

Universidad de San Carlos de Guatemala
Centro Universitario de Occidente
División de Ciencias de la Ingeniería
Teoría de Sistemas 1
Ing. Pedro Domingo



“DOCUMENTACIÓN TÉCNICA”

Alan Giovany Morales Tay

Carné: 201830731

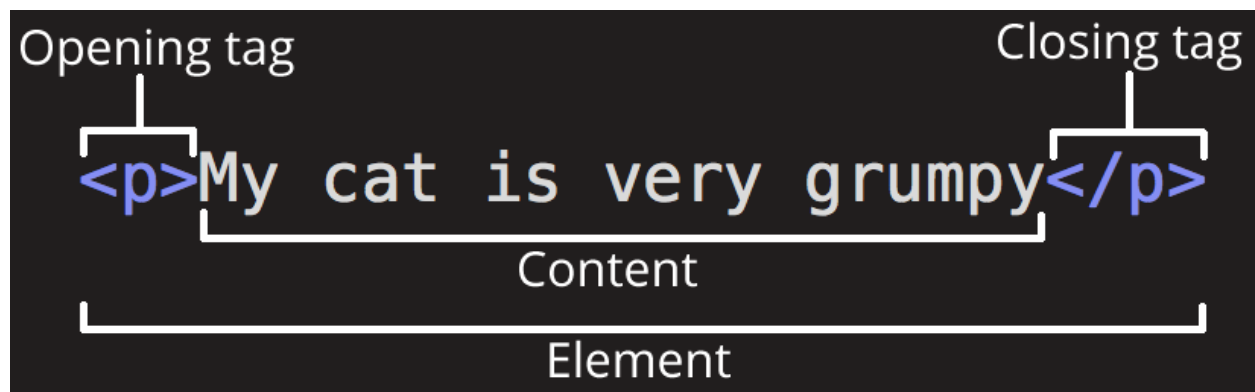
Enlace de repositorio: <https://github.com/Alan-MT/ProyectoTorismo>

23 de Agosto de 2023

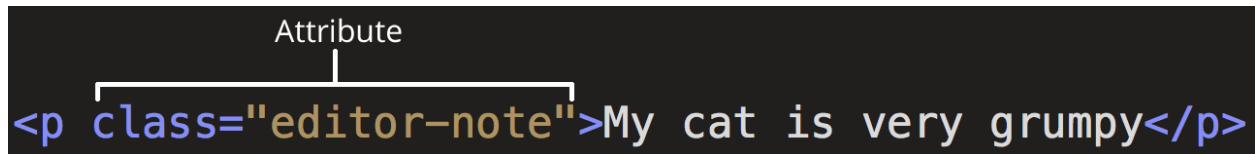
HTML

El Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos. Como lo sugiere el título, este artículo te dará una comprensión básica de HTML y cuál es su función.

Anatomía de un elemento HTML



1. La etiqueta de apertura: consiste en el nombre del elemento (en este caso, p), encerrado por paréntesis angulares (< >) de apertura y cierre. Establece dónde comienza o empieza a tener efecto el elemento —en este caso, dónde es el comienzo del párrafo—.
2. La etiqueta de cierre: es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (/) antes del nombre de la etiqueta. Establece dónde termina el elemento —en este caso dónde termina el párrafo—.
3. El contenido: este es el contenido del elemento, que en este caso es sólo texto.
4. El elemento: la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivale al elemento.



Los atributos contienen información adicional acerca del elemento, la cual no quieres que aparezca en el contenido real del elemento. Aquí `class` es el nombre del atributo y `editor-note` el valor del atributo. En este caso, el atributo `class` permite darle al elemento un nombre identificativo, que se puede utilizar luego para apuntarle al elemento información de estilo y demás cosas.

Un atributo debe tener siempre:

1. Un espacio entre este y el nombre del elemento (o del atributo previo, si el elemento ya posee uno o más atributos).
2. El nombre del atributo, seguido por un signo de igual (=).
3. Comillas de apertura y de cierre, encerrando el valor del atributo.

Los atributos siempre se incluyen en la etiqueta de apertura de un elemento, nunca en la de cierre.

Qué es CSS

CSS son las siglas en inglés para «hojas de estilo en cascada» (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de los sitios.

Se les denomina hojas de estilo «en cascada» porque puedes tener varias y una de ellas con las propiedades heredadas (o «en cascada») de otras.

Para muchas personas, una simple plantilla de blog es suficiente. Aun así, cuando quieras personalizar la apariencia de un sitio, necesitarás implementar CSS que, en conjunto con un buen [CMS](#), te ayudará a potenciar el alcance de tu contenido.

Para qué sirve CSS

Con CSS, puedes crear reglas para decirle a tu sitio web cómo quieres mostrar la información y guardar los comandos para elementos de estilo (como fuentes, colores, tamaños, etc.) separados de los que configuran el contenido.

Además, puedes crear formatos específicos útiles para comunicar tus ideas y producir experiencias más agradables, en el aspecto visual, para los usuarios del sitio web.

1. CSS externo

El CSS externo existe en su propio archivo, el cual está enlazado a un documento HTML con una etiqueta `<link>`. Es el método más usado para añadir CSS a HTML, ya que una hoja de estilo externa puede dictar el estilo de diferentes documentos HTML. Esto permite que los desarrolladores hagan cambios a lo ancho del sitio con un solo archivo CSS.

Para crear un archivo CSS, escribe tu código en cualquier editor de texto o editor de código y guarda el archivo con la extensión `.css` para vincularlo con un archivo de HTML; almacena tus HTML y CSS en la misma carpeta y luego, pega el siguiente código dentro de la sección `<head>` del archivo HTML:

```
<link rel="stylesheet" target="_blank" rel="noopener" href="style.css">
```

"Agrega en «style.css» el nombre de tu archivo CSS. Las reglas de este archivo se aplicarán a cualquier archivo HTML que haga referencia a él con el elemento <link> de arriba.



```
HTML
1 <head>
2   <link rel="stylesheet"
3     href="style.css">
4 </head>
5 <body>
6   <div>Soy un div con CSS externo.
7 </div>
8 </body>

CSS
1 /* style.css */
2
3 div {
4   background-color: #FD8F62;
5   width: 300px;
6   text-align: center
7 }
```

2. CSS Interno

El CSS interno es un código CSS incrustado en un documento HTML. Está escrito dentro del elemento <style>, que se acomoda en la sección de <head>:



```
HTML
1 <head>
2   <style>
3     div {
4       background-color: #FD8F62;
5       width: 300px;
6       text-align: center
7     }
8   </style>
9 </head>
10
11 <body>
12   <div>Soy un div con CSS interno.
13 </div>
14 </body>
```

El CSS interno puede utilizarse para algunos proyectos pequeños de web y páginas individuales con su propio estilo. En estos casos, será más fácil guardar todo tu código en el mismo archivo y no en dos.

3. CSS en línea

CSS en línea se ubica dentro de una etiqueta HTML para cambiar el estilo de un elemento en específico. La sintaxis inline es un poco diferente de lo que hemos visto, pues la declaración está escrita como el valor del atributo de estilo.

A screenshot of a code editor with a dark theme. The editor shows a single line of HTML code: `<div style="background-color: #FD8F62; width: 300px; text-align: center">Soy un div con CSS en línea o inline.</div>`. The text "Soy un div con CSS en línea o inline." is displayed in the center of the editor, with a light orange background color. To the right of the code editor, there is a small orange box containing the text "Soy un div con CSS en línea o inline." in white. The code editor has a tab labeled "HTML" and a dropdown arrow.

Aunque es posible hacerlo, usar este método viola la práctica de separar el estilo del contenido, por eso no se recomienda. El CSS en línea es ineficiente para programar y resulta más difícil de comprender que los otros dos. Aun así, es bueno que lo conozcas para que lo puedas identificar, si se presenta el caso.

¿Qué es JavaScript?

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable “click-to-show” (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

¿Para qué se utiliza JavaScript?

Anteriormente, las páginas web eran estáticas, similares a las páginas de un libro. Una página estática mostraba principalmente información en un diseño fijo y no todo aquello que esperamos de un sitio web moderno. JavaScript surgió como una tecnología del lado del navegador para hacer que las aplicaciones web fueran más dinámicas. Por medio de JavaScript, los navegadores eran capaces de responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.

A medida que el lenguaje evolucionó, los desarrolladores de JavaScript establecieron bibliotecas, marcos y prácticas de programación y comenzaron a utilizarlo fuera de los navegadores web. En la actualidad, puede utilizar JavaScript para el desarrollo tanto del lado del cliente como del lado del servidor. En las siguientes subsecciones se presentan algunos casos de uso comunes:

Anteriormente, las páginas web eran estáticas, similares a las páginas de un libro. Una página estática mostraba principalmente información en un diseño fijo y no todo aquello que esperamos de un sitio web moderno. JavaScript surgió como una tecnología del lado del navegador para hacer que las aplicaciones web fueran más dinámicas. Por medio de JavaScript, los navegadores eran capaces de responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.

A medida que el lenguaje evolucionó, los desarrolladores de JavaScript establecieron bibliotecas, marcos y prácticas de programación y comenzaron a utilizarlo fuera de los navegadores web. En la actualidad, puede utilizar JavaScript para el desarrollo tanto del lado del cliente como del lado del servidor. En las siguientes subsecciones se presentan algunos casos de uso comunes:

MYSQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basado en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle.

Las versiones Enterprise, diseñadas para aquellas empresas que quieran incorporarlo en productos privativos, incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial.

