Servicio Nacional de Aprendizaje – SENA

Centro de Diseño y Metrologia

Informe php gestión odontologica

Aprendiz:

Edwin Camilo Lozano Chaparro

No Ficha 2848530-A

Introducción

El presente sistema tiene como objetivo gestionar de manera eficiente la información de pacientes, citas y médicos en un contexto odontológico. El código proporcionado corresponde a la interfaz web de un Sistema de Gestión Odontológica, diseñado para facilitar la administración de citas médicas, permitiendo a los usuarios asignar, consultar y cancelar citas de manera rápida y sencilla. En esta página específica, se presenta la estructura básica del sistema, que incluye un menú de navegación y una descripción general de las funcionalidades del sistema. Además, se proporciona acceso a las diferentes secciones, como la asignación de citas, la consulta y la cancelación de las mismas.

Contenido

Principal

Index.php

```php
<!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
    require_once 'Controlador/Controlador.php';
    require_once 'Modelo/GestorCita.php';
    require_once 'Modelo/Cita.php';
    require_once 'Modelo/Paciente.php';
    require_once 'Modelo/Conexion.php';
    $controlador = new Controlador();
    if (isset($_GET["accion"])) {
        if ($_GET["accion"] == "asignar") {
            $controlador->verPagina('Vista/html/asignar.php');
        } elseif ($_GET["accion"] == "consultar") {
            $controlador->verPagina('Vista/html/consultar.php');
        } elseif ($_GET["accion"] == "cancelar") {
            $controlador->verPagina('Vista/html/cancelar.php');
        } elseif ($_GET["accion"] == "guardarCita") {
            $controlador->agregarCita(
                $_POST["asignarDocumento"],

                $_POST["medico"],
                $_POST["fecha"],
                $_POST["hora"],
                $_POST["consultorio"]
            );
        } elseif ($_GET["accion"] == "consultarCita") {
            $controlador->consultarCitas($_POST["consultarDocumento"]);
        } elseif ($_GET["accion"] == "cancelarCita") {
            $controlador->cancelarCitas($_POST["cancelarDocumento"]);
        } elseif ($_GET["accion"] == "ConsultarPaciente") {
            $controlador->consultarPaciente($_GET["documento"]);
        } elseif ($_GET["accion"] == "ingresarPaciente") {
            $controlador->agregarPaciente(
                $_GET["PacDocumento"],
```

```
                $_GET["PacNombres"],
                $_GET["PacApellidos"],
                $_GET["PacNacimiento"],
                $_GET["PacSexo"]
            );
        }
    } else {
        $controlador->verPagina('Vista/html/inicio.php');
    }

    ?>
</body>

</html>
```

El código presentado corresponde a la lógica principal de una aplicación web desarrollada en PHP. Su propósito es servir como un controlador central que administra las solicitudes del usuario y las redirige hacia acciones específicas en el sistema. Este archivo se encarga de interpretar las acciones solicitadas a través de parámetros GET o POST, como la asignación, consulta, o cancelación de citas, así como la gestión de pacientes. Además, emplea una arquitectura basada en capas, incluyendo un controlador principal, modelos y vistas, lo cual facilita la separación de responsabilidades y el mantenimiento del código.

Explicación del código

1. **Estructura básica y dependencias**:
    - Se incluye la declaración de documentos requeridos como Controlador.php, GestorCita.php, Cita.php, Paciente.php, y Conexion.php. Estos archivos son fundamentales para la funcionalidad del sistema.
    - El controlador principal (Controlador) se instancia para manejar las acciones de usuario.
2. **Rutas y acciones**:
    - El parámetro $_GET["accion"] se utiliza para determinar la acción que el usuario desea realizar.
    - Dependiendo de la acción, se llama a métodos específicos del controlador para:
        - Mostrar páginas (asignar, consultar, cancelar citas).
        - Agregar nuevas citas o pacientes al sistema.
        - Consultar información de citas o pacientes.
        - Cancelar citas existentes.

- En caso de no especificar ninguna acción, se muestra la página de inicio predeterminada (inicio.php).

3. **Gestión de datos**:
   - Utiliza parámetros POST o GET para obtener datos relevantes como documento del paciente, nombre, médico asignado, fecha y hora de la cita, entre otros.
   - Implementa métodos como agregarCita, consultarCitas, cancelarCitas, entre otros, para realizar operaciones sobre los datos mediante las capas modelo y controlador.
4. **Separación de responsabilidades**:
   - El código demuestra una buena práctica al mantener la lógica de negocios en el controlador (Controlador.php) y delegar la interacción con la base de datos a los modelos (GestorCita, Paciente, etc.).
   - Las vistas se encuentran separadas en la carpeta Vista/html.

Controlador

Controlador.php

```php
<?php
class Controlador
{
    public function verPagina($ruta)
    {
        require_once $ruta;
    }
    public function agregarCita($doc, $med, $fec, $hor, $con)
    {
        $cita = new Cita(null, $fec, $hor, $doc, $med, $con, "Solicitada",
"Ninguna");
        $gestorCita = new GestorCita();
        $id = $gestorCita->agregarCita($cita);
        $result = $gestorCita->consultarCitaPorId($id);
        require_once 'Vista/html/confirmarCita.php';
    }
    public function consultarCitas($doc)
    {
        $gestorCita = new GestorCita();
        $result = $gestorCita->consultarCitasPorDocumento($doc);
        require_once 'Vista/html/consultarCitas.php';
    }
    public function cancelarCitas($doc)
```

```
    {
        $gestorCita = new GestorCita();
        $result = $gestorCita->consultarCitasPorDocumento($doc);
        require_once 'Vista/html/cancelarCitas.php';
    }
    public function consultarPaciente($doc)
    {
        $gestorCita = new GestorCita();
        $result = $gestorCita->consultarPaciente($doc);
        require_once 'Vista/html/consultarPaciente.php';
    }
    public function agregarPaciente($doc, $nom, $ape, $fec, $sex)
    {
        $paciente = new Paciente($doc, $nom, $ape, $fec, $sex);
        $gestorCita = new GestorCita();
        $registros = $gestorCita->agregarPaciente($paciente);
        if ($registros > 0) {
            echo "Se insertó el paciente con exito";
        } else {
            echo "Error al grabar el paciente";
        }
    }
}
```

La clase Controlador actúa como el punto intermedio entre la vista y los modelos en la arquitectura MVC (Modelo-Vista-Controlador). A continuación, se explican sus principales características y funciones:

1. **Método verPagina**:
    - Carga dinámicamente las vistas especificadas a través del parámetro $ruta.
    - Es utilizado para mostrar las diferentes interfaces de usuario según las acciones requeridas.
2. **Método agregarCita**:
    - Crea una nueva instancia de la clase Cita con los datos proporcionados (documento del paciente, médico, fecha, hora, consultorio, y estado inicial).
    - Utiliza el modelo GestorCita para agregar la cita a la base de datos.
    - Posteriormente, consulta la cita recién creada por su ID y muestra la confirmación a través de la vista confirmarCita.
3. **Método consultarCitas**:

- Llama al modelo GestorCita para recuperar todas las citas asociadas con un documento de paciente.
- Los resultados se pasan a la vista consultarCitas para su visualización.

