



NEXO GLOBAL SERVICES

NGS

☎ 661669381

🌐 ngs.com

~DOCUMENTACIÓN DEL PROYECTO:

ÍNDICE:

Introducción:

- > Propósito del proyecto. 3
- > Objetivos del proyecto. 3

Descripción General:

- > Descripción de la página web de la agencia de viajes. 4
- > Funcionalidades principales. 4
- > Descripción del programa de gestión de la base de datos. 4
- > Relación entre la página web y el programa de gestión. 4

Arquitectura del Sistema:

- > Diagrama de la arquitectura. 5
- > Tecnologías utilizadas. 6

Base de Datos:

- > Estructura de la base de datos. 6
- > Tipos de datos almacenados. 7
- > Relación con el programa de gestión. 7

Seguridad:

- > Medidas de seguridad implementadas. 8

Desarrollo y Tecnologías Utilizadas:

- > Lenguajes de programación. 10
- > Frameworks y librerías. 10
- > Herramientas de desarrollo. 10

Instalación y Configuración:

- > Pasos para configurar el sistema. 11

Mantenimiento y Actualizaciones:

- > Procedimientos de mantenimiento. 11

Dificultades encontradas y decisiones al respecto.

- > Dificultades y decisiones.

12-16

~PROPÓSITO DEL PROYECTO:

El propósito de este proyecto es desarrollar una página web tipo blog para una agencia de viajes, que servirá como plataforma para compartir artículos y noticias relacionadas con destinos turísticos, consejos de viaje, y experiencias de clientes. Además, la página web incluirá un formulario de contacto para que los visitantes puedan comunicarse fácilmente con la agencia y un sistema de inicio de sesión para acceder a un programa de gestión de la base de datos de la empresa, el cual permitirá a la empresa gestionar los datos de la base de datos y hacer inserción de datos nuevos, además de un sistema de generación de facturas en formato pdf.

~OBJETIVOS DEL PROYECTO:

- > Informar y atraer a potenciales clientes:

Crear una plataforma informativa y atractiva que capte la atención de viajeros interesados y potenciales clientes de la agencia.

- > Facilitar la comunicación:

Proveer un formulario de contacto sencillo y efectivo para que los visitantes puedan realizar consultas, solicitar información adicional o recibir asesoría personalizada.

- > Optimizar la gestión interna:

Implementar un sistema de inicio de sesión seguro que permita al personal autorizado acceder a un programa de gestión de la base de datos, mejorando la eficiencia en la administración de reservas, datos de clientes y otros aspectos operativos.

- > Ofrecer una experiencia de usuario intuitiva:

Diseñar una interfaz amigable y fácil de usar, tanto para los visitantes del blog como para los administradores del programa de gestión, garantizando una experiencia de navegación y uso óptima.

- > Asegurar la seguridad de los datos:

Implementar medidas de seguridad adecuadas para proteger la información personal de los usuarios y los datos sensibles gestionados a través del programa interno, así como implementar copias de seguridad periódicas.

~DESCRIPCIÓN GENERAL:

La página web de la agencia de viajes está diseñada para servir como una plataforma informativa y atractiva para los viajeros. La página principal actúa como un blog interactivo donde se publican artículos y noticias sobre destinos turísticos y experiencias de clientes. Además, la página incluye un formulario de contacto que permite a los visitantes comunicarse fácilmente con la agencia para realizar consultas o solicitar más información.

Funcionalidades principales:

Blog de Viajes: Publicación regular de artículos sobre diversos destinos turísticos, experiencias de viaje, y consejos prácticos para viajeros.

Formulario de Contacto: Sección dedicada donde los usuarios pueden enviar consultas o solicitudes de información, facilitando la comunicación directa con la agencia.

Inicio de Sesión: Área exclusiva para el personal de la agencia que permite acceder al programa de gestión de la base de datos mediante un sistema de autenticación seguro.

Categorías: Organización de los apartados por categorías y etiquetas para facilitar la búsqueda y navegación por el contenido de la web.

