

# Rock Fish



**IES Luis Vélez de Guevara  
Curso 2021-22**

**Realizado: Alberto Arraz Alé**

**DAW**

**Tutor: José Antonio Muñoz**

# ÍNDICE DE CONTENIDO

<b>Descripción general del proyecto</b>	<b>3</b>
<b>Objetivos generales y objetivos específicos.</b>	<b>3</b>
<b>Análisis previo de ideas según los objetivos del proyecto</b>	<b>3</b>
<b>Justificación de la opción, camino o solución escogida</b>	<b>7</b>
<b>Recursos necesarios</b>	<b>8</b>
<b>Desarrollo del Proyecto</b>	<b>9</b>
<b>Manual básico de utilización para el usuario</b>	<b>16</b>
<b>Dificultades encontradas</b>	<b>20</b>
<b>Conclusiones</b>	<b>20</b>
<b>Posibles Mejoras</b>	<b>21</b>
<b>Fuentes de Información y recursos utilizados</b>	<b>21</b>

# 1. Descripción general del proyecto

La idea es hacer una página web con un catálogo de peces, que puedas iniciar sesión y hacer pedidos personalizados.

El objetivo principal es poder tener en un sitio los peces que dispone la pecera y poder hacer pedidos a la página, y desde ella se puedan ver y administrar los peces y los pedidos.

# 2. Objetivos generales y objetivos específicos.

Desarrollar una aplicación web para la gestión de distintos tipos de peces, que contarán con todo tipo de información sobre su especie o cuidado y hacer pedidos

- Mostrar un catálogo de peces
- Soportar usuarios de tipo cliente y administrador
- Permitir al usuario administrador hacer CRUD de los peces y categorías de la pecera
- Permitir registro de usuarios con autenticación de google
- Permitir al administrador hacer CRUD de los pedidos del cliente
- Ofrecer una zona donde el administrador pueda ver todos los pedidos hechos y que los clientes puedan ver los suyos
- Notificar mediante correo electrónico la confirmación y el envío de su pedido

# 3. Análisis previo de ideas según los objetivos del proyecto



Symfony es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, 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.

Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.

La presentación usa templates y layouts que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.

Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las bases de datos.

El manejo de caché reduce el uso de banda ancha y la carga del servidor.

El soporte de autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.

El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.

Permite la paginación, clasificación y filtrado automático.

Los plugins proveen un alto nivel de extensibilidad.

Dispone de interacción con AJAX.



Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios web y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.



MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,<sup>12</sup> y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

### **phpMyAdmin**

phpMyAdmin es una herramienta gratuita y de código abierto escrita en PHP destinada a manejar la administración de MySQL con el uso de un navegador web. Puede realizar diversas tareas como crear, modificar o eliminar bases de datos, tablas, campos o filas; ejecutar sentencias SQL o gestionar usuarios y permisos. El software, que está disponible en 78 idiomas,<sup>16</sup> es mantenido por The phpMyAdmin Project.<sup>17</sup>

Puede importar datos de CSV y SQL, y transformar los datos almacenados en cualquier formato utilizando un conjunto de funciones predefinidas, como la visualización de datos BLOB como imágenes o enlaces de descarga.



Heroku es una plataforma como servicio (PaaS) de computación en la Nube que soporta distintos lenguajes de programación.

- Elasticidad y crecimiento. La cantidad de Dynos asignados a una aplicación se puede cambiar en cualquier momento a través de la línea de comandos o el dashboard.
- Tamaño. Heroku ofrece diferentes tipos de dynos, cada uno con diferentes capacidades de procesamiento y memoria.
- Routing. Internamente los routers realizan un seguimiento de la ubicación de los Dynos que estén corriendo, y redirigen el tráfico de acuerdo a la misma.
- Seguimiento. Existe un manejador de Dynos, el cual monitorea de forma continua los dynos que se estén ejecutando. En caso de una falla en un Dyno, este es eliminado y creado nuevamente.
- Distribución y redundancia. Los Dynos se encuentran aislados uno del otro. Esto implica que de existir fallos en la infraestructura interna de alguno de ellos, los otros dynos no se ven afectados, y consecuentemente tampoco la aplicación.

