Informe Gestión Odontológica

Aprendiz

Dylan Robayo Tavera

Julio Roberto Galvis Instructor

Servicio Nacional de Aprendizaje

Análisis y desarrollo de Software

2024

Bogotá D.C

Introducción

El propósito principal de este documento es examinar y registrar el desarrollo de un proyecto enfocado en la creación de un sitio web para la gestión de citas médicas, utilizando el lenguaje PHP. Este ejercicio buscó profundizar en la comprensión de su sintaxis y explorar las diversas herramientas que ofrece, como XAMPP, un entorno que soporta múltiples lenguajes. En el desarrollo se trataron temas como el uso de clases, constructores y formularios, entre otros aspectos clave. El informe también aborda los desafíos encontrados y las estrategias empleadas para solucionarlos. Este enfoque tiene como objetivo destacar tanto las ventajas del lenguaje como la relevancia de aplicar buenas prácticas de programación para asegurar soluciones eficaces, claras y escalables.

Gestión Odontológica

El sistema de gestión odontológica es una aplicación que utiliza HTML y CSS para el diseño frontend, mientras que emplea PHP para la lógica en el backend. En esta implementación, el código estará organizado en tres directorios principales: modelo, vista y controlador, cada uno cumpliendo con un rol específico dentro de la arquitectura del proyecto. Vista: todos los archivos relacionados con HTML y CSS.

- Modelo: archivos que serán funcionales para la ejecución del programa.
- Controlador: archivos que da la funcionalidad de las opciones en el sistema.

Carpeta Vista: Archivo

HTML en formato php

Dentro de su contenido aplica un enlace de referencia a archivo css, también aplica un formulario de asignación en donde se pide diferentes datos del usuario y contiene un menú de navegación en el cual tiene diferentes secciones para la gestión de citas médicas, en donde asigne, consulte, confirme si la cita fue guardada por medio de un programa que almacena y ejecuta la función que dentro del código se haya ejecutado.

1. Archivo html asignar

index.php | plantilla.php | estilos.css | Asignar.php

Fernanda Ibañez > GESTION ODONTOLOGICA > VISTA > HTML > Asignar.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 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 cita</h2>
  <p>Contenido de la página</p>