Principales sentencias de MySQL

MySQL comparte las sentencias del lenguaje SQL. Lo que en la práctica se traduce en una plena compatibilidad. Al ser un elemento muy ligado al backend de cualquier aplicación web, normalmente desarrollada sobre algún lenguaje como PHP, puede ser un poco difícil de ejemplificar pues sería necesario abarcar todo un artículo pero se puede realizar la siguiente relación de sentencias básicas:

- **SELECT** es usada para consultar datos.
- **DISTINCT** Sirve para eliminar los duplicados de las consultas de datos.
- **WHERE** Es usada para incluir las condiciones de los datos que queremos consultar.
- **AND y OR** es usada para incluir 2 o más condiciones a una consulta.
- **ORDER BY** Es usada para ordenar los resultados de una consulta.
- **INSERT** Es usada para insertar datos.

- UPDATE Es usada para actualizar o modificar datos ya existentes.
- DELETE Es usada para borrar datos.

PHP

PHP es un lenguaje de programación de código abierto del lado del servidor que se utiliza principalmente para crear páginas web dinámicas. La abreviatura nació originariamente de “Personal Home Page Tools”, aunque hoy en día se ha convertido en el acrónimo recursivo para “PHP: Hypertext Preprocessor”.

Mientras que los lenguajes del lado del cliente como HTML, CSS o JavaScript son interpretados primero por el navegador web en el momento de abrir una página, el código PHP se ejecuta en el servidor web. Allí, los scripts de PHP generan el código HTML que se envía después al navegador. Este no recibe el código real (el script de PHP), sino el resultado de la ejecución del mismo.

El ámbito de aplicación principal de PHP es la programación del lado del servidor, sobre todo de páginas dinámicas y aplicaciones. Otras áreas de implementación son la creación de aplicaciones de escritorio o la programación de líneas de comandos. A pesar de tener una sintaxis sencilla para principiantes, PHP ofrece una cantidad de funciones remarcable. Este lenguaje de programación se distingue por su amplio soporte a bases de datos, puede utilizarse en todo tipo de plataformas y está cubierto por una licencia PHP especial que permite su libre utilización y modificación del código fuente, una combinación más que convincente.

XAMPP

XAMPP es una distribución de Apache que incluye varios software libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen: el servidor web Apache, los sistemas relacionales de administración de bases de datos MySQL y MariaDB, así como los lenguajes de programación Perl y PHP. La inicial X se usa para representar a los sistemas operativos Linux, Windows y Mac OS X.

Apache: el servidor web de código abierto es la aplicación más usada globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como software libre por la Apache Software Foundation.

MySQL/MariaDB: y MySQL, XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web. En las versiones actuales de XAMPP esta base de datos se ha sustituido por MariaDB, una ramificación (“Fork”) del proyecto MySQL.

PHP: Es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente de plataforma y soporta varios sistemas de bases de datos.

Perl: este lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red. También permite programar aplicaciones web dinámicas.

Además de estos componentes principales, esta distribución gratuita también incluye, según el sistema operativo, otras herramientas como el servidor de correo Mercury, el programa de administración de bases de datos phpMyAdmin, el software de analítica web Webalizer, OpenSSL, Apache Tomcat y los servidores FTP FileZilla o ProFTPd.

Instalar XAMPP

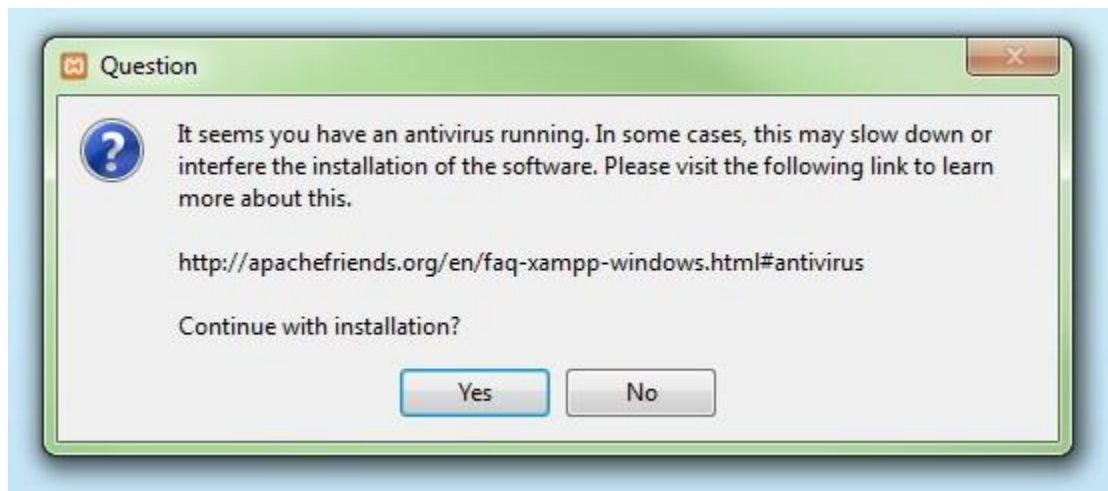
Paso 1: Descarga las versiones con PHP 5.5, 5.6 o 7 se pueden descargar gratuitamente desde la página del proyecto Apache Friends.

Paso 2: Ejecutar el archivo .exe

Una vez descargado el paquete, puedes ejecutar el archivo .exe haciendo doble clic en él.

Paso 3: Desactivar el programa antivirus

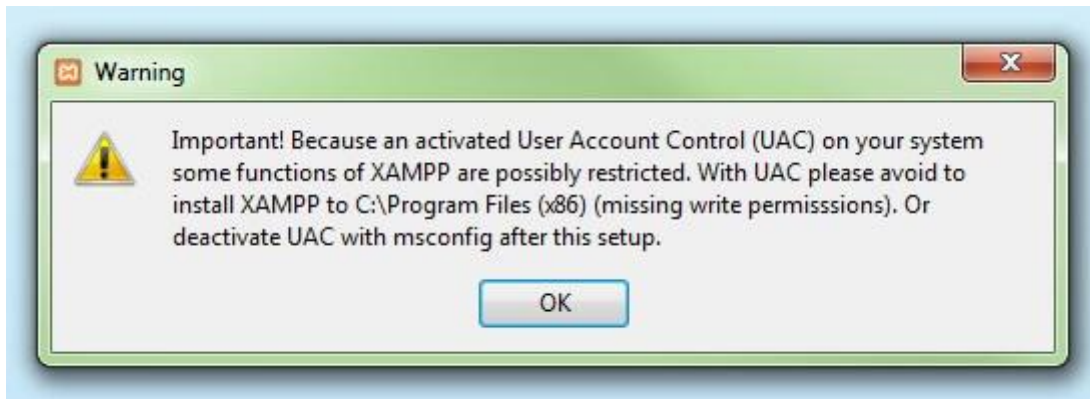
Se recomienda desactivar el programa antivirus hasta que todos los componentes estén instalados, ya que puede obstaculizar el proceso de instalación.



Antes de iniciar la instalación de XAMPP es recomendable desactivar temporalmente el antivirus

Paso 4: Desactivar el UAC

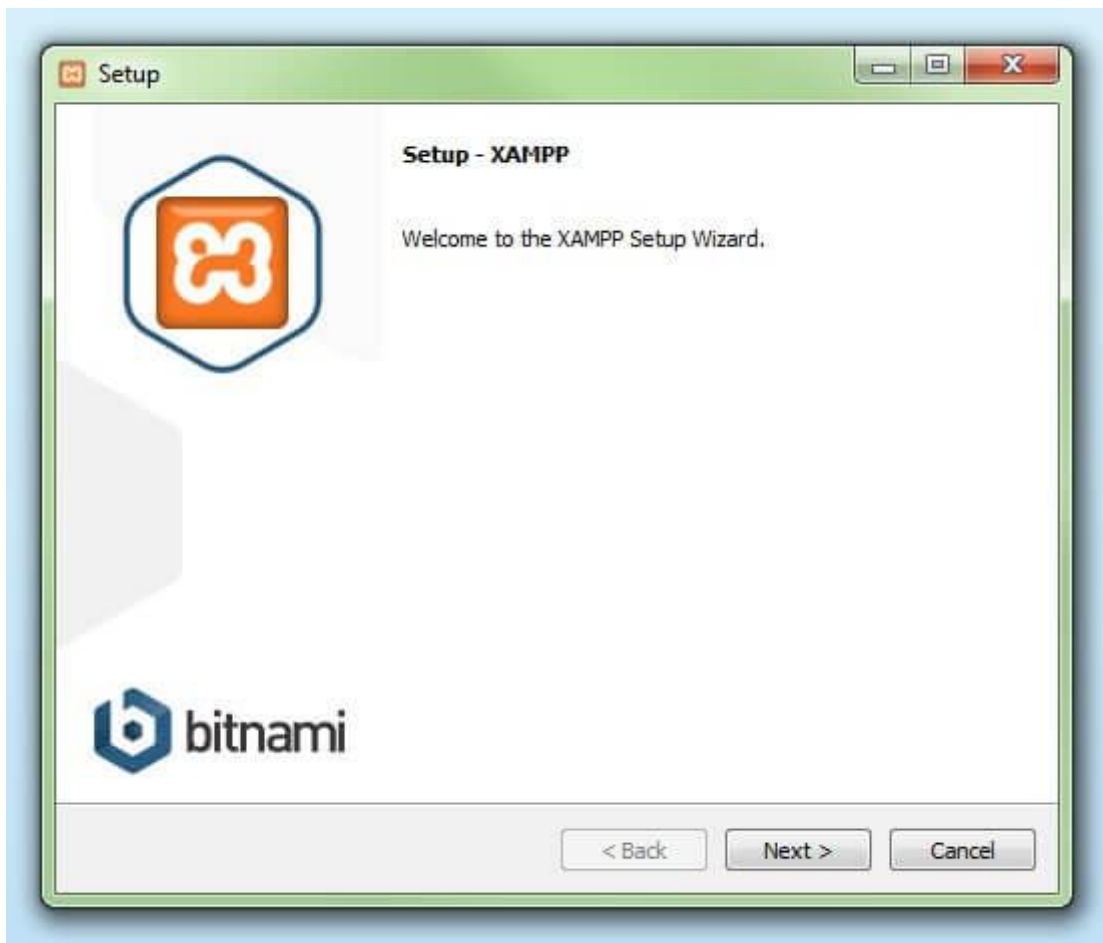
También el control de cuentas de usuario (User Account Control, UAC) puede interferir en la instalación, ya que limita los derechos de escritura en la unidad de disco C:\. Para saber cómo desactivar temporalmente el UCA puedes dirigirte a las páginas de soporte de Microsoft.



