



PROTECTIVE ANIMALS

Alberto Ortega Padilla

Final project DAW

1. User Manual

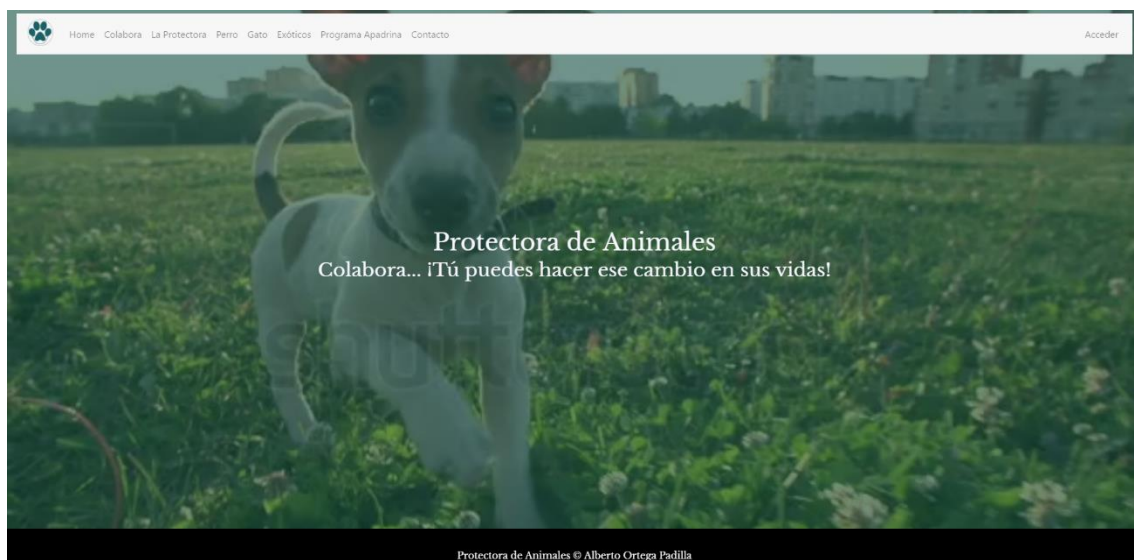
1.1. Description of the application.

The protective animals web application is a platform whose purpose is to facilitate adoption, sponsor, volunteering, donation in an association whose function is an animal shelter.

The application is divided into two parts, a public part, that shows the data of the protective, contact, the selection of animals that have in the shelter (cats, dogs, exotics, sponsors), to be sponsor or adopted, etc. Then the private part, which is accessed from the section of the login, where you Access the administration of the animals that are shown in the public part.

1.2. Features and characteristics.

- Front with background video.



- Collaboration information, history of the protector and contact.

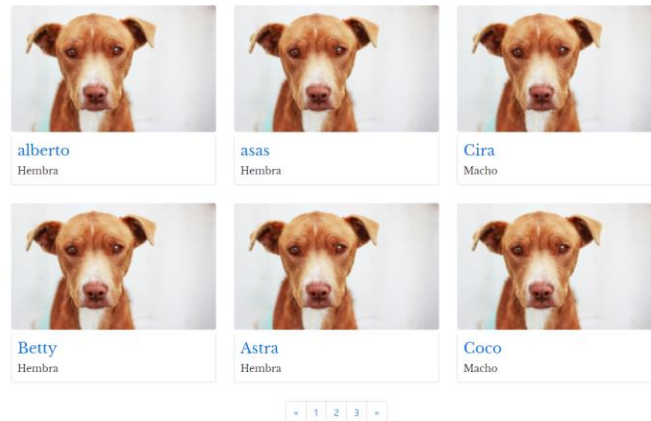


- List with cats pagination, cat detail with Scroll images and hidden video for the page simple have less weight load, with button to show and hide animal video (video connected to a YouTube account).



- List with dogs pagination, cat detail with Scroll images and hidden video for the page simple have less weight load, with button to show and hide animal video (video connected to a YouTube account).

