

---

# MEMORIA SEGUNDA Y TERCERA SECCIÓN

---

# PRIMER APARTADO

## Introducción

En esta práctica se ha trabajado en la integración de funcionalidades de servidor utilizando PHP en una página web diseñada previamente con HTML y Bootstrap. El objetivo ha sido añadir dinamismo y funcionalidades interactivas, como la capacidad de realizar y visualizar reservas de hoteles, mediante la creación de formularios y el manejo de datos de usuario.

## Estructura y Diseño de la Página

### PHP

Se han desarrollado varias páginas y clases PHP:

- cliente.php y servidor.php: Permiten a los usuarios realizar reservas y visualizar los resultados de estas, respectivamente.
- LogInCliente.php y LogInServidor.php: Gestionan la autenticación de los usuarios y el servidor.

### Formulario PHP

Manejo de formularios: Los formularios han sido implementados para recoger y validar datos del usuario, utilizando métodos POST y GET para el envío y recepción de estos datos. Esto incluye:

- Validación de entradas mediante expresiones regulares (REGEXP) para prevenir inyecciones SQL y otras vulnerabilidades.
- Diseño responsivo y fácil de usar, asegurando una experiencia de usuario coherente y accesible.

### LogInCliente y LogInServidor

Estos archivos se encargan de manejar el proceso de autenticación para dos tipos de usuarios: clientes y servidores (personal administrativo o sistema de gestión del hotel). La autenticación es un paso crucial en cualquier aplicación web, ya que asegura que solo los usuarios autorizados puedan acceder a ciertas funcionalidades y datos.

#### LogInCliente

- Propósito: Sirve como una prueba para capturar y manejar datos de inicio de sesión enviados por los clientes utilizando un formulario HTML. Aunque no conlleva un cambio visual en la interfaz, el proceso subyacente es crucial para entender cómo se transmiten y reciben los datos en aplicaciones web.
- Implementación:
  - El formulario utiliza el método POST para enviar los datos del usuario (como el nombre de usuario y la contraseña) de manera segura al servidor.
  - PHP recibe estos datos y puede simular una verificación contra credenciales predefinidas o almacenadas en alguna estructura de datos.

- Se utiliza este proceso para ilustrar cómo se manejarían las sesiones y la autenticación en un escenario real, aunque en este caso, la interfaz del usuario permanece inalterada tras el envío del formulario.

## LogInServidor

- Propósito: Demuestra la captura y manejo de datos de inicio de sesión para el personal administrativo o servidor, utilizando el mismo enfoque práctico que LogInCliente.php. El foco está en el proceso backend y no implica cambios en la interfaz de usuario tras el inicio de sesión.
- Implementación:
  - Similar a LogInCliente.php, este archivo maneja los datos enviados mediante el método POST, simulando el proceso de autenticación del servidor.
  - Es un ejemplo práctico de cómo se recibirían y procesarían las credenciales del personal administrativo, enfatizando la importancia de métodos seguros de transmisión de datos como POST.

## Reservar hoteles

### cliente

- Propósito: cliente.php actúa como la interfaz principal para que los usuarios realicen reservas. A través de esta página, un cliente puede seleccionar un hotel y enviar la información necesaria para hacer una reserva, utilizando un formulario.
- Implementación
  - Formulario de Reserva: La página incluye un formulario donde los usuarios pueden seleccionar opciones como el hotel de interés, fechas de estancia, tipo de habitación, y cualquier otro detalle relevante para la reserva.
  - Método de Envío: El formulario utiliza el método POST para enviar los datos de manera segura al servidor, donde serán procesados.
  - Simulación de la Reserva: Al enviar el formulario, los datos son recogidos por servidor.php. Aunque esta implementación puede no conectar con una base de datos real para completar la reserva, simula el proceso de recibir y gestionar los datos de reserva, como lo haría un sistema de gestión de reservas.