También el Control de cuentas de usuarios (UAC) puede impedir la instalación de XAMPP

Paso 5: Iniciar el asistente de instalación

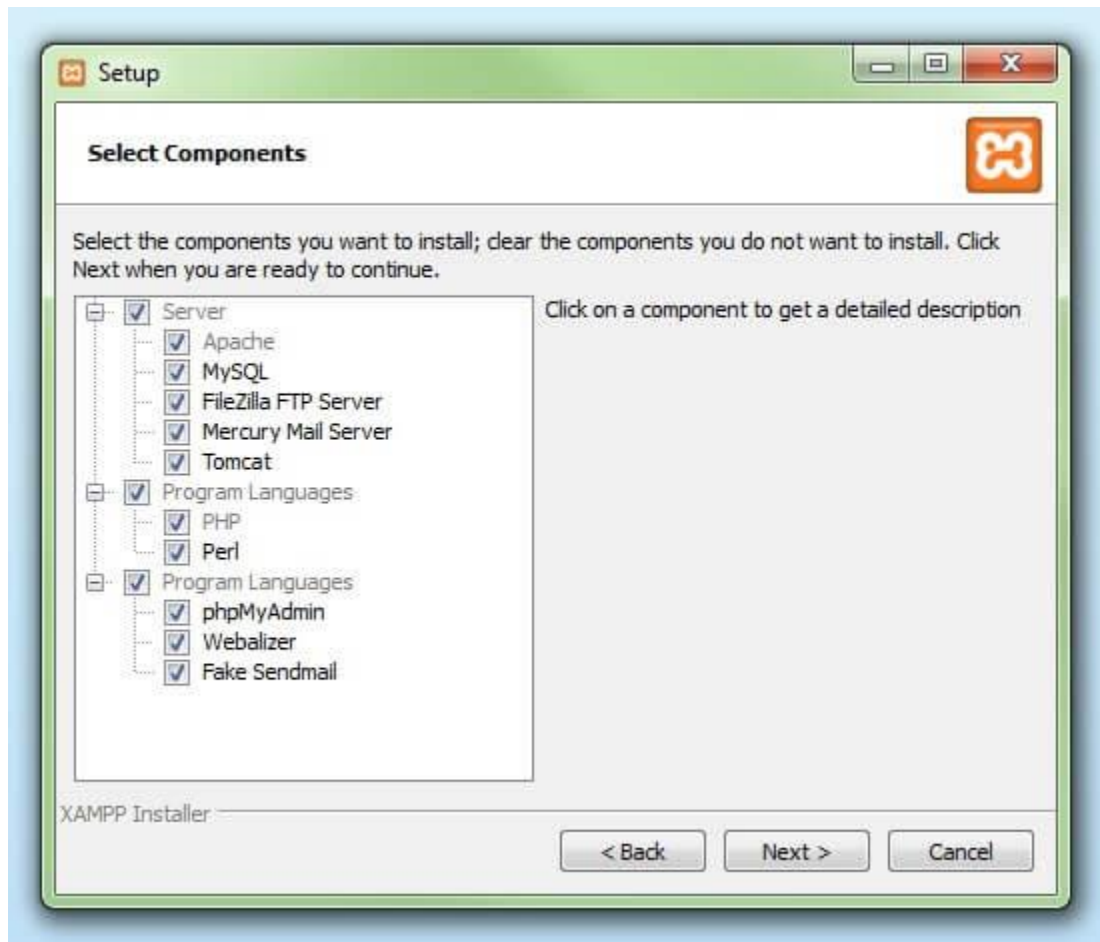
Una vez superados estos pasos, aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en "Next".



Con la aparición de la pantalla de inicio del asistente da comienzo la instalación de XAMPP

Paso 6: Selección de los componentes del software

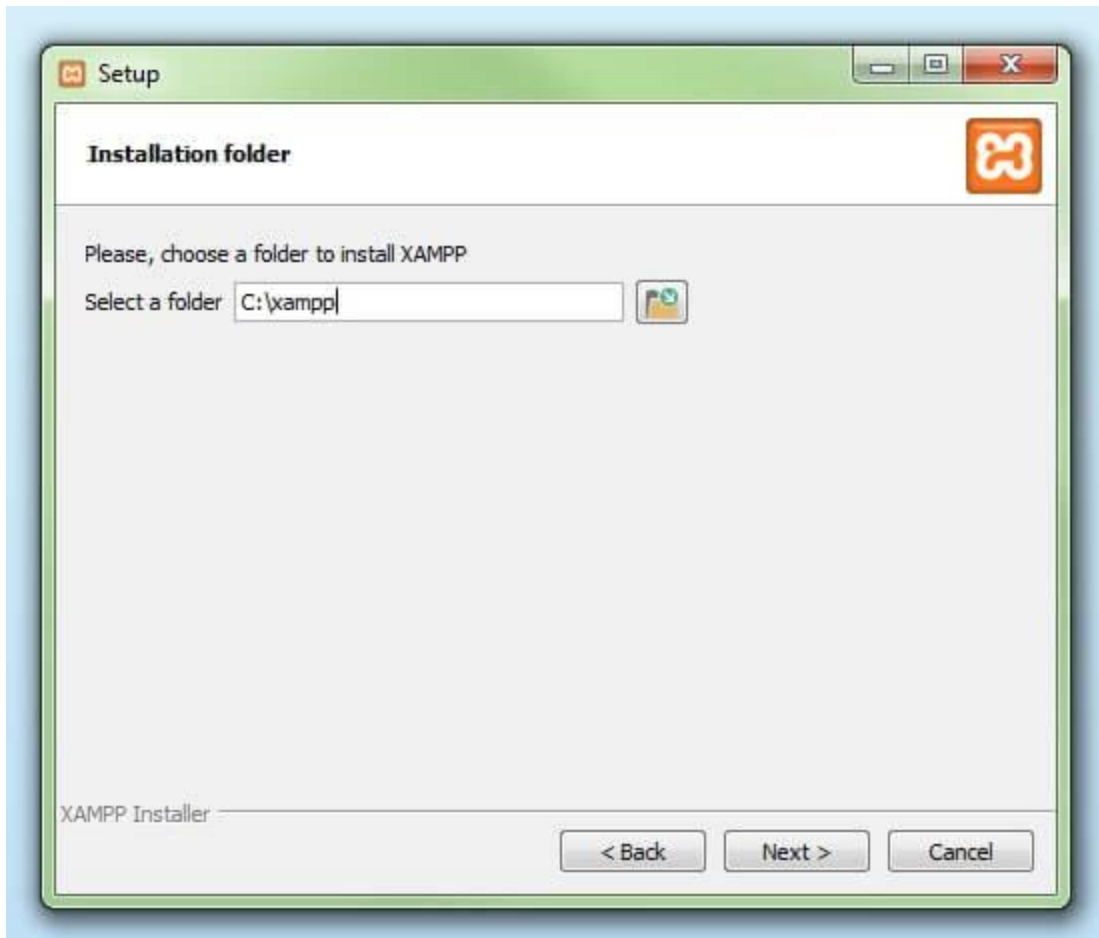
En la rúbrica “Select components” se pueden excluir de la instalación componentes aislados del paquete de software de XAMPP. Se recomienda la configuración estándar para un servidor de prueba local, con la cual se instalan todos los componentes disponibles. Confirma la selección haciendo clic en “Next”.



En el cuadro de diálogo “Select Components” se pueden seleccionar o deseleccionar los componentes que se instalarán

Paso 7: Selección del directorio para la instalación

En este paso se escoge el directorio donde se instalará el paquete. Si se ha escogido la configuración estándar se creará una carpeta con el nombre XAMPP en C:\.



En un siguiente paso, se selecciona el directorio donde se instalarán los archivos

Paso 8: Iniciar el proceso de instalación

El asistente extrae los componentes seleccionados y los guarda en el directorio escogido en un proceso que puede durar algunos minutos. El avance de la instalación se muestra como una barra de carga de color verde.

