora continua y el ajuste de procesos según sea necesario.

### 4.2. Control de versiones

Para el control de versiones se ha utilizado GitHub [21] y Fork.[19]

GitHub es plataforma de desarrollo colaborativo que utiliza Git como sistema de control de versiones. Permite a los desarrolladores gestionar y compartir código, colaborar en proyectos y mantener un historial de cambios detallado. GitHub también ofrece funcionalidades adicionales como la integración continua, revisiones de código y documentación en línea, que son esenciales para el desarrollo eficiente del proyecto.

Por otro lado, Fork es una herramienta gráfica para Git que facilita la gestión de repositorios. Permite realizar commits, ramas y merges de manera más intuitiva en comparación con la línea de comandos de Android Studio. Fork ofrece una interfaz visual que simplifica la resolución de conflictos y el seguimiento de cambios, además de que crear ramas para trabajar en paralelo al proyecto es muy sencillo y cómodo.

### 4.3. Gestión del proyecto

Para la gestión del proyecto, se ha utilizado Zube [35] debido a su integración con GitHub y sus funcionalidades de gestión ágil. Zube permite una organización eficiente del trabajo y una mejor colaboración entre los miembros del equipo, proporcionando una plataforma centralizada para la planificación, ejecución y seguimiento del progreso del proyecto.

Aunque se consideró el uso de Jira, esta opción fue descartada debido a su coste. Jira es una opción viable y recomendada para proyectos más grandes y en entornos empresariales, pero Zube proporciona la funcionalidad necesaria para este proyecto de manera más accesible.

### 4.4. Entorno de desarrollo integrado

Se consideraron varios entornos de desarrollo integrado (IDE) para el desarrollo de la aplicación:

- **Android Studio:** fue elegido el entorno principal de desarrollo debido a su integración completa con herramientas de desarrollo Android, su interfaz visual intuitiva y sus funcionalidades avanzadas. Permite una fácil visualización y edición de la interfaz de usuario, así como la depuración y prueba de la aplicación. Android Studio soporta plugins que extienden su funcionalidad, mejorando aún más la eficiencia del desarrollo.
- **Visual Studio:** utilizado como soporte adicional, principalmente para funciones específicas que complementan el desarrollo en Android Studio. Visual Studio ofrece herramientas avanzadas de depuración y desarrollo multiplataforma que pueden ser útiles para ciertas partes del proyecto.
- **Eclipse:** Aunque se consideró, finalmente no se utilizó debido a la superioridad de Android Studio en términos de integración y usabilidad. Eclipse es una opción sólida para el desarrollo Java, pero Android Studio proporciona un entorno más especializado y optimizado para aplicaciones Android.

### 4.5. Lenguaje de programación

Para llevar a cabo el desarrollo de la aplicación se utilizó Java.

Es un lenguaje de programación robusto y ampliamente utilizado en el desarrollo de aplicaciones Android. Aunque se consideró el uso de Kotlin, la falta de experiencia en este lenguaje inclinó la balanza hacia Java, asegurando un desarrollo más fluido y

eficiente. Java ofrece una vasta cantidad de recursos y bibliotecas que facilitan el desarrollo y mantenimiento de la aplicación.

## **4.6. Edición de texto**

Para la redacción de la documentación y otros textos se evaluaron varias opciones, Microsoft Word, Open Office o LaTeX, pero finalmente la herramienta elegida fue Microsoft Word, el cual fue elegido por su familiaridad y funcionalidad completa, fue la opción preferida por su facilidad de uso y capacidades avanzadas de edición. Microsoft Word permite una colaboración eficiente mediante el seguimiento de cambios y comentarios, lo que es esencial para la documentación colaborativa, como en ese caso ha sido para las correcciones y comentarios del tutor.

## **4.7. Hosting del repositorio [21]**

Para el hosting del repositorio se han considerado GitHub, BitBucket y GitLab, pero finalmente la elegida fue GitHub.