Sobre el programa de gestión de la base de datos:

El programa de gestión es una herramienta interna diseñada para ayudar al administrador de la agencia a administrar de manera eficiente los datos relacionados con los clientes, las reservas y otros aspectos operativos. Este programa está integrado con la base de datos de la empresa y es accesible únicamente a través de un inicio de sesión seguro con cifrado sha256. Entre sus funcionalidades principales se incluyen la actualización de información de clientes, la gestión de reservas y la generación de facturas detalladas.

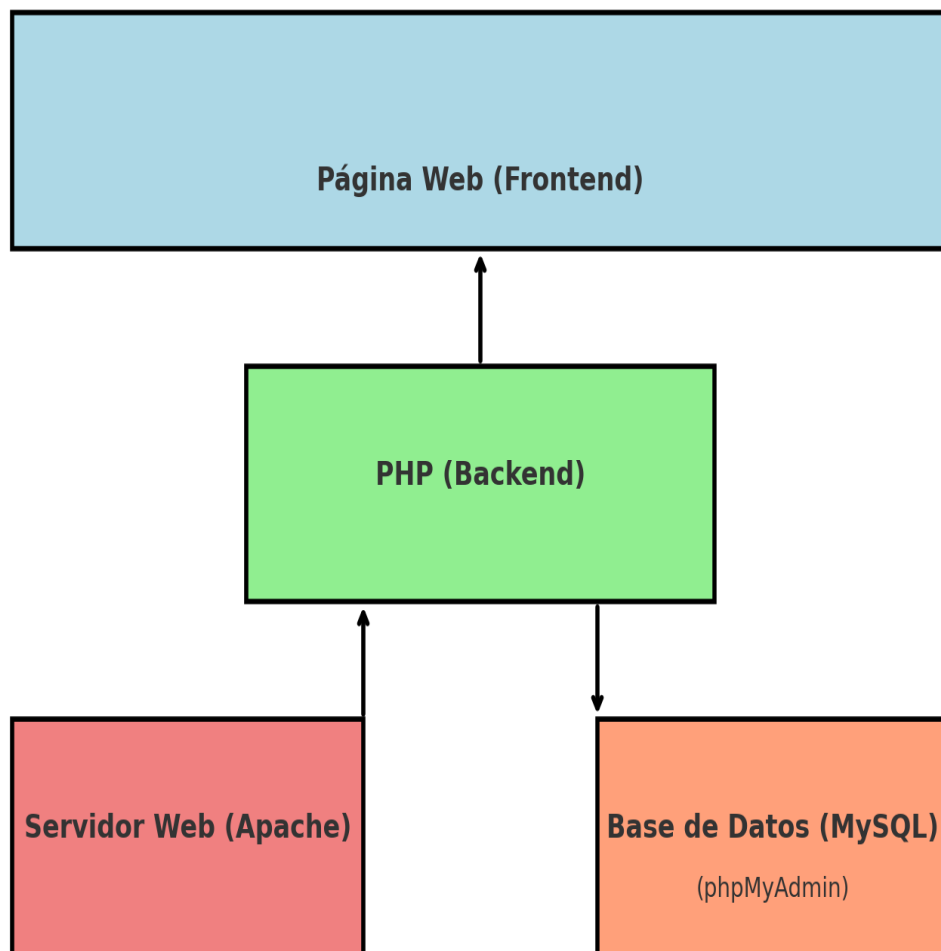
Relación entre la página web y el programa de gestión:

La página web y el programa de gestión están interconectados para asegurar una administración eficiente y centralizada de la información. Donde el programa de gestión permite al personal autorizado acceder y administrar dicha información. Esta integración garantiza que toda la información se mantenga actualizada y precisa, mejorando tanto la experiencia del usuario como la eficiencia operativa de la agencia.

~ARQUITECTURA DEL SISTEMA:

Diagrama de la Arquitectura:

Diagrama de la Arquitectura del Sistema



> Tecnologías Utilizadas:

XAMPP:

> Apache:

Servidor web que aloja la página web de la agencia de viajes, permitiendo que los usuarios accedan a través de un navegador o en local.

> MySQL (phpMyAdmin):

Sistema de gestión de bases de datos que almacena información sobre clientes, reservas y mas información.

phpMyAdmin se utiliza para administrar la base de datos de manera visual y eficiente.

