
APLICACIÓN WEB PARA ADOPTAR ANIMALES

APIPET

Saúl Aznárez Rodríguez

Grado superior en desarrollo de aplicaciones web

Trabajo fin de grado

Tutor de TFG: Antonio López Dios

Tutor de prácticas: Xavier Venancio Rosillo Guerrero

Convocatoria Junio 2022

FICHA TÉCNICA DEL TRABAJO FINAL DE CURSO

Título del trabajo: Apipet

Nombre del autor: Saúl Aznárez Rodríguez

Nombre del consultor: Antonio López Dios

Fecha de entrega: 03 de Junio del 2022

Titulación: Desarrollo de aplicaciones web

Idioma del proyecto: Castellano

Resumen

Un usuario cuando haya realizado el login si no tiene intereses guardados se le preguntara sobre sus intereses por el animal que desea adoptar (tipo de animal y ubicación), una vez lo haya elegido se le redirige a su “home” donde se le realiza una búsqueda automática con estos intereses.

El usuario que tiene animales en adopción tendrá una opción de ver sus animales. Esta vista solo podrá ser accesible por el dueño de los animales.

Una vez que se desee adoptar un animal se pulsara el botón de ¡adóptame! y automáticamente se envía un mensaje vía email al usuario que da en adopción para que se pongan en contacto, ya que la confirmación de la adopción la

realiza el que da en adopción. Una vez realizada dicha confirmación el animal desaparecerá de la lista de animales en adopción.

ABSTRACT

When a user has log in and has no saved interests, he will be asked about his interests in the animal he wishes to adopt (type of animal and location), once he has chosen it, he is redirected to his "home" where an automatic search with these interests.

The user who has animals for adoption will have an option to see their animals. This view can only be accessible by the owner of the animals.

Once you want to adopt an animal, you will press the button adopt me! and a message is automatically sent via email to the user who gives up for adoption so that they can get in touch, since the confirmation of the adoption is made by the person who gives up for adoption. Once said confirmation has been made, the animal will disappear from the list of animals for adoption.

ÍNDICE

I INTRODUCCIÓN	6
1.1 Contexto	6
1.2 Motivación y Objetivos	7
1.2.1 cliente final	8
1.2.2 Necesidades y requerimientos del cliente	9
1.2.3 imagen corporativa	9
II ANALISIS DE MERCADO	11
2.1. Competencia	11
2.2 Marketing	12
2.3 Fuentes de ingresos y viabilidad económica	12
2.4 Viabilidad técnica	13
2.5 Estrategia de evolución	14
III ARQUITECTURA Y TECNOLOGÍAS UTILIZADAS	15
3.1 Tecnologías - cliente (Frontend)	17
3.2 Tecnologías - servidor (backend).....	17
IV PLANIFICACION DEL PROYECTO	24
4.1 Requisitos no funcionales	24
4.2 Requisitos funcionales	24
4.3 Metodología ágil	25
4.3.1 Manifiesto ágil	25
4.3.2 Scrum	27
4.4 Casos de uso y estructura	35
4.5 Base de datos	48

V CONCLUSIONES Y POSIBLES AMPLIACIONES	49
5.1 CONCLUSIONES	49
5.2 TRABAJO FUTURO	49
VI BIBLIOGRAFÍA	51

I INTRODUCCIÓN

1.1 CONTEXTO

Es preocupante que uno de los titulares de un periódico sea: "España, líder europea en abandono de animales: 700 cada día."

España se sitúa desde hace años a la cabeza de Europa en cifras de abandono animal, la falta de concienciación y la dejadez política ponen en jaque un sistema de protectoras y perreras completamente saturadas.

Las comunidades autónomas donde se recogen más gatos y perros extraviados son **Andalucía, Madrid y Cataluña**. Esto significa por un lado algo positivo: que son comunidades donde se hace un gran esfuerzo por recoger a estos animales y llevarlos a refugios, sin embargo, inevitablemente también significa que el índice de abandono es alto.

Existen diferentes condicionantes claves para que los animales sean abandonados, el más importante es que todavía las mascotas son entendidas por muchas familias como **un bien de consumo**, y regalan animales a personas que no se sienten cómodas recibiendo esta responsabilidad, lo que puede desembocar en abandonos pese a que las campañas de sensibilización llevan varias décadas tratando de evitarlo, pero también existen otros motivos de abandono como:

Motivos (%)
Factores económicos 18,8
Camadas indeseadas 13,9
Problemas de comportamiento animal 11,5
Fin de la temporada de caza 10,1
Pérdida de interés por el animal 8,6
Falta de tiempo o espacio 7,6
Cambio de domicilio (piso más pequeño. De alquiler) o traslado 5,9
Pérdida de empleo 5,9
Vacaciones 3,6

Figura 1. Factores por abandono

En general, los meses entre mayo y julio es la época del año en que más animales abandonados o extraviados se recogen, en proporción al total que entran en los refugios.

Las cifras son muy preocupantes porque, aunque en 2021 se produjeron 627 detenciones por maltrato o abandono de animales domésticos en España, se sigue abandonando, a todo esto, se le suma que cada año se producen detenciones por tráfico ilegal de especies protegidas o en peligro de extinción.

Perros y gatos recogidos por refugios y protectoras en España

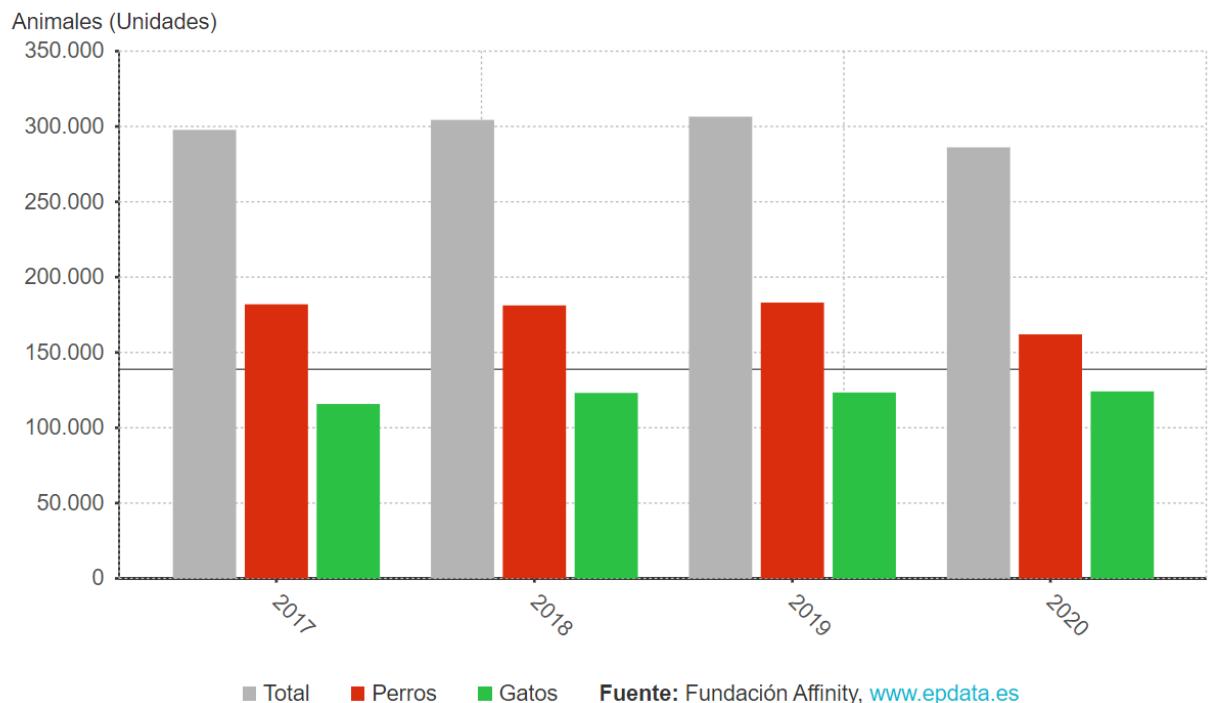


Figura 2. Estadísticas de acogidas.

1.2 MOTIVACIÓN Y OBJETIVOS

Dado todos los datos anteriores, se entiende que la motivación principal para la realización de este proyecto es esencialmente poder reducir a lo máximo posible el índice de abandono de

todo tipo de animales, no solamente de perros y gatos que son los más recurrentes en el abandono, si no recoger a cualquier animal y poderles dar una familia para que tenga una vida mejor.

Los datos son alarmantes, la ciudadanía no está concienciada, hacerse cargo de un animal significa haber sopesado la gran responsabilidad que tienes sobre él, antes de adoptar o “comprar” hay que pensar si te puedes hacer cargo al 100%.

El objetivo fundamental es dar a conocer a los animales que por cualquier motivo no pueden continuar con su actual dueño o han sido abandonados dándoles la oportunidad para que otras familias les puedan acoger y esto reduciría el abandono.

Esto conlleva que se reduzca el número de compras de animales dado que se estaría fomentando un negocio donde los animales son mera mercancía de trabajo y de hacer dinero.

Apipet es una pagina donde NO se puede vender animales, es restrictivamente para la adopción.

1.2.1 CLIENTE FINAL

El cliente final de esta aplicación serían todas aquellas personas interesadas en adoptar animales o dar en adopción algún animal en posesión de habla española ya que solamente está traducida a un solo idioma.

La plataforma web se centra en un público de habla hispana (peninsular) concienciado con la adopción de animales y que dispone de acceso a internet; y son tantos los usuarios de la plataforma (particulares) como las protectoras de animales de ámbito nacional.



Figura 5 - Actores de la plataforma

1.2.2 NECESIDADES Y REQUERIMIENTOS DEL CLIENTE

Apipet ofrece a todos los usuarios un servicio muy concreto conectar con cualquier persona ya sea bien para exponer un animal o aquellos interesados en un animal.

La aplicación ofrece una navegación sencilla dinámica y moderna para que el usuario tenga la mejor experiencia virtual con un entorno visual agradable y facilidad para interactuar con otros usuarios vía email o vía mensaje.

Los usuarios podrán compartir publicaciones o guardarse en favoritos a los animales, poder filtrar según raza, tipo y provincia para una mejor búsqueda y aparte de todo esto tendrán la posibilidad de guardarse unos intereses para que nada más entrar a la aplicación se filtre por dichos intereses es decir si a un usuario le gustan los perros le saldrán las publicaciones filtradas por perros.

1.2.3 IMAGEN CORPORATIVA

El nombre de la empresa se ha conformado por dos partes “Api” de aplicación y “pet” de mascota en inglés, ya que está dirigida a ellos.

El eslogan elegido es: “Una mascota quiere conocerte” incitando a las personas a la acogida de mascotas en vez de comprarlas. Además, es un slogan sentimental dando valor a una vida desprotegida.

Los colores que se han usado han sido azul y gris oscuro, el azul se ha elegido porque es un color que transmite seguridad y protección.

La imagen corporativa se conforma de cuatro rombos cada 1 de ellos tiene una imagen diferente con dibujos que dan pistas al usuario de qué es lo que se va a encontrar dentro de esta aplicación. Estos cuatro elementos forman un rombo para este integrado el nombre de la empresa junto con la marca de la empresa.



Apipet

UNA MASCOTA QUIERE CONOCERTE

II ANÁLISIS DE MERCADO

La aplicación va a ser una actividad sin ánimo de lucro es decir no va a tener ningún coste, si un usuario quiere publicar sus anuncios o simplemente buscar podrá hacerlo totalmente gratuita, aunque se buscará patrocinadores como protectoras, marcas que estén especializadas en animales o cualquier entorno que pertenezca a este sector.

2.1 COMPETENCIA

Para analizar la competencia se buscó en internet plataformas web que hiciesen algo parecido. Se analizaron sus redes sociales y se utilizó SimilarWeb para generar informes sobre cada una de sus webs.

a) AdoptaMe

Esta página web se consideró como competencia directa. Es una aplicación española que, a pesar de llevar tan solo unas semanas en Play Store, acumula ya más de 10.000 descargas y se está compartiendo bastante por las redes sociales. Se trata de una iniciativa para dar visibilidad a animales en adopción.

b) Miwuki Pet Shelter

Esta app posee una base de datos con más de 90.000 mascotas, repartidas por todo el territorio nacional. Muestra toda la información del animal elegido, como el sexo, la raza, si está vacunado o no, etc. Un dato curioso es que no se puede buscar por raza, la app asegura que en caso contrario muchos animales quedarían excluidos en las búsquedas.

c) Amazdog

El Amazon de las mascotas, una de las aplicaciones para adoptar donde sí o sí encontraremos a nuestra media naranja peluda. Una app muy visual, sencilla e intuitiva donde encontremos mucho más que adopciones, también otros servicios como hoteles para perros, clínicas veterinarias cercanas a nuestra ubicación, playas para perros, etc.

El inconveniente que tiene es que solo se pueden adoptar perros.

2.2 MARKETING

Para el marketing del proyecto se utiliza las siguientes herramientas y estrategias:

- Google Ads como posicionamiento SEM: plataforma de anuncios y principal fuente de ingresos de Google. Con Google Ads las empresas tienen la posibilidad de pagarle a Google para que las destaque en internet. Es invertir en audiencia calificada y por lo tanto, que esté relacionada con tu negocio.
- Google Analytics: es una herramienta gratuita para analítica web. Permite acceder a una gran cantidad de información y realizar un análisis profundo del mismo y así poder realizar mejores estrategias de marketing digital, optimizar el sitio web para mejorarlo y encontrar oportunidades o tendencias.
- Posicionamiento SEO: se trata del posicionamiento natural de las aplicaciones en los principales canales de descarga. De esta manera, se consigue que la app aparezca entre las primeras búsquedas y consiga más descargas. Se utilizarán herramientas SEO para ayudar en mejorar el posicionamiento y optimizar el contenido y con ello lograr los objetivos.

2.3 FUENTE DE INGRESOS Y VIABILIDAD ECONÓMICA

Las principales fuentes de ingresos será:

- Mediante publicidad dentro de la plataforma web. Para poder costear el mantenimiento. Ya que para los usuarios el acceso y uso será gratuito.

2.4 VIABILIDAD TÉCNICA

La aplicación está planificada como un sistema MVC (Modelo – Vista – Controlador). Se desarrollará con Angular en la parte Frontend, y Symfony para el Backend con Xampp que incluye Apache, MySQL/Mariadb para la base de datos y PHP.

Se utiliza un control de versiones gratuito (GitHub) manteniendo el proyecto de forma privada. En cuanto al despliegue, se utilizará la plataforma Heroku para la aplicación.

