

# PROYECTO DE FIN DE CICLO



Autor:

Alba Núñez Arranz

Curso 2022-2023

Desarrollo de Aplicaciones Web

Github: <https://github.com/Alba1999/ProyectoDAW.git>



<b>Autor:</b>	<b>1</b>
Alba Núñez Arranz	1
<b>Índice:</b>	<b>2</b>
Introducción	3
<b>Justificación del proyecto y objetivos</b>	<b>4</b>
<b>Planificación</b>	<b>5</b>
<b>Parte experimental</b>	<b>8</b>
Análisis:	8
Diseño:	11
Implementación y pruebas:	13
Implantación y documentación:	16
<b>Resultados y discusión:</b>	<b>18</b>
Tiempos	18
Dificultades	19
Mejoras e implementaciones	20
<b>Conclusiones</b>	<b>21</b>
<b>Bibliografía y referencias</b>	<b>22</b>
Protectoras que han colaborado en investigación	22
<a href="https://albaonline.org/">https://albaonline.org/</a>	22
<a href="https://www.anaaweb.org/">https://www.anaaweb.org/</a>	22
<a href="https://www.huellaranda.org/">https://www.huellaranda.org/</a>	22
<a href="https://www.fundacion-affinity.org/">https://www.fundacion-affinity.org/</a>	22
Diseños:	22
Herramientas:	22
<b>Anexos</b>	<b>23</b>
Investigación:	23



## Introducción

El proyecto que desarrollé es un sitio web que permite administrar de manera eficiente todos los animales de una protectora, incluyendo gatos y perros, que se encuentren en diferentes estados, como aquellos que están pendientes de un hogar, en trámites de adopción o en acogida temporal.

La idea principal del proyecto es que ofrezca la oportunidad de ver un pequeño registro de los datos del animal con los datos principales, su estado, personalidad,... que ofrecerá la oportunidad al usuario interesado en poder saber más.

Además, la aplicación cuenta con un sistema de avisos para los animales que estén en peligro de ser sacrificados, con el fin de darles visibilidad para que tengan una segunda oportunidad y conseguirles un hogar de manera rápida y efectiva. La web ofrece la posibilidad de que los usuarios interesados en adoptar un animal puedan informarse de manera sencilla, consultando la información de cada uno de ellos, y contactando directamente con la protectora para iniciar los diferentes procesos.

En resumen, esta aplicación es una herramienta para mejorar la gestión y cuidado de los animales de una protectora, lo que permitirá reducir la tasa de abandono y mejorar la calidad de vida de estos seres que tanto lo necesitan.



## Justificación del proyecto y objetivos

Mi idea del proyecto surge debido a la gran tasa de abandono de animales en España, que es una de las más altas de Europa, el objetivo principal de la aplicación es fomentar la adopción responsable, promoviendo la concienciación y ofreciendo información detallada sobre cada uno de ellos. Según las cifras recogidas por el estudio "Él nunca lo haría 2022" de la Fundación Affinity, alrededor de 285.000 perros y gatos son recogidos cada año por protectoras en todo el territorio nacional, y estas se encuentran desbordadas.

Año tras año las nefastas cifras en España se traducen en más de 700 animales cada día que terminan en la calle o perreras, según los datos del estudio de abandono y adopción 2021 que recoge la Fundación Affinity.

Creo que es realmente importante profundizar en este proyecto, aunque no tenga unas opciones reales para venderse pero si puede ser algo realmente útil, la mayoría de protectoras no tienen suficientes recursos económicos pero es realmente necesario tener un sitio web que les permita y facilite que las gente les conozca y sepa más sobre los animales o para animarse a colaborar.

Los objetivos del proyecto son crear una aplicación capaz de administrar los animales que están sin hogar incluyendo un sistema de avisos para animales que van ser sacrificados para intentar facilitar la gestión de animales abandonados.



## Planificación

Esta es la planificación que se sigue, dividida en 5 sprints diferentes:

SPRINTS	FECHAS	TAREAS
1º Sprint	17 mar - 31 mar	<ul style="list-style-type: none"><li>- Diseñar el logo</li><li>- Documentación</li><li>- Preparación del entorno</li><li>- Crear la introducción</li></ul>
2º Sprint	31 mar - 14 abr	<ul style="list-style-type: none"><li>- Investigación</li><li>- Análisis</li><li>- Modelo E/R</li><li>- Diseño de tablas</li><li>- Diseño del front</li><li>- Hacer las imágenes del front</li></ul>
3º Sprint	14 abr - 28 abr	<ul style="list-style-type: none"><li>- Modelo relacional</li><li>- Explicar las relaciones</li><li>- Instalar MySQL y crear el fichero</li><li>- Redactar la investigación</li><li>- Explicar las tecnologías</li></ul>
4º Sprint	28 abr - 12 may	<ul style="list-style-type: none"><li>- Crear datos de carga</li><li>- Terminar el front</li><li>- Desarrollo de código</li><li>- Modificar documentación</li><li>- Explicar los lenguajes</li><li>- Documentación de los sprints</li></ul>
5º Sprint	12 may - 26 may	<ul style="list-style-type: none"><li>- Pasar el front a componentes</li><li>- Hacer servicios en Angular</li><li>- Crear las sociales</li><li>- Insertar widgets de las redes sociales</li><li>- Testeo de conexión con la base de datos</li><li>- Hacer la documentación</li><li>- Manual</li></ul>