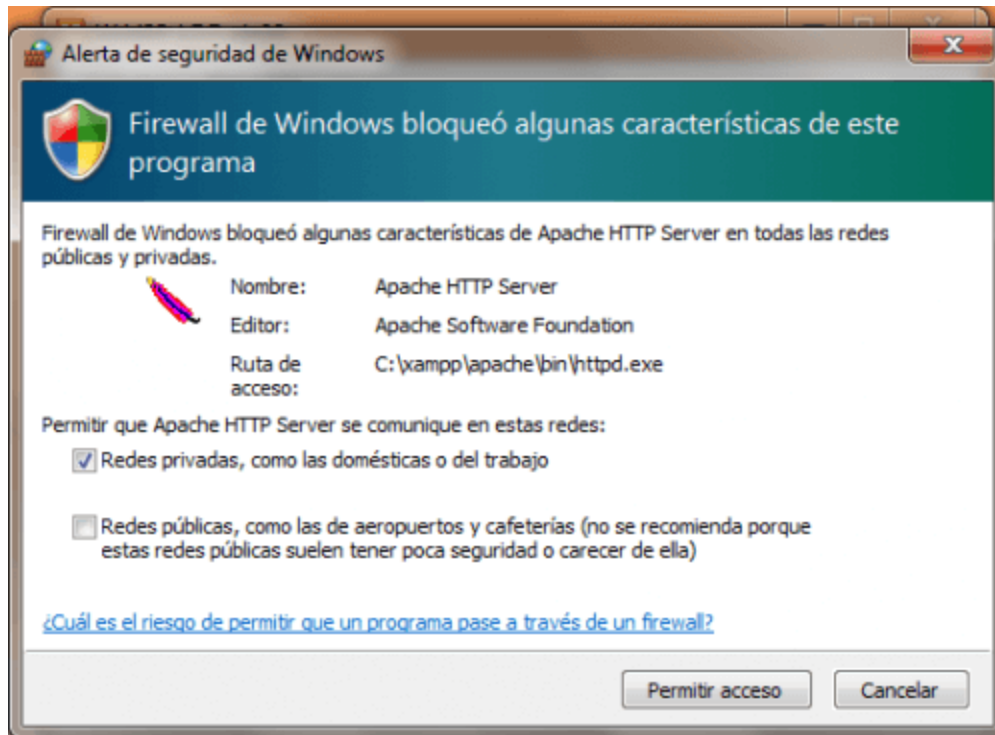


A

continuación da comienzo el proceso de instalación en el cual se descomprimen los elementos de software seleccionados y se instalan en el directorio que se ha definido en los preajustes

Paso 9: Configurar Firewall

Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública.



Durante la instalación será necesario reconfigurar el cortafuegos para que no bloquee componentes del servidor Apache

Paso 10: Cerrar la instalación

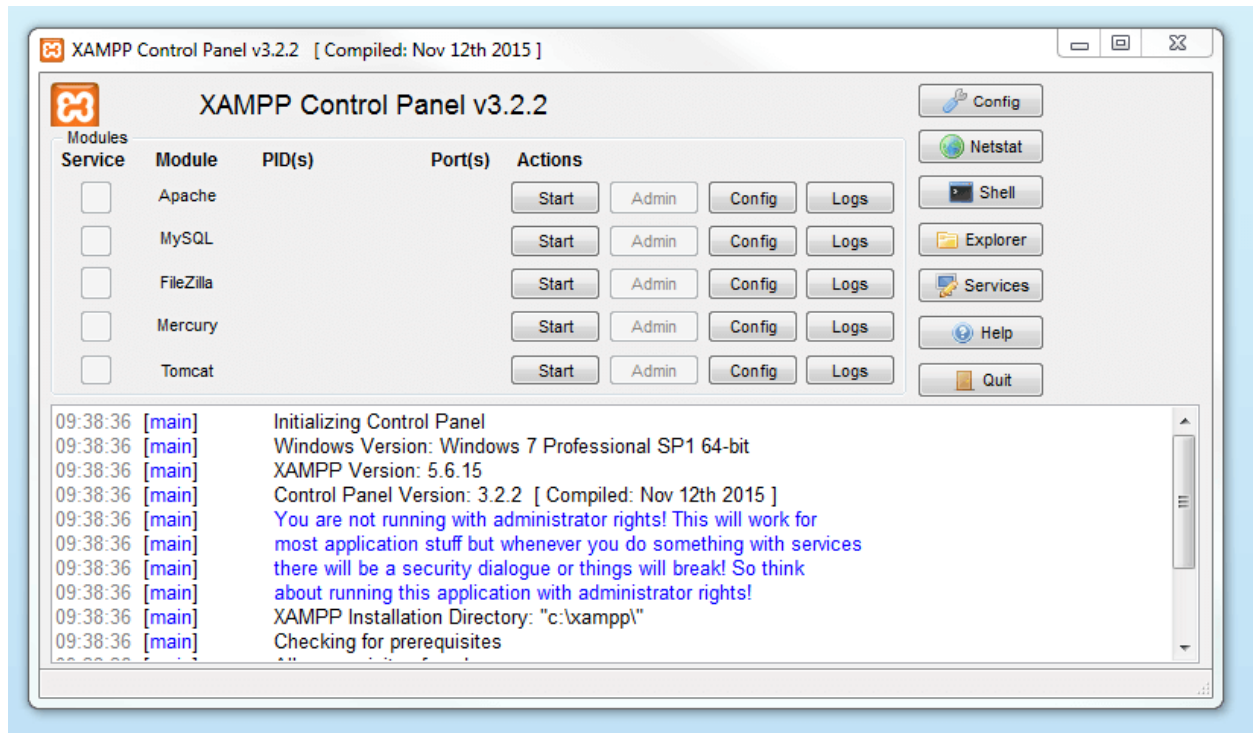
Una vez extraídos e instalados todos los componentes puedes cerrar el asistente con la tecla “Finish”. Para acceder inmediatamente al panel de control solo es necesario marcar la casilla que pregunta si deseamos hacerlo.



Haciendo clic en "Finish" se cierra el asistente de instalación de XAMPP
Panel de control de XAMPP

En la diáfana interfaz de usuario del panel de control se protocolan todas las acciones y es posible activar o desactivar los módulos por separado con un simple clic. Además, se dispone de diversas utilidades como:

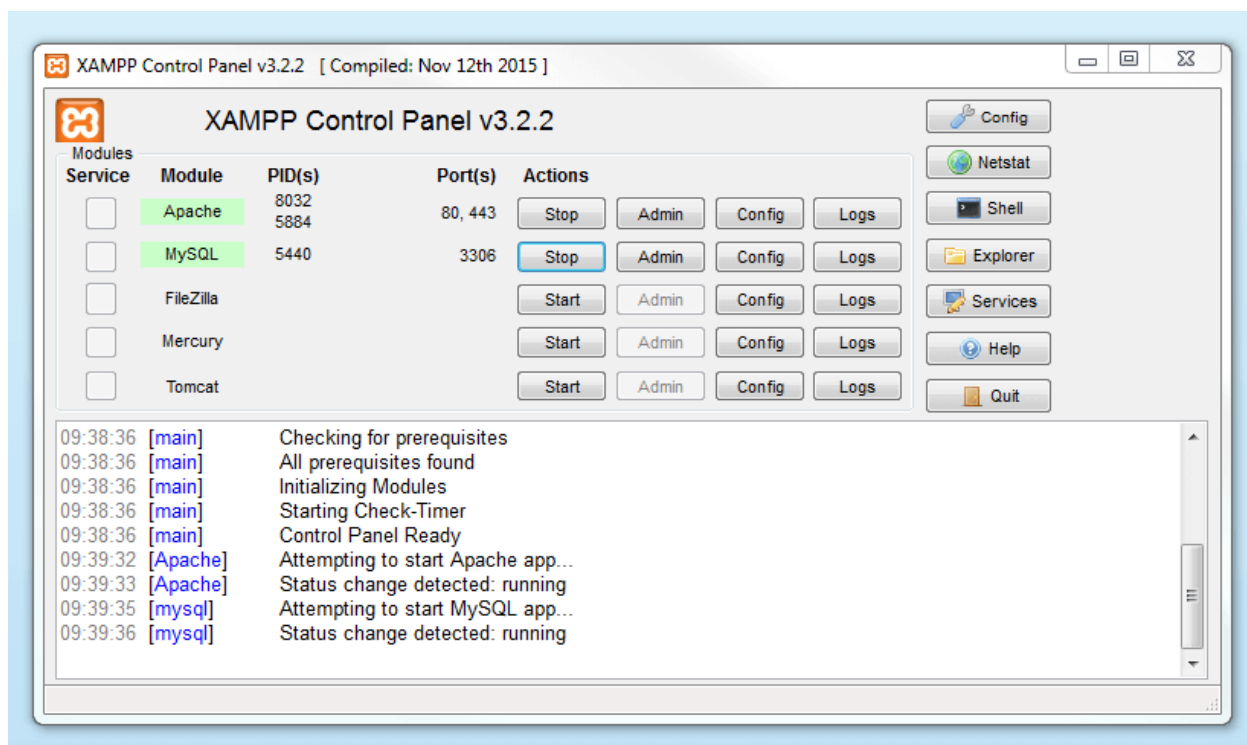
- Config: para configurar XAMPP así como otros componentes aislados.
- [Netstat](#): muestra todos los procesos en funcionamiento en el ordenador local
- Shell: lanza una ventana de comandos UNIX
- Explorer: abre la carpeta XAMPP en el explorador de Windows
- Services: muestra todos los servicios en funcionamiento
- Help: incluye enlaces a foros de usuarios
- Quit: se usar para salir del panel de control



En el Panel de Control el usuario puede iniciar o finalizar los diferentes módulos por separado

Iniciar módulos

En la parte superior se pueden iniciar o interrumpir los módulos de XAMPP por separado mediante los comandos “Start” y “Stop” bajo “Actions”. Los módulos que se activaron aparecen marcados en verde.



Los módulos activos aparecen en el Panel de Control marcados en verde. Si uno de ellos no pudiera ser iniciado por un error, se mostrará marcado en rojo. Las notificaciones de error protocolizadas en la parte de abajo ayudan a encontrar las causas del error.