# GitHub

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera

GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena generalmente de forma pública.



JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,<sup>2</sup> basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

## **4. Justificación de la opción, camino o solución escogida**

Symfony: Para empezar he pensado en usar symfony como tecnología base porque creo que en el curso lo que más completo y que más me va a aportar a la hora de crear mi aplicación va a ser un framework con el que ya he trabajado en clase.

Bootstrap 5: es lo que necesitare para el apartado visual, todo que tenga que ver con cómo se vea la página será trabajado con esta tecnología.

MySQL: Es un sistema de gestión de base de datos que trabaja bien con Symfony y el que más facilidades te da para trabajar con él.

Heroku: Es la plataforma que voy a usar para desplegar el proyecto, no voy a necesitar más pues symfony es un MVC y va todo el despliegue junto.

GitHub: Lo usare para el control de versiones, es la plataforma que más control en este caso, nada más.

JavaScript: Sobre todo para temas visuales, posibles animaciones y puede que para gestión de datos, pero es tan poca cosa que prefiero usar js vanilla que un framework.

## 5. Recursos necesarios

- Hardware

Ordenador sobremesa

- Ryzen 5 3600
- 16 GB RAM
- B450
- SO: Ubuntu

Ratón: Logitech g206

Teclado: Razer huntsman mini

Monitor 1: 1080p 75Hz

Monitor 2: 900p 60Hz

- Software

- Google Chrome
- Visual Studio Code

- Servicios online (repositorios, servidores de despliegue...)

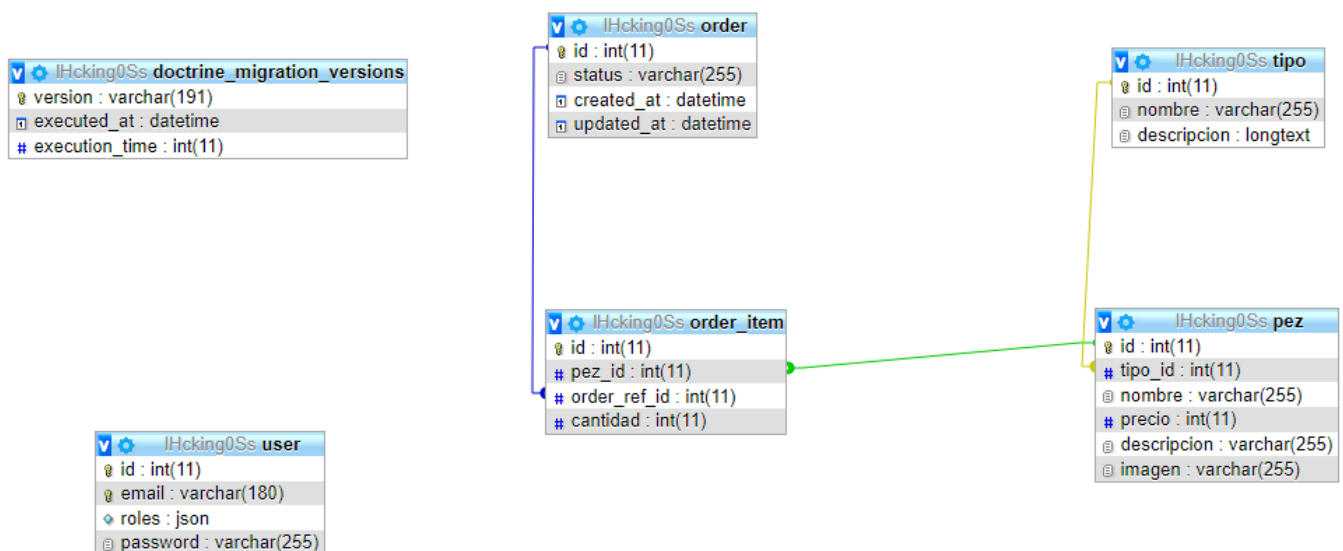
- Heroku
- GitHub



# 6. Desarrollo del Proyecto

## Modelado de la información

Cómo estamos utilizando el mapeador objeto-relacional (ORM) Doctrine, las entidades son objetos PHP que contienen variables que son las propiedades que se guardan y devuelven a una base de datos.