Heroku es una de las plataformas como servicio(PaaS) más utilizados en la actualidad por su fuerte enfoque en resolver el despliegue de una aplicación además tiene un modelo de servicio gratuito para proyectos pequeños. Características de Heroku

- Soporta diferentes lenguajes de programación: Node, Ruby, Java, Clojure, Scala, Go, Python, PHP
- Tiene una versión gratuita fácil de usar
- Ejecuta las aplicaciones a través de sus contenedores, también conocidos como Dynos
- Tiene Dynos que pueden ser de tres tipos: web, worker o cron
- Ofrece más de 200 complementos con los que ampliar las aplicaciones al instante
- Ofrece varias características de seguridad, incluyendo SSL, autenticación y cumplimiento de PCI

2.5 ESTRATEGIA DE EVOLUCIÓN

El proyecto se desarrollará en 3 fases:

- Fase 1 – Documentación y la funcionalidad.
- Fase 2 – Captación de clientes y de protectoras pequeñas (Marketing).
- Fase 3 – Incremento de funcionalidad y expansión nacional.

Fase 1:

- Generar un Plan de Empresa y una especificación de requisitos básica. Detallar las funcionalidades básicas y futuras, la Idea de negocio, realizar un estudio de mercado y buscar, pensar y analizar las tecnologías que se van a utilizar.
- Realizar una investigación exhaustiva de las tecnologías para aprender a usarlas y ver si son viables antes de empezar con el desarrollo.
- Realizar una planificación sobre el desarrollo del proyecto utilizando metodologías ágiles, usaremos Scrum ya que permite hacer una división muy específica del proyecto.
- Desarrollar la plataforma para obtener una funcionalidad básica.

Fase 2:

- Realizar pruebas de la aplicación con clientes con el objetivo de que la utilicen y puedan dar las primeras impresiones para posteriormente llevarla ante protectoras de animales.
- Seguir trabajando para optimizar las estrategias de marketing.

III ARQUITECTURA Y TECNOLOGÍAS UTILIZADAS

3.1 TECNOLOGÍAS - CLIENTE (FRONTEND)

a) Javascript

Sigue siendo uno de los más usados en los últimos años. Está ganando impulso continuamente tanto en aplicaciones del lado del servidor como del cliente.

Los frameworks como Angular y las librerías están orientados a acelerar el proceso de desarrollo aprovechando el poder de herramientas creadas por desarrolladores experimentados y dan solución a muchos problemas comunes.



Figura 4.

b) Angular

Angular es un marco de interfaz de usuario de código abierto que fue diseñado para aplicaciones web modernas y potentes. Continúa afirmando su dominio como el nombre más consolidado en el mundo de los frameworks de JavaScript front-end.

La razón principal detrás de su popularidad es que es fácil y simple de usar, también simplifica el proceso de desarrollo y la estructura del código JavaScript.

Es muy demandado principalmente por su funcionalidad de enlace de datos bidireccional. Esto significa que cuando el marco se activa por un evento en el navegador web, puede hacer que el módulo y las acciones del usuario en la página web se actualicen y cambien los patrones necesarios y esenciales. Utilizando el enlace de datos bidireccional, la aplicación simplificará su capa de presentación. Esto permite un enfoque más simplista y menos intrusivo de la visualización del DOM para crear la interfaz de usuario.

Por lo tanto Angular sigue estando entre los tres frameworks mas usados en 2022.

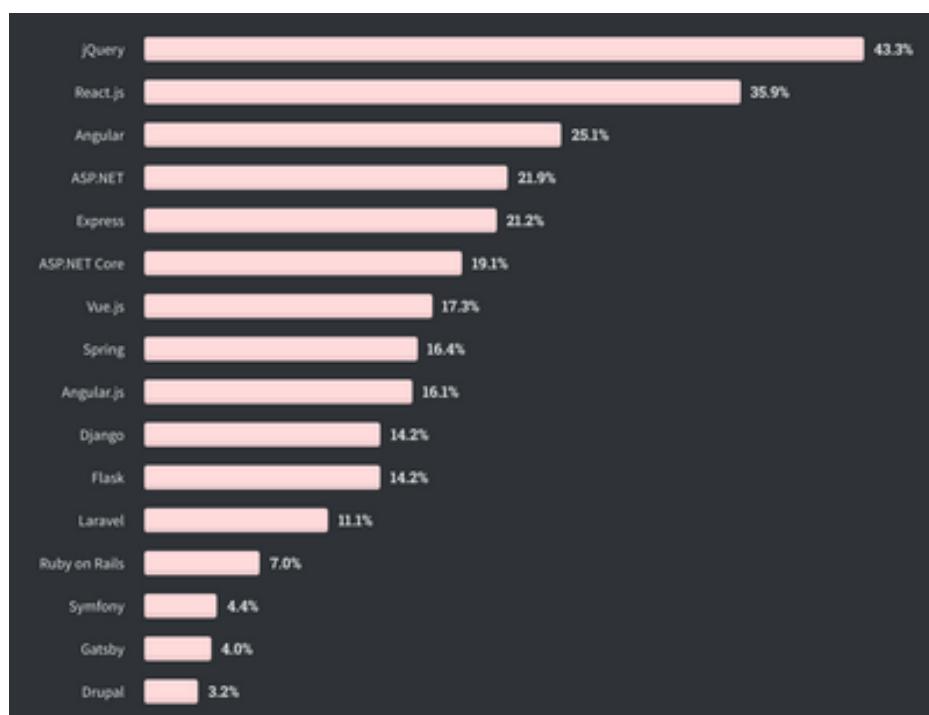


Figura 5.

c) Bootstrap

Se trata de un *framework* de CSS que podemos incluir en nuestro sitio web y comenzar a maquetarlo en poco tiempo y de manera sencilla.

Además nos provee de todas las reglas CSS para que nuestro sitio web se adapte de manera dinámica a la mayoría de las pantallas. Es muy fácil e intuitivo. Es extensible y cuenta con múltiples implementaciones externas.

Cuenta con múltiples implementaciones externas: WordPress, Drupal, SASS o jQuery UI.

Hace uso de componentes y servicios que han sido creados por la comunidad web. Como, por ejemplo: HTML5 shim, Normalize.css, OOCSS, jQuery UI, LESS o GitHub.

d) HTML5 Y CSS

HTML sus siglas en inglés significan «lenguaje de marcas de hipertexto» (HyperText Markup Language), y hacen referencia al código que define el significado de las instrucciones dadas a una plataforma computacional. Sirve básicamente para definir la estructura y el contenido de una página Web.

CSS es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de las páginas.

Puedes crear reglas para decirle a tu sitio web cómo quieras mostrar la información y puedes guardar los comandos para elementos de estilo (como fuentes, colores, tamaños, etc.)

3.2 TECNOLOGÍAS - SERVIDOR (BACKEND)

a) PHP

PHP es el lenguaje de programación del lado del servidor más utilizado para crear sitios web dinámicos, aunque puede tener otras aplicaciones. De hecho, el 79,2% de todos los sitios web

dependen de PHP en algún grado, lo que lo convierte en uno de los lenguajes más populares entre los programadores y desarrolladores web debido a su uso generalizado.

Se utiliza usualmente del lado del servidor para crear sitios web dinámicos, aunque puede tener otras aplicaciones.

En general, el servidor web (habitualmente Apache) ejecuta el código y lo transforma en un archivo HTML, XHTML o CSS y en datos, como imágenes jpeg o gif. Todo esto es interpretado por el navegador para generar la página web.

PHP se caracteriza por ser un lenguaje de programación flexible y pragmático. Esto lo convierte en muy amigable para principiantes. Pero también puede ser la solución perfecta para problemas complejos.



Figura 6.

b) Symfony

Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Detrás de Symfony existe una gran comunidad trabajando, que colabora y fomenta su utilización. Esta comunidad ha crecido de forma exponencial durante los últimos años y la documentación ha ido aumentando y mejorando.

Además es fácil de instalar y configurar en la mayoría de plataformas.

Independiente del sistema gestor de bases de datos.

Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP.

Usa variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

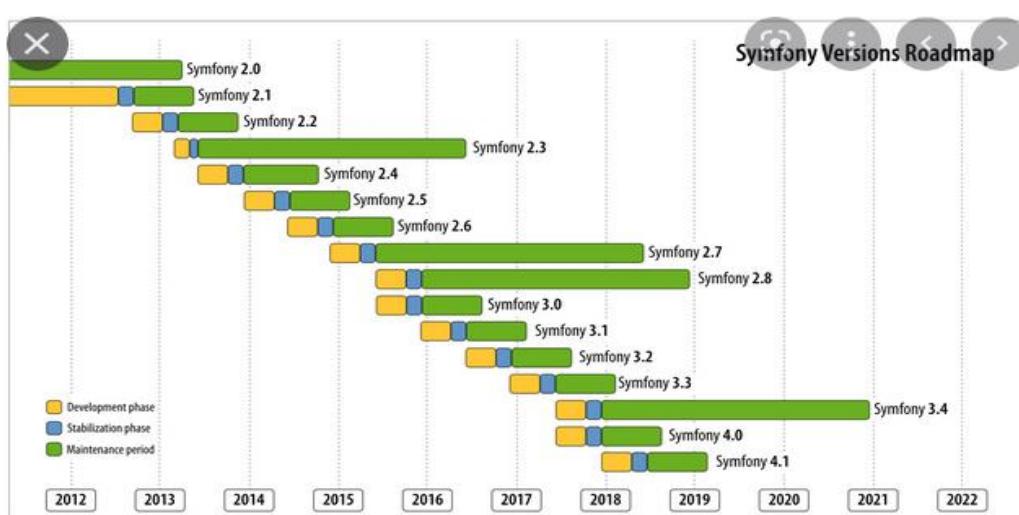


Figura 7

c) XAMPP

Es una herramienta de desarrollo que te permite probar tu desarrollo web basado en PHP en tu propio ordenador sin necesidad de tener acceso a internet. Te provee de una configuración totalmente funcional desde el momento que lo instalas.

Básicamente es una distribución de Apache que incluye diferentes softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen:

1. Linux: Es el sistema operativo donde estará instalado nuestra aplicación. A diferencia de Windows, Linux es una distribución libre que es segura, no requiere pago de licencias y tiene alto rendimiento.
2. Apache: el servidor web de código abierto es la aplicación usado globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como software libre por la Apache Software Foundation.
3. MySQL/MariaDB: XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web. En las versiones actuales de XAMPP esta base de datos se ha sustituido por MariaDB.
4. PHP: es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente de plataforma y soporta varios sistemas de bases de datos.
5. Perl: este lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red. También permite programar aplicaciones web dinámicas.

Es una herramienta muy práctica que además de esta distribución gratuita también incluye otras herramientas como el servidor de correo Mercury, el programa de administración de bases de datos phpMyAdmin, el software de analítica web Webalizer, OpenSSL, Apache Tomcat y los servidores FTP FileZilla o ProFTPD.

d) GiT Y GITHUB

GitHub es una página web en donde podrás alojar el código de tus proyectos utilizando el sistema de control de versiones Git.

Git es una herramienta de código abierto que los desarrolladores instalan localmente para gestionar el código fuente

Git y GitHub son dos entidades diferentes que te ayudan a administrar y alojar archivos. En otras palabras, Git sirve para controlar las versiones de los archivos mientras que GitHub es una plataforma para alojar tus repositorios Git.

Una de las más grandes ventajas de Git es su capacidad de ramificación, ya que son fáciles y económicas de fusionar, lo que facilita el flujo de trabajo.

Si unimos todo: las ramas de funciones, el desarrollo distribuido, las solicitudes de incorporación de cambios y una comunidad estable. Todas estas razones o características facilitan un flujo de trabajo ágil y veloz en el que se anima a los desarrolladores a compartir cambios más pequeños con mayor frecuencia.

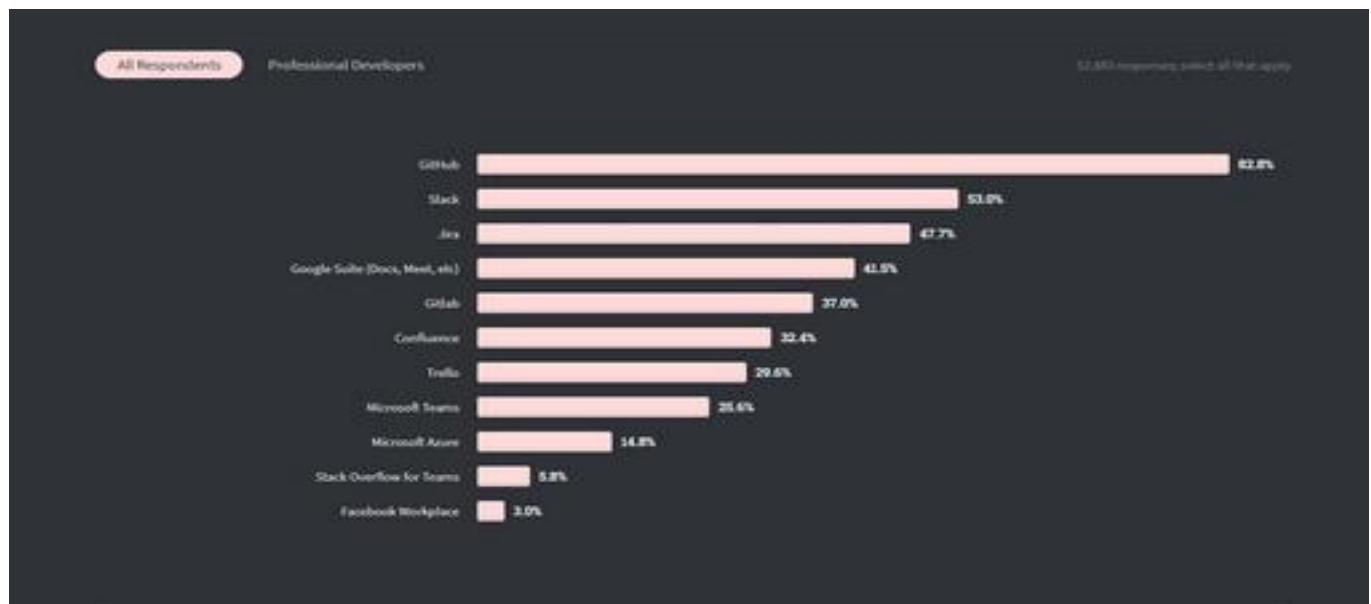


Figura 8.

e) Postman

Postman nace como una herramienta que principalmente nos permite crear peticiones sobre APIs de una forma muy sencilla y poder, de esta manera, probar las APIs. Todo basado en una extensión de *Google Chrome*. El usuario de Postman puede ser un *desarrollador* que esté comprobando el funcionamiento de una API para desarrollar sobre ella o un *operador* el cual esté realizando tareas de monitorización sobre un API.