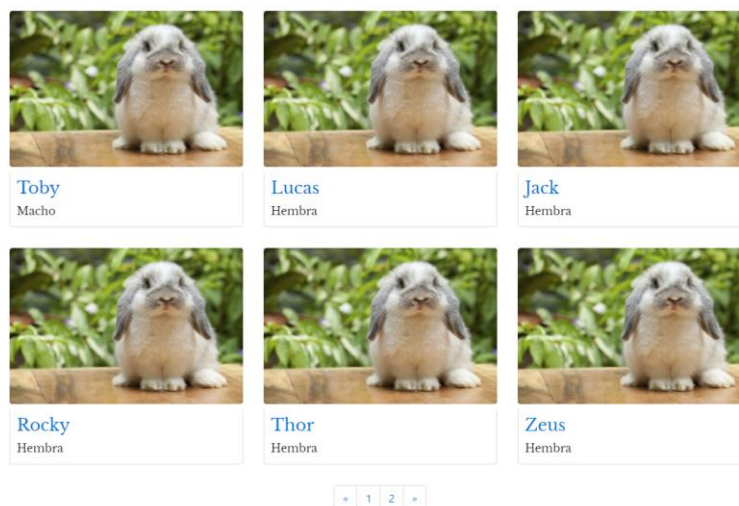
PERROS EN ADOPCIÓN



Protectora de Animales © Alberto Ortega Padilla


- List with cat pagination, exotics detail with Scroll images and hidden video for the page simple have less weight load, with button to show and hide animal video (video connected to a YouTube account).

EXÓTICOS EN ADOPCIÓN



Protectora de Animales © Alberto Ortega Padilla

- Logged in for the administrative part of the animals.

 [Home](#) [Colabora](#) [La Protectora](#) [Perro](#) [Gato](#) [Exóticos](#) [Programa Apadrina](#) [Contacto](#) [Acceder](#)

Login not Valid!!

Login

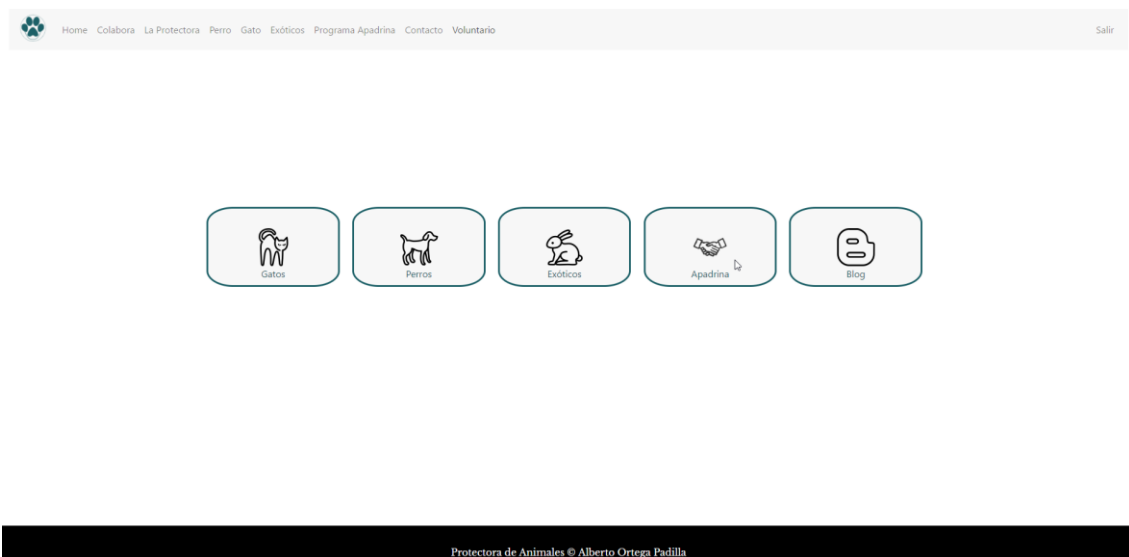
Email:
jane.doe@example.com

Password:
Password


Login

Protectora de Animales © Alberto Ortega Padilla

- Menu of the animal modules (only active the first 3) and emphasize that this private section is responsive until Tablet.



- Table of cats, with options to add, remove, modify and search:
 - ✓ Add is hidden as in the detail video of the animal, with button to add and hide the form with validation of each input.
 - ✓ Animal delete button.
 - ✓ Modify button, which externally opens a form with the details of the database of that cat already entered, forcing the user to check before accepting the changes, if others are correct.