<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>
```

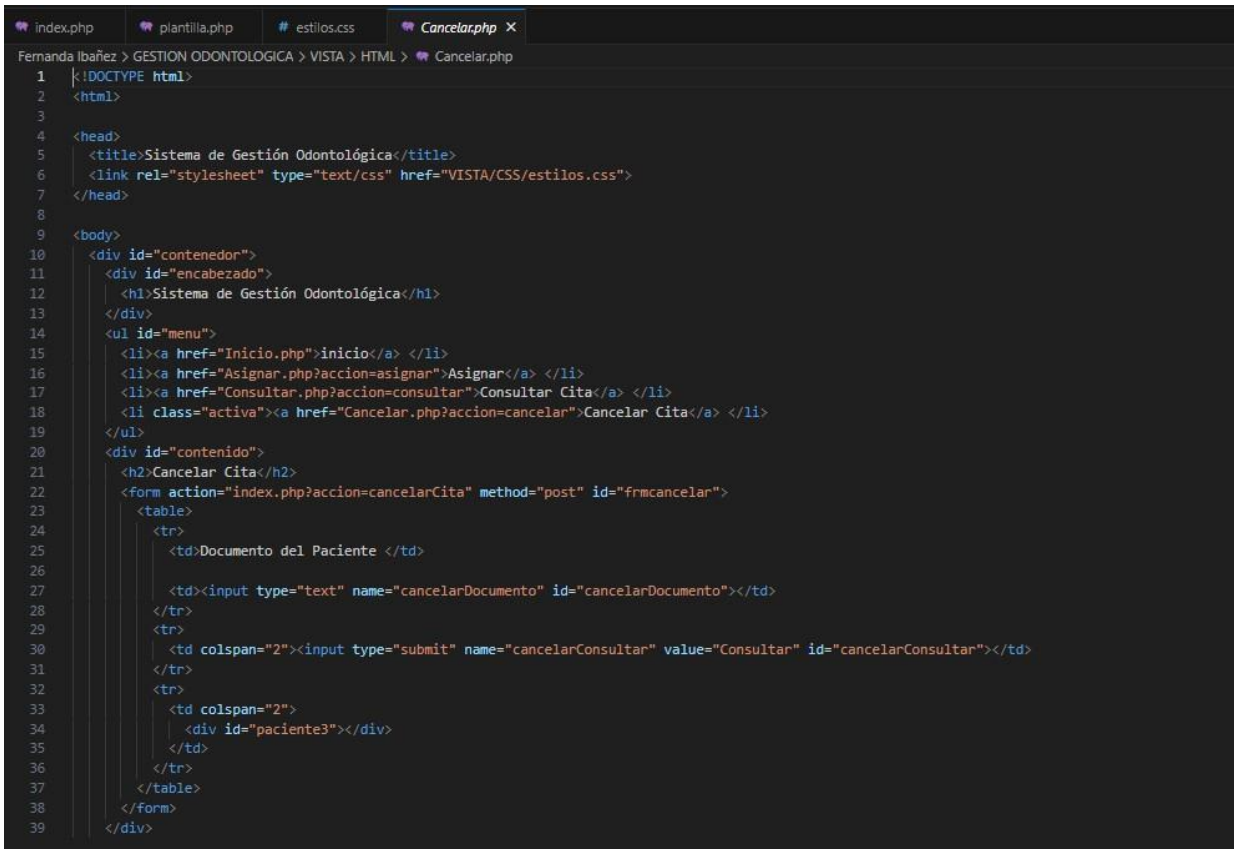
```
      <td>
        <input type="date" id="fecha" name="fecha" />
      </td>
    </tr>
    <tr>
      <td>Hora</td>
      <td>
        <select id="hora" name="hora">
          <option value="-1" selected="selected">
            ---Seleccione la hora ---
          </option>
          <option>08:00:00</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>
  </table>
</form>
</div>
</html>
```

Se implementa una etiqueta de formulario diseñada para recopilar información del paciente. Entre las funcionalidades incluidas, el

usuario puede elegir una opción de un conjunto predefinido, logrando esto mediante el atributo selected. Además, se utiliza la etiqueta ul para generar una lista no ordenada, proporcionando un menú alternativo al convencional. El código también incorpora el uso de clases, lo que facilita su referencia en el archivo CSS para aplicar estilos personalizados a los distintos elementos.

2. Archivo HTML - Cancelar



De manera similar al caso anterior, se implementa un formulario, aunque en esta ocasión se utiliza para verificar si los datos ingresados están registrados en la base de datos correspondiente. Cabe destacar que este archivo HTML,

al igual que los siguientes, sigue una estructura base que reutiliza ciertas etiquetas previamente descritas.

3. Archivo HTML – confirmar cita:

Confirmar_cita.php

GESTION ODONTOLOGICA > VISTA > HTML > Confirmar_cita.php

```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>
```

```
        <table>
          <tr>
            <td>Documento del Paciente</td>
            <td><input type="text" name="consultarDocumento" id="consultarDocumento"></td>
          </tr>
          <tr>
            <td colspan="2"><input type="submit" name="consultarConsultar" value="Consultar" id="consultarConsult
            </td>
          </tr>
          <tr>
            <td colspan="2">
              <div id="paciente2"></div>
            </td>
          </tr>
        </table>
      </form>
    </div>
```

```
                    <td>
                        <?php echo $fila->CitEstado; ?>
                    </td>
                </tr>
                <tr>
                    <td>Observaciones</td>
                    <td>
                        <?php echo $fila->CitObservaciones; ?>
                    </td>
                </tr>
            </table>
        </div>
    </div>
</body>

</html>
```

4. Archivo html – consultar

Consultar.php

GESTION ODONTOLOGICA > VISTA > HTML > Consultar.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 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="consultarDocumento" id="consultarDocumento"></td>
          </tr>
          <tr>
            <td colspan="2"><input type="submit" name="consultarConsultar" value="Consultar" id="consultarConsult
            </td>
          </tr>
          <tr>
            <td colspan="2">
              <div id="paciente2"></div>
            </td>
          </tr>
        </table>
      </form>
    </div>
```

5. Archivo html – Inicio

Inicio.php

GESTION ODONTOLOGICA > VISTA > HTML > 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>
```

En esta situación no se utiliza ningún formulario. En su lugar, se emplean etiquetas para estructurar el panel principal del sitio web, incluyendo títulos, listas sin orden para el menú de navegación y detalles generales sobre el contenido de la página.

Modelo

- **Archivo cita**

Cita.php

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;
    }


}
```

- Se diseña una clase destinada a gestionar la información, con atributos definidos como privados. Además, se implementa un constructor que simplifica los nombres de los atributos, optimizando el tiempo de desarrollo y previniendo redundancias. En el constructor se incluyen métodos públicos para devolver los valores almacenados.