Alrededor de la idea de testear las APIs, Postman nos ofrece un conjunto de utilidades adicionales para poder gestionar las APIs de una forma más sencilla. Es por ello que nos va a proporcionar herramientas para documentar los APIs, realizar una monitorización sobre las APIs, crear equipos sobre un API para que trabajen de forma colaborativa... convirtiendo a Postman plataforma de desarrollo de APIs que se basa por un modelo de desarrollo API First.

A día de hoy Postman nos ofrece aplicaciones *Windows*, *Linux* y *Mac*, así como un módulo colaborativo para equipos basado en Cloud.

En la aplicación se utiliza Postman para realizar y comprobar que las peticiones vía API funcionan antes de generarlas en el frontend.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Home, Workspaces, API Network, Reports, and Explore. Under 'My Workspace', there's a 'Collections' section with 'ApiPet' expanded, showing several API endpoints: POST http://127.0.0.1:8000/registro/..., POST http://127.0.0.1:8000/animal/a..., POST Crear animal, GET Get Users, POST Login Autorizacion, POST Prueba validacion token, POST crear usuario, POST editar usuario, POST editar animal, POST lista animales de usuario, POST mostrar un animal de user, POST Buscador de animales, POST eliminar animal usuario, and POST get animal. The main workspace area shows a 'POST Login Autorizacion' request. The 'Body' tab is selected, showing a JSON payload: {"email": "mario@prueba.com", "password": "1"}. The response status is 200 OK, with a status message: "Status: 200 OK Time: 122 ms Size: 1.08 KB". The response body is a large JSON string starting with "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9, eyJzdW1I0jcsImVtYWIslJoibmFyaW9acHJ1ZWJhLmNvbSisImShbWU101JQYXJ1ZGVzIiwiaWF0IjoxNjU0MjQ3M...".

Figura 9.

f) API Rest

La parte de servidor se constituye como un servicio API de transferencia de estado y datos representacional (en inglés representational state transfer) o REST¹³ que sigue una serie de reglas que se deben cumplir para crear un estilo de arquitectura software, la cual se apoya en el protocolo HTTP. Las operaciones más importantes que maneja son “GET” para consultar y leer, “POST” para crear y editar, y “DELETE” para eliminar.

g) Node package manager (NPM)

Este sistema facilita a los desarrolladores compartir y reutilizar código generado como paquetes o a veces también denominados módulos. Al crear un proyecto Node con NPM, crea por defecto un archivo llamado “package.json” donde viene toda la configuración previa necesaria para que un proyecto Node pueda ejecutarse.

h) Node.js

Es un entorno de tiempo de ejecución de JavaScript. Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript.

Utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente (con entrada nos referimos a solicitudes y con salida a respuestas). Puede referirse a cualquier operación, desde leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP.

IV PLANIFICACIÓN DEL PROYECTO

4.1. REQUISITOS NO FUNCIONALES

Para la correcta implementación del proyecto se han establecido los siguientes requisitos no funcionales:

- La plataforma tiene su interfaz adaptada a dispositivos escritorio(ordenadores). Para la futura actualización se adaptara para ser multiplataforma. (móvil, tablets,etc).
- El código del proyecto se almacena en un control de versiones gratuito manteniendo el código de forma privada.
- Los datos de los usuarios y animales se almacenan en una base de datos accesible solo por la plataforma. En la futura actualización se realizará el despliegue y estos datos serán almacenados en una base de datos en la nube, desplegada por Heroku.
- Para que un usuario pueda acceder al panel de administración es necesario su registro.

4.2 REQUISITOS FUNCIONALES

A continuación, se describen las distintas funcionalidades que debe ofrecer la aplicación:

- API REST:
 - Usuario:
 - Verificar cuenta usuario.

- Login usuario.
- Subir imagen de perfil de usuario a MySQL Database.
- Ver todos los animales del usuario que ha hecho login.

○ Animales:

- Registrar animales para darlos en adopción.
- Subir datos del perfil del animal a MySQL Database.
- Subir galería de imágenes del animal a MySQL Database.
- Buscar animales para adoptar.
- Ver individualmente los datos e imágenes de un animal para adoptar
- Editar un animal.

○ Configuración y utilidades:

- Buscador de animales.
- Envío de email a usuario dueño del animal.
- Elegir unos intereses de preferencias para adoptar.

4.3 METODOLOGÍA ÁGIL

4.3.1 MANIFIESTO ÁGIL

En Estados Unidos se reunieron 17 reconocidos expertos de la ingeniería del Software, los creadores de Scrum, XP, Crystal, DSDM, Extreme Programming, Adaptive Sotfware Development, Feature-Driven Development...

El objetivo de la reunión fue debatir y buscar alternativas a los procesos tradicionales de desarrollo de software, caracterizados por la rigidez de su carácter normativo y su gran dependencia de la planificación detallada previa al desarrollo, se decidieron por el termino Métodos Ágiles.

El manifiesto ágil se compone de cuatro valores fundamentales de los que se derivan doce principios.

Los 4 Valores del Manifiesto Ágil.

El manifiesto ágil comienza así: *"Estamos descubriendo mejores formas de desarrollar software haciendo y ayudando a otros a hacerlo. A través de este trabajo hemos llegado a valorar:*

- a los individuos y las interacciones del equipo sobre el proceso y las herramientas.
- el desarrollar software que funcione más que obtener una exhaustiva documentación.
- la colaboración con el cliente más que negociación de un contrato.
- el responder a los cambios más que seguir estrictamente un plan.

12 principios del manifiesto ágil



nextidealab.pe



Figura 10.

4.3.2 SCRUM

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible. Estas prácticas se apoyan unas a otras.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

El proceso

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

Planificación de la iteración

Se realiza la planificación de la iteración. Tiene dos partes:

- 1. Selección de requisitos.** El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

2. Planificación de la iteración. El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas grupal o individualmente.

En este proyecto se utiliza SCRUM como metodología de trabajo.

Algunos aspectos a tener en cuenta:

- En este caso no existe equipo, ni en sí la figura de “Product Owner” y “Scrum Master” al ser un producto propio.
- Se decidió que las iteraciones tuvieran una duración de una semana. Del tiempo disponible (académico), se optó por:
 - Se comienza el proyecto a marzo de 2022.
 - Invertir 1 semana en una fase de investigación y generar la arquitectura base del proyecto. Una vez analizadas las tecnologías necesarias, se construye los diagramas generales de casos de uso de la aplicación y definimos la interfaz gráfica del usuario (GUI), la estructura de los componentes (HTML y CSS) dejando para el resto de los Sprints el desarrollo de la aplicación.
- El número de días trabajados por semana serán por lo general tres (viernes, sábado y domingo). Por lo que cada iteración aplica a solo estos tres días.
- Al final de cada iteración se observan los resultados obtenidos, para ver si las tareas se han realizado correctamente, ver qué cosas se pueden mejorar y eliminar aquellas que no dan resultado con el objetivo de mejorar en las siguientes iteraciones.

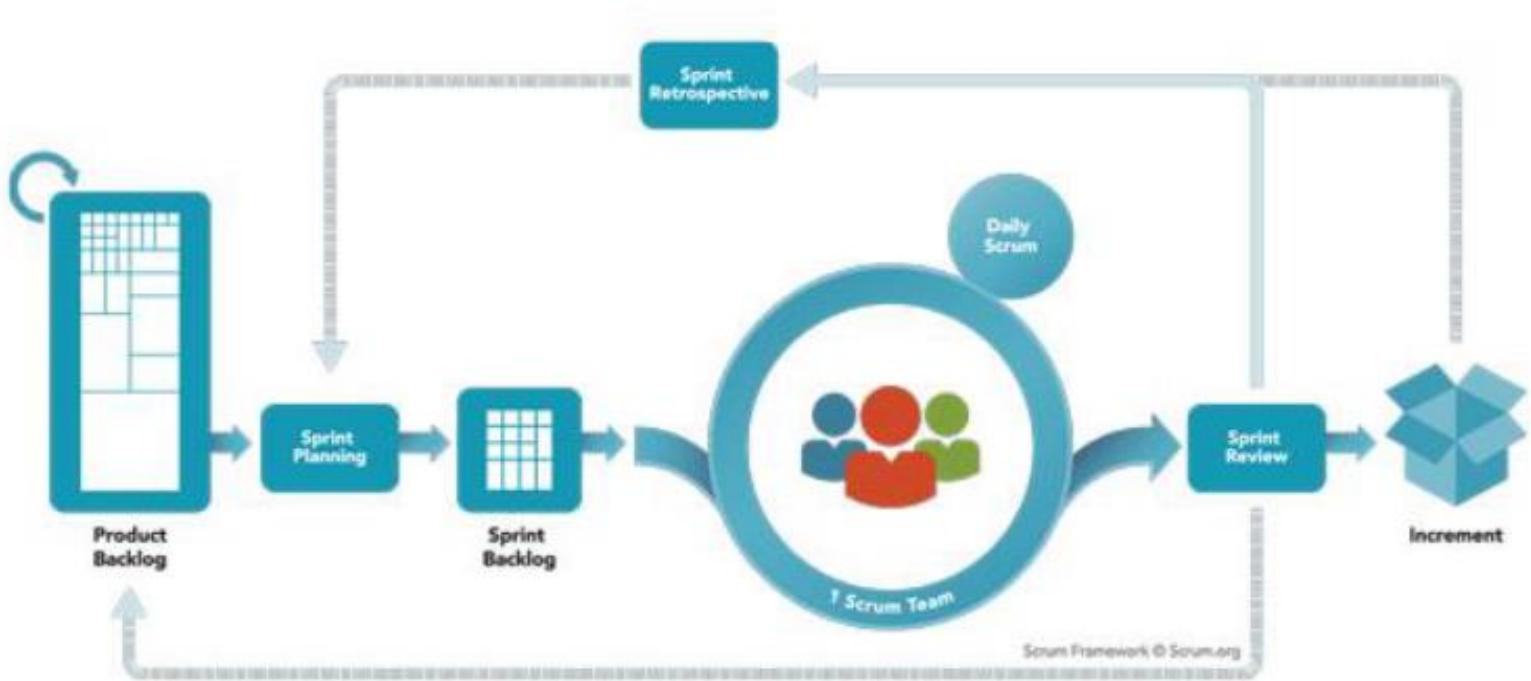
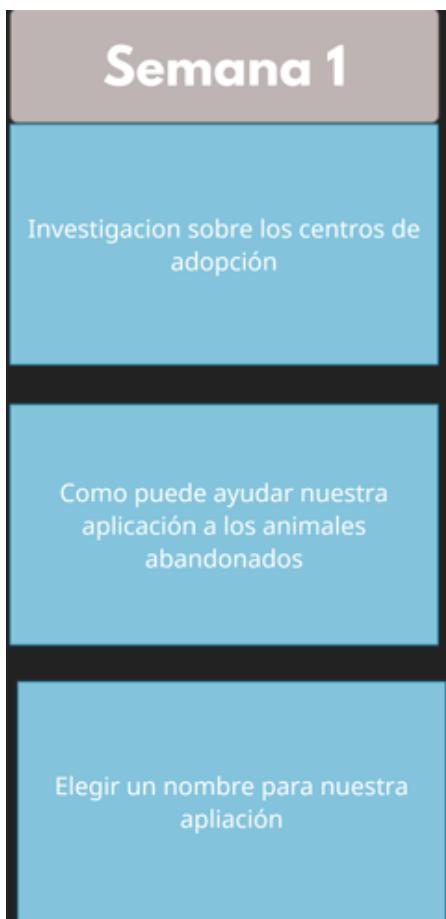


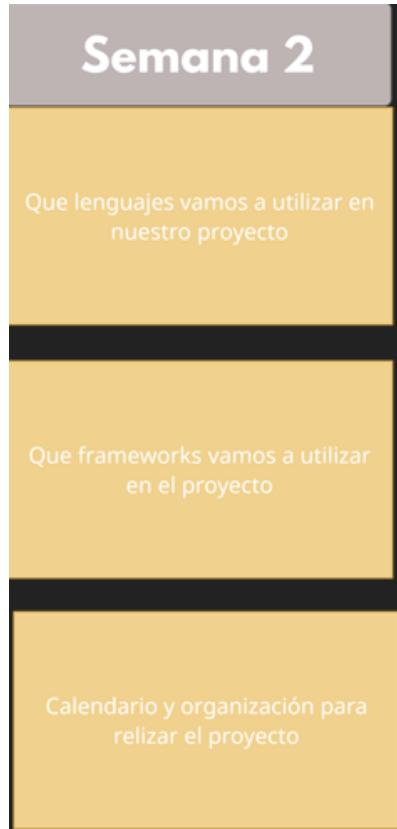
Figura 11.

Estructura y planificación



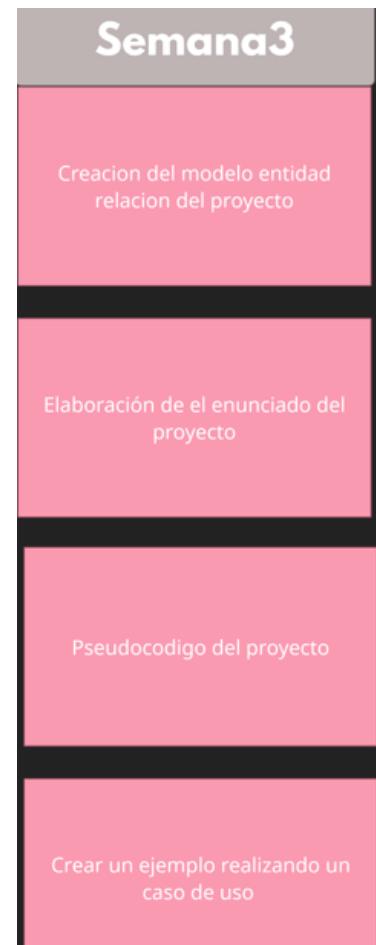
Semana 1. El trabajo se enfocó en recopilar y buscar información sobre los problemas que supone el abandono de animales, como se puede ayudar a que esto no suceda.

Plantear que tipo de aplicación se desea realizar, ir encaminado que se requieren requisitos, nombre... etc.



