ACTIVIDAD #2

Ingeniería de Software Web – 03MASW

Equipo 3 - 20/11/2021

Irene Gómez González  
Alberto Mateo Martínez   
María Cortés Menéndez

Índice de contenido

[1. Introducción 2](#_Toc88249954)

[2. Funcionalidades a Requisitos 2](#_Toc88249955)

[3. Desarrollo conceptual 5](#_Toc88249956)

[3.1 Análisis de los requisitos funcionales 6](#_Toc88249957)

[3.2 Análisis de los requisitos de contenido 10](#_Toc88249958)

[3.3 Análisis de los requisitos de navegación 11](#_Toc88249959)

[3.4. Análisis de los requisitos de presentación 17](#_Toc88249960)

[4. Conclusiones 19](#_Toc88249961)

[5. Bibliografía 19](#_Toc88249962)

# 1. Introducción

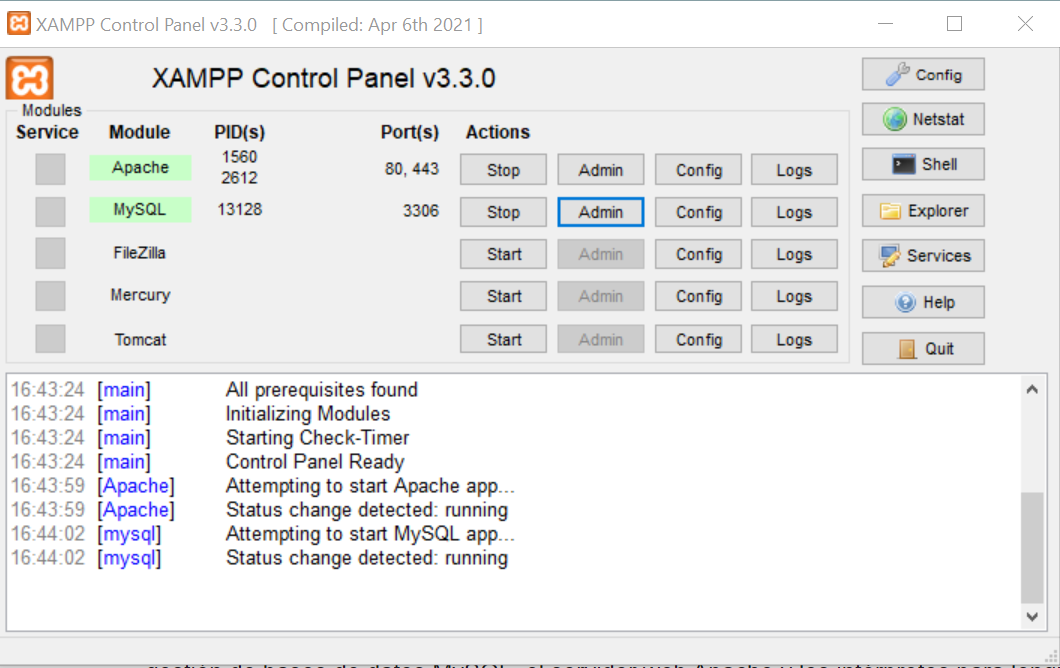
El objetivo de esta actividad es la programación de un sitio web con el lenguaje de programación PHP del lado del servidor.

El sitito web solicitado es una biblioteca de Series con los datos de Plataforma, Idioma, Directores, Actores, etc.

Se documenta con un informe breve de Requisitos, Diseño conceptual y se desarrolla con más detalle la implementación que esta más en la línea de la asignatura

Como servidor local de Apache se puede utilizar XAMP, MAMP o cualquier otro. Sobre dicho servidor se da la opción de usar como base de datos PostgreeSQL, MySQL o MariaDB.

Por la facilidad de uso, la integración con Visual Estudio Code, y sobre todo por el conocimiento del conjunto de herramientas se elige realizar la practica con **MySQL** sobre el paquete o pila de aplicaciones de código abierto **XAMP. Versión 3.3.0** (Compilación 6 de abril de 2021)



Como se explicará con más detalle más adelante se utilizará el patrón de MVC, Modelo Vista Controlador

# 2. Requisitos Funcionales

La actividad establece unos requisitos generales. Se trata de realizar una aplicación Web para almacenar una biblioteca de series. El sistema realizado ha de almacenar las series de manera que sean recuperables. Se fijan otros condicionantes para asegurar la integridad de los datos como el control de duplicados

Las funcionalidades posibles son muy numerosas y todas muy interesantes. Se podrían crear más tablas, vistas, sistemas de localización etc. Se ha optado por una simplificación del sistema al objeto de poder experimentar más características y no dedicar el escaso tiempo disponible a repetir una y otra vez la misma estructura. Por ello se realizan las siguientes simplificaciones:

* Una serie solo puede estar en una plataforma
* Una serie solo puede tener un idioma
* Una serie solo puede tener un director.

En cambio, un actor puede participar “Cast” en varias series y una Serie tiene varios Actores. Por ello, la relación “n a m”, muchos a muchos, se transforma en una tabla para dar soporte a las relaciones entre Actores y Series.

Las funcionalidades definitivas son:

1. Creación y listado de **Plataformas**. Desde el listado se pueden borrar y editar las plataformas existentes.
2. Creación y listado de **Idiomas**. Desde el listado se pueden borrar y editar los idiomas existentes.
3. Creación y listado de D**irectores**. Desde el listado se pueden borrar y editar los directores existentes.
4. Creación y listado de **Actores**. Desde el listado se pueden borrar y editar los Actores existentes.
5. Creación y listado de **Series**. Desde el listado se pueden borrar y editar los directores existentes.
6. Creación y listado de **Relaciones entre Actores y Series**. Desde el listado se pueden borrar y editar los directores existentes.
7. Consulta de las series de un Actor desde el listado de Actores.
8. Consulta de los Actores de una Serie desde el listado de Series.

Así mismo se establecen como requisitos no funcionales:

* Las fechas se muestran en formato español.
* Se ha de evitar que la creación de registros duplicados
* Se ha de evitar que se borren registros de entidades con claves foráneas.

# 3. Desarrollo conceptual

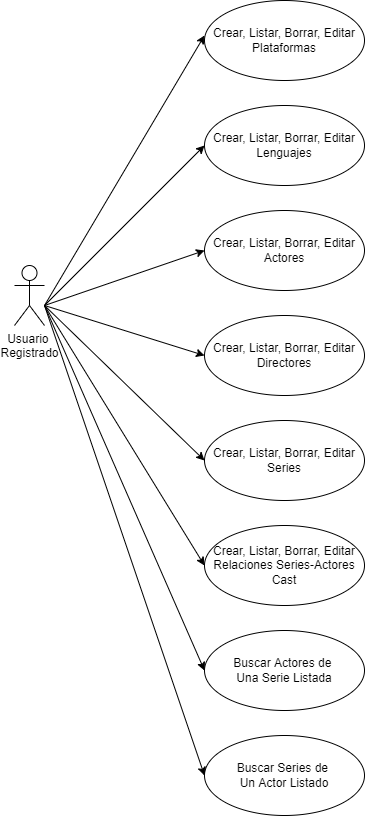


Figura 4. Diagrama de casos de uso .

**CASO DE USO**

1. **Mantenimiento de Plataformas.** El usuario puede crear nuevas plataformas, listar las existentes. Sobre el listado de las existente puede editar o eliminar
2. **Mantenimiento de Lenguajes.** El usuario puede crear nuevos lenguajes, listar los existentes. Sobre el listado de los existente puede editar o eliminar.
3. **Mantenimiento de Actores.** El usuario puede crear nuevos Actores, listar los existentes. Sobre el listado de los existentes puede editar o eliminar.
4. **Mantenimiento de Directores.** El usuario puede crear nuevos Directores, listar los existentes. Sobre el listado de los existentes puede editar o eliminar.
5. **Mantenimiento de Series.** El usuario puede crear nuevas Series, listar las existentes. Sobre el listado de las existente puede editar o eliminar
6. **Mantenimiento de Relaciones Series-Actor.** El usuario puede crear nuevas Relaciones, listar las existentes. Sobre el listado de las existente puede editar o eliminar
7. A partir del listado de series, el usuario puede seleccionar una serie y ver el detalle de todos los actores que participan en dicha serie.
8. A partir del listado de Actores, el usuario puede seleccionar un actor y ver el detalle de todas las series en las que participa.

Se elabora también un **diagrama Entidad-Relación** con el fin de organizar y representar los datos de contenido relativos a cada una de las entidades principales y las relaciones existentes entre ellos:

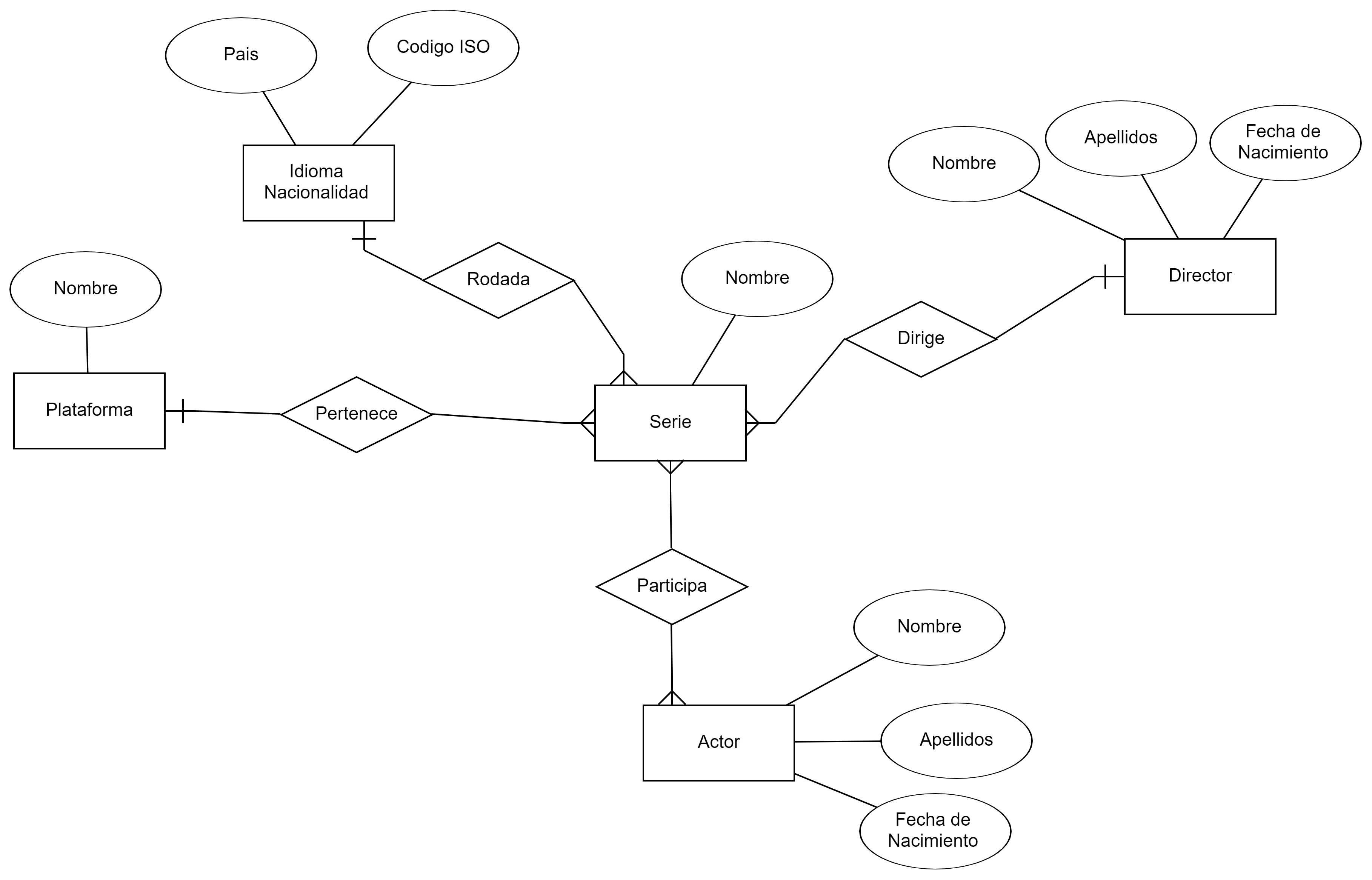


Figura 6. Diagrama Entidad – Relación.