Archivo conexión:

Conexion.php

MODELO > Conexion.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;
    }
}
```

El archivo de conexión corresponde a una clase diseñada para manejar la interacción con la base de datos mediante el atributo $mysqli. Adicionalmente, otros atributos, como $sql, se utilizan para guardar las consultas realizadas en la base de datos.

El método abrir() establece la conexión con la base de datos utilizando la clase mysqli, donde se deben proporcionar parámetros como el servidor (host), el nombre de usuario, la contraseña (si existe) y el nombre exacto de la base de datos registrado en el panel de administración. Dentro de esta función, se incluye una condición que verifica si ocurrió un error durante la conexión: en caso de fallo, devuelve un valor de 0, indicando el problema; de lo contrario, retorna 1, lo que confirma que la conexión fue exitosa. Una vez completada la operación, la conexión se finaliza mediante otro método.

$this->mySQLI->query() este método ejecuta la consulta en la base de datos, $this-

>filasAfectadas almacena el numero de filas afectadas en la consulta ejecutada con el método anterior.

Después se ejecutan 3 funciones finales:

- **obtenerResult():** retorna el resultado de la última consulta ejecutada , esta es útil para acceder a resultados que se ejecuten a operaciones como SELECT.
- **obtenerFilasAfectadas():** devuelve el numero de filas afectadas por la ultima consulta, este se utiliza para verficiar el impacto del operaciones como UPDATE, DELETE O **INSERT.**

- **obtenerCitaId():** retorna el ultimo ID insertado, este es útil para operaciones como

  INSERT.

  - ✓ **Archivo- gestor_cita**

```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;
    }
```

```php
    public function consultarCitaPorId($id)
    {
        $conexion = new Conexion();
        $conexion->abrir();
        $sql = "SELECT pacientes.* , medicos.*, 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;
    }
```

```php
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;
}
```

```php
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;
}
```

```php
public function agregarPaciente(Paciente $paciente)
{
    $conexion = new Conexion();
    $conexion->abrir();
    $identificacion = $paciente->obtenerIdentificacion();
    $nombres = $paciente->obtenerNombres();
    $apellidos = $paciente->obtenerApellidos();
    $fechaNacimiento = $paciente->obtenerFechaNacimiento();
    $sexo = $paciente->obtenerSexo();
    $sql = "INSERT INTO pacientes VALUES (
                '$identificacion','$nombres','$apellidos',"
        . "'$fechaNacimiento','$sexo')";
    $conexion->consulta($sql);
    $filasAfectadas = $conexion->obtenerFilasAfectadas();
    $conexion->cerrar();
    return $filasAfectadas;
}
```

```
80
81      public function consultarMedicos(){
82          $conexion = new Conexion();
83          $conexion->abrir();
84          $sql = "SELECT * FROM medicos ";
85          $conexion->consulta($sql);
86          $result = $conexion->obtenerResult();
87          $conexion->cerrar();
88          return $result ;
89          }
90  }
91
```

El código gestor cita está diseñado para gestionar operaciones relacionadas con citas médicas, pacientes y médicos, este interactúa con la base de datos que se utiliza desde la clase conexión.

- ✓ **Métodos de clase:**

  - ❿ **Agregar cita:** Tal como indica su propósito, agrega una nueva entrada en la tabla de citas utilizando un comando INSERT. Este proceso recopila información como la fecha, hora, paciente, médico, consultorio, estado y observaciones. Una vez que estos datos se registran, el identificador único de la cita se genera automáticamente.

  - ❿ **Consultar cita por Id:** Este método obtiene la información de una cita específica, combinando datos relacionados con el paciente, el médico y el

consultorio, a través de una consulta basada en el identificador único (ID).

- **Consulta Citas por documento:** Este proceso reúne las citas vinculadas al usuario identificado mediante su documento, siempre que su estado sea 'Solicitada'.

- **Consultar Paciente:** Esta operación busca en la tabla de pacientes un dato que corresponda con el documento ingresado por el usuario.

- **Agregar paciente:** Agrega un nuevo paciente a la tabla de pacientes, empleando métodos como 'obtener' para recopilar información. Devuelve datos del paciente, incluyendo identificación, nombres, apellidos, fecha de nacimiento y género, además de indicar la cantidad de filas afectadas por la operación

- **Consultar médicos:** Recupera los datos almacenados en la tabla de médicos mediante una consulta ejecutada con el comando SELECT sobre dicha tabla.

- **Archivo Paciente**