> PHP:

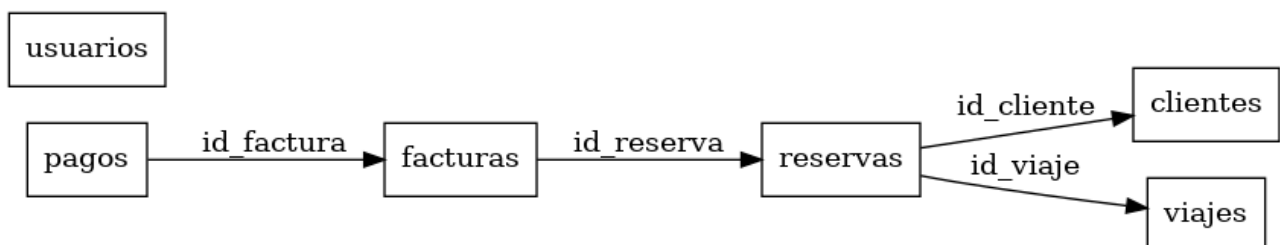
Lenguaje de programación del lado del servidor que facilita la conexión entre la página web y la base de datos. PHP se encarga de procesar las solicitudes del usuario, interactuar con la base de datos y generar el contenido dinámico de la web.

SQL Backup:

> Sistema para realizar backups periódicos de la base de datos, con confirmación vía correo electrónico.

~SOBRE LA BASE DE DATOS:

Estructura de la base de datos



- > clientes: almacena información de los clientes.
- > facturas: guarda los detalles de las facturas emitidas.
- > pagos: contiene la información de los pagos realizados.
- > reservas: almacena las reservas de los clientes.
- > usuarios: maneja los datos de los usuarios del sistema.
- > viajes: guarda la información de los viajes disponibles.

Tipos de datos almacenados

clientes: ID de cliente, nombre, apellido, correo electrónico, teléfono.

facturas: ID de factura, ID de reserva, fecha de emisión, total a pagar, estado de la factura, método de pago.

pagos: ID de pago, ID de factura, método de pago, monto, fecha de pago, estado del pago.

reservas: ID de reserva, ID de cliente, ID de viaje, fecha de reserva, estado de la reserva.

usuarios: nombre de usuario, contraseña.

viajes: ID de viaje, destino, fecha de salida, fecha de regreso, precio, descripción.