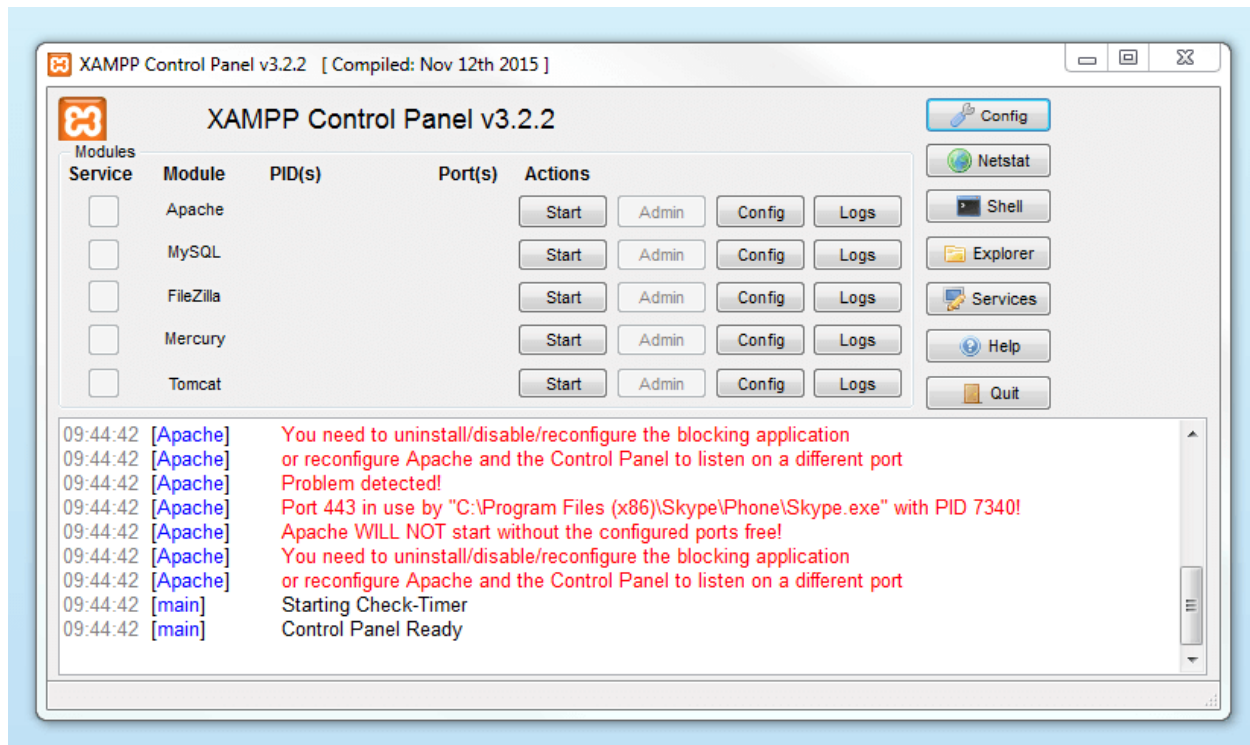
Ajustar XAMPP

Una causa frecuente de fallos en el uso de Apache es un puerto bloqueado. La configuración estándar XAMPP suele asignar al servidor web el puerto principal 80 y el puerto SSL 443, pero suelen estar bloqueados por otros programas. En la figura anterior se muestra un conflicto con el programa de mensajería instantánea Skype, que está usando los puertos 80 y 443, por lo que el servidor no se puede iniciar. Existen tres posibles soluciones:

- Cambiar el puerto en Skype: para ello se abre Skype y en “Herramientas” > “Opciones” > “Avanzada” > “Conexión” se desmarca la casilla donde se indica “Usar los puertos 80 y 443 para las conexiones entrantes adicionales”.
- Cambiar las configuraciones de los puertos en Apache: haz clic en la configuración del módulo Apache (“Config”) y abre los archivos httpd.conf y httpd-ssl.conf. En httpd.conf cambia el puerto 80 por el puerto que deseas

usar y lo mismo se realiza en httpd-ssl.conf con el puerto 443. Guarda los archivos. En la tecla “Config” de la derecha selecciona “Ajustar puertos” para poder introducir los cambios realizados en los archivos conf.

c. Finalizar Skype: el método más sencillo de evitar conflictos con Skype consiste en cerrarlo y volver a iniciarlo solo cuando el servidor Apache ya esté en funcionamiento.



Las notificaciones de error aparecen en rojo en el Panel de Control con indicaciones para resolver el conflicto


Administrar los módulos

Para cada módulo existe una función “Admin”.

- Al hacer clic en la tecla “Admin” del servidor Apache, accederás a la dirección web del servidor a través del navegador estándar del equipo. Serás redirigido a la página principal de XAMPP en el local host, el dominio de la máquina local. El dashboard incluye multitud de enlaces a páginas web con información útil así como al proyecto open source BitNami, que ofrece diversas aplicaciones gratuitas para XAMPP como WordPress u otros CMS. Para acceder a esta página principal se introduce la dirección localhost/dashboard/.

Apache Friends

ApplicationsFAQsHOW-TO GuidesPHPInfophpMyAdmin

 **XAMPP** Apache + MariaDB + PHP + Perl

Welcome to XAMPP for Windows 5.6.15

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

Start the XAMPP Control Panel to check the server status.

Community






XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#), adding yourself to the [Mailing List](#), and liking us on [Facebook](#), following our exploits on [Twitter](#), or adding us to your [Google+](#) circles.




Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We have set up a site, translate.apachefriends.org, where users can contribute translations.

Install applications on XAMPP using Bitnami

Apache Friends and Bitnami are cooperating to make dozens of open source applications available on XAMPP, for free. Bitnami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the [Bitnami XAMPP page](#) for details on the currently available apps.



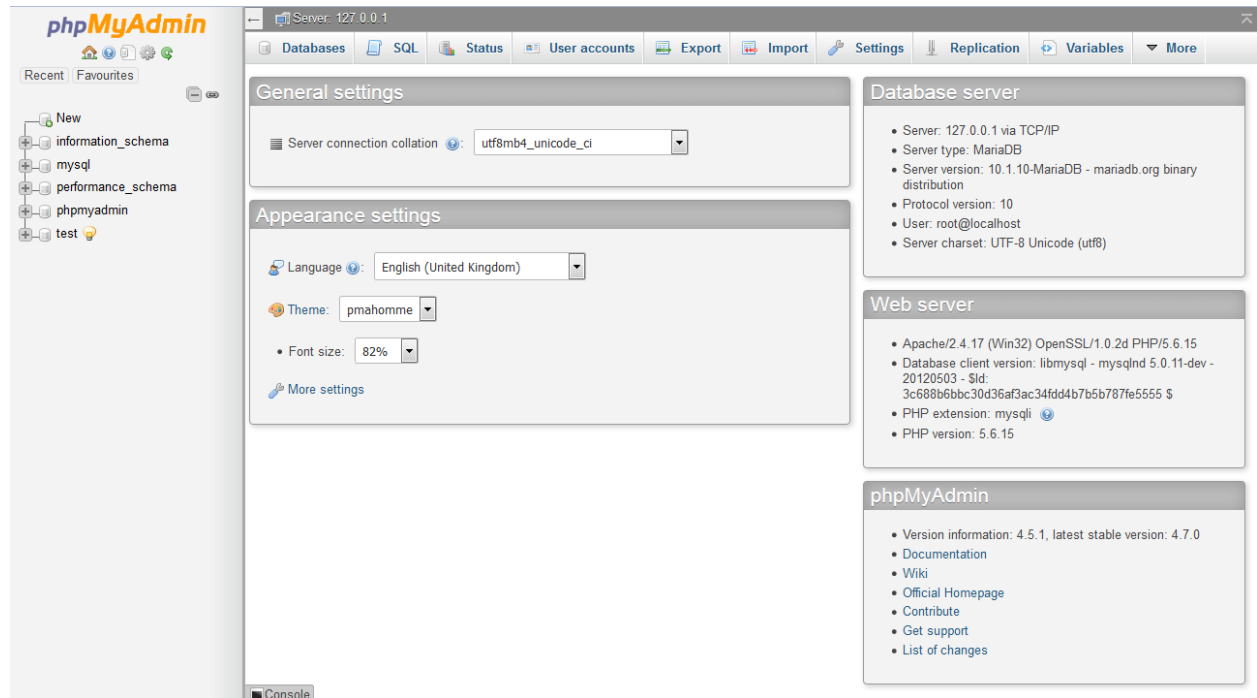


[Blog](#) [Privacy Policy](#) [CDN provided by **fastly**](#)

Copyright (c) 2015, Apache Friends

Haciendo clic en el botón “Admin” del módulo Apache el usuario es redirigido al dashboard local de XAMPP

- Haciendo clic en la tecla “Admin” de la base de datos se abre phpMyAdmin, donde se pueden administrar las bases de datos del proyecto web que se quiere probar con XAMPP. También podemos acceder a la interfaz de administración para la base de datos MySQL en `localhost/phpmyadmin/`.