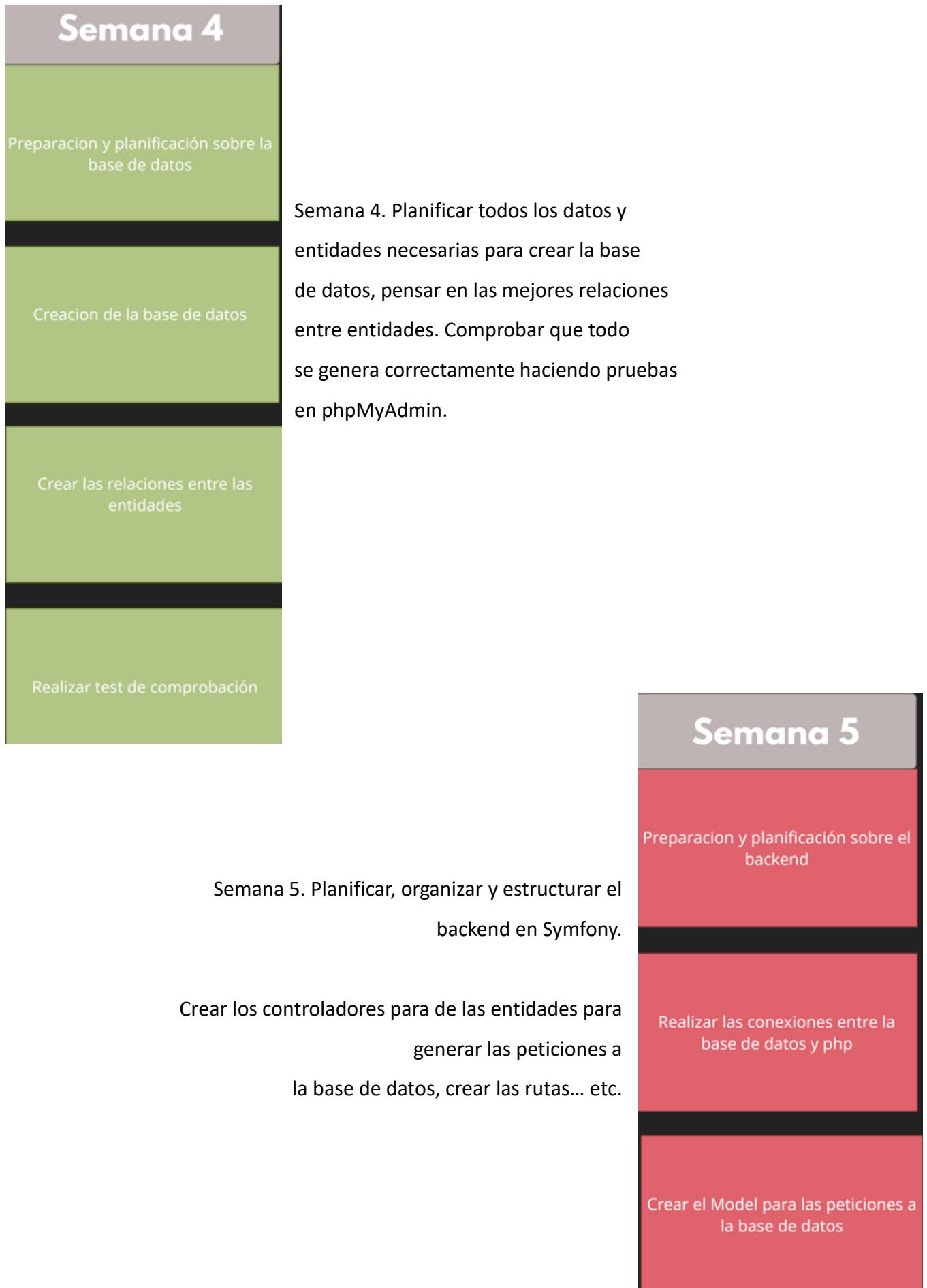
Semana 2. En la segunda semana se estudia que estructura, planificación, herramientas y lenguajes va a ser más efectivo y por qué.

También la organización y disposición de trabajo, que metodología y porqué.



Semana 3. Creación del enunciado con los requisitos de la aplicación. Realizar algún caso de uso para plantear el camino a seguir y si esta todo correcto.

Crear modelo entidad relación para ver cómo se va a estructurar.



Semana 6

Realizar test sobre el backend y la base de datos

Semana 6. Realizar varias pruebas subiendo datos con Postman para ver si surgen problemas con las relaciones y funciona todo de manera correcta. Corregir todos los errores que vayan surgiendo.

Corregir todos los errores que surjan

Empezar a planificar como montar el frontend y ver que necesidades pueden surgir.

Preparación y planificación sobre el frontend

Semana 7. Realizar la conexión entregado Angular y Symfony.

Creación de login y registro tanto de usuario como de animales.

Comprobación de que se insertan en la base de datos de manera correcta y si no es así corregir errores.

Semana 7

Realizar las conexiones y peticiones entre Javascript y php

Crear login y registro de usuarios y el registro de los animales

Comprobar que se insertan correctamente en la base de datos

Semana 8

Elaboracion de las distintas vistas de los usuarios

Semana 8. Crear y maquetar las distintas funciones de los usuarios (ver un animal, ver los animales del usuario, ver y editar los datos del propio usuario...).

Crear y comprobar la seguridad de las vistas de los usuarios

Comprobaciones de seguridad, que un usuario sin hacer el login no pueda acceder a ninguna ruta.

Semana 9

Semana 9. Creación del entorno visual tipo barra de menú y footer que será común para casi todas las vistas.

Elaboración de la página del home del usuario.

Volver a comprobar desde el principio que todo está correcto en frontend y backend.

Corregir los errores que surjan.

Crear el header, footer y barra de menu que sera comun para todos

Crear pagina de inicio

Realizar test de todo lo realizado hasta el momento

Semana 10

Empezar con el diseño del proyecto

Semana 10. Comienzo del diseño, moldeado y estilos de la aplicación.

Crear un entorno visual agradable, bonito, que invite al usuario a entrar y quiera adoptar animales y se sienta cómodo y satisfecho.

Crear los CSS para las distintas vistas

Subir el proyecto a Git hub.

Subir el proyecto a git y al dominio

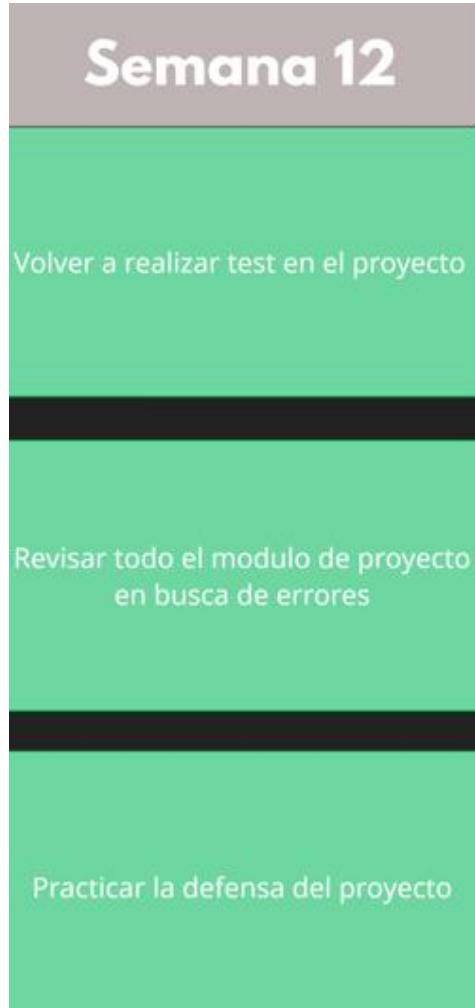
Semana 11

Semana 11. Una vez realizado todo y diseñado la aplicación volver a probar desde el principio test en busca de errores y posibles brechas de seguridad.

Realizar comprobaciones del escrito del proyecto que se ha ido generando en busca de errores, si esta todo correcto realizar la encuadernación.

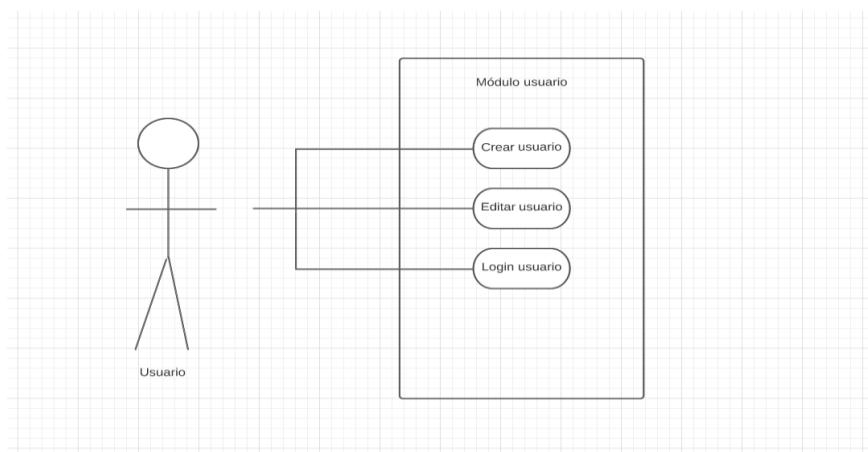
Comprobar y buscar errores y brechas de seguridad

Elaborar, preparar y encuadrinar todo el modulo de poryecto



4.4 CASOS DE USO Y ESTRUCTURA

A continuación, se muestra el diagrama de casos de uso de este módulo:



En este caso de uso cuando un usuario llega a la aplicación puede realizar tres acciones:

1. Registrarse como nuevo usuario, donde deberá llenar sus datos: nombre, apellidos, email, password, edad, ciudad, teléfono e imagen de perfil.
2. Una vez que el usuario este registrado aparece un mensaje diciendo que todo está correcto y desde ese mensaje puede acceder al login, donde se le pide el email y el password.
3. Antes de llegar al home aparece una pantalla donde se podrá elegir unos intereses o preferencias para encontrar nuestra mascota.
4. Realizada la anterior tarea se redirige al home de la aplicación donde podrá ver sus datos y editar su perfil entre otras cosas.

En la siguiente imagen se puede observar el registro en Angular que obtiene los datos de un formulario.

```

EXPLORADOR ... ts register.component.ts ✘
EDITORIOS ABIERTOS src > app > components > ts register.component.ts > RegisterComponent
ANGULAR
src
app
components
ts animal.adoptar.component.ts
ts animal.detail.component.ts
ts animal.edit.component.ts
ts animal.new.component.ts
ts animalesUser.component.ts
ts inicio.ts
ts login.component.ts
ts register.component.ts
ts user.component.ts
ts user.edit.component.ts
models
services
ts animal.services.ts
ts global.ts
ts imagen.services.ts
ts user.services.ts
views
animal.adoptar.html
animal.detail.html
animal.edit.html
animal.new.html
animalesUser.html
inicio.html
login.html
ESQUEMA
LÍNEA DE TIEMPO
SONARLINT RULES
SONARLINT ISSUE LOCATIONS
... register.component.ts ...
25 private _userService: UserService
26 )
27 {
28     this.title = "Registrate";
29     this.user = new User("", "", "", 0, "", 0, "", "");
30 }
31 ngOnInit() {
32     console.log("registro.component cargado");
33 }
34
35 //Enviamos los datos del usuario al servidor
36 enviarDatos():any{
37     this._userService.register(this.user, this.file).subscribe(
38         response => {
39             this.status = response;
40             if(this.status.status != "error"){
41                 this.status = "success";
42                 this.user = new User("", "", "", 0, "", 0, "", "");
43             }else{
44                 this.status = "error";
45                 this.user = new User("", "", "", 0, "", 0, "", "");
46             }
47         }
48     );
49 }
50 }
51
52 fileChangeEvent(event: any): void {
53     this.imageChangedEvent = event;
54     this.nombreFoto = event.target.files[0].name;
55 }
56
57
58 onFileChange(event: any) {
59     if (event.target.files.length > 0) {
60         const file = event.target.files[0];
61         this.file = file;
62         console.log(file);
63     }
64 }

```

aster* ⌂ ⌂ 0 △ 0 [TypeScript Importer] Symbols: 48 - INSERT - Lin. 35, col. 45 Espacios: 4 UTF-8

En esta imagen se encuentra el registro de lado de la API en Symfony. Donde llegan los datos como json y se decodifican para poderlos tratar y guardarlos en la base de datos.

```

SaulMint [Running]
/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Controller/UserController.php (symfony) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
syfony
app
BackendApiPet
bin
src
  AppBundle
    Controller
      AnimalController.php
      DefaultController.php
      UserController.php
    Resources
    Services
      AppBundle.php
  BackendBundle
    uploads
    .htaccess
tests
var
vendor
web
  .gitignore
  composer.json
  composer.lock
  LICENSE
  phpunit.xml.dist
  README.md

Line 1, Column 1
master (T)
Spaces: 4

```

```

UserController.php

16 class UserController extends Controller{
17     public function newAction(Request $request){
18         $helpers = $this->get('Helpers::class');
19         $json = $request->get('json', null);
20         $params = json_decode($json);
21
22         $data = array(
23             'status' => 'error',
24             'code' => 400,
25             'msg' => 'User not created'
26         );
27
28         if($json != null){
29             $email = (isset($params->email)) ? $params->email : null;
30             $nombre = (isset($params->nombrel)) ? $params->nombrel : null;
31             $apellidos = (isset($params->apellidos)) ? $params->apellidos : null;
32             $password = (isset($params->password)) ? $params->password : null;
33             $ciudad = (isset($params->ciudad)) ? $params->ciudad : null;
34             $telefono = (isset($params->telefono)) ? $params->telefono : null;
35             $edad = (isset($params->edad)) ? $params->edad : null;
36             $imagen = (isset($params->file)) ? $params->file : null;
37             $role = 'ROLE_USER';
38
39             $emailConstraint = new Assert\Email();
40             $emailConstraint->message = "El email no es válido";
41             $validate_email = $this->get("validator")->validate($email, $emailConstraint);
42
43             if(\count($validate_email) == 0 && $email != null && $nombre != null && $apellidos != null && $password != null && $ciudad != null && $telefono != null && $edad != null && $imagen != null){
44                 $user = new User();
45                 $user->setEmail($email);
46                 $user->setNombre($nombre);
47                 $user->setApellidos($apellidos);
48                 $user->setRoles($role);
49                 $user->setCiudad($ciudad);
50                 $user->setTelefono($telefono);
51                 $user->setEdad($edad);
52             }
53         }
54     }
55 }
56

```

En la siguiente imagen se observa el login en Symfony que es la parte del backend. Donde se comprueba si el email es correcto y existe, igualmente con la contraseña.

```

SaulMint [Running]
/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Controller/DefaultController.php (symfony) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
syfony
app
BackendApiPet
bin
src
  AppBundle
    Controller
      AnimalController.php
      DefaultController.php
      UserController.php
    Resources
    Services
      AppBundle.php
  BackendBundle
    uploads
    .htaccess
tests
var
vendor
web
  .gitignore
  composer.json
  composer.lock
  LICENSE
  phpunit.xml.dist
  README.md

Line 1, Column 1
master (T)
Spaces: 4

