

Documentación del Proyecto Clínica Castiñeira

Resumen del Proyecto

La Clínica Castiñeira se basa en un sistema web diseñado para gestionar las citas, documentos y facturas de una clínica especializada en logopedia. El proyecto utiliza tecnologías web como HTML, CSS, jQuery, AJAX en el front-end, y PHP y MySQL en el back-end.

Estructura del Proyecto

El proyecto se organiza en las siguientes carpetas y archivos principales:

assets: Contiene archivos estáticos como imágenes y svg.

estilos: Almacena los archivos CSS utilizados en el proyecto.

scripts: Contiene scripts jQuery y archivos relacionados con AJAX.

back: Contiene archivos PHP reutilizables y comunes.

assets/uploads: Directorio para almacenar archivos subidos, como imágenes de usuario.

vistas: Directorio con las vistas del lado cliente.

Tecnologías Utilizadas

Front-end:

HTML: Estructura de las páginas web.

CSS: Estilo y presentación de las páginas web.

jQuery: Biblioteca de JavaScript para interactividad y manipulación del DOM.

AJAX: Comunicación asíncrona con el servidor.

Back-end:

PHP: Lenguaje de servidor para el desarrollo del lado del servidor.

MySQL: Sistema de gestión de bases de datos relacionales.

Base de Datos

Estructura de la Base de Datos

La base de datos clinica se compone de las siguientes tablas principales:

citas: Almacena información sobre las citas programadas.

citas_posibles: Contiene horas posibles para las citas.

documentos: Guarda documentos relacionados con pacientes y trabajadores.

documento_compartido: Asocia documentos compartidos entre usuarios.

facturas: Registra información sobre las facturas generadas.

user: Almacena información de los usuarios, tanto pacientes como trabajadores.

Procedimientos Almacenados

CrearNuevoDocumento: Procedimiento para crear un nuevo documento con información adicional y devuelve el id del documento.

Front-end

Páginas Principales

index.php: Página de inicio con enlaces a secciones principales.

datosUsuario.php: Página que muestra y permite editar la información del usuario.

facturas.php: Página que muestra las facturas generadas.

citas.php: Página que permite ver y gestionar las citas de un usuario.

crearUsuario.php: Página que permite crear un nuevo usuario sin permisos.

documentos.php: Página de gestión de documentos. Con los permisos oportunos de trabajador, podrán crear tanto nuevos documentos como facturas.

editor.php: Página para crear nuevos documentos.

facturas.php: Página para ver las facturas de un usuario. Si el usuario es administrador, podrá ver todas.

formularioCitas.php: Vista del formulario para generar una nueva cita.

gestorUsuarios.php: Página sólo para el usuario administrador. Podrá gestionar los usuarios, borrarlos y cambiar permisos.

header.php: vista del menú principal.

home.php: Página principal de la página. Es la que carga index.php.

login.php: Página para iniciar sesión.

nuevoUsuarioAdmin.php: Página solamente de administrador en la que podrá crear un usuario de un trabajador.

plantillaFactura.php: Página que sirve de plantilla para las facturas con inputs de formulario.

prevista.php: página de previsualización de los documentos.

previstaFactura.php: página de previsualización de las facturas.

Interactividad

La interactividad se logra mediante jQuery y AJAX para la carga asíncrona de datos y la mejora de la experiencia del usuario. Las razones principales fueron:

- La paginación de los datos que pudiesen ser más complicados de manejar todos juntos.

- Evitar la actualización de la página al guardar los documentos, ya que el usuario puede querer guardar cada cierto tiempo para no perder sus textos.

Los scripts están guardados en la carpeta /scripts y la lógica de la librería Rich-Text-Editor-jQuery-RichText en la carpeta correspondiente.

Back-end

Archivos Principales

conexion.php: Configuración de la conexión a la base de datos.

bd.php: Todas las operaciones de CRUD que se realizarán en la base de datos.

functional.php: funciones comunes.

actualizarUsuario.php: Lógica relacionada con las actualización de un usuario.

borrarCita.php: Lógica para borrar una cita.

borrarDocumento.php: Lógica para borrar un documento llamando a bd.

borrarUsuario.php: Lógica para borrar un usuario llamando a bd.

cerrarSesion.php: Destruye la sesión y la cookie.

compartirDocumento.php: Lógica para compartir un documento llamando a bd.

comprobarLogin.php: Lógica para comprobar las credenciales del usuario.

getCitas.php: lógica para obtener las citas según id del usuario o su rol.

getDatosCliente.php: lógica para obtener los datos del cliente por id.

getDocumentos.php: lógica para obtener los documentos dando el id de usuario y un índice de inicio.

getFacturas.php: lógica para obtener las facturas dando el id de usuario y un índice de inicio.

getPáginas.php: lógica para obtener el número de páginas en cada paginación de documentos.

getPáginasFacturas.php: lógica para obtener el número de páginas en cada paginación de facturas.

guardarFactura.php: Lógica para guardar una nueva factura en la bd.

guardarUsuario.php: Lógica para guardar una nueva usuario en la bd. Puede ser trabajador o paciente.

cerrarSesion.php: Crea la sesión y las variables de sesión y almacena una cookie de login.

Seguridad

Se implementan prácticas de seguridad como la validación de datos, el uso de consultas preparadas para prevenir inyecciones SQL.

Instalación y Configuración

1.- Clonar el Repositorio: `git clone https://github.com/Aitorif/PFC.git`

2.- Configurar la Base de Datos: Importar el archivo `clinica.sql` y configurar las credenciales en `conexion.php`.

Uso del Sistema

Acceder a la página de inicio.

Iniciar sesión.

Navegar a "Mis Datos" para ver y editar información personal. Esto podemos hacerlo clicando en la foto de usuario del header para abrir el desplegable

Visitar "Mis Facturas" para consultar las facturas generadas.

Visitar "Mis documentos" para ver los documentos compartidos o generar nuevos.

Visitar "Citas" para ver las citas o pedir nuevas.

Visitar "Usuarios" para gestionar estos y crear nuevos usuarios de trabajador..

Explorar otras secciones según sea necesario.

Futuras Mejoras

Implementar mejoras de seguridad en materia de sesión y contraseñas.

Mejorar la interfaz de usuario con estilos más modernos.

Agregar funcionalidades adicionales según los requisitos del usuario.

Hacerlo responsive para móvil.