4. **Método cancelarCitas**:
   - Recupera las citas asociadas a un documento específico utilizando el modelo GestorCita.
   - Los datos se envían a la vista cancelarCitas, que permite realizar la acción de cancelación.
5. **Método consultarPaciente**:
   - Busca información de un paciente en el sistema utilizando el modelo GestorCita.
   - Los resultados se muestran en la vista consultarPaciente.
6. **Método agregarPaciente**:
   - Crea una instancia de la clase Paciente con los datos proporcionados (documento, nombres, apellidos, fecha de nacimiento y sexo).
   - Llama al método del modelo GestorCita para insertar los datos en la base de datos.
   - Proporciona un mensaje de éxito o error dependiendo del resultado de la operación.

Modelo

Cita.php

```php
<?php
class Cita
{
    private $numero;
    private $fecha;
    private $hora;
    private $paciente;
    private $medico;
    private $consultorio;
    private $estado;
    private $observaciones;
    public function __construct($num, $fec, $hor, $pac, $med, $con, $est,
$obs)
    {
```

```
        $this->numero = $num;
        $this->fecha = $fec;
        $this->hora = $hor;
        $this->paciente = $pac;
        $this->medico = $med;
        $this->consultorio = $con;
        $this->estado = $est;
        $this->observaciones = $obs;
    }
    public function obtenerNumero()
    {
        return $this->numero;
    }
    public function obtenerFecha()
    {
        return $this->fecha;
    }
    public function obtenerHora()
    {
        return $this->hora;
    }
    public function obtenerPaciente()
    {
        return $this->paciente;
    }
    public function obtenerMedico()
    {
        return $this->medico;
    }
    public function obtenerConsultorio()
    {
        return $this->consultorio;
    }
    public function obtenerEstado()
    {
        return $this->estado;
    }
    public function obtenerObservaciones()
    {
        return $this->observaciones;
    }
}
```

La clase Cita representa un modelo para gestionar las citas médicas en un sistema. A continuación, se detallan los aspectos más importantes:

Atributos:

1. **$numero**:
    - Identificador único para la cita.
2. **$fecha**:
    - Fecha en la que se llevará a cabo la cita.
3. **$hora**:
    - Hora programada para la cita.
4. **$paciente**:
    - Documento o identificador del paciente que solicita la cita.
5. **$medico**:
    - Identificador o nombre del médico asignado a la cita.
6. **$consultorio**:
    - Consultorio donde se llevará a cabo la cita.
7. **$estado**:
    - Estado actual de la cita (ejemplo: "Solicitada", "Confirmada", "Cancelada").
8. **$observaciones**:
    - Notas adicionales relacionadas con la cita.

Constructor:

- El método __construct permite inicializar una instancia de Cita con todos los atributos necesarios.
- Los parámetros corresponden a cada atributo de la clase.

Explicación del Código: Clase Cita

La clase Cita representa un modelo para gestionar las citas médicas en un sistema. A continuación, se detallan los aspectos más importantes:

Atributos:

1. **$numero**:
    - Identificador único para la cita.
2. **$fecha**:

- Fecha en la que se llevará a cabo la cita.

3. **$hora**:
    - Hora programada para la cita.
4. **$paciente**:
    - Documento o identificador del paciente que solicita la cita.
5. **$medico**:
    - Identificador o nombre del médico asignado a la cita.
6. **$consultorio**:
    - Consultorio donde se llevará a cabo la cita.
7. **$estado**:
    - Estado actual de la cita (ejemplo: "Solicitada", "Confirmada", "Cancelada").
8. **$observaciones**:
    - Notas adicionales relacionadas con la cita.

Constructor:

- El método __construct permite inicializar una instancia de Cita con todos los atributos necesarios.
- Los parámetros corresponden a cada atributo de la clase.

Métodos Públicos:

Cada atributo cuenta con un método público para obtener su valor:

1. obtenerNumero(): Devuelve el número de la cita.
2. obtenerFecha(): Devuelve la fecha de la cita.
3. obtenerHora(): Devuelve la hora de la cita.
4. obtenerPaciente(): Devuelve el identificador del paciente.
5. obtenerMedico(): Devuelve el médico asignado.
6. obtenerConsultorio(): Devuelve el consultorio asignado.
7. obtenerEstado(): Devuelve el estado actual de la cita.
8. obtenerObservaciones(): Devuelve las observaciones de la cita.

Ventajas y Aplicaciones

1. **Encapsulación**:
    - Los atributos están definidos como privados, lo que evita accesos directos desde fuera de la clase.
    - Los métodos públicos permiten un acceso controlado a los valores de los atributos.
2. **Flexibilidad**:
    - La clase se puede reutilizar en distintos escenarios donde se necesite gestionar citas.
3. **Facilidad de Integración**:
    - Este modelo se puede integrar con otros componentes, como un gestor de base de datos o un controlador que maneje las operaciones CRUD (crear, leer, actualizar y eliminar).

Conexión.php

```php
<?php
class Conexion
{
    private $mySQLI;
    private $sql;
    private $result;
    private $filasAfectadas;
    private $citaId;
    public function abrir()
    {
        $this->mySQLI = new mysqli("localhost", "root", "", "citas");
        if (mysqli_connect_error()) {
            return 0;
        } else {
            return 1;
        }
    }
    public function cerrar()
    {
        $this->mySQLI->close();
    }
    public function consulta($sql)
    {
        $this->sql = $sql;
        $this->result = $this->mySQLI->query($this->sql);
        $this->filasAfectadas = $this->mySQLI->affected_rows;
```

```
        $this->citaId = $this->mySQLI->insert_id;
    }
    public function obtenerResult()
    {
        return $this->result;
    }
    public function obtenerFilasAfectadas()
    {
        return $this->filasAfectadas;
    }
    public function obtenerCitaId()
    {
        return $this->citaId;
    }
}
```

La clase Conexion está diseñada para gestionar las operaciones básicas de conexión y consulta a una base de datos MySQL. Es un componente clave en un sistema que interactúa con bases de datos.

Atributos:

1. **$mySQLI**:
    - Objeto que representa la conexión activa con la base de datos.
2. **$sql**:
    - Almacena la consulta SQL que se va a ejecutar.
3. **$result**:
    - Almacena el resultado de la consulta SQL.
4. **$filasAfectadas**:
    - Cantidad de filas afectadas por la consulta ejecutada.
5. **$citaId**:
    - Identificador de la última fila insertada (útil para operaciones INSERT).

Métodos:

1. **abrir()**:
    - Establece una conexión con la base de datos utilizando mysqli.
    - Parámetros de conexión:
        - **Host**: localhost.
        - **Usuario**: root.

        - **Contraseña**: "" (vacía, común en entornos locales).
        - **Base de datos**: citas.
    - Devuelve 1 si la conexión es exitosa y 0 si hay un error.
2. **cerrar()**:
    - Cierra la conexión activa con la base de datos.
3. **consulta($sql)**:
    - Recibe una consulta SQL como parámetro y la ejecuta.
    - Almacena:
        - El resultado en $result.
        - Las filas afectadas en $filasAfectadas.
        - El ID de la última fila insertada en $citaId (si la consulta fue un INSERT).
4. **obtenerResult()**:
    - Devuelve el resultado de la consulta ejecutada (útil para consultas SELECT).
5. **obtenerFilasAfectadas()**:
    - Devuelve la cantidad de filas afectadas por la última consulta ejecutada.
6. **obtenerCitaId()**:
    - Devuelve el ID de la última fila insertada.