Las entidades necesarias para el desarrollo de la aplicación son las siguientes:

### User:

- Esta entidad es para tener un usuario administrador que es el que puede hacer crud
- Propiedades:
  - id: identificador
  - email: email para iniciar sesión
  - roles: los roles de los usuarios, en este caso admin
  - password: contraseña

## **Tipo:**

- **Los distintos tipos de peces que hay en la tienda**
- **Propiedades:**
  - id: identificador
  - nombre: el nombre de la especie/tipo de pez
  - descripcion: descripcion rapida de como es, lo que come etc

## **Pez:**

- **Los peces que existen en la tienda**
- **Propiedades:**
  - id: identificador
  - tipo\_id: el identificador del tipo al que pertenece
  - nombre: nombre científico del pez
  - precio: lo que cuesta el pez
  - descripción: descripción del pez ya sea tamaño, edad, etc.
  - imagen: aqui se guarda la referencia a la imagen que tiene el pez

## **Order:**

- **Son los pedidos, ya están en trámite(cart) o pedidos**
- **Propiedades:**
  - id: identificador
  - status: estado del pedido (cart por defecto)
  - created\_at: fecha cuando se crea el carrito
  - updated\_at: fecha cuando se edita/guarda el carrito

## **Order\_item:**

- **Son los peces que hay dentro de cada pedido**
- **Propiedades:**
  - id: identificador
  - pez\_id: pez que está en el pedido
  - order\_ref\_id: pedido/order al que pertenece el pez del pedido

## Programación

### Guía de estilo de codificación

Para la codificación he seguido la recomendación PHP Standards Recommendations - 2 <https://www.php-fig.org/psr/psr-2/>.

Aquí puedes encontrar todas las normas seguidas y que debe mantener el código.

Algunos ejemplos que cumplimos en nuestra codificación, son los siguientes:

- Los nombre de clase deben cumplir la nomenclatura Upper Camel Case, es decir, la primera letra de cada palabra en mayúscula.
- Para el nombre de las funciones se ha utilizado la nomenclatura Lower Camel Case, a diferencia de la anterior, la primera letra es minúscula.
- Las variables van a utilizar la nomenclatura Lower Camel Case y en caso de haber alguna constante, utilizará la nomenclatura Upper Case (todas las letras del identificador se encontrarán en mayúscula).
- Los espacios de nombre se declarará mediante la nomenclatura Upper Camel Case (la primera letra de cada palabra en mayúscula).

## Organización física del código (ficheros)

```
├── bin/ |
    │   └── console
├── config/
    │   ├── packages/
    │   ├── routes/
    │   └── routes.yaml
├── public/
├── src/
    │   ├── Command/
    │   ├── Controller/
    │   ├── DataFixtures/
    │   ├── Entity/
    │   ├── Factory/
    │   ├── Form/
    │   │   └── EventListener/
    │   ├── Manager/
    │   ├── Repository/
    │   ├── Security/
    │   └── Storage/
└── templates/
```

### El directorio bin/

En la directorio bin se encontrarán los archivos ejecutables en binario, por ejemplo:

bin/console.

### El directorio config/

En este directorio se guarda configuración de la aplicación, plantillas, paquetes

instalados y traducción. Es aquí donde se encuentra unos de los archivos más importante de la

aplicación:

- **routes.yaml**. Archivo que contiene las rotas de redireccionamiento, en mi caso cada una está definida en su controller

## El directorio public/

Es el **directorio raíz** de la página web. Todo lo de esta carpeta es accesible al usuario;

imágenes, documentos o archivos de estilos necesarios para la renderización del código, así

como los paquetes de Bootstrap. Se organiza en carpetas:

- /img. Contiene fotos para presentar la web.
- /img\_pez. Contiene las fotos de cada pez.
- /css. Contiene estilos adicionales.
- /js. Contiene archivos javascript adicionales.