Las bases de datos se gestionan en phpMyAdmin, accesible en el botón “Admin” del módulo Base de datos

Diagrama de Despliegue

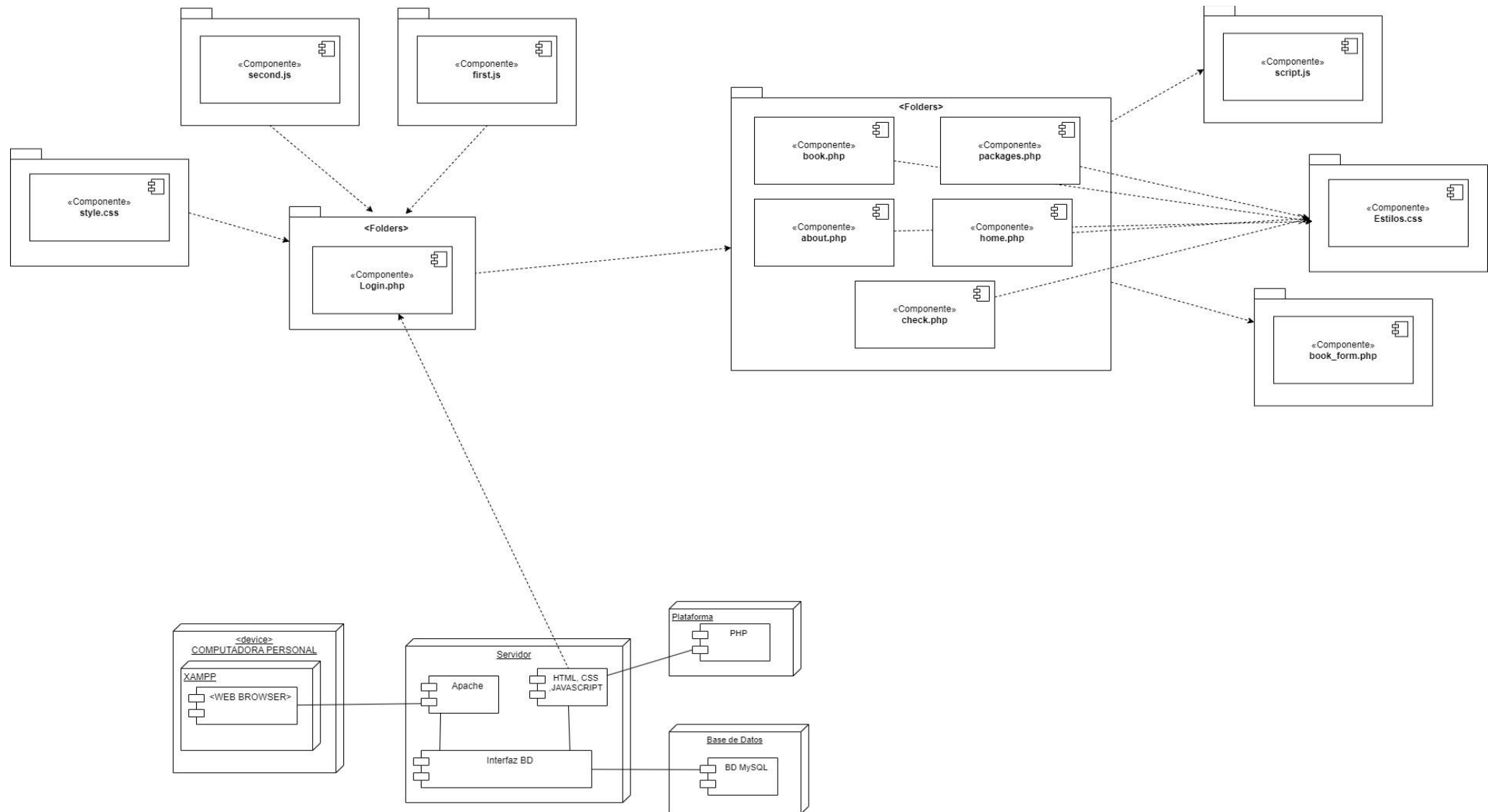
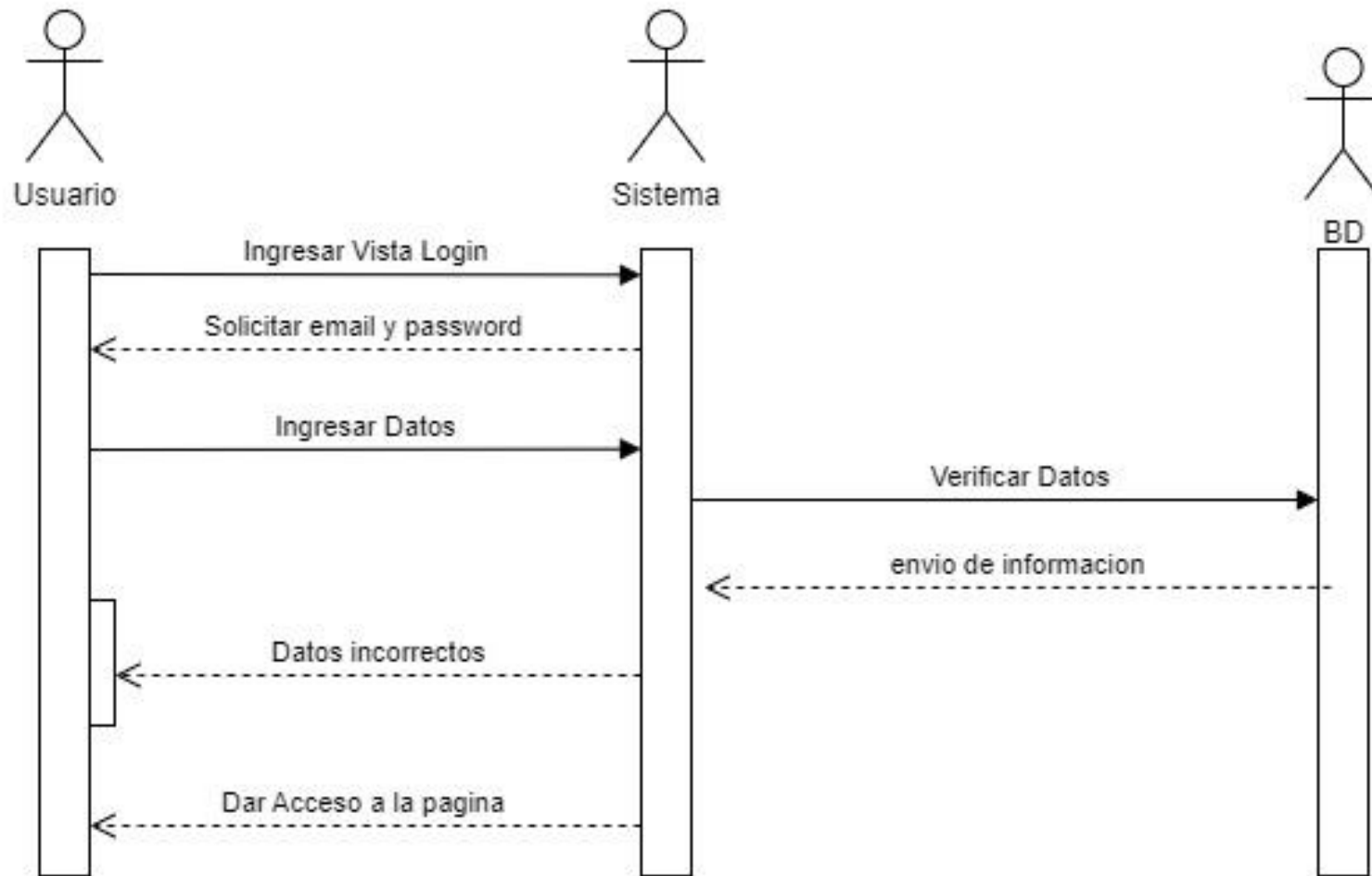
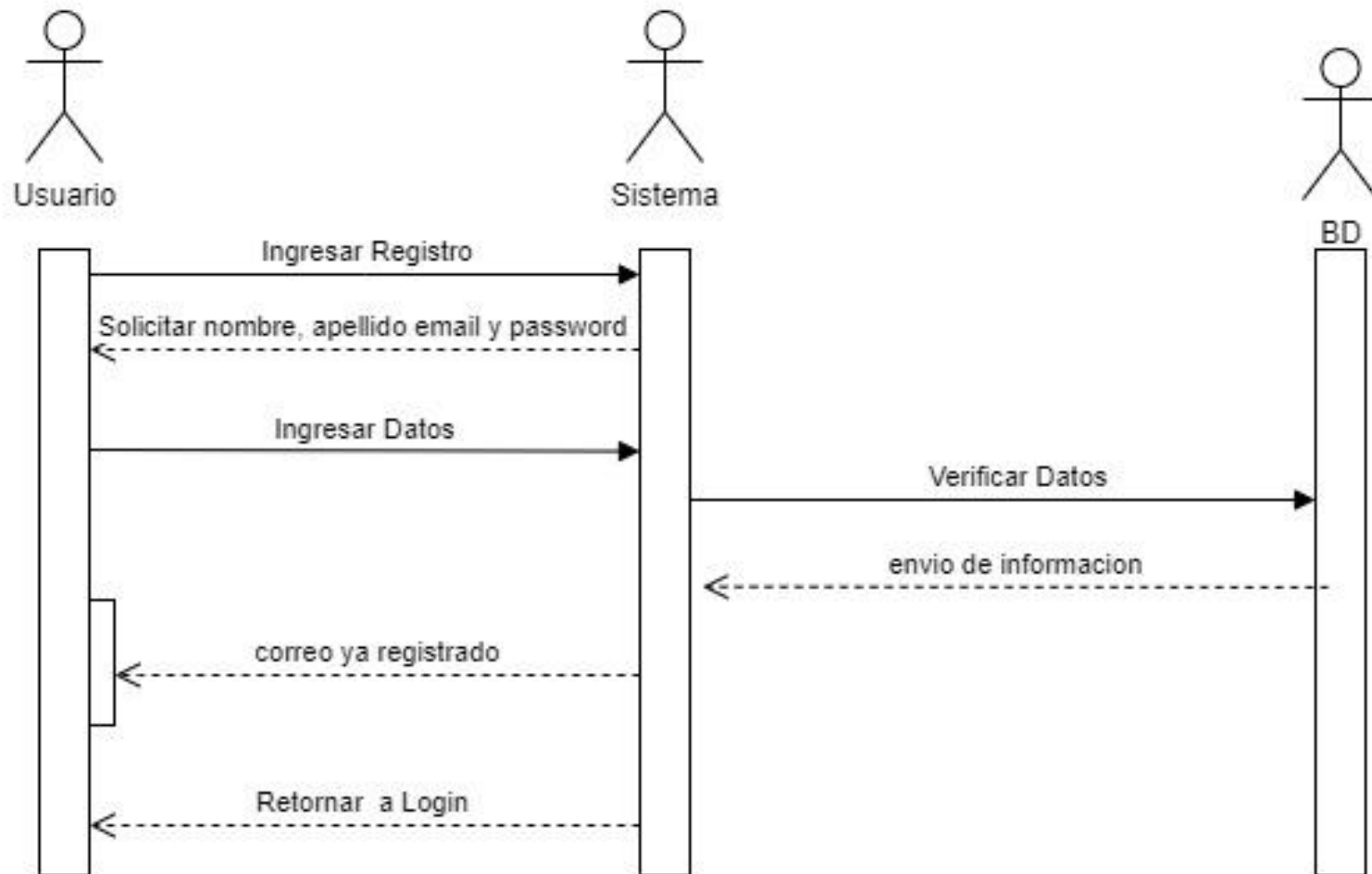


