

Tecnologías de la Información, Área Desarrollo de Software  
Multiplataforma

# Investigación de proyecto final



Asignatura: Aplicaciones web para 14.0



08 DE FEBRERO DE 2024

BENJAMIN PEÑA MARIN

VANIA DONAJI VELAZQUEZ TORRES

Grupo. 1

Docente: Edel Meza Hernandez

# Entorno y configuración

## ¿Qué es?

CodeIgniter es un marco de desarrollo de aplicaciones (un conjunto de herramientas) para personas que crean sitios web utilizando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría hacerlo si estuviera escribiendo código desde cero, proporcionando un amplio conjunto de bibliotecas para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas. CodeIgniter le permite concentrarse creativamente en su proyecto minimizando la cantidad de código necesario para una tarea determinada.

Siempre que es posible, CodeIgniter se ha mantenido lo más flexible posible, permitiéndole trabajar de la manera que desee, sin verse obligado a trabajar de una manera determinada. La estructura puede tener partes centrales fácilmente ampliadas o reemplazadas por completo para que el sistema funcione como usted lo necesita. En resumen, CodeIgniter es el marco maleable que intenta proporcionar las herramientas que necesita sin estorbar.

## Requisitos

### Servidor

#### PHP y extensiones requeridas

Se requiere PHP versión 7.4 o posterior, con las siguientes extensiones de PHP habilitadas:

- internacional
- cadenamb
- json

### Extensiones PHP

Las siguientes extensiones PHP deben estar habilitadas en su servidor:

- mysqlnd (si usas MySQL)
- curl (si usa CURLRequest )
- imagick (si usa la clase Image ImageMagickHandler)
- gd (si usa la clase de imagen GDHandler)
- simplexml (si formatea XML)
- Se requieren las siguientes extensiones de PHP cuando utiliza un servidor de caché:
  - memcache (si usa la clase Cache MemcachedHandler con Memcache)
  - memcached (si usa la clase Cache MemcachedHandler con Memcached)
  - redis (si usa la clase Cache RedisHandler)

Se requieren las siguientes extensiones PHP cuando usa PHPUnit:

- dom (si usa la clase TestResponse )
- libxml (si usa la clase TestResponse )
- xdebug (si usas CIUnitTestCase::assertHeaderEmitted())

## Bases de datos compatibles

Se requiere una base de datos para la mayoría de la programación de aplicaciones web.

Las bases de datos actualmente soportadas son:

- MySQL a través del MySQLi controlador (solo versión 5.1 y superior)
- PostgreSQL a través del Postgre controlador (solo versión 7.4 y superior)
- SQLite3 a través del SQLite3controlador
- Microsoft SQL Server a través del SQLSRV controlador (solo versión 2005 y superior)

- Base de datos Oracle a través del OCI8controlador (solo versión 12.1 y superior)

No todos los controladores se han convertido/reescritos para CodeIgniter4. La siguiente lista muestra los más destacados.

- MySQL (5.1+) a través del controlador pdo
- Oracle a través de los controladores pdo
- PostgreSQL a través del controlador pdo
- MSSQL a través del controlador pdo
- SQLite a través de los controladores sqlite (versión 2) y pdo
- CUBRID a través de los controladores cubrid y pdo
- Interbase/Firebird a través de los controladores ibase y pdo
- ODBC a través de los controladores odbc y pdo (debe saber que ODBC es en realidad una capa de abstracción)

## Descarga y Configuración del Servidor

Investigamos un poco sobre las condiciones que nos pide CodeIgniter para poder trabajar con él, y para eso buscamos la opción más óptima la cual es un servidor ya configurado con todas las especificaciones y características.

En este caso encontramos Laragon, para su descarga nos vamos directamente a la página de laragon.

The screenshot shows a Google search results page for the query "laragon". The top navigation bar includes tabs for "Reporte\_CodeIgniter - Documentación", "La licencia MIT (MIT) — Documentación", "psr que es en php - Buscar con Google", and "laragon - Buscar con Google". Below the search bar, there are filters for "Imágenes", "Videos", "Download", "Mac", "Linux", "Laravel", "150mg", "PHP", and "Características". The main search results are for "Laragon", showing the official website at <https://laragon.org>. The first result is titled "Download" and provides links for "Benefits of Laragon", "Download Laragon - Full (173 MB)", "Download Laragon - Portable (38 MB)", and "Laragon 5.0 (32-bit) - Migrate from XAMPP - Migrate from WAMP - Install Rails". Below the results, there is a section titled "Más preguntas :" with expandable questions about Laragon's purpose, usage, creation capabilities, and database creation.

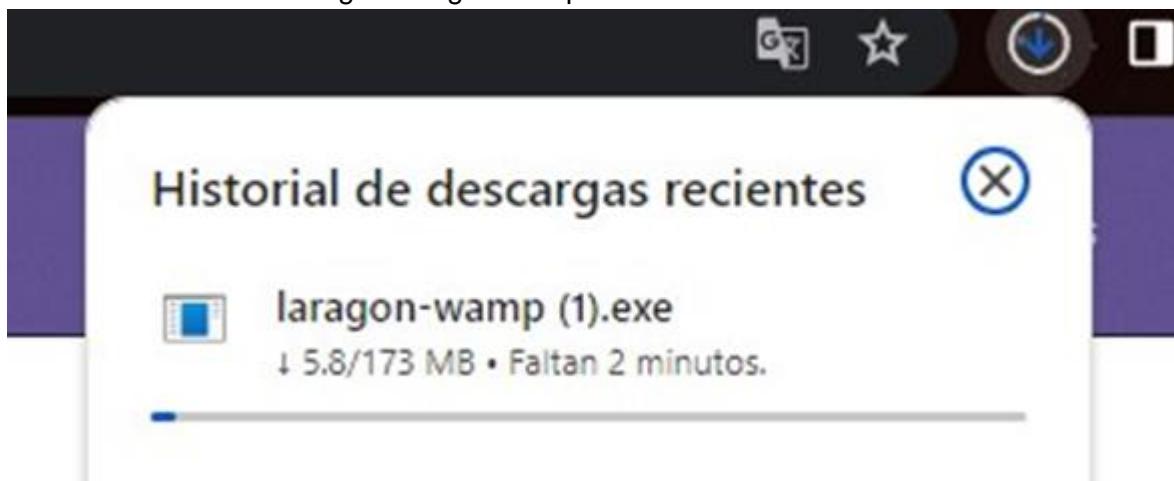
Laragon es un entorno de desarrollo universal. Tiene muchas características para hacerte más productivo:

**Beneficios de Laragon**

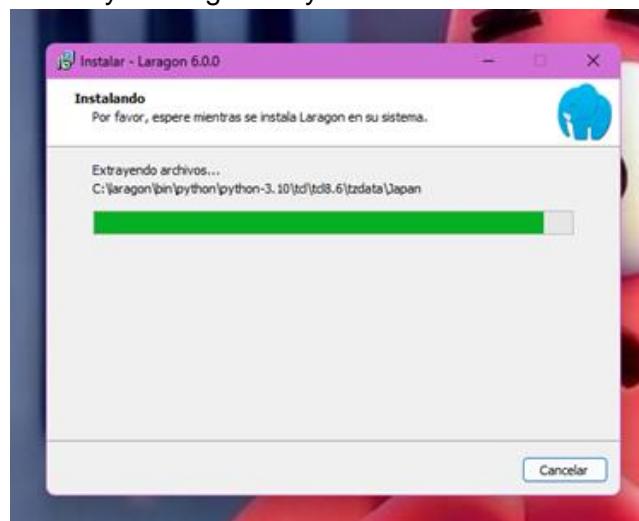
Después de la descarga, puedes agregar git, phpmyadmin, Node.js/MongoDB, Python/Django/Flask/Postgres, Ruby, Java, Go usando "Herramientas > Agregar rápido".

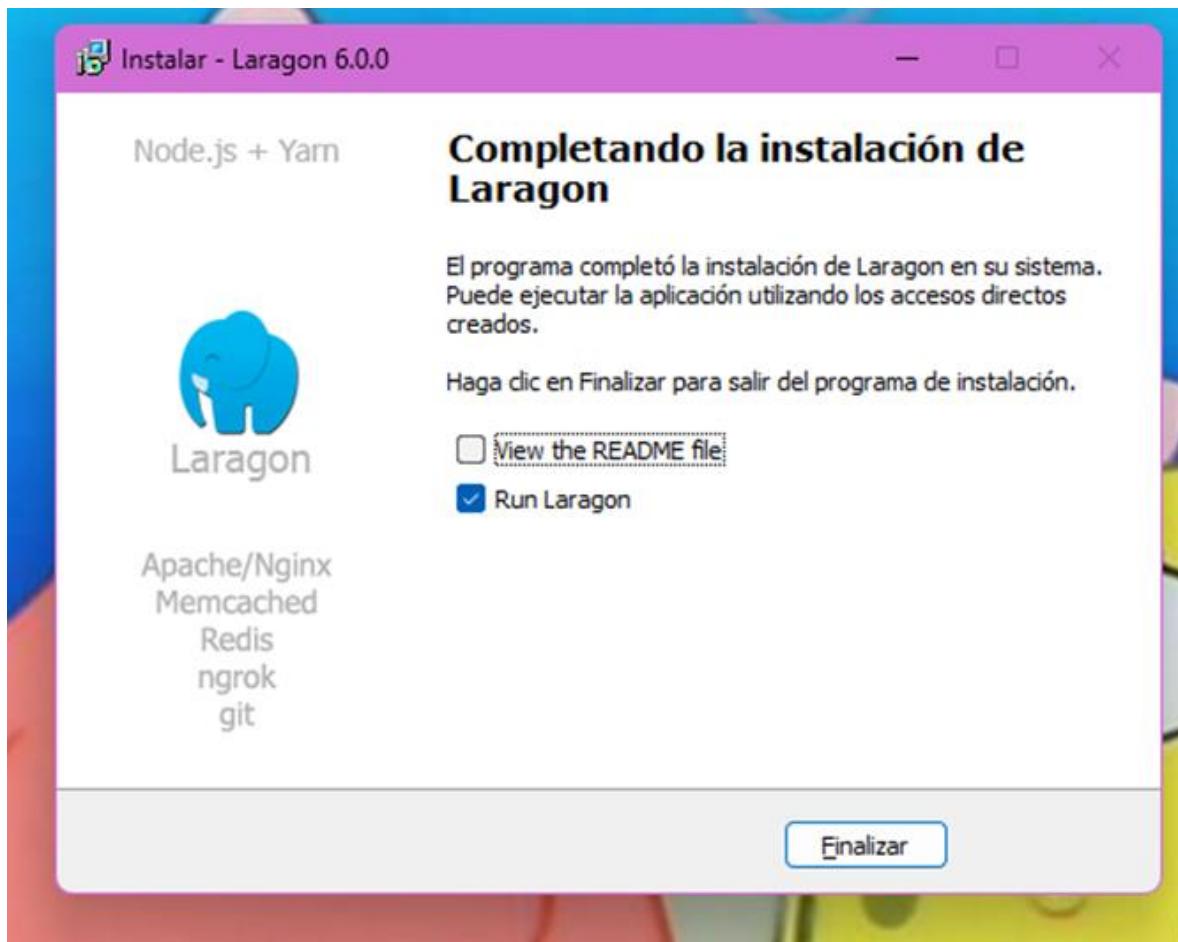
Nota: También puedes descargar desde GitHub

Le damos click en descargar Laragon Completo.

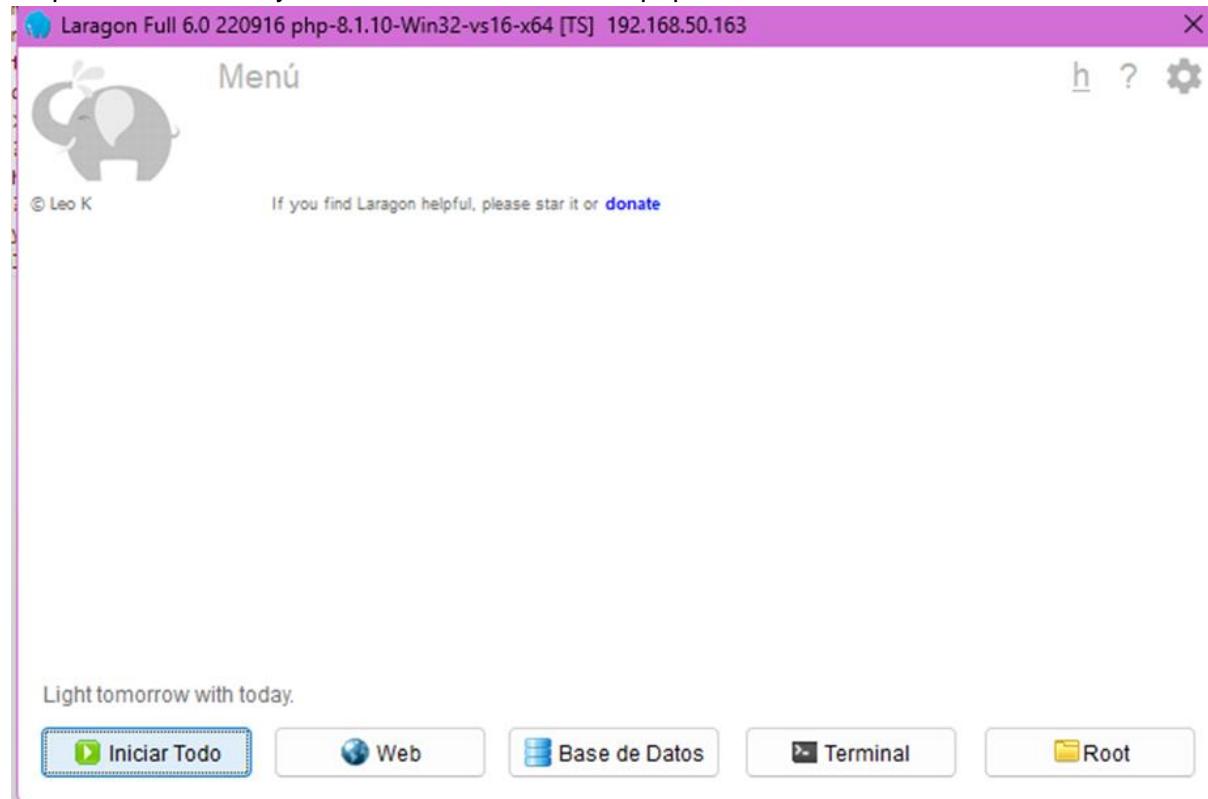


Esperamos a que se descargue. Este navegador ya viene con PHP configurado, un servidor web que es Apache también ya configurado y una conexión a base de datos con MySQL.





Ya que está instalado, lo único que debemos hacer es iniciar nuestro servidor para poder empezar a codificar y ver nuestros archivos con php.



Le damos en iniciar todo.

Después nos iremos a la carpeta que se creó cuando instalamos laragon que es en C.

Nombre	Fecha de modificación	Tipo	Tamaño
Drivers	13/01/2024 03:03 a. m.	Carpeta de archivos	
Gradle	22/01/2024 08:16 a. m.	Carpeta de archivos	
Intel	08/02/2024 12:28 p. m.	Carpeta de archivos	
laragon	07/02/2024 03:34 p. m.	Carpeta de archivos	

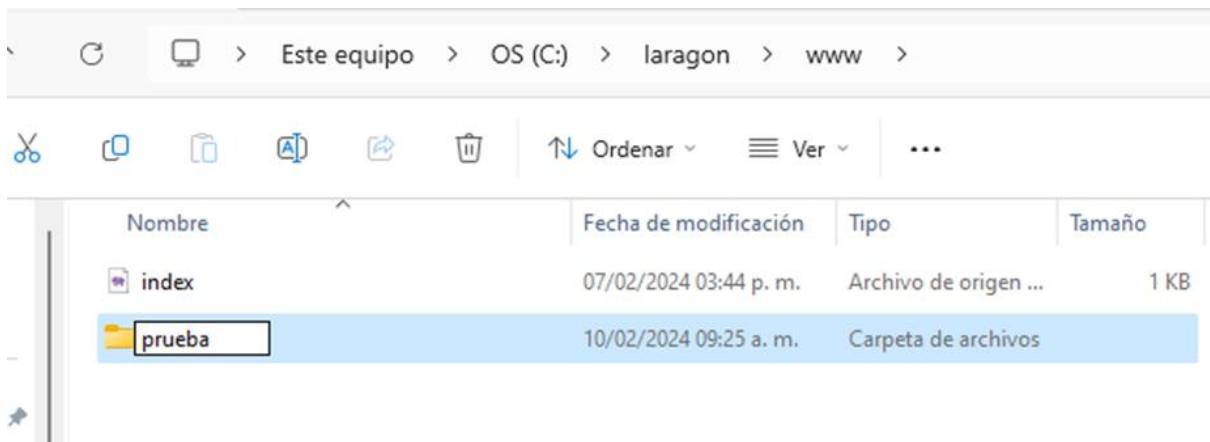
Damos clic y nos dará las siguientes carpetas.

Nombre	Fecha de modificación	Tipo	Tamaño
bin	07/02/2024 03:34 p. m.	Carpeta de archivos	
data	07/02/2024 03:35 p. m.	Carpeta de archivos	
etc	07/02/2024 03:34 p. m.	Carpeta de archivos	
tmp	07/02/2024 04:48 p. m.	Carpeta de archivos	
usr	07/02/2024 03:35 p. m.	Carpeta de archivos	
www	07/02/2024 03:38 p. m.	Carpeta de archivos	
laragon	16/09/2022 04:55 p. m.	Aplicación	2,108 KB

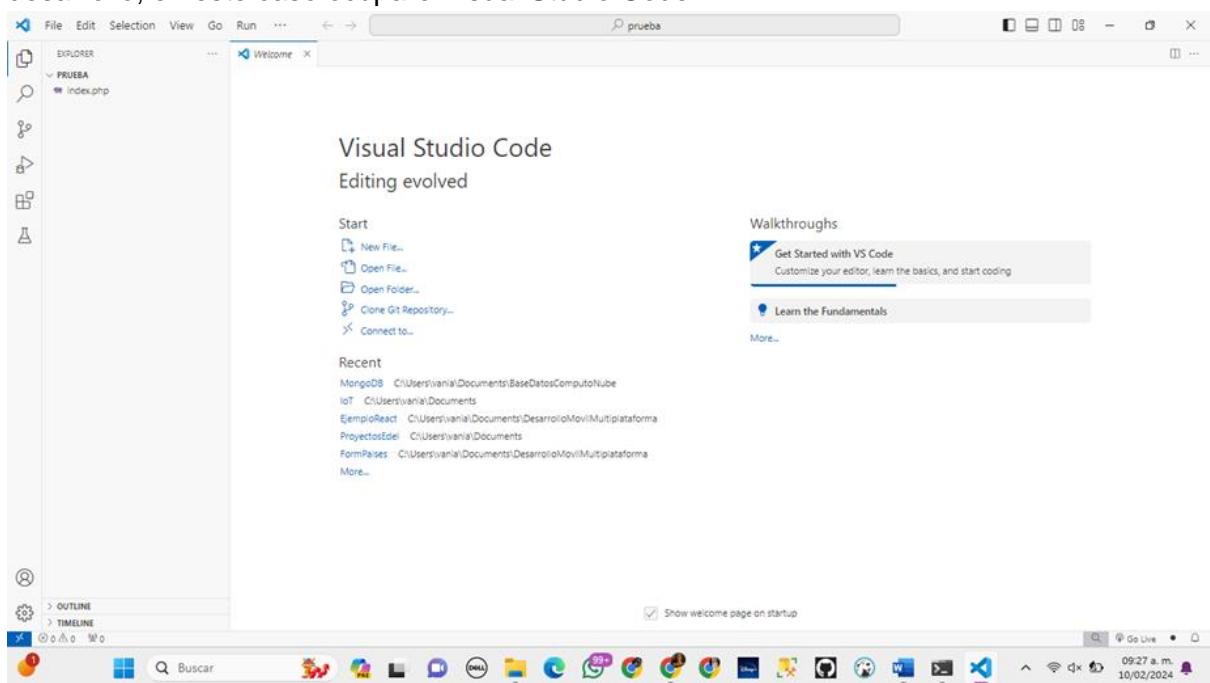
Nos vamos a la carpeta de www que es aquí donde se ejecutarán nuestros proyectos de php.

Nombre	Fecha de modificación	Tipo	Tamaño
index	07/02/2024 03:44 p. m.	Archivo de origen ...	1 KB

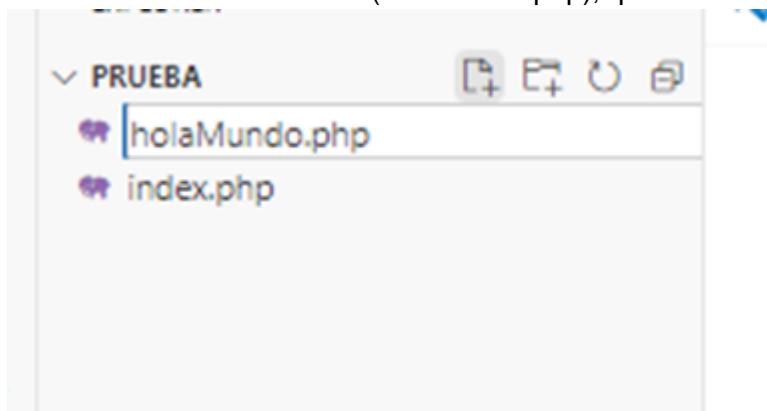
Veremos que esta carpeta ya tiene un index.php por default.



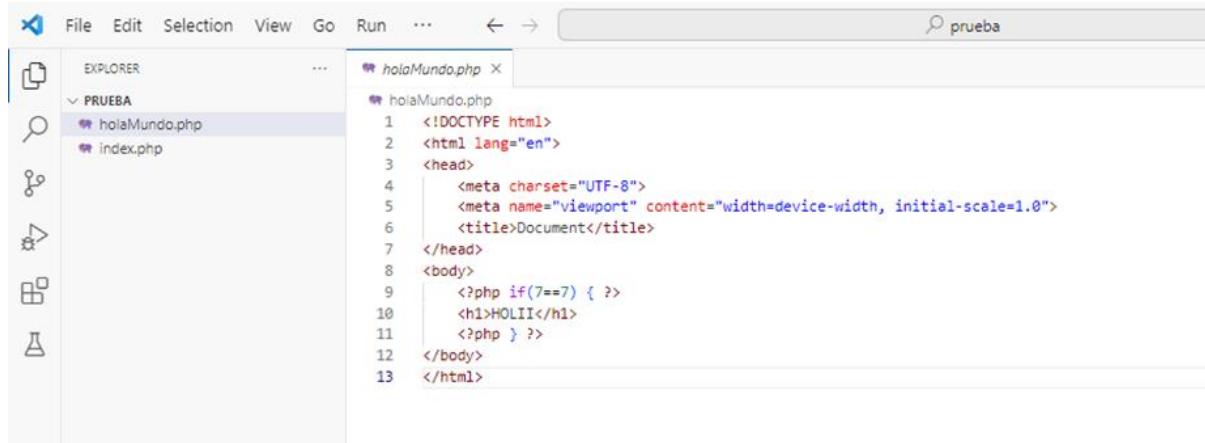
Haremos una nueva carpeta que es donde haremos una pequeña prueba php.  
Metemos el index.php en la nueva carpeta que hicimos y la abrimos en nuestro entorno de desarrollo, en este caso ocuparé Visual Studio Code.