```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 incluye propiedades privadas destinadas a guardar datos del paciente, como su identificación, nombres, apellidos, fecha de nacimiento y

género. Implementa un constructor que define los métodos necesarios, empleando getters y setters para gestionar y acceder a la información de manera eficiente.

o CONTROLADOR

★ **Archivo controlador**

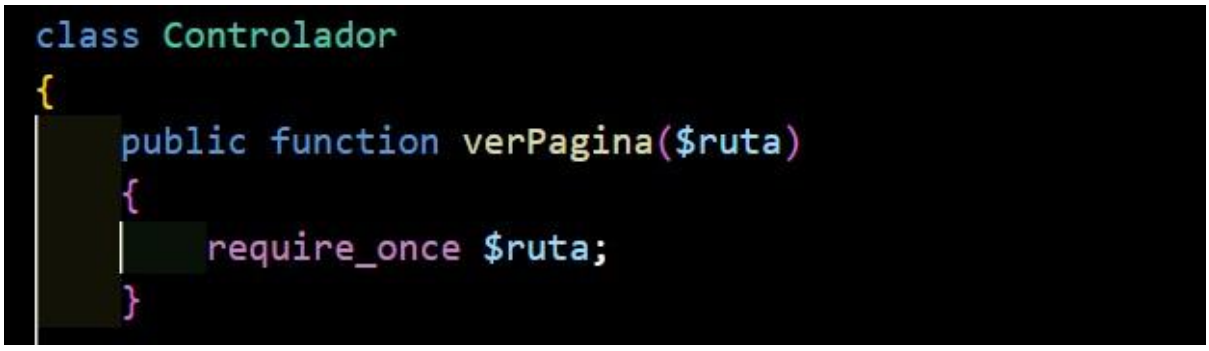
```php
    public function agregarPaciente($doc, $nom, $ape, $fec, $sex)
    {
        $paciente = new Paciente($doc, $nom, $ape, $fec, $sex);
        $gestorCita = new GestorCita();
        $registros = $gestorCita->agregarPaciente($paciente);
        if ($registros > 0) {
            echo "Se insertó el paciente con exito";
        } else {
            echo "Error al grabar el paciente";
        }
    }

    public function cargarAsignar()
    {
        $gestorCita = new GestorCita();
        $result = $gestorCita->consultarMedicos();
        $result2 = $gestorCita->consultarConsultorios();
        require_once 'VISTA/HTML/Asignar.php';
    }

    public function consultarHorasDisponibles($medico, $fecha)
    {
        $gestorCita = new GestorCita();
        $result = $gestorCita->consultarHorasDisponibles(
            $medico,
            $fecha
        );
        require_once 'Vista/html/consultarHoras.php';
    }

    public function consultarConsultorios()
    {
        $conexion = new Conexion();
        $conexion->abrir();
        $sql = "SELECT * FROM consultorios ";
        $conexion->consulta($sql);
        $result = $conexion->obtenerResult();
        $conexion->cerrar();
        return $result;
    }
```

```
89      public function verCita($cita)
90      {
91          $gestorCita = new GestorCita();
92          $result = $gestorCita->consultarCitaPorId($cita);
93          require_once 'Vista/html/confirmarCita.php';
94      }
95
96      public function confirmarCancelarCita($cita){
97          $gestorCita = new GestorCita();
98          $registros = $gestorCita->cancelarCita($cita);
99          if($registros > 0){
00          echo "La cita se ha cancelado con éxito";
01          } else {
02          echo "Hubo un error al cancelar la cita";
03          }
04          }
05  }
06
```

★ **Definición de la clase**

La clase Controlador centraliza la lógica necesaria para coordinar la interacción entre los diferentes componentes del sistema, como el modelo y las vistas. Su función principal es procesar las solicitudes provenientes de la interfaz web y gestionar acciones como registrar o buscar pacientes.