## El directorio src/

Si indagamos en este directorio, encontraremos todo el código fuente (source code) PHP. Es aquí donde se almacenan los controladores de la aplicación, las entidades, formularios o los DataFixtures organizados de la siguiente forma:

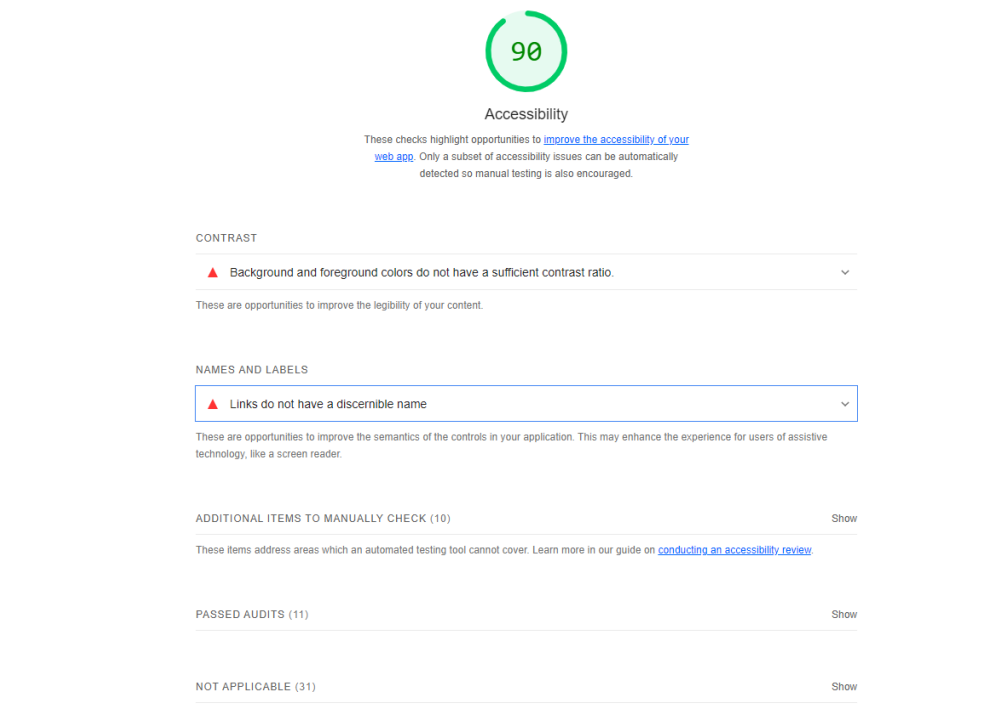
- /Command. Contiene comandos personalizados de symfony
- /Controller. Contiene los controladores de la aplicación.
- /DataFixtures. Se almacenan los archivos que permiten crear datos iniciales.
- /Entity. Guarda las entidades de la aplicaciones.
- /Form. Aquí se almacena la estructura de los formularios de la aplicación.
- /Repository. Incluye las clases de la aplicación.

## El directorio templates/

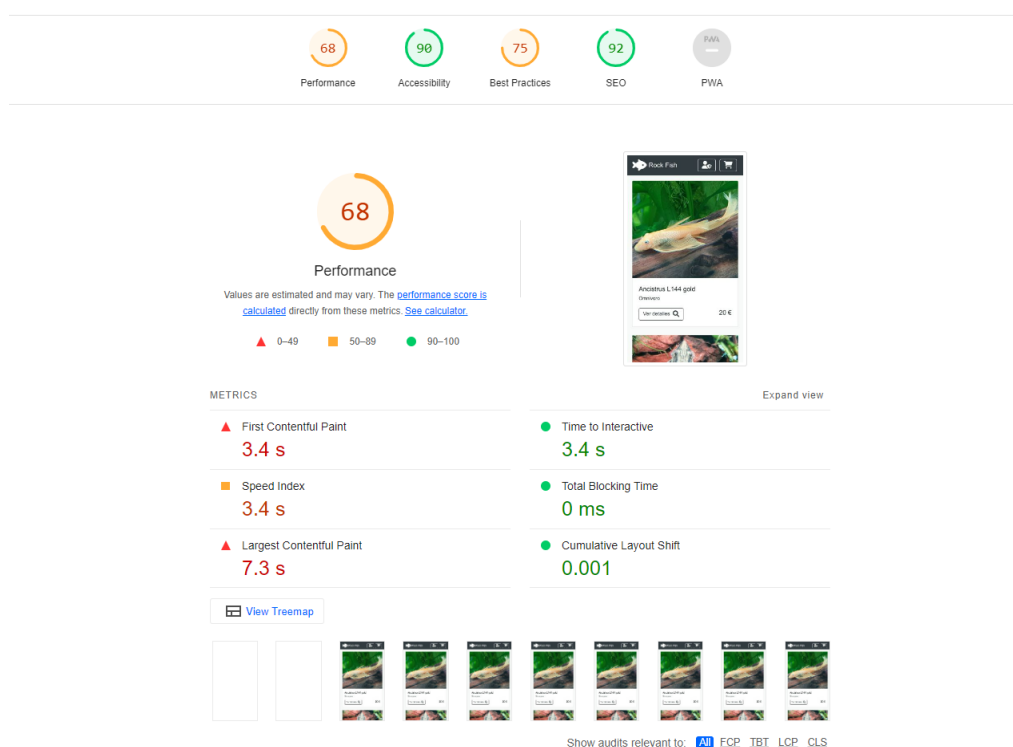
Por último, nos encontramos el directorio templates/, donde se almacenarán las plantillas twig organizadas por carpetas.