```

```

DefaultController.php

19     'base_dir' => realpath($this->getParameter('kernel.project_dir')).DIRECTORY_SEPARATOR,
20     ]);
21 }
22
23 public function loginAction(Request $request){
24     $helpers = $this->get('Helpers::class');
25
26     //Recibir json por Post
27     $json = $request->get('json', null);
28
29     //Array de datos a devolver por defecto
30     $data = array(
31         'status' => 'error',
32         'code' => 400,
33         'msg' => 'Login no válido'
34     );
35
36     if($json != null){
37         //Hacemos el login
38         //Convertimos un json a un objeto
39         $params = json_decode($json);
40
41         //Comprobamos si los datos son correctos y no estan vacios
42         $email = (isset($params->email)) ? $params->email : null;
43         $password = (isset($params->password)) ? $params->password : null;
44         $getHash = (isset($params->getHash)) ? $params->getHash : null;
45
46         //Validamos el email
47         $emailConstraint = new Assert\Email();
48         $emailConstraint->message = "El email no es válido";
49         $validate_email = $this->get("validator")->validate($email, $emailConstraint);
50
51         //Ciframos la contraseña
52         $pwd = hash('sha256', $password);
53
54         //Hacemos las comprobaciones
55         if(\count($validate_email) == 0 && $email != null && $password != null){
56             $jwt_Auth = $this->get(jwtAuth::class);
57
58             if($getHash == null || $getHash == false){$f
59

```

Una vez efectuado el login se crea un método que genera un token de autenticación con los datos del usuario.

```

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
symfony
app
BackendApiPet
bin
src
  AppBundle
    Controller
      AnimalController.php
      DefaultController.php
      UserController.php
    Resources
      config
    Services
      Helpers.php
    jwtAuth.php
      AppBundle.php
      BackendBundle
      uploads
        .htaccess
tests
var
vendor
web
.gitignore
/* composer.json
composer.lock
LICENSE
phpunit.xml.dist
README.md
Line 1, Column 1

```

```

/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Services/jwtAuth.php (Symfony) - Sublime Text (UNREGISTERED)
public function singUp($email, $password, $getHash = NULL){
    $user = $this->manager->getRepository('BackendBundle:User')->findOneBy(array(
        'email' => $email,
        'password' => $password
    ));
    $singUp = false;
    $data = array();
    if(is_object($user)){
        $singUp = true;
    }
    if($singUp == true){
        //Generamos el token JWT
        $token = array(
            'sub' => $user->getId(),
            'email' => $user->getEmail(),
            'name' => $user->getNombre(),
            'surname' => $user->getApellidos(),
            'iat' => time(),
            'exp' => time() + (7 * 24 * 60 * 60),
            'user' => [
                'id' => $user->getId(),
                'email' => $user->getEmail(),
                'name' => $user->getNombre(),
                'surname' => $user->getApellidos(),
                'ciudad' => $user->getCiudad(),
                'edad' => $user->getEdad(),
                'telefono' => $user->getTelefono(),
                'roles' => $user->getRoles(),
                'password' => $user->getPassword(),
            ],
        );
        $jwt = JWT::encode($token, $this->key, 'HS256');
        $decoded = JWT::decode($jwt, $this->key, array('HS256'));
        if($getHash == NULL){
    
```

El siguiente método sirve para verificar que el token pertenece al usuario que esta “logueado”. Es un método de seguridad.

```

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
symfony
app
BackendApiPet
bin
src
  AppBundle
    Controller
      AnimalController.php
      DefaultController.php
      UserController.php
    Resources
      config
    Services
      Helpers.php
    jwtAuth.php
      AppBundle.php
      BackendBundle
      uploads
        .htaccess
tests
var
vendor
web
.gitignore
/* composer.json
composer.lock
LICENSE
phpunit.xml.dist
README.md
Line 1, Column 1

```

```

/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Services/jwtAuth.php (Symfony) - Sublime Text (UNREGISTERED)
    }
} else{
    $data = array(
        'status' => 'error',
        'code' => 400,
        'msg' => 'El usuario no se ha podido identificar'
    );
}
return $data;
}

public function checkToken($jwt, $getIdentity = false){
    $auth = false;
    try{
        $decoded = JWT::decode($jwt, $this->key, array('HS256'));
    }catch(\UnexpectedValueException $e){
        $auth = false;
    }catch(\DomainException $e){
        $auth = false;
    }
    if(isset($decoded) && is_object($decoded) && isset($decoded->sub)){
        $auth = true;
    }else{
        $auth = false;
    }
    if($getIdentity == true){
        return $decoded;
    }else{
        return $auth;
    }
}

```

En la parte de Symfony. Función que sirve para editar el usuario que está “logueado”. Donde se verificar que coincide el token con ese usuario y si le pasa por parámetro el “id” del usuario para posteriormente modificar los datos.

```

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  angular
  .angular
  .vscode
  node_modules
  src
    app
      components
        animal.adoptar.component.ts
        animal.detail.component.ts
        animal.edit.component.ts
        animal.new.component.ts
        animalesUser.component.ts
        inicio.ts
        login.component.ts
        register.component.ts
        user.component.ts
        user.edit.component.ts
      models
      services
      views
    app-routing.module.ts
    app.component.css
    app.component.html
    app.component.spec.ts
    app.component.ts
    app.module.ts
  assets
  environments
  favicon.ico
Line 1, Column 1

```

```

/login.component.ts
1 import { Component, OnInit } from "@angular/core";
2 import { Router, ActivatedRoute, Params } from "@angular/router";
3 import { UserService } from "../services/user.services";
4
5 @Component({
6   selector: "login",
7   templateUrl: "../views/login.html",
8   providers: [UserService]
9 })
10 export class LoginComponent implements OnInit {
11   public title: string;
12   public user: any;
13   public token: any;
14   public identity: any;
15
16   constructor(
17     private route: ActivatedRoute,
18     private router: Router,
19     private _userService: UserService
20   ) {
21     this.title = "Identificate";
22     this.user = {
23       "email": "",
24       "password": "",
25       "getHash": "true"
26     };
27   }
28
29   ngOnInit() {
30     console.log("Login cargado");
31     this.logout();
32     this.redirectIfIdentity();
33   }
34
35   redirectIfIdentity(){
36     console.log("redirectIfIdentity");
37     let identity = this._userService.getIdentity();
38
39     if(identity != null && identity.sub){
40       this._router.navigate(["/user"]);
41     }
42   }
}

```

Login en Angular donde recogemos los datos del usuario y los enviamos a la API Symfony.

```

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  symfony
    app
    BackendApiPet
    bin
    src
      AppBundle
        Controller
          AnimalController.php
          DefaultController.php
          UserController.php
        Resources
        config
        Services
          Helpers.php
          jwtAuth.php
        AppBundle.php
      BackendBundle
      uploads
        .htaccess
    tests
    var
    vendor
    web
      .gitignore
    composer.json
    composer.lock
    LICENSE
    phpunit.xml.dist
    README.md
Line 1, Column 1

```

```

/UserController.php
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
public function editAction(Request $request){
  $helpers = $this->get('Helpers');
  $jwt_auth = $this->get('jwtAuth');
  $token = $request->get('authorization', null);
  $authCheck = $jwt_auth->checkToken($token);

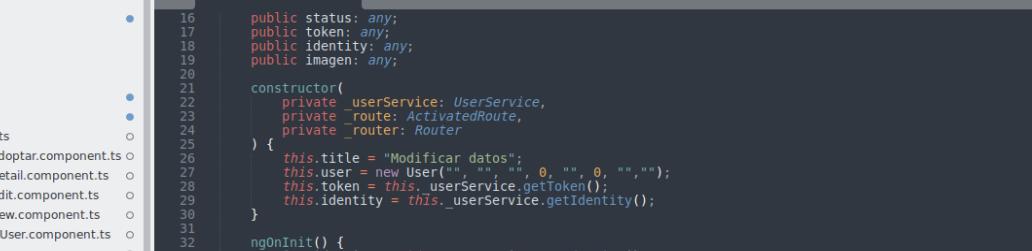
  if($authCheck){
    //entity manager
    $em = $this->getDoctrine()->getManager();
    //conseguir los datos del usuario vía token
    $identity = $jwt_auth->checkToken($token, true);
    //conseguir el usuario a actualizar
    $id = "";
    $id = $identity->sub;

    //recoger datos del post
    $json = $request->get('json', null);
    $params = json_decode($json);

    //array de error por defecto
    $data = array(
      'status' => 'error',
      'code' => 400,
      'msg' => 'Error al editar el usuario'
    );
    if($json != null){
      $email = (isset($params->email)) ? $params->email : null;
      $nombre = (isset($params->nombre)) ? $params->nombre : null;
      $apellidos = (isset($params->apellidos)) ? $params->apellidos : null;
      $password = (isset($params->password)) ? $params->password : null;
      $ciudad = (isset($params->ciudad)) ? $params->ciudad : null;
      $telefono = (isset($params->telefono)) ? $params->telefono : null;
      $edad = (isset($params->edad)) ? $params->edad : null;
      //Imagen = (isset($params->imagen)) ? $params->imagen : null;
      $emailConstraint = new Assert\Email();
      $emailConstraint->message = "El email no es válido";
      $validate_email = $this->get('validator')->validate($email, $emailConstraint);
      if($validate_email->count() == 0 && $email != null && $nombre != null && $apellidos != null && $ciudad != null && $telefono != null && $edad != null && $password != null && $telefono != null && $email != null && $email != ''){
        $user = $em->getRepository('AppBundle:User')->findOneBy(array(
          'email' => $email
        ));
        if($user != null){
          $user->setNombre($nombre);
          $user->setApellidos($apellidos);
          $user->setCiudad($ciudad);
          $user->setTelefono($telefono);
          $user->setEdad($edad);
          $user->setPassword($password);
          $em->persist($user);
          $em->flush();
        }
      }
    }
  }
}

```

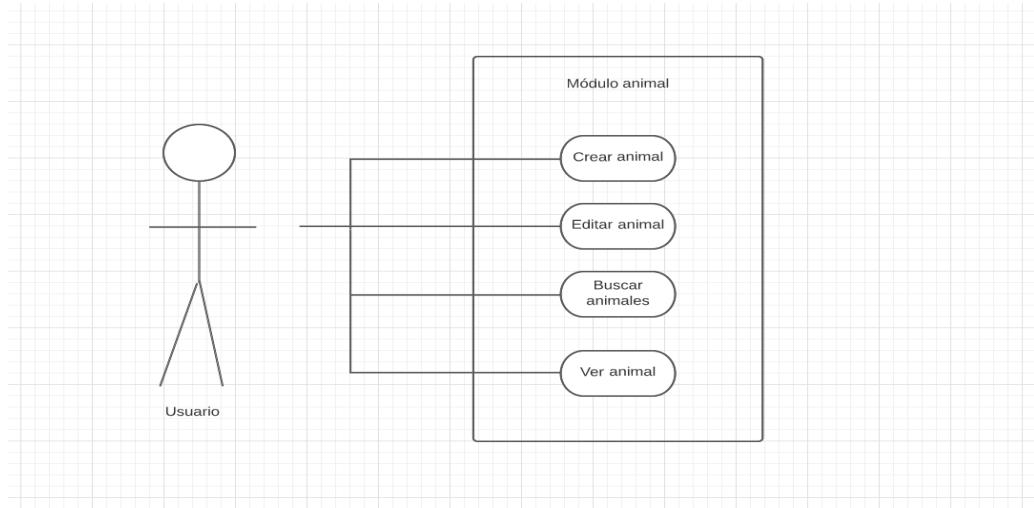
Editar usuario en Angular, se recogen los datos del usuario almacenados en el LocalStorage y se muestran en la vista para ser modificados.



```
user.edit.component.ts x
16  public status: any;
17  public token: any;
18  public identity: any;
19  public imagen: any;
20
21  constructor(
22    private _userService: UserService,
23    private _route: ActivatedRoute,
24    private _router: Router
25  ) {
26    this.title = "Modificar datos";
27    this.user = new User("", "", "", 0, "", 0, "", "");
28    this.token = this._userService.getToken();
29    this.identity = this._userService.getIdentity();
30  }
31
32  ngOnInit() {
33    let identity = this._userService.getIdentity();
34
35    if(identity == null || !identity.sub || identity == null){
36      this._router.navigate(["/login"]);
37    }else{
38      this.user = new User(this.identity.name, this.identity.surname, this.identity.email, this.identity.user
39    }
40  }
41
42  enviarDatos(){
43    this.userService.update(this.user).subscribe(
44      response => {
45        this.status = response;
46        if(this.status.status != "error"){
47          this.status = "success";
48          this.user = new User("", "", "", 0, "", 0, "", "");
49        }else{
50          this.status = "error";
51          this.user = new User("", "", "", 0, "", 0, "", "");
52        }
53      },
54      error => {
55        console.log(error);
56      }
57    );
58  }
59
60
```

El siguiente caso de uso es del módulo animal.

Donde el usuario logueado podrá crear un animal, modificar los animales que el haya puesto en adopción, buscar animales para poder adoptarlo y ver los datos específicos del animal que deseé.



Método que sirve para crear un animal en Angular y se envían los datos a la Api en Symfony.