Ventajas y Aplicaciones

1. **Centralización de la Conexión**:
    - Esta clase centraliza las operaciones de conexión, facilitando la reutilización y mantenimiento del código.
2. **Facilidad de Uso**:
    - Proporciona métodos claros y directos para abrir/cerrar conexiones y ejecutar consultas.
3. **Compatibilidad con Operaciones CRUD**:
    - Puede integrarse fácilmente con otros componentes del sistema para realizar operaciones de creación, lectura, actualización y eliminación en la base de datos.

Gestorcita.php

```php
<?php
class GestorCita
{
    public function agregarCita(Cita $cita)
    {
        $conexion = new Conexion();
        $conexion->abrir();
        $fecha = $cita->obtenerFecha();
        $hora = $cita->obtenerHora();
        $paciente = $cita->obtenerPaciente();
        $medico = $cita->obtenerMedico();
        $consultorio = $cita->obtenerConsultorio();
        $estado = $cita->obtenerEstado();
        $observaciones = $cita->obtenerObservaciones();
        $sql = "INSERT INTO citas VALUES ( null,'$fecha','$hora',
'$paciente','$medico','$consultorio','$estado','$observaciones')";
        $conexion->consulta($sql);
        $citaId = $conexion->obtenerCitaId();
        $conexion->cerrar();
        return $citaId;
    }
    public function consultarCitaPorId($id)
    {
        $conexion = new Conexion();
        $conexion->abrir();
        $sql = "SELECT pacientes.* , pedicos.*, consultorios.*, citas.*"
            . "FROM Pacientes as pacientes, Medicos as medicos, Consultorios
        as consultorios ,citas "
            . "WHERE citas.CitPaciente = pacientes.PacIdentificacion "
            . " AND citas.CitMedico = medicos.MedIdentificacion "
            . " AND citas.CitNumero = $id";
        $conexion->consulta($sql);
        $result = $conexion->obtenerResult();
        $conexion->cerrar();
        return $result;
    }
    public function consultarCitasPorDocumento($doc)
    {
        $conexion = new Conexion();
        $conexion->abrir();
        $sql = "SELECT * FROM citas "
            . "WHERE CitPaciente = '$doc' "
            . " AND CitEstado = 'Solicitada' ";
        $conexion->consulta($sql);
```

```
        $result = $conexion->obtenerResult();
        $conexion->cerrar();
        return $result;
    }
    public function consultarPaciente($doc)
    {
        $conexion = new Conexion();
        $conexion->abrir();
        $sql = "SELECT * FROM Pacientes WHERE PacIdentificacion = '$doc' ";
        $conexion->consulta($sql);
        $result = $conexion->obtenerResult();
        $conexion->cerrar();
        return $result;
    }
    public function agregarPaciente(Paciente $paciente){
        $conexion = new Conexion();
        $conexion->abrir();
        $doc = $paciente->obtenerIdentificacion();
        $nom = $paciente->obtenerNombres();
        $ape = $paciente->obtenerApellidos();
        $fec = $paciente->obtenerFechaNacimiento();
        $sex = $paciente->obtenerSexo();
        $sql = "INSERT INTO pacientes VALUES (
'$doc','$nom','$ape','$fec','$sex')";
        $conexion->consulta($sql);
        $conexion->cerrar();
    }
}
```

La clase GestorCita actúa como intermediaria entre la lógica de negocio y la base de datos. Su objetivo es realizar operaciones relacionadas con las citas y los pacientes en la base de datos.

Métodos Principales

1. **agregarCita(Cita $cita)**
    - Este método inserta una nueva cita en la base de datos.
    - Extrae los atributos de un objeto de la clase Cita (fecha, hora, paciente, médico, consultorio, etc.).
    - Ejecuta un comando INSERT INTO para almacenar estos datos.
    - Devuelve el ID de la cita recién creada.
2. **consultarCitaPorId($id)**

- Busca una cita específica por su ID.
- Realiza un SELECT que une varias tablas relacionadas: Pacientes, Medicos, Consultorios y Citas.
- Retorna el resultado de la consulta.

3. **consultarCitasPorDocumento($doc)**
    - Busca todas las citas asociadas a un paciente usando su documento de identificación.
    - Filtra las citas que están en estado Solicitada.
4. **consultarPaciente($doc)**
    - Recupera la información de un paciente específico por su documento de identificación.
    - Inserta un nuevo registro en la tabla Pacientes.
    - Utiliza los métodos de la clase Paciente para obtener los datos necesarios (documento, nombre, apellidos, fecha de nacimiento, sexo).

Ventajas de la Implementación

1. **Separación de Responsabilidades**:
    - Mantiene la lógica de la base de datos separada de la lógica de presentación y negocio.
2. **Reutilización**:
    - Los métodos pueden ser invocados desde diferentes partes del sistema, permitiendo un diseño más modular.
3. **Manejo Eficiente de la Base de Datos**:
    - Cada operación abre y cierra explícitamente la conexión, minimizando riesgos de saturación del servidor.

Mejoras Potenciales

1. **Validación y Escapado de Datos**:
    - Utilizar sentencias preparadas para prevenir inyecciones SQL, en lugar de interpolar datos directamente en las consultas.
2. **Manejo de Excepciones**:
    - Agregar bloques try-catch para capturar y manejar errores en las consultas o la conexión.
3. **Logs de Auditoría**:

- Registrar las consultas realizadas y los resultados obtenidos para monitorear el sistema y depurar errores.

4. **Generalización de Consultas**:
   - Crear métodos más genéricos para ejecutar consultas (SELECT, INSERT, UPDATE, DELETE) que reciban parámetros dinámicos, reduciendo la redundancia.

Paciente.php

```php
<?php
class Paciente
{
    private $identificacion;
    private $nombres;
    private $apellidos;
    private $fechaNacimiento;
    private $sexo;
    public function __construct($ide, $nom, $ape, $fNa, $sex)
    {
        $this->identificacion = $ide;
        $this->nombres = $nom;
        $this->apellidos = $ape;
        $this->fechaNacimiento = $fNa;
        $this->sexo = $sex;
    }
    public function obtenerIdentificacion()
    {
        return $this->identificacion;
    }
    public function obtenerNombres()
    {
        return $this->nombres;
    }
    public function obtenerApellidos()
    {
        return $this->apellidos;
    }
    public function obtenerFechaNacimiento()
    {
        return $this->fechaNacimiento;
    }
    public function obtenerSexo()
    {
        return $this->sexo;
```

```
    }
}
```

La clase Paciente define una estructura de datos para almacenar y manipular la información básica de un paciente en un sistema de gestión de citas médicas. Esencialmente, encapsula los detalles personales de un paciente y proporciona métodos para acceder a ellos de manera controlada.

Atributos

1. **$identificacion**:
    - Representa un identificador único para el paciente, como un número de documento.
2. **$nombres**:
    - Almacena el o los nombres del paciente.
3. **$apellidos**:
    - Contiene los apellidos del paciente.
4. **$fechaNacimiento**:
    - Especifica la fecha de nacimiento del paciente, posiblemente en formato YYYY-MM-DD.
5. **$sexo**:
    - Indica el género del paciente (por ejemplo, "M" para masculino o "F" para femenino).

Constructor

__construct($ide, $nom, $ape, $fNa, $sex)
El constructor inicializa un objeto Paciente con la información básica del mismo.