## Accesibilidad

La aplicación da una puntuación de 90/100 en accesibilidad lo que quiere decir que es muy accesible.



El único problema que tenemos es la Performance que es muy difícil o casi imposible de subir, ya que tenemos desplegado en un subdominio gratuito y no es muy rápido todo lo que podemos hacer es poner las imagenes en webp y no tener nada más que lo necesario, todos los css y js que traigamos a la app han de ser min.



## Seguridad

Para la adecuada seguridad de la aplicación web se han creado diferentes tipos de usuarios para restringir a algunos de estos a las determinadas páginas.

Tipos de usuarios de la aplicación:

- Usuario Administrador. Los administradores tendrán el control total sobre la aplicación. Estos pueden crear, editar y eliminar peces.

Usuario Anónimo. A este grupo pertenece cualquier usuario tendrá acceso a ver todos los peces, ver los detalles de estos, añadirlos a su carrito, borrar peces de sus pedidos, limpiar el carrito o realizar el pedido

## Proceso de despliegue

Para realizar el despliegue de la aplicación se ha realizado a través de Heroku, siguiendo los siguientes pasos:

1. Crear cuenta en Heroku. Elegir el plan Free, que aunque está algo limitado es gratis.
2. Iniciar el repositorio en Git.
3. Crear la aplicación en Heroku con el nombre deseado.
4. Crear el archivo 'Procfile' a través del comando echo 'web:

```
heroku-php-apache2
```

```
public/' > Procfile
```

Este archivo le dice a Heroku que comando usar para iniciar el servidor con la configuración correcta.

5. Cambiamos la variable de entorno APP\_ENV de dev a prod, es decir, cambiamos el proyecto al modo de producción.

```
heroku config:set APP_ENV=prod
```

6. Hacemos commit de todo el repositorio y subimos los cambios a Heroku:

```
git push
```


```
heroku master
```



7. Abrimos la aplicación: heroku open


8. Al principio nos dará error, ya que la pantalla de inicio necesita una conexión a la base de datos y aún no la tenemos configurada, para ello no dirigimos al apartado Config Vars de Heroku y agregamos la variable de entorno DATABASE\_URL con los parámetros de conexión a la BD.

# 7. Manual básico de utilización para el usuario


Los usuarios **anónimos** tienen acceso completo a la tienda salvo por la posibilidad de crear, editar y borrar peces, tanto ver como añadir al carrito y guardar sus carritos es posible, aunque no se pueden volver a obtener si pierden la sesión pero se quedan guardados en la base de datos.