Durante el primer sprint me enfoqué en llevar a cabo algunas tareas iniciales como fue la configuración del entorno de trabajo. Esto implicó la instalación y la configuración de las herramientas y programas necesarios. Entre otras tareas estuvo la creación de los logos para el proyecto, utilizando Canvas. Otra tarea en la que me concentré fue el desarrollo de la introducción del proyecto y la documentación, intentando establecer una base que incluyera los objetivos y la metodología. Además, invertí tiempo en el desarrollo de ideas que fueron convertidas en tareas funcionales y accesibles.

Durante el segundo sprint me enfoque en la investigación tanto de los lenguajes como del propio tema del proyecto, contactando con protectoras reales y gente que colabora con ellos. Gracias a toda esta labor de investigación pude hacer un diseño de modelo E/R adaptable para una protectora que pueda ser más grande o más pequeña teniendo necesidades diferentes. También me ocupe del diseño que quería que tuviera la interfaz gráfica.

En el tercer sprint decidí enfocarme en el desarrollo del modelo relacional y las relaciones que se iban a establecer y cómo solucionarlas de la forma más eficiente. También estuve haciendo la instalación y configuración de MySQL, además de redactar el punto sobre las tecnologías usadas a lo largo del proyecto.

Durante el cuarto sprint me enfoqué en la implementación y desarrollo del código en Spring y la creación de unos datos de carga que posteriormente necesité cargar. Finalicé el desarrollo de la interfaz gráfica (con Bootstrap, HTML y CSS). Además, estuve haciendo modificaciones y cambios en la documentación previa. Estuve redactando y mejorando los apartados de Taiga para clarificar el propio proyecto.



En el quinto y último sprint me centré en algunas de las tareas finales como la preparación para la última entrega. Tuve que migrar todo el frontend a componentes que el Angular CLI pudiera interpretar además de añadirle la parte de los servicios que me permitían desde el frontend hacer cambios y que se vieran reflejados en la base de datos. Entre otras tareas tuve la creación de la redes sociales para posteriormente poder acceder a través de widgets y que el usuario pudiera tener un interfaz más visual y dinámica. Estuve haciendo los testeos de conexión con la base de datos. Finalmente estuve completando algunos apartados de documentación final del proyecto.

Uno de los aspectos importantes en los que invertí tiempo fue el testing desde Angular utilizando Karma. Realicé pruebas unitarias para asegurar el correcto funcionamiento de los componentes y servicios desarrollados. Además, aproveché este tiempo para pulir detalles y solucionar problemas relacionados con la base de datos, especialmente en lo que respecta a las relaciones entre las tablas para garantizar la integridad de los datos. Otro de los puntos fue la actualización de la documentación del proyecto. Registré todos los cambios y mejoras realizados, detallando los procesos y decisiones tomadas durante el desarrollo.



## Parte experimental

### Análisis:

El diseño del sitio web presenta una estructura bien definida. En la parte superior, se encuentra un encabezado con una imagen distintiva que incluye un botón de inicio de sesión que te abre un formulario que permite loguear y registrarse mandando ese correo directamente a la base de datos, el logo del sitio y una imagen, brindando una identidad visual clara. Justo debajo del encabezado, se encuentran los contadores de animales (diferenciado por tipo de animal) y un carrousel de imágenes que destacan a los animales sin hogar, generando conciencia sobre su situación.

En la siguiente sección, se muestran 4 tarjetas con información básica de los animales, como su nombre, sexo y fecha de nacimiento. Estas tarjetas proporcionan una introducción visual atractiva y despiertan el interés.