- **Parámetros:**
    - $ide: Identificación del paciente.
    - $nom: Nombres del paciente.
    - $ape: Apellidos del paciente.
    - $fNa: Fecha de nacimiento.
    - $sex: Género.

Métodos Públicos

1. **obtenerIdentificacion()**

- Retorna el número de identificación del paciente.

2. **obtenerNombres()**
    - Devuelve el nombre o nombres del paciente.
3. **obtenerApellidos()**
    - Proporciona los apellidos del paciente.
4. **obtenerFechaNacimiento()**
    - Retorna la fecha de nacimiento del paciente.
5. **obtenerSexo()**
    - Devuelve el género del paciente.

Ventajas de la Implementación

1. **Encapsulación**:
    - Protege los datos del paciente al hacerlos privados y solo accesibles mediante métodos públicos.
2. **Reutilización**:
    - Puede integrarse fácilmente con otras clases, como gestores o controladores, para manipular la información del paciente.
3. **Legibilidad y Mantenibilidad**:
    - Cada método tiene una función clara, lo que facilita su uso y comprensión.

Vista

Css

estilos.css

```css
body {
background-color: #dfdfdf;
}

#contenedor {
    font-size:1em;
    background-color: #ffffff;
    border:#d3d3d3 1px solid;
    text-align: justify;
    margin: 10px auto;
    width : 900px ;
}
```

```css
#encabezado {
    width: 900px;
    height: 150px;
    background : url('../imagenes/Untitled.png') no-repeat ;
    border: #d3d3d3 1px solid ;
    }
    #encabezado h1 {
        text-align: center ;
        margin-top: 50px;
    }
    #contenido {
        border: #d3d3d3 1px solid ;
        margin: 20px 20px 10px;
        padding: 10px 15px;
     }
     #contenido h2 {
        color : #0075ff;
        border-bottom: #CFBFB5 1px solid;
    }
#menu {
    float: left;
    width: 120px;
}
div#contenido {
    border: #d3d3d3 1px solid ;
    margin: 20px 20px 10px 180px;
    padding: 10px 15px;
}
#menu {
    float: left;
    width: 120px;
    background-color: #D4FFFF ;
}
#menu li {
    border: #d3d3d3 3px solid;
    background-color: #ffffff;
    margin-bottom: 5px;
    font-size: 0.9em;
    text-align: center;
    line-height: 30px;
}
#menu a {
    display: block;
    text-decoration: none;
    color : #000000;
```

```
}
#menu li:hover {
    background-color: #4294ff;
}
#menu li:activa,#menu li:hover {
    background-color: #7092be;
}
```

Este código CSS establece el diseño visual y la estructura básica para una página web. A continuación, se explica en detalle cada sección:

Estilos Globales

- Configura un color de fondo gris claro para toda la página (body).

Contenedor Principal

- **Propósito**: Define un contenedor principal centrado con un ancho fijo de 900px.
- **Características**:
    - Fondo blanco.
    - Borde gris claro.
    - Texto justificado.
    - Centración automática (margin: 10px auto).

Encabezado

- **Propósito**: Diseña la sección del encabezado con un fondo personalizado (Untitled.png).
- **Características**:
    - Tamaño: 900px de ancho y 150px de alto.
    - Borde gris claro.
    - Título (h1) centrado con un margen superior de 50px.

Contenido Principal

- **Propósito**: Define el área principal de contenido.
- **Características**:
    - Borde gris claro.
    - Márgenes y relleno definidos para espaciado interno y externo.
    - Títulos (h2) con color azul (#0075ff) y una línea de separación en la parte inferior.

Menú de Navegación

- **Propósito**: Estiliza un menú de navegación vertical ubicado a la izquierda.
- **Características**:
    - **#menu**:
        - Flotante a la izquierda con un ancho de 120px.
        - Fondo azul claro (#D4FFFF).
    - **#menu li**:
        - Elementos de lista con borde gris, fondo blanco, y márgenes inferiores.
        - Texto centrado vertical y horizontalmente.
    - **#menu a**:
        - Enlaces sin subrayado y color negro.
    - **Hover y Activo**:
        - Cambia el fondo a azul (#4294ff) cuando el cursor está sobre un elemento.
        - Color diferente para el estado activo y hover simultáneo.

Diseño Ajustado para Contenido y Menú

- Ajusta el margen izquierdo del contenido principal (#contenido) para acomodarse al menú flotante (#menu).

Js

script.js

```
function consultarPaciente() {
    var url = "index.php?accion=consultarPaciente&documento=" +
        $("#asignarDocumento").val();
    $("#paciente").load(url, function () {
    });
}
$(document).ready(function () {
    $("#frmPaciente").dialog({
        autoOpen: false,
        height: 310,
        width: 400,
        modal: true,
        buttons: {
            "Insertar": insertarPaciente,
            "Cancelar": cancelar
        }
```

```
    });
});
function consultarPaciente() {
    var url = "index.php?accion=consultarPaciente&documento=" +
        $("#asignarDocumento").val();
    $("#paciente").load(url, function () {
    });
}
function mostrarFormulario() {
    documento = "" + $("#asignarDocumento").val();
    $("#PacDocumento").attr("value", documento);
    $("#frmPaciente").dialog('open');
}
function insertarPaciente() {
    queryString = $("#agregarPaciente").serialize();
    url = "index.php?accion=ingresarPaciente&" + queryString;
    $("#paciente").load(url);
    $("#frmPaciente").dialog('close');
}
function cancelar() {
    $(this).dialog('close');
}
```

Este fragmento de código implementa funcionalidades relacionadas con la consulta, inserción y manejo de pacientes a través de una interfaz de usuario interactiva. A continuación, se explica cada parte

1. consultarPaciente

- **Propósito**: Consulta información de un paciente utilizando su documento de identificación.
- **Funcionamiento**:
    1. Construye la URL de consulta (index.php) con parámetros accion y documento.
    2. Obtiene el valor del campo con ID asignarDocumento.
    3. Usa el método load para cargar el contenido desde la URL en el elemento con ID paciente.

2. Inicialización del Diálogo ($(document).ready)

- **Propósito**: Configura un cuadro de diálogo (dialog) para el formulario de pacientes.
- **Características**:
    - **autoOpen: false**: El diálogo no se abre automáticamente al cargar la página.

- Dimensiones: Altura (310px) y ancho (400px).
- **modal: true**: Hace que el diálogo sea modal (bloquea la interacción con el resto de la página).
- Define botones:
  - **Insertar**: Ejecuta la función insertarPaciente.
  - **Cancelar**: Ejecuta la función cancelar.

3. mostrarFormulario

- **Propósito**: Prepara y abre el formulario para agregar un nuevo paciente.
- **Funcionamiento**:
  1. Recupera el valor del campo asignarDocumento y lo asigna a la variable documento.
  2. Actualiza el valor del campo PacDocumento en el formulario.
  3. Abre el cuadro de diálogo (dialog) del formulario.

4. insertarPaciente

- **Propósito**: Envía los datos del formulario para registrar un nuevo paciente.
- **Funcionamiento**:
  1. Serializa los datos del formulario con ID agregarPaciente en un formato adecuado para la URL.
  2. Construye la URL con la acción ingresarPaciente y los datos del formulario.
  3. Usa load para actualizar el contenido del elemento #paciente con la respuesta del servidor.
  4. Cierra el cuadro de diálogo (dialog).