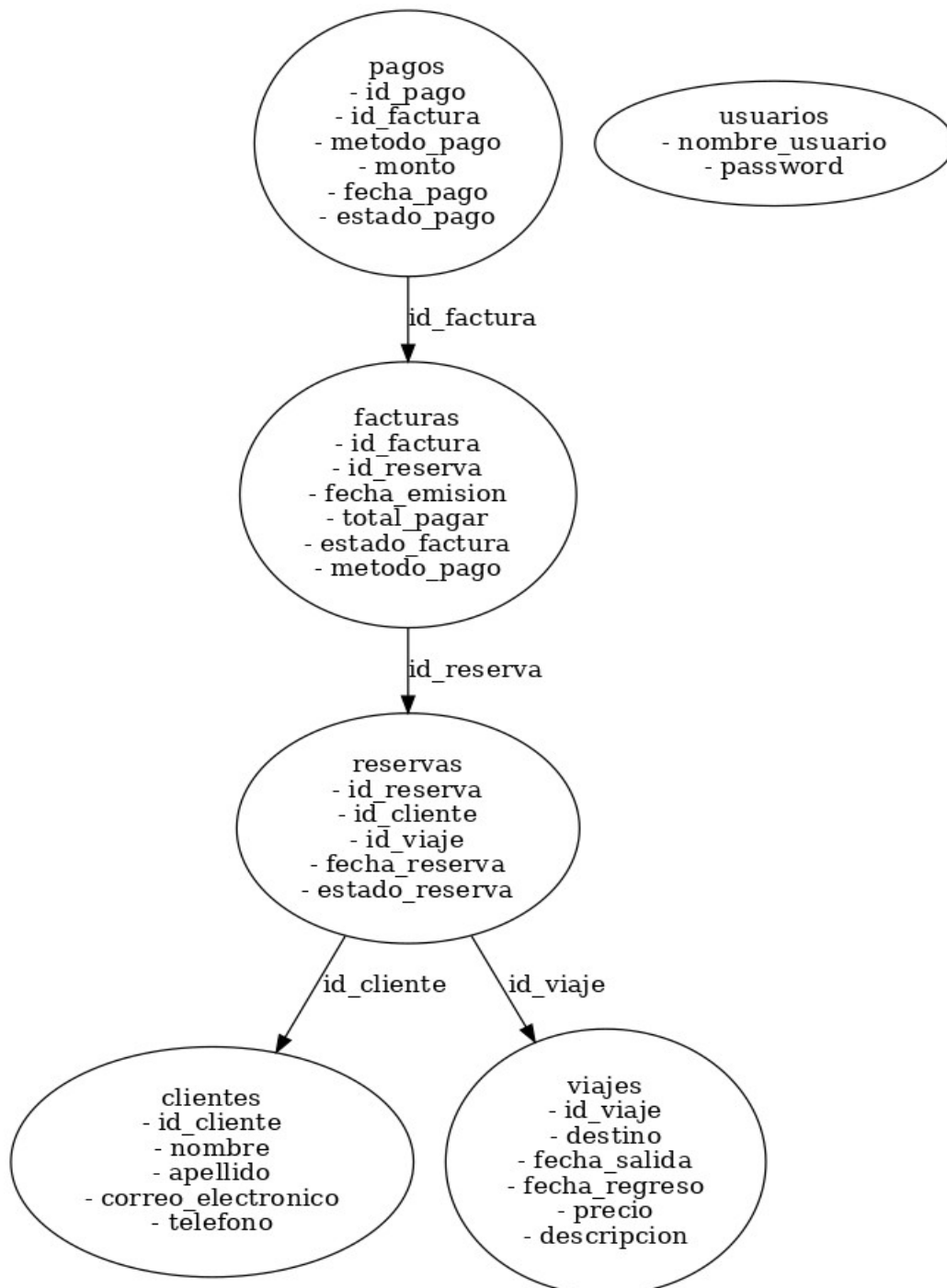
Relación con el programa de gestión

El programa de gestión se encarga de interactuar con la base de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre las distintas tablas. A través del inicio de sesión, los usuarios autorizados pueden acceder al sistema y gestionar información crítica como reservas, clientes, pagos y facturas.

Relación entre las tablas

- La tabla reservas está relacionada con las tablas clientes y viajes mediante las claves foráneas id_cliente e id_viaje.
- La tabla facturas está relacionada con la tabla reservas mediante la clave foránea id_reserva.
- La tabla pagos está relacionada con la tabla facturas mediante la clave foránea id_factura.

Vista general de la estructura:



~SEGURIDAD:

Medidas de seguridad implementadas:

> Autenticación y Autorización:

Implementación de un sistema de inicio de sesión seguro con nombre de usuario y contraseña encriptada.

Utilización de sesiones para mantener la autenticación del usuario durante la navegación.

> Encriptación de Datos:

Encriptación de las contraseñas de los usuarios utilizando algoritmos de hashing seguros SHA-256.

Implementación de SSL para cifrar la comunicación entre el cliente y el servidor.

> Validación y Sanitización de Datos:

Uso de validaciones en el lado del servidor para evitar la inyección de SQL y otros ataques basados en la entrada del usuario.

Sanitización de todas las entradas de usuario para prevenir la inyección de scripts.

> Políticas de Acceso:

Definición de roles y permisos para controlar el acceso a diferentes partes del sistema. Restricciones en la base de datos para asegurar que solo usuarios autorizados puedan realizar operaciones críticas.

> Respaldo y Recuperación:

Programación de respaldos regulares de la base de datos para prevenir la pérdida de datos.

Implementación de procedimientos de recuperación ante desastres para garantizar la seguridad de los datos.

- > **Políticas de privacidad y protección de datos:**
- > **Confidencialidad:**

Asegurar que solo las personas autorizadas tienen acceso a los datos personales de los clientes.

No compartir información personal con terceros sin el consentimiento explícito del cliente.