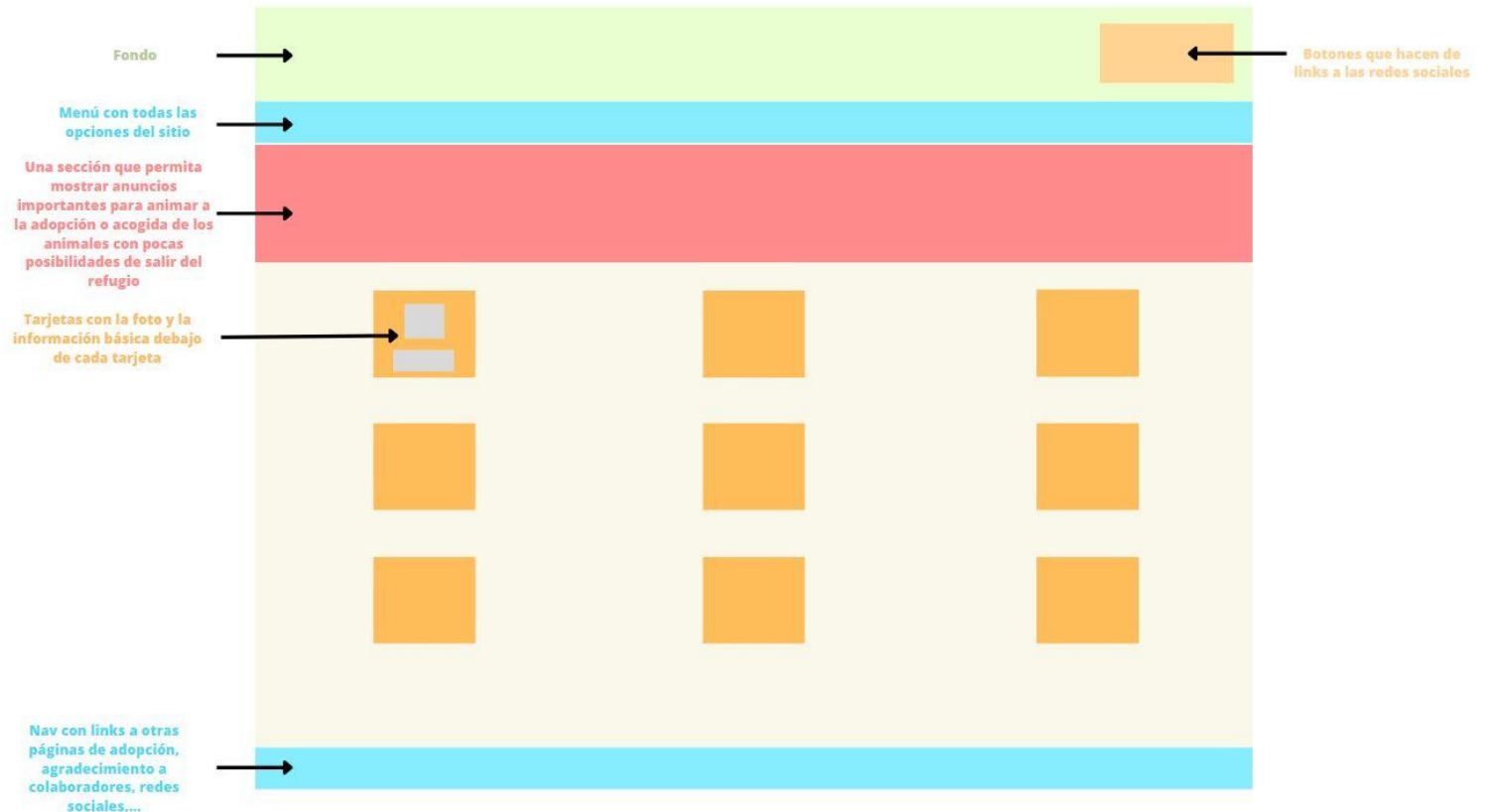
En la parte inferior del sitio, se incluyen widgets de redes sociales que permiten a los usuarios explorar y acceder a más fotos relacionadas con los animales, así como realizar donaciones para apoyar la causa.

El footer del sitio web es una sección relevante que contiene varios elementos importantes. En primer lugar, se muestra un mapa que indica la ubicación del centro, utilizando datos prestados de una protectora real para mayor autenticidad. También se encuentra un pequeño formulario funcional, que invita a los visitantes a suscribirse que envía un correo y registra el que te dan en una hoja de cálculo de google. Por último, se muestra un calendario de eventos que permite a los usuarios informarse de eventos relevantes, compartirlos por enlaces y correo electrónico para mantenerse al tanto de las actividades programadas. Para finalizar el diseño, se incluyen el logo del centro, el logo del proyecto.