## 3.3 Análisis de los requisitos de navegación

Para analizar los requisitos de navegación se realizan dos **Diagramas de Secuencia UML** para modelar los distintos recorridos que los usuarios realizan en su interacción con la aplicación. Se elige este diagrama porque representa de manera clara el manejo de la aplicación web.

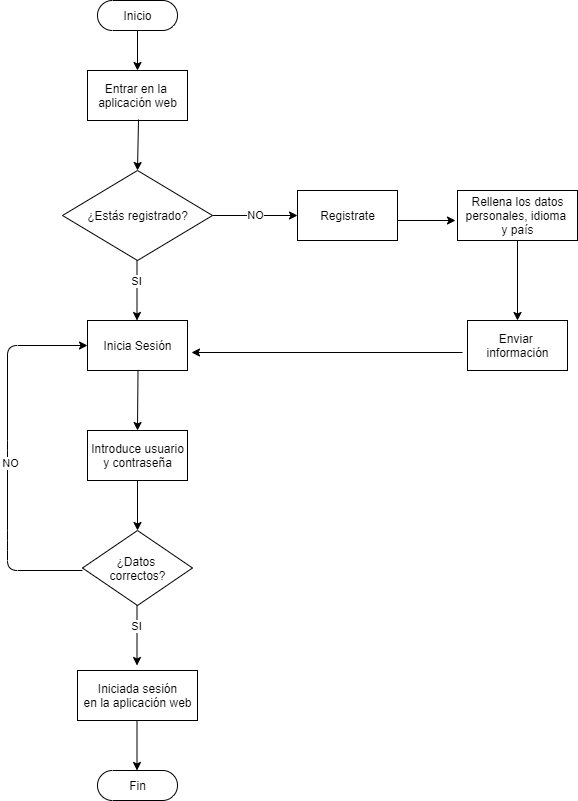


Figura 9. Diagrama de flujo inicio de sesión de la aplicación.

Desde la vista **Principal/General** se podrán seleccionar y añadir los productos deseados, hacer una búsqueda específica de productos, acceder a visualizar la lista de productos seleccionados para efectuar la compra, ver las características de las piedras de una manera más didáctica y completa e iniciar/cerrar sesión.

* **Búsqueda de productos.** Información de la empresa de venta, resultados de la búsqueda incluyendo las características y los aspectos de cada gema.

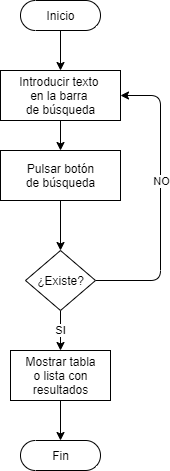


Figura 10. Diagrama de flujo de la búsqueda de productos.

* **Visualización detallada de los productos.** Tras la búsqueda se han de poder seleccionar los productos para acceder a su descripción y características físicas completas.



Figura 11. Diagrama de flujo de la visualización detallada de productos.

* **Proceso de compra.** Acceso al carrito de la compra, en el que aparecerán todos los productos seleccionados por el usuario, y acceso al proceso final de compra, al verificar la lista de la compra.

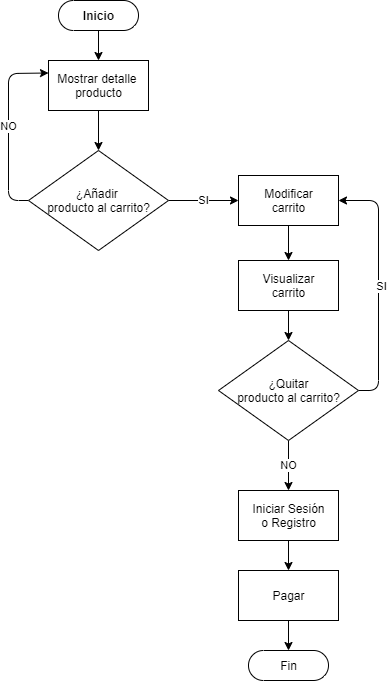


Figura 12. Diagrama de flujo del proceso de compra de productos.

También se ilustra la navegación por medio de un **mapa navegacional de la aplicación** web completa. Para que sea entendido y validado con claridad por el cliente.

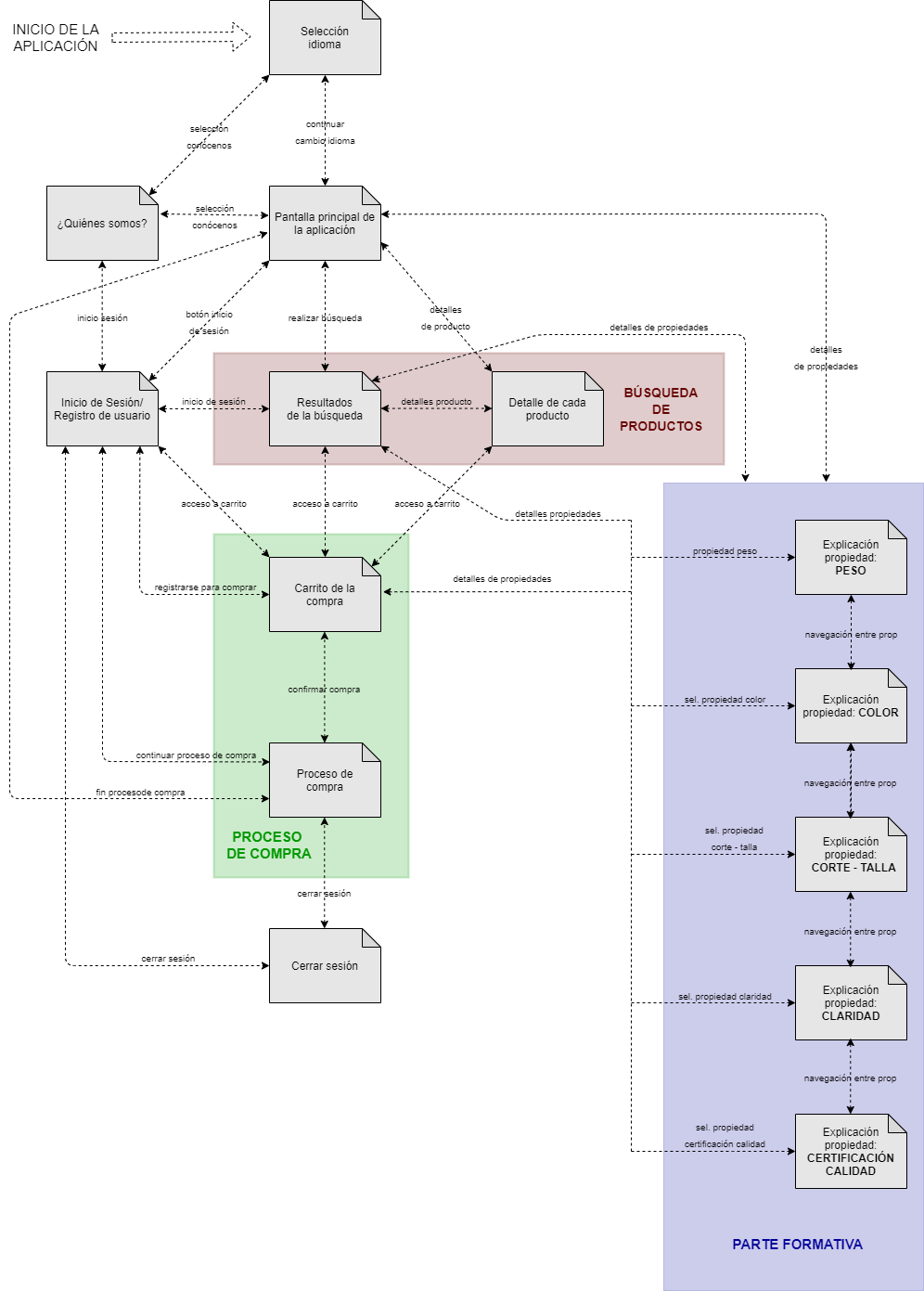


Figura 13. Mapa navegacional de la aplicación.

## 3.4. Análisis de los requisitos de presentación

Tras realizar el análisis de los requisitos de navegación se establecen las siguientes vistas principales. La primera vista con la que se encontrará el usuario al entrar en la aplicación será la **vista de selección de idioma**. En esta vista el usuario tendrá disponibles los idiomas indicados anteriormente para seleccionar.

Posteriormente, se pasará a la vista de aplicación principal, que se llamará **Principal/General**. También tendrá la posibilidad de conocer quién es la empresa, es decir, acceder a la vista de **¿Quiénes somos?**

**¿Quiénes somos?** Albergará información detallada de la empresa, incluyendo datos de contacto de tener alguna incidencia y dirección de asentamiento de la empresa.

La vista **Principal/General** mostrará todos los productos que se encuentran a la venta, incluyendo los detalles del producto que se han especificado en el área de requisitos de contenido de cada producto. Mostrará un botón de selección para añadir los productos deseados al carrito de compra, otro para realizar la búsqueda de productos, otro para ver con detalle las características y propiedades de las piedras, uno para acceder al carrito y otro para iniciar/cerrar sesión. En definitiva, esta será la vista padre desde la que se tendrá la posibilidad de acceder al resto de vistas.

**Vista de resultados de búsqueda**, aparecerá siempre tras realizar una búsqueda ya sea de tipo general o de tipo específico (previa selección de filtrado). En ella se dispondrán todos los productos que cumplan con los detalles de búsqueda descritos. Al igual que desde la vista general se podrá: seleccionar productos y añadirlos al carrito de la compra, visualizar el carrito de la compra, acceder a la vista de detalle de productos y a la de detalle de características, iniciar/cerrar sesión.

Además de la vista de resultados de búsqueda se tendrá una **vista de detalle de producto.** En ella se visualizarán imágenes/vídeos reales de la piedra, se detallarán las propiedades y las características físicas y se indicará su precio. Se tendrá la posibilidad de añadir al carrito este producto, visualizar el carrito completo, acceder a la información detallada de las propiedades y características o regresar a la vista anterior o a la vista principal.

Con más detalles aún si cabe se describirán las propiedades y características de cada piedra, añadiendo así la parte formativa y educativa que requería el cliente. Todas estas vistas tendrán el mismo formato y serán diferentes para cada propiedad, por lo que se tendrá: **vista de detalle propiedad peso, vista de detalle propiedad color, vista de detalle propiedad talla o corte, vista de detalle de claridad** y **vista de detalle de certificación de calidad del producto**.

**Vista del carrito de la compra o vista de lista de productos seleccionados**, desde aquí se podrán modificar el número de unidades a comprar, deseleccionar producto y finalizar el proceso de compra en caso de estar seguros de comprar los productos mostrados.

**Proceso de compra** se han de introducir los datos del usuario relativos al pago de los productos, y en caso de que el usuario no esté registrado y no se quiera registrar, los datos del usuario. Una vez finalizado se podrá regresar a la vista principal de la aplicación.