★ **Método ver pagina**



Recibe un parámetro que define una ruta e incorpora el flujo de ejecución mediante require_once. Su propósito es cargar las vistas o los archivos esenciales para renderizar una página web.

- ★ **Método agregar cita**

```php
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/Confirmar_cita.php';
}
```

- ★ Este método gestiona la generación de citas médicas, recibiendo información como el documento del paciente, el médico asignado, la fecha, la hora y el consultorio. Integra un objeto Cita que encapsula estos datos y utiliza el objeto GestorCita para registrar la cita en la base de datos.

- ★ **Método consultar cita**

```php
public function consultarCitas($doc)
{
    $gestorCita = new GestorCita();
    $result = $gestorCita->consultarCitasPorDocumento($doc);
    require_once 'VISTA/HTML/Consultar_citas.php';
}
```

```php
public function cancelarCitas($doc)
{
    $gestorCita = new GestorCita();
    $result = $gestorCita->consultarCitasPorDocumento($doc);
    require_once 'VISTA/HTML/Cancelar_citas.php';
}
```

- Este método realiza una búsqueda de las citas vinculadas a un paciente utilizando su número de documento. Los resultados obtenidos se presentan en la interfaz de consulta de citas.

- **Método cancelar cita**

Este método, además de buscar las citas utilizando el número de documento, también ofrece la funcionalidad de cancelar dichas citas..

- **Método consultar paciente**

```php
public function consultarPaciente($doc)
{
    $gestorCita = new GestorCita();
    $result = $gestorCita->consultarPaciente($doc);
    require_once 'VISTA/HTML/Consultar_paciente.php';
}
```

Este procedimiento permite localizar pacientes utilizando su documento de identificación. Los datos obtenidos desde la base de datos se presentan en la interfaz de consulta de pacientes.

- ★ **Método agregar paciente**

```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";
    }
}
```

- ★ Este procedimiento gestiona el registro de un paciente nuevo, recopilando su información y empleando el administrador de citas para incorporarlo a la base de datos. Cuenta con una condición que, al completarse correctamente, despliega un mensaje de confirmación; de lo contrario, informa sobre un fallo.

- ★ **Método Cargar asignar**

```php
public function cargarAsignar()
{
    $gestorCita = new GestorCita();
    $result = $gestorCita->consultarMedicos();
    $result2 = $gestorCita->consultarConsultorios();
    require_once 'VISTA/HTML/Asignar.php';
}
```

★ Este procedimiento crea una instancia de la clase encargada de gestionar citas. Utiliza el método de dicha clase para obtener una lista de médicos y también invoca otra función para recuperar la lista de consultorios disponibles. Su objetivo principal es recopilar y organizar la información necesaria para presentarla en la interfaz de asignación de citas.

★ **Método consultar horas disponibles**

```php
public function consultarHorasDisponibles($medico, $fecha)
{
    $gestorCita = new GestorCita();
    $result = $gestorCita->consultarHorasDisponibles(
        $medico,
        $fecha
    );
    require_once 'VISTA/HTML/Consultar_horas.php';
}
```

★ Este procedimiento acepta parámetros que determinan tanto la identidad de los médicos como las fechas concretas. Hace uso de la clase encargada de la gestión de citas, invocando un método que tiene como propósito obtener los horarios disponibles del médico. Los resultados se

presentan posteriormente en la interfaz destinada a la consulta de horarios.

★ **Consultar consultorios:**

```php
public function consultarConsultorios()
{
    $conexion = new Conexion();
    $conexion->abrir();
    $sql = "SELECT * FROM consultorios ";
    $conexion->consulta($sql);
    $result = $conexion->obtenerResult();
    $conexion->cerrar();
    return $result;
}
```

Este procedimiento establece una conexión con la base de datos y realiza una consulta que ejecuta la instrucción SELECT * FROM consultorios. Esta consulta recupera todos los registros almacenados en la tabla de consultorios.