5. cancelar

- **Propósito**: Cierra el cuadro de diálogo.
- **Funcionamiento**:
  - Utiliza $(this) para referirse al cuadro de diálogo actual y cierra el formulario modal.

Html

Asignar.php

```
<!DOCTYPE html>
```

```
<html>

<head>
    <title>Sistema de Gestión Odontológica</title>
    <link rel="stylesheet" type="text/css" href="Vista/css/estilos.css">
    <script src="Vista/jquery/jquery-3.7.1.min.js"
type="text/javascript"></script>
    <script src="Vista/jquery/jquery-ui.js"
        type="text/javascript"></script>
    <link href="Vista/jquery/jquery-ui.min.css"
        rel="stylesheet" type="text/css" />
    <script src="Vista/js/script.js" type="text/javascript"></script>
    <script>
    </script>
</head>

<body>
    <div id="contenedor">
        <div id="encabezado">
            <h1>Sistema de Gestión Odontológica</h1>
        </div>
        <ul id="menu">
            <li><a href="index.php">inicio</a> </li>
            <li class="active"><a
href="index.php?accion=asignar">Asignar</a> </li>
            <li><a href="index.php?accion=consultar">Consultar Cita</a>
</li>
            <li><a href="index.php?accion=cancelar">Cancelar Cita</a> </li>
        </ul>
        <div id="contenido">
            <h2>Asignar citas</h2>
            <form id="frmasignar" action="index.php?accion=guardarCita"
method="post">
                <table>
                    <tr>
                        <td>Documento del paciente</td>
                        <td><input type="text" name="asignarDocumento"
id="asignarDocumento">
                                    td>
                    </tr>
                    <tr>
                        <td colspan="2"><input type="button"
value="Consultar"
                                    name="asignarConsultar"
id="asignarConsultar"></td>
```

```
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="paciente"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Médico</td>
                        <td>
                            <select id="medico" name="medico">
                                <option value="-1" selected="selected">---
Selecione el
                                    Médico</option>
                                <option value="12345">12345-Pepito
Pérez</option>
                                <option value="67890">67890-Pepita
Mendieta</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td>Fecha</td>
                        <td>
                            <input type="date" id="fecha" name="fecha">
                        </td>
                    </tr>
                    <tr>
                        <td>Hora</td>
                        <td>
                            <select id="hora" name="hora">
                                <option value="-1" selected="selected">---
Seleccione la
                                    hora ---</option>
                                <option>08:00:00</option>
                                <option>08:20:00</option>
                                <option>08:40:00</option>
                                <option>09:00:00</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td>Consultorio</td>
                        <td>
                            <select id="consultorio" name="consultorio">
```

```
                                <option value="-1" selected="selected">---
Seleccione el
                                    Consultorio---</option>
                                <option value="1">1 Consultas 1</option>
                                <option value="2">2 Tratamientos 1</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <input type="submit" name="asignarEnviar"
value="Enviar"
                                id="asignarEnviar">
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2"><input type="button"
value="Consultar" name="asignarConsultar"
                                id="asignarConsultar"
onclick="consultarPaciente()"></td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
    <div id="frmPaciente" title="Agregar Nuevo Paciente">
        <form id="agregarPaciente">
            <table>
                <tr>
                    <td>Documento</td>

                    <td><input type="text" name="PacDocumento"

                            id="PacDocumento"></td>
                </tr>
                <tr>

                    <td>Nombres</td>

                    <td><input type="text" name="PacNombres"

                            id="PacNombres"></td>
                </tr>
                <tr>
```

```
                    <td>Apellidos</td>

                    <td><input type="text" name="PacApellidos"

                            id="PacApellidos"></td>
                </tr>
                <tr>

                    <td>Fecha de Nacimiento</td>

                    <td><input type="text" name="PacNacimiento"

                            id="PacNacimiento"></td>
                </tr>
                <tr>

                    <td>Sexo</td>

                    <td>

                        <select id="pacSexo" name="PacSexo">
                            <option value="-1"

                                FAVA - Formación en Ambientes Virtuales de
Aprendizaje

                                SENA - Servicio Nacional de Aprendizaje. 85

                                Desarrollo de aplicaciones web en PHP

                                selected="selected">--Selecione el sexo ---
</option>
                            <option value="M">Masculino</option>
                            <option value="F">Femenino</option>

                        </select>
                    </td>
                </tr>
            </table>
        </form>
    </div>
</body>

</html>
```

El código presenta una interfaz web para un **Sistema de Gestión Odontológica** que permite asignar citas, consultar información, y gestionar datos de pacientes. Incluye formularios, menús de navegación y elementos dinámicos para la interacción con usuarios. A continuación, se desglosan sus componentes principales.

Estructura General del Documento

1. **Encabezado (<head>):**
    - Incluye hojas de estilo para la personalización visual.
    - Importa bibliotecas de JavaScript:
        - jQuery (jquery-3.7.1.min.js) y jQuery UI (jquery-ui.js) para interactividad y elementos de interfaz.
        - Archivo de scripts personalizado (script.js).
2. **Cuerpo (<body>):**
    - Contiene un contenedor principal (#contenedor) con un encabezado, menú de navegación y un área de contenido.

Sección del Menú

- **Propósito**: Facilitar la navegación entre las diferentes funcionalidades del sistema.
- Contiene enlaces a las acciones principales:
    - **Inicio**
    - **Asignar Cita**: Página actual (active).
    - **Consultar Cita**
    - **Cancelar Cita**

Formulario para Asignar Citas

Componentes Clave:

- **Campo Documento:** Permite ingresar el número de identificación del paciente para consultarlo.
- **Datos de la Cita:** Incluyen médico, fecha, hora y consultorio.
- **Botón Enviar:** Envía la información para registrar una nueva cita.

Formulario Modal para Nuevo Paciente

Características:

- Se muestra al agregar un paciente nuevo.
- Incluye campos básicos (documento, nombres, apellidos, fecha de nacimiento, sexo).

Interacción con JavaScript

1. **Consultar Paciente:**
    - El botón Consultar ejecuta la función consultarPaciente, cargando datos relacionados en el elemento #paciente.
2. **Asignar Cita:**
    - Al enviar el formulario, los datos se envían al servidor (index.php) para registrar la cita.
3. **Agregar Paciente:**
    - Se utiliza el cuadro de diálogo modal #frmPaciente, controlado con jQuery UI.

Cancelar.php

```
<!DOCTYPE html>
<html>

<head>
    <title>Sistema de Gestión Odontológica</title>
    <link rel="stylesheet" type="text/css" href="Vista/css/estilos.css">
</head>

<body>
    <div id="contenedor">
        <div id="encabezado">
            <h1>Sistema de Gestión Odontológica</h1>
        </div>
        <ul id="menu">
            <li><a href="index.php">inicio</a> </li>
            <li><a href="index.php?accion=asignar">Asignar</a> </li>
            <li><a href="index.php?accion=consultar">Consultar Cita</a>
</li>
            <li><a href="index.php?accion=cancelar">Cancelar Cita</a> </li>
        </ul>
        <div id="contenido">
            <h2>Cancelar Cita</h2>
            <form action="index.php?accion=cancelarCita" method="post"
                id="frmcancelar">
                <table>
                    <tr>
                        <td>Documento del Paciente </td>

                        <td><input type="text" name="cancelarDocumento"
```

```
                                id="cancelarDocumento"></td>
                    </tr>
                    <tr>
                        <td colspan="2"><input type="submit"
                                name="cancelarConsultar" value="Consultar"
id="cancelarConsultar"></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="paciente3"></div>
                        </td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
</body>

</html>
```

Este código HTML presenta una página dentro de un **Sistema de Gestión Odontológica** que permite a los usuarios cancelar una cita. La página sigue el diseño y estructura general del sistema, incluyendo encabezado, menú de navegación y un área de contenido específica para esta funcionalidad.