Es la plataforma web de hospedaje de repositorios por excelencia. Ofrece todas las funcionalidades de Git, revisión de código, documentación, bug tracking, gestión de tareas, wikis, red social y numerosas integraciones con otros servicios. Es gratuita para proyectos open source y proporciona una infraestructura robusta para el desarrollo colaborativo.

Utilizamos GitHub como plataforma principal donde hospedamos el código del proyecto, la gestión de proyecto (gracias a Zube) y la documentación. Además, el repositorio está integrado con varios servicios de integración continua de Android Studio, lo que facilita la entrega continua y la calidad del código

## **4.8. Gestor de referencias**

Para la gestión de referencias bibliográficas, se ha utilizado Zotero [4], es un software de gestión de referencias bibliográficas de código abierto que permite recopilar, organizar, citar y compartir investigaciones.

Zotero facilita la importación de referencias desde diversos formatos y fuentes, incluyendo bases de datos académicas y páginas web. Además, permite la creación automática de bibliografías en diversos estilos de citación, lo que agiliza el proceso de documentación y asegura la consistencia en las referencias utilizadas a lo largo del proyecto.

## 5. Aspectos relevantes del desarrollo del proyecto

---

En este apartado se recogen las partes más importantes que se han tenido en cuenta a la hora del desarrollo del proyecto, empezando por las primeras ideas, los problemas y el resultado final

### 5.1. Inicio del proyecto

La idea del proyecto surgió de la experiencia personal y familiar en el ámbito de la ganadería. Toda mi familia se ha dedicado siempre a la ganadería, y desde muy pequeña he observado de primera mano todas las dificultades y atrasos que tiene este sector, porque como se comentaba en la introducción, a pesar de ser un sector primario y vital es uno de los grandes olvidados.

Por ello, al ver los problemas a la hora de hacer registros, el tener que hacer desplazamientos a una ciudad cada vez que pasa algo con un animal, o el tener que hacer todo manualmente llevándolos más trabajo aún del que da de por sí una ganadería, se me ocurrió la idea de desarrollar una solución que permitiera gestionar toda la información administrativa desde dispositivos móviles, facilitando el acceso y la organización de datos para los ganaderos.

Con esta idea en mente, presenté el proyecto a mi tutor, quien aprobó mi propuesta y comenzamos el desarrollo bajo la metodología ágil de sprints.

### 5.2. Metodología de trabajo

Desde el primer momento de empezar el proyecto, se propuso el utilizar la metodología ágil por sprints, para intentar ser lo más constantes y responsables posibles a la hora de cumplir con los tiempos.

Se siguió la metodología scrum, aunque no se ha podido cumplir del todo dado que no somos un equipo de varias personas, si no que al final es un proyecto educativo e individual, pero junto con el tutor se intentó seguir al máximo posible reuniéndonos al final de cada sprint para enseñar el funcional y programar las siguientes tarjetas. En líneas generales, lo que íbamos haciendo es:

- Seguir una planificación por sprints y revisiones.
- Establecer una duración media de los sprints en dos semanas.
- Cada vez que terminaba un sprint se enseñaba lo que se había hecho y lo que no.
- Las reuniones de final de sprint nos servían para el punto anterior y para programar el siguiente sprint y concertar la siguiente reunión.
- Por cada planificación del sprint se cogían tareas del backlog para realizarlas, o en el caso de que se nos ocurriese una nueva idea, se añadía una nueva tarjeta.

- A estas tarjetas les estimábamos un tiempo, una prioridad y un sprint en el tablero Kanban Board de Zube.
- Para controlar los tiempos y la evolución de nuestro proyecto se hacía un seguimiento a través de los gráficos burndown.

Para el desarrollo de la aplicación de Android lo primero que se hizo fue crear las principales pantallas para ir haciendo toma de contacto y empezar a hacer flujos para comprobar el funcionamiento. Una vez teníamos esto, se continuo con la base de datos.