★ **Método ver cita**

```php
public function verCita($cita){
    $gestorCita = new GestorCita();
    $result = $gestorCita->consultarCitaPorId($cita);
    require_once 'VISTA/HTML/Confirmar_cita.php';
    }
```

Este procedimiento acepta como parámetro el identificador de una cita. Utiliza la clase encargada de

gestionar citas para invocar el método destinado a buscar una cita por su ID, devolviendo los detalles específicos de dicha cita.

★ **Confirmar cancelar cita**

```
    public function confirmarCancelarCita($cita){
        $gestorCita = new GestorCita();
        $registros = $gestorCita->cancelarCita($cita);
        if($registros > 0){
        echo "La cita se ha cancelado con éxito";
        } else {
        echo "Hubo un error al cancelar la cita";
        }
        }
}
```

Este procedimiento toma como parámetro el identificador de una cita y emplea la clase encargada de la gestión de citas para invocar el método que realiza la cancelación de la misma, devolviendo la cantidad de registros modificados. Asimismo, verifica si la operación de cancelación se ejecutó correctamente

INDEX

Controlador.php index.php

index.php

```php
<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") {
            $controlador->cargarAsignar();
        }
    } elseif ($_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"],
```

Esto incluye las clases necesarias para permitir la

```php
            $_POST["consultorio"]
        );
    } elseif($_GET["accion"] == "consultarCita"){
        $controlador->consultarCitas($_GET["consultarDocumento"]);

    } elseif($_GET["accion"] == "cancelarCita"){
        $controlador->cancelarCitas($_GET["cancelarDocumento"]);

    } elseif ($_GET["accion"] == "ConsultarPaciente") {
        $controlador->consultarPaciente($_GET["documento"]);
    } elseif ($_GET["accion"] == "ingresarPaciente") {
        $controlador->agregarPaciente(
            $_GET["PacDocumento"],
            $_GET["PacNombres"],
            $_GET["PacApellidos"],
            $_GET["PacNacimiento"],
            $_GET["PacSexo"]
        );
    } elseif ($_GET["accion"] == "consultarHora") {
        $controlador->consultarHorasDisponibles($_GET["medico"], $_GET["fecha"]);
    } elseif($_GET["accion"] == "verCita"){
```

interacción del **controlador**. Con el sistema, el

```
46          $controlador->verCita($_GET["numero"]);
47      }elseif($_GET["accion"] == "confirmarCancelar"){
48              $controlador->confirmarCancelarCita($_GET["numero"]);
49
50      }else {
51          $controlador->verPagina('VISTA/HTML/Inicio.php');
52      }
53
54      ?>
55  </body>
```

controlador gestiona la lógica Principal. Incorpora componentes como el administrador de citas, las clases de cita y paciente, así como la conexión para operar con la información almacenada en la base de datos.

Este sigue una línea de control de acciones con if y else if , en donde revisa el valor acción, este ejecuta el método correspondiente del controlador.

- **Acciones disponibles:**
  - **Acción-asignar:** esta llama al método de cargar asignar del controlador, este prepara los datos para asignar citas (médicos y consultorios).
  - **Acción-consultar:** esta llama al método ver página () del controlador con la ruta vista, esta

carga una página para consultar la información general.

- **Acción-cancelar:** este es similar al anterior y carga una página específica para cancelar citas.

- **Acción-guardar cita:** esta llama al método agregar cita () del controlador, pasando los datos enviados por el formulario mediante el método post para guardar citas nuevas.

- **Acción-consultar cita:** esta llama al método consultar citas pasando el documento del paciente, esta muestra las citas asociadas al paciente.

- **Acción- cancelar cita:** esta llama al método cancelar citas, este se utiliza para cancelar una cita en específico usando el documento del paciente.

- **Acción- consultar paciente e ingresar paciente:** método que consulta la información detallada

sobre un paciente especifico, además hay un método adicional que es ingresar paciente pasando datos del paciente que este registrado en la base de datos.

- **Acción- consultar hora:** este método consulta las horas disponibles de un médico en fechas específicas.

- **Acción- ver cita:** esta muestra los detalles de la cita.

- **Acción-confirmar cita:** este se utiliza para cancelar una cita especifica identificada por el número de documento.