Estructura del Código

1. **Encabezado del Documento (<head>):**
    - Contiene el título de la página y una referencia a la hoja de estilos para darle formato.
2. **Cuerpo de la Página (<body>):**
    - **Contenedor Principal (#contenedor):** Incluye las siguientes secciones:
        - **Encabezado (#encabezado):** Muestra el título principal del sistema.
        - **Menú de Navegación (#menu):** Proporciona enlaces a las principales funcionalidades del sistema. Aquí, el enlace **Cancelar Cita** lleva al usuario a la página actual.
        - **Contenido Principal (#contenido):** Contiene un formulario para cancelar citas.

Análisis de los Elementos del Formulario

1. **Campo de Documento del Paciente:**

- Permite ingresar el número de documento del paciente para consultar la cita que se desea cancelar.

2. **Botón Consultar:**
    - Envía el número de documento al servidor mediante el método POST para procesar la consulta.
3. **Espacio para Información del Paciente:**
    - Se utiliza para mostrar la información del paciente y detalles de la cita a cancelar.

Flujo del Proceso de Cancelación

1. **Entrada del Documento del Paciente:**
    - El usuario ingresa el número de identificación en el campo correspondiente.
2. **Consulta de Datos:**
    - Al presionar el botón **Consultar**, el formulario envía los datos al servidor (index.php?accion=cancelarCita).
    - El servidor busca la cita correspondiente y devuelve la información del paciente o un mensaje de error si no se encuentra la cita.
3. **Confirmación de Cancelación:**
    - Aunque no está explícitamente indicado en este código, se podría agregar un paso adicional para confirmar la cancelación de la cita.

cancelarCitas.php

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>

<head>
    <meta charset="UTF-8">
    <title></title>
</head>

<body>
    <?php
    if ($result->num_rows > 0) {
```

```
    ?>
        <table>
            <tr>
                <th>Número</th>
                <th>Fecha</th>
                <th>Hora</th>
            </tr>
            <?php
            while ($fila = $result->fetch_object()) {
            ?>
                <tr>
                    <td><?php echo $fila->CitNumero; ?></td>
                    <td><?php echo $fila->CitFecha; ?></td>
                    <td><?php echo $fila->CitHora; ?></td>
                    <td><a href="#" onclick="confirmarCancelar(<?php echo
$fila->CitNumero; ?>)">Cancelar</a></td>
                </tr>
            <?php
            }
            ?>
        </table>
    <?php
    } else {
    ?>
        <p>El paciente no tiene citas asignadas</p>
    <?php
    }
    ?>
</body>

</html>
```

Este código genera dinámicamente una tabla en HTML que muestra las citas asignadas a un paciente. Se incluye la funcionalidad para cancelar una cita específica mediante un enlace que ejecuta una función JavaScript.

Estructura del Código

1. **Encabezado del Documento (<head>):**
    - Establece la codificación de caracteres como UTF-8 para garantizar la compatibilidad con caracteres especiales y acentos.
2. **Cuerpo del Documento (<body>):**

- Contiene un bloque de código PHP que verifica si hay citas registradas para el paciente en la variable $result (resultado de una consulta previa a la base de datos).

Flujo de Ejecución

1. **Verificación de Datos Disponibles:**
    - Comprueba si el objeto $result contiene al menos una fila de datos. Si hay citas registradas, se genera una tabla en HTML.
2. **Generación de la Tabla:**
    - Se crea la estructura inicial de la tabla, con encabezados para mostrar el **número de cita**, **fecha** y **hora**.
3. **Iteración sobre las Citas:**
    - Recorre cada fila del resultado de la consulta y genera dinámicamente una fila en la tabla para cada cita encontrada.
4. **Contenido Dinámico de la Fila:**
    - Muestra:
        - **Número de cita:** $fila->CitNumero
        - **Fecha:** $fila->CitFecha
        - **Hora:** $fila->CitHora
    - Incluye un enlace con un evento onclick para cancelar la cita. El enlace llama a una función JavaScript, confirmarCancelar, pasando como argumento el número de la cita.
5. **Mensaje cuando No Hay Citas:**
    - Si no hay resultados en $result, se muestra este mensaje.

Análisis de Funcionalidades Clave

1. **Tabla Dinámica:**
    - La tabla se genera automáticamente según los resultados de la consulta, asegurando que los datos mostrados estén actualizados.
2. **Función JavaScript para Cancelar:**
    - El enlace de cada fila llama a una función JavaScript para confirmar la cancelación:
    - Aunque no está definida en el código proporcionado, esta función podría abrir un cuadro de confirmación y, si se acepta, enviar una solicitud al servidor para cancelar la cita.

3. **Diseño Responsivo:**
    - Al ser código HTML básico, es fácil de integrar con estilos CSS para mejorar la presentación y hacerlo más atractivo.

confirmarCita.php

```
<!DOCTYPE html>
<html>

<head>
    <title>Sistema de Gestión Odontológica</title>
    <link rel="stylesheet" type="text/css" href="Vista/css/estilos.css">
</head>

<body>
    <div id="contenedor">
        <div id="encabezado">
            <h1>Sistema de Gestión Odontológica</h1>
        </div>
        <ul id="menu">
            <li class="activa"><a href="index.php">inicio</a> </li>
            <li><a href="index.php?accion=asignar">Asignar</a> </li>
            <li><a href="index.php?accion=consultar">Consultar Cita</a>
</li>
            <li><a href="index.php?accion=cancelar">Cancelar Cita</a> </li>
        </ul>
        <div id="contenido">
            <?php $fila = $result->fetch_object(); ?>
            <h2>Información Cita</h2>
            <table>
                <tr>
                    <th colspan="2">Datos del Paciente</th>
                </tr>
                <tr>
                    <td>Documento</td>
                    <td><?php echo $fila->PacIdentificacion; ?></td>
                </tr>
                <tr>
                    <td>Nombre</td>
                    <td><?php echo $fila->PacNombres . " " . $fila-
>PacApellidos; ?></td>
                </tr>
                <tr>
                    <th colspan="2">Datos del Médico</th>
```

```
                </tr>
                <tr>
                    <td>Documento</td>
                    <td><?php echo $fila->MedIdentificacion; ?></td>
                </tr>
                <tr>
                    <td>Nombre</td>
                    <td><?php echo $fila->MedNombres . " " . $fila-
>MedApellidos; ?></td>
                </tr>
                <tr>
                    <th colspan="2">Datos de la Cita</th>
                </tr>
                <tr>
                    <td>Número</td>
                    <td><?php echo $fila->CitNumero; ?></td>
                </tr>
                <tr>
                    <td>Fecha</td>
                    <td><?php echo $fila->CitFecha; ?></td>
                </tr>
                <tr>
                    <td>Hora</td>
                    <td><?php echo $fila->CitHora; ?></td>
                </tr>
                <tr>
                    <td>Número de Consultorio</td>
                    <td><?php echo $fila->ConNombre; ?></td>
                </tr>
                <tr>
                    <td>Estado</td>
                    <td><?php echo $fila->CitEstado; ?></td>
                </tr>
                <tr>
                    <td>Observaciones</td>
                    <td><?php echo $fila->CitObservaciones; ?></td>
                </tr>
            </table>
        </div>
    </div>
</body>