De acuerdo con los datos que se han de conocer de los usuarios y que marcará la diferenciación entre los tipos de usuario (usuario no registrado de usuario registrado) se tienen las siguientes vistas: **inicio de sesión**, en la que el usuario previamente registrado introducirá las credenciales necesarias para el acceso; **registro de usuario**, será un formulario estándar en el que se rellenarán los datos como: nombre, apellidos, email de registro y contraseña, teléfono de contacto y dirección; y **cierre de sesión**, simplemente se seleccionará el cierre de sesión y se volverá a la vista principal.

Por lo tanto, entre los requisitos de presentación se tiene:

* **Botón para la visualización del idioma y país.** Para hacer el cambio de idioma, la aplicación ha de tener un botón o algún elemento de interacción intuitivos con el que efectuar ese cambio. La aplicación está dirigida a usuarios de Europa y América por lo que se han de incorporar diferentes idiomas. Los idiomas principales que se han seleccionado para incorporar tras analizarlo con el cliente son: inglés británico, inglés estadounidense, español de España, español de Latinoamérica, alemán, italiano, francés, portugués de Portugal y portugués de Brasil.
* **Visualización de opciones de búsqueda.** La búsqueda ha de poder realizarse a través de botones u otros elementos de interacción con todas las opciones de búsqueda disponibles, tanto de características como de propiedades.
* **Visualización de los resultados de la búsqueda.** Tras finalizar la búsqueda se ha de poder representar los resultados obtenidos en la búsqueda en una tabla o rejilla al efecto. Con algún elemento que nos lleve a la vista detallada individual.
* **Visualización detallada individual de la información de cada gema.** Se ha de mostrar la información completa, características, gráficas e imágenes reales, y el precio de cada uno de los productos seleccionados.
* **Botón para inicializar el proceso de Registro o Iniciar (Iniciar Sesión o Regístrate)**
* **Botón que permita seleccionar los productos para realizar la compra.** Para efectuar el proceso de compra se han de seleccionar los productos deseados, por lo que se precisa de un botón de selección. La imagen será un carrito clásico de supermercado.
* **Botón que permita navegar a la vista de compra.** Antes de finalizar el proceso de compra se ha de comprobar que todos los datos son correctos, por lo que se ha de poder acceder a la lista/carrito de la compra. Para ello se dispondrá de este botón.
* **Botón de Realizar el Pago**. Con enlace que nos lleva a una pasarela bancaria de compra.

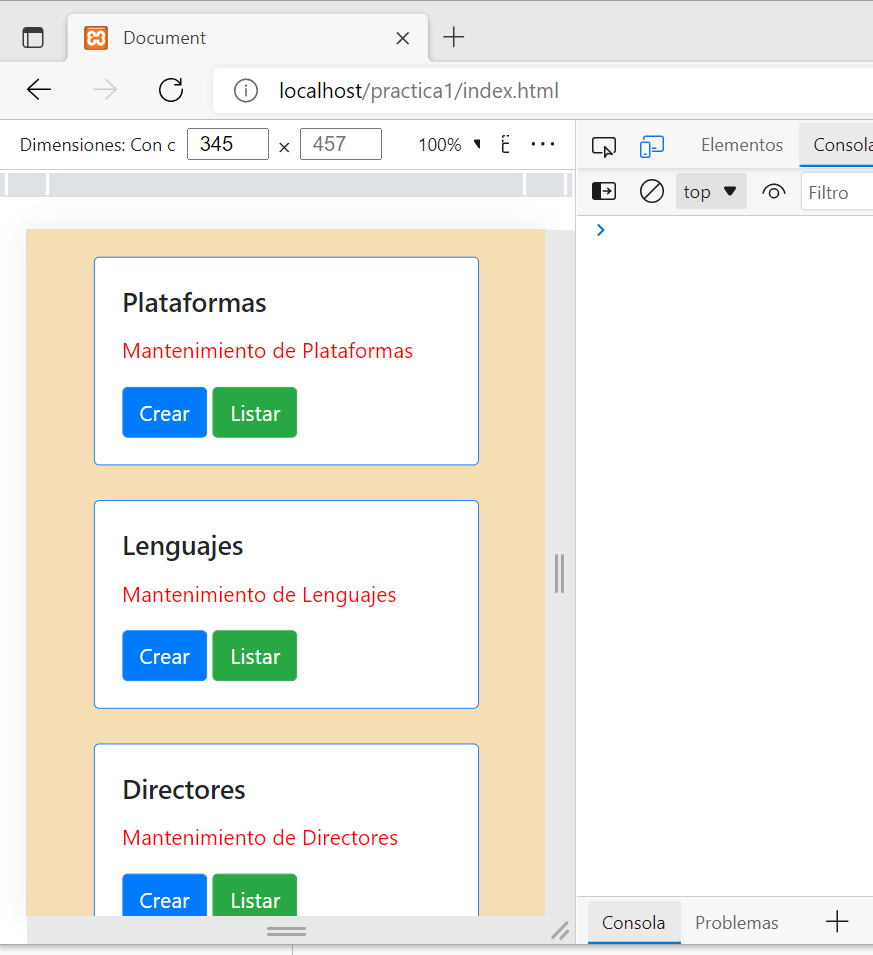
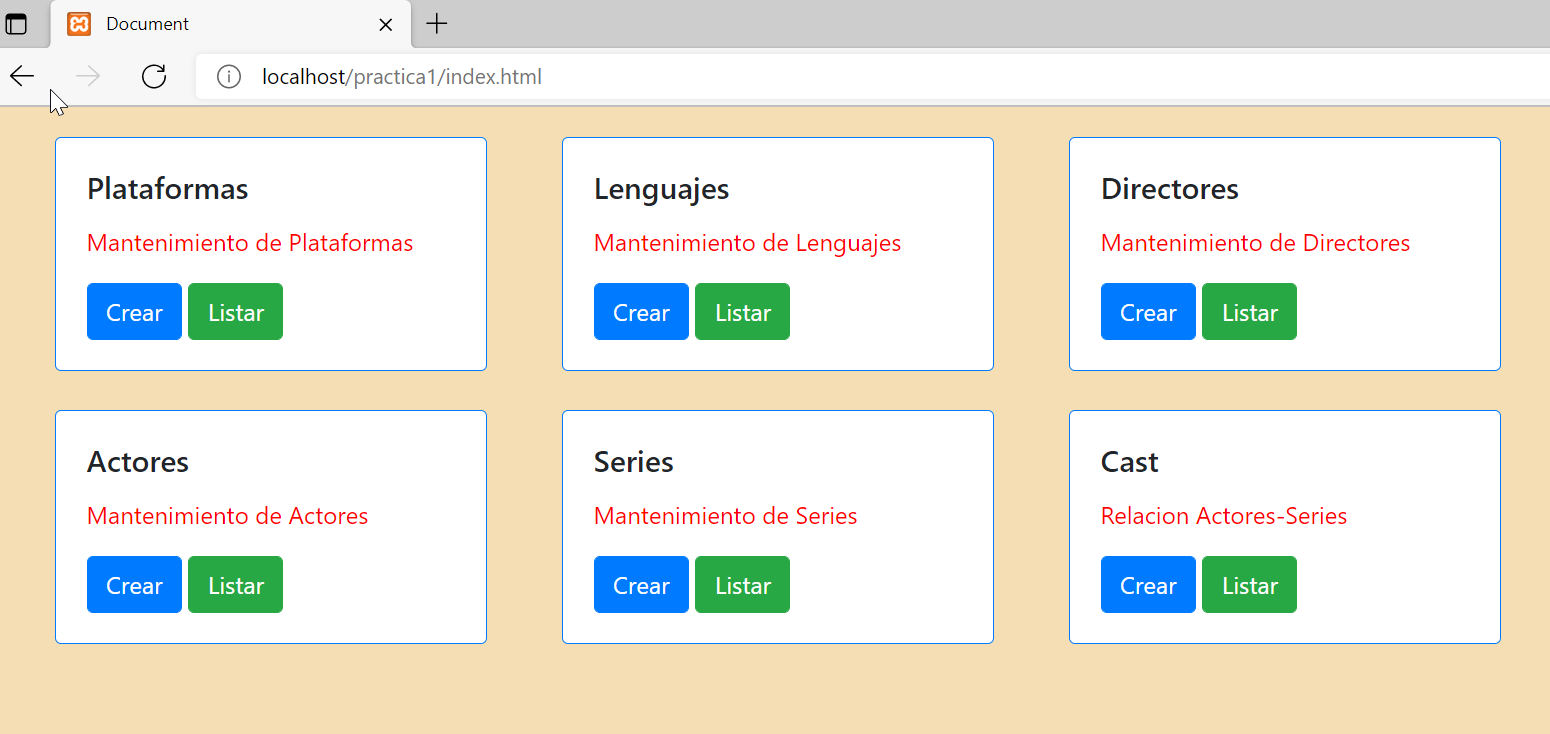
Interfaz de usuario gráfica

Descripción generada automáticamente

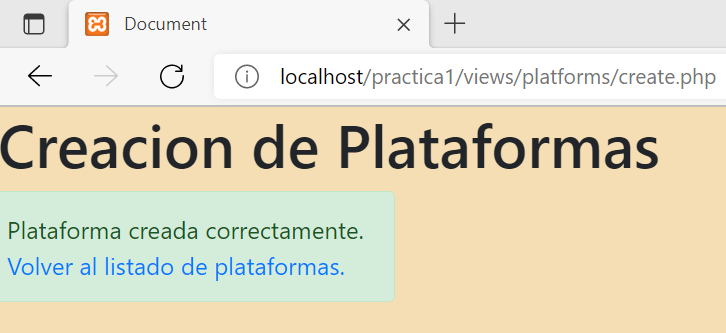
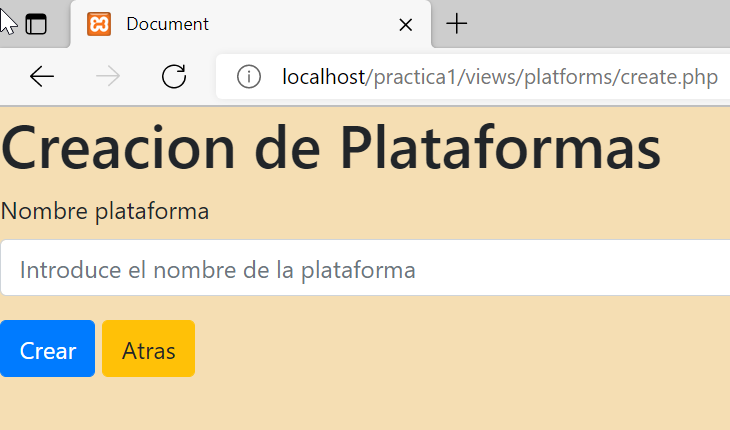
Figura 14. Diagrama de representación de vistas de la aplicación web.