Hacemos un nuevo archivo (holaMundo.php), que es solo un Hola Mundo.



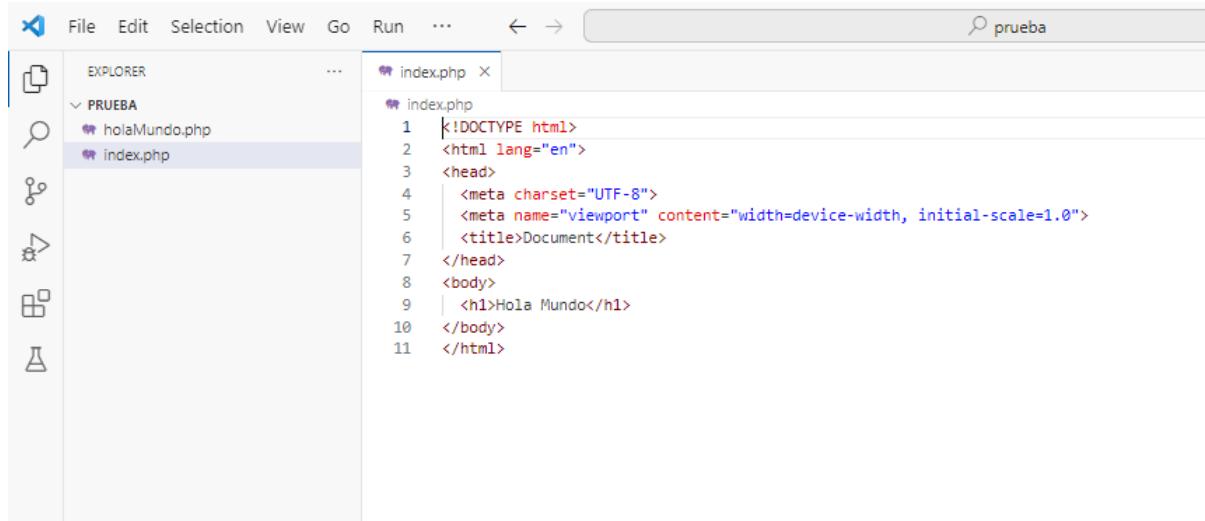
En este solo se va hacer una condición que será true para poder visualizar un "HOLII".



The screenshot shows a code editor interface with a toolbar at the top. The left sidebar is labeled 'EXPLORER' and shows a folder named 'PRUEBA' containing two files: 'holaMundo.php' and 'index.php'. The right pane displays the content of 'holaMundo.php'. The code is as follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<?php if(7==7) { ?>
<h1>HOLII</h1>
<?php } ?>
</body>
</html>
```

El index también lo modificaremos de la siguiente manera para que podamos comprar algo.



The screenshot shows a code editor interface with a toolbar at the top. The left sidebar is labeled 'EXPLORER' and shows a folder named 'PRUEBA' containing two files: 'holaMundo.php' and 'index.php'. The right pane displays the content of 'index.php'. The code is as follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<h1>Hola Mundo</h1>
</body>
</html>
```

Para correrlo debemos verificar que los servicios de php, apache estén activos por medio de Laragon. Después vamos a nuestro navegador favorito y nos vamos al localhost, al hacer eso, nos abrirá la siguiente ventana:



Le damos clic en donde dice prueba, esta es la carpeta que anteriormente se hizo, si hacemos más carpetas ahí se verán reflejadas, pero como tenemos una, solo nos muestra esa, al darle clic nos da la siguiente ventana:



Hola Mundo

Este es el index.php que realizamos, esto porque nuestro servidor de Apache levanta por default el index, si queremos entrar a otro archivo, en este caso nuestro holaMundo, necesitamos poner el nombre junto con su extensión.



HOLII

Listo, este es nuestro otro archivo.

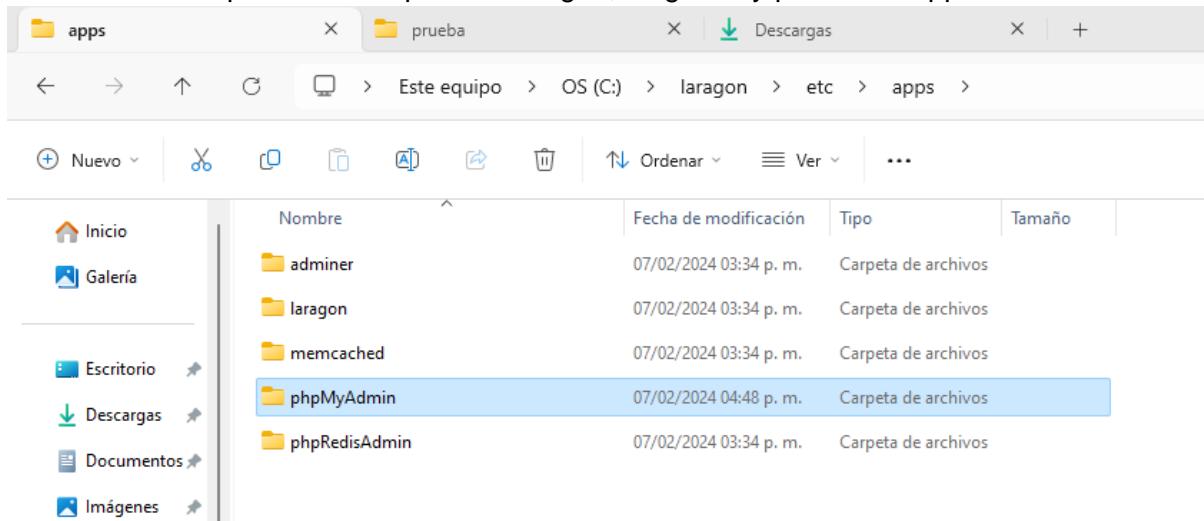
Ya tenemos configurado php, apache.

Ahora para las bases de datos, podemos descargar phpMyAdmin, es una interfaz gráfica para manipular las bases de datos de MySQL.

Para la configuración de phpMyAdmin, solo ocupamos descargarlo desde la página oficial, ya que tenemos nuestro zip:

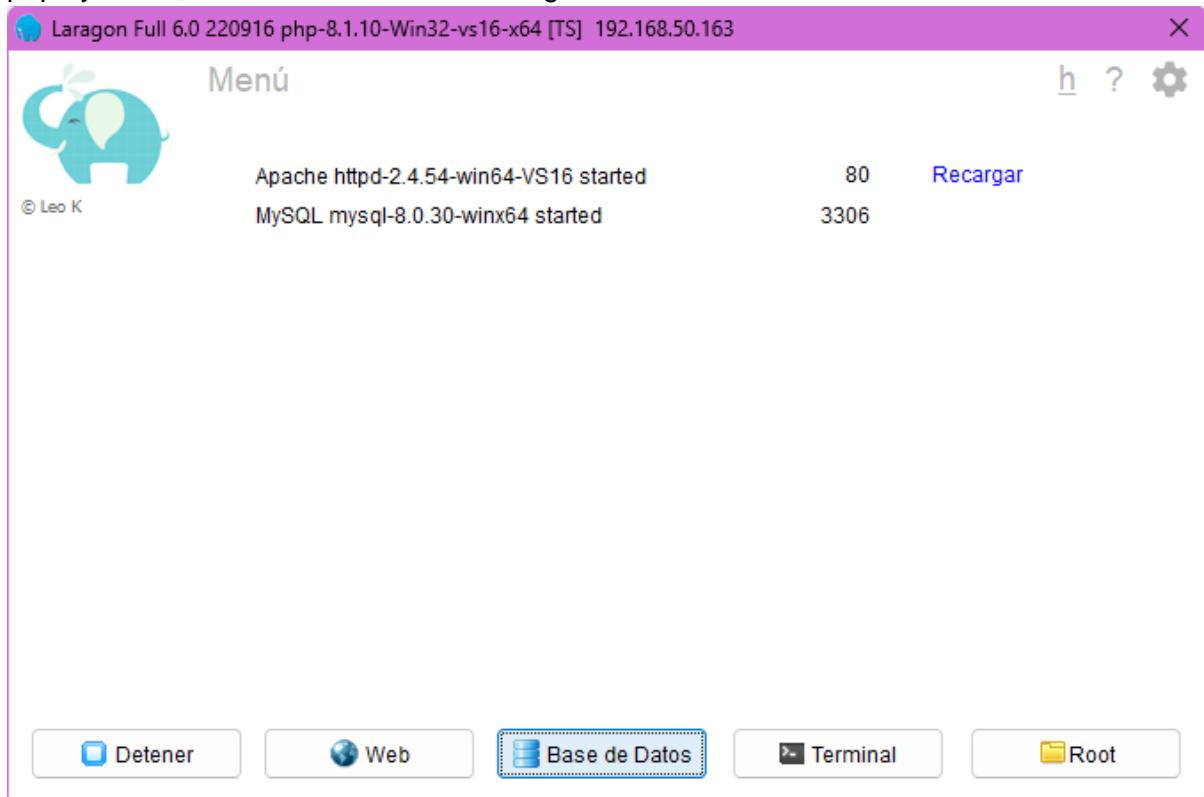
Nombre	Fecha de modificación	Tipo	Tamaño
phpMyAdmin-5.2.1-all-languages	04/02/2024 10:10 p. m.	Carpeta comprimi...	14,743 KB

Extraemos la carpeta en la carpeta de laragon, luego etc y por último apps:



Nombre	Fecha de modificación	Tipo	Tamaño
adminer	07/02/2024 03:34 p. m.	Carpeta de archivos	
laragon	07/02/2024 03:34 p. m.	Carpeta de archivos	
memcached	07/02/2024 03:34 p. m.	Carpeta de archivos	
phpMyAdmin	07/02/2024 04:48 p. m.	Carpeta de archivos	
phpRedisAdmin	07/02/2024 03:34 p. m.	Carpeta de archivos	

Y con eso ya lo tenemos, ya podemos entrar a nuestra base de datos con una interfaz gráfica, este paso solo es si queremos, no es necesario. Sin embargo, para entrar ya a phpMyAdmin, en nuestro servidor de laragon damos clic en Base de Datos:



Laragon Full 6.0 220916 php-8.1.10-Win32-vs16-x64 [TS] 192.168.50.163

Menú

Apache httpd-2.4.54-win64-VS16 started 80 Recargar

MySQL mysql-8.0.30-winx64 started 3306

Detener Web Base de Datos Terminal Root

Y nos abrirá una pestaña en nuestro navegador, donde nos pide iniciar sesión, iniciamos

solo con el usuario root y listo.

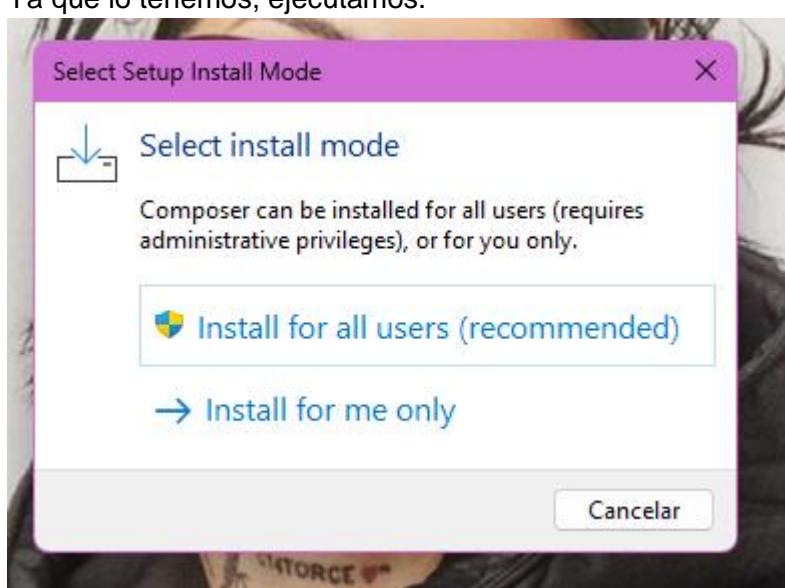
The screenshot shows the phpMyAdmin interface running on a local host. On the left, there's a sidebar with database navigation. The main area has several tabs: 'Configuraciones generales' (General configurations), 'Configuraciones de apariencia' (Appearance configurations), 'Servidor de base de datos' (Database server information), 'Servidor web' (Web server information), and 'phpMyAdmin' (About page). The 'Configuraciones generales' tab shows a dropdown for 'Cambiando de contraseña' (Changing password) set to 'utf8mb4\_unicode\_ci'. The 'Servidor de base de datos' tab lists MySQL server details like version 8.0.30 and user root@localhost. The 'Servidor web' tab shows Apache 2.4.54 and PHP 8.1.10. The 'phpMyAdmin' tab provides links to the About page, Documentation, and Support.

Ahora para descargar CodeIgniter.

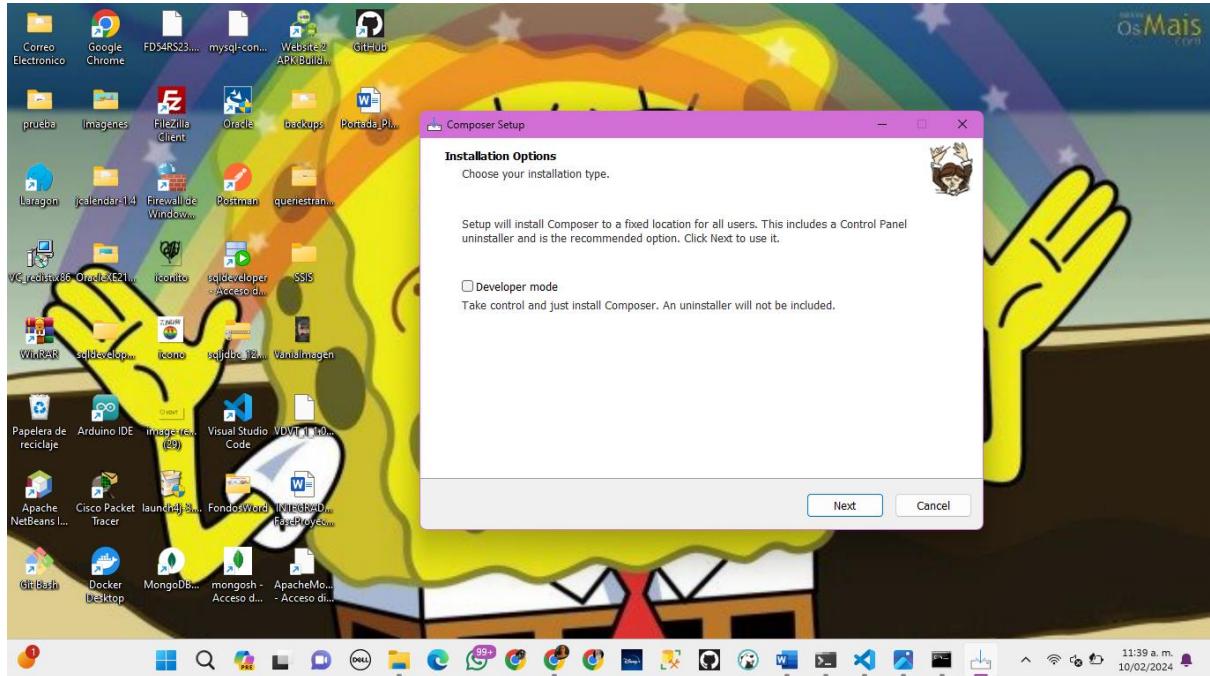
Necesitamos la ayuda de COMPOSER, para descargarlo, nos iremos a la página oficial de composer: <https://getcomposer.org/> y damos clic en descargar, descargamos el ejecutable para windows:

The screenshot shows the Composer website at getcomposer.org/download/. It features a navigation bar with Home, Getting Started, Download, Documentation, and Browse Packages. Below the navigation is a heading 'Download Composer' with a note about the latest version v2.7.1. A 'Windows Installer' section explains the installer requires PHP and provides a link to download the installer. A 'Command Line Installation' section also includes a download link. The page is clean with a white background and standard web fonts.

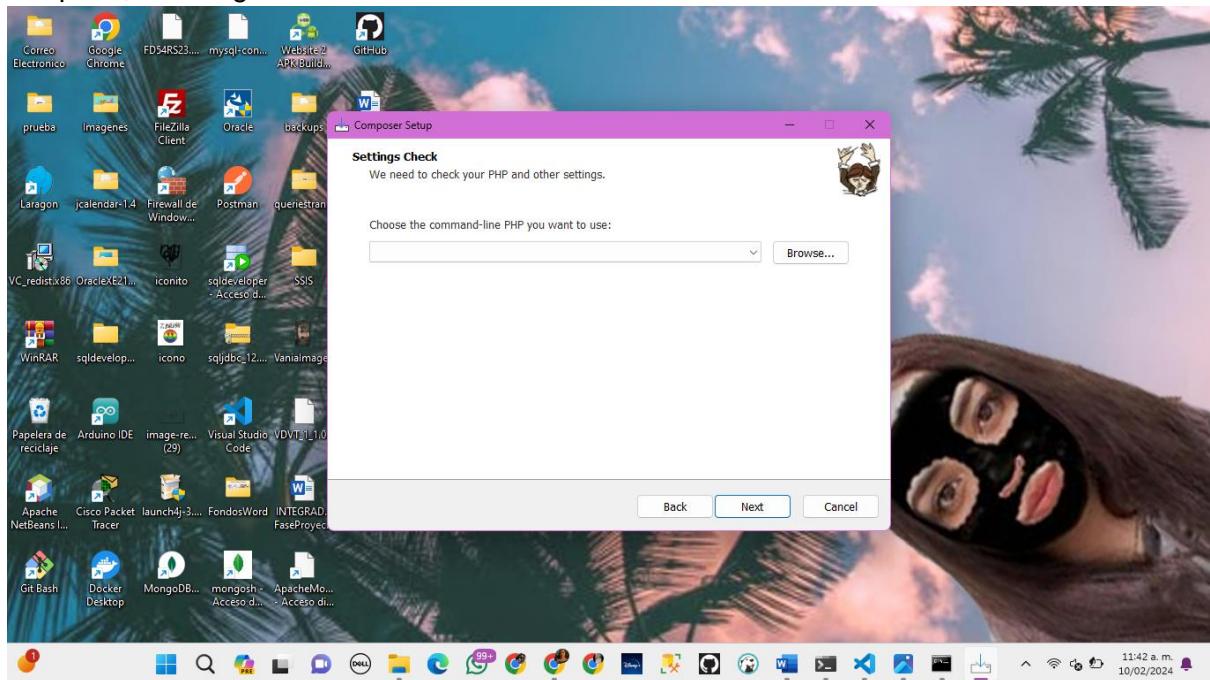
Ya que lo tenemos, ejecutamos.



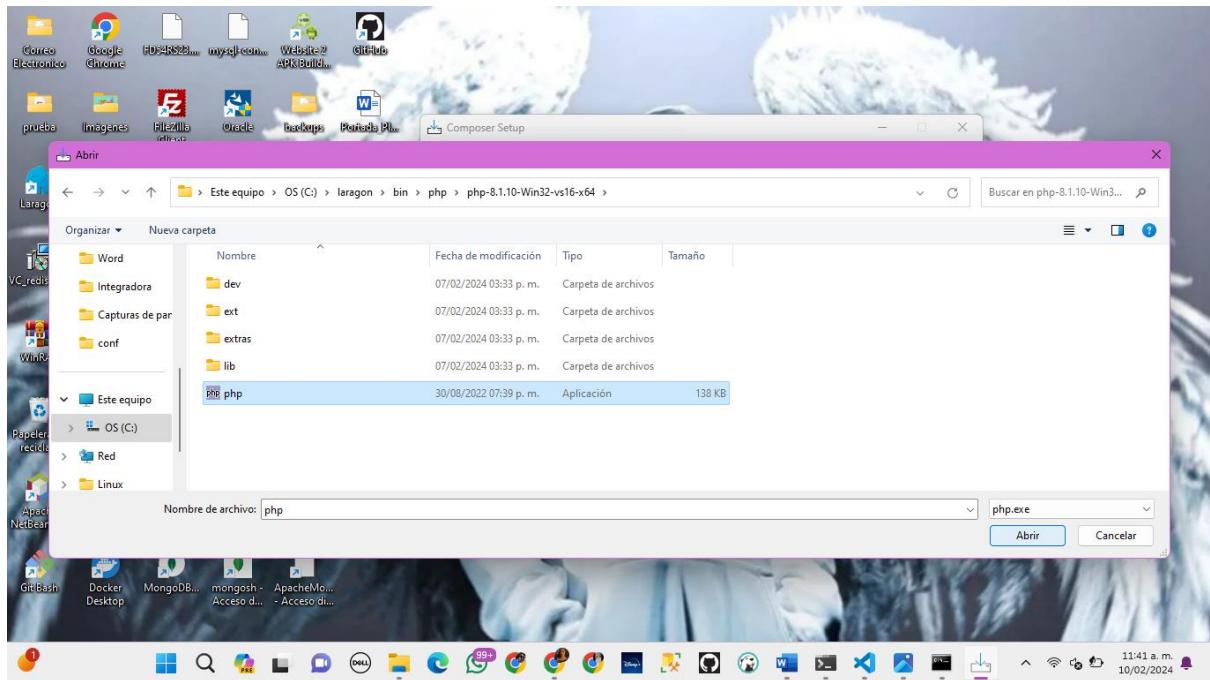
Le damos clic en instalar para todos los usuarios o podemos dejarlo para solo el usuario que estamos ocupando, según tus necesidades, en este caso le daré en: todo los usuarios. En esta pestaña solo le damos next.