</html>
```

Este código presenta un diseño estructurado en HTML y PHP para mostrar detalles completos de una cita médica en un sistema de gestión odontológica.

Estructura del Código

1. **Encabezado del Documento (<head>):**
    - Define el título de la página.
    - Incluye la hoja de estilos Vista/css/estilos.css para estilizar el diseño.
2. **Cuerpo del Documento (<body>):**
    - Contiene los elementos principales del sistema: el **contenedor**, el **encabezado**, el **menú de navegación**, y la sección de **contenido** donde se muestra la información de la cita.

Flujo de Ejecución

1. **Obtención de los Datos:**
    - Se obtiene la primera fila de los resultados de la consulta a la base de datos y se almacena en $fila para ser utilizada en la página.
2. **Sección "Información Cita":**
    - Encabezado que describe el contenido que se mostrará a continuación.
3. **Tabla de Detalles:**
    - La tabla está dividida en tres secciones principales:
        - **Datos del Paciente**
        - **Datos del Médico**
        - **Datos de la Cita**
    - Cada sección utiliza un encabezado con colspan="2" para abarcar ambas columnas de la tabla.
    - **Datos del Paciente:**
        - Muestra el documento y nombre completo del paciente.
    - **Datos del Médico:**
        - Muestra el documento y nombre completo del médico asignado.
    - **Datos de la Cita:**
        - Incluye detalles como número, fecha, hora, número de consultorio, estado de la cita, y observaciones adicionales.

Ventajas del Código

1. **Estructura Clara:**
    - El uso de tablas organiza la información en una forma legible y fácil de entender.
2. **Dinamicidad:**
    - Los datos son extraídos dinámicamente de una base de datos, lo que asegura que la información mostrada sea actual.
3. **Reusabilidad:**
    - Esta plantilla puede ser reutilizada para mostrar información de citas de otros pacientes sin cambios significativos en el código.

Consultar.php

```
<!DOCTYPE html>
<html>

<head>
<title>Sistema de Gestión Odontológica</title>
<link rel="stylesheet" type="text/css" href="Vista/css/estilos.css">
<script type="text/javascript" src="Vista/html/jquery/jquery-3.2.1-
min.js"></script>
</head>

<body>
    <div id="contenedor">
        <div id="encabezado">
            <h1>Sistema de Gestión Odontológica</h1>
        </div>
        <ul id="menu">
            <li><a href="index.php">inicio</a> </li>
            <li><a href="index.php?accion=asignar">Asignar</a> </li>
            <li class="activa"><a
href="index.php?accion=consultar">Consultar Cita</a></li>
            <li><a href="index.php?accion=cancelar">Cancelar Cita</a> </li>
        </ul>
        <div id="contenido">
            <h2>Consultar Cita</h2>
            <form action="index.php?accion=consultarCita" method="post"
id="frmconsultar">
                <table>
                    <tr>
                        <td>Documento del paciente</td>
```

```
                        <td><input type="text" name="asignarDocumento"
id="asignarDocumento"></td>
                    </tr>
                    <tr>
                        <td colspan="2"><input type="button"
value="Consultar" name="asignarConsultar"
                                id="asignarConsultar"></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="paciente"></div>
                        </td>
                    </tr>
                    <td colspan="2">
                        <div id="paciente2"></div>
                    </td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
</body>

</html>
```

Este código HTML permite consultar citas en un sistema de gestión odontológica mediante un formulario que solicita el documento de un paciente. Se incluyen elementos interactivos para mostrar los resultados de la consulta.

Estructura del Código

1. **Encabezado del Documento (<head>):**
    - Título de la página: "Sistema de Gestión Odontológica".
    - Referencia a una hoja de estilos externa: Vista/css/estilos.css.
    - Inclusión de la librería **jQuery** para facilitar la manipulación del DOM y la realización de peticiones dinámicas.
2. **Cuerpo del Documento (<body>):**
    - **Contenedor Principal:**
        - Encabezado: muestra el título del sistema.
        - Menú de navegación: permite acceder a diferentes funcionalidades del sistema (Asignar, Consultar, Cancelar).

- **Formulario de Consulta:**
    - Solicita el documento del paciente en un campo de texto.
    - Botón Consultar que activará una acción de búsqueda.
    - Espacios ( ) para mostrar resultados dinámicamente (#paciente y #paciente2).

Flujo de Ejecución

1. **Formulario de Consulta:**
    - El formulario tiene como objetivo enviar el documento del paciente al controlador (index.php) para realizar la consulta. Sin embargo, el botón está configurado para usar JavaScript en lugar de un envío estándar del formulario.
2. **Campos del Formulario:**
    - Entrada de texto para el documento del paciente:
        - Este botón no envía directamente el formulario al servidor, sino que probablemente llama a una función JavaScript que realiza una consulta asincrónica.
3. **Divs para Mostrar Resultados:**
    - Espacios reservados para mostrar los resultados de la consulta (p. ej., información del paciente o sus citas).
4. **Interactividad con jQuery (Suponiendo):**
    - Aunque el archivo JavaScript no se incluye en el fragmento, se espera que al presionar el botón de consulta:
        - Se obtenga el valor del documento del paciente.
        - Se realice una solicitud asíncrona (AJAX) al servidor para buscar las citas correspondientes.
        - Los resultados se inserten dinámicamente en los elementos (#paciente y #paciente2).

Ventajas del Código

1. **Interactividad:**
    - El uso de jQuery facilita una experiencia de usuario fluida y sin necesidad de recargar la página.
2. **Organización:**
    - Estructura clara y dividida en secciones: menú, encabezado, contenido y resultados.

3. **Extensibilidad:**
    - Se puede adaptar fácilmente para incluir validaciones, agregar nuevos campos o modificar el comportamiento de la consulta.

consultarCitas.php

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>

<head>
    <meta charset="UTF-8">
    <title></title>
</head>

<body>
    <?php
    if ($result->num_rows > 0) {
    ?>
        <table>
            <tr>
                <th>Número</th>
                <th>Fecha</th>
                <th>Hora</th>
            </tr>
            <?php
            while ($fila = $result->fetch_object()) {
            ?>
                <tr>
                    <td><?php echo $fila->CitNumero; ?></td>
                    <td><?php echo $fila->CitFecha; ?></td>
                    <td><?php echo $fila->CitHora; ?></td>
                    <td><a href="#" onclick="confirmarConsultar(<?php echo
$fila->CitNumero; ?>)">Consultar</a></td>
                </tr>
            <?php
            }
            ?>
        </table>
    <?php
```

```
    } else {
    ?>
        <p>El paciente tiene citas asignadas</p>
    <?php
    }
    ?>
</body>
```

El código muestra una tabla dinámica de citas asignadas a un paciente, obtenidas desde una consulta a la base de datos. Además, incluye una opción para interactuar con cada cita, activando una función JavaScript.