### servidor

- Propósito: servidor.php simula el lado del servidor de la gestión de reservas. Recibe los datos enviados desde cliente.php y simula el proceso de registrar la reserva, proporcionando un ejemplo práctico de cómo se manejarían estas operaciones en una aplicación real.
- Implementación
  - Recepción de Datos: Al recibir los datos del formulario a través del método POST, servidor.php procesa la información de la reserva. Aunque en este caso no se realiza una operación real con una base de datos, se podría mostrar un mensaje de confirmación o resumen de la reserva simulada para completar la experiencia del usuario.

- Validación y Procesamiento: Se pueden implementar validaciones para asegurar que la información recibida es correcta y completa. Esta etapa también podría incluir la simulación de disponibilidad de habitaciones y la aplicación de cualquier regla de negocio necesaria.

## Conclusiones

La integración de funcionalidades de servidor mediante PHP ha enriquecido significativamente la página web, permitiendo una interacción más dinámica y personalizada. Esta práctica ha demostrado la importancia de la validación de datos y la seguridad en el desarrollo web. Los retos encontrados durante el desarrollo, como la gestión de sesiones de usuario y la validación de formularios, han sido superados mediante el uso efectivo de PHP. Como posibles mejoras, se podrían considerar la implementación de una API REST para manejar las reservas y la autenticación de manera más eficiente, así como la incorporación de AJAX para actualizar los contenidos de la página sin necesidad de recargarla.

# SEGUNDO APARTADO

## Introducción

En esta sección de la práctica, nos centramos en la integración y gestión de bases de datos dentro de una aplicación web, utilizando PhpMyAdmin para manejar las bases de datos imdb\_small, simpsons, y world. La práctica se orienta hacia la ejecución de consultas SQL desde el sitio web para interactuar con estas bases de datos, específicamente para insertar nuevos clientes y leer listas de ciudades, además de validar la existencia de ciudades y correos electrónicos dentro de las bases de datos mencionadas.

## Gestión de Base de Datos y Implementación Web

### PhpMyAdmin y SQL

- Configuración Inicial: Asumimos que las bases de datos requeridas ya están importadas y disponibles en el entorno de corrección. Nuestro sitio web está diseñado para verificar y crear, si es necesario, las tablas adicionales para almacenar información específica, como datos de nuevos clientes.
- Consultas SQL: Se implementan consultas SQL para la inserción de datos (como nuevos clientes) y la lectura de datos (listado de ciudades) directamente desde el sitio web, facilitando la interacción dinámica con la base de datos.

### Implementación Específica

- buscarhoteles.php: Este archivo se encarga de verificar la existencia de una ciudad en la base de datos world cuando un usuario realiza una búsqueda de hoteles. Si la

ciudad existe, se redirige al usuario a una página de reserva de hoteles para esa ciudad; si no, se informa al usuario que la ciudad no existe.

- Modificación del Login: Se adapta el proceso de inicio de sesión para verificar que el correo electrónico ingresado por el usuario exista en la base de datos simpsons, sin verificar la contraseña. Esto sirve como una práctica inicial para familiarizarse con la validación de datos de usuario contra una base de datos.

## Conclusiones

Esta sección de la práctica ha sido una valiosa oportunidad para familiarizarse con la gestión de bases de datos dentro de un entorno web utilizando PhpMyAdmin y PHP. Las implementaciones realizadas demuestran cómo las consultas SQL pueden integrarse en un sitio web para ofrecer funcionalidades dinámicas y cómo se pueden manejar los datos de manera segura y eficiente. A través de ejercicios prácticos como la validación de la existencia de ciudades y correos electrónicos en las bases de datos, hemos aprendido la importancia de las interacciones entre la aplicación web y la base de datos para proporcionar una experiencia de usuario coherente y fiable. Estos conocimientos sientan las bases para desarrollos futuros más complejos, donde la integración y gestión de datos juegan un papel central en la funcionalidad y el éxito de las aplicaciones web.