Función de Angular que se utiliza para editar un animal comprobando el id para ver si coincide.

```

Line 1, Column 1
master (62) Spaces: 4

```

```

13  export class AnimalNewComponent implements OnInit {
14    public title: string;
15    public user: any;
16    public animal: any;
17    public token: any;
18    public identity: any;
19    public status: any;
20
21    constructor(
22      private _route: ActivatedRoute,
23      private _router: Router,
24      private _userService: UserService,
25      private _animalService: AnimalService
26    ) {
27      this.title = "Dar en adopción";
28      this.user = new User("", "", "", 0, "", "", "");
29      this.token = this._userService.getToken();
30      this.identity = this._userService.getIdentity();
31    }
32
33    ngOnInit() {
34      let identity = this._userService.getIdentity();
35      if(identity == null || !identity.sub || identity == null){
36        this._router.navigate(["/login"]);
37      }else{
38        this.animal = new Animal(0, "", "", 0, "", "", "", 0);
39        this.user = new User(this.identity.name, this.identity.surname, this.identity.email, this.identity.user);
40      }
41    }
42
43    enviarDatos(){
44      this._animalService.create(this.token, this.animal).subscribe(
45        response =>{
46          this.status = response;
47          if(this.status.status != "error"){
48            this.status = "success";
49            this._router.navigate(["/animalesUser"]);
50          }else{
51            this.status = "error";
52            this.animal = new Animal(0, "", "", 0, "", "", "", 0);
53          }
54        }
55      );
56    }
57
58  }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

```

Método que se utiliza para realizar el filtro de búsqueda en Angular y se envían los datos vía API.

```

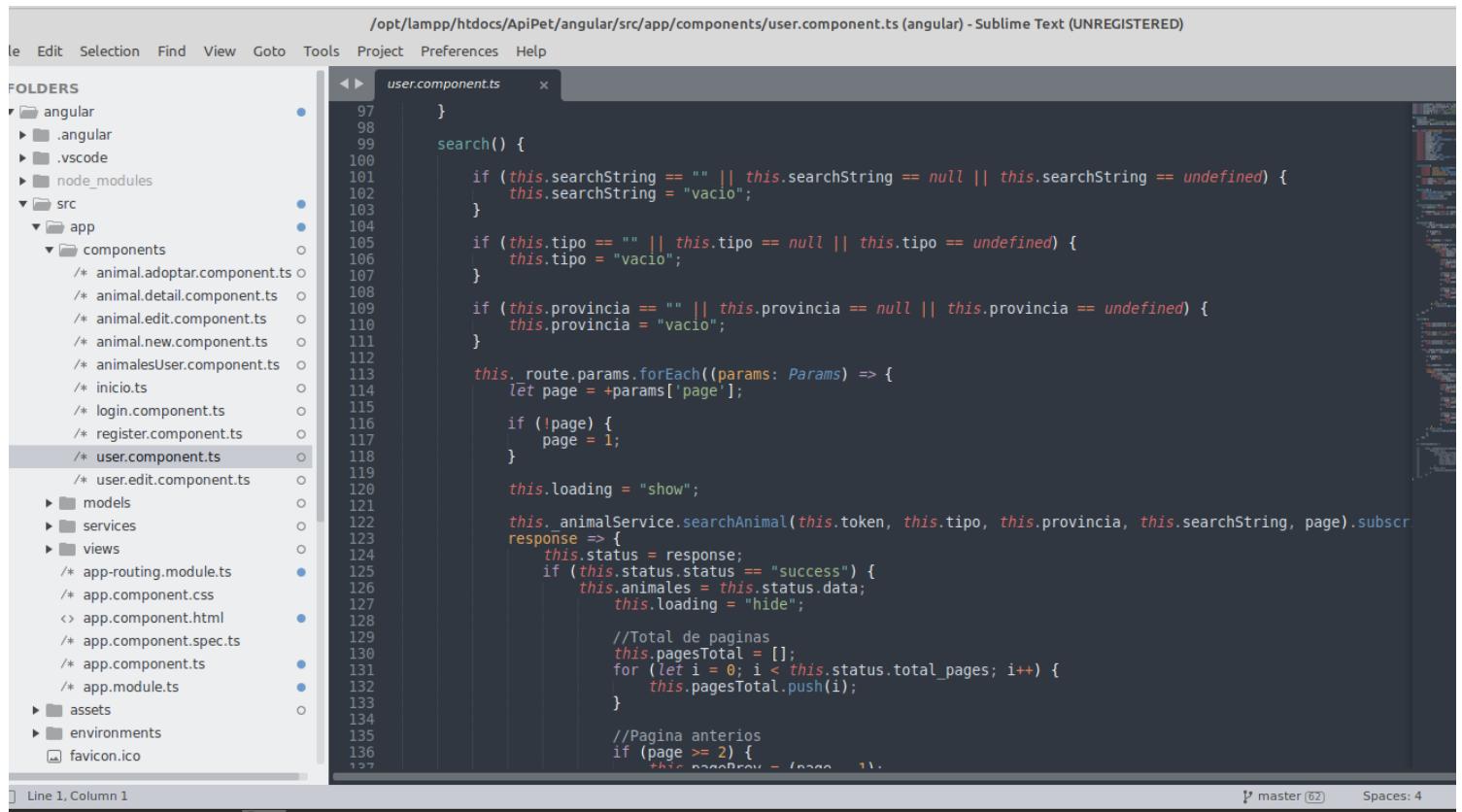
Line 1, Column 1
master (62) Spaces: 4

```

```

43  getAnimal(){
44    this.loading = "show";
45    this._route.params.forEach((params: Params) => {
46      let id = params["id"];
47
48      this._animalService.getAnimal(this.token, id).subscribe(
49        response =>{
50          this.status = response;
51          if(this.status.status == "success"){
52            this.loading = "hide";
53            this.animal = this.status.animal;
54            if(this.animal.userId != this.identity.sub){
55              this._router.navigate(["/user"]);
56            }else{
57              this.status = "success";
58            }
59          }else{
60            this.status = "error";
61          }
62          this.loading = "hide";
63        },
64        error =>{
65          this.status = "error";
66          this.loading = "hide";
67          console.log(<any>error);
68        }
69      );
70    });
71  }
72
73
74  enviarDatos(){
75    this.loading = "show";
76    this._route.params.forEach((params: Params) => {
77      let id = params["id"];
78      this._animalService.editAnimal(this.token, this.animal, id).subscribe(
79        response =>{
80          this.status = response;
81          if(this.status.status != "error"){
82            this.status = "success";
83            this.loading = "hide";
84            this._router.navigate(["/animalesUser"]);
85          }
86        }
87      );
88    });
89  }
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900

```



```

Line 1, Column 1
master (62) Spaces: 4

```

```

    ...
    }

    search() {
        if (this.searchString == "" || this.searchString == null || this.searchString == undefined) {
            this.searchString = "vacio";
        }

        if (this.tipo == "" || this.tipo == null || this.tipo == undefined) {
            this.tipo = "vacio";
        }

        if (this.provincia == "" || this.provincia == null || this.provincia == undefined) {
            this.provincia = "vacio";
        }

        this._route.params.forEach((params: Params) => {
            let page = +params['page'];

            if (!page) {
                page = 1;
            }

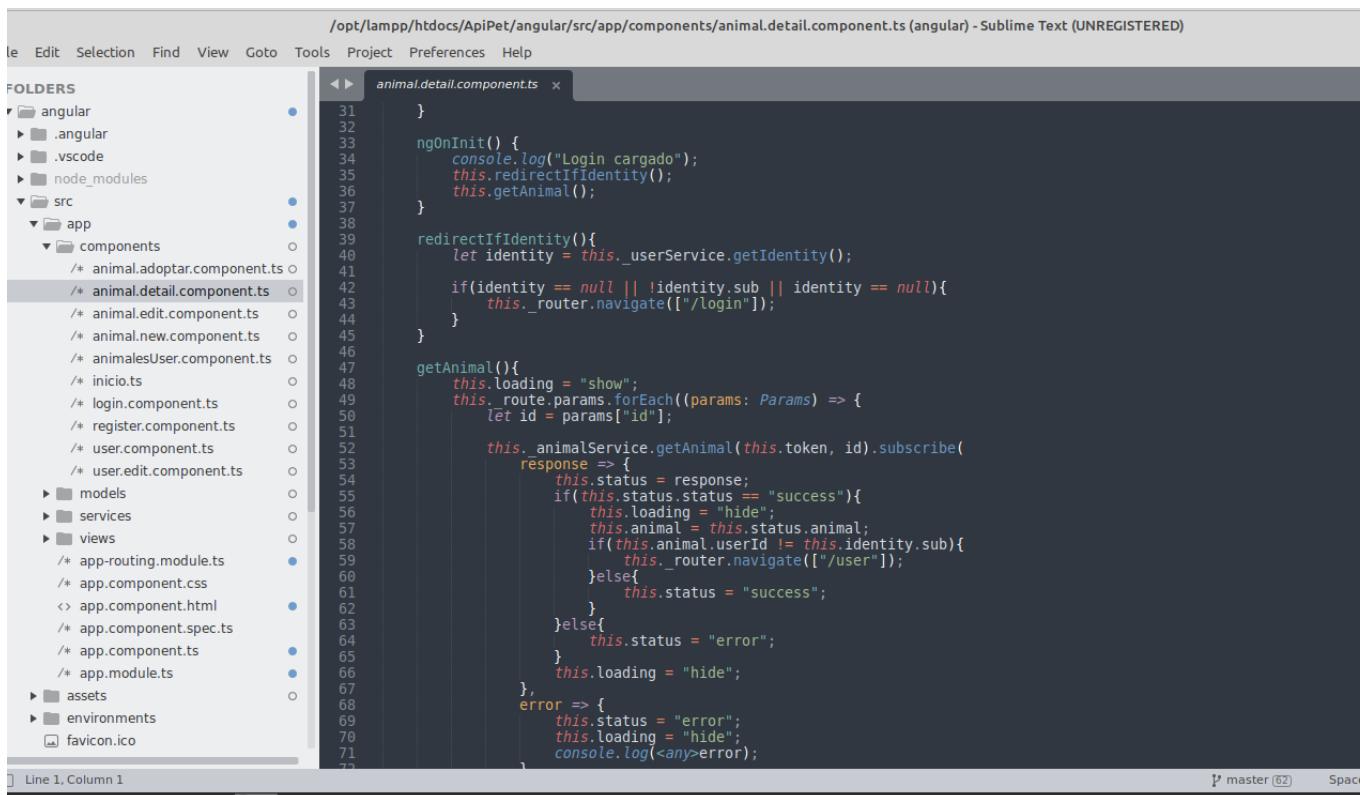
            this.loading = "show";

            this.animalService.searchAnimal(this.token, this.tipo, this.provincia, this.searchString, page).subscribe(
                response => {
                    this.status = response;
                    if (this.status.status == "success") {
                        this.animales = this.status.data;
                        this.loading = "hide";

                        //Total de paginas
                        this.pagesTotal = [];
                        for (let i = 0; i < this.status.total_pages; i++) {
                            this.pagesTotal.push(i);
                        }
                    }
                    //Pagina anteriores
                    if (page >= 2) {
                        this.pagePrev = page - 1;
                    }
                }
            );
        });
    }
}

```

Similar al de editar, muestra los datos del animal solo para visualizar y poder adoptarlo en Angular.



```

Line 1, Column 1
master (62) Spaces: 4

```

```

    ...
    }

    ngOnInit() {
        console.log("Login cargado");
        this.redirectIfIdentity();
        this.getAnimal();
    }

    redirectIfIdentity(){
        let identity = this._userService.getIdentity();

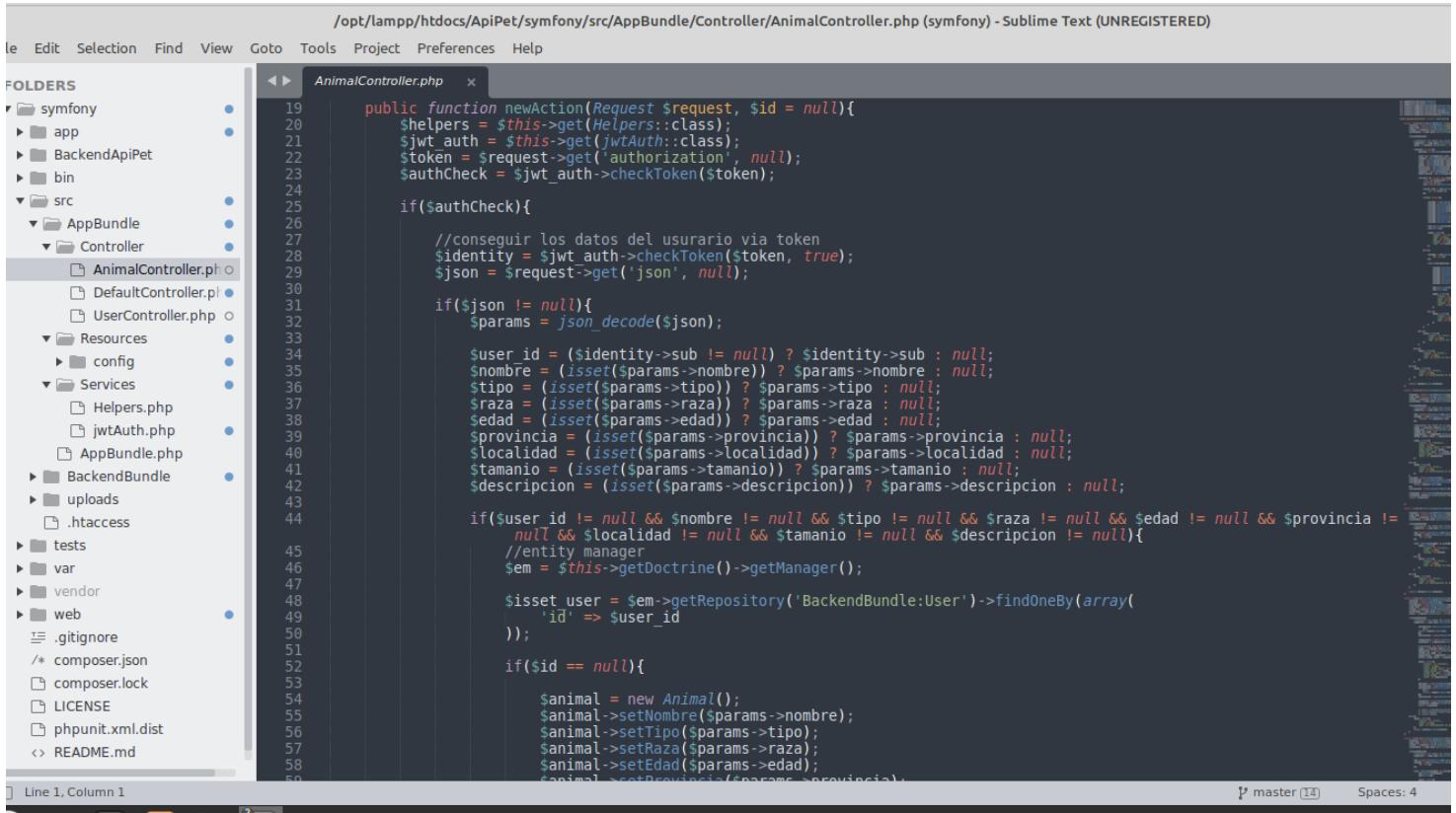
        if(identity == null || !identity.sub || identity == null){
            this._router.navigate(["/login"]);
        }
    }

    getAnimal(){
        this.loading = "show";
        this._route.params.forEach((params: Params) => {
            let id = params['id'];

            this.animalService.getAnimal(this.token, id).subscribe(
                response => {
                    this.status = response;
                    if(this.status.status == "success"){
                        this.loading = "hide";
                        this.animal = this.status.animal;
                        if(this.animal.userId != this.identity.sub){
                            this._router.navigate(["/user"]);
                        }else{
                            this.status = "success";
                        }
                    }else{
                        this.status = "error";
                    }
                    this.loading = "hide";
                },
                error => {
                    this.status = "error";
                    this.loading = "hide";
                    console.log(<any>error);
                }
            );
        });
    }
}