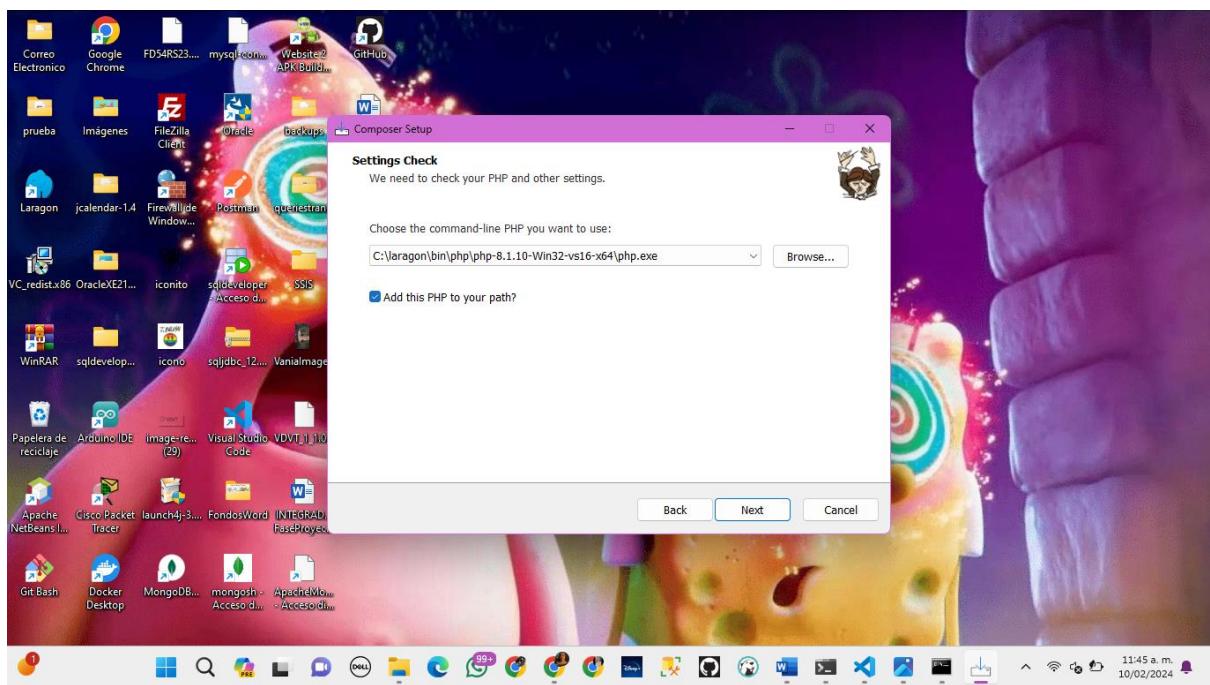
Después, en la siguiente ventana le damos en browse...



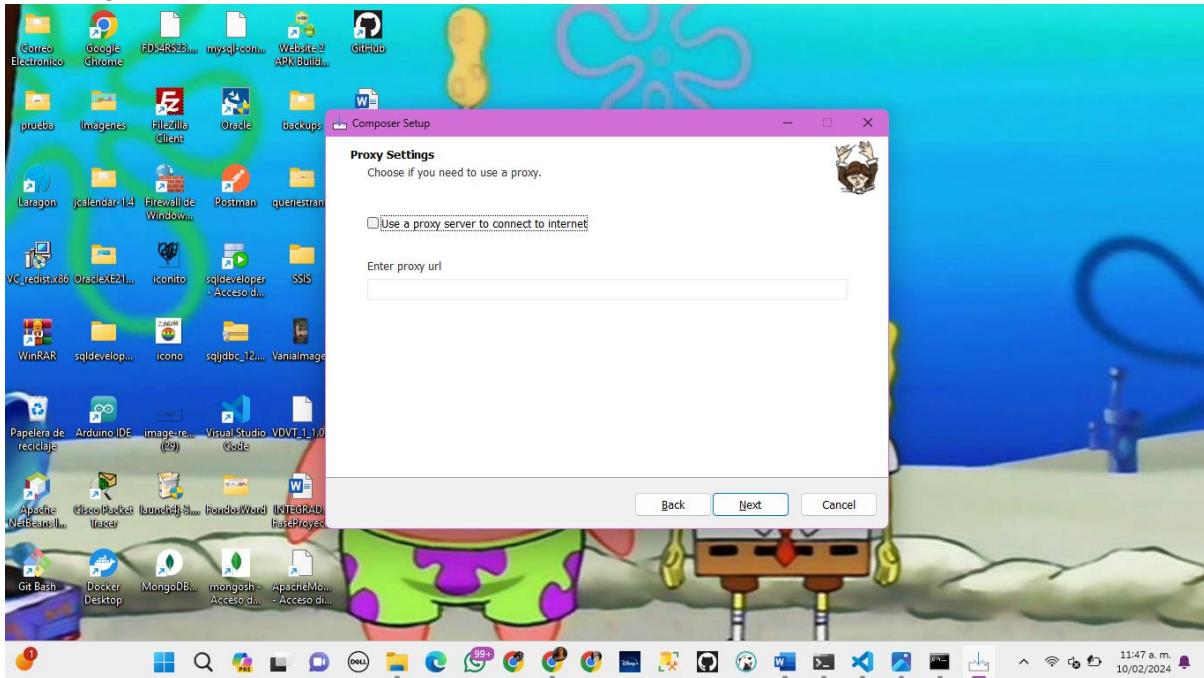
Y buscamos en C la carpeta de laragon, luego bin, después php, nos vamos a la única carpeta de php que hay y seleccionamos php.exec que es la aplicación o el ejecutable de php, y le damos abrir.



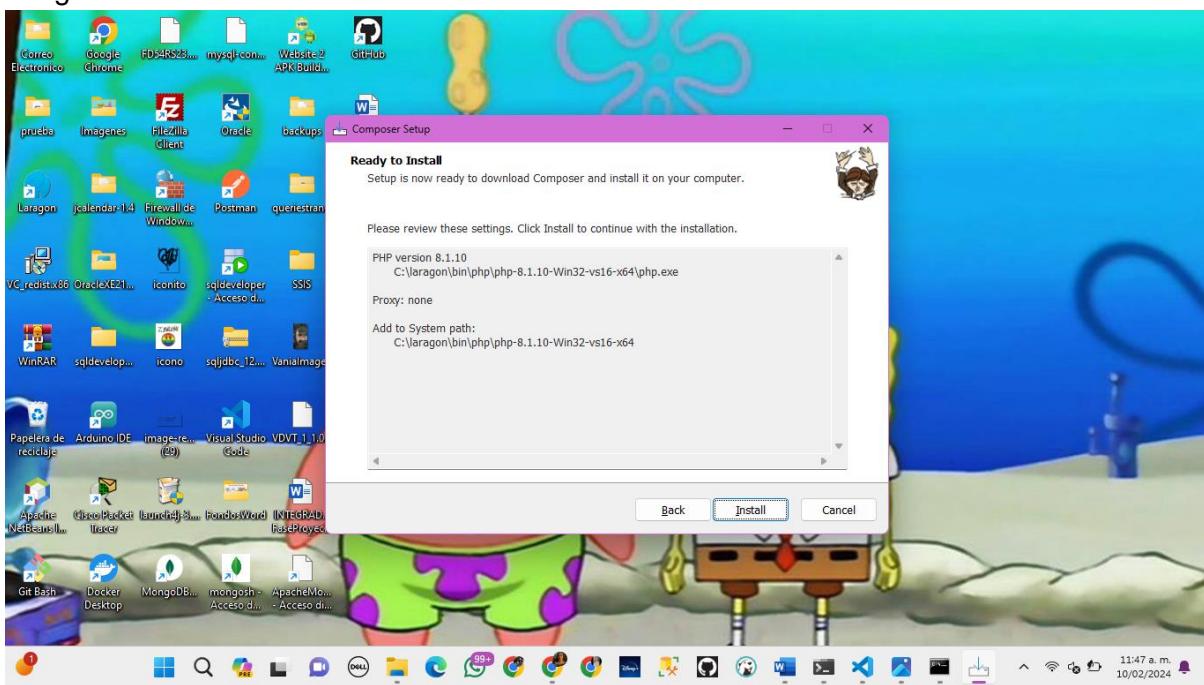
Ya que lo tenemos seleccionamos la opción que nos da y le damos next, dejamos que se instale.



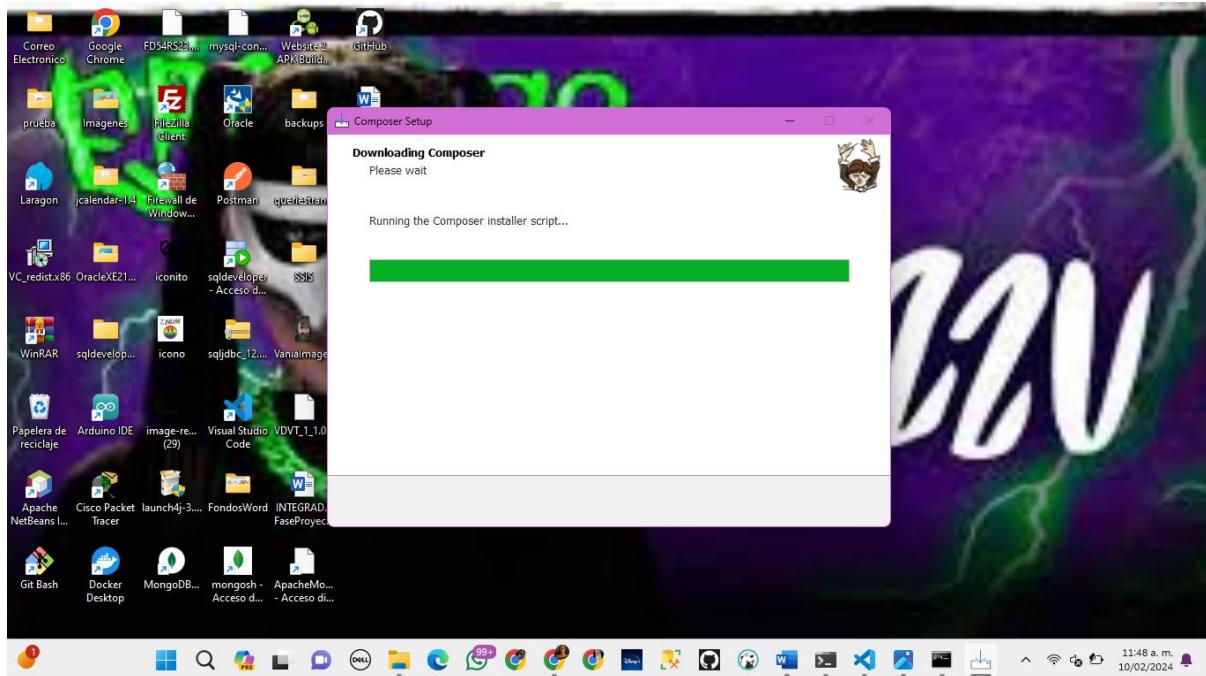
En la siguiente pantalla solo le damos next.



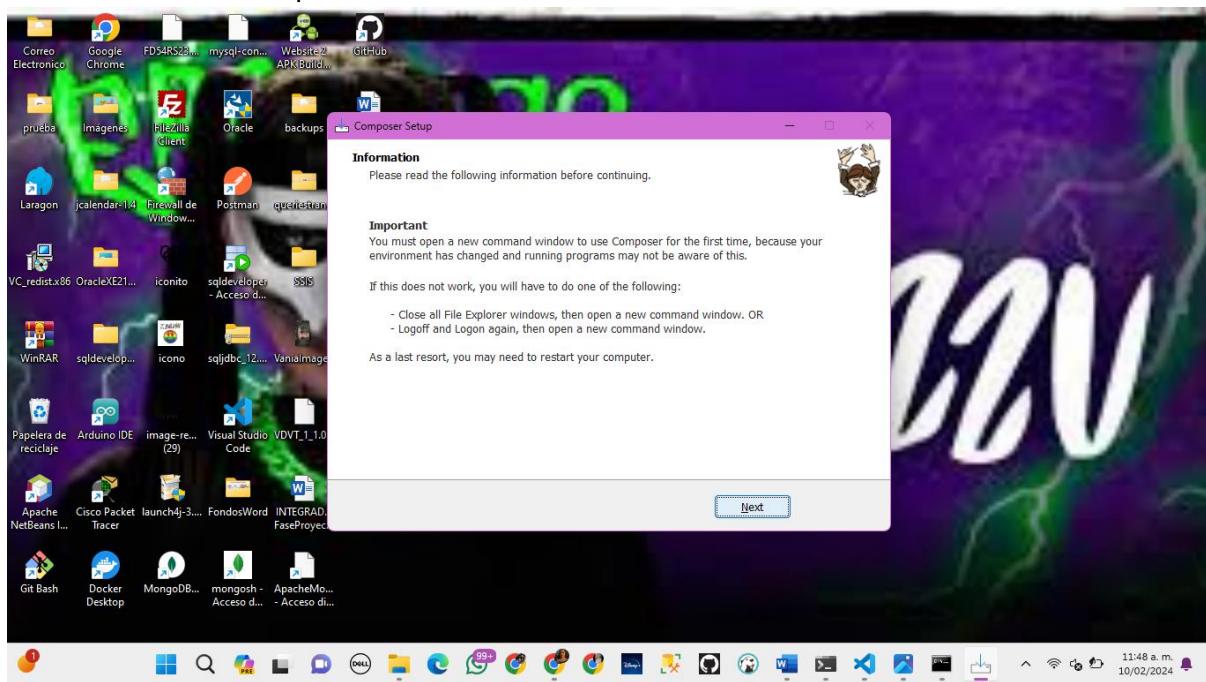
Luego le damos Install.



Esperamos que se instale y listo.



Le damos next a esta pantalla:





Bienvenido a CodeIgniter

Encendedor de código

Conversar Contribuir Aprender

CODEIGNITER 4

CodeIgniter 4 es la última versión del marco, diseñada para usarse con PHP 7.4+ (incluido 8.2). El lanzamiento inicial fue el 24 de febrero de 2020. La versión actual es la v4.4.5. 'Podrías' descargar esta versión del framework usando el botón a continuación, pero te recomendamos que consultes la [sección Instalación](#) de la Guía del usuario para usar Composer :)

Descargar Conversar Fuentes Traducciones Guía del usuario

Bienvenido a CodeIgniter

Encendedor de código

Conversar Contribuir Aprender

CODEIGNITER 3

CodeIgniter 3 es la versión heredada del marco, diseñada para usarse con PHP 5.6+. Esta versión está en mantenimiento y recibe principalmente actualizaciones de seguridad, y la

Le damos clic en descargar la última versión y esperamos a que se descargue.

Historial de descargas recientes

codeigniter4-framework-v4.4.5-0-gf5844cb.zip  
1,122 KB • Listo

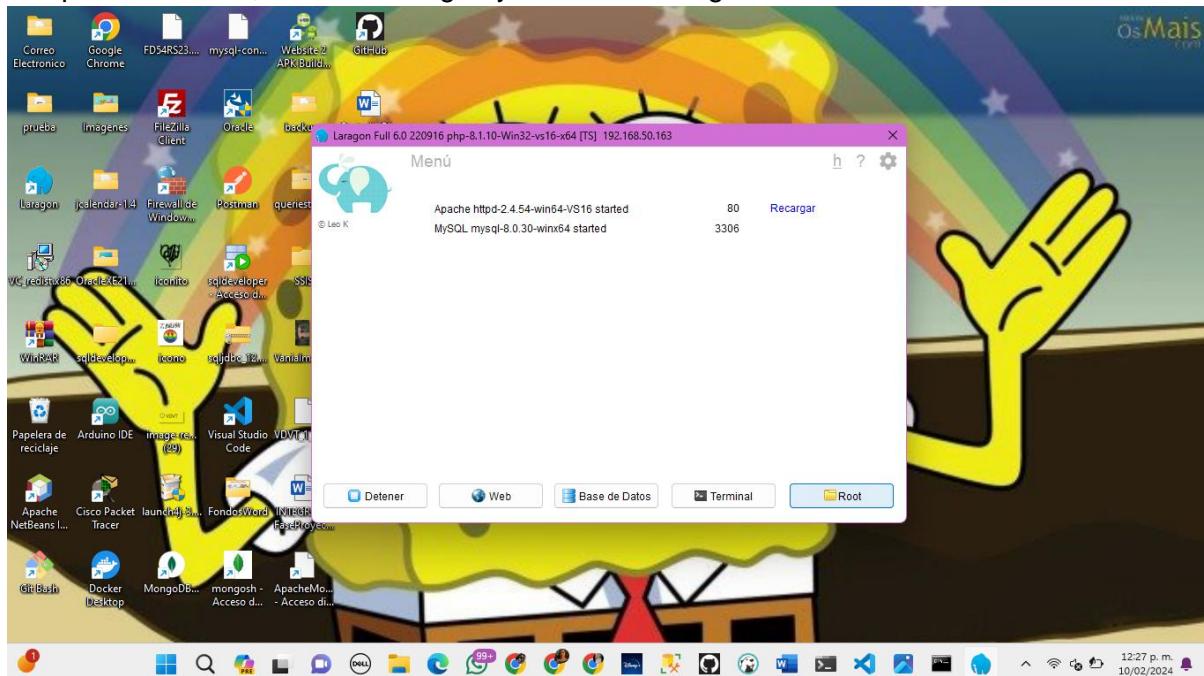
Abrimos nuestros archivos y vamos a copiar la carpeta zip que se descargó con el framework.

↑ ⌄ Descargas >

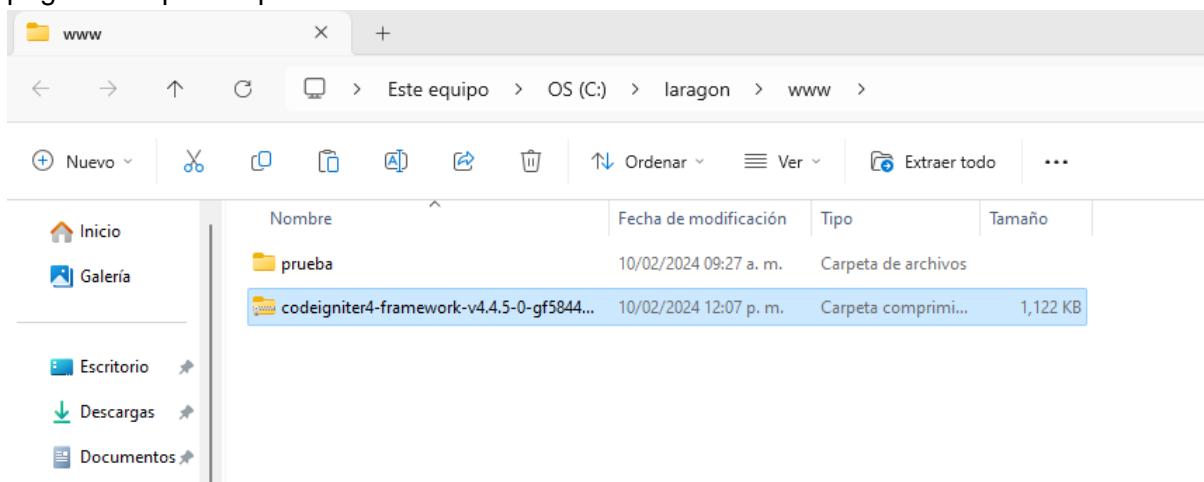
Nombre Fecha de modificación Tipo Tamaño

codeigniter4-framework-v4.4.5-0-gf5844... 10/02/2024 12:07 p. m. Carpeta comprimi... 1,122 KB

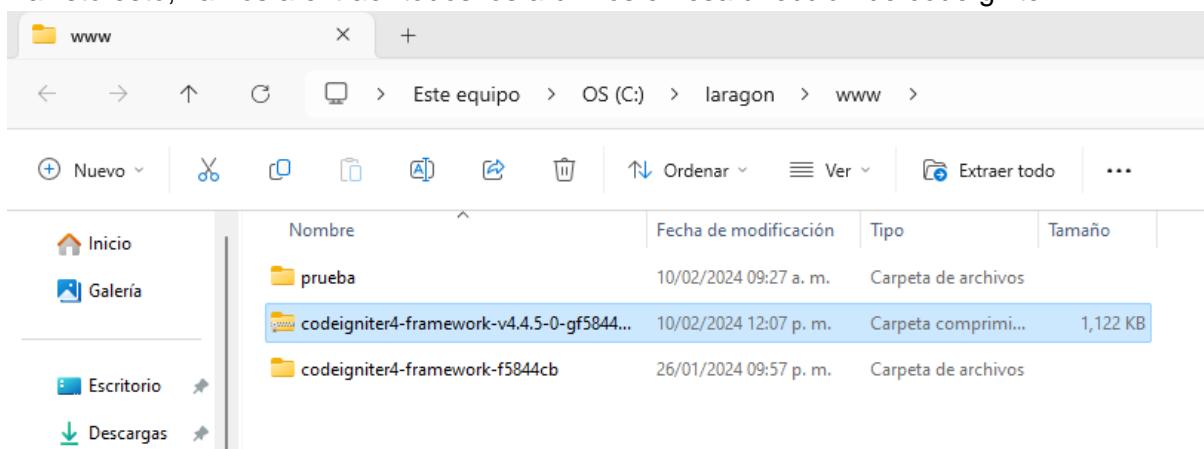
Ya que lo tenemos, abrimos Laragon y haremos las siguientes cosas.



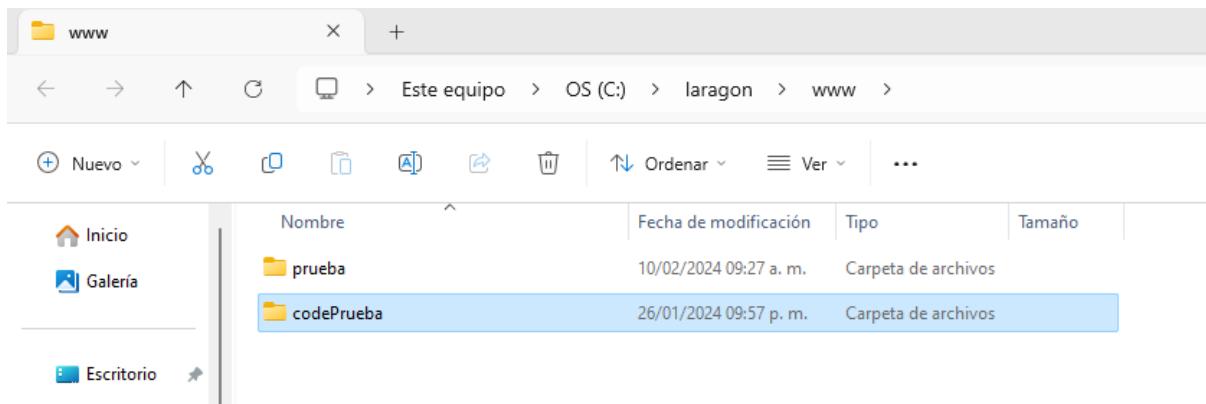
Damos clic en la carpeta de root y nos mandará justo a la dirección en donde vamos a pegar la carpeta copiada de nuestro framework.