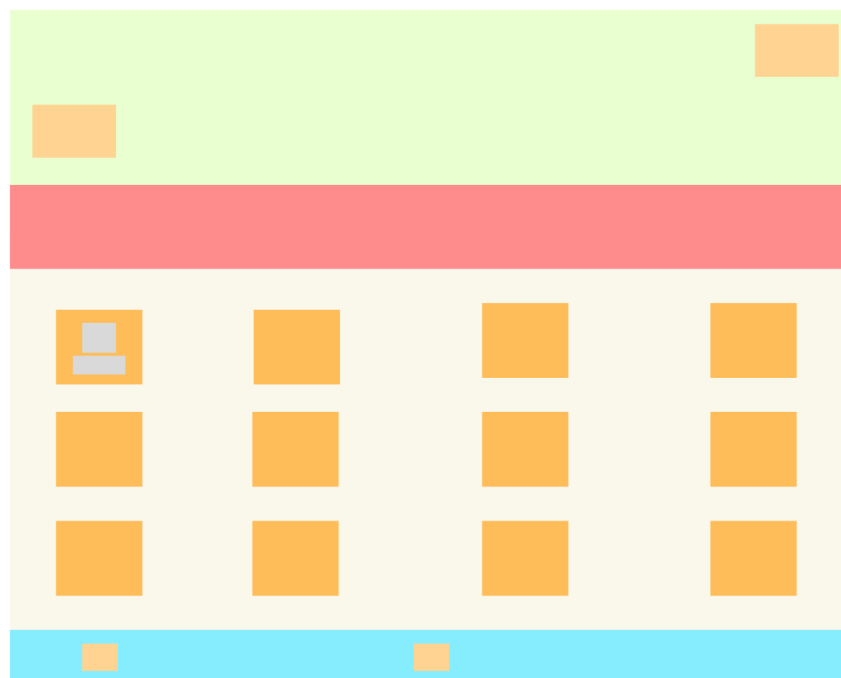




*(Este solo es un diseño aproximado que hice de como me quería que se viera la parte del frontend antes de empezar a escribir el código)*

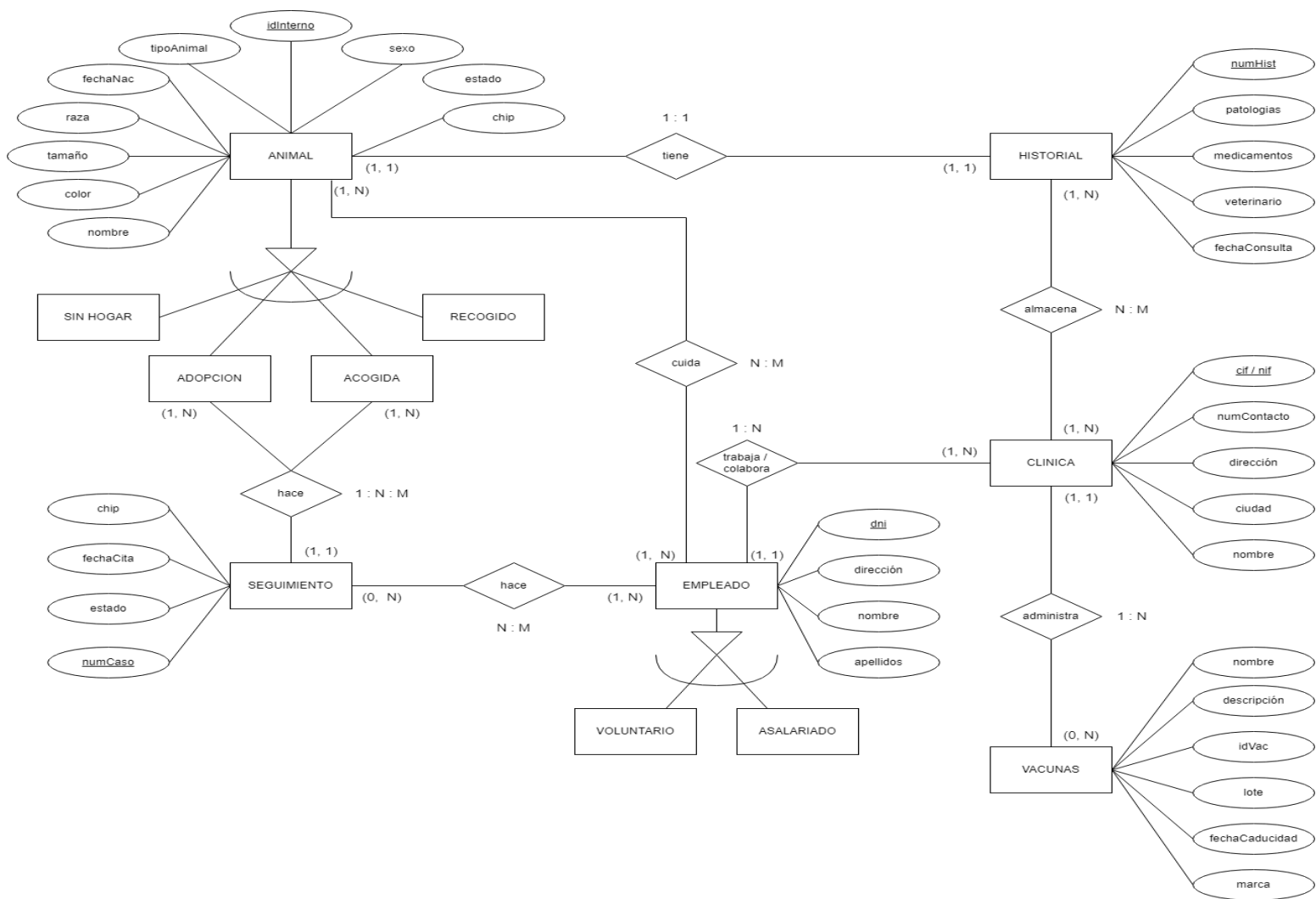


*(Este es un diseño de la vista actual)*





En el siguiente diagrama se incluyen las entidades se generan en la base de datos, así como sus relaciones y atributos correspondientes para planificar y estructurar la información.





