



Universidad  
Pontificia  
de Salamanca

Arquitectura del software  
Grado en Ingeniería Informática

## Link de repositorio Github

<https://github.com/Gonzarrion/PracticaFinalSymfony>

## Rutas

Aunque hemos incorporado navegación en el proyecto. Se pueden utilizar las siguientes rutas para acceder directamente una funcionalidad/método concreto.

### **/inicio**

Ruta inicial de la aplicación. se te redirecciona directamente desde 127.0.0.1:8000(ruta por defecto).

### **/funcionalidades**

Menú de funcionalidades disponibles.

### **/búsqueda**

Lugar en el que se encuentra el menú de los métodos de búsqueda.

### **/listar/bibliotecas**

Todas las bibliotecas existentes. desde aquí se puede utilizar **/borrar/biblioteca/{id}** para borrar una biblioteca concreta.

### **/buscar/biblioteca/ciudad**

Buscar biblioteca introduciendo una ciudad.

### **/buscar/biblioteca/nombre**

Buscar biblioteca introduciendo un nombre.

### **/insertar/biblioteca**

Añade una biblioteca nueva a la base de datos, los datos introducidos deben ser válidos.

### **/editar/biblioteca/{id}**

Editar campos de una biblioteca existente.

**/buscar/libro/autor/biblioteca**

Buscar libros de un autor concreto en una biblioteca.

**/buscar/libro/editorial/biblioteca**

Buscar libros de una editorial concreta en una biblioteca .

**/buscar/libro/titulo/biblioteca**

Buscar libros con un determinado título en una biblioteca.

**/buscar/libro/titulo**

Buscar libros introduciendo su título.

**/buscar/libro/titulo/disponibilidad**

Buscar libros y su disponibilidad (número de ejemplares totales de una biblioteca para los libros con el mismo nombre).

**/listar/libros/biblioteca**

Buscar todos los libros de una biblioteca, desde aquí se puede utilizar **/borrar/libro/{id}** para eliminar un libro de una biblioteca.

**/editar/libro/{id}**

Editar los campos de un libro existente

**/añadir/nuevo/libro**

Añade un nuevo libro a la base de datos, los datos introducidos deben ser válidos.

## Análisis de Requisitos

### Requisitos funcionales

#### Gestión de bibliotecas

##### 1. Búsquedas

- Consultar lista de bibliotecas.
- Buscar biblioteca por nombre.
- Buscar biblioteca por ciudad.

## **2. Interacciones con la Base de Datos**

- Añadir nueva biblioteca.
- Editar biblioteca existente.
- Eliminar biblioteca existente.

## **Gestión de libros**

### **1. Búsquedas**

- Consultar lista de libros de una biblioteca.
- Buscar libro por título en una biblioteca.
- Buscar libro por autor en una biblioteca.
- Buscar libro por editorial en una biblioteca.
- Buscar libro por título.
- Buscar libro por título y disponibilidad, además hemos añadido la posibilidad de ver los ejemplares totales de un libro en una biblioteca.

## **2. Interacciones con la Base de Datos**

- Añadir nuevo libro.
- Editar libro existente.
- Eliminar libro existente.

## **Requisitos no funcionales**

### **1. Usabilidad**

El sistema está diseñado de manera que es fácil de usar y navegar por él sin necesidad de escribir las rutas, aunque también se puede acceder a las funcionalidades a través de ellas.

# Diagrama del Modelo de Base de Datos

## Entidades

### Biblioteca:

- Id (int, Primary Key)
- Nombre (varchar(255))
- Dirección (varchar(255))
- Ciudad (varchar(255))
- Horario\_apertura (time)
- Horario\_cierre (time)
- Fecha\_fundacion (date)

### Libro:

- Id (int, Primary Key)
- Biblioteca\_id (int, Foreign Key)
- Título (varchar(255))
- Autor (varchar(255))
- Sinopsis (text)
- Año\_publicacion (date)
- Editorial (varchar(255))
- ISBN (varchar(255))
- Numero\_ejemplares (int)

