**Proyecto Final De Programación Nivel Integrador “Adoptapp”**

Andrés Felipe Escobar Vásquez

Jorge Andrés Restrepo Cataño

Alejandro Pérez Ramírez

Julián David Giraldo Murillo

Ministerio de Tecnologías de la Información y Comunicaciones de Colombia

Grupo 66 (UdeA)

Eder Lara Trujillo

2025

Tabla de contenido

[Introducción 2](#_Toc194228352)

[Justificación 3](#_Toc194228353)

[Alcance del Proyecto 4](#_Toc194228354)

[Objetivos del Proyecto 5](#_Toc194228355)

[Objetivo General 5](#_Toc194228356)

[Objetivos Específicos 5](#_Toc194228357)

[Diseño y Desarrollo 5](#_Toc194228358)

[Diseño de la Base de Datos 5](#_Toc194228359)

[Desarrollo de la Aplicación 10](#_Toc194228360)

[Conclusiones y Recomendaciones 11](#_Toc194228361)

# Introducción

Este documento presenta el desarrollo de una aplicación web para la adopción de perros y gatos, cuyo propósito es reducir el índice de mascotas en estado de abandono y fomentar la adopción responsable. La plataforma conecta a personas interesadas en adoptar con albergues, refugios y otros particulares que buscan dar en adopción a sus mascotas, ofreciendo una manera sencilla y organizada de visualizar las opciones disponibles.

La aplicación permite a los usuarios registrarse con diferentes roles, incluyendo adoptantes, personas que ofrecen mascotas en adopción, tiendas y cuidadores. Actualmente, la plataforma facilita el registro, inicio de sesión y la visualización de mascotas para adopción, con planes futuros de integrar un sistema de seguimiento post-adopción y la posibilidad de registrar tiendas y cuidadores para ofrecer sus servicios.

El desarrollo del sistema se basa en tecnologías como Java y JavaScript con los frameworks Spring Boot, React, y gestión de bases de datos con MySQL, garantizando una experiencia interactiva y dinámica. A largo plazo, se busca implementar un proceso de adopción más riguroso, incluyendo un seguimiento periódico de las mascotas adoptadas para asegurar su bienestar.

# Justificación

En Colombia, el abandono de mascotas es una problemática significativa que afecta a millones de animales. Se estima que alrededor de 3 millones de perros y gatos se encuentran en situación de calle en el país. Esta cifra refleja la magnitud del desafío que enfrentan las organizaciones y comunidades dedicadas al bienestar animal.

En Bogotá, por ejemplo, se calcula que aproximadamente 133.637 perros y gatos vivían en las calles en 2022, de los cuales cerca del 10% fueron víctimas de abandono directo por parte de sus dueños. Además, durante el mismo año, se registraron 52 casos de perros abandonados en el sistema de transporte público TransMilenio.

A nivel nacional, la Fiscalía General de la Nación reportó que, entre enero y septiembre de 2023, se registraron 1.002 casos de maltrato animal, lo que evidencia la gravedad de la situación y la necesidad de implementar soluciones efectivas.

Estas estadísticas subrayan la urgencia de desarrollar iniciativas que promuevan la adopción responsable y mejoren la conexión entre los adoptantes y los refugios. Al proporcionar una plataforma que facilite la visibilidad de las mascotas disponibles y agilice el proceso de adopción, se puede contribuir significativamente a reducir el número de animales en situación de abandono y fomentar una cultura de tenencia responsable en Colombia.

# Alcance del Proyecto

Inicialmente, la aplicación tendrá cobertura en el **Valle de Aburrá**, en el departamento de Antioquia, Colombia, focalizándose en conectar a adoptantes con albergues, refugios y otros usuarios que ofrecen mascotas en adopción. A futuro, se proyecta ampliar la cobertura a otras regiones del país, consolidándose como la plataforma líder en adopción responsable de mascotas y como un puente entre entidades gubernamentales y privadas dedicadas al bienestar animal.