- > **Integridad:**

Garantizar que los datos almacenados son precisos y están protegidos contra alteraciones no autorizadas.

Monitoreo constante y registro de todas las transacciones y cambios en la base de datos.

- > **Transparencia:**

Informar a los clientes sobre las políticas de privacidad y cómo se manejan sus datos. Proveer opciones a los clientes para revisar, actualizar o eliminar su información personal.

~DESARROLLO Y TECNOLOGÍAS UTILIZADAS:

- > **Lenguajes de programación:**

HTML/CSS: Para la estructura y diseño de la página web.

JavaScript: Para la interactividad en el cliente.

PHP: Para el desarrollo del backend y la lógica del servidor.

SQL: Para la gestión y manipulación de la base de datos.

- > **Frameworks y librerías:**

FPDF: Utilizado para el desarrollo y creación de la factura generada en el programa de gestión.

Google Fonts: Fuentes que he usado para la página web.

Font Awesome: Emoticonos y algunas fuentes mas empleadas en la web.

> Herramientas de desarrollo:

XAMPP: Como entorno de desarrollo local que incluye Apache, MySQL y PHPMyAdmin.

Visual Studio Code: Para la edición y desarrollo de código.

phpMyAdmin: Para la administración de la base de datos MySQL.

Git: Para el control de versiones y la colaboración en el código.

~INSTALACIÓN Y CONFIGURACIÓN:

> Software:

- Sistema Operativo:** Windows 10 o una distribución de Linux.
- XAMPP:** Para instalar Apache, MySQL, PHP, y phpMyAdmin.
- Navegador web:** Google Chrome, Mozilla Firefox, o cualquier navegador compatible.

> Dependencias:

- PHP 7.4 o superior.
- MySQL 5.7 o superior (incluido en XAMPP).
- Librerías PHP necesarias

> Pasos para configurar el sistema:

Instalar XAMPP:

Descargar XAMPP desde apachefriends.org.

Ejecutar el instalador y seguir las instrucciones para completar la instalación.

Configurar Apache y MySQL:

Abrir el panel de control de XAMPP y arrancar los servicios de Apache y MySQL.

Configurar la base de datos:

Abrir phpMyAdmin desde el panel de control de XAMPP.

Crear una nueva base de datos llamada ngs.

Importar el archivo SQL proporcionado para crear las tablas y datos iniciales.

Configurar los archivos del proyecto:

Clonar o descargar el repositorio del proyecto en el directorio htdocs de XAMPP (generalmente ubicado en C:\xampp\htdocs en Windows).

~MANTENIMIENTO Y ACTUALIZACIONES:

Respaldo regular:

Programar respaldos automáticos de la base de datos usando herramientas como SQL backup (mi caso) o scripts PHP.

Realizar copias de seguridad de los archivos del proyecto periódicamente.

Monitoreo y logs:

Habilitar y revisar regularmente los logs de Apache y PHP para identificar y solucionar problemas.

Hacer respaldos periódicos de los cambios hechos en la web.

Optimización de la base de datos:

Ejecutar consultas de optimización y análisis periódicamente para mantener la base de datos eficiente.

Revisar y ajustar los índices de las tablas según sea necesario.

Seguridad:

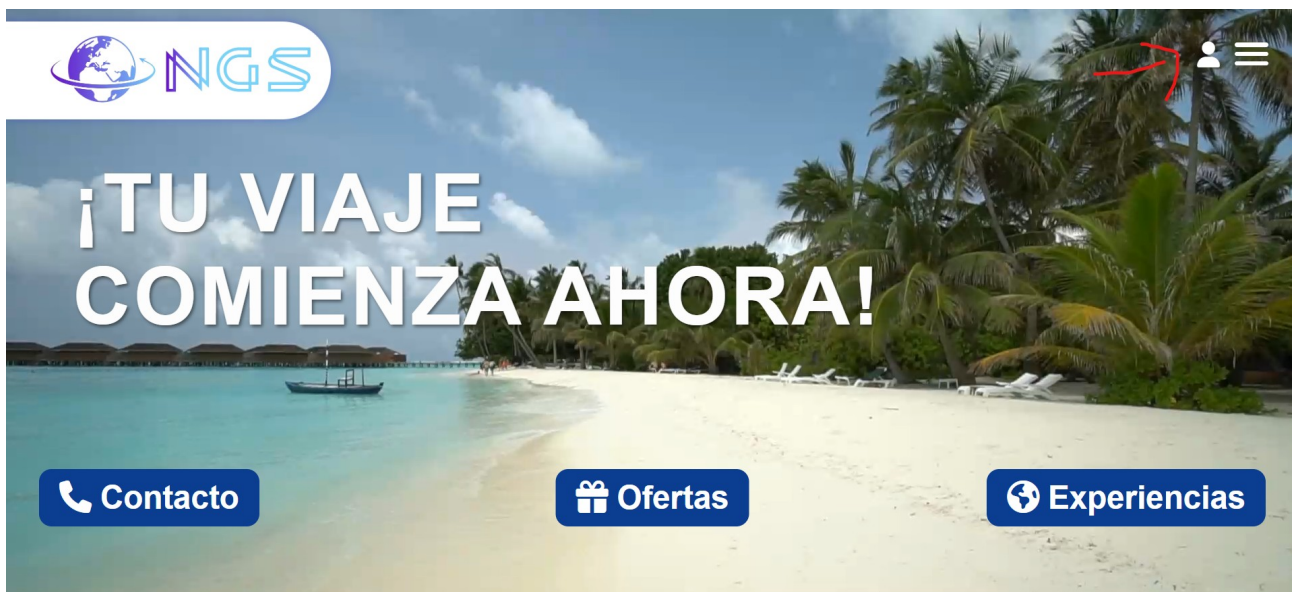
Mantener el servidor y el software actualizado con los últimos parches de seguridad.

Revisar y fortalecer las políticas de acceso y autenticación periódicamente.

~DIFICULTADES ENCONTRADAS Y DECISIONES AL RESPECTO:

> Escoger el diseño de la web

Este fue un punto bastante complicado pero finalmente me decidí por un estilo moderno de web, muy vistosa para el cliente y fácil de usar e interactiva.



> Creacion del inicio de sesión

Este punto me dio varios problemas a la hora de poner el cifrado en el código etc, ya que daba muchos errores y algunas veces el cifrado no se aplicaba o fallaba la seguridad anti inyección SQL, finalmente se solventó aplicándole un cifrado SHA256 y varios parámetros para evitar inyecciones SQL o scripts.

```
<?php
session_start(); // Iniciar sesión

// Conexión a la base de datos
$servername = "localhost";
$username = "NGS";
$password = "NGS2024..";
$dbname = "NGS";

// Crear conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}

// Función para limpiar y preparar los datos antes de ejecutar consultas SQL
2 references
function clean_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

// Verificar si se ha enviado el formulario de inicio de sesión
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Limpiar y obtener los datos del formulario
    $username = clean_input($_POST['username']);
    $password = clean_input($_POST['password']);

    // Consulta preparada para evitar inyección SQL
    $sql = "SELECT * FROM usuarios WHERE nombre_usuario=? LIMIT 1";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows == 1) {
        $user = $result->fetch_assoc();
        // Verificar la contraseña con hash SHA-256
        $hashed_password = hash('sha256', $password);
        if ($hashed_password == $user['password']) {
            // Inicio de sesión exitoso
            $_SESSION['logged_in'] = true;
            $_SESSION['username'] = $username;
            header("Location: ../!bddd/dashboard.php");
            exit();
        } else {
            // Contraseña incorrecta
            echo "Usuario y/o contraseña incorrectos.";
        }
    } else {
        // Usuario no encontrado
        echo "Usuario y/o contraseña incorrectos.";
    }

    // Cerrar la declaración y la conexión
    $stmt->close();
}

$conn->close();
?> The closing ?> tag should be omitted from files containing only PHP.
```

Aparte de esto, también en la base de datos se almacena la contraseña encriptada en SHA256

	nombre_usuario	password
   NGS	90fa4327acb33783d02265a03124c734d153126275d55f69f8...	