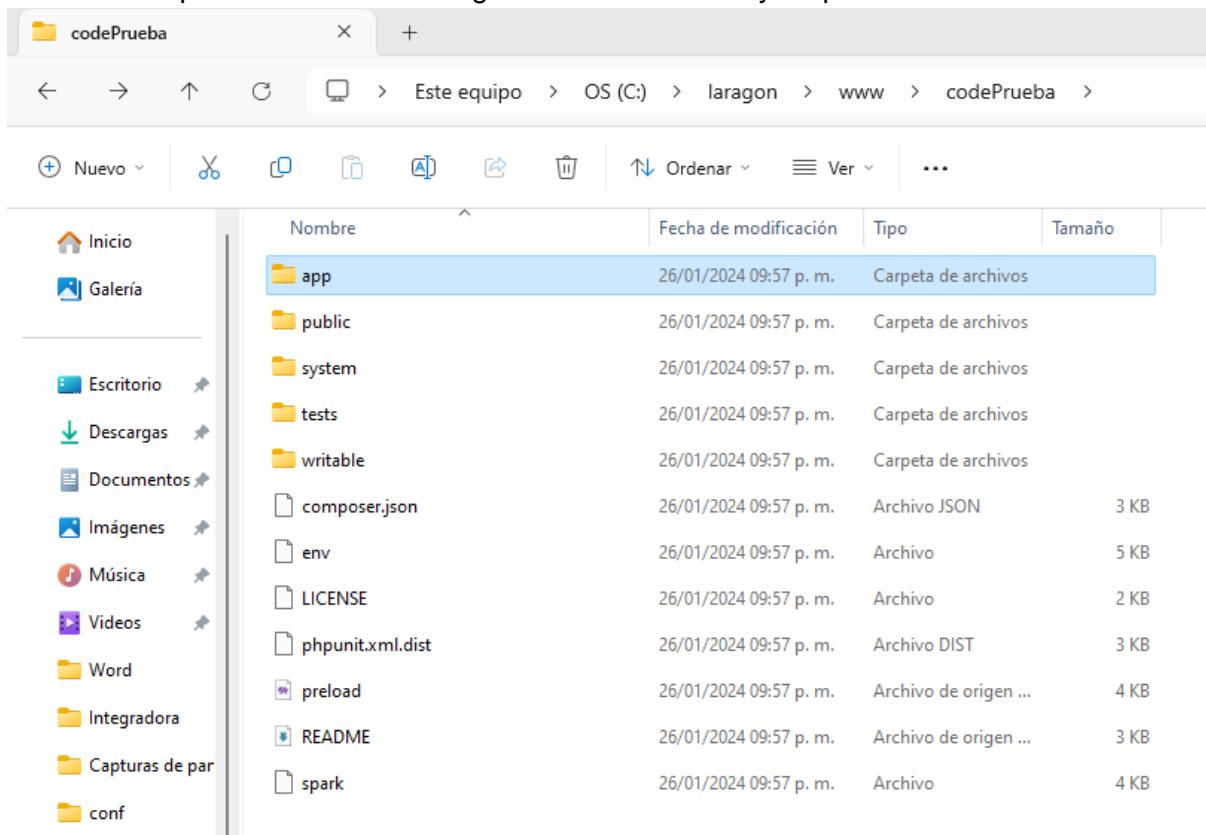
Ya listo esto, vamos a extraer todos los archivos en esa dirección de codeigniter.



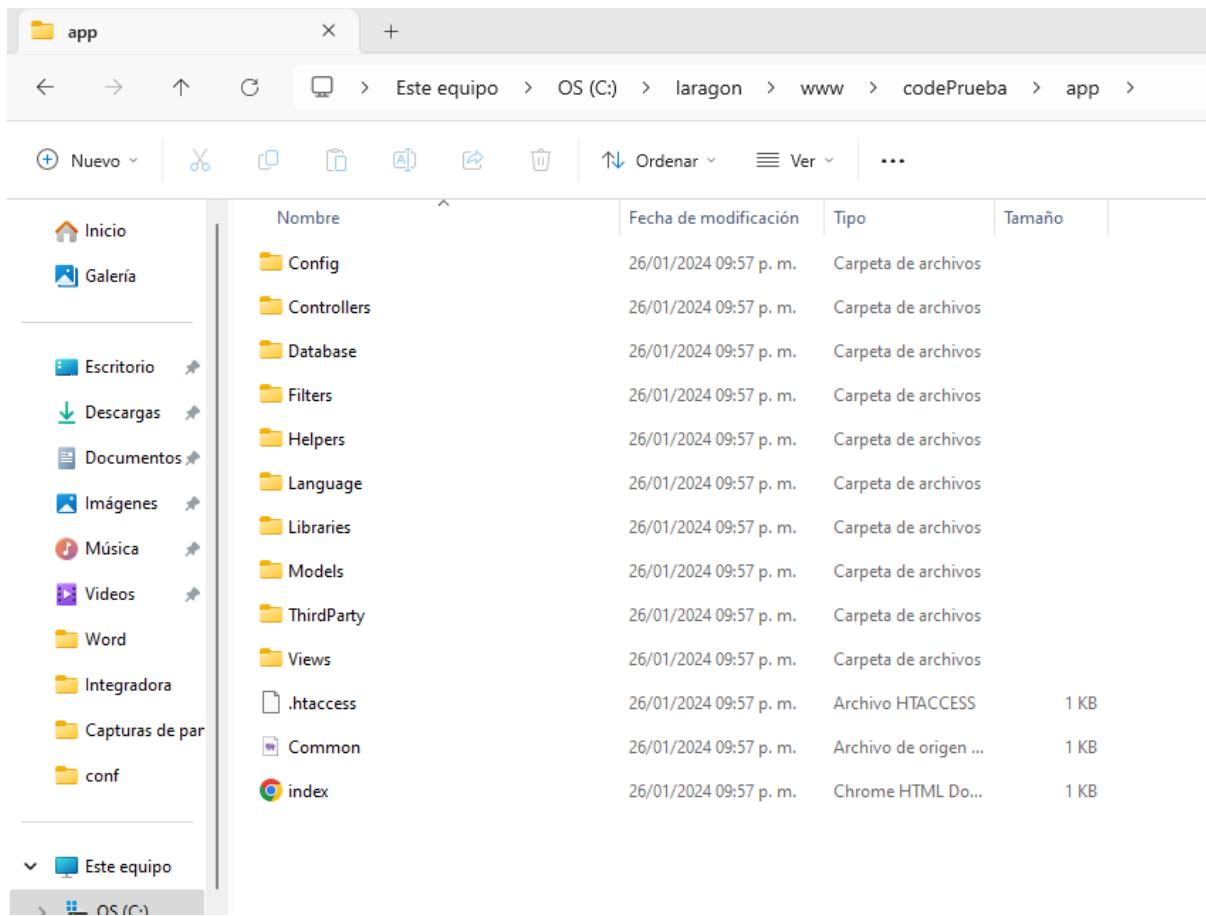
Ahora eliminamos el zip y renombramos nuestra carpeta con un nombre representativo a nuestro proyecto que desarrollaremos.



Al abrir la carpeta nos saldrán los siguientes documentos y carpetas:



En donde en la carpeta app estarán las carpetas en donde estaremos trabajando con el modelo MVC que es el modelo con el que trabaja este framework.



Las carpetas son Controllers, Views y Models.

Listo ya tenemos listo nuestro framework, nuestro servidor, nuestra base de datos y php instalado y configurado correctamente para trabajar con ellos en conjunto.

## Cumplimiento de PSR

### PSR-1: Estándar de codificación básica

Esta recomendación cubre estándares básicos de clases, métodos y nombres de archivos. Nuestra guía de estilo cumple con PSR-1 y le agrega sus propios requisitos.

### PSR-12: estilo de codificación extendido

Nuestra guía de estilo sigue las recomendaciones además de un conjunto de nuestras propias convenciones de estilo.

### PSR-3: Interfaz de registrador

El registrador de CodeIgniter implementa todas las interfaces proporcionadas por este PSR.

### PSR-4: Estándar de carga automática

Este PSR proporciona un método para organizar archivos y espacios de nombres para permitir un método estándar de carga automática de clases. Nuestro Autoloader cumple con las recomendaciones PSR-4.

## PSR-6: Interfaces de almacenamiento en caché PSR-16: Interfaz SimpleCache

Si bien los componentes de la caché del marco no se adhieren a PSR-6 o PSR-16, la organización CodeIgniter4 ofrece un conjunto separado de adaptadores como módulo complementario. Se recomienda que los proyectos utilicen los controladores de caché nativos directamente, ya que los adaptadores solo están diseñados para ser compatibles con bibliotecas de terceros. Para obtener más información, visite el repositorio de caché de CodeIgniter4.

## PSR-7: Interfaz de mensajes HTTP

Este PSR estandariza una forma de representar las interacciones HTTP. Si bien muchos de los conceptos se convirtieron en parte de nuestra capa HTTP, CodeIgniter no se esfuerza por lograr compatibilidad con esta recomendación.

# Hosting

En esta investigación encontramos diversas posibles alternativas para la publicación de nuestro proyecto una vez que esté finalizado, buscamos servicios de alojamiento web que sea compatible con php, MySQL y que sea gratuito, aplicando estos filtros de búsqueda hemos encontrado los siguientes posibles hosting del cual en la práctica elegiremos uno, el que nos parezca más indicado.

**InfinityFree:** InfinityFree es un proveedor de alojamiento web gratuito que ofrece un amplio conjunto de características, incluyendo soporte para PHP, bases de datos MySQL, y un panel de control fácil de usar.

**000webhost:** 000webhost es otro proveedor de alojamiento web gratuito que ofrece soporte para PHP, MySQL y una amplia gama de funciones, incluyendo un constructor de sitios web y un panel de control intuitivo.

**FreeHosting.com:** Ofrece alojamiento web gratuito con soporte para PHP, bases de datos MySQL y un panel de control fácil de usar. También proporciona soporte para tecnologías web estándar como HTML, CSS y JavaScript.

**AwardSpace:** Este proveedor de alojamiento web gratuito ofrece soporte para PHP, bases de datos MySQL, y proporciona un creador de sitios web fácil de usar. También es compatible con tecnologías web estándar como HTML, CSS y JavaScript.

**Byet.Host:** Ofrece alojamiento web gratuito con soporte para PHP, bases de datos MySQL, y una amplia gama de características adicionales. También es compatible con HTML, CSS, Bootstrap y JavaScript.

**x10Hosting:** Proporciona alojamiento web gratuito con soporte para PHP, bases de datos MySQL, y una variedad de características adicionales. También es compatible con tecnologías web estándar como HTML, CSS y JavaScript.

**5GBFree:** Ofrece alojamiento web gratuito con soporte para PHP, bases de datos MySQL, y un generador de sitios web fácil de usar. También es compatible con HTML, CSS, Bootstrap y JavaScript.

# Pasos a seguir para la elección de un buen hosting:

Lo primero es ir revisando la reputación de las empresas de servicio de hosting, para eso existen sitios en internet donde se puede ver eso, también es importante ver las valoraciones de sus usuarios, esto podemos lograrlo buscando opiniones de clientes reales, un sitio muy confiable para eso es [trustpilot.com/](https://es.trustpilot.com/)

The screenshot shows the Trustpilot website homepage. At the top, there's a navigation bar with links for 'Categorías', 'Blog', 'Iniciar sesión', and 'Para empresas'. Below the header, a large green banner features the text 'Lee opiniones. Escribe reseñas. Encuentra empresas de confianza.' in white. To the right of the text is a collage of various user reviews and photos. A search bar with the placeholder 'Empresa o categoría' and a blue 'Buscar' button is positioned below the banner. At the bottom of the page, a cookie consent banner is displayed, asking if the user wants to use cookies for personalization and advertising, with options for 'Configuración de cookies' and 'Entendido'.

En esta página podemos buscar en la categoría de hosting

The screenshot shows the Trustpilot website homepage after a search for 'hosting'. The search results are displayed on the left side of the page. The first result is 'IONOS | ionos.de' with a rating of 4.4 stars. Below it are 'ZAP-Hosting' (4.2 stars), 'Miss Hosting' (4.8 stars), 'Hosting Studio' (4.6 stars), and 'Hosting Power' (4.6 stars). On the right side of the page, there are several categories represented by icons and text: 'Concesionario de autos', 'Tienda de muebles', 'Electrónica y tecnología', 'Servicio de fitness y nutrición', 'Agentes inmobiliarios', and 'Compañía de seguros'. A 'Ver todas' button is located near the bottom of the category section. The Trustpilot logo is at the top left, and the navigation bar is at the top right.

Normalmente los hosting cuentan con diferentes planes, ahí viene información como el espacio en disco que ofrecen, lo cual es una característica importante de revisar y elegir lo que se adapte a nuestras necesidades.



Otro tema importante en el mundo de los hosting son los **INODOS**

En el contexto del hosting, especialmente en el entorno de servidores y redes, "inodo" se refiere a una estructura de datos utilizada para mantener información sobre un archivo o directorio en un sistema de archivos.

Los inodos son importantes en el contexto del hosting para la publicación de páginas web por varias razones:

- Gestión de archivos y directorios: En un servidor web, los inodos ayudan a gestionar todos los archivos y directorios que componen un sitio web. Cada archivo (como páginas HTML, imágenes, scripts, etc.) y cada directorio (carpetas que contienen estos archivos) están representados por inodos en el sistema de archivos del servidor.
- Control de acceso y permisos: Los inodos almacenan información sobre los permisos de acceso de cada archivo y directorio. Esto es crucial para la seguridad del sitio web, ya que permite controlar quién puede ver, modificar o eliminar los archivos.
- Rendimiento del servidor: Los sistemas de archivos eficientes utilizan inodos de manera efectiva para organizar y acceder a los archivos de manera rápida y eficiente. Un número adecuado de inodos y una gestión eficiente de los mismos pueden contribuir al rendimiento general del servidor web.
- Límite de archivos: En algunos casos, el número de inodos disponibles en el sistema de archivos puede limitar la cantidad de archivos que se pueden almacenar en el servidor. Esto puede ser importante al seleccionar un plan de hosting, ya que algunos proveedores pueden tener límites en la cantidad de inodos que puedes utilizar.

En resumen, los inodos son fundamentales para la gestión, seguridad y rendimiento de los archivos en un servidor web, lo que los convierte en un elemento crucial en el proceso de publicación y administración de páginas web en el entorno de hosting.

Es importante ver la cantidad de inodos que ofrece un hosting, normalmente una buena cantidad de inodos en un hosting compartido son 200,000 un hosting que te ofrezca por debajo de los 100,000 inodos no es tan buena opción.

También hay que revisar la **Transferencia de datos o ancho de banda** el cual es el límite del consumo de datos que se transmiten cada vez que un usuario accede a nuestro sitio web. Cada vez que un usuario accede a nuestro sitio web y cada vez que cambia de sección o página se consume parte de este ancho de banda, es muy común que los proveedores tengan transferencia de datos ilimitados en sus planes de hosting, aunque es importante revisar eso ya que podemos encontrarnos con empresas que SÍ tengan un límite, es importante elegir uno que no tenga ese límite.

✓ Copias de seguridad <b>semanales</b>	?	✓ Copias de seguridad <b>diarias</b> (valor: MX\$ 468.00)	?	✓ Copias de...
✓ SSL <b>ilimitado</b> gratis	?	✓ SSL <b>ilimitado</b> gratis	?	✓ SSL <b>ilimit...</b>
✓ Ancho de banda <b>ilimitado</b>	?	✓ Ancho de banda <b>ilimitado</b>	?	✓ Ancho de...
✓ Email <b>gratis</b>	?	✓ Email <b>gratis</b>	?	✓ Email <b>gra...</b>
✓ Dominio <b>gratis</b> (MX\$ 176.99)	?			

Otro punto importante a revisar al elegir un hosting es la **compatibilidad** con tus herramientas, es decir revisar que ese servicio de hosting que te interesa sea compatible con la tecnología que vas a usar para crear tu sitio web, en este caso PHP, codeigniter, html, css, bootstrap, JavaScript, y la base de datos de MySql.

Otro punto es el **certificado SSL** es lo que te permite tener el candado a un lado de tu dirección web



Este es obligatorio solo para sitios web que vendan en línea, sin embargo es recomendable elegir un hosting que ofrezca este servicio gratis para agregarlo a todos los sitios web que agregues.

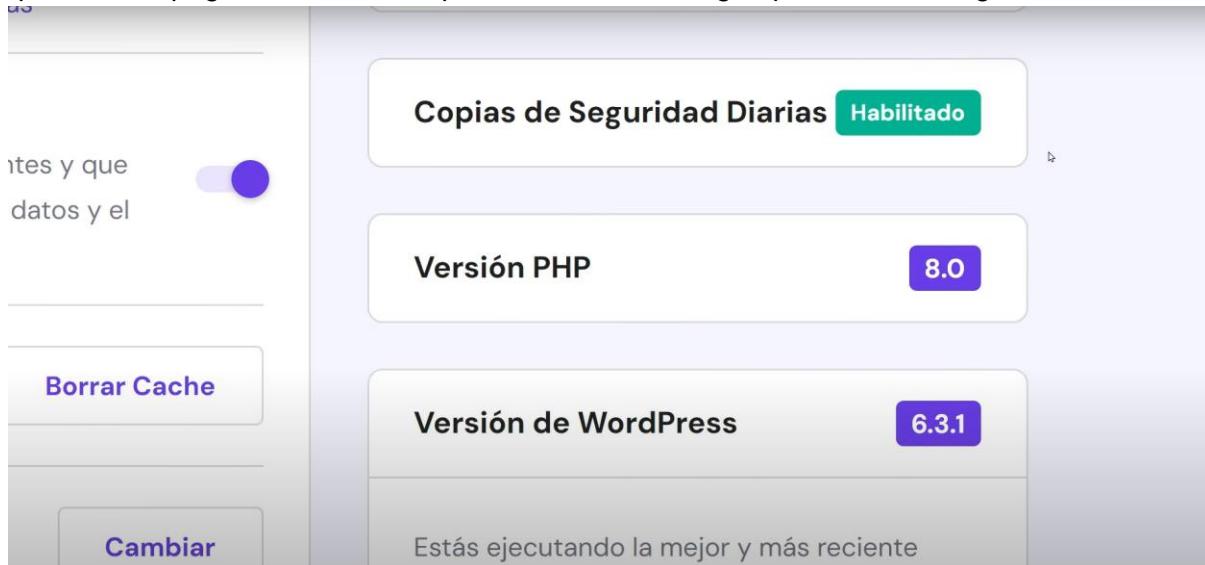
Otro punto es el **Dominio gratis**, es común que algunos proveedores te regalen el dominio.com gratis para el primer año, hay que aprovecharlo ya que un dominio puede rondar los 10 dólares. Y existen servicios de hosting que te quieren cobrar el dominio aparte del hosting y a precios altos.

Otro punto es el correo electrónico personalizado, las cuentas de correo electrónico profesional son las cuentas de correo electrónico personalizadas que llevan el nombre de tu dominio, por ejemplo:

**hola@markethax.com**

Es recomendable contratar con un proveedor que ofrezca por lo menos 10 cuentas de correo, de lo contrario sera muy costoso agregar un correo cada vez que necesitemos uno.

Otro punto importante son los **Respaldos**, en caso de algún incidente siempre es una buena idea contar con un respaldo de nuestro sitio web, es importante revisar que el proveedor que elijamos ofrezca respaldos diarios o por lo menos semanales para garantizar que nuestra página web sea recuperada en caso de algún problema de seguridad.



Es importante revisar que el hosting te permite alojar múltiples sitios web, hoy en dia este tipo de servicios son muy comunes.

Después de mencionar todas esas características hemos investigado más acerca de los hosting y encontramos uno que se adapta bien a nuestras necesidades, es económico y cumple con las características que debe de tener un buen hosting:

Hostinger es una empresa de alojamiento web que ofrece una variedad de servicios de hosting compartido, VPS (servidores privados virtuales) y alojamiento en la nube. Aquí hay algunas características destacadas de Hostinger:

Precio competitivo: Hostinger es conocido por ofrecer precios muy bajos, especialmente en sus planes de hosting compartido. Esto lo hace atractivo para personas que están comenzando con un presupuesto limitado.

Rendimiento sólido: A pesar de sus precios bajos, Hostinger ofrece un buen rendimiento en términos de velocidad y tiempo de actividad. Utilizan tecnologías como el almacenamiento SSD y la optimización de servidores para garantizar un buen rendimiento para sus usuarios.

Panel de control fácil de usar: Hostinger proporciona un panel de control personalizado que facilita la gestión de tu sitio web, correos electrónicos, bases de datos y otros aspectos relacionados con el hosting. Es intuitivo y fácil de navegar, lo que lo hace adecuado para usuarios principiantes.

Seguridad: Hostinger ofrece medidas de seguridad sólidas, como copias de seguridad diarias gratuitas, protección contra malware y firewall de aplicaciones web (WAF) en algunos de sus planes.

**Soporte técnico:** Ofrecen soporte técnico las 24 horas del día, los 7 días de la semana, a través de chat en vivo y ticket de soporte. Su equipo de soporte está capacitado para ayudar con una variedad de problemas relacionados con el hosting.

**Amplia gama de servicios:** Además del hosting compartido, Hostinger también ofrece servicios de VPS, alojamiento en la nube, registro de dominios y más, lo que permite a los usuarios escalar según sus necesidades.

Hostinger permite tener **HTTPS** en tus sitios web. HTTPS (Hypertext Transfer Protocol Secure) es una versión segura del protocolo HTTP que utiliza cifrado SSL/TLS para proteger la comunicación entre el navegador del usuario y el servidor web. Esto es fundamental para garantizar la seguridad y privacidad de los datos transmitidos entre el usuario y el sitio web.

Para habilitar HTTPS en tu sitio web alojado en Hostinger, generalmente puedes seguir estos pasos:

**Obtener un certificado SSL:** Hostinger ofrece certificados SSL gratuitos con la mayoría de sus planes de hosting. Puedes obtener un certificado SSL gratuito desde tu panel de control o comprar uno si necesitas características avanzadas.