El sistema permitirá:

* **Registro de usuarios**: Diferentes roles se podrán registrar, incluyendo adoptantes, personas que ofrecen mascotas en adopción, tiendas y cuidadores temporales.
* **Inicio de sesión y gestión de perfiles**: Permitiendo la autenticación segura de cada usuario.
* **Visualización y filtrado de mascotas**: Los usuarios podrán buscar y filtrar mascotas por ubicación dentro del Valle de Aburrá, facilitando la búsqueda de una opción adecuada.
* **Gestión de datos**: Toda la información se almacenará y gestionará a través de una base de datos relacional (MySQL).

Adicionalmente, se contempla la implementación de mejoras futuras, tales como:

* **Verificación biométrica**: Para garantizar la identidad de adoptantes y cuidadores, reforzando el proceso de adopción responsable.
* **Seguimiento post-adopción**: Un sistema que, en versiones posteriores, permitirá dar seguimiento al bienestar de las mascotas adoptadas.

# Objetivos del Proyecto

## Objetivo General

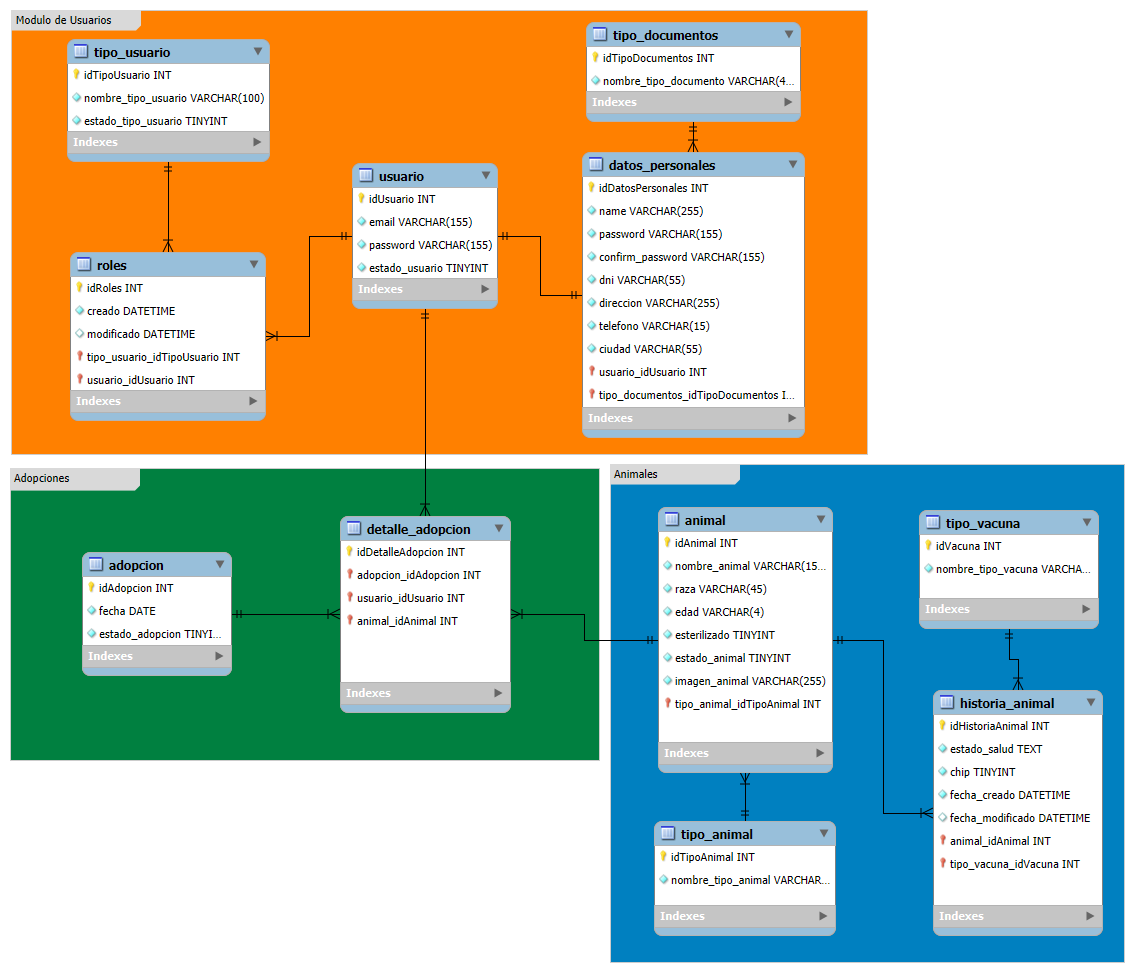
Desarrollar una aplicación web para la adopción de perros y gatos que conecte adoptantes con refugios, albergues y otros oferentes de mascotas, reduciendo el abandono y promoviendo la adopción responsable.