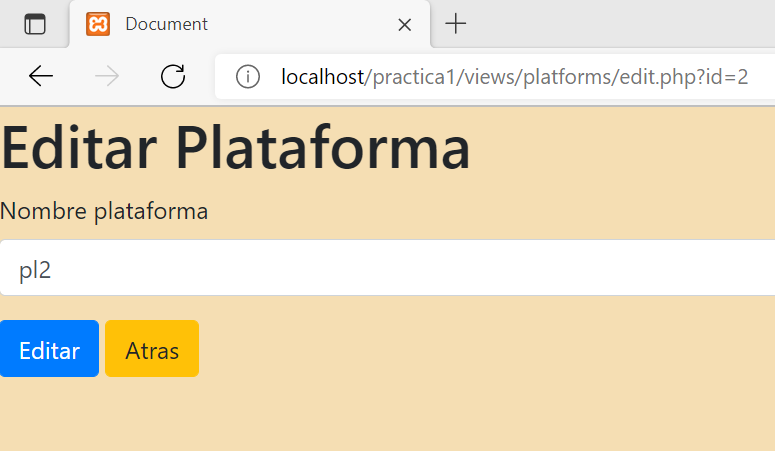
# Vistas de la Aplicación

### Pantalla inicial

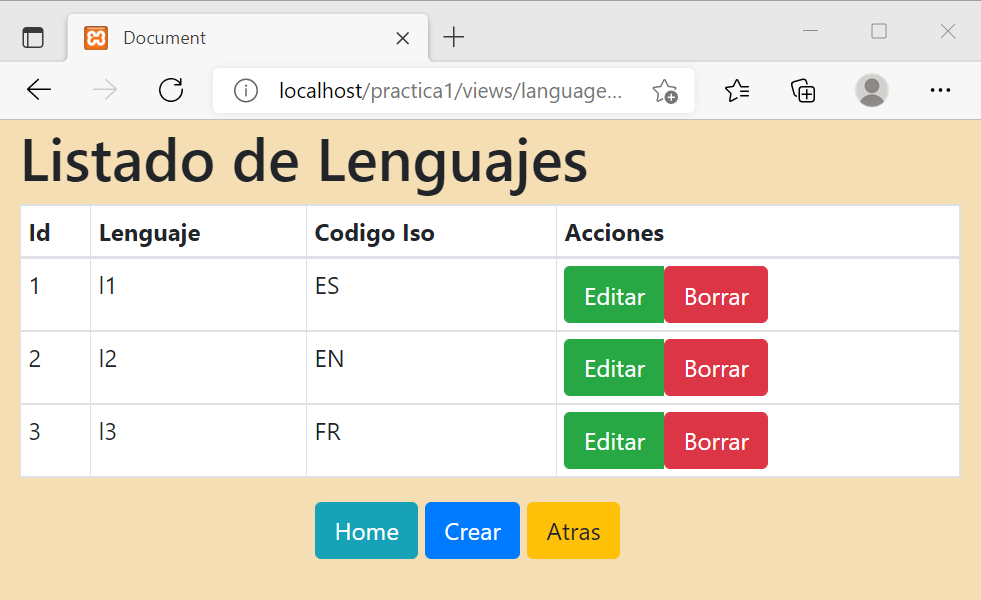
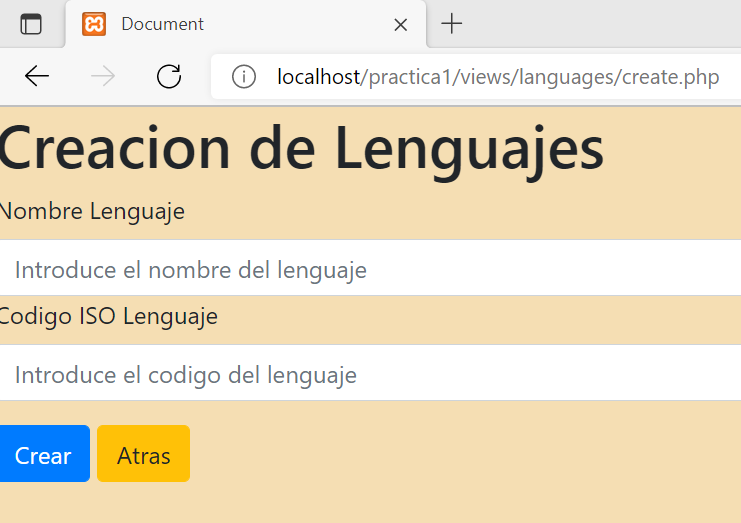


### Creación, Listado, Edición y Borrado de Plataformas





### Creación, Listado, Edición y Borrado de Lenguajes



### Creación, Listado, Edición y Borrado de directores



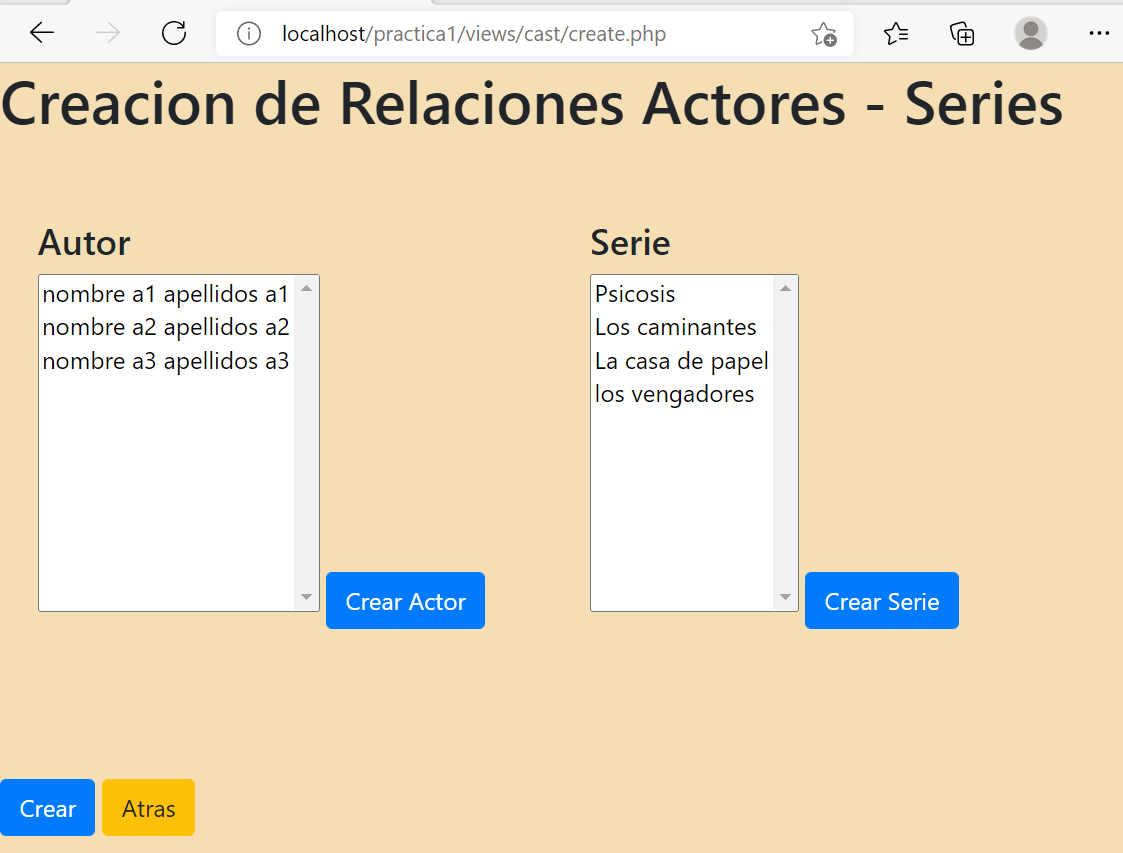
### Creación, Listado, Edición y Borrado de Actores