## Diseño

Para clarificar la estructura antes de empezar a crear la base de datos decidí incluir el modelo relacional con la intención de solventar las relaciones, cuando se genera una nueva tabla o cuando los atributos se mueven de una tabla a otra.

En mi caso tengo solo una relación 1:1 en la que ambas se llevan su PK a la otra tabla como una FK.

También encuentro 2 relaciones 1:N en la que en la relación 1 se va a la tabla que tiene la N. En la primera el cif de la clínica se va a las vacunas y en la segunda se va el dni de empleado a la clínica.

Me salen 3 relaciones N:M (muchos a muchos) las cuales generan otra tabla cuyas. También encontré 2 relaciones ISA para poder diferenciar los diferentes tipos de empleados y las fases por las que pasa un animal antes de ser adoptado.

Este es el diseño más aproximado a como se vería la estructura de tablas en la base de datos.

```
1
2 ANIMAL: chip, sexo, idInterno (PK), tipoAnimal, fechaNac, raza, tamaño, color, nombre, numHistorial (FK)
3
4 HISTORIAL: numHistorial (PK), fecha, veterinario, medicamento, patologías, idInterno (FK)
5
6 CLINICA: cif (PK), numContacto, dirección, ciudad, fechaConsulta, dni (FK)
7
8 SEGUIMIENTO: chip, fechaCita, estado, numCaso (PK)
9
10 VACUNAS: nombre, descripción, idVac (PK), lote, fechaCaducidad, marca, cif (FK)
11
12 EMPLEADO: dni (PK), dirección, nombre, apellidos
13
14 EMPLEADO_HACE_SEGUIMIENTO (dni (PK, FK), numCaso (PK, FK))
15
16 HISTORIAL_DE_CLINICA (numHistorial (PK, FK), cif (PK, FK))
17
18 EMPLEADO_CUIDA_ANIMAL (dni (PK, FK), idInterno (PK, FK))
19
```



Con respecto a las tecnologías usadas he decidido Spring se que se encarga del backend y Angular se encarga del frontend, la separación de responsabilidades facilita el mantenimiento. Como base de datos decidí usar MySQL ya que Spring tiene una buena integración con bases de datos relacionales debido a su conjunto de herramientas y bibliotecas que facilitan la conexión y la consulta.

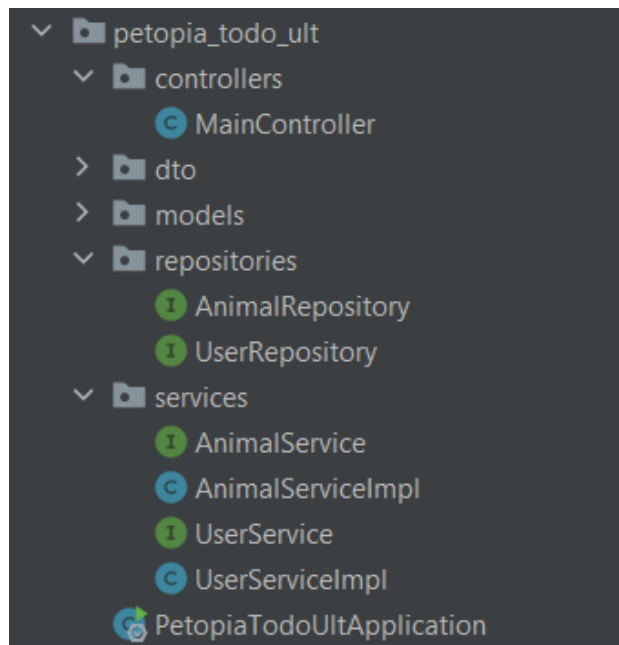
Todos los elementos se comunican a través de servicios. En el backend, se generan tablas basadas en el contenido de las clases, cada atributo es una columna cuyo tipo de dato es el indicado desde la clase. Los datos son cargados mediante los servicios y los repositorios, obteniendo la información desde InitialData. Estos cambios se reflejan en MySQL Workbench mediante consultas, al igual que todos los datos que se crean y cargan en la base de datos desde el frontend.