## Objetivos Específicos

* Implementar un sistema orientado a objetos en Java (Spring Boot) para gestionar la información de usuarios, mascotas y adopciones.
* Diseñar e implementar una base de datos relacional en MySQL.
* Implementar funcionalidades de usuario (registro, inicio de sesión y gestión de perfiles).
* Garantizar una experiencia de usuario óptima mediante una interfaz web responsiva e interactiva desarrollada con React y JavaScript.
* Planificar futuras mejoras, como la verificación biométrica y el seguimiento post-adopción.

# Diseño y Desarrollo

Diseño de la Base de Datos

La base de datos de Adoptapp está diseñada para gestionar la información de usuarios, mascotas, adopciones, albergues y tiendas. A continuación, se describen las principales tablas y sus relaciones: 

**Tablas y relaciones en la base de datos**

La base de datos del proyecto se denomina **adoptApp** y está diseñada para gestionar la adopción de mascotas. A continuación, se describen las principales tablas y sus relaciones.

1. **Tabla “usuario”**

**Propósito**: Almacena la información básica de los usuarios registrados en el sistema.

**Principales columnas**:

* idUsuario (INT, PRIMARY KEY, AUTO\_INCREMENT): Identificador único.
* email (VARCHAR 155, UNIQUE): Correo electrónico del usuario.
* password (VARCHAR 155): Contraseña encriptada.
* estado\_usuario (TINYINT, DEFAULT 1): Indica si el usuario está activo.

**Relaciones**: Se relaciona con “datos\_personales” y “roles”.

1. **Tabla “tipo\_documentos”**

**Propósito**: Almacena los tipos de documentos de identificación.

**Principales columnas**:

* idTipoDocumentos (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre\_tipo\_documento (VARCHAR 45): Nombre del tipo de documento.

**Relaciones**: Se relaciona con “datos\_personales”.

1. **Tabla “datos\_personales”**

**Propósito**: Contiene información personal de los usuarios.

**Principales columnas**:

* idDatosPersonales (INT, PRIMARY KEY, AUTO\_INCREMENT)
* name, dni, direccion, telefono, ciudad (VARCHAR)
* usuario\_idUsuario (INT, FOREIGN KEY a usuario.idUsuario)
* tipo\_documentos\_idTipoDocumentos (INT, FOREIGN KEY a tipo\_documentos.idTipoDocumentos)

**Relaciones**: Se relaciona con “usuario” y “tipo\_documentos”.

1. **Tabla “tipo\_usuario”**

**Propósito**: Define los tipos de usuarios (adoptantes, administradores, etc.).

**Principales columnas**:

* idTipoUsuario (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre\_tipo\_usuario (VARCHAR 100)
* estado\_tipo\_usuario (TINYINT, DEFAULT 1)

**Relaciones**: Se relaciona con “roles”.

1. **Tabla “roles”**

**Propósito**: Almacena los roles asignados a los usuarios.

**Principales columnas**:

* idRoles (INT, PRIMARY KEY, AUTO\_INCREMENT)
* tipo\_usuario\_idTipoUsuario (INT, FOREIGN KEY a tipo\_usuario.idTipoUsuario)
* usuario\_idUsuario (INT, FOREIGN KEY a usuario.idUsuario)