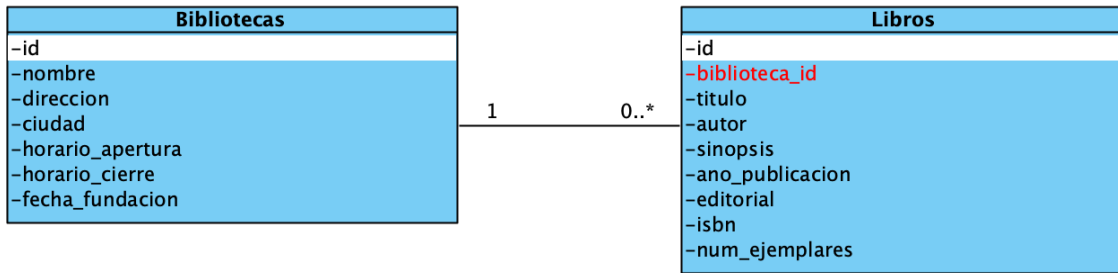
## Relaciones

1. Una biblioteca puede tener muchos libros.
2. Un libro pertenece a una única biblioteca.

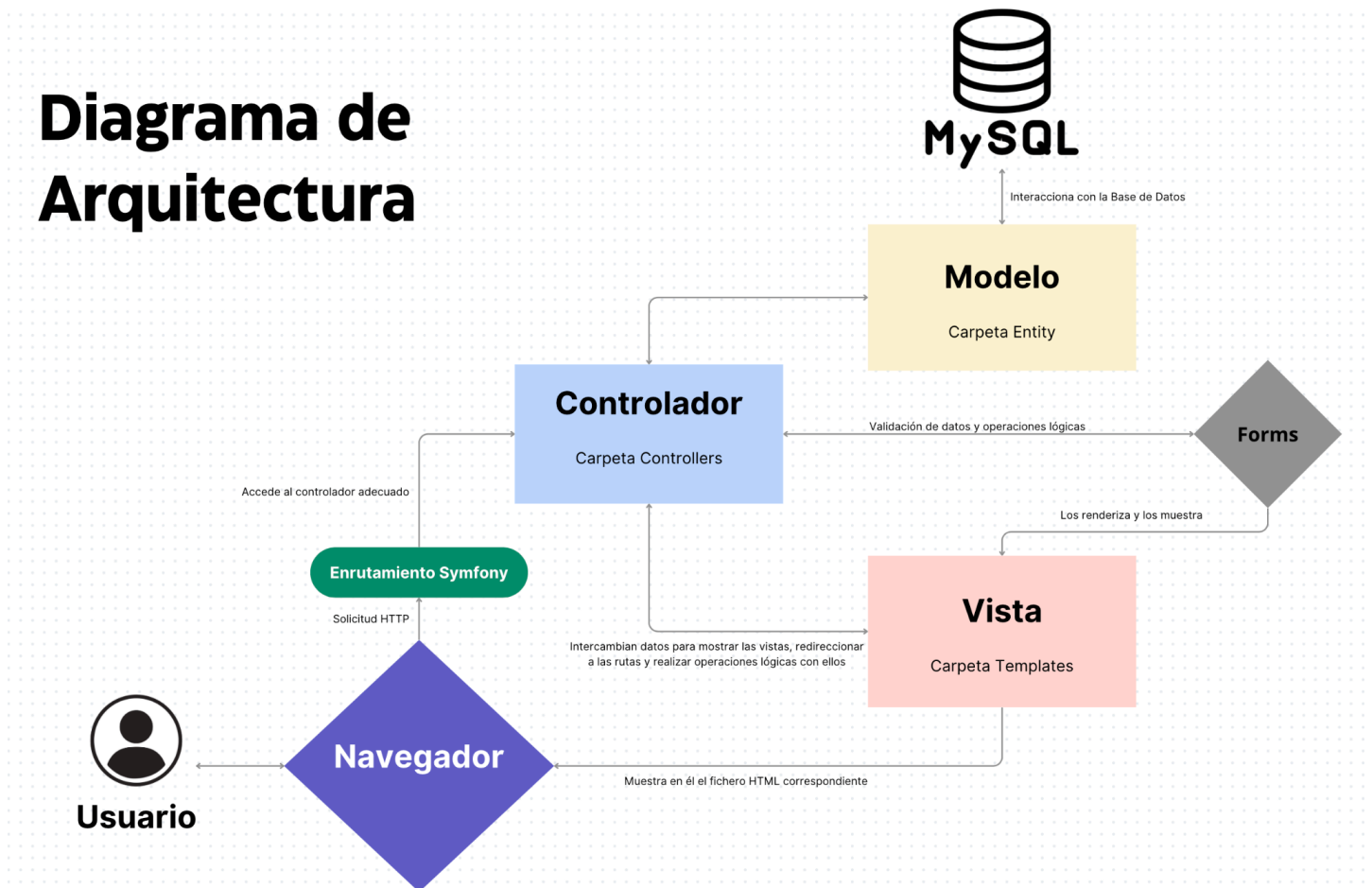
## Restricciones

- El campo **ISBN** debe tener un formato válido. La restricción nos pide que contenga 4 grupos de dígitos separados por un guión (-), por ejemplo: 12-1234-1-23.
- El campo **Dirección** debe contener un número al final.
- El campo **id** de cada tabla es su clave primaria.
- El campo **biblioteca\_id** de la tabla **libro** es clave foránea y representa la biblioteca a la que pertenece.

## Diagrama de la Base de Datos



## Diagrama de Arquitectura



Para explicar la arquitectura del proyecto, lo haremos con un ejemplo del flujo de trabajo.

1. **Solicitud:** El usuario envía una solicitud HTTP (GET, POST, etc.) a la aplicación a través del navegador.

2. **Enrutamiento:** Symfony utiliza su sistema de enrutamiento para dirigir la solicitud al controlador adecuado.
3. **Controlador:** El controlador procesa la solicitud, interactúa con las entidades para realizar operaciones con los datos y devuelve una respuesta.
4. **Modelo:** Interactúa con la base de datos para realizar las operaciones CRUD indicadas por el controlador.
5. **Formulario:** Los formularios se utilizan para validar y procesar los datos de entrada del usuario.
6. **Plantilla:** Finalmente, la respuesta se renderiza utilizando una plantilla Twig y se envía de vuelta al navegador del usuario.

Esta arquitectura sigue el patrón MVC (Modelo-Vista-Controlador), en la cual se separan claramente la lógica de negocio, la presentación y el manejo de solicitudes.