En una primera instancia, se empezó a hacer la base de datos con SQLite, pero se contempló la opción de hacer la base de datos externa a la aplicación con MySQL, para ello se probó con XAMPP a realizar una conexión a una base de datos externa a nuestra aplicación y funcionaba, pero finalmente se descartó porque para que exista esa conexión necesitábamos que la app estuviese en conexión con internet y al ser una aplicación para la ganadería es muy probable que la mayoría de las veces que se utilice el dispositivo móvil no va a tener opción de conectarse a ninguna red de internet. Por lo que en este momento se decidió volver a cambiarlo y continuar con nuestra base de datos SQLite.

Para terminar con el desarrollo, se implementó que las pantallas de la aplicación se vean iguales en todos los dispositivos móviles, ya que al trabajar con el emulador esto no se comprobaba del todo, asique se decidió empezar a hacer pruebas en dispositivos físicos de diferentes tamaños y marcas para comprobar el funcionamiento.

### 5.3. Formación

Para poder llevar a cabo la realización de este proyecto necesitaba de muchos conocimientos previos que no tenía, por lo que busqué recursos educativos para poder llevar a cabo este proceso. Estos fueron:

Para la formación de AndroidStudio:

- Curso de Android desde cero [1]
- Aspectos básicos de Android con Compose [2].
- Estructura de un proyecto Android Studio [3].

Para la formación de la base de datos:

- Cómo guardar datos con SQLite [25]
- [SQLite en Android: creación y acceso base de datos e inserción de registros – Academia Android](#)
- [SQLiteDatabase | Android Developers](#)

Para la formación de Zube:

[Zube, un sistema de gestión de proyectos para programadores \(www.whatsnew.com\)](http://www.whatsnew.com)

Para la formación de GitHub:

Curso de Git para principiantes [23]

En relación a la resolución de errores que iban saliendo hice mucho uso de [Stack Overflow en español](#) y también del buscador de código con inteligencia artificial [Your Personalized AI Assistant](#).

## 5.4. Desarrollo del proyecto

Una vez teniendo claro lo que se quería implementar y el tutor lo había aprobado, se empezó a estudiar las diferentes herramientas para determinar cómo se podía llevar a cabo. El primer punto para discutir era que entorno integrado usaría y en que lenguaje se programaría.

Como se determinó que iba a ser una aplicación móvil, se decidió que se usaría Android Studio para llevar a cabo el desarrollo.

El primer paso fue crear el proyecto, con actividades vacías, y se empezó a configurar para que usara Gradle, y el lenguaje de programación java. Me decidí por este lenguaje dado que con Kotlin no había tenido ninguna experiencia y Java se ha estudiado en varias asignaturas del grado como son, entre otras, Metodología de la programación y Sistemas distribuidos.

Se empezó a crear el módulo de animales, con sus altas y bajas. Primero de todo se crearon todas las actividades más importantes con sus correspondientes controladores para ir haciendo flujos y comprobando el correcto funcionamiento de los botones entre sí. Una vez hecho esto, había que empezar a mostrar datos, para ello se construyó la base de datos con SQLite. En esta se guardarán en el dispositivo donde se ejecute la aplicación los datos insertados por cada animal, y podrán ser consultados por el usuario en todo momento, ya que como se mencionó anteriormente, no es necesaria la conexión a internet para poder hacer uso de la aplicación.



## 6. Trabajos relacionados

---

Como se comentaba anteriormente, ya existen aplicaciones y softwares para la gestión de las ganaderías, el problema es que la gran mayoría tienen un precio demasiado elevado o son muy complicadas de entender y de usar fácilmente.

En este apartado se van a mostrar y explicar algunos de los trabajos que están relacionados con el proyecto que se ha implementado:

- **AppGRO, Agricultura, Ganadería y Recursos Organizados :**

Es un proyecto diseñado para la universidad de Cataluña, como trabajo de fin de grado de ingeniería informática.