## Implementación y pruebas

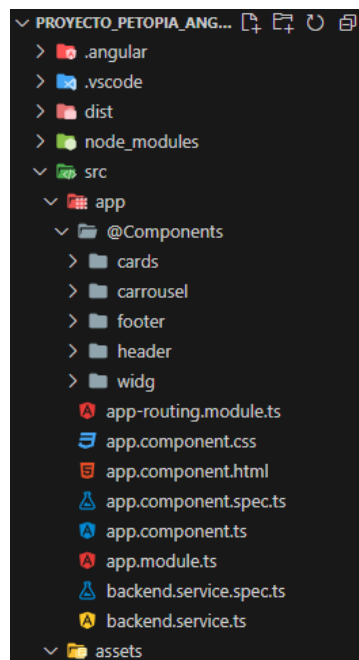
Quisiera comenzar explicando los lenguajes y aplicaciones utilizados durante el proyecto. En primer lugar, instalé MySQL para luego crear un usuario con una contraseña que se utiliza para enlazar con Spring.

Para el desarrollo del backend, utilicé IntelliJ. Tuve que agregar librerías (use las de yakarta) que me permitieran crear las clases y constructores entre otros, así como modificar las propiedades y crear todos los servicios y repositorio para poder acceder a la base de datos (incluyendo el usuario, la contraseña, el puerto,...) cargando todos los datos.





En cuanto al desarrollo del frontend, considero que fue la parte más compleja, ya que teníamos experiencia previa con Spring y lenguajes SQL. Para llevarlo a cabo, utilicé Visual Studio y Node.js. Para comenzar, tuve que habilitar la ejecución de scripts desde Windows Powershell y luego instalé extensiones de Angular en Visual Studio para crear el proyecto. A través de la terminal, pude crear los componentes y carpetas. Los servicios tuve que crearlos después para poder conectar el front con la base de datos y el front.



Como parte de las pruebas realizadas la primera comprobación que hice fue la conexión de spring con la base de datos, un test capaz de comprobar si se conectaba correctamente.

```
@SpringBootTest
public class PruebaConexionJdbc {
    1 usage
    @Autowired
    private DataSource dataSource;
    @Test
    public void pruebaConexion() throws SQLException {
        try (Connection connection = dataSource.getConnection()) {
            System.out.println("Se conecta correctamente");
        } catch (SQLException e) {
            System.out.println("La conexión falló");
            e.printStackTrace();
        }
    }
}
```



En Angular, al generar un componente o servicio, se crea automáticamente un archivo con extensión **.spec.ts** en la misma carpeta. Este archivo se utiliza para almacenar y ejecutar pruebas unitarias. Para realizar estas pruebas, se utiliza Karma, una herramienta que se instala por defecto junto con todas las demás dependencias y herramientas adicionales de Angular.

Cuando se ejecuta el comando **ng test**, se ejecutan todas las pruebas unitarias que se hayan creado en el proyecto y que estén almacenadas en los archivos con extensión **.spec.ts**. Este comando facilita la ejecución para posteriormente mostrarte todos los errores a través de la terminal o en el navegador con todos los detalles.

Desde angular tengo algunas pruebas unitarias:

- widg.component.spec.ts
- footer.component.spec.ts
- cards.component.spec.ts
- header.component.spec.ts (da error)
- app.component.spec.ts (da error)
- backend.component.spec.ts
- animal.service.spec.ts (da error)

Como posible implantación y mejora que me hubiera gustado hacer si tuviera más tiempo sería mejorar y arreglar las pruebas unitarias ya creadas para que funcionen correctamente y pasen sin fallo, ya que no son errores complicados y crear algunas pruebas más.



## Implantación y documentación

Primero debí instalar Node.js (18.16.0) y desde Windows Powershell habilitar la ejecución de scripts ejecutando como administrador y utilizando el comando ***Set-ExecutionPolicy RemoteSigned***. Esto permite la creación y ejecución desde el propio terminal desde Visual Studio.

Desde VisualStudio debí instalar Angular con ***npm install -g @angular/cli*** para posteriormente instalar las dependencias adicionales con ***npm install*** y algunos plugins útiles de Bootstrap y Angular para el proyecto:

- Angular 10 Snippets (para facilitar algunos atajos pero no es necesario)
- Angular Essentials (herramientas para el desarrollo como guías,...)
- Angular Schematics (ofrece funcionalidades y sugerencias útiles)
- Bootstrap 4 Snippets, Font Awesome 4 Snippets, Font Awesome 5 Free & Pro
- Material Icon Theme (no es necesario pero ofrece iconos para el explorador de archivos que ayudan a ver más claramente que es cada cosa)
- EditorConfig for VS Code (para aplicar reglas de estilo sobre el proyecto)

En cuanto a la configuración del entorno de desarrollo con Spring, como ya había estado utilizando Spring durante el curso y tenía mi entorno configurado, no fue necesario instalar nada adicional. Solo me aseguré de tener la versión correcta de Java y realicé algunos cambios en la configuración del proyecto. Es importante mencionar que se requiere Java 11 como SDK para trabajar con Spring.