**Instalar el certificado SSL:** Una vez que hayas obtenido el certificado SSL, puedes instalarlo en tu sitio web a través del panel de control de Hostinger. Este proceso generalmente implica generar una solicitud de certificado (CSR), validar la propiedad del dominio y luego instalar el certificado en tu servidor.

**Configurar tu sitio web para HTTPS:** Una vez que el certificado SSL esté instalado correctamente, necesitas configurar tu sitio web para que use HTTPS en lugar de HTTP. Esto generalmente implica actualizar los enlaces internos, asegurarse de que todas las imágenes, hojas de estilo y scripts se carguen a través de HTTPS y redireccionar el tráfico HTTP a HTTPS.

Hostinger proporciona guías detalladas y soporte técnico para ayudarte en este proceso. Una vez que HTTPS esté habilitado en tu sitio web, los visitantes verán el icono del candado en la barra de direcciones de su navegador, lo que indica que la conexión es segura y que sus datos están protegidos durante la transmisión.

### **¿Hostinger permite tener https?**

Sí, Hostinger permite tener HTTPS en tus sitios web. HTTPS (Hypertext Transfer Protocol Secure) es una versión segura del protocolo HTTP que utiliza cifrado SSL/TLS para proteger la comunicación entre el navegador del usuario y el servidor web. Esto es fundamental para garantizar la seguridad y privacidad de los datos transmitidos entre el usuario y el sitio web.

Para habilitar HTTPS en tu sitio web alojado en Hostinger, generalmente puedes seguir estos pasos:

Obtener un certificado SSL: Hostinger ofrece certificados SSL gratuitos con la mayoría de sus planes de hosting. Puedes obtener un certificado SSL gratuito desde tu panel de control o comprar uno si necesitas características avanzadas.

Instalar el certificado SSL: Una vez que hayas obtenido el certificado SSL, puedes instalarlo en tu sitio web a través del panel de control de Hostinger. Este proceso generalmente implica generar una solicitud de certificado (CSR), validar la propiedad del dominio y luego instalar el certificado en tu servidor.

Configurar tu sitio web para HTTPS: Una vez que el certificado SSL esté instalado correctamente, necesitas configurar tu sitio web para que use HTTPS en lugar de HTTP. Esto generalmente implica actualizar los enlaces internos, asegurarse de que todas las imágenes, hojas de estilo y scripts se carguen a través de HTTPS y redireccionar el tráfico HTTP a HTTPS.

Hostinger proporciona guías detalladas y soporte técnico para ayudarte en este proceso. Una vez que HTTPS esté habilitado en tu sitio web, los visitantes verán el icono del candado en la barra de direcciones de su navegador, lo que indica que la conexión es segura y que sus datos están protegidos durante la transmisión.

#### **Compatibilidad de nuestras herramientas con hostinger:**

Hostinger es compatible con PHP, HTML, CSS, JavaScript y bases de datos MySQL, por lo que sería una opción adecuada para alojar una página web que utiliza estas tecnologías como la que estamos creando,

#### **¿Hostinger es compatible con PHP?**

Sí, Hostinger es compatible con páginas web que utilizan PHP. PHP es uno de los lenguajes de programación más comunes para el desarrollo web dinámico y muchas páginas web, incluidos sistemas de gestión de contenido (CMS) como WordPress, Joomla y Drupal, están escritas en PHP.

Hostinger ofrece soporte para PHP en sus planes de hosting compartido, VPS y alojamiento en la nube. Puedes alojar y ejecutar páginas web PHP en sus servidores sin ningún problema.

Para utilizar PHP en Hostinger, normalmente no necesitas hacer nada especial. Los servidores de Hostinger están configurados para admitir PHP de forma predeterminada. Solo necesitas subir tus archivos PHP al servidor y asegurarte de que tus páginas web estén configuradas correctamente para utilizar PHP.

Además de PHP, Hostinger también es compatible con otros lenguajes de programación comunes como HTML, CSS y JavaScript. Esto significa que puedes construir y alojar sitios web completos con tecnologías como HTML, CSS y JavaScript junto con la funcionalidad dinámica proporcionada por PHP.

En resumen, Hostinger es una opción adecuada si estás buscando un hosting web que admite PHP y te permite crear y alojar páginas web dinámicas y funcionales.

# Envío de correo electrónico:

Lo primero que hay que hacer para poder enviar un correo electrónico con php según nuestra investigación es tener instalado composer en nuestro servidor que estamos utilizando para correr código php, lo podemos instalar en <https://getcomposer.org/> dando clic en Download, podemos verificar si está instalado, abriendo nuestra terminal cmd y poniendo composer como en el siguiente ejemplo:

The screenshot shows a Windows desktop environment. In the center, there's a terminal window titled 'cmd' with the path 'C:\Windows\system32\cmd.exe'. It displays the output of the 'composer' command, which shows the version 'Composer version 2.7.1 2024-02-09 15:26:28' and a detailed usage and options help message. Above the terminal, several browser tabs are visible, including 'Enviar correos electrónicos...', 'Composer', '(S) WhatsApp', and 'Reporte\_Codeligner - Documento'. The taskbar at the bottom shows various pinned icons and the system tray indicating the date and time as '09/02/2024'.

```
C:\Windows\system32\cmd.exe: x + x
Microsoft Windows [Versión 10.0.22631.3085]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\bengi>composer
Composer version 2.7.1 2024-02-09 15:26:28
Usage:
  command [options] [arguments]

Options:
  -h, --help           Display help for the given command. When no command is given display help for the list command
  -q, --quiet          Do not output any message
  -V, --version         Display this application version
  --ansi[=no-ansi]     Force (or disable --no-ansi) ANSI output
  -n, --no-interaction Do not ask any interactive question
  --profile            Display timing and memory usage information
  --no-plugins          Whether to disable plugins.
  --no-scripts          Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache            Prevent use of the cache
  -v[v|vv|vvv], --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

nuestra investigación es tener instalado composer en nuestro servidor que estamos
utilizando para correr código php, lo podemos instalar en https://getcomposer.org/ dando clic
en Download, podemos verificar si está instalado, abriendo nuestra terminal cmd y poniendo
composer como en el siguiente ejemplo:
```

Si sale algo parecido como en la anterior captura, significa que tenemos instalado composer correctamente.

También podemos verificarlo en una nueva terminal dentro de Visual Studio Code:

The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORER' sidebar is visible with a folder named 'PRUEBAS' containing files like 'index.php', 'enviar\_correo.php', and 'composer.lock'. The main area is a terminal window with the title 'Pruebas'. It shows the output of the 'composer' command, identical to the one in the previous screenshot. Below the terminal, the 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL' tabs are visible. The status bar at the bottom shows the file path 'PS C:\Jaranon\www\Pruebas>', the date and time '09/02/2024 10:18 p.m.', and other system information.

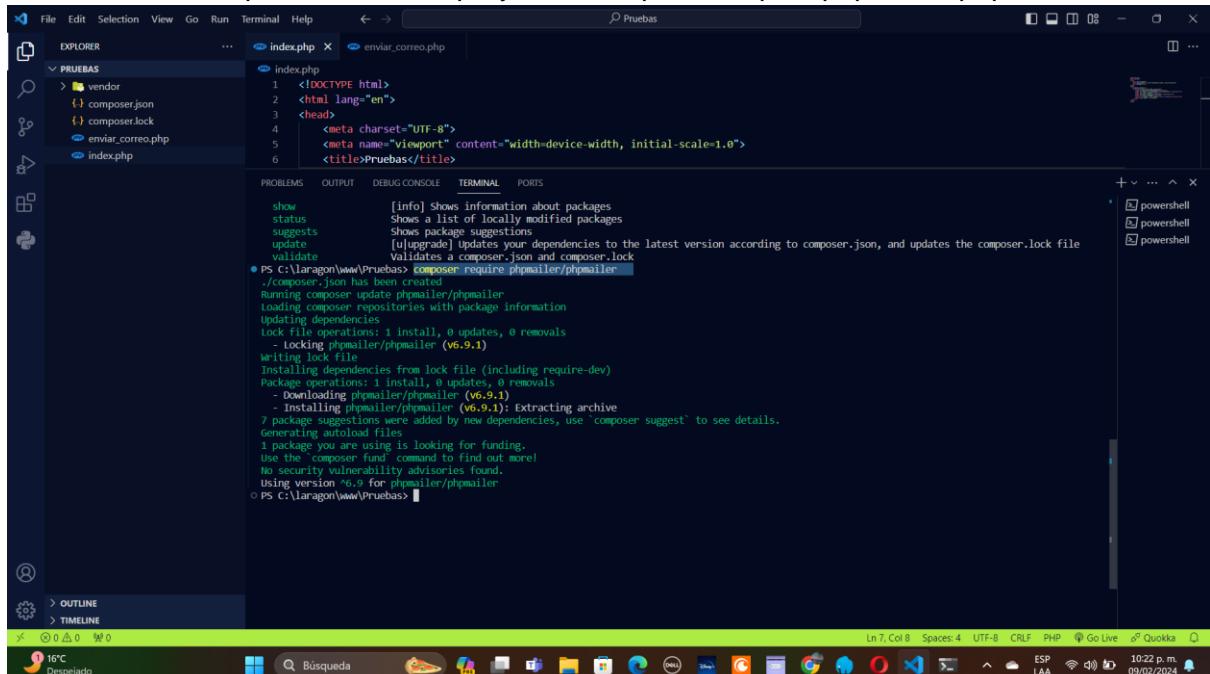
```
index.php x enviar_correo.php
PS C:\Jaranon\www\Pruebas> composer
Composer version 2.7.1 2024-02-09 15:26:28
Usage:
  command [options] [arguments]

Options:
  -h, --help           Display help for the given command. When no command is given display help for the list command
  -q, --quiet          Do not output any message
  -V, --version         Display this application version
  --ansi[=no-ansi]     Force (or disable --no-ansi) ANSI output
  -n, --no-interaction Do not ask any interactive question
  --profile            Display timing and memory usage information
  --no-plugins          Whether to disable plugins.
  --no-scripts          Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache            Prevent use of the cache
  -v[v|vv|vvv], --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about      Shows a short information about Composer
  archive    Creates an archive of this composer package
  audit      Checks for security vulnerability advisories for installed packages
  browse     [home] Opens the package's repository URL or homepage in your browser
  bump       Increases the lower limit of your composer.json requirements to the currently installed versions
  check-platform-req  Checks that platform requirements are satisfied
  clear-cache [clearcache][cc] Clears composer's internal package cache
  completion Dump the shell completion script

Ln 7, Col 8  Spaces: 4  UFT-8  CRLF  PHP  Go Live  Quokka  10:21 p.m.  09/02/2024
```

Luego de verificar, colocaremos el siguiente comando, cuidando que en la terminal estemos ubicados en la carpeta de nuestro proyecto: composer require phpmailer/phpmailer

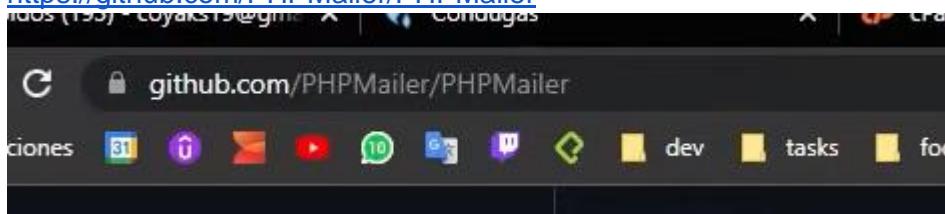


A screenshot of the Visual Studio Code interface. The terminal tab is active, showing the command `composer require phpmailer/phpmailer` being run. The output shows the creation of a `composer.json` file, the loading of composer repositories, and the updating of dependencies, specifically installing `phpmailer/phpmailer (v6.9.1)`. The terminal also displays security information and the generation of autoload files. The status bar at the bottom right shows the date and time as 09/02/2024.

Una vez ejecutemos este comando, y si todo es correcto, nos va a crear la carpeta vendor en nuestro proyecto.

Entonces nos dirigimos a la siguiente dirección para encontrar un código de la documentación para el envío de correos con el uso de phpmailer:

<https://github.com/PHPMailer/PHPMailer>



Una vez teniendo eso, ya podemos empezar a implementar la lógica para que funcione nuestro código, en el php que tiene el funcionamiento de la página, tenemos que añadir lo siguiente:



A screenshot of a code editor showing two files: `index.php` and `enviar_correo.php`. The `enviar_correo.php` file contains the following code:

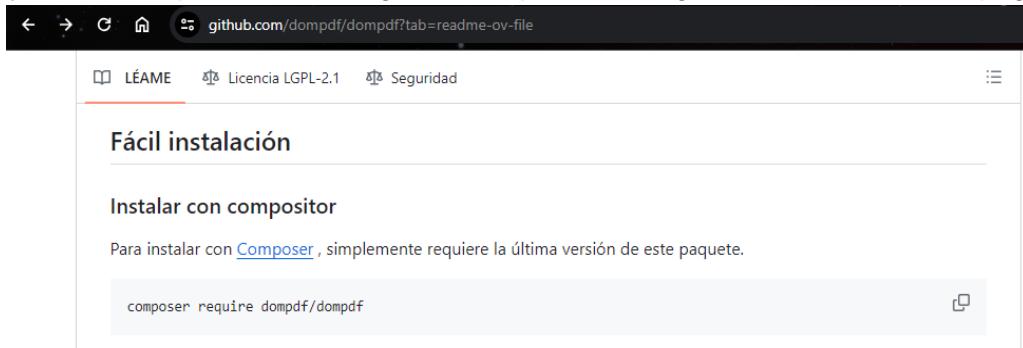
```
1 <?php
2 use PHPMailer\PHPMailer\PHPMailer;
3 use PHPMailer\PHPMailer\SMTP;
4 use PHPMailer\PHPMailer\Exception;
5
6 require 'vendor/autoload.php';
7
8 $mail = new PHPMailer(true);
```

# Crear y Descargar un PDF

Para crear un pdf, ocuparemos “dompdf”, este lo podemos descargar desde aquí:

<https://github.com/dompdf/dompdf?tab=readme-ov-file>

Y con ayuda de composer lo descargamos. Copiamos el siguiente comando de la página de



The screenshot shows a GitHub repository page for 'dompdf'. The URL in the address bar is 'github.com/dompdf/dompdf?tab=readme-ov-file'. The page has tabs for 'LÉAME', 'Licencia LGPL-2.1', and 'Seguridad'. The main content area is titled 'Fácil instalación' and contains the heading 'Instalar con composer'. Below it, there is a note: 'Para instalar con [Composer](#), simplemente requiere la última versión de este paquete.' A code block shows the command: 'composer require dompdf/dompdf'.

github:

Nos vamos a nuestro proyecto de visual studio code, abrimos una terminal y pegamos el comando para la instalación de dompdf.

Dejamos que se instale. Composer nos va a ayudar a configurar todo por nosotros. Y nos guiamos de git para poder empezar nuestro pdf y al mismo tiempo se pueda descargar.

Es necesario tener en cuenta que por medio de html se va ir creando lo que es nuestro pdf. Así trabaja dompdf.

Primero comenzamos por usar una referencia:

```
// reference the Dompdf namespace
use Dompdf\Dompdf;
```

Hacemos una instancia de esa clase y le damos un nombre:

```
// instantiate and use the dompdf class
$dompdf = new Dompdf();
$dompdf->loadHtml('hello world');
```

Con esto configuramos el tamaño y la orientación del papel.

```
// (Optional) Setup the paper size and orientation
$dompdf->setPaper('A4', 'landscape');
```

Renderizamos el html a pdf, esto quiere decir que vamos a convertir el html que ocupamos o hicimos para el pdf y lo convertimos a formatos pdf.

```
// Render the HTML as PDF
$dompdf->render();
```

Esto último es para la descarga del pdf:

```
// Output the generated PDF to Browser
$dompdf->stream();
```

# Página Responsiva.

Al estar en php y con el framework CodeIgniter trabajamos con lo que es html en conjunto, entonces para que nuestras pantallas se vean responsivas no es tan complejo, solo necesitamos los links de bootstrap.

Entramos al link <https://getbootstrap.com/docs/5.3/getting-started/download/>

The screenshot shows the Bootstrap 5.3 documentation page. In the left sidebar, under the 'Getting started' section, the 'Download' option is selected. The main content area shows two code snippets. The first snippet, titled 'CDN via jsDelivr', provides links to compiled CSS and JS files. The second snippet, titled 'Alternative CDNs', provides links to files from other providers like cdnjs or unpkg. A yellow warning bar at the bottom of the content area states: 'If the SRI hash differs for a given file, you shouldn't use the file from that CDN, because it means that the file was modified by someone else.'

Copiamos los links que nos da justo donde dice: CDN via jsDelivr, varian dependiendo la versión, en este caso ocupamos la versión más actual que es la 5.3 y estos son los links:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
      rel="stylesheet" integrity="sha384-T3c6Coli6uLrA9TneNEoa7RxnatzcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
      crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
      crossorigin="anonymous"></script>
```

Estos enlaces los vamos a pegar justo abajo de donde ponemos el título a nuestro html.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Lista de Libros</title>
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
8          rel="stylesheet" integrity="sha384-T3c6Coli6uLrA9TneNEoa7RxnatzcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">
9      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
10         integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>
11  </head>
12  <body style="background-color: #B9C9E4;">
```

Y listo, ya podemos ocupar bootstrap para que nuestra aplicación sea responsive.

Por ejemplo, lo que es mi página principal, es una tabla que ocupamos y está hecha con bootstrap.

```
<br>
<a href="=base_url('crear')?&gt;" class="btn btn-primary mb-2" type="button"&gt;Crear un Libro&lt;/a&gt;
&lt;br&gt;
&lt;table class="table table-ligth mt-2"&gt;
    &lt;thead class="table-ligth"&gt;
        &lt;tr&gt;
            &lt;th&gt;#&lt;/th&gt;
            &lt;th&gt;Imagen&lt;/th&gt;
            &lt;th&gt;Nombres&lt;/th&gt;
            &lt;th&gt;Acciones&lt;/th&gt;
        &lt;/tr&gt;
    &lt;/thead&gt;
    &lt;tbody&gt;
        &lt;?php foreach ($libros as $libro):?&gt;
        &lt;tr&gt;
            &lt;td&gt;&lt;?=$libro['id'];?&gt;&lt;/td&gt;
            &lt;td&gt;
                &lt;img class="img-thumbnail" src="<?=base_url()?/uploads/<?=$libro['imagen'];?>" width="100" alt="">
            </td>
            <td><?=$libro['nombre']; ?></td>
            <td>
                <a href="=base_url('editar/'.$libro['id'])?&gt;" class="btn btn-info m-1" type="button"&gt;Editar&lt;/a&gt;
                &lt;a href="<?=base_url('borrar/'.$libro['id'])?&gt;" class="btn btn-danger m-1" type="button"&gt;Borrar&lt;/a&gt;
            &lt;/td&gt;
        &lt;/tr&gt;
        &lt;?php endforeach; ?&gt;
    &lt;/tbody&gt;
&lt;/table&gt;</pre
```

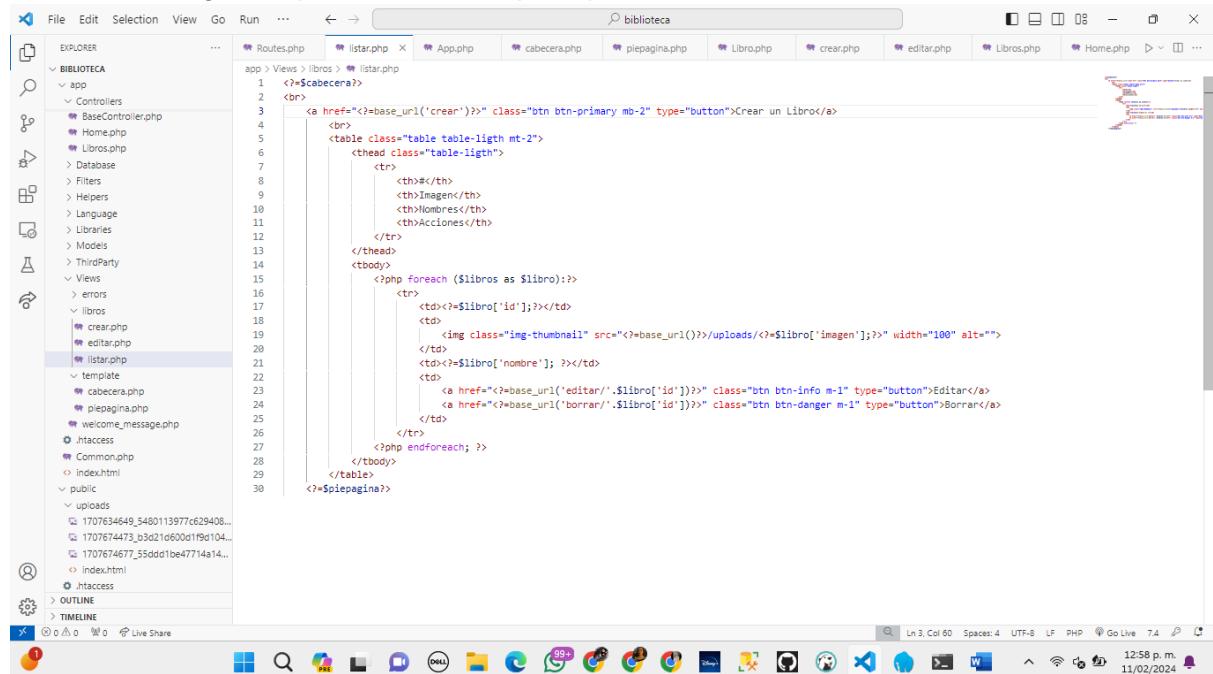
Y se visualiza así.