### Creación, Listado, Edición y Borrado de series



### Creación, Listado, Edición y Borrado de la relación Actores - Series



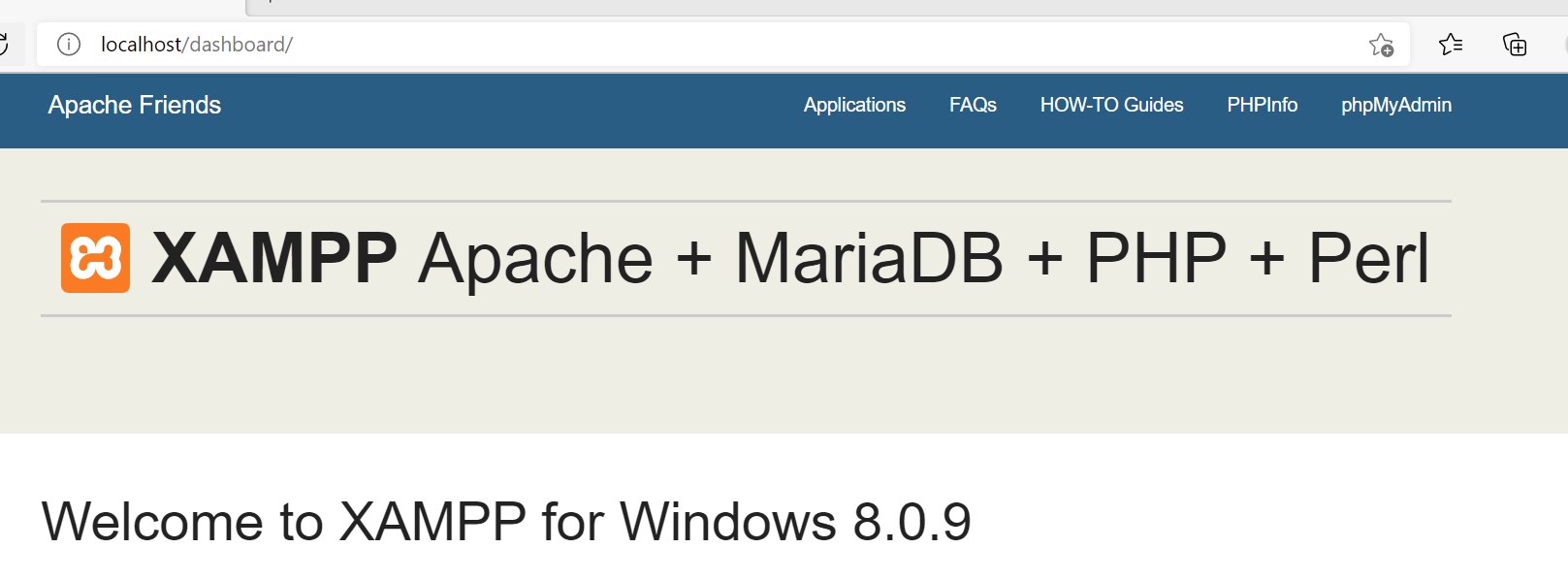
# 4. Implementación

**4.1 configuración del servidor**

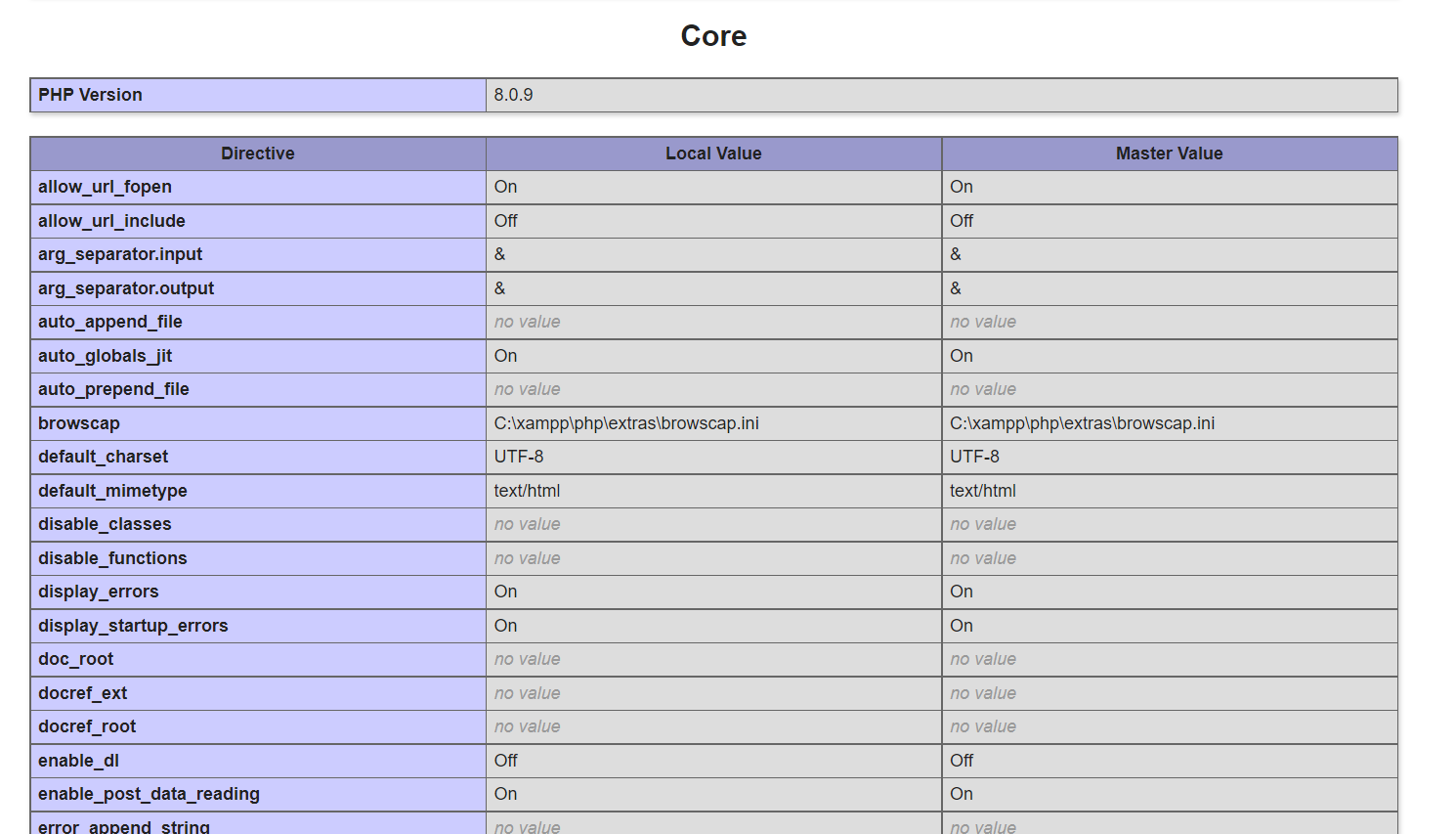
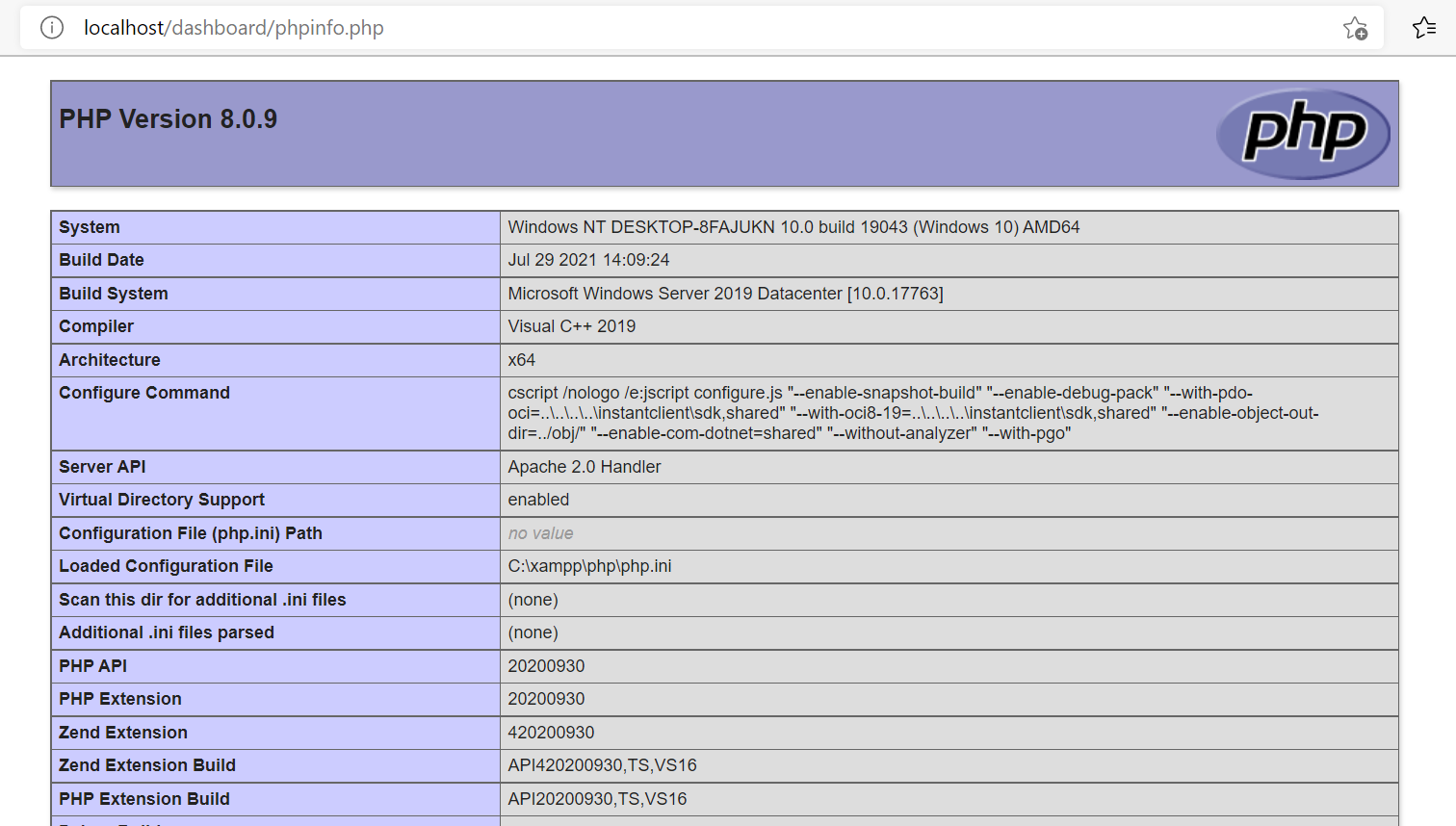
**Paquete de Aplicaciones XAMP**

Se instala un servidor PHP a través del paquete XAMP con las opciones por defecto.

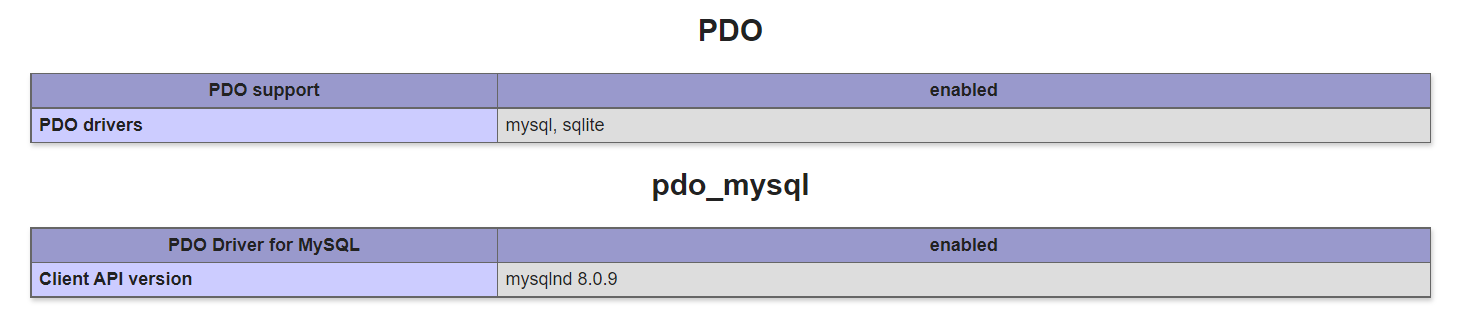
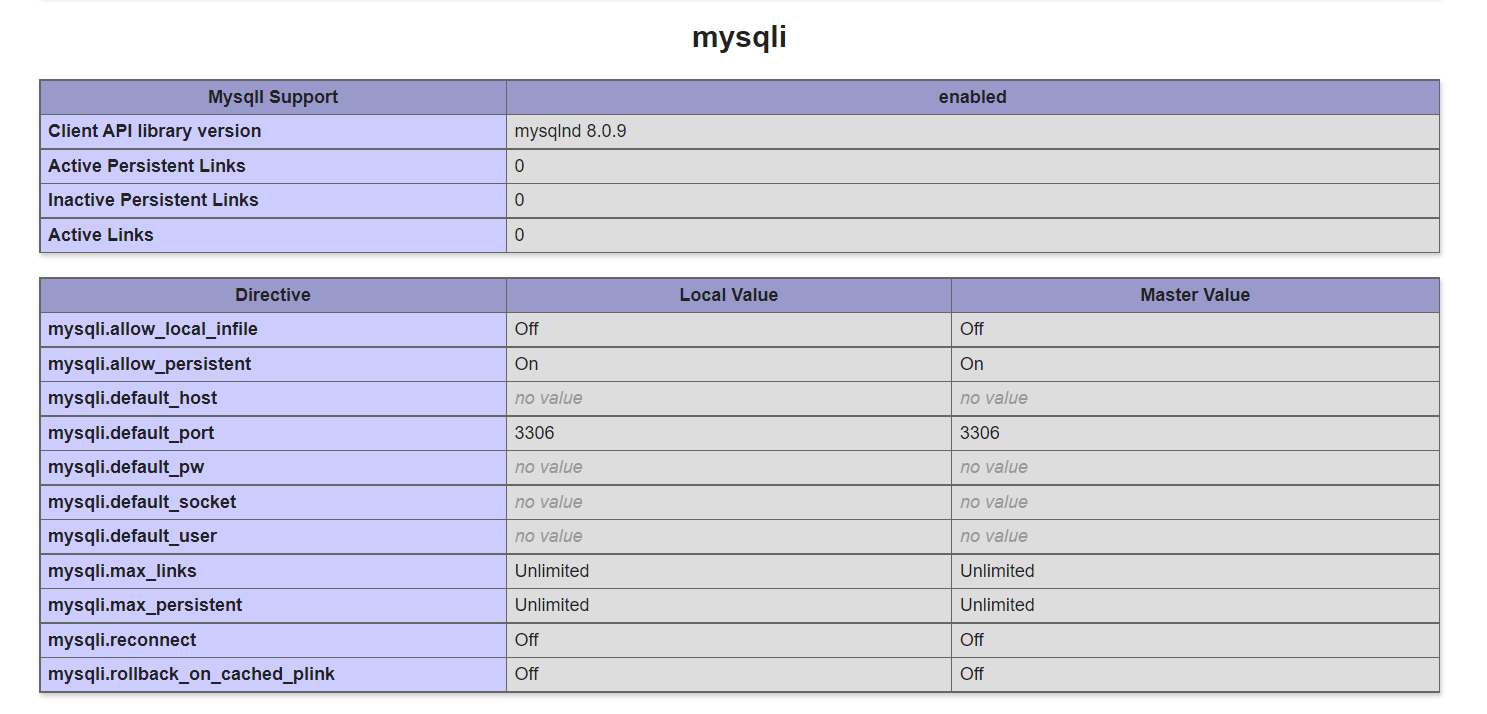
Se habilita la opción de mostrar errores: display errors on