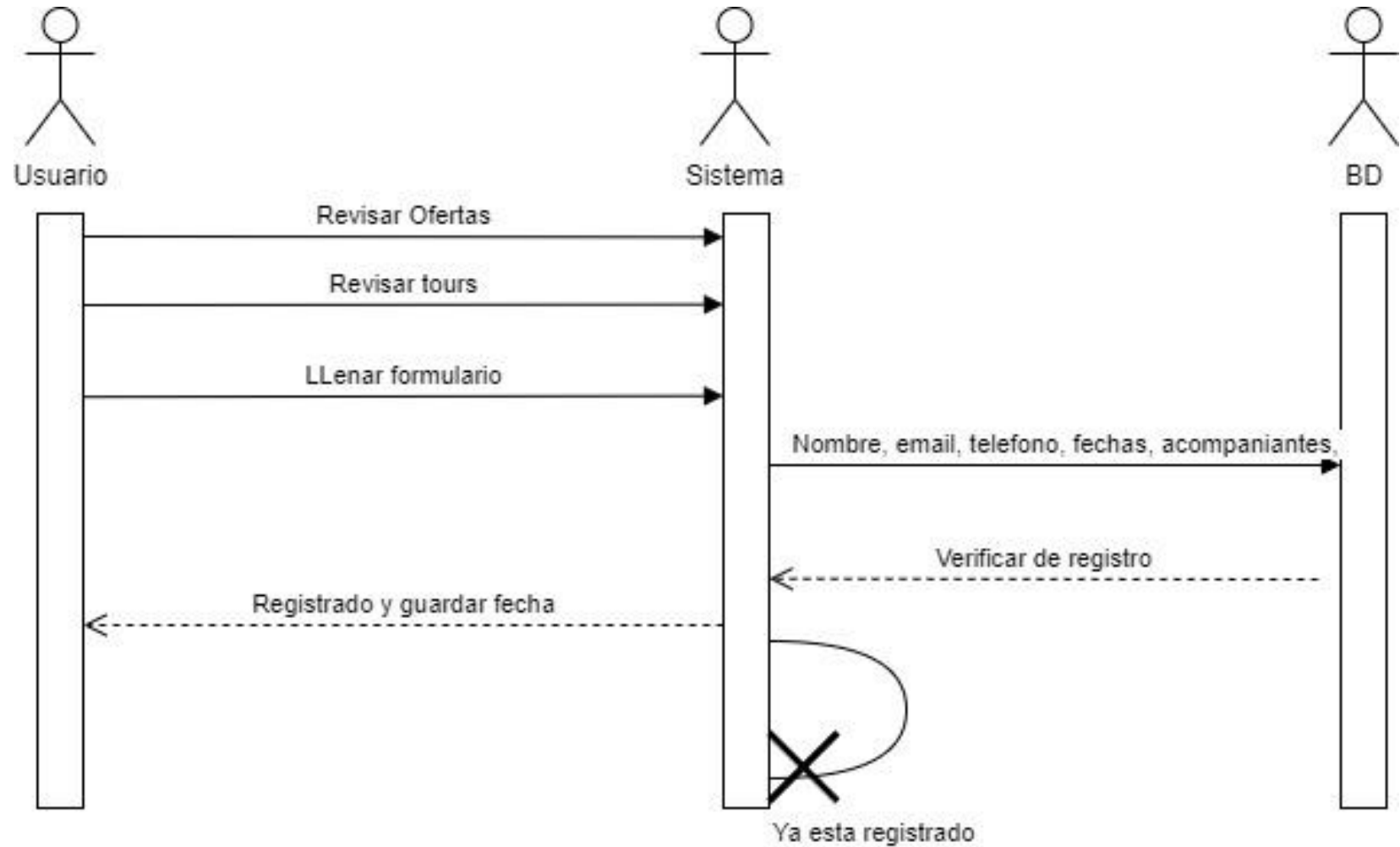
DIAGRAMA DE SECUENCIA (LOGIN)



(REGISTRO)



(SISTEMAS PRINCIPAL)



CÓDIGO

```
<div class="login-container" id="login">
  <div class="top">
    <span>No tienes una cuenta? <a href="#" onclick="register()">Registrarse</a></span>
    <header>Login</header>
  </div>
  <form action="book_form.php" method="POST">
    <div class="input-box">
      <input type="text" class="input-field" placeholder="Email" name="email" required="true">
      <i class="bx bx-user"></i>
    </div>
    <div class="input-box">
      <input type="password" class="input-field" placeholder="Contraseña" name="contrasenia" required="true">
      <i class="bx bx-lock-alt"></i>
    </div>
    <div class="input-box">
      <input type="submit" class="submit" value="Iniciar Sesion" name="sesion" >
    </div>
    <div class="two-col">
      <div class="one">
        <input type="checkbox" id="login-check" >
        <label for="login-check"> Recordar</label>
      </div>
      <div class="two">
        <label><a href="#">Olvidaste contraseña?</a></label>
      </div>
    </div>
  </form>
</div>
```

En esta parte se crea un ‘form’ para mostrar en la vista que es lo que vamos a pedir en este caso pedimos email y contraseña ya que es el principio o el login y nos damos cuenta como que sean requeridos ambos campos para que no sea sencillo entra solo presionando el botón.

```

<div class="register-container" id="register">
  <div class="top">
    <span>Tienes una cuenta? <a href="#" onclick="login()">Login</a></span>
    <header>Registrarse</header>
  </div>
  <form action="book_form.php" method="post">
    <div class="two-forms">
      <div class="input-box">
        <input type="text" class="input-field" placeholder="Nombre" name="nombre" required="true">
        <i class="bx bx-user"></i>
      </div>
      <div class="input-box">
        <input type="text" class="input-field" placeholder="Apellido" name="apellido" required="true">
        <i class="bx bx-user"></i>
      </div>
      <div class="input-box">
        <input type="email" class="input-field" placeholder="Email" name="email" required="true">
        <i class="bx bx-envelope"></i>
      </div>
      <div class="input-box">
        <input type="password" class="input-field" placeholder="Contraseña" name="contrasenia" required="true">
        <i class="bx bx-lock-alt"></i>
      </div>
      <div class="input-box">
        <input type="submit" class="submit" value="Registrar" name="regis">
      </div>
      <div class="two-col">
        <div class="one">
          <input type="checkbox" id="register-check">
          <label for="register-check"> Recordar</label>
        </div>
      </div>
    </form>
  </div>

```

Ahora bien, en este caso estamos en la parte de registro. En esta parte pedimos lo que es el nombre apellido, email y contraseña y al igual que el login son espacios requeridos que no podemos dejar en blanco. Si nos damos cuenta en el tipo de “input” están ‘text’ para los nombres y apellidos, email para el mismo y password para la contraseña.

```

<script src="/js/first.js"></script>
<script src="/js/second.js"></script>

```

Esta es la última parte de este archivo que hace un llamado a javascript en otros archivos.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>home</title>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swiper@10/swiper-bundle.min.css"/>

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css"/>

  <link rel="stylesheet" href="/css/estilos.css"/>
</head>
<body>
  <section class="header">
    <a href="home.php" class="logo">HelloWorld</a>

    <nav class="navega">
      <a href="home.php">Principio</a>
      <a href="about.php">Acerca de</a>
      <a href="package.php">Paquete</a>
      <a href="book.php">Registro</a>
      <a href="/login.php">Salir</a>
    </nav>

    <div id="menu-btn" class="fas fa-bars"></div>
  </section>

  <section class="home">
    <div class="home-slider">
      <div class="w">
        <div class="slide" style="background:url(imagen/rio.jpg)">
          <div class="content">
            <span>Explorar, Descubrir, Viajar</span>
            <h3>Viaja alrededor del mundo</h3>
            <a href="package.php" class="btn">Descubrir mas</a>
          </div>
        </div>
      </div>
    </div>
  </section>

```

En esta parte estamos en el archivo home si nos damos cuenta en la parte de section class header eso se repetirá en todos los los archivos que veremos ya que es el encabezado

```

<section class="pie">
  <div class="principal-cotenedor">
    <div class="caja">
      <h3>Links</h3>
      <a href="home.php"><i class="fas fa-angle-right"></i>Principio</a>
      <a href="about.php"><i class="fas fa-angle-right"></i>Acerca de</a>
      <a href="package.php"><i class="fas fa-angle-right"></i>Paquete</a>
      <a href="book.php"><i class="fas fa-angle-right"></i>book</a>
    </div>

    <div class="caja">
      <h3>Extra Links</h3>
      <a href="#"><i class="fas fa-angle-right"></i>Preguntas?</a>
      <a href="#"><i class="fas fa-angle-right"></i>Acerca de nosotros</a>
      <a href="#"><i class="fas fa-angle-right"></i>Politicass</a>
      <a href="#"><i class="fas fa-angle-right"></i>Terminos</a>
    </div>

    <div class="caja">
      <h3>Informacion</h3>
      <a href="#"><i class="fas fa-phone"></i>+50245625889</a>
      <a href="#"><i class="fas fa-phone"></i>+50241342459</a>
      <a href="#"><i class="fas fa-envelope"></i>moralesalan235@gmail.com</a>
      <a href="#"><i class="fas fa-map"></i>Guatemala, Guatemala</a>
    </div>

    <div class="caja">
      <h3>Redes Sociales</h3>
      <a href="#"><i class="fab fa-facebook-f"></i> facebok </a>
      <a href="#"><i class="fab fa-twitter"></i> twitter </a>
      <a href="#"><i class="fab fa-instagram"></i> instagram </a>
      <a href="#"><i class="fab fa-linkedin"></i> linkedin </a>
    </div>
  </div>

  <div class="creditos">Created by <snap>AGMT</snap> | Derechos Reservados</div>
</section>

```

Al igual que en esta parte se seguirá repitiendo en los demás ya que en el pie de página del sistema es repetitivo en este sistema.