➤ Creacion de facturas

Este es un aspecto que se me dificultó bastante ya que no sabia como hacerlo, finalmente recurri a un repositorio de codigo abierto llamado fpdf, el cual configurando un par de archivos se encarga de poder generar pdf para facturas en mi caso de manera relativamente sencilla.

```
// Crear una nueva instancia de FPDF
$pdf = new FPDF('P', 'mm', 'A4');
$pdf->AddPage();

// Encabezado con logotipo
if (file_exists('../media/ngslogo.png')) {
    $pdf->Image('../media/ngslogo.png', 10, 10, 80);
}

// Espaciado debajo del logo
$pdf->Ln(40);

// Información de la empresa y del cliente en la misma línea
$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(120, 10, 'Datos de la Empresa', 0, 0, 'L');
$pdf->Cell(70, 10, 'Datos del Cliente', 0, 1, 'R');

$pdf->SetFont('Arial', '', 12);
$pdf->Cell(120, 10, 'Nexo Global Services', 0, 0, 'L');
$pdf->Cell(70, 10, $row['nombre'] . ' ' . $row['apellido'], 0, 1, 'R');

$pdf->Cell(120, 10, 'Plaza Espana, Cartagena', 0, 0, 'L');
$pdf->Cell(70, 10, 'Telefono: ' . $row['telefono'], 0, 1, 'R');

$pdf->Cell(120, 10, 'Telefono: 661669381', 0, 0, 'L');
$pdf->Cell(70, 10, 'Email: ' . $row['correo_electronico'], 0, 1, 'R');

$pdf->Cell(120, 10, 'Email: ngs@gmail.com', 0, 0, 'L');
$pdf->Ln(20);

// Título de la sección de la factura
$pdf->SetFont('Arial', 'B', 14);
$pdf->Cell(0, 10, 'Datos de la Factura', 0, 1, 'L');
$pdf->Ln(10);

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(40, 10, 'Factura ID: ', 1);
$pdf->SetFont('Arial', '', 12);
$pdf->Cell(60, 10, $row['id_factura'], 1, 1, 'C');

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(40, 10, 'ID Reserva: ', 1);
$pdf->SetFont('Arial', '', 12);
$pdf->Cell(60, 10, $row['id_reserva'], 1, 1, 'C');

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(40, 10, 'Fecha de Emision: ', 1);
$pdf->SetFont('Arial', '', 12);
$pdf->Cell(60, 10, $row['fecha_emision'], 1, 1, 'C');

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(40, 10, 'Total a Pagar: ', 1);
$pdf->SetFont('Arial', '', 12);
$pdf->Cell(60, 10, '$' . $row['total_pagar'], 1, 1, 'C');

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(40, 10, 'Metodo de Pago: ', 1);
$pdf->SetFont('Arial', '', 12);
$pdf->Cell(60, 10, $row['metodo_pago'], 1, 1, 'C');

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(40, 10, 'Estado de Factura: ', 1);
$pdf->SetFont('Arial', '', 12);
```

➤ Formulario de contacto

El formulario de contacto de la web puede que sea de los aspectos mas complicados que contiene, ya que es la única parte de la web que contiene php, y una de las partes mas difíciles fue la parte de estructurar los datos para la llegada a email y la protección contra bots.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Función para filtrar y escapar datos
    // references
    function filtrarDatos($dato) {
        return htmlspecialchars(trim($dato));
    }

    // Filtrar y escapar datos del formulario
    $nombre = filtrarDatos($_POST["nombre"]);
    $apellidos = filtrarDatos($_POST["apellidos"]);
    $email = filter_var(filtrarDatos($_POST["email"]), FILTER_SANITIZE_EMAIL);
    $ciudad = filtrarDatos($_POST["ciudad"]);
    $telefono = filtrarDatos($_POST["telefono"]);
    $mensaje = filtrarDatos($_POST["mensaje"]);

    // Validar campos requeridos
    if (empty($nombre) || empty($apellidos) || empty($email) || empty($ciudad) || empty($telefono) || empty($mensaje)) {
        die("Por favor, completa todos los campos del formulario.");
    }

    // Validar formato de correo electrónico
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        die("Por favor, introduce una dirección de correo electrónico válida.");
    }