The screenshot shows a web application titled "Sistema de Libros". The main content is a table listing three books:

#	Imagen	Nombres	Acciones
5		Holiii	<a href="#">Editar</a> <a href="#">Borrar</a>
7		Pase de Parámetros con método POST.	<a href="#">Editar</a> <a href="#">Borrar</a>
8		Pase de Parámetros con método POST y Base de Datos MySQL.	<a href="#">Editar</a> <a href="#">Borrar</a>

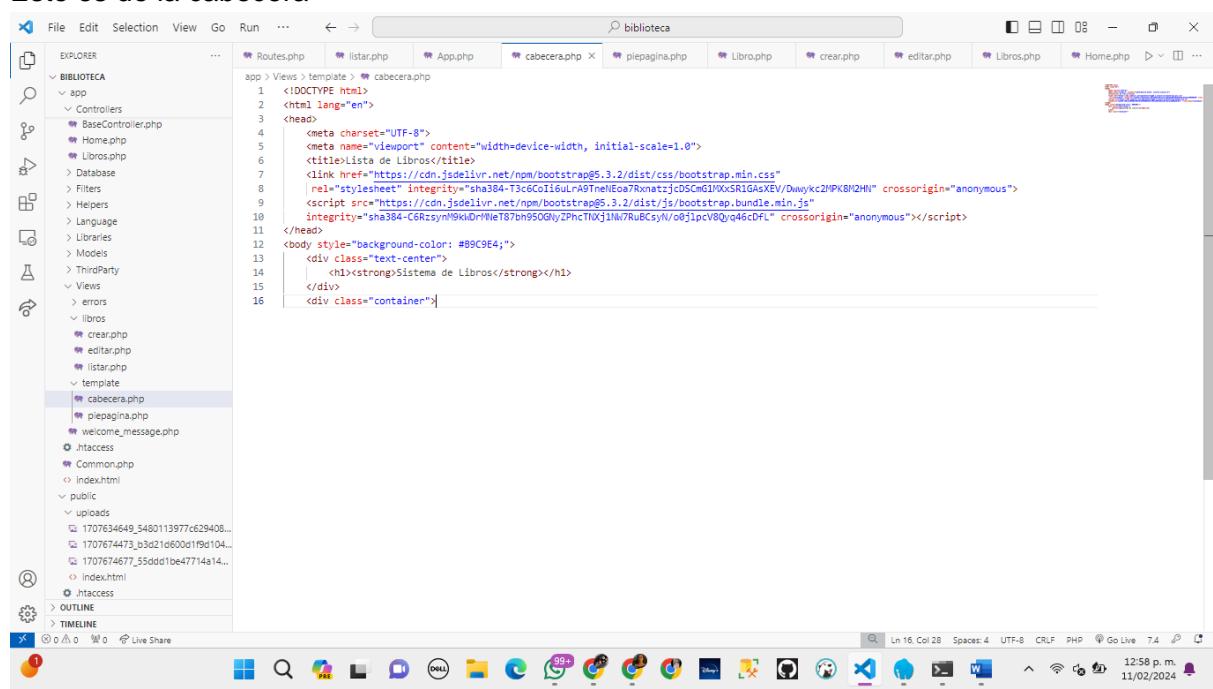
Gracias a php, pude dividir mi html por partes e hice la cabecera y el pie de manera genérica para no tener que hacerlos cada que haga una nueva vista.

Este es el código completo de mi vista principal:



```
<?php ><?>$cabecera>
<br>
<table class="table table-light mt-2">
<thead class="table-light">
<tr>
<th></th>
<thImagen/>
<thNombres/>
<thAcciones/>
</tr>
</thead>
<tbody>
<?php foreach ($libros as $libro):?>
<tr>
<td><?= $libro['id']?></td>
<td>
<img class="img-thumbnail" src=<?><?base_url()?>/uploads/<?>$libro['imagen']?> width="100" alt="">
</td>
<td><?= $libro['nombre']?></td>
<td>
<a href=<?><?base_url('editar/'.$libro['id'])?><?> class="btn btn-info m-1" type="button">Editar</a>
<a href=<?><?base_url('borrar/'.$libro['id'])?><?> class="btn btn-danger m-1" type="button">Borrar</a>
</td>
</tr>
<?php endforeach?>
</tbody>
</table>
<?= $piepagina?>
```

Este es de la cabecera



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Lista de Libros</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3cEdIiUlA0TneIeo7RxnatjzDSCmG1WxSR1GAsXEV/DwiyK2MPKBH2HN" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C8Rzsyn9XiDDrT87bh95OGNyZPnCNIjIW7RuBCsyN/o8jpcv8Q/q46cDFL" crossorigin="anonymous"></script>
</head>
<body style="background-color: #B0C4DE;">
<div class="text-center">
<h1><strong>Sistema de Libros</strong></h1>
</div>
<div class="container">
```

app > Views > template > piepagina.php

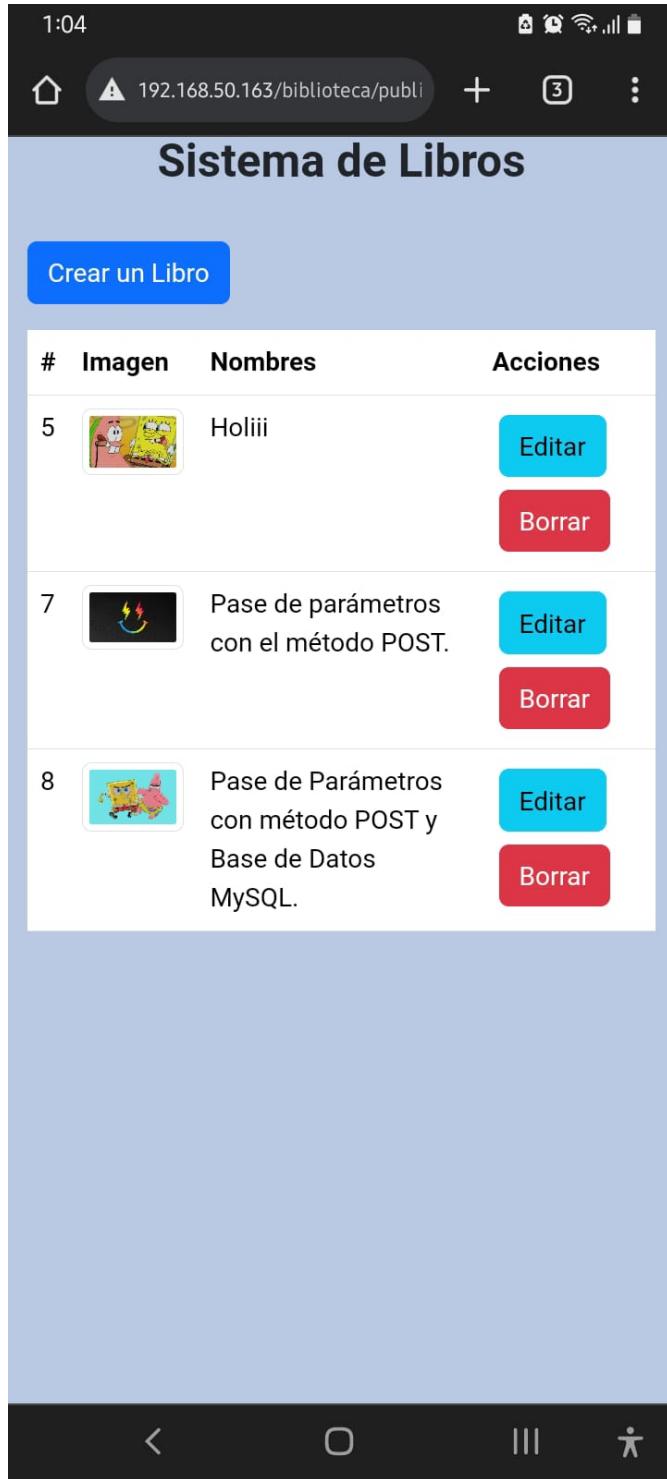
```
1   </div>
2   </body>
3   </html>
```

Y del pie de página

Y solo realizamos esto para poder ocuparlos, lo que hacemos es que esas vistas se mandaran como una variable a la vista en donde se visualizará.

```
public function crear(){
    $datos['cabecera'] = view('template/cabecera');
    $datos['piepagina'] = view('template/piepagina');
    return view("libros/crear", $datos);
}
```

Si lo vemos en un celular móvil, se verá de una manera amigable y responsive:



# Almacenar Imágenes.

Primero vemos que en nuestra base de datos solo se va a guardar el nombre de nuestra imagen:

Lo que hace que se puedan ver aunque las pongamos desde otro dispositivo es el uso del método guardar en nuestro controlador.

```
public function guardar(){
    $libro = new Libro();
    $nombre = $this->request->getVar("nombre");
    $imagen=$this->request->getFile("imagen");

    if($imagen=$this->request->getFile("imagen")){
        $nuevoNombre=$imagen->getRandomName();
        $imagen->move("../public/uploads/", $nuevoNombre);
        $datos=[ "nombre"=>$nombre, "imagen"=>$nuevoNombre];
    }
    $libro->insert($datos);
    print_r($this->request->getFile("imagen"));
    print_r($this->request->getVar("nombre"));
    return $this->response->redirect(site_url('/listar'));
}
```

En donde primero hacemos una instancia de la clase Libro que es nuestro Modelo.

```
app > Models > Libro.php > App\Models\Libro
6   class Libro extends Model{
    18 references
7     protected $table      = 'libros';
8     // Uncomment below if you want add primary key
9     // protected $primaryKey = 'id';
    21 references
10    protected $primaryKey = 'id';
    7 references
11    protected $allowedFields = [ 'nombre', 'imagen' ];
12 }
```

Volviendo al método guardar, lo que hacemos es verificar que haya una imagen y un nombre, asignando una variable para cada campo según corresponda, luego validamos si hay una imagen la vamos a renombrar con un nombre aleatorio en caso de que ya haya una imagen así guardada y para evitar problemas en los nombres hacemos esto, estas imágenes se van a guardar en una carpeta que por default se va a crear, en este caso es uploads y ahí se van a guardar las imágenes.

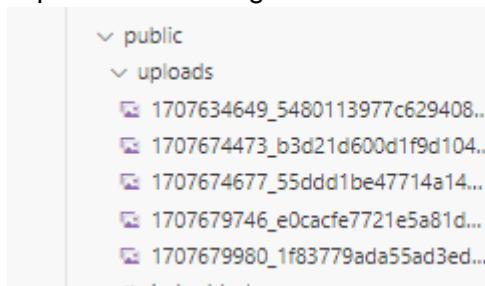
Después hacemos un arreglo en donde irá el nombre y el nuevo nombre de la imagen y es lo que se va a insertar en la base de datos.

Imprimimos los parámetros en una nueva pestaña, esto se puede saltar, solo es importante realizarlo al principio para verificar que se estén mandando. Luego vamos a redirigir al usuario a la página principal en “listar”, donde va a poder ver que ya está insertado su libro.

#	Imagen	Nombres	Acciones
5		Holiii	<button>Editar</button> <button>Borrar</button>
7		Pase de Parámetros con método POST.	<button>Editar</button> <button>Borrar</button>
8		Pase de Parámetros con método POST y Base de Datos MySQL.	<button>Editar</button> <button>Borrar</button>
9		Harry Potter	<button>Editar</button> <button>Borrar</button>
10		Español	<button>Editar</button> <button>Borrar</button>

Aquí vemos que ya se ven las imágenes.

Aquí están las imágenes almacenadas en la carpeta que creamos.



Como vemos son las 5 imágenes que están insertadas en la bd.

# Pase de Parámetros.

## Método Get

Hay varias maneras de hacer esto, la primera es pasar los parámetros por medio de url con el método get, en este caso hice un pequeño ejemplo.

Aquí tengo un pequeño crud, en donde para realizar la eliminación, se necesita pasar por medio de la url el id del objeto que se va a eliminar:

```
public function borrar($id=null){  
    echo('Borrar registro'. $id );  
    $libro = new Libro();  
    $datosLibro = $libro->where('id',$id)->first();  
    $ruta ='../public/uploads/'. $datosLibro['imagen'];  
    unlink($ruta);  
    $libro->where('id',$id)->delete($id);  
    echo('Se elimino el libro '.$datosLibro['nombre']. 'con el id: '.$id);  
}
```

La url nos va a dar el id y esta función lo va a atrapar y lo va a eliminar.

Esta es la url:

```
3 $routes->get('borrar/(:num)', 'Libros::borrar/$1');
```

Aquí le mandamos la url y ahí mismo le estamos diciendo que el método borrar del controlador Libros lo va a agarrar.

Primero entramos a nuestra página principal:

#	Imagen	Nombres	Acciones
5		Hollii	<button>Editar</button> <button>Borrar</button>
6		Ando Feliii Jajaja	<button>Editar</button> <button>Borrar</button>
7		Pase de Parámetros con método POST.	<button>Editar</button> <button>Borrar</button>
8		Pase de Parámetros con método POST y Base de Datos MySQL.	<button>Editar</button> <button>Borrar</button>

Después al darle en borrar a cualquiera, nos va a borrar ese registro y se verá por url el id con el cual se borrará.



Y como vemos ya no está el registro 6:

A screenshot of a Windows desktop environment. On the screen, there is a web browser window titled "Lista de Libros" displaying a table of books. The table has columns for "#", "Imagen" (Image), "Nombres" (Names), and "Acciones" (Actions). There are three rows of data:

#	Imagen	Nombres	Acciones
5		Holiii	<button>Editar</button> <button>Borrar</button>
7		Pase de Parámetros con método POST.	<button>Editar</button> <button>Borrar</button>
8		Pase de Parámetros con método POST y Base de Datos MySQL.	<button>Editar</button> <button>Borrar</button>

The taskbar at the bottom of the screen shows various pinned icons and the system tray indicates the date and time as "12:19 p.m. 11/02/2024".

## Método Post

Para este método es insertar en este ejemplo un libro, y le vamos a pasar los parámetros recogidos no sólo a otra pestaña, si no también a la base de datos.

Para esto tenemos nuestro método guardar, lo que hace es guardar los datos en nuestra bd.

```
0 references | 0 overrides
21 public function guardar(){
22     $libro = new Libro();
23     $nombre = $this->request->getVar("nombre");
24     $imagen=$this->request->getFile("imagen");
25
26     if($imagen==$this->request->getFile("imagen")){
27         $nuevoNombre=$imagen->getRandomName();
28         $imagen->move("../public/uploads/", $nuevoNombre);
29         $datos=[ "nombre"=>$nombre, "imagen"=>$nuevoNombre];
30     }
31     $libro->insert($datos);
32     print_r($this->request->getFile("imagen"));
33     print_r($this->request->getVar("nombre"));
34 }
```

Y nuestra url, aquí nos daremos cuenta que en esta ocasión no se pasarán los parámetros de esa manera, la url se verá normal.

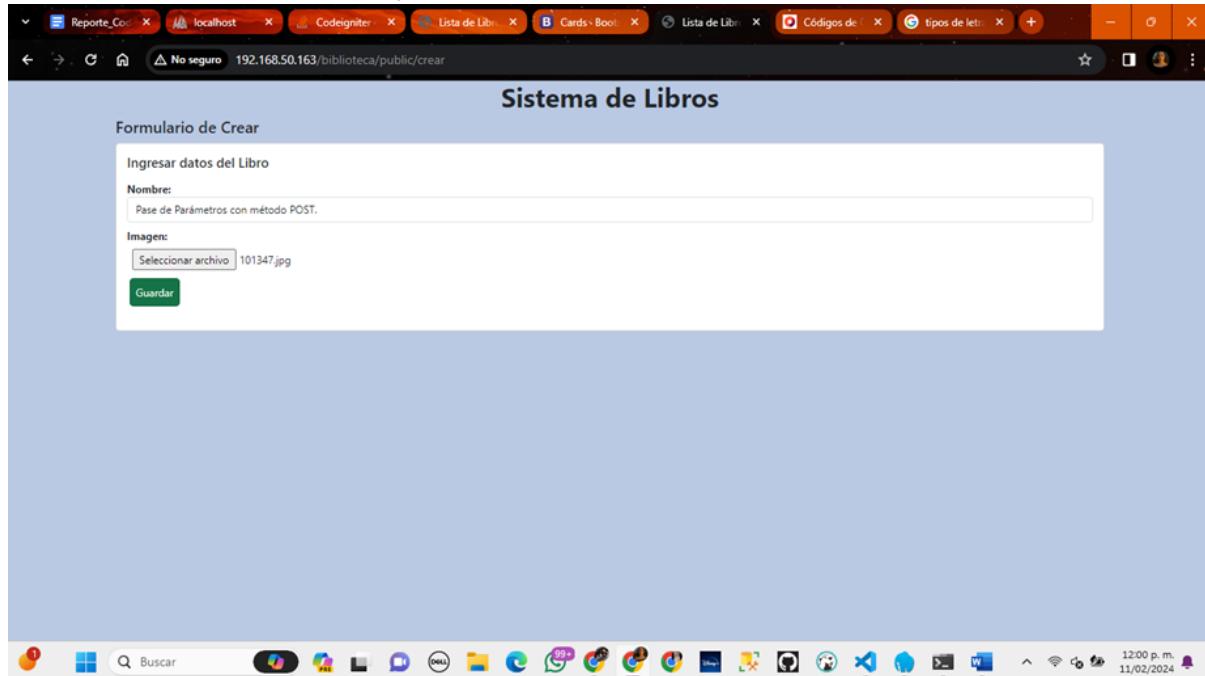
```
$routes->post('guardar', 'Libros::guardar');
```

Si lo ejecutamos, se vería algo así.

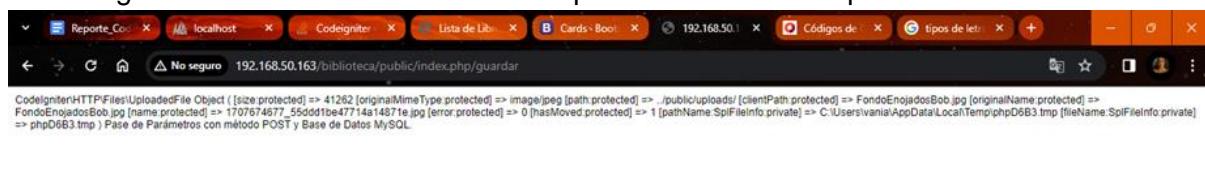
#	Imagen	Nombres	Acciones
5		Holii	<a href="#">Editar</a> <a href="#">Borrar</a>
6		Ando Feliii JajaJaja	<a href="#">Editar</a> <a href="#">Borrar</a>

Le damos en crear un nuevo libro primero.

Nos mandará a un formulario y lo llenamos:



Al darle guardar nos mandará a una nueva pestaña en donde imprimimos los datos.



Así pasamos parámetros con el método post.

# Validaciones

En el formulario para crear un libro ponemos algunas validaciones para que este se cree y sobre todo para que el usuario no haga lo que quiera. En este caso, para la imagen y el nombre.

```
$validacion = $this->validate([
    "nombre"=> "required|min_length[3]",
    'imagen'=>
        'uploaded[imagen]',
        'mime_in[imagen,image/jpg,image/jpeg,image/png,image/webp]',
        'max_size[imagen,1024]',
]);
```

Le estamos diciendo que el nombre debe de contener más de 3 letras y la imagen debe de tener la extensión de una imagen y un máximo de 1024px.

Si esto se cumple, entonces se cambia el nombre de la imagen, se guarda y se inserta en la base de datos.

Como estas, hay más validaciones que se pueden realizar.

```
if($validacion){
    if($imagen=$this->request->getFile("imagen")){
        $nuevoNombre=$imagen->getRandomName();
        $imagen->move("../public/uploads/", $nuevoNombre);
        $datos=[ "nombre"=>$nombre, "imagen"=>$nuevoNombre];
    }
}
$libro->insert($datos);
```

Si no se cumplen estas validaciones, entonces se va a redireccionar a la pantalla de inicio.

```
}else{
    return $this->response->redirect(site_url('/listar'));
}
```

# Sesiones

En este ejemplo de sesiones, yo ocupo lo que es la función de session, para también validar registros de libros correctos.

```
if($validacion){
    if($imagen=$this->request->getFile("imagen")){
        $nuevoNombre=$imagen->getRandomName();
        $imagen->move("../public/uploads/", $nuevoNombre);
        $datos=[ "nombre"=>$nombre, "imagen"=>$nuevoNombre];
    }
} else{
    $session = session();
    $session->setflashdata("mensaje","Información inválida, el libro debe contener más de 3 caracteres
y el archivo debe ser una imagen, extensión (.png, .jpg, .jpeg o .webp)");

    return $this->response->redirect(site_url('/listar'));
}
```

Esto lo hago para mandarle mensajes al usuario, no solo redireccionarlo, porque el usuario no sabe porque no se insertó y porque se salió de su registro, para eso estamos ocupando aquí las sesiones.

En el formulario también hacemos mención de las sesiones, ya que le vamos a mandar mensaje al usuario por medio de la sesión con una alerta.

```
<h4>Formulario de Crear</h4>

<?php
    if(session('mensaje')){
?>

<div class="alert alert-danger" role="alert">
    <?php
        echo session('mensaje');
    ?>
</div>

<?php
    }
?>
```

# Como encriptar una contraseña o clave en php con hash (la manera más recomendable):

Para esto tenemos una función llamada `password_hash()` que recibe 2 parámetros, el string que se va a encriptar y el algoritmo que se va a usar, para ver los algoritmos que admite esta función hash mas moderna podemos ir a la documentación:

<https://www.php.net/manual/es/function.password-hash.php>

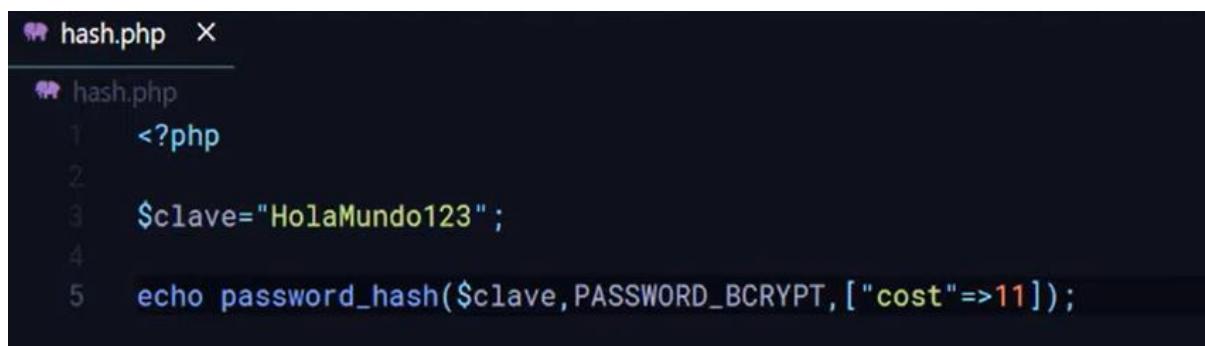
Por ejemplo nos da los siguientes algoritmos:

En este caso utilizó el algoritmo `PASSWORD_DEFAULT` pero la diferencia es que siempre va a cambiar la manera en que encripta mi String, si yo refresco la pagina varias veces, observare que la encriptación siempre cambia, aunque sea el mismo String que estoy encriptando, cosa que no era así con los anteriores algoritmos antiguos, por eso es más recomendable la función `password_hash()`.



```
hash.php <?php  
$clave="HolaMundo123";  
echo password_hash($clave,PASSWORD_DEFAULT);
```

También puedo mandarle un tercer parámetro si quiero para hacer mas fuerte la encriptación, el default es 10 pero aquí lo cambio a 11, si yo no mando este tercer parámetro a la función `password_hash` mantiene el 10.