Flujo de Ejecución

1. **Condicional para Verificar Resultados:**
    - Se verifica si la consulta a la base de datos devuelve resultados (citas asignadas al paciente).
2. **Generación de la Tabla:**
    - Si existen citas, se construye una tabla HTML con las siguientes columnas:
        - **Número de Cita (CitNumero)**
        - **Fecha (CitFecha)**
        - **Hora (CitHora)**
        - **Enlace para Consultar Detalles**
    - Ejemplo de una fila generada dinámicamente:
3. **Manejo de Casos Sin Resultados:**
    - Si no hay citas asignadas, muestra el mensaje:
4. **Interactividad:**
    - Cada cita tiene un enlace que ejecuta la función JavaScript confirmarConsultar, pasando como parámetro el número de la cita:

Ventajas del Código

1. **Generación Dinámica:**
    - El uso de PHP para crear filas dinámicamente asegura que la tabla refleje el contenido de la base de datos en tiempo real.
2. **Interactividad:**
    - La integración con JavaScript (función confirmarConsultar) permite una navegación fluida hacia los detalles de cada cita.

3. **Condicional para Sin Resultados:**
    - Maneja adecuadamente el caso en que no hay citas asignadas, evitando errores o una tabla vacía.

consultarPaciente.php

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title></title>
</head>

<body>
    <?php
    if ($result->num_rows > 0) {
    ?>
        <table>
            <tr>
                <th>Documento</th>
                <th>Nombre Completo</th>
                <th>Sexo</th>
            </tr>
            <?php
            while ($fila = $result->fetch_object()) {
            ?>
                <tr>
                    <td><?php echo $fila->PacIdentificacion; ?></td>
                    <td><?php echo $fila->PacNombres . " " . $fila-
>PacApellidos; ?></td>
                    <td><?php echo $fila->PacSexo; ?></td>
                    <td>Ver</td>
                </tr>
            <?php
            }
            ?>
        </table>
    <?php
    } else {
    ?>
        <p>El paciente no existe en la base de datos.</p>
        <input type="button" name="ingPaciente" id="ingPaciente"
value="Ingresar
```

```
Paciente" onclik="ingPaciente()">
    <?php
    }
    ?>
</body>

</html>
```

Este código genera una tabla dinámica con información básica de los pacientes registrados en la base de datos. También maneja casos en los que no se encuentran registros, permitiendo la opción de ingresar un nuevo paciente.

Estructura del Código

1. **Encabezado del Documento (<head>):**
    - Establece la codificación de caracteres como UTF-8 para soportar caracteres especiales y acentos.
    - Incluye un título vacío que podría ser actualizado para mejorar la claridad del contenido.
2. **Cuerpo del Documento (<body>):**
    - Contiene la lógica principal para mostrar la información de los pacientes o manejar el caso en que no existan registros.

Flujo de Ejecución

1. **Verificación de Resultados:**
    - Comprueba si la consulta a la base de datos devuelve resultados (pacientes registrados). Si hay registros, genera una tabla HTML para mostrar la información.
2. **Generación de la Tabla:**
    - Si existen pacientes, se construye dinámicamente una tabla con las siguientes columnas:
        - **Documento:** $fila->PacIdentificacion
        - **Nombre Completo:** $fila->PacNombres y $fila->PacApellidos
        - **Sexo:** $fila->PacSexo
    - Ejemplo de una fila generada:
3. **Caso Sin Resultados:**
    - Si no se encuentran pacientes en la base de datos:
        - Se muestra un mensaje informativo al usuario.

- Incluye un botón para ingresar un nuevo paciente.
- **Nota:** Hay un error tipográfico en onclik. Debe ser onclick.

Ventajas del Código

1. **Dinamicidad:**
    - Utiliza PHP para generar contenido dinámico en función de los resultados de la consulta.
2. **Manejo de Casos Sin Datos:**
    - Proporciona un mensaje claro y una opción para registrar un nuevo paciente si no existen resultados.
3. **Extensibilidad:**
    - La tabla puede ampliarse fácilmente para incluir más información o acciones, como editar o eliminar pacientes.

Inicio.php

```
<!DOCTYPE html>
<html>

<head>
    <title>Sistema de Gestión Odontológica</title>
    <link rel="stylesheet" type="text/css" href="Vista/css/estilos.css">
</head>

<body>
    <div id="contenedor">
        <div id="encabezado">
            <h1>Sistema de Gestión Odontológica</h1>
        </div>
        <ul id="menu">
            <li><a href="index.php">inicio</a> </li>
            <li><a href="index.php?accion=asignar">Asignar</a> </li>
            <li><a href="index.php?accion=consultar">Consultar Cita</a>
</li>
            <li><a href="index.php?accion=cancelar">Cancelar Cita</a> </li>
        </ul>
        <div id="contenido">
            <h2>Información General</h2>
            <p>El Sistema de Gestión Odontológica permite administrar la
información de los
```

```
                pacientes,tratamientos y citas a través de una interfaz
web.</p>
            <p>El sistema cuenta con las siguientes secciones:
            <ul>
                <li>Asignar cita</li>
                <li>Consultar cita</li>
                <li>Cancelar cita</li>
            </ul>
            </p>
        </div>
    </div>
</body>

</html>
```

Este código HTML describe la página principal de un Sistema de Gestión Odontológica, con un menú de navegación y una sección de información general sobre el sistema. A continuación, se explica cada una de las partes que componen este código:

Estructura del Código

1. **Encabezado (<head>):**
    - Se incluye un título para la página web (<title>) que es "Sistema de Gestión Odontológica".
    - Se hace referencia a un archivo CSS externo para los estilos, ubicado en Vista/css/estilos.css.
2. **Cuerpo del Documento (<body>):**
    - El contenido principal de la página se encuentra dentro de un contenedor , que organiza y agrupa todos los elementos.
    - **Encabezado de la Página ( ):**
        - Contiene un título principal (<h1>) con el nombre del sistema.
    - **Menú de Navegación (<ul id="menu">):**
        - Se incluye una lista de enlaces (
<a>) para navegar entre las diferentes secciones del sistema, como:
            - **Inicio** (index.php)
            - **Asignar cita** (index.php?accion=asignar)
            - **Consultar cita** (index.php?accion=consultar)
            - **Cancelar cita** (index.php?accion=cancelar)

- **Contenido Principal ( ):**
    - Se incluye un subtítulo <h2> con el texto "Información General".
    - Una breve descripción sobre el sistema de gestión odontológica se muestra con los párrafos (<p>).
    - Además, se enumera la funcionalidad del sistema a través de una lista no ordenada (<ul>), destacando tres funciones principales:
        - Asignar cita
        - Consultar cita
        - Cancelar cita

Flujo de Ejecución

- Cuando el usuario visita esta página, se muestra el encabezado con el nombre del sistema y un menú de navegación que le permite ir a diferentes secciones del sistema.
- La sección de contenido proporciona una descripción general del sistema y sus funcionalidades clave, presentadas de forma sencilla y clara para el usuario.

Conclusión

En resumen, el código HTML presentado establece una estructura clara y funcional para el Sistema de Gestión Odontológica, con un menú intuitivo y una sección informativa sobre el sistema. Aunque es un sistema básico, su diseño modular y su enfoque en la usabilidad permiten una gestión eficaz de las citas odontológicas. Para mejorar la experiencia del usuario, sería conveniente optimizar el diseño visual, implementar funcionalidades interactivas adicionales y asegurar la accesibilidad del sitio. Con estas mejoras, el sistema podría convertirse en una herramienta más robusta y eficiente para la gestión de pacientes y citas médicas en clínicas odontológicas.