Lo que propone es crear una aplicación multiplataforma, ósea que sirva tanto para iOS como para Android, que permita tanto a los agricultores como a los ganaderos gestionar su modelo de negocio de tal manera que se vuelque el beneficio en la “España vaciada”.

Este proyecto también está basado en una metodología ágil como es scrum y sigue el patrón de software Container/View al igual que el mío.

También destacan que, en la fase de desarrollo, gracias a la previsión y amplitud de márgenes temporales han podido llegar a realizar todas las funciones especificadas desde el inicio.

- **StockLive, una herramienta para la gestión de ganado vacuno de leche.**

Esta herramienta hace que se puedan registrar todas las reproducciones dentro de una ganadería, por lo tanto, esto facilita al usuario el determinar cuándo tiene que realizar las inseminaciones, secados y cuando podemos prever el parto.

El usuario tiene también disponible la opción de generar informes para gestionar los trabajos en la granja y llevar así un control junto con un calendario.

En esta aplicación, para el tratamiento de los datos, se importan los de las asociaciones AFRICOR Y CONAFE, que llevan el control lechero y anima de granjas de vacuno lechero.

- **GoBees, monitorización del estado de una colmena mediante la cámara de un smartphone [18].**

Aunque esta aplicación trata de colmenas y abejas en vez de vacas y ganaderías, es bastante similar a lo que se quiere realizar. Además, sigue la misma metodología, usa el mismo lenguaje de programación y el mismo entorno de desarrollo integrado.

Por otro lado, se trata de un trabajo de fin de grado de la misma universidad, por lo que ha servido mucho de cara a ser una “guía” para desarrollar y gestionar este proyecto, tanto para la parte de la memoria como para el desarrollo.

## 7. Conclusiones y líneas de trabajo futuras

---

Una vez terminado el proyecto, se va a hacer un análisis final donde sacaremos conclusiones acerca de cómo ha ido el desarrollo y se estudiarán las posibles líneas de trabajo futuras.

### 7.1. Conclusiones

El principal objetivo por el que se empezó este proyecto ha sido alcanzado, que fue el poder gestionar una ganadería desde nuestro dispositivo móvil, pero no se ha podido alcanzar todo lo que se quería.

Actualmente en el proyecto se pueden añadir, eliminar, editar y consultar los diferentes animales, pero no se puede hacer nada más allá de eso, ya que la organización no ha sido del todo satisfactoria por lo que no he dispuesto del tiempo necesario para poder realizar más módulos, como hubiese sido la alimentación, por ejemplo.

Se cumple con el objetivo de una interfaz sencilla y fácil de usar, bastante intuitiva que no da lugar a errores por dudas. Esto va a permitir que diferentes personas con diferentes conocimientos y habilidades sean capaces de utilizar la aplicación diseñada sin tener ningún tipo de dificultad o sin que sea necesario tener conocimientos en tecnologías para poder hacerlo.

Se ha cumplido con los objetivos personales, he obtenido bastantes conocimientos acerca de las aplicaciones móviles y además he contribuido al sector al que mi familia lleva dedicada toda la vida.

Se ha aprendido muchos conocimientos acerca de la herramienta GitHub, Fork y repositorios. También se han adquirido conocimientos acerca de scrum y se ha intentado cumplir con el plan de desarrollo que se propuso en un inicio, aunque a veces los tiempos no han ido del todo acordes.

### 7.2. Líneas de trabajo futuras

En un futuro, se quiere que esta aplicación cumpla con los requisitos establecidos para poder lanzarla a GooglePlay, así como implementar más módulos para gestionar más cosas de una ganadería que no sean solamente los animales.

Además, estando en la comunidad autónoma de Castilla y León, donde la ganadería es un sector muy importante y del que dependen muchas familias, estaría bien implementar una nueva funcionalidad en la aplicación de tal manera que a través del listado que se descargan los ganaderos en el portal de JCYL del módulo ganadero, se pudiese hacer una carga masiva de datos, aparte de añadir funciones para que la aplicación no sea solamente para bovinos.