[Home](#)
[Colabora](#)
[La Protectora](#)
[Perro](#)
[Gato](#)
[Exóticos](#)
[Programa Apadrina](#)
[Contacto](#)
[Voluntario](#)
Sair


Listado Gatos

NOMBRE	RAZA	GÉNERO	TAMAÑO	STATUS	EDAD	TEXTO	CONTACTO	TELÉFONO	CORREO	VIDEO	CREADO		
Suli	Mestizo	Hembra	Mediano	Adopción	1	Qué dise u...	Alberto	123456787	alb@aozte...	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Felipe	Europeo	Macho	Mediano	Adoptado	3	Aliqua me...	Luna	578433910	luna@aort...	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Tana	Ruso	Hembra	Mediano	Adopción	3	Caballo bl...	África	878926900	afri@aorte...	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Sheldon	Siamés	Macho	Mediano	Adopción	4	Lorem fist...	Adrián	123445678	adri@aort...	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Isabela	Mestizo	Hembra	Mediano	Adopción	3	Lorem fist...	Luna	578433910	luna@aort...	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Lynx	Siamés	Hembra	Pequeño	Adopción	1	Lorem fist...	Lorena	382910405	lore@aort...	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Protectora de Animales © Alberto Ortega Padilla


- Table of dogs, with options to add, remove, modify and search:

- ✓ Add is hidden as in the detail video of the animal, with button to add and hide the form with validation of each input.
- ✓ Animal delete button.
- ✓ Modify button, which externally opens a form with the details of the database of that cat already entered, forcing the user to check before accepting the changes, if others are correct.


[Home](#)
[Colabora](#)
[La Protectora](#)
[Perro](#)
[Gato](#)
[Exóticos](#)
[Programa Apadrina](#)
[Contacto](#)
[Voluntario](#)
Salir

Listado Perros


NOMBRE	RAZA	GENERO	TAMAÑO	STATUS	EDAD	TEXTO	CONTACTO	TELEFONO	CORREO	VIDEO	CREADO		
alberto	sddldds	Hembra	Grande	Adoptado	2	1zxx	esd	12344567...	es@es.com	<iframe w...	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
asasa	ssasa	Hembra	Grande	Adoptado	1	assasa	asa	123456789	asta@asta...	asss	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Cira	Persa	Macho	Grande	Adopción	1	Gato gord...	Tomm	76585483	tonmm@t.c...	as	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Betty	Persa	Hembra	Pequeño	Adopción	2	Gato gord...	Tomm	765854834	tonmm@t.c...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Astra	Persa	Hembra	Pequeño	Adopción	2	Gato gord...	Tomm	765854834	tonmm@t.c...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Coco	Persa	Macho	Pequeño	Adopción	2	Gato gord...	Tomm	765854834	tonmm@t.c...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>



Protectora de Animales © Alberto Ortega Padilla


- Table of exotics, with options to add, remove, modify and search:

- ✓ Add is hidden as in the detail video of the animal, with button to add and hide the form with validation of each input.
- ✓ Animal delete button.
- ✓ Modify button, which externally opens a form with the details of the database of that cat already entered, forcing the user to check before accepting the changes, if others are correct.


[Home](#)
[Colabora](#)
[La Protectora](#)
[Perro](#)
[Gato](#)
[Exóticos](#)
[Programa Apadrina](#)
[Contacto](#)
[Voluntario](#)
Salir

Listado Exoticos

NOMBRE	CLASE	RAZA	GENERO	TAMANO	ESTADO	EDAD	TEXTO	CONTACTO	TELEFONO	CORREO	VIDEO	CREADO		
Toby	Conejo	Belier	Macho	Mediano	Adopción	2	Gato gor...	Tomm	765854834	tomm@t...	algo	Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Lucas	Hámsteres		Hembra	Pequeño	Adopción	2	Gato gor...	Tomm	765854834	tomm@t...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Jack	Hurones		Hembra	Pequeño	Adopción	2	Gato gor...	Tomm	765854834	tomm@t...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Rocky	Hurones		Hembra	Pequeño	Adopción	2	Gato gor...	Tomm	765854834	tomm@t...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Thor	Hámsteres		Hembra	Pequeño	Adopción	2	Gato gor...	Tomm	765854834	tomm@t...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Zeus	Conejo		Hembra	Pequeño	Adopción	2	Gato gor...	Tomm	765854834	tomm@t...		Alber	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>



Protectora de Animales © Alberto Ortega Padilla

1.3. Minimum requirements.

As it is through the browser, the only requirement is for the browser, but for all interface functionality to work. Another requirement is that for the public part can be displayed on mobile, Tablet, and computer. While the private part is only optimized for computer and Tablet.

2. Technical manual.

2.1. Backend.

2.1.1.1. Data Model.

The database is made in MySQL (relational database management system), managed in Apache, in PhpMyAdmin.

2.1.1.2. Structure and operation.

This environment is made with PHP and the Framework Symfony 3, which is based on the model, view, controller, but since this environment is server only, it has no interface, only use the model and controller section.