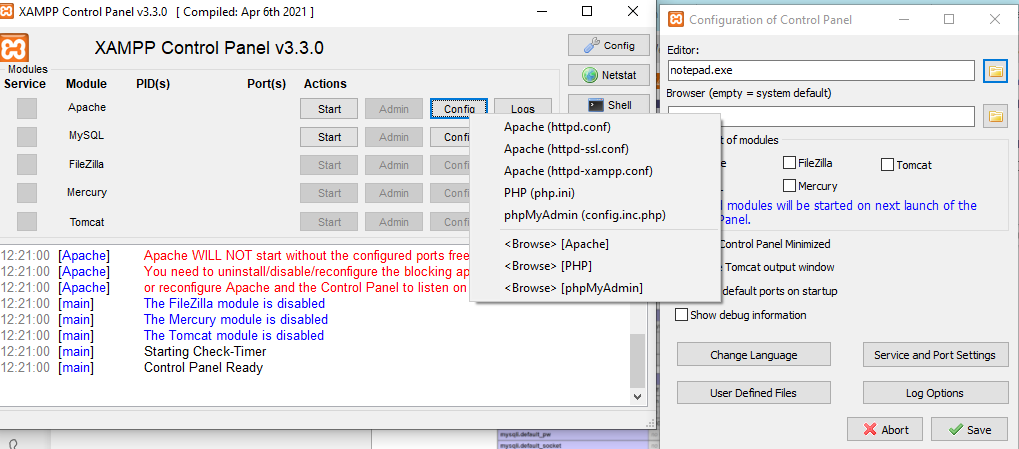
Esta “pila” de aplicaciones trae PHP y los drives de conexión necesarios configurados



Driver



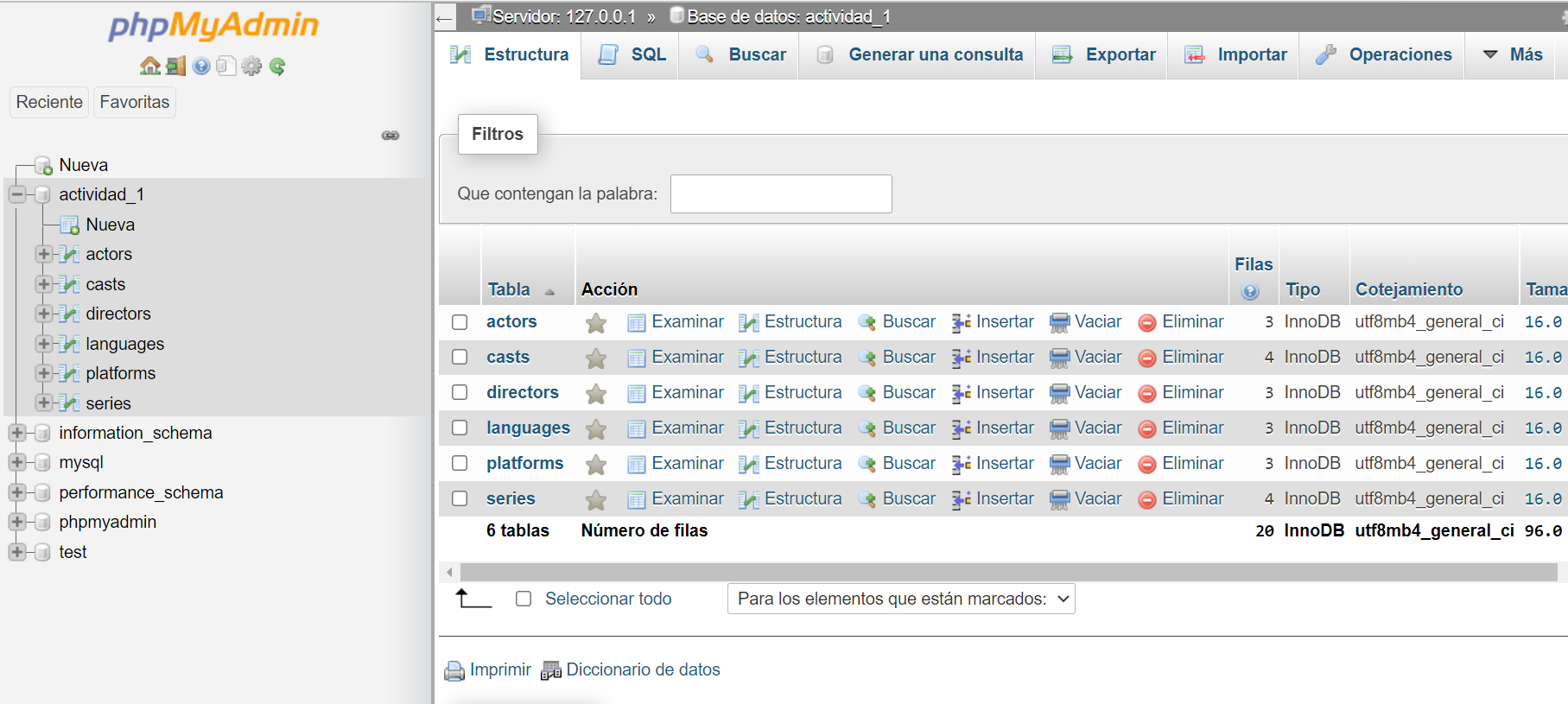
XAMP trae un panel de control desde donde acceder con comodidad a todas los fichero y opciones normales de configuración



**Creación de la tablas y campos**

Se utiliza tanto phpMyAdmin como Visual studio code con una extensión para acceder al SGBD mysql

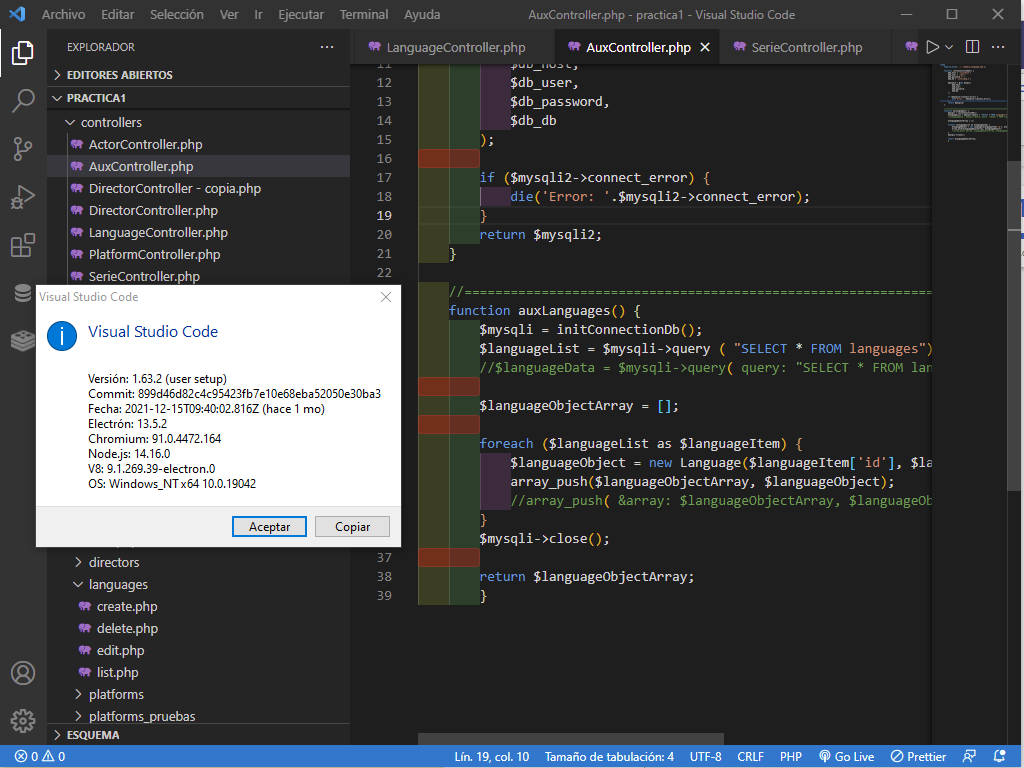
phpMyAdmin



**Visual Estudio Code**

Como entorno de programación, por el conocimiento de la herramienta, se utiliza Visual studio Code.

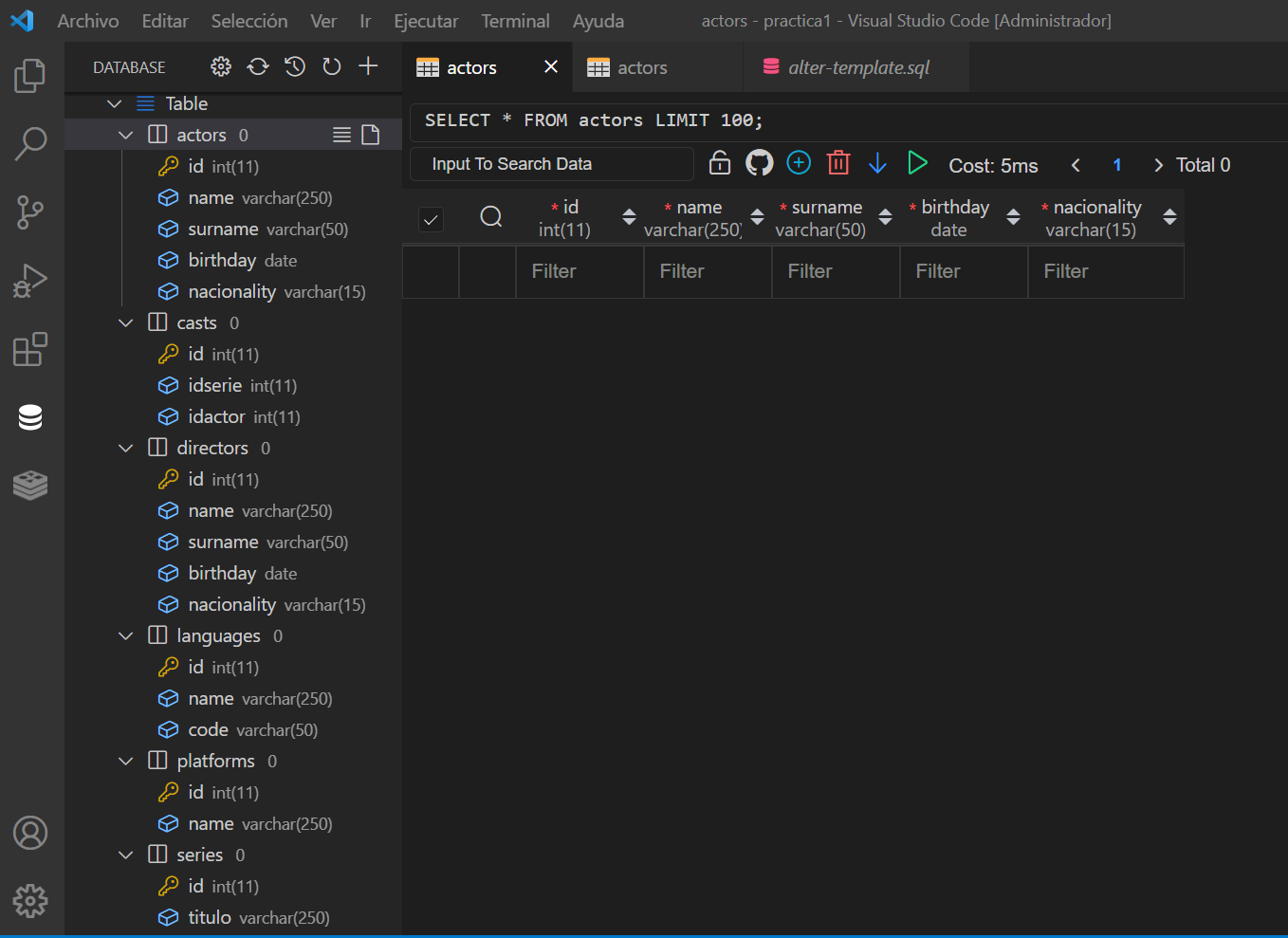
Visual Studio Code



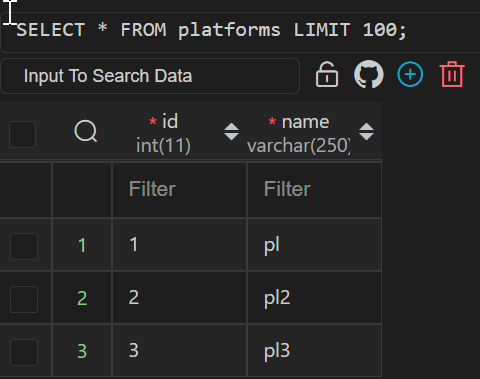
Extensión para acceder a MySQL (o Mariadb)



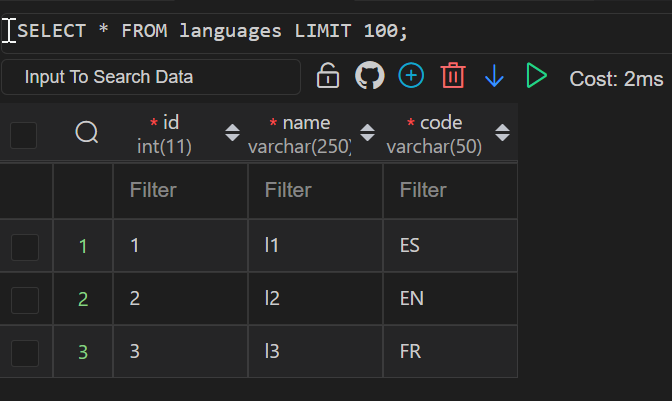
Se usa el acceso a la base de datos desde Visual Studio Code, tanto para operaciones CRUD como para rediseñar la base de datos.