```

En este caso en el lado de Symfony para ahorrar y reutilizar código se ha creado el método de registrar y editar animal juntos. Si se recibe un “id” por parámetro es para editar comprobando si coincide y existe y si no se recibe es para crear uno nuevo.



```

/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Controller/AnimalController.php (symfony) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
src
  AppBundle
    Controller
      AnimalController.php
      DefaultController.php
      UserController.php
  Resources
  config
  Services
    Helpers.php
    jwtAuth.php
  AppBundle.php
  BackendBundle
  uploads
  .htaccess
  tests
  var
  vendor
  web
  .gitignore
  composer.json
  composer.lock
  LICENSE
  phpunit.xml.dist
  README.md

Line 1, Column 1

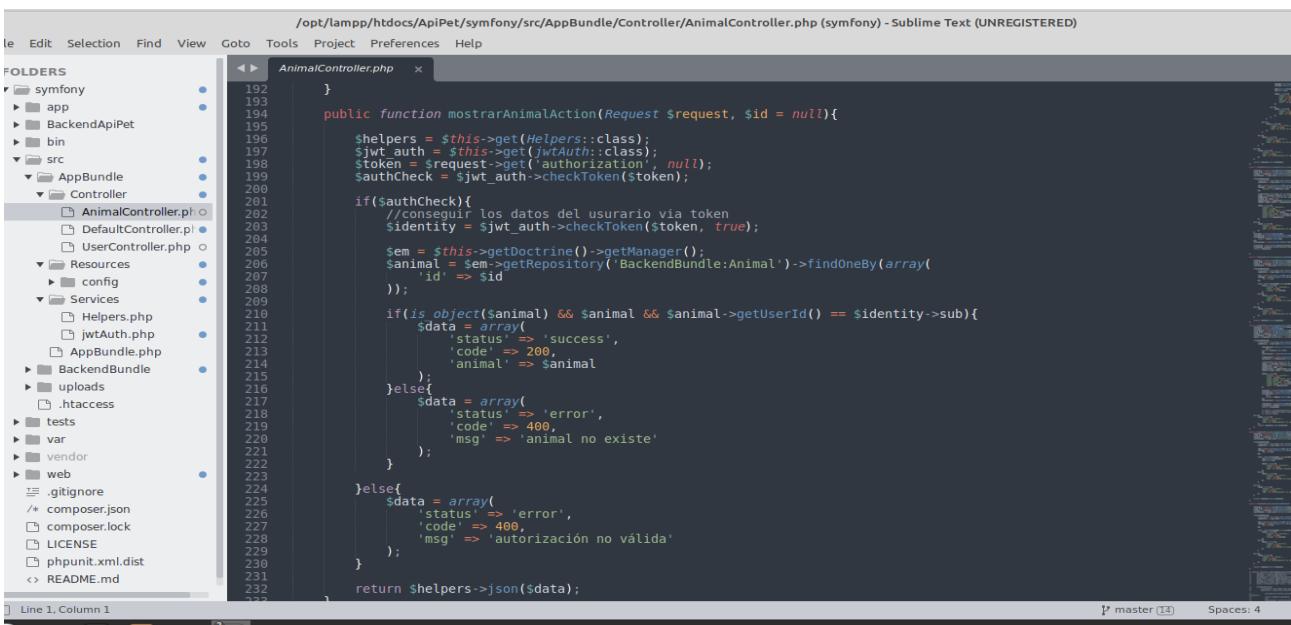
```

```

19 public function newAction(Request $request, $id = null){
20     $helpers = $this->get(Helpers::class);
21     $jwt_auth = $this->get(JwtAuth::class);
22     $stoken = $request->get('authorization', null);
23     $authCheck = $jwt_auth->checkToken($stoken);
24
25     if($authCheck){
26         //conseguir los datos del usuario via token
27         $identity = $jwt_auth->checkToken($stoken, true);
28         $json = $request->get('json', null);
29
30         if($json != null){
31             $params = json_decode($json);
32
33             $user_id = ($identity->sub != null) ? $identity->sub : null;
34             $nombre = (isset($params->nOMBRE)) ? $params->nOMBRE : null;
35             $tipo = (isset($params->tipo)) ? $params->tipo : null;
36             $raza = (isset($params->raZA)) ? $params->raZA : null;
37             $edad = (isset($params->edad)) ? $params->edad : null;
38             $provincia = (isset($params->proVINCIA)) ? $params->proVINCIA : null;
39             $localidad = (isset($params->localIDAD)) ? $params->localIDAD : null;
40             $tamanio = (isset($params->tamANO)) ? $params->tamANO : null;
41             $descripcion = (isset($params->desCRIPCION)) ? $params->desCRIPCION : null;
42
43             if($user_id != null && $nombre != null && $tipo != null && $raza != null && $edad != null && $provincia != null && $localidad != null && $tamanio != null && $descripcion != null){
44                 //entity manager
45                 $em = $this->getDoctrine()->getManager();
46
47                 $isset_user = $em->getRepository('BackendBundle:User')->findOneBy(array(
48                     'id' => $user_id
49                 ));
50
51                 if($id == null){
52                     $animal = new Animal();
53                     $animal->setNombre($params->nOMBRE);
54                     $animal->setTipo($params->tipo);
55                     $animal->setRaza($params->raZA);
56                     $animal->setEdad($params->edad);
57                     $animal->setProvincia($params->proVINCIA);
58
59                 }
60
61             }
62
63         }
64
65     }
66
67 }

```

Para ver un animal en la vista se envía por API los datos del animal a través de un “id” para comparar.



```

/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Controller/AnimalController.php (symfony) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
src
  AppBundle
    Controller
      AnimalController.php
      DefaultController.php
      UserController.php
  Resources
  config
  Services
    Helpers.php
    jwtAuth.php
  AppBundle.php
  BackendBundle
  uploads
  .htaccess
  tests
  var
  vendor
  web
  .gitignore
  composer.json
  composer.lock
  LICENSE
  phpunit.xml.dist
  README.md

Line 1, Column 1

```

```

192     }
193
194     public function mostrarAnimalAction(Request $request, $id = null){
195         $helpers = $this->get(Helpers::class);
196         $jwt_auth = $this->get(JwtAuth::class);
197         $stoken = $request->get('authorization', null);
198         $authCheck = $jwt_auth->checkToken($stoken);
199
200         if($authCheck){
201             //conseguir los datos del usuario via token
202             $identity = $jwt_auth->checkToken($stoken, true);
203
204             $em = $this->getDoctrine()->getManager();
205             $animal = $em->getRepository('BackendBundle:Animal')->findOneBy(array(
206                 'id' => $id
207             ));
208
209             if(is_object($animal) && $animal && $animal->getUser() == $identity->sub){
210                 $data = array(
211                     'status' => 'success',
212                     'code' => 200,
213                     'animal' => $animal
214                 );
215             }else{
216                 $data = array(
217                     'status' => 'error',
218                     'code' => 400,
219                     'msg' => 'animal no existe'
220                 );
221             }
222
223         }
224
225         else{
226             $data = array(
227                 'status' => 'error',
228                 'code' => 400,
229                 'msg' => 'autorización no válida'
230             );
231
232         }
233
234         return $helpers->json($data);
235
236     }
237
238 }

```

Función para buscar los animales y filtrarlos según los parámetros que recibamos.

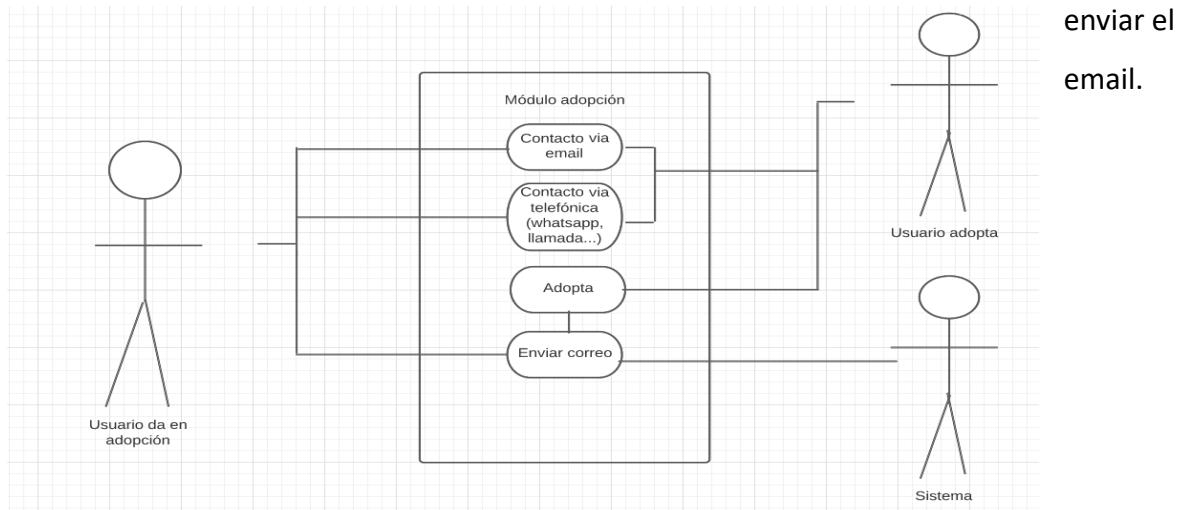
The screenshot shows the Sublime Text interface with the following details:

- File Path:** /opt/lampp/htdocs/apiPet/symfony/src/AppBundle/Controller/AnimalController.php
- Project Structure:** The sidebar displays the project structure under "FOLDERS". Key files shown include AnimalController.php, DefaultController.php, UserController.php, and various configuration and service files.
- Code Editor:** The main window shows the code for the AnimalController.php file. The code handles search requests for animals based on type, name, and province using Doctrine and JWT authentication.

```
449     public function searchAnimalesAction(Request $request, $tipo = null, $nombre = null, $provincia = null){
450         $helpers = $this->get('Helpers');
451         $encoder = new JsonEncoder();
452         $normalizer = new ObjectNormalizer();
453         $jwt_auth = $this->get('jwtAuth');
454         $token = $request->get('authorization', null);
455         $authCheck = $jwt_auth->checkToken($token);
456         $json = $request->get('json', null);
457
458         if($authCheck){
459             //conseguir los datos del usurario via token
460             $identity = $jwt_auth->checkToken($token, true);
461
462             $normalizer->setCircularReferenceHandler(function ($object, string $format = null, array $context = array(
463                 )) {
464                 return $object->getId();
465             });
466             $serializer = new Serializer(array($normalizer), array($encoder));
467
468             $repository = $this->getDoctrine()->getRepository(Animal::class);
469
470             if($tipo != "vacio" && $nombre != "vacio" && $provincia != "vacio"){
471                 $animales = $repository->findBy(array(
472                     'tipo' => $tipo,
473                     'nombre' => $nombre,
474                     'provincia' => $provincia
475                 ));
476             }elseif($tipo != "vacio" && $nombre != "vacio"){
477                 $animales = $repository->findBy(array(
478                     'tipo' => $tipo,
479                     'nombre' => $nombre
480                 ));
481             }elseif($tipo != "vacio" && $provincia != "vacio"){
482                 $animales = $repository->findBy(array(
483                     'tipo' => $tipo,
484                     'provincia' => $provincia
485                 ));
486             }elseif($nombre != "vacio" && $provincia != "vacio"){
487                 $animales = $repository->findBy(array(
488                     'nombre' => $nombre,
489                     'provincia' => $provincia
490                 ));
491             }
492         }
493     }
```

En el siguiente caso de uso vemos como un usuario desea adoptar un animal. Al pulsar el botón automáticamente el sistema envía un email al usuario que le pertenece el animal para que se pongan en contacto vía email, teléfono o como deseen.

Función de Angular que sirve para mostrar el animal que se desea adoptar para posteriormente



```

Line 1, Column 1
master (14) Spaces: 4
/opt/lampp/htdocs/ApiPet/angular/src/app/components/animal.adoptar.component.ts (angular) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  angular
    .angular
    .vscode
    node_modules
  src
    app
      components
        animal.adoptar.component.ts
        animal.detail.component.ts
        animal.edit.component.ts
        animal.new.component.ts
        animalesUser.component.ts
        inicio.ts
        login.component.ts
        register.component.ts
        user.component.ts
        user.edit.component.ts
      models
      services
      views
      app-routing.module.ts
      app.component.css
      app.component.html
      app.component.spec.ts
      app.component.ts
      app.module.ts
    assets
    environments
    favicon.ico
animal.adoptar.component.ts
10  providers: [UserService, AnimalService]
11 }
12
13 export class AnimalAdoptarComponent implements OnInit {
14   public title: string;
15   public animal: any;
16   public token: any;
17   public identity: any;
18   public status: any;
19   public loading: any;
20
21   constructor(
22     private _route: ActivatedRoute,
23     private _router: Router,
24     private _userService: UserService,
25     private _animalService: AnimalService
26   ) {
27     this.title = "¡Adoptame!";
28     this.identity = this._userService.getIdentity();
29     this.token = this._userService.getToken();
30   }
31
32   ngOnInit() {
33     console.log("Adoptar cargado");
34     this.redirectIfIdentity();
35     this.getAnimal();
36   }
37
38   redirectIfIdentity(){
39     let identity = this._userService.getIdentity();
40
41     if(identity == null || !identity.sub || identity == null){
42       this._router.navigate(['/login']);
43     }
44   }
45
46   getAnimal(){
47     this.loading = "show";
48     this._route.params.forEach((params: Params) => {
49       let id = params["id"];
50     });
51

```

Método de Symfony que devuelve el animal que se desea adoptar.