The structure of this section is:

- ✓ **App:** Configuration, and translations, of the application.
- ✓ **Bin:** This directory resides the console executable, which serves to call the client utilities from our terminal.
- ✓ **Src:** Is the source code of the application, where it contains the entities, bundles, controllers, etc.
- ✓ **Test:** Stores the test sources, basically a copy src .
- ✓ **Var:** Temporary directory, here is stored log, cache, sesión, etc.
- ✓ **Vendor:** Contains the external libraries that i use, as well as the Symfony core, or basically the directory that the composer uses to maintain dependencies.

- ▶ app
- ▶ bin
- ▼ src
 - ▼ AppBundle
 - ▼ Controller
 - BlogController.php
 - CatController.php
 - DefaultController.php
 - DogController.php
 - ExoticController.php
 - GenderController.php
 - RoleController.php
 - SizeController.php
 - SponsorController.php
 - StatusController.php
 - TypeAnimalController.php
 - TypeBlogController.php
 - TypeExoticController.php
 - UserController.php
 - ▼ Resources
 - ▼ config
 - ▶ routing
 - routing.yml
 - ▼ Services
 - Helpers.php
 - JwtAuth.php
 - AppBundle.php
 - ▼ BackendBundle
 - ▶ Controller
 - ▼ Entity
 - Blog.php
 - Cat.php
 - Dog.php
 - Exotic.php
 - Gender.php
 - ImgAnimal.php
 - ImgBlog.php
 - ImgCat.php
 - ImgDog.php
 - ImgExotic.php
 - Role.php
 - Size.php
 - Sponsor.php
 - Status.php
 - TypeAnimal.php
 - TypeBlog.php
 - TypeExotic.php
 - User.php
 - ▶ Resources
 - ▶ Tests

The controller handles the request, calls and returns the result from the database, to the client, in this case to Angular v4.

```
<?php
namespace AppBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Validator\Constraints as Assert;
use BackendBundle\Entity\Cat;

class CatController extends Controller {

    public function newAction(Request $request) {
        $helpers = $this->get("app.helpers");

        $hash = $request->get("authorization", null);
        $authCheck = $helpers->authCheck($hash);

        if ($authCheck == true) {
            $identity = $helpers->authCheck($hash, true);

            $json = $request->get("json", null);

            if ($json != null) {
                $params = json_decode($json);

                $createdAt = new \DateTime('now');
                $updatedAt = new \DateTime('now');

                $name = (isset($params->name) && preg_match('/^[a-z àéïóñüâëîöñ]+$/i', $params->name)) ? $params->name : null;
                $breed = (isset($params->breed) && preg_match('/^[a-z àéïóñüâëîöñ]+$/i', $params->breed)) ? $params->breed : null;
                $age = (isset($params->age) && ctype_digit($params->age)) ? $params->age : null;
                $description = (isset($params->description)) ? $params->description : null;
                $contact_person = (isset($params->contact_person) && preg_match('/^[a-z àéïóñüâëîöñ]+$/i', $params->contact_person)) ? $params->contact_person : null;
                $phone = (isset($params->phone) && ctype_digit($params->phone)) ? $params->phone : null;
                $email = (isset($params->email)) ? $params->email : null;
                $url_video = (isset($params->url_video)) ? $params->url_video : null;
                $id_gender = (isset($params->id_gender)) ? $params->id_gender : null;
                $id_size = (isset($params->id_size)) ? $params->id_size : null;
                $id_user = ($identity->sub != null) ? $identity->sub : null;
                $id_status = (isset($params->id_status)) ? $params->id_status : null;

                $emailContrain = new Assert\Email();
                $emailContrain->message = "This email is not valid !";
                $validate_email = $this->get("validator")->validate($email, $emailContrain);
            }
        }
    }
}
```