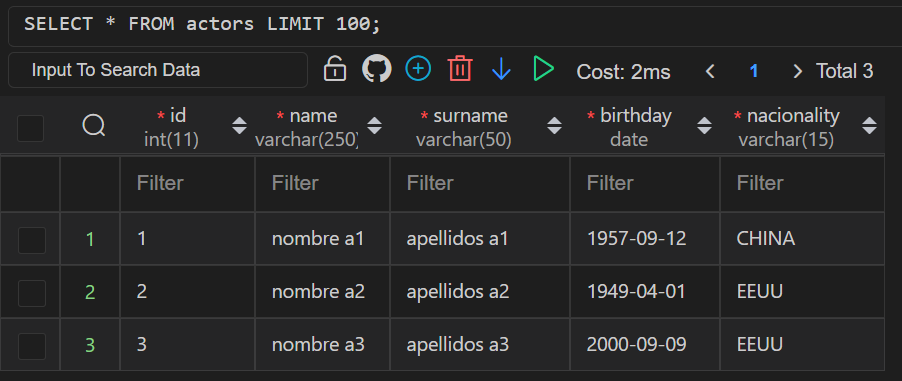
Listado de Plataformas



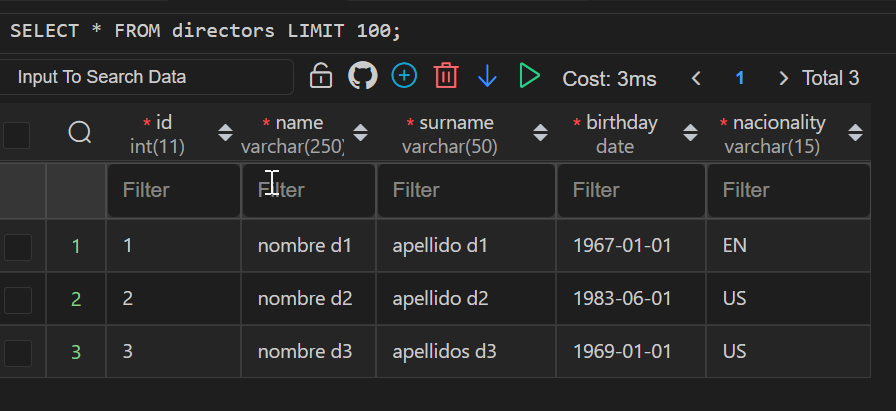
Listado de Lenguajes



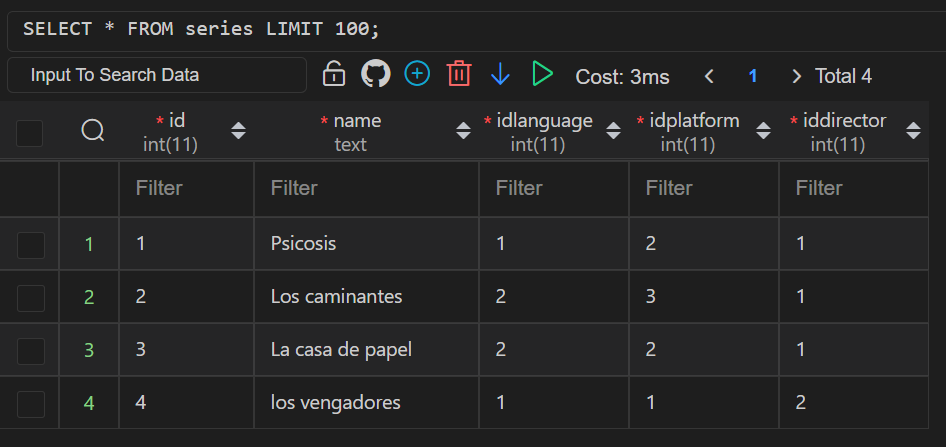
Listado de Actores



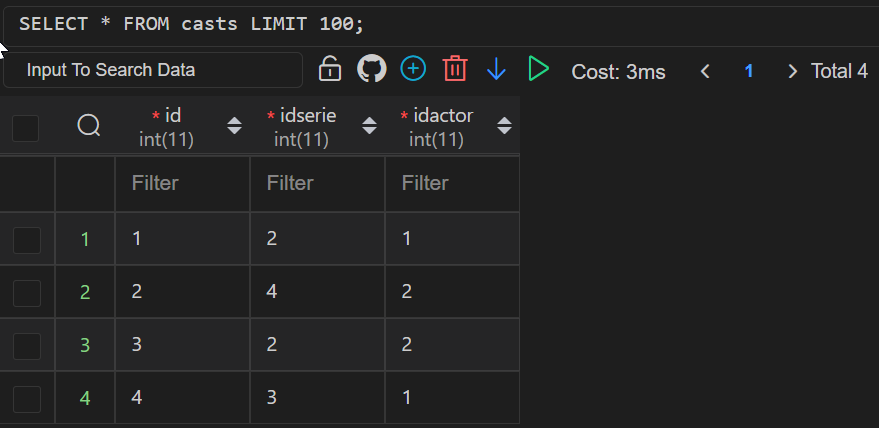
Listado de Directores



Listado de Series



Listado de Relaciones Actores-Series



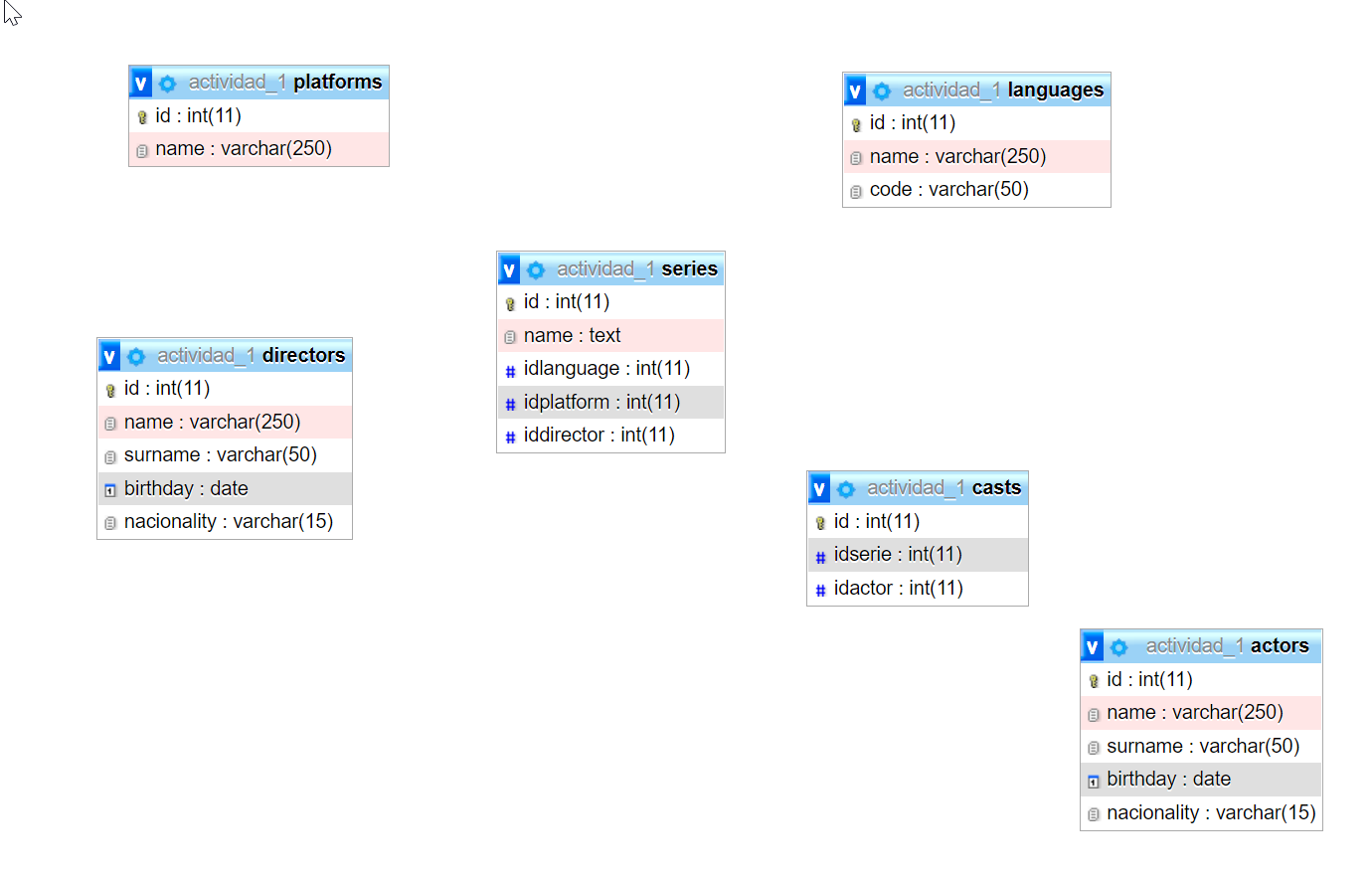
### **Codificación**

Se incluye ciertos detalles de código considerados de más interesantes.

### Creación de la Base de datos Biblioteca de Series

Se crean en total seis entidades. cinco de ellas (Plataforma, Idioma, Serie, Actor y Directo) vienen definido por los requisitos definidos en la actividad. Además, se crea una entidad más para representar la Relación “muchos a muchos” entre series y actores.

Con el diseñador de phpmyadmin, podemos ver el detalle de las seis tablas creadas



No se establecen relaciones entre campo de las tablas. Las relaciones quedan establecidas en la Lógica de la propia aplicación. En desarrollos posteriores se pueden definir relaciones en la base de datos y capturar los errores producidos en los borrados y actualizaciones.