**Relaciones**: Se relaciona con “tipo\_usuario” y “usuario”.

1. **Tabla “adopción”**

**Propósito**: Registra las adopciones realizadas.

**Principales columnas**:

* idAdopcion (INT, PRIMARY KEY, AUTO\_INCREMENT)
* fecha (DATE)
* estado\_adopcion (TINYINT, DEFAULT 1)

**Relaciones**: Se relaciona con “detalle\_adopcion”

1. **Tabla “tipo\_animal”**

**Propósito**: Define los tipos de animales disponibles (perros, gatos, etc.).

**Principales columnas**:

* idTipoAnimal (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre\_tipo\_animal (VARCHAR 155)

**Relaciones**: Se relaciona con “animal”.

1. **Tabla “animal”**

**Propósito**: Contiene la información de los animales en adopción.

**Principales columnas**:

* idAnimal (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre\_animal, raza, edad, esterilizado, estado\_animal, imagen\_animal (VARCHAR)
* tipo\_animal\_idTipoAnimal (INT, FOREIGN KEY a tipo\_animal.idTipoAnimal)

**Relaciones**: Se relaciona con “tipo\_animal”, “detalle\_adopcion” y “historia\_animal”.

1. **Tabla “detalle\_adopcion”**

**Propósito**: Registra el detalle de cada adopción.

**Principales columnas**:

* idDetalleAdopcion (INT, PRIMARY KEY, AUTO\_INCREMENT)
* adopcion\_idAdopcion (INT, FOREIGN KEY a adopcion.idAdopcion)
* usuario\_idUsuario (INT, FOREIGN KEY a usuario.idUsuario)
* animal\_idAnimal (INT, FOREIGN KEY a animal.idAnimal)

**Relaciones**: Se relaciona con “adopcion”, “usuario” y “animal”.

1. **Tabla “tipo\_vacuna”**

**Propósito**: Contiene los tipos de vacunas.

**Principales columnas**:

* idVacuna (INT, PRIMARY KEY, AUTO\_INCREMENT)
* nombre\_tipo\_vacuna (VARCHAR 155)

**Relaciones**: Se relaciona con “historia\_animal”.

1. **Tabla “historia\_animal”**

**Propósito**: Registra el historial médico de los animales.

**Principales columnas**:

* idHistoriaAnimal (INT, PRIMARY KEY, AUTO\_INCREMENT)
* estado\_salud, chip, fecha\_creado, fecha\_modificado (TEXT, TINYINT, DATETIME)
* animal\_idAnimal (INT, FOREIGN KEY a animal.idAnimal)
* tipo\_vacuna\_idVacuna (INT, FOREIGN KEY a tipo\_vacuna.idVacuna)

**Relaciones**: Se relaciona con “animal” y “tipo\_vacuna”.

Cada una de estas tablas está relacionada para garantizar la integridad de la información y permitir una gestión eficiente de la adopción de mascotas.

## Desarrollo de la Aplicación

* Implementación de autenticación y gestión de usuarios en Java.
* Creación de la API REST para la interacción con la base de datos.
* Implementación de funcionalidades de búsqueda y filtrado de mascotas.
* Desarrollo de la interfaz web utilizando tecnologías modernas.

**Funcionalidades Clave**

* La aplicación servirá como puente entre adoptantes y albergues, perreras y cuidadores temporales.
* Los cuidadores temporales podrán registrarse de manera independiente, aunque en futuras versiones se integrará un sistema de validación de identidad para mayor seguridad.
* Se implementará un sistema de calificación y reseñas para los albergues y cuidadores.
* No se habilitará comunicación directa dentro de la plataforma; cada albergue o cuidador tendrá en su perfil su información de contacto.
* No se realizará seguimiento post-adopción desde la plataforma; cada albergue definirá si desea implementarlo en sus protocolos.
* No se emitirán certificaciones de adopción; la plataforma solo actuará como intermediaria.

# Conclusiones y Recomendaciones

* Evaluación del cumplimiento de los objetivos.
* Descripción de los principales retos enfrentados y soluciones aplicadas.
* Posibles mejoras y futuras actualizaciones del sistema.