Para MySQL solo tuve que descargar y posteriormente abrirlo, en la primera pantalla elegir Developer Default, en la siguiente pantalla comprobar que tiene las dependencias y las que no tiene instalarlas hasta que todo está correcto y ir a la siguiente pantalla donde se instala todos los componentes y seguir hasta la pantalla donde tienes que poner en usuario y la contraseña los cuales serán los del root y seguir hasta la pantalla donde te conectas con el servidor al cual accedes con las credenciales el root y continuar con todo lo predefinido hasta terminar la instalación.





Con estas configuraciones y herramientas instaladas, pude comenzar el desarrollo del proyecto.

1. Para poder ver los resultados del proyecto angular (sin nada más) se debe escribir desde el terminal: **ng serve**.
2. Desde el propio buscador acceder a: **localhost:4200**

---

Para MySQL se accede por el puerto 3306

```
spring.datasource.url=jdbc:mysql://localhost:3306/protectora_db
spring.datasource.username=root
spring.datasource.password=admin123
```

- 
1. Desde el propio IntelliJ tienes una opción de Run para el proyecto
  2. Para el backend se accede desde: **localhost:9000**



## Resultados y discusión

### Tiempos

En relación a los tiempos establecidos para el proyecto, tanto los estimados por el centro como los que me había marcado personalmente no se cumplieron. Durante el desarrollo, nos encontramos con diversos problemas con los que no habíamos tenido experiencia previa, lo cual me ralentizó bastante.

Por otro lado, la base de datos resultó ser más complicada de lo que esperaba. El diseño del modelo entidad-relación (E/R) no fue demasiado difícil y pude resolverlo en un par de horas. Sin embargo, el modelo relacional presentó mayores dificultades. Dedique casi un día entero para solucionar las relaciones entre tablas, incluyendo las relaciones ISA, las relaciones que generan tabla y una relación 1:N:M. En retrospectiva, considero que hubo cierta pérdida de tiempo al diseñar un archivo SQL que no terminé utilizando. A través de dicho archivo, se generaban todas las tablas y se cargaban datos de prueba, pero luego recordé que podía crear las tablas directamente desde las clases Spring con todos sus atributos y cargar los datos de forma más sencilla mediante el InitialData. Los datos de prueba que había preparado para cargarlos pude reutilizados para cargar los datos desde Spring.

La parte del backend con Spring no fue realmente mucho más complicada de lo que esperaba, las clases fueron relativamente sencillas de crear siempre clases API REST con la notación de SQL. Lo más complicado de todo esto fueron las relaciones N:M y algunos de los controladores. Tardé entre 3-4 días en tenerlo salvo pequeños detalles que debí modificar después.

En cuanto al desarrollo del frontend, decidí comenzar con Bootstrap, en aproximadamente 7-8 horas logré desarrollar una parte considerable del frontend. Sin embargo, cuando migré todo el código a Angular, solo la creación del proyecto, incluyendo la estructura de carpetas y componentes vacíos, me llevó entre 4 y 5 horas, desde habilitar la ejecución de scripts hasta la instalación de los plugins necesarios y la creación de las carpetas con todo lo necesario dentro.



## Dificultades

Una de ellas fue aprender a utilizar los comandos específicos de Angular y entender los cambios que se producen al incluir o excluir incluso una sola letra, lo que te puede generar o si te genera todo lo necesario para poder usarlo.

Además, comprender cómo se fragmenta el código en Angular no fue una tarea fácil de asimilar. Aunque ya teníamos cierta experiencia con fragmentos en Thymeleaf durante el curso, en Angular no solo se fragmenta el propio HTML, sino las llamadas a los componentes, cómo se carga su correspondiente archivo CSS (en el propio componente, el del proyecto y el del propio index),...

Uno de los desafíos más remarcados que encontré fue la creación de servicios. Una vez generados mediante el comando ng, escribir el código requería seguir notaciones específicas que modificaban la forma en que se utilizaban.

En cuanto a la base de datos, enfrenté problemas con las tablas que generaban una relación de muchos a muchos (N:M). Estas tablas no se generaban correctamente y establecer las relaciones adecuadas en Spring fue tarea fácil. Fue necesario dedicar tiempo adicional para solucionar estos problemas.

Hacer las pruebas unitarias de Angular tampoco fue tarea fácil, algunos errores eran fáciles de encontrar y arreglar pero otros eran más complejos y aun sabiendo lo que significaba no sabía muy como arreglarlo correctamente.



## Mejoras e implementaciones

Dentro de las posibles mejoras del proyecto, se plantea la integración de funciones de pago o suscripciones reales. Sin embargo, esta tarea resulta complicada debido a la necesidad de utilizar una cuenta real que permita operar con dichas funcionalidades.