 Rock Fish







Ancistrus L144 gold  
Omnivoro

Ver detalles 


20 €




Sturisoma panamense  
Herbívoro

Ver detalles 

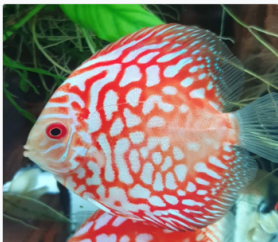
30 €




Escalar manacapuru  
Carnívoro

Ver detalles 

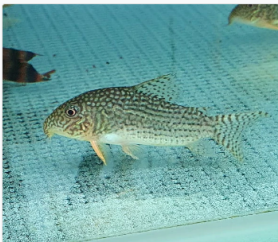
200 €




Symphysodon Pigeon Blood  
Carnívoro

Ver detalles 

300 €






Corydora sterbai  
Omnívoro


Ver detalles 

15 €

## RockFish

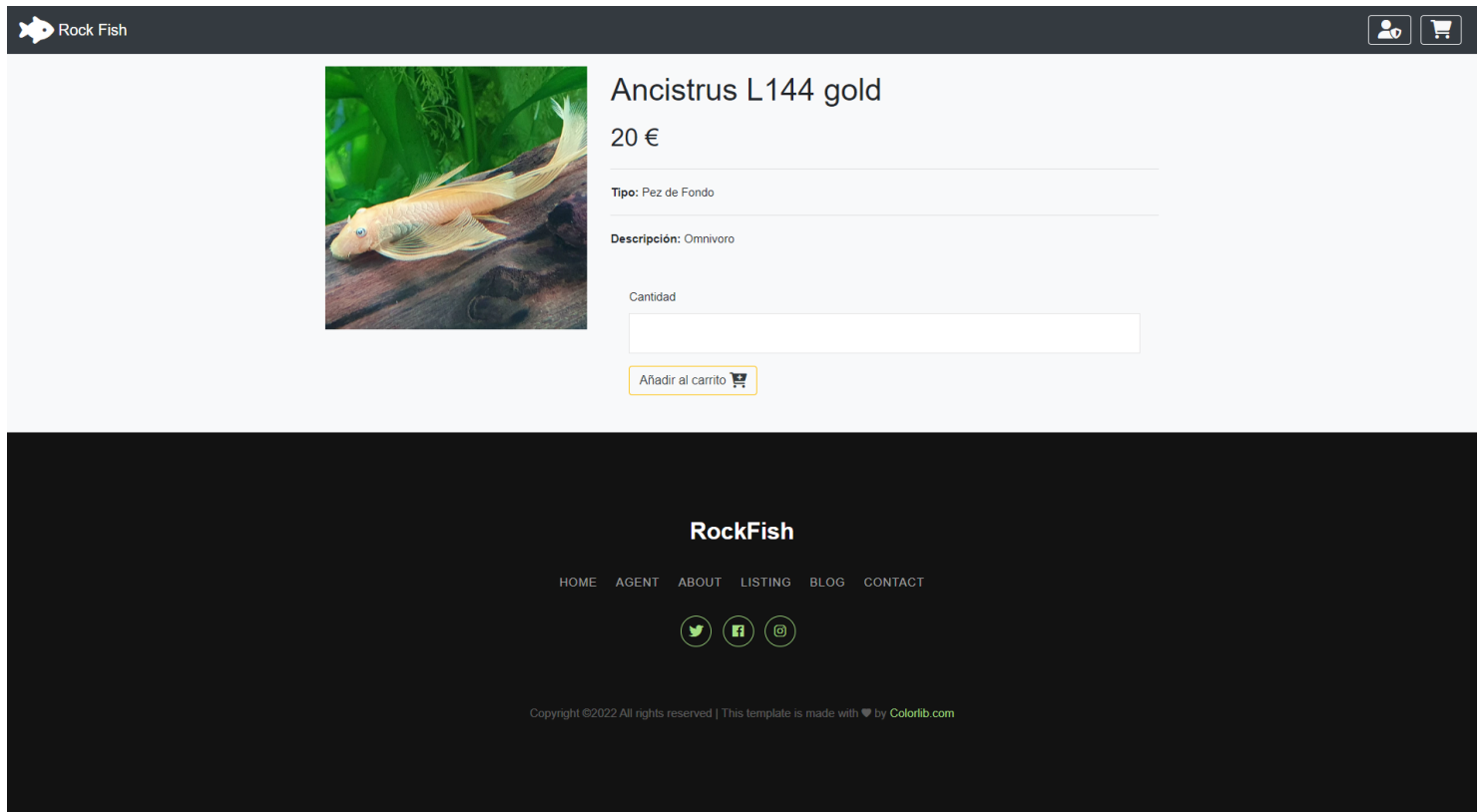
HOME AGENT ABOUT LISTING BLOG CONTACT



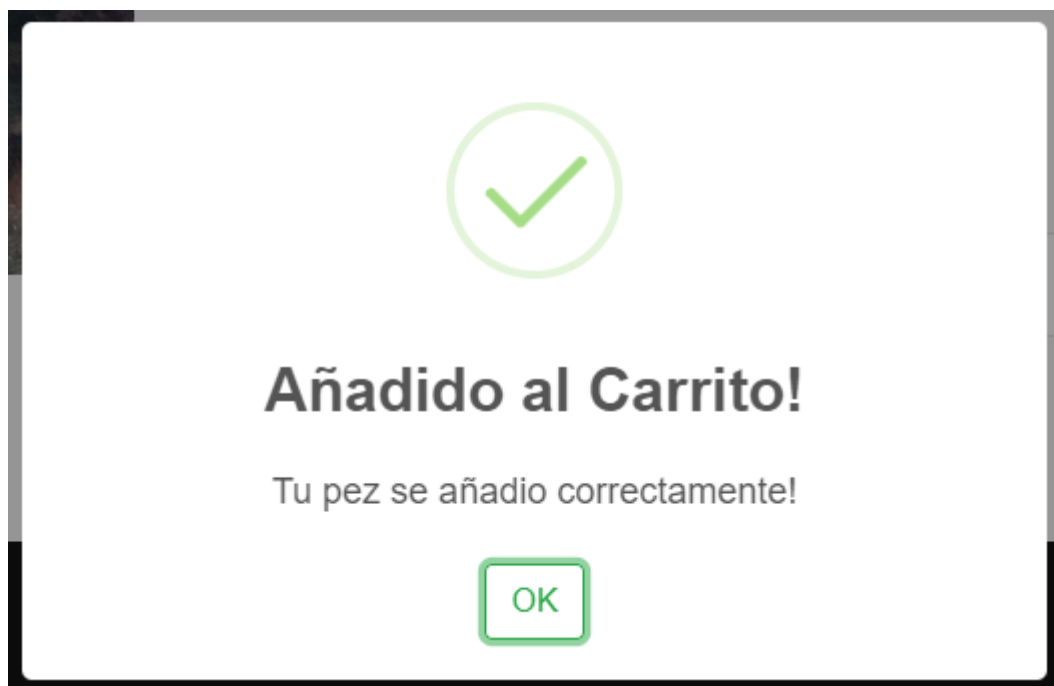
Copyright ©2022 All rights reserved | This template is made with  by [Colorlib.com](https://colorlib.com)



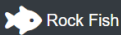
En cada página de detalle del pez podrán añadir al carrito la cantidad que quieran de peces. En cualquier momento podemos volver a la página de inicio pulsando en Rockfish





Al introducir una cantidad correcta(1 o más) nos saldrá la siguiente alerta y ya tendremos nuestro pez/peces añadidos al carrito, repítelo cuanto quieras!




Ahora en nuestro carrito tenemos los distintos peces que hemos añadido, ahora podemos cambiar la cantidad que queremos de cada uno y guardar el carrito, vaciar el carrito o borrar uno que no queremos o hacer el pedido.





## Tu Carrito



Ancistrus L144 gold

Omnivoro...

Borrar

Symphysodon Pigeon Blood

Carnivoro...

Borrar

Guardar

Limpiar




Resumen

Total940 €

Checkout


RockFish



HOMEAGENTABOUTLISTINGBLOGCONTACT



Copyright ©2022 All rights reserved | This template is made with ♥ by [Colorlib.com](#)

Ahora es hora del usuario Admin, para ello lo primero es iniciar sesión pulsando en el icono del usuario con escudo.