# Bibliografía

---

- [1] «Curso Android desde cero con Android Studio - YouTube». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.youtube.com/watch?v=tyx05coXixw&list=PLyvsggKtwbLX06iMtXnRGX5lyjiiMaT2y>
- [2] «Curso sobre los aspectos básicos de Android con Compose | Aspectos básicos de Android con Compose», Android Developers. Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://developer.android.com/courses/android-basics-compose/course?hl=es-419>
- [3] «Android Studio.-Estructura de un proyecto en Android Studio - YouTube». Accedido: 9 de julio de 2024. [En línea]. Disponible en: [https://www.youtube.com/watch?v=tQQ3\\_mHOvz4](https://www.youtube.com/watch?v=tQQ3_mHOvz4)
- [4] «Zotero | Your personal research assistant». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.zotero.org/>
- [5] L. Jiménez, «8 pasos esenciales para planificar un proyecto de desarrollo de software |». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://leojimzdev.com/8-pasos-esenciales-para-planificar-un-proyecto-de-desarrollo-de-software/>
- [6] L. Jiménez, «8 pasos esenciales para planificar un proyecto de desarrollo de software |». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://leojimzdev.com/8-pasos-esenciales-para-planificar-un-proyecto-de-desarrollo-de-software/>
- [7] *Android studio 4.1 conectando botones, EditText y TextView con el MainActivity | Marzo 2021*, (4 de marzo de 2021). Accedido: 7 de mayo de 2024. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=wwLnOlzzAMM>
- [8] «Android Studio.-Estructura de un proyecto en Android Studio - YouTube». Accedido: 9 de julio de 2024. [En línea]. Disponible en: [https://www.youtube.com/watch?v=tQQ3\\_mHOvz4](https://www.youtube.com/watch?v=tQQ3_mHOvz4)
- [9] M. Piano, «Answer to “¿Cómo cambiar el tamaño a mi imagen en Android?”», Stack Overflow en español. Accedido: 30 de mayo de 2024. [En línea]. Disponible en: <https://es.stackoverflow.com/a/334661>
- [10] N. IA, «¿Cómo cambiar el tamaño a mi imagen en Android?», Stack Overflow en español. Accedido: 30 de mayo de 2024. [En línea]. Disponible en: <https://es.stackoverflow.com/q/43396>
- [11] *Conectar Mysql a Android Studio (Insertar y mostrar datos)*, (13 de octubre de 2020). Accedido: 30 de mayo de 2024. [En línea Video]. Disponible en: [https://www.youtube.com/watch?v=C1GlrLi\\_vU](https://www.youtube.com/watch?v=C1GlrLi_vU)
- [12] «Curso sobre los aspectos básicos de Android con Compose | Aspectos básicos de Android con Compose», Android Developers. Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://developer.android.com/courses/android-basics-compose/course?hl=es-419>