Personalmente, me habría gustado tener más tiempo para poder integrar un componente que permita la incorporación de un pequeño vlog. Muchas páginas web de protectoras cuentan con un apartado donde los usuarios pueden realizar consultas a la protectora o a otros usuarios sobre temas diversos, como alimentación, tratamientos y cuidados.

Además, hubiera sido fantástico incluir un apartado relacionado con los trámites de adopción en la página con información detallada sobre los diferentes procesos y cómo iniciarlos. Esto permitiría a los usuarios tener un mayor conocimiento sobre el proceso de adopción.

Se hubiera podido incluir una página extra sobre voluntariado en la que se explique como se puede ayudar en las diferentes tareas y con un formulario de solicitud para proponerse voluntario.

Un apartado que suelen tener otras webs es el de recursos educativos para que la gente sepa cómo cuidar correctamente un animal y donde se conciencie sobre la adopción responsable, esterilización,...

Creo que hubiera sido interesante el poder realizar más parte de testing, me hubiera gustado poder profundizar en las pruebas unitarias con Jasmine.

Por último, me habría encantado poder implementar una pequeña galería de fotos para cada animal con información más detallada. Esta mejora proporcionaría a los usuarios información adicional sobre el animal.



## Conclusiones

Creo que este proyecto me ha brindado una experiencia práctica que refleja los plazos que se encuentran en un entorno laboral real. Tuve que cumplir con los requerimientos establecidos dentro de un marco de tiempo específico dependiendo principalmente de mi propia investigación y resolución de problemas.

El desarrollo de este proyecto me ha permitido expandir mis conocimientos en el uso de Spring, aprender y entender cómo funciona Angular y su distribución de los proyectos y aprender cómo se integran ambas tecnologías.

Además, he adquirido una comprensión más profunda de cómo se organizan los proyectos conjuntamente para crear aplicaciones web completas y funcionales.

Además de ampliar mi dominio de las tecnologías, con este proyecto también he aprendido la importancia de la organización y la planificación, he aprendido a priorizar tareas, administrar el tiempo y adaptarme a los posibles problemas que puedan surgir durante el proceso.

Creo que el proyecto también me ha permitido profundizar en un tema que realmente es importante y que no se suele tratar, comprender de una forma mucho más cercana las necesidades de las protectoras de animales y encontrar una nueva forma de ayudar y colaborar.



## Bibliografía y referencias

Protectoras que han colaborado en investigación

<https://albaonline.org/>

<https://www.anaaweb.org/>

<https://www.huellaranda.org/>

Estudios y documentación sobre datos

<https://www.fundacion-affinity.org/>

Diseños:

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

Herramientas:

<https://angular.io/>

<https://getbootstrap.com/>

<https://www.w3schools.com/>

<https://codepen.io/trending>

<https://openai.com/blog/chatgpt>

<https://taiga.io/es>

<https://fonts.google.com>



## Anexos

### Investigación

Como parte de mi labor de investigación, tuve la oportunidad de hablar con varias protectoras de animales para obtener sus recomendaciones sobre cómo mejorar el proyecto. Me di cuenta de que sus sugerencias podrían ser útiles para hacer que el proyecto sea más realista y adaptable, y cómo se podrían cubrir las diferentes necesidades que pueden tener según su tamaño.

- Por ejemplo, me comentaron que las protectoras pequeñas suelen trabajar con voluntarios y no con empleados, en este caso veo necesario hacer una diferencia ya que las que son más grandes si necesitan empleados para poder hacer el mantenimiento de las instalaciones, etc.
- Otra sugerencia que recibí fue ampliar la base de datos para incluir información sobre los tratamientos que los veterinarios dan a los animales, no sólo en lo referente a vacunas, sino también a desparasitación, curas, etc. Asimismo, me pareció importante incluir el número de colegiado del veterinario para poder identificarlos correctamente.
- Debido a que los chips de identificación de animales no siempre son únicos (dependiendo de la comunidad autónoma pueden varias), opté por incluir un campo de IdInterno en la base de datos. Esto permitiría diferenciar a los animales con certeza en caso de ser necesario, especialmente si se trabaja a nivel nacional.
- Por otro lado, algunos voluntarios sugirieron que se incluyeran los materiales disponibles (comida, vendas, etc.) en la base de datos, así como la cantidad de comida necesaria para el animal o su dieta. Aunque creo que esta información es importante, decidí dejarla como una opción para incluirla solo si daba tiempo.

En resumen, la estructura provisional del proyecto se diseñó teniendo en cuenta la importancia de relacionar el trabajo de las protectoras con las clínicas, los voluntarios y los empleados, así como la necesidad de diferenciar entre los diferentes estados y fases por los que puede pasar un animal.