```

<section class="booking">
  <h1 class="heading-title">Registra tu viaje</h1>
  <form action="book_form.php" method="post" class="book-form">
    <div class="flex">
      <div class="inputBox">
        <span>Nombre:</span>
        <input type="text" placeholder="Ingresa tu nombre" name="name" required="true">
      </div>
      <div class="inputBox">
        <span>Correo Electronico:</span>
        <input type="email" placeholder="Ingresa tu correo" name="email" required="true">
      </div>
      <div class="inputBox">
        <span>Telefono:</span>
        <input type="number" placeholder="Ingresa tu numero" name="phone" required="true">
      </div>
      <div class="inputBox">
        <span>Direccion:</span>
        <input type="text" placeholder="Ingresa tu direccion" name="address" required="true">
      </div>
      <div class="inputBox">
        <span>Hacia donde:</span>
        <input type="text" placeholder="Lugar a visitar" name="location" required="true">
      </div>
      <div class="inputBox">
        <span>Cuanto:</span>
        <input type="number" placeholder="Acompañantes" name="guests" required="true">
      </div>
      <div class="inputBox">
        <span>Llegada:</span>
        <input type="date" name="arrivals" required="true">
      </div>
      <div class="inputBox">
        <span>Salida:</span>
        <input type="date" name="leaving" required="true">
      </div>
    </div>
  </form>
</section>

```

El archivo book o registro este nos sirve para visualizar nuestro viaje y registrarlo en el sistema y como nos damos cuenta todos los datos son requeridos al llenarlos y son nombre, email, teléfono, dirección, localidad o lugar de visita, acompañantes, llegada y salida

```

$connection = mysqli_connect('localhost','root','','book_db');

if(isset($_POST['send'])){
    $name = $_POST['name'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $address = $_POST['address'];
    $location = $_POST['location'];
    $guests = $_POST['guests'];
    $arrivals = $_POST['arrivals'];
    $leaving = $_POST['leaving'];

    $request = "insert into book_form(name, email, phone, address, location, guests, arrivals, leaving) values
    ('$name','$email','$phone','$address','$location','$guests','$arrivals','$leaving')";

    $verificar_correo = $verificar_correo = mysqli_query($connection, "SELECT * FROM book_form WHERE email = '$email'");

    if(mysqli_num_rows($verificar_correo) > 0){
        echo '
        <script>
        alert("Este correo ya esta registrado");
        window.location = "/book.php";
        </script>
        ';
    } else {
        $ejecutar = mysqli_query($connection, $request);
        if($ejecutar){
            echo'
            <script>
            alert("Usuario Registrado");
            window.location = "/home.php";
            </script>
            ';
        }
    }
}

```

El archivo book_form es el archivo que envía o que hace la conexión con el la base de datos ya creada anteriormente en el sistema de xampp ahora bien si nos damos cuenta creamos variables en el entorno de PHP una que conecta y las van dentro del condicional if y por supuesto que cuando se enviar para aca van por el método POST para mayor seguridad ahora bien dentro el primer condiciona hay otro condicional que hace una verificación de correo esto va conectado al registro de book y verifica si el correo ya existe dentro de la base de datos y cuando lo encuentra manda una alerta diciendolo que ya está registrado en cambio si es diferente de lo que ya está entonces envía otro mensaje de alerta que el usuario ya está registrado y lo regresa para ver otras ofertas.


```

    }else if(isset($_POST['regis'])){
        $nombre = $_POST['nombre'];
        $apellido = $_POST['apellido'];
        $email = $_POST['email'];
        $contrasenia = $_POST['contrasenia'];

        $request = "insert into usuarios (nombre, apellido, email, contrasenia) values ('" . $nombre . "', '" . $apellido . "', '" . $email . "', '" . $contrasenia . "')";

        $verificar_correo = mysqli_query($connection, "SELECT * FROM usuarios WHERE email = '$email'");

        if(mysqli_num_rows($verificar_correo) > 0){
            echo '
            <script>
            alert("Este correo ya esta registrado");
            window.location = "/login.php";
            </script>
            ';
        } else {
            $ejecutar = mysqli_query($connection, $request);

            if($ejecutar){
                echo '
                <script>
                alert("Usuario Registrado");
                window.location = "/login.php";
                </script>
                ';
            }
        }
    }
}

```

El siguiente condicional nos sirve para registrar a alguien en sistema para poder entrar al principal al igual que el anterior este verifica si existe el correo en el sistema para que no haya una confusión cuando quiera iniciar sesión en el sistema entonces al hacer la verificación alerta si ya existe este correo o es un nuevo usuario.

```

}else if(isset($_POST['sesion'])){
    $email = $_POST['email'];
    $contrasenia = $_POST['contrasenia'];

    $verificar_correo = mysqli_query($connection, "SELECT * FROM usuarios WHERE email = '$email'");
    $verificar_contra = mysqli_query($connection, "SELECT * FROM usuarios WHERE contrasenia = '$contrasenia'");

    if(mysqli_num_rows($verificar_correo) > 0 && mysqli_num_rows($verificar_contra) > 0){
        echo '
        <script>
        alert("BIENVENIDO");
        window.location = "/home.php";
        </script>
        ';
    } else {
        echo '
        <script>
        alert("No existe este usuario");
        window.location = "/login.php";
        </script>
        ';
    }
} else{
    echo 'Algo esta mal intentalo de nuevo';
}

```

Para el último condicional es para verificar la entrada en sesión del login y en este caso hace dos verificaciones, la primera es el email como el anterior para ver si existe y dos es para verificar la contraseña en el sistema y de igual forma tiene dos condicionales más dentro del sistema para si es igual que en este caso sea bienvenido y si no lo envía a registro para que lo haga.

SCRIPTS

```

function myMenuFunction() {
    var i = document.getElementById("navMenu");
    if(i.className === "nav-menu") {
        i.className += " responsive";
    } else {
        i.className = "nav-menu";
    }
}

```

```
let menu = document.querySelector('#menu-btn');
let navega = document.querySelector('.header .navega');

menu.onclick = () =>{
    menu.classList.toggle('fa-times');
    navega.classList.toggle('active');
}
```

```
var a = document.getElementById("loginBtn");
var b = document.getElementById("registerBtn");
var x = document.getElementById("login");
var y = document.getElementById("register");
function login() {
    x.style.left = "4px";
    y.style.right = "-520px";
    a.className += " white-btn";
    b.className = "btn";
    x.style.opacity = 1;
    y.style.opacity = 0;
}
function register() {
    x.style.left = "-510px";
    y.style.right = "5px";
    a.className = "btn";
    b.className += " white-btn";
    x.style.opacity = 0;
    y.style.opacity = 1;
}
```

CSS

existe el primer css que se hace llamar estilos el cual hace una buen estilo al sistema principal y los demas archivos dentro del principal

```

--main-color: #8e44ad;
--black: #222;
--white: #fff;
--light-black: #777;
--light-white: #fff9;
--dark-bg: rgba(0,0,0,7);
--light-bg: #eee;
--border:.1rem solid var(--black);
--box-shadow:0 .5rem 1rem rgba(0,0,0,.1);
--text-shadow:0 .5rem 1rem rgba(0,0,0,.1);

(
font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
margin:0 ;
padding: 0;
box-sizing: border-box;
outline: none;
text-decoration: none;
text-transform: capitalize;

tml{
font-size: 62.5%;
overflow-x: hidden;

tml::-webkit-scrollbar{
width: 1rem;

tml::-webkit-scrollbar-track{
background-color: var(--white);

tml::-webkit-scrollbar-thumb{
background-color: var(--white);

ection{
padding: 5rem 10%;

```

Para el segundo le pone estilo a login principal y le da las texturas

```

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap');

*{
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: 'Poppins', sans-serif;
}
body{
background: url("/imagen/anti.jpg");
background-size: cover;
background-repeat: no-repeat;
background-attachment: fixed;
overflow: hidden;
}
.wrapper{
display: flex;
justify-content: center;
align-items: center;
min-height: 110vh;
background: rgba(39, 39, 39, 0.4);
}
.nav{
position: fixed;
top: 0;
display: flex;
justify-content: space-around;
width: 100%;
height: 100px;
line-height: 100px;
background: linear-gradient(rgba(39,39,39, 0.6), transparent);
z-index: 100;
}
.nav-logo p{
color: white;
font-size: 25px;
font-weight: 600;

```