Se realiza la copia de seguridad y script con phpMyAdmin

------------------------------------

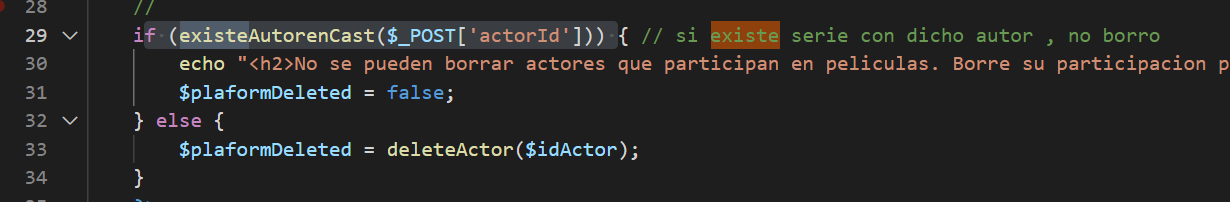
I\*\*\*\* inserta pantalla

I

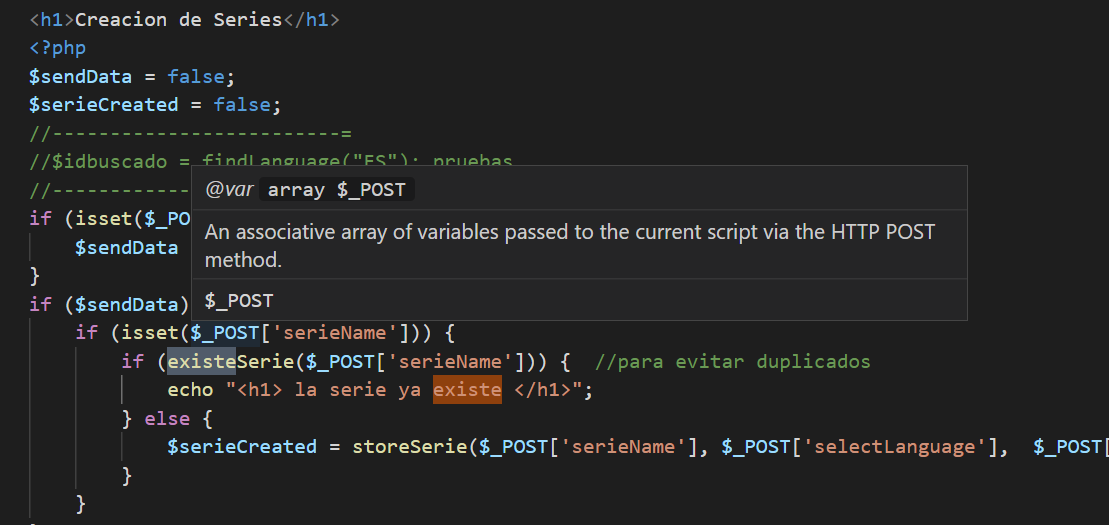
I

----------------------------------------------

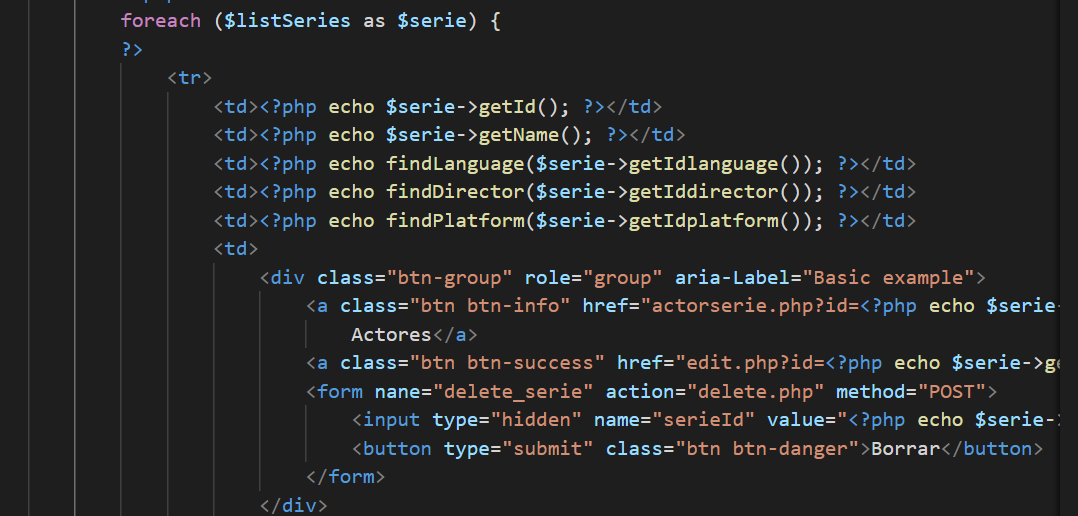
Ejemplo de funciones creadas dentro de los controladores para localizar la existencia de un registro en una tabla. Se ha usado tanto para evitar duplicados, como para controlar el borrado de registros que son referenciados por una clave externa.

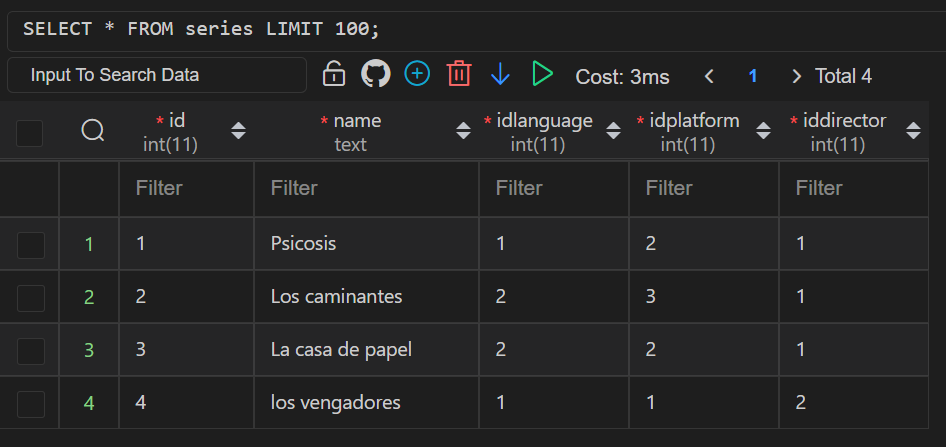


Ejemplo en el que se evitan duplicados en Series. Tablas que previsiblemente tendrán pocos registros como la de plataformas, no tienen implementado esta funcionalidad

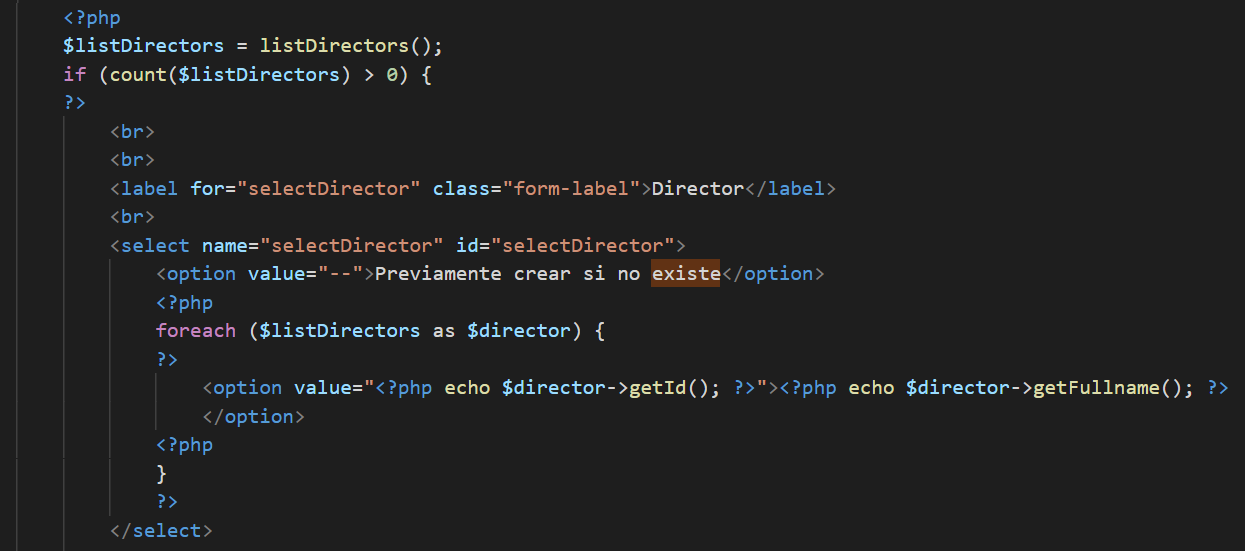


Ejemplo de código usado para mostrar los registros un **select**. También muestras las funciones findLanguage, findDirector y findPlatform para que la tabla muestre los nombres y no los números correspondientes a las claves foráneas. Se usa en todas las tablas con clave foránea



 se muestra como 

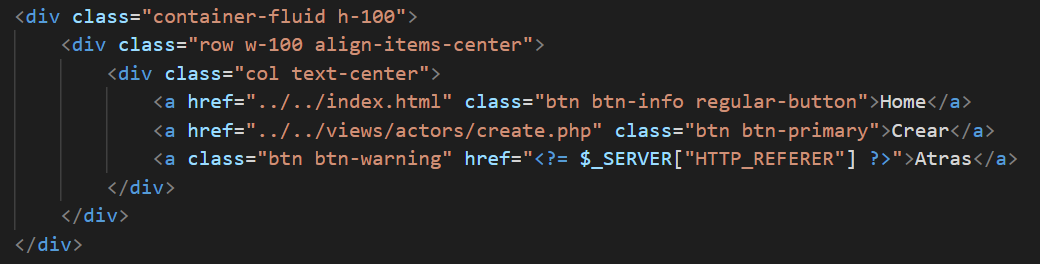
Otro ejemplo de Creación de un Select



Ejemplo de función que cambia el formato año-mes-dia a dia/mes/año



Botón de retroceder. Se ve también como mediante ../.. se recorre la estructura



# 5. Conclusiones

A través de esta Actividad se ha puesto en práctica muchos de los conocimientos teóricos aprendidos en la asignatura.

Se ha puesto en práctica el patrón MVC Modelo Vista Controlador, creándose un Modelo, Uno o varios controladores y vistas para las distintas operaciones CRUD.

En el modelo se han mapeado los campos de las tablas a través de una clase que representa la tabla. Dichas clases tienen definidas sus propiedades y métodos respectivos.+

En los controladores se han definido la conexión a la base de datos, y las distintas funciones y operaciones que relacionan las vistas y el modelo.

Se crea una vista por cada una de las pantallas de la aplicación.

LA relación entre tablas, aunque simplificada, se ha controlado por código. No se ha escogido para esta actividad un control de la integridad realizado por el sistema de gestión de base de datos, bien a través de reglas definidas, bien a través del uso de disparadores.

Se ha trabajado mucho la codificación html-css-php. Constantemente se ha estado pasando de código HTML a PHP y viceversa. En este punto, si bien mi experiencia con PHP era nula, me ha ayudado mi experiencia con PHP clásico.

Se ha trabajado con css en línea, utilizando bootstrap. El resultado de presentación ha sido satisfactorio incluido la respuesta responsive a diversas pantallas.

Por último, a pesar de que mi equipo de desarrollo es un portátil i5, de quinta generación, la velocidad de ejecución es casi instantánea lo que mejora mucho la experiencia de usuario.

Los posibles desarrollos futuros pueden ser: Una mejor ordenación de los controller, Una ampliación y rediseño de algunas tablas, un sistema de consultas más avanzado y flexible y una unificación de todas las características de diseño en un fichero css.

# 5. Bibliografía

[1] Definición y descripción del marco de referencia Fraternali, recuperado de: <https://eprints.ucm.es/id/eprint/4553/1/T25910.pdf>

[2] *Figura 1.* Recuperada íntegramente de los apuntes de la asignatura: Ingeniería de Software Web: Bianchini, A. (2014). WEBFDM: un método flexible para el desarrollo de aplicaciones Web. Ph.D.Thesis Universidad Simón Bolivar, Caracas.

[3] Información sobre los diagramas de uso UML, recuperado de: <https://www.lucidchart.com/pages/uml-use-case-diagram>

<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml/#section_8>

<https://blog.comparasoftware.com/diagramas-de-uml-que-significa-esta-metodologia/>