    // Validar respuesta del captcha
    $respuesta = intval($_POST["respuesta"]); // Convertir respuesta a entero
    if ($respuesta != 9) {
        die("Por favor, responde la pregunta correctamente.");
    }

    // Configuración del correo
    $destinatario = "nexoglobalservices.contact@gmail.com"; // Reemplaza con tu dirección de correo destino
    $asunto = "Mensaje enviado desde el formulario de contacto de mi sitio web";

    // Construir el mensaje de correo
    $mensaje_email = "Nombre: $nombre\n";
    $mensaje_email .= "Apellidos: $apellidos\n";
    $mensaje_email .= "Email: $email\n";
    $mensaje_email .= "Ciudad: $ciudad\n";
    $mensaje_email .= "Teléfono: $telefono\n";
    $mensaje_email .= "Mensaje: \n$mensaje\n";

    $headers = "From: $nombre $apellidos <$email>\r\n";
    $headers .= "Reply-To: $email\r\n";

    // Enviar el correo
    if (mail($destinatario, $asunto, $mensaje_email, $headers)) {
        echo "Tu mensaje ha sido enviado correctamente, ya puedes cerrar esta ventana y te contactaremos con la mayor brevedad posible. ¡Gracias por contactar con nosotros!";
    } else {
        echo "Lo sentimos, ha ocurrido un error al enviar el mensaje.";
    }
}
```

> Carrusel de imágenes

Otro punto que se me complicó bastante fue crear el carrusel de imágenes en javascript.

```
//CARRUSEL

document.addEventListener('DOMContentLoaded', () => {
  const sliderGeneral = document.querySelector('.slider-general');
  const sliderSlide = document.querySelectorAll('.slider-slide');
  const sliderControlPrev = document.querySelector('.slider-control-prev');
  const sliderControlNext = document.querySelector('.slider-control-next');

  let index = 0;

  // Función para mover el slider a la izquierda
  function slideToLeft() {
    if (index === 0) {
      index = sliderSlide.length - 1;
    } else {
      index--;
    }
    sliderGeneral.style.transform = `translateX(-${index * (100/sliderSlide.length)}%)`;
  }

  // Función para mover el slider a la derecha
  function slideToRight() {
    if (index === sliderSlide.length - 1) {
      index = 0;
    } else {
      index++;
    }
    sliderGeneral.style.transform = `translateX(-${index * (100/sliderSlide.length)}%)`;
  }

  // Event listeners para los botones de control
  sliderControlPrev.addEventListener('click', slideToLeft);
  sliderControlNext.addEventListener('click', slideToRight);
});
```


~BIBLIOGRAFÍA:

- GitHub. (n.d.). **GitHub Documentation**. <https://docs.github.com/es>
- MySQL Documentation Team. (n.d.). **MySQL 8.0 Reference Manual**. Oracle. <https://dev.mysql.com/doc/refman/8.0/en/>
- OpenAI. (2024). **ChatGPT Documentation**. <https://www.openai.com/chatgpt>
- phpMyAdmin Team. (2021). **phpMyAdmin Documentation**. <https://docs.phpmyadmin.net/es/latest/>
- PHP Documentation Group. (n.d.). **PHP Manual**. <https://www.php.net/manual/es/>
- Stack Overflow. (n.d.). **Stack Overflow Community**. <https://stackoverflow.com/>
- Visual Studio Code Documentation Team. (n.d.). **Visual Studio Code Documentation**. Microsoft. <https://code.visualstudio.com/docs>
- W3Schools. (n.d.). **HTML, CSS, JavaScript, PHP, SQL Tutorial**. <https://www.w3schools.com/>
- Mozilla Developer Network (MDN). (n.d.). **Web Technologies Documentation**. <https://developer.mozilla.org/es/docs/Web>



Texto legal: <https://creativecommons.org/licenses/by/4.0/deed.es>

Links de ayuda:

<https://creativecommons.org/licenses/?lang=es> [EShhttps://creativecommons.org/choose/results-one?license_code=by&jurisdiction=&version=4.0&lang=es](https://creativecommons.org/choose/results-one?license_code=by&jurisdiction=&version=4.0&lang=es)