- [13] Movistar, «Fibra óptica en casa sin permanencia. Movistar Internet en casa». Accedido: 4 de julio de 2024. [En línea]. Disponible en: <https://www.movistar.es/internet/>
- [14] M. García, «MVC (Modelo-Vista-Controlador): ¿qué es y para qué sirve?». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve>
- [15] «mvc-modelo-vista-controlador.png (800×800)». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://codingornot.com/wp-content/uploads/2017/10/mvc-modelo-vista-controlador.png>
- [16] «Linea principal de trabajo | Kanban». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://zube.io/tfg-4/app-movil-para-la-gestion-de-una-explotacion-ganadera-bovina/w/linea-principal-de-trabajo/kanban>
- [17] Unknown, «INFOTELECO: Tutorial: Cómo conectarse a MySQL desde Android con XAMPP», INFOTELECO. Accedido: 30 de mayo de 2024. [En línea]. Disponible en: <https://infoteleco.blogspot.com/2015/06/tutorial-como-conectarse-mysql-desde.html>
- [18] «go-bees/app/src/main/java/com/davidmiguel/gobees at master · davidmigloz/go-bees», GitHub. Accedido: 7 de mayo de 2024. [En línea]. Disponible en: <https://github.com/davidmigloz/go-bees/tree/master/app/src/main/java/com/davidmiguel/gobees>
- [19] «Fork - a fast and friendly git client for Mac and Windows». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://git-fork.com/>
- [20] S. Sánchez y A. Zarzalejos, «Esto es lo que realmente gana un desarrollador de aplicaciones en España: hasta 7.000 euros de diferencia entre provincias», Business Insider España. Accedido: 4 de julio de 2024. [En línea]. Disponible en: <https://www.businessinsider.es/cuanto-gana-desarrollador-aplicaciones-espana-476483>
- [21] «elenarecio/TFG: Diseño y programación de una aplicación móvil para ayudar a los ganaderos (explotaciones bovinas) con las gestiones cotidianas más comunes. Sencilla y fácil de usar para que pueda ser usada de forma rápida y correcta por todo tipo de edades.», GitHub. Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://github.com/elenarecio/TFG>
- [22] E. Foundation, «Eclipse Downloads | The Eclipse Foundation». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.eclipse.org/downloads/>
- [23] *Curso de Git y Github para Principiantes (2022)*, (17 de enero de 2022). Accedido: 9 de julio de 2024. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=CK5ZcKZsMRs>
- [24] «Curso Android desde cero con Android Studio - YouTube». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.youtube.com/watch?v=tyx05coXixw&list=PLyvsggKtwbLX06iMtXnRGX5lyjiMaT2y>
- [25] «Cómo usar SQL para leer y escribir en una base de datos», Android Developers. Accedido: 7 de mayo de 2024. [En línea]. Disponible en: <https://developer.android.com/codelabs/basic-android-kotlin-compose-sql?hl=es-419>

- [26] Jorgesys, «Answer to “¿Cómo cambiar el tamaño a mi imagen en Android?”», Stack Overflow en español. Accedido: 30 de mayo de 2024. [En línea]. Disponible en: <https://es.stackoverflow.com/a/43405>
- [27] «mvc-modelo-vista-controlador-portada.png (770×458)». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://codingornot.com/wp-content/uploads/2017/10/mvc-modelo-vista-controlador-portada.png>
- [28] *MySQL con Android Studio (Web Services)*, (8 de octubre de 2018). Accedido: 29 de mayo de 2024. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=ZmOOMCSVe20>
- [29] *MySQL con Android Studio (Insertar datos)*, (9 de octubre de 2018). Accedido: 30 de mayo de 2024. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=i-eKtws0EnI>
- [30] «Patrón Modelo Vista Controlador | Marco de Desarrollo de la Junta de Andalucía». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>
- [31] «Qué es Gradle y sus características | OpenWebinars». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://openwebinars.net/blog/que-es-gradle/>
- [32] fjmeler, «Viabilidad de proyectos», UGRemprendedora. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://ugremprendedora.ugr.es/viabilidad-de-proyectos/>
- [33] «YouTube». Accedido: 8 de julio de 2024. [En línea]. Disponible en: <https://www.youtube.com/>
- [34] «Samsung Galaxy S8 Smartphone, 4GB RAM, 64GB, 12MP, Android 9, (Versión española: incluye Samsung Pay, compatibilidad de redes), Negro, 5.8" : Amazon.es: Electrónica». Accedido: 4 de julio de 2024. [En línea]. Disponible en: <https://www.amazon.es/Samsung-Galaxy-Smartphone-Versi%C3%B3n-espa%C3%B1ola/dp/B06XXFHG6J>
- [35] «Zube, un sistema de gestión de proyectos para programadores». Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.whatsnew.com/2015/10/04/zube-un-sistema-de-gestion-de-proyectos-para-programadores/>