```
hash.php <?php  
$clave="HolaMundo123";  
echo password_hash($clave,PASSWORD_BCRYPT, [ "cost"=>11]);
```

Ahora con la función `password_verify()` verifica, comparando la clave original con la encriptada, y si es igual nos da un valor booleano, false o true en caso de que si coinciden. En este caso muestra un 1 que significa true ya que si coinciden.

```
hash.php
1 <?php
2
3 $clave="HolaMundo123";
4
5 $clave_procesada=password_hash($clave,PASSWORD_BCRYPT,[ "cost"=>11]);
6
7 echo password_verify($clave,$clave_procesada);
```

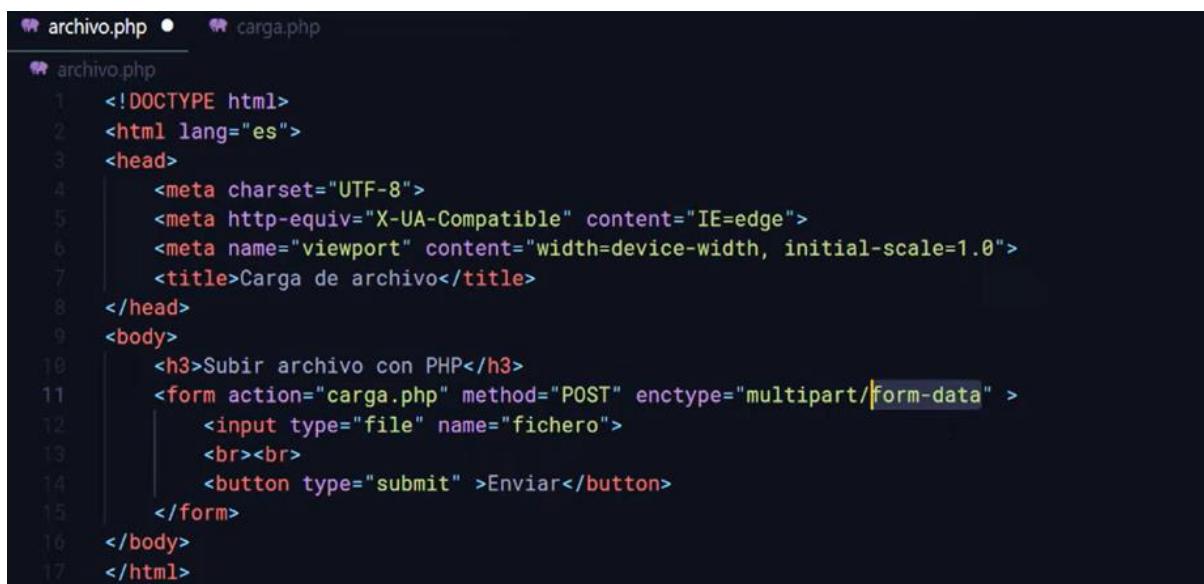
Esto puede servirnos ya que podemos ocuparlo en una condición if

```
hash.php X
hash.php
1 <?php
2
3 $clave="HolaMundo123";
4
5 $clave_procesada=password_hash($clave,PASSWORD_BCRYPT,[ "cost"=>11]);
6
7 if(password_verify($clave,$clave_procesada)){
8     echo "Las claves coinciden";
9 }
```

# Como subir o enviar archivos a servidor php con formularios:

Para esto tenemos el siguiente formulario y en la etiqueta form colocamos **action** para indicar a que archivo queremos pasar los datos o en este caso el archivo, el método con **method** a ocupar (solo se puede con POST) y un nuevo atributo que estamos viendo que también es necesario **enctype="multipart/form-data"** para que se puedan enviar correctamente los archivos, con ese valor indicamos que queremos enviar archivos desde ese formulario que tenemos.

NOTA: Únicamente se puede con el método POST



The screenshot shows a code editor with two files open. The file 'archivo.php' contains the following code:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Carga de archivo</title>
</head>
<body>
    <h3>Subir archivo con PHP</h3>
    <form action="carga.php" method="POST" enctype="multipart/form-data" >
        <input type="file" name="fichero">
        <br><br>
        <button type="submit" >Enviar</button>
    </form>
</body>
</html>
```

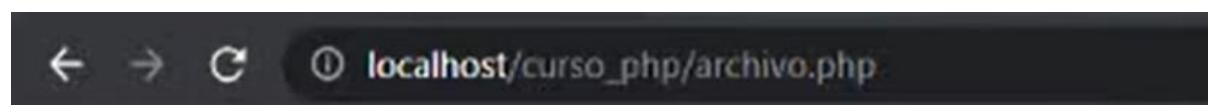
De esta manera accedemos a información del archivo que se selecciono y se envio en mi formulario, con name accedemos al nombre del archivo, con tmp\_name accedemos a la ruta en la que se encuentra el archivo, el type para el tipo de archivo que estamos cargando, error para detectar si el archivo se cargo bien o no, y size para el tamaño del archivo en bytes



The screenshot shows the 'carga.php' file from the code editor. The code is as follows:

```
<?php
echo $_FILES['fichero']['name']."<br>";
echo $_FILES['fichero'][ 'tmp_name' ]."<br>";
echo $_FILES['fichero'][ 'type' ]."<br>";
echo $_FILES['fichero'][ 'error' ]."<br>";
echo $_FILES['fichero'][ 'size' ]."<br>";
```

Por ejemplo:



## Subir archivo con PHP

No se eligió archivo



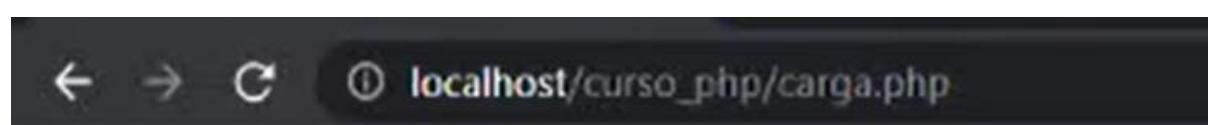
Si selecciono un archivo y doy clic al botón para enviar:



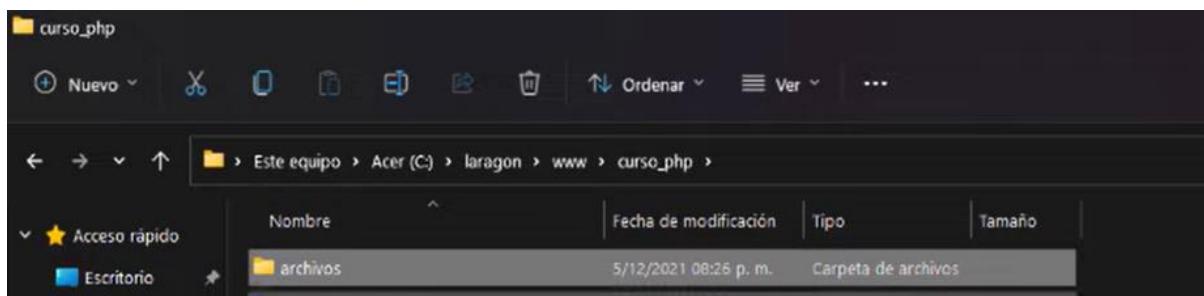
## Subir archivo con PHP

cover\_php.jpg

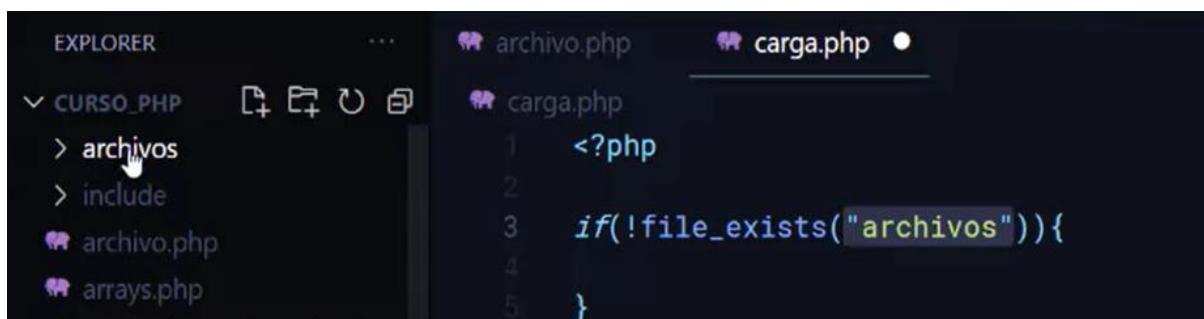
Y me muestra los datos del archivo:



Bueno para continuar vamos a crear una carpeta en el proyecto, que es en donde se van a almacenar todos los archivos que mandemos mediante el formulario en este caso se llama archivos.



Aquí ocupamos la función `file_exists(" ")` para saber si la carpeta o el archivo existe, entre las comillas va la ruta de la carpeta, esta función nos devuelve valores booleanos, en este caso decimos que si el formulario existe nos de FALSE ya que lo estamos negando con !



Aquí decimos que si el directorio o la carpeta no esta creado, entonces se va a crear con el nombre de archivos usando la función `mkdir(" ")`, dentro de las comillas va el **nombre de la carpeta que queremos crear**, para eso sirve esa función, y también recibe otro parámetro que son los permisos, con 0777 damos todos los permisos de escritura y lectura para la carpeta para no tener problemas de permisos en el servidor. Y para **detener la ejecución de un script** lo hacemos con la función `exit()`.

Entonces aquí estamos diciendo que si no existe una carpeta llamada archivos, la cree, y si no se puede crear me mande un error y se deje de ejecutar el script.



Ahora utilizaremos la función `chmod(" ")` que sirve para dar permisos de lectura y escritura a una carpeta a esta función se le envían 2 parámetros, en este caso queremos hacer eso con la carpeta que YA tenemos creada llamada archivos.

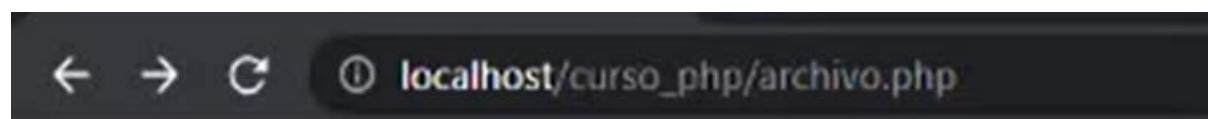
Y también usamos la función `move_uploaded_file()` en donde ocupa la ruta donde esta almacenado el archivo esto con `tmp_name` como ya habíamos visto un poco mas atrás, luego ponemos en donde queremos colocar el archivo que se encuentra en la carpeta temporal asignándole el nombre que tenia nuevamente.

Entonces aquí le estamos diciendo que mueva el archivo almacenado en la carpeta temporal, que vaya a mi carpeta archivos y que entre ( / ), después que mueva el archivo a esa carpeta asignándole su nombre.



```
archivo.php carga.php
carga.php
1 <?php
2
3 if(!file_exists("archivos")){
4     if(!mkdir("archivos",0777)){
5         echo "Error al crear el directorio";
6         exit();
7     }
8 }
9
10 chmod("archivos",0777);
11
12 if(move_uploaded_file($_FILES['fichero']['tmp_name'], "archivos/".$_FILES['fichero']['name'])){
13     echo "Archivo subido con exito";
14 }else{
15     echo "Error al subir el archivo";
16 }
17 }
```

Este código ya es capaz de mover un archivo a una carpeta, si yo selecciono un archivo y lo muevo ocurrirá lo siguiente:

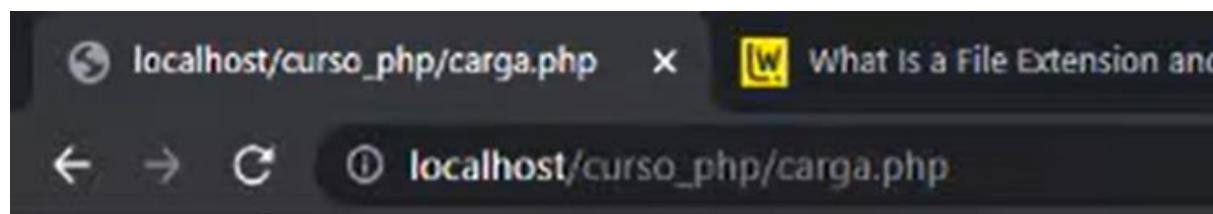


## Subir archivo con PHP

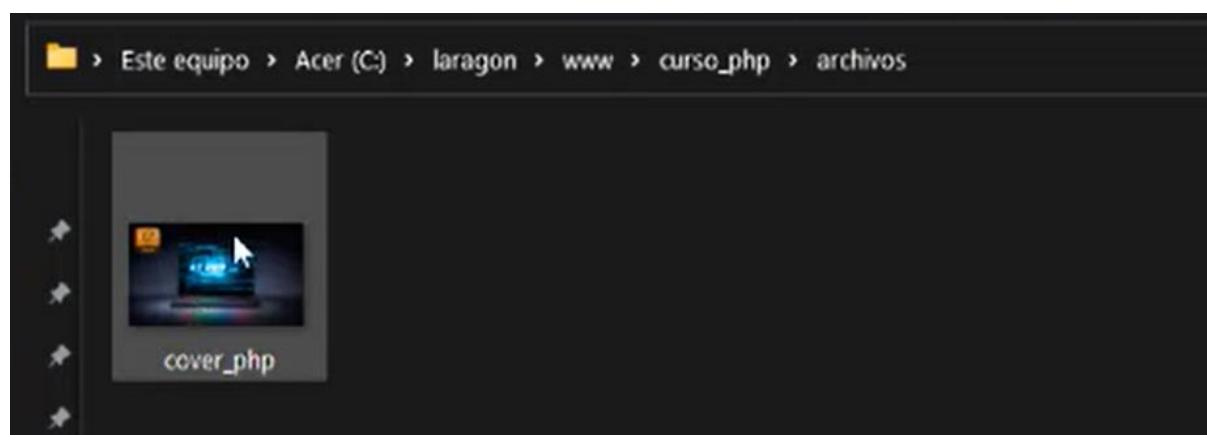


Seleccionar archivo cover\_php.jpg

Enviar



Si vemos la carpeta que creamos, observaremos que ya tiene el archivo, en caso de no existir la carpeta la va a crear y se va a mover el archivo a esa carpeta:



## Manejo de sesiones en php con variables de **\$\_SESSION**:

### Sesiones en PHP

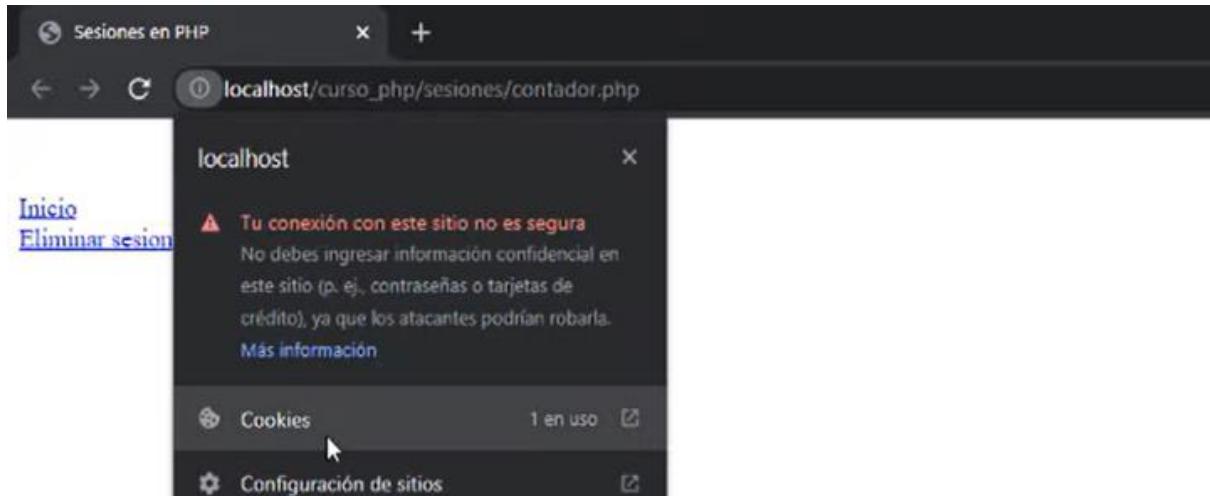
Las **sesiones** en aplicaciones web desarrolladas con **PHP** nos sirven para **almacenar información** durante toda la visita de un usuario a un **sitio web**. Dicha información se almacena en el **servidor**.

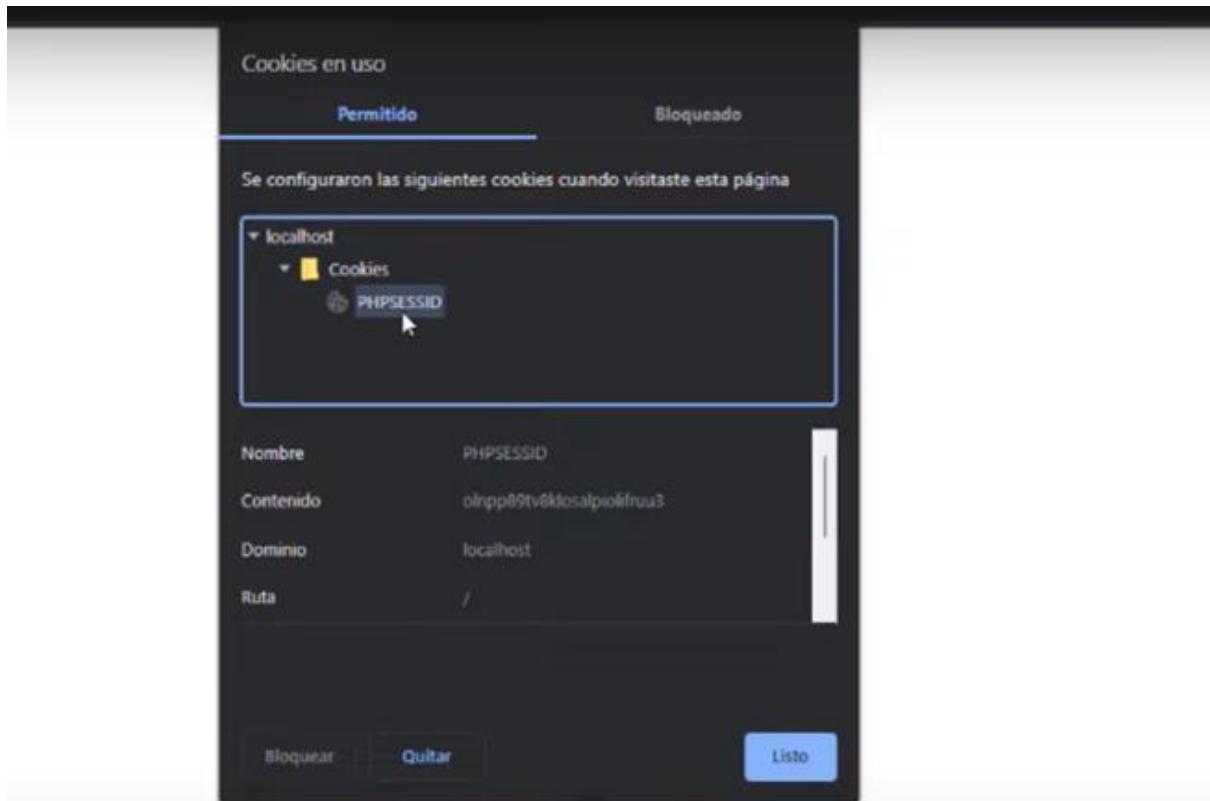
**Ejemplo de uso de sesiones:** inicio de sesión (login), buscadores (que recuerde término de búsqueda) etc.