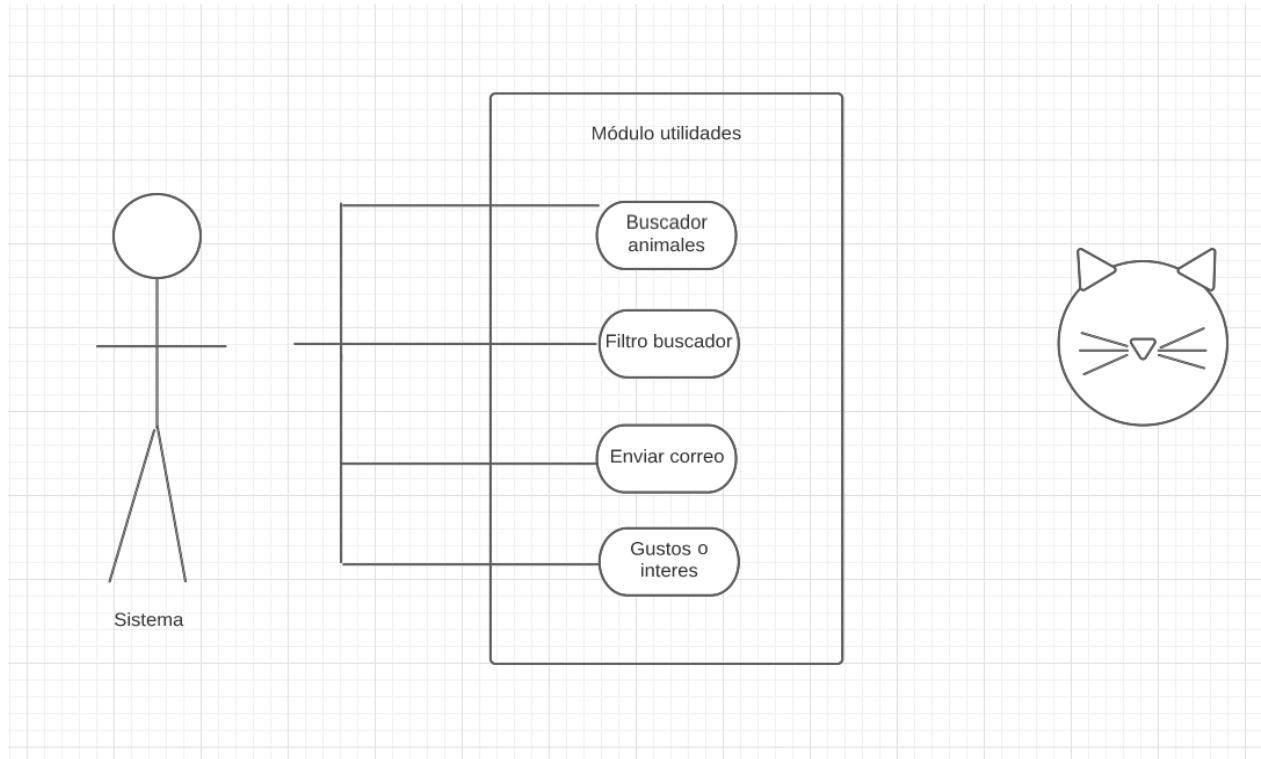
```

Line 1, Column 1
master (14) Spaces: 4
/opt/lampp/htdocs/ApiPet/symfony/src/AppBundle/Controller/AnimalController.php (symfony) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  symfony
    app
    BackendApiPet
    bin
    src
      AppBundle
        Controller
          AnimalController.php
          DefaultController.php
          UserController.php
      Resources
        config
      Services
        Helpers.php
        jwtAuth.php
      AppBundle.php
    BackendBundle
    uploads
    .htaccess
  tests
  var
  vendor
  web
  .gitignore
  composer.json
  composer.lock
  LICENSE
  phpunit.xml.dist
  README.md
AnimalController.php
337   return $helpers->json($data);
338 }
339
340 public function getAnimalAction(Request $request, $id = null){
341
342   $helpers = $this->get('Helpers');
343   $jwt_auth = $this->get('jwtAuth');
344   $token = $request->get('authorization', null);
345   $authCheck = $jwt_auth->checkToken($token);
346
347   if($authCheck){
348     //conseguir los datos del usuario via token
349     $identity = $jwt_auth->checkToken($token, true);
350
351     $em = $this->getDoctrine()->getManager();
352     $animal = $em->getRepository('BackendBundle:Animal')->findOneBy(array(
353       'id' => $id
354     ));
355
356     if(is_object($animal) && $animal){
357       $data = array(
358         'status' => 'success',
359         'code' => 200,
360         'animal' => $animal
361       );
362     }else{
363       $data = array(
364         'status' => 'error',
365         'code' => 400,
366         'msg' => 'animal no existe'
367       );
368     }
369
370   }else{
371     $data = array(
372       'status' => 'error',
373       'code' => 400,
374       'msg' => 'autorización no válida'
375     );
376   }
377
378   return $helpers->json($data);
379 }


```

El siguiente caso de uso son utilidades y funciones que ofrece el sistema para favorecer el uso de la aplicación al usuario.



Método que nos muestra los animales que pertenecen a un usuario en Angular.

```

// Code for animalesUser.component.ts
export class AnimalesUserComponent implements OnInit {
    ...
    getAnimalUser() {
        this.route.params.forEach((params: Params) => {
            let page = +params['page'];
            if(!page){
                page = 1;
            }
            this.loading = "show";
            this.animalService.animalesUsuario(this.token, page).subscribe(
                response => {
                    this.status = response;
                    if(this.status.status == "success"){
                        this.animales = this.status.data;
                        this.loading = "hide";
                    }
                    //Total de paginas
                    this.pagesTotal = [];
                    for(let i = 0; i < this.status.total_pages; i++){
                        this.pagesTotal.push(i);
                    }
                    //Pagina anteriores
                    if(page >= 2){
                        this.pagePrev = (page - 1);
                    }else{
                        this.pagePrev = page;
                    }
                    //Pagina siguiente
                    if(page < this.status.total_pages){
                        this.pageNext = (page + 1);
                    }else{
                        this.pageNext = page;
                    }
                }, error => {
                    console.log(error);
                }
            );
        });
    }
}

```

Función que nos devuelve todos los animales de la base de datos para mostrarlos en la vista de Angular junto con unos “paginator” para poder ir pasando de página y ver cuantas hay en total y así ayudar y favorecer al usuario.

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- symfony
- app
- BackendApiPet
- bin
- src
 - AppBundle
 - Controller
 - AnimalController.php
 - DefaultController.php
 - UserController.php
 - Resources
 - config
 - Services
 - Helpers.php
 - jwtAuth.php
 - AppBundle.php
- BackendBundle
- uploads
- .htaccess
- tests
- var
- vendor
- web
- .gitignore
- /* composer.json
- composer.lock
- LICENSE
- phunit.xml.dist
- README.md

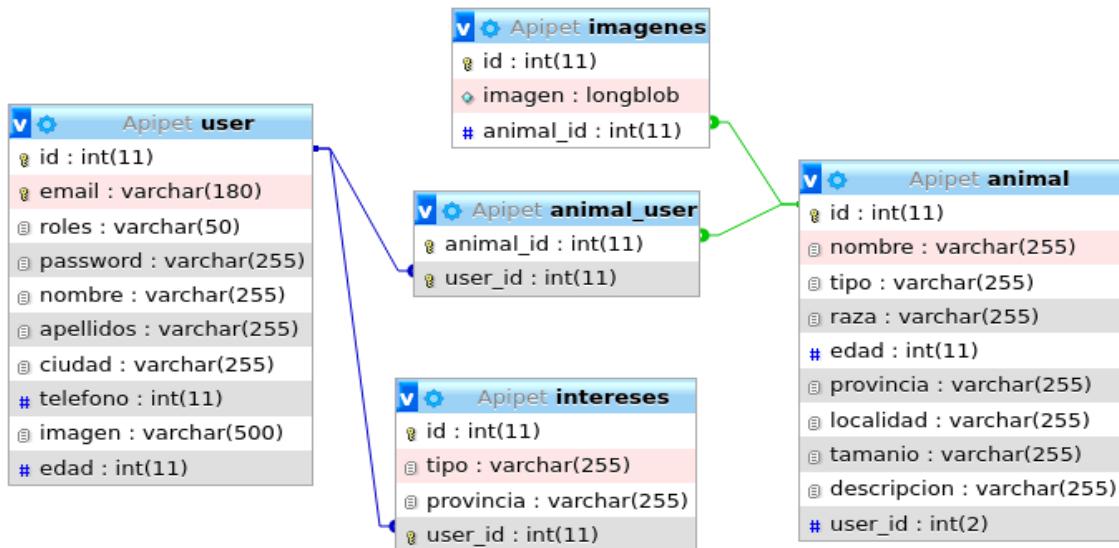
AnimalController.php x

```
234
235     public function buscadorAnimalesAction(Request $request){
236         $helpers = $this->get('Helpers::class');
237         $encoder = new JsonEncoder();
238         $normalizer = new ObjectNormalizer();
239         $jwt_auth = $this->get('jwtAuth::class');
240         $stoken = $request->get('authorization', null);
241         $authCheck = $jwt_auth->checkToken($stoken);
242         $json = $request->get('json', null);
243
244         if($authCheck){
245             //conseguir los datos del usurario via token
246             $identity = $jwt_auth->checkToken($token, true);
247
248             $normalizer->setCircularReferenceHandler(function ($object, string $format = null, array $context = array())
249                 ) {
250                 return $object->getId();
251             };
252             $serializer = new Serializer(array($normalizer), array($encoder));
253
254             $repository = $this->getDoctrine()->getRepository(Animal::class);
255             $animales = $repository->findAll();
256
257             //Recogemos los datos de la pagina que nos vienen por GET
258             $page = $request->query->getInt('page', 1);
259             $paginator = $this->get('knp_paginator');
260             $items_per_page = 9;
261             $pagination = $paginator->paginate($animales, $page, $items_per_page);
262             $total_items_count = $pagination->getTotalItemCount();
263
264             $data = array(
265                 'status' => 'success',
266                 'code' => 200,
267                 'total_items_count' => $total_items_count,
268                 'page actual' => $page,
269                 'items per page' => $items_per_page,
270                 'total pages' => ceil($total_items_count / $items_per_page),
271                 'data' => $pagination
272             );
273
274             $encoder = new JsonEncoder();
275             $normalizer = new ObjectNormalizer();
276         }
277     }
```

Line 1, Column 1

master 14 Spaces: 4

4.5 BASE DE DATOS



La base de datos ha sido creada desde Symfony mediante comandos en la terminal siguiendo los pasos que te da el propio Symfony, con esto se genera simultáneamente las tablas y relaciones en phpMyAdmin y las entidades y las mismas relaciones en Symfony.

En la tabla usuarios se pueden observar los atributos que tiene un usuario y las relaciones con intereses y animales.

La tabla animales tiene los atributos para generar un animal y las relaciones con un usuario y con imágenes ya un animal puede tener más de una imagen.

La tabla animal_user está en proyecto ya que es para la siguiente actualización, va a ser una tabla de favoritos y está relacionada con usuarios y animales de muchos a muchos. Un animal puede ser el favorito de muchos usuarios y un usuario puede tener muchos animales como favorito.

V CONCLUSIONES Y POSIBLES AMPLIACIONES

En este apartado se analizan las conclusiones a las que se ha llegado desarrollando este proyecto y las posibles actualizaciones y mejoras a realizar a futuro.

5.1. CONCLUSIONES

Con el desarrollo de este proyecto se ha conseguido aprender a generar una arquitectura capaz de crear una plataforma basada en componentes y fácilmente mantenible.

Además, desde el comienzo del proyecto se ha utilizado un marco de trabajo ágil facilitando el desarrollo del proyecto y su seguimiento.

5.2. TRABAJO FUTURO

Aunque se ha desarrollado los objetivos propuestos al comienzo del proyecto se pueden incluir al sistema las siguientes funcionalidades:

- Incluir geolocalización del usuario (animales).
- Incluir un mapa que permita localizar animales en adopción próximos a la posición de un usuario.
- Incluir un mapa para poder buscar animales a través del mismo.
- Permitir a los usuarios eliminar su cuenta y todos los animales asociados al usuario.

- Incluir un sistema de animales favoritos para los usuarios.
- Mejorar el sistema de búsquedas para permitir filtrar por más campos.
- Actualizar Angular, Symfony y cualquier librería y tecnología con forme vayan surgiendo novedades para que el proyecto este siempre actualizado y sea menos vulnerable.

VI BIBLIOGRAFÍA

[1] B. (2021, 8 enero). *Las mejores protectoras de animales*. Tienda Veterinaria Blog.

<https://www.barakaldotiendaveterinaria.es/blog/las-mejores-protectoras-de-animales/>

[2] R. (2019, 1 diciembre). *¿Por qué son imprescindibles las protectoras de animales?* La Vanguardia. <https://www.lavanguardia.com/participacion/las-fotos-de-los-lectores/20191201/471922736302/pendi-dita-historia-gatas-acogidas-refugio-protectoras-animales.html>

[3] *Los beneficios del vínculo entre niños y animales de compañía* / Fundación Affinity. (2022). Fundacion Affinity. <https://www.fundacion-affinity.org/perros-gatos-y-personas/busco-un-animal-de-compania/los-beneficios-del-vinculo-entre-ninos-y>

[4] *El poder de las mascotas: unas grandes aliadas para alcanzar el bienestar.* (2021, 22 abril). Iberdrola. <https://www.iberdrola.com/compromiso-social/beneficios-de-tener-mascotas>

[5] Soft, A. (2020, 2 noviembre). *Planificar en un proyecto de software*. Alboka Soft. <https://www.albokasoftware.com/index.php/blog/80-proyecto-de-software-a-medida>

[6] Velasco, R. (2021, 11 marzo). *¿Quieres crear páginas web? Aprende estos lenguajes de programación*. SoftZone. <https://www.softzone.es/programas/lenguajes/lenguajes-programacion-web/>

[7] Back4App

<https://blog.back4app.com/es/los-10-mejores-marcos-de-frontend-y-backend/>

[8] S. (2022, 3 enero). *Los mejores Frameworks de backend para 2022*. Softtek.

<https://softtek.eu/tech-magazine/software-trends/los-mejores-frameworks-de-backend-para-2022/>

[9] Angular. (2022). Angular. <https://angular.io/>

[10] *symfony.es, el mejor framework PHP para crear aplicaciones web*. (2022). symfony.es.

<https://symfony.es/>

[11] *Curso Angular API REST Angular 10 - Rick and Morty API*. (2020, 9 julio). YouTube.

<https://www.youtube.com/watch?v=hGU2ceM8aVY>

[12] *Curso de Symfony 5. Creando una API desde cero. Capítulo 1. Configuración del proyecto*.

(2020, 2 mayo). YouTube. <https://www.youtube.com/watch?v=cYCCCgrFSi4&t=287s>

[12] *Develoteca*. (2022). DEVELOTECA. <https://cursos.develoteca.com/>

[13] *Lucid visual collaboration suite: Log in*. (2022). Lucid.

<https://lucid.app/users/login?referredProduct=lucidchart>

[14] *Qué es SCRUM*. (2021, 20 septiembre). Proyectos Ágiles. <https://proyectosagiles.org/que-es-scrum/>

[15] Atlassian. (2022). *Scrum: qué es, cómo funciona y por qué es excelente*.

<https://www.atlassian.com/es/agile/scrum>

[16] *Preguntas de «angular» más nuevas.* (2022). Stack Overflow en español.

<https://es.stackoverflow.com/questions/tagged/angular>

[17] *Newest «symfony» Questions.* (2022). Stack Overflow.

<https://stackoverflow.com/questions/tagged/symfony>

[18] *CSS / MDN.* (2021, 7 julio). CSS | MDN. <https://developer.mozilla.org/es/docs/Web/CSS>

[19] *Universe of UI elements made with HTML & CSS.* (2022). Universe. <https://uiverse.io/>

[20] *GitHub: Where the world builds software.* (2022). GitHub. <https://github.com/>