The section of the model is the logic and that interacts with the Database.

```

    * @return Cat
    */
    public function setBreed($breed)
    {
        $this->breed = $breed;

        return $this;
    }

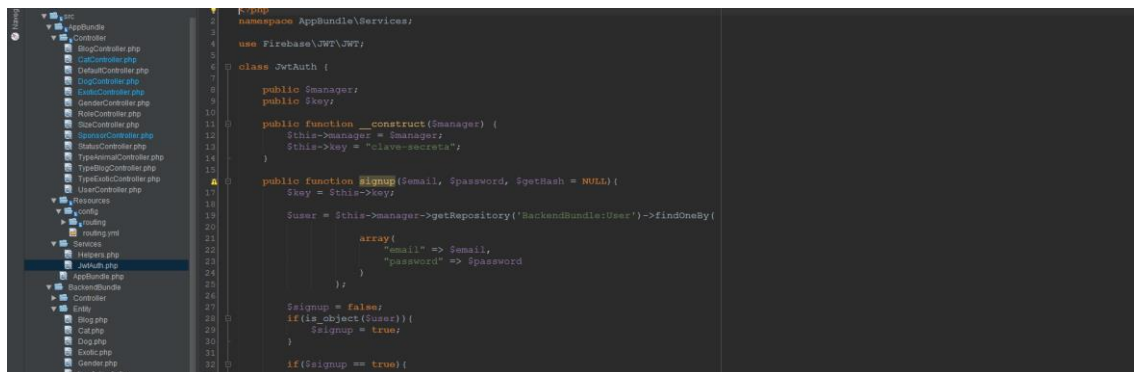
    /**
     * Get breed
     *
     * @return string
     */
    public function getBreed()
    {
        return $this->breed;
    }

    /**
     * Set name
     *
     * @param string $name
     *
     * @return Cat
     */
    public function setName($name)
    {
        $this->name = $name;

        return $this;
    }

    /**
     * Get name
     *
     * @return string
     */
    public function getName()
    {
        return $this->name;
    }
}
```

Services deal with authorization by login, Json, Token check.



2.2. FrontEnd.

2.2.1.1. Structure and operation.

The environment is made with TypeScript and with the Angular Framework v4, which is based on the application view, is the interface of the application, and through the requests by https, calls the Symfony Backend and transmits the data in Json.

Antes de explicar la estructura destacar, que, para poder utilizar este entorno, es necesario antes, instalarte globalmente Node.js y Angular Cli.

La estructura de este apartado es:

- ✓ **Node_module:** This folder contains all the libraries of the package.json that will be used later for the application.
- ✓ **Src:** Contains the source code of the application, divided into folders, and each folder contains a component, which contains the application's functions, the html, containing the views of the component, the scss, which contains the css of this particular component and spec that is the component test.

The start of the application starts in index.html, and then from the selectors, initially app.component.html, and depending on if you use

as it is in my case `app.route`, from the routing, you go from one component to another, forming what would be the application.

The assets folder contains the images, videos, js, etc. Of the application.

- ✓ There would be the pre-configuration of the application, such as `angular-cli.json`, `karma.conf.js`, `package.json`, `protractor.conf.js`, `tsconfig.json`, `tslint.json`.

- node_modules
- src
 - app
 - home
 - login
 - pipes
 - shared
 - voluntary
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - app.route.ts
 - assets
 - img
 - video
 - .gitkeep
 - environments
 - ★ favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.css
 - test.ts
 - {...} tsconfig.app.json
 - {...} tsconfig.spec.json
 - typings.d.ts
 - .angular-cli.json
 - .editorconfig
 - .gitignore
 - karma.conf.js
 - package.json
 - protractor.conf.js
 - README.md
 - {...} tsconfig.json
 - {...} tslint.json

- ✓ **Pipes:** the pipes folder that is inside `/src/app/` are the ones that allow us to alter the way in which the data will be visualized, they are a kind of data filters.
- ✓ **Shared:** Here it is on the same path as the pipes, and contains the models, directives and services of the application.

3. Technologies used.

- **Backend:**

- ✓ **Symfony 3:**

To install Symfony, you need to install a package installer called Composer. By counting the serial dependencies that a blank Project brings, I added new libraries in the dependencies:

- `firebase/php-jwt`: For authentication of Tokens, the translation of jwt is Json Web Token.
 - `knplabs/knp-paginator-bundle`: That uses it for the pagination of the data brought from the Database.

- **FrontEnd:**

- ✓ **Node.js:**

It is a JavaScript execution environment built with the JavaScript V8 for Chrome.

- ✓ **Angular v4:**

As in Symfony you need other components to install an application of Angular, in this case you need node.js and Angular Cli.

Once installed, add new libraries to the dependencies of the package.json:

- Bootstrap: for css in the application.
 - JQuery: to use their bookstore.
 - ✓ **Inkscape:** For the logo of the application transformed into svg and for the favicons, then transformed to the .ico extension from the internet.
 - ✓ **CSS:** the css is divided between the predefined Bootstrap and own, as is the example of the images in scroll, mixed the Bootstrap css for the animation, but then using the css own @media of the media queries. Other cases are like the menú of the animal module in the private part, tables, etc.
- **Networks:**

The application being composed in two parts, are housed in two different hostings, the server is hosted in one that runs with Apache and PhpMyAdmin and the client environment is hosted in another that uses node.js, which is essential to run the interface made by Angular v4.