Las sesiones normalmente se crean al inicio de cualquier otro código, por eso es lo primero que tenemos en este ejemplo, creamos una sesión con la función `sesión_start()`

```
contador.php index.php login.php cerrar.php ...  
sesiones > contador.php  
1 <?php  
2     session_start();  
3 ?>  
4 <!DOCTYPE html>  
5 <html lang="es">  
6 <head>  
7     <meta charset="UTF-8">  
8     <meta http-equiv="X-UA-Compatible" content="IE=edge">  
9     <meta name="viewport" content="width=device-width,  
10        initial-scale=1.0">  
11     <title>Sesiones en PHP</title>  
12 </head>  
13 <body>  
14     <br><br>  
15     <a href="index.php">Inicio</a>  
16     <br>  
17     <a href="cerrar.php">Eliminar sesión</a>  
18 </body>  
19 </html>
```

Al crear una sesión se crea una cookie para tener una conexión entre cliente servidor, para ver las cookies que tenemos es así:



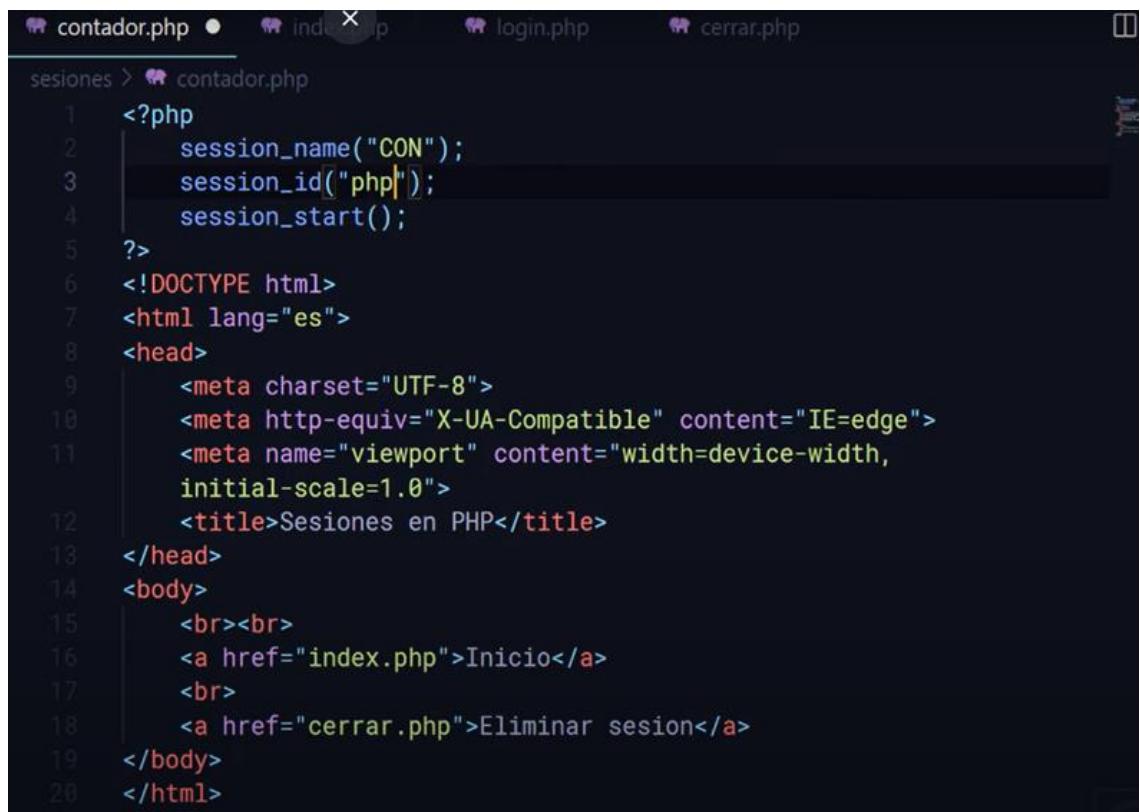


También puedo verlo en aplicaciones al inspeccionar, aquí me dice el nombre y clave que tiene:

Nombre	Valor	Domain	Path	Expires / Max-Age	Tamaño
PHPSESSID	olnpp89tv6klosalpolfruu3	localhost	/	Sesión	35

Cookie Value: olnpp89tv6klosalpolfruu3

Si yo quiero que el nombre y la clave sean diferentes es de la siguiente forma, utilizando la función `sesión_name(" ")` para cambiarle el nombre y la función `session_id(" ")` para cambiar el id o el valor.



```
contador.php • index.php • login.php • cerrar.php
sesiones > contador.php
1 <?php
2     session_name("CON");
3     session_id("php");
4     session_start();
5 ?>
6 <!DOCTYPE html>
7 <html lang="es">
8 <head>
9     <meta charset="UTF-8">
10    <meta http-equiv="X-UA-Compatible" content="IE=edge">
11    <meta name="viewport" content="width=device-width,
12        initial-scale=1.0">
13    <title>Sesiones en PHP</title>
14 </head>
15 <body>
16     <br><br>
17     <a href="index.php">Inicio</a>
18     <br>
19     <a href="cerrar.php">Eliminar sesión</a>
20 </body>
21 </html>
```

Podemos usar un contador de sesiones añadiendo estos códigos a mi html. }



```
contador.php • index.php • login.php • cerrar.php
sesiones > contador.php
1 <?php
2     session_name("CON");
3     session_start();
4
5     if(isset($_SESSION['contador'])){
6         $_SESSION['contador']++;
7     }else{
8         $_SESSION['contador']=1;
9     }
10 ?>
11 <!DOCTYPE html>
12 <html lang="es">

<body>
<?php echo "Has recargado esta pagina ".$_SESSION['contador']." veces"; ?>
    <form action="login.php" method="POST">
        <label>Usuario</label>
```

Ahora para hacer una simulación de un login deje mi html de la siguiente manera:

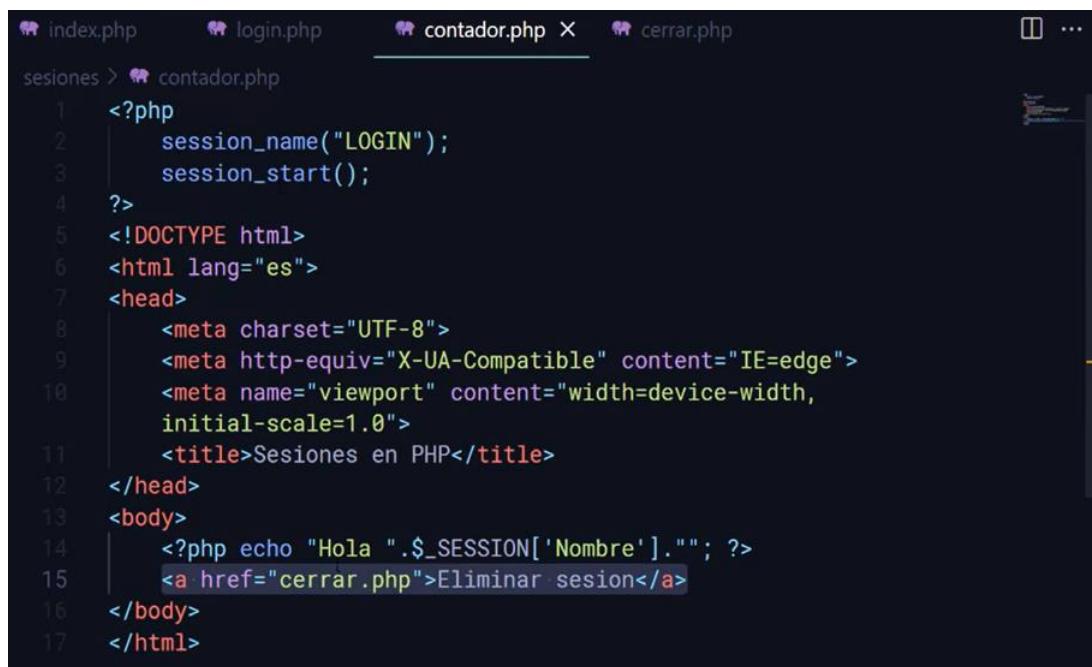
```
index.php X login.php cerrar.php
sesiones > index.php
 7  <head>
 8    <meta charset="UTF-8">
 9    <meta http-equiv="X-UA-Compatible" content="IE=edge">
10    <meta name="viewport" content="width=device-width,
11      initial-scale=1.0">
12    <title>Sesiones en PHP</title>
13  </head>
14  <body>
15    <form action="login.php" method="POST">
16      <label>Usuario</label>
17      <input type="text" name="usuario">
18      <br>
19      <label>Clave</label>
20      <input type="password" name="clave">
21      <br><br>
22      <button type="submit">Login</button>
23    </form>
24  </body>
25 </html>
```

Y en el otro archivo con código php hago lo siguiente, en donde digo que si los datos ingresados en el input con el name usuario son iguales a Carlos, y en el input con el name clave sea igual a 1234, entonces se cree o inicie la sesión con nombre Login, cree 3 variables de sesión y me diga sesión iniciada, sino datos incorrectos.

```
index.php X login.php X contador.php cerrar.php
sesiones > login.php
 1  <?php
 2
 3  if($_POST['usuario']=="Carlos" && $_POST['clave']=="1234"){
 4      session_name("LOGIN");
 5      session_start();
 6
 7      $_SESSION["Nombre"]="Carlos";
 8      $_SESSION["Apellido"]="Alfaro";
 9      $_SESSION["Pais"]="El Salvador";
10
11      echo "Sesion iniciada";
12  }else{
13      echo "Datos incorrectos";
14  }
```

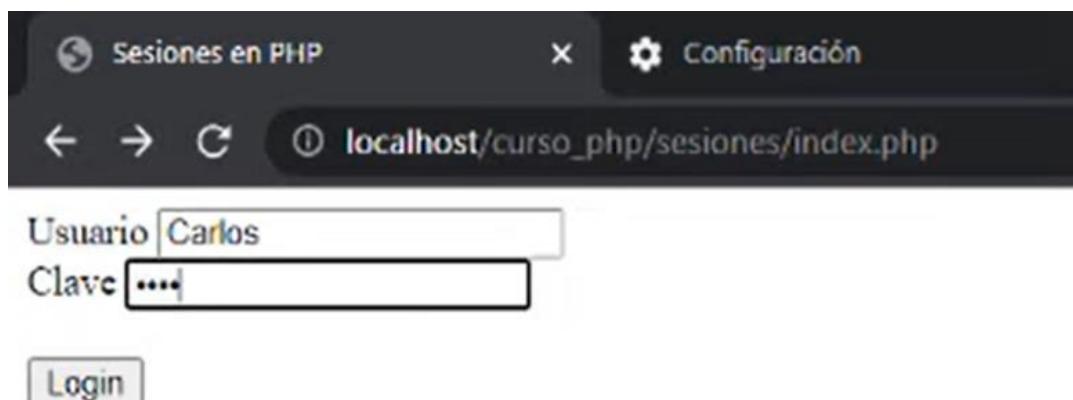
Tenemos que tener el nombre de la sesión y el session\_start( ) para que funcione también en el archivo donde se van a mostrar los datos almacenados en las variables de sesión que

se crearon en el archivo login.php, en este caso mostramos la variable Nombre que sabemos que contiene el valor Carlos.

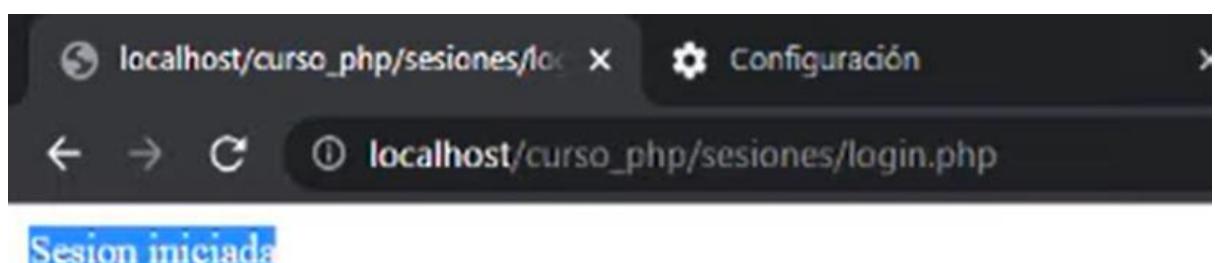


```
index.php login.php contador.php X cerrar.php
sesiones > contador.php
1 <?php
2     session_name("LOGIN");
3     session_start();
4 ?>
5 <!DOCTYPE html>
6 <html lang="es">
7 <head>
8     <meta charset="UTF-8">
9     <meta http-equiv="X-UA-Compatible" content="IE=edge">
10    <meta name="viewport" content="width=device-width,
11        initial-scale=1.0">
12    <title>Sesiones en PHP</title>
13 </head>
14 <body>
15     <?php echo "Hola ".$_SESSION['Nombre']."'"; ?>
16     <a href="cerrar.php">Eliminar sesion</a>
17 </body>
</html>
```

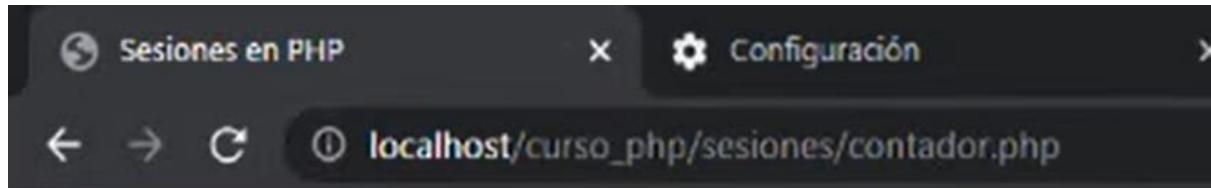
Ahora si coloco Carlos, colocó en el password 1234 y doy clic al botón de Login me dirá sesión iniciada.



Significa que se han creado correctamente los datos o variables de sesión



Si ahora me voy al archivo contador.php observamos que me muestra el valor de sesión correctamente que almacenamos en nuestra variable se sesión llamada Nombre



Hola Carlos [Eliminar sesión](#)

Si ahora coloco Apellido observaremos que también se muestra, o si muestro País también:

```
index.php login.php contador.php cerrar.php
sesiones > contador.php
 7  <head>
 8    <meta charset="UTF-8">
 9    <meta http-equiv="X-UA-Compatible" content="IE=edge">
10    <meta name="viewport" content="width=device-width,
11      initial-scale=1.0">
12    <title>Sesiones en PHP</title>
13  </head>
14  <body>
15    <?php echo "Hola ".$_SESSION['Apellido']. """; ?>
16    <a href="cerrar.php">Eliminar sesión</a>
17  </body>
18 </html>
```

## Como eliminar una sesión:

Para eso tenemos la etiqueta `<a/>` que es un enlace que al presionarlo nos va a llevar al archivo que tenemos llamado cerrar.php

```
index.php login.php contador.php cerrar.php
sesiones > contador.php
 1  <?php
 2    session_name("LOGIN");
 3    session_start();
 4  ?>
 5  <!DOCTYPE html>
 6  <html lang="es">
 7    <head>
 8      <meta charset="UTF-8">
 9      <meta http-equiv="X-UA-Compatible" content="IE=edge">
10      <meta name="viewport" content="width=device-width,
11        initial-scale=1.0">
12      <title>Sesiones en PHP</title>
13    </head>
14    <body>
15      <?php echo "Hola ".$_SESSION['Apellido']. """; ?>
16      <a href="cerrar.php">Eliminar sesión</a>
17    </body>
18  </html>
```

Recordemos para usar las variables de sesión o trabajar en la misma sesión dentro de otro archivo es necesario colocar el nombre y el `session_start()` esto solo en caso de que

hayamos cambiado el nombre como lo hicimos en este ejemplo que le colocamos LOGIN, sino lo cambiamos solo es necesario el session\_start()

Ahora para eliminar una sesión, ocupamos la función session\_destroy() la cual elimina toda la información registrada, incluyendo variables de sesión y el id de la sesión (el valor que aparece al inspeccionar la cookie de muchos números y letras)



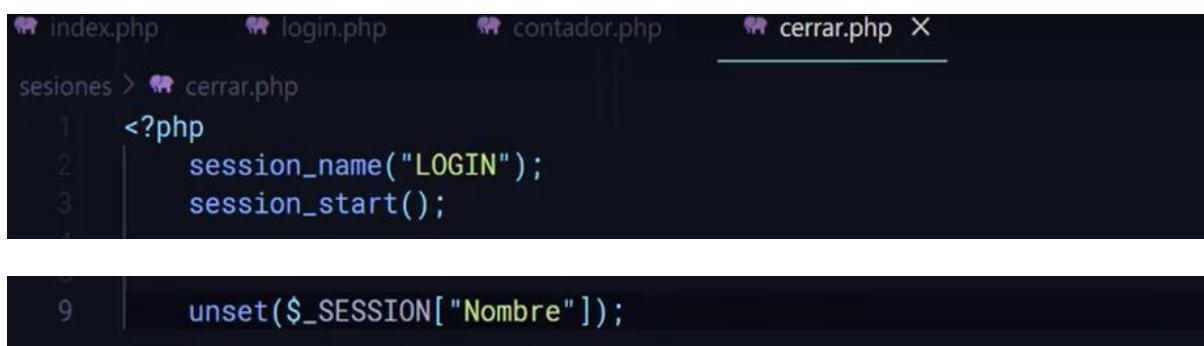
```
index.php login.php contador.php cerrar.php
sesiones > cerrar.php
1 <?php
2     session_name("LOGIN");
3     session_start();
4
5     session_destroy();
```

O también podemos utilizar la función session\_unset() que lo que hace es únicamente eliminar las variables de sesión, pero en los login reales lo que se ocupa es session\_destroy() para eliminar completamente la sesión.



```
index.php login.php contador.php cerrar.php
sesiones > cerrar.php
1 <?php
2     session_name("LOGIN");
3     session_start();
4
5     session_unset();
```

O si queremos eliminar únicamente una variable de sesión podemos hacerlo con unset() colocando dentro la variable de sesión que queremos eliminar de la siguiente manera :



```
index.php login.php contador.php cerrar.php
sesiones > cerrar.php
1 <?php
2     session_name("LOGIN");
3     session_start();
4
9     unset($_SESSION["Nombre"]);
```

Al trabajar con php y mysql en un login real los datos se extraen de una BD y se comparan con los que escribimos en el formulario con los que queremos iniciar sesión.

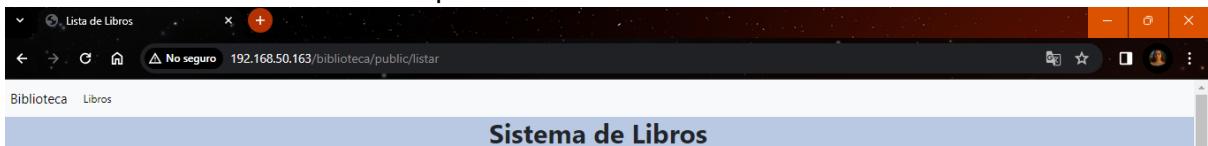


# Menú

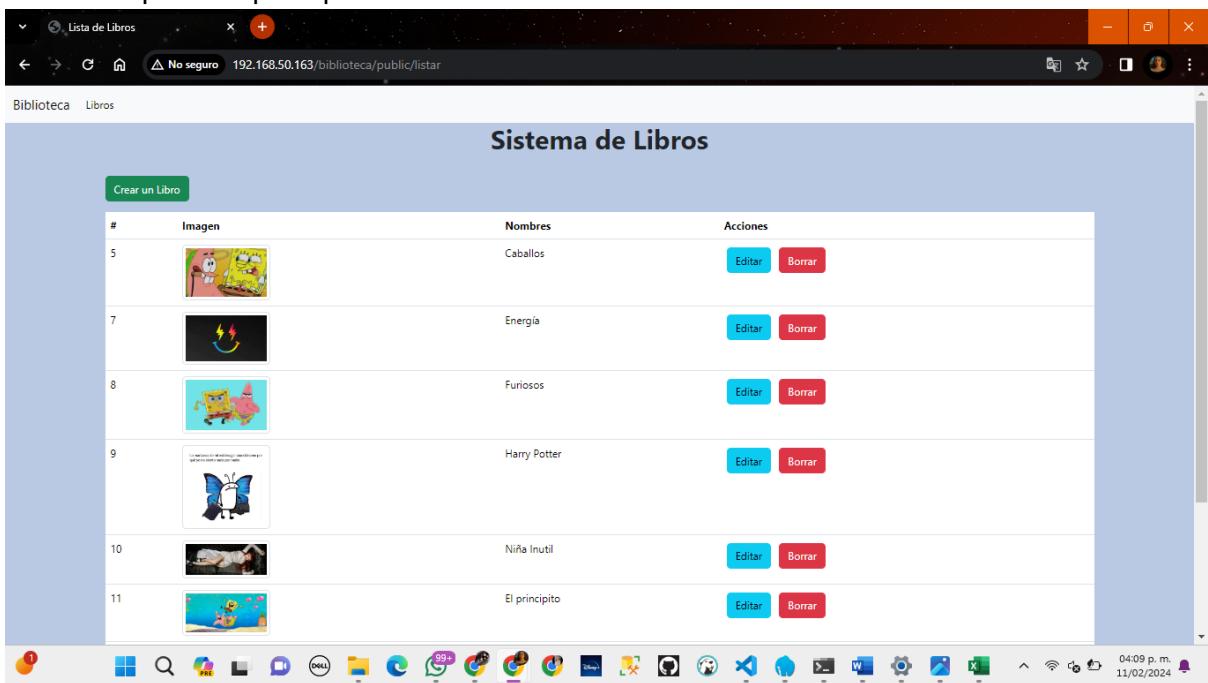
Para realizar un menú, en este ejemplo, solo se ocupará de tener bootstrap y listo. En nuestra cabecera vamos a insertar un nav.

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Biblioteca</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="=base_url('listar');?">Libros</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

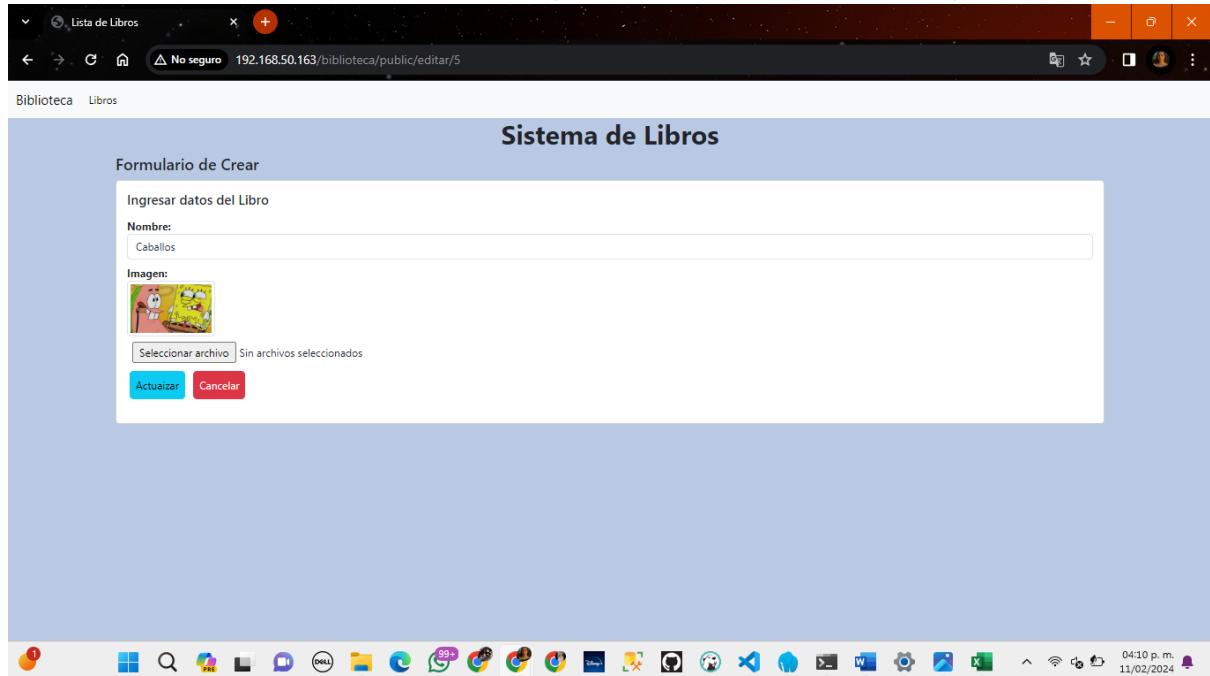
Así se ve nuestro menú en una aplicación web.



Lo único que hace es dirigirnos a una pantalla, según la opción que eligió el usuario. Esta es la pantalla principal:



Cuando nos vamos a editar a un libro, lo que hace es irnos a una nueva pantalla:



Pero si le damos clic a la opción de Libros nos redirige a la página principal.

**Sistema de Libros**

Formulario de Crear

Ingresar datos del Libro

Nombre: Caballos

Imagen:

Seleccionar archivo Sin archivos seleccionados

Actualizar Cancelar

**Sistema de Libros**

Crear un Libro

#	Imagen	Nombres	Acciones
5		Caballos	Editar Borrar
7		Energía	Editar Borrar
8		Furiosos	Editar Borrar
9		Harry Potter	Editar Borrar
10		Niña Inutil	Editar Borrar
11		El principito	Editar Borrar

Al ponernos sobre la opción, abajo de la pantalla nos aparecerá el link al que nos dirigirá.