Please sign in




Email

Contraseña

Iniciar sesión

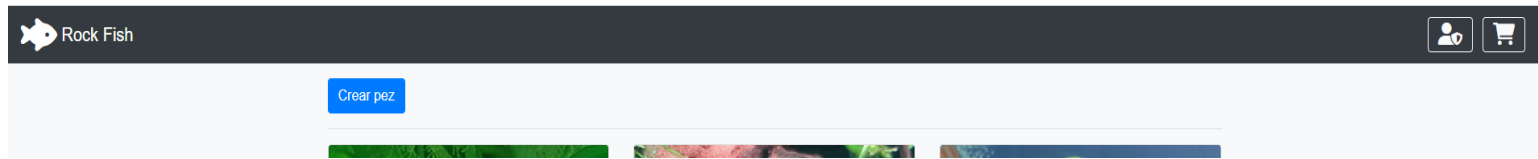
RockFish

HOMEAGENTABOUTLISTINGBLOGCONTACT

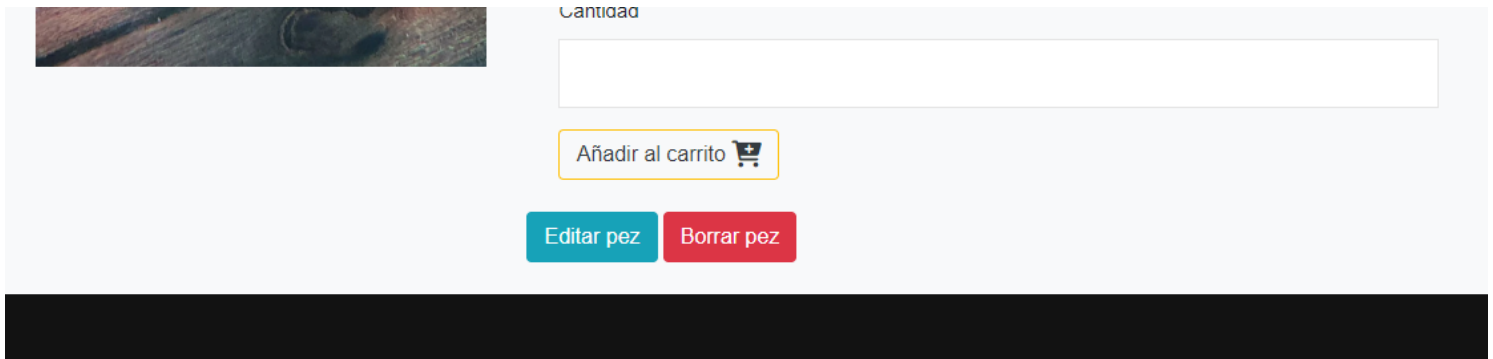


Copyright ©2022 All rights reserved | This template is made with ♥ by [Colorlib.com](#)

Ahora en la página de inicio nos encontramos este botón para crear peces



Y dentro de cada página de detalle de los peces tenemos para editar y borrarlo



Si entras en cualquiera de las rutas de administrador sin iniciar sesión te pedirá que la inicies sesión.

## 8. Dificultades encontradas

- Las sesiones del carrito han sido más difíciles de lo que pensaba hacer.
- Al editar los peces, tienes que poner de nuevo la foto si o si, no encontré manera de que coja la que tenía de antes.

## 9. Conclusiones

Después de acabar el proyecto me he dado cuenta como hay cosas que las tenemos en todos lados como el carrito y pueden ser muy complicadas de desarrollar, sin embargo hay cosas que parecen difíciles y son por el contrario algo rápido que es muy fácil implementar en tu proyecto.

Al principio pensaba en hacer el proyecto de otra manera y con otras entidades que he tenido que cambiar en mitad del camino.

## 10. Posibles Mejoras

- Pagina para ver los pedidos y gestionarlos
- Poder iniciar sesion como usuario y guardar tus carritos
- Ahora mismo si le das a checkout solo te vacía el carrito y te dice que se ha realizado correctamente pero no se manda realmente ni se hace nada, seria cambiar el estado del pedido a send.
- Poder hacer que el pedido le llegue al correo de la tienda

## 11. Fuentes de Información y recursos utilizados

Footer

<https://colorlib.com/wp/template/bootstrap-footer-17/>

PSR-1: Basic Coding Standard:

<https://www.php-fig.org/psr/psr-1/>

PSR-2: Coding Style Guide:

<https://www.php-fig.org/psr/psr-2/>

Sweetalert2:

<https://sweetalert2.github.io/>

Fontawesome

<https://fontawesome.com/>