



Universidad Autónoma
de San Luis Potosí

Universidad Autónoma de San Luis Potosí



FACULTAD DE
INGENIERÍA

Facultad de Ingeniería

Aplicaciones Web Interactivas

MANUAL DEL PROGRAMADOR

Alumno:

- *Jesús Antulio Jiménez Delgado*

Profesor:

- *Francisco Everardo Estrada Velázquez*

02 / 12 / 2022

Tabla de contenido

Instalación	2
Descripción de funciones	2
Migraciones.....	2
User_table	2
Documentos	3
Citas	3
Cuestionario.....	3
Respuestas_x_usuario.....	3
HomeController	4
Index()	4
Dashboard()	4
verPerfil().....	4
Documentos()	4
Respuestas().....	4
Ver()	4
Descargar()	4
Contacto()	4
enviarCorreo()	4
exportUser().....	5
exportCitas()	5
Citas	5
citas()	5
crearCita()	5
comprobante()	5
Cuestionario()	6
crearCuestionario()	6
terminarCita().....	6
cancelarCita()	6
Laravel Breeze.....	6

Instalación

Para poder ejecutar el sistema, es necesario descargar la herramienta de Laragon, de preferencia la versión instalable y no la portable.

Una vez descargado e instalado Laragon, hay que verificar que la versión de PHP sea la 8.1.10 o superior y la versión de Node.js número 18 o superior. En caso de que una de las dos versiones no sea la correcta, es necesario descargarlas de su pagina oficial y guardarlas en la carpeta correspondiente de laragon

- **PHP**
 - Descarga: www.php.net/downloads
 - Carpeta en Laragon: laragon\bin\php\
 - Descomprimir la carpeta en el espacio de laragon mencionado anteriormente.
- **Node.js**
 - Descarga: nodejs.org/es/
 - Carpeta en Laragon: laragon\bin\nodejs\
 - Descomprimir la carpeta en el espacio de laragon mencionado anteriormente.

También es necesario contar con un IDE o editor de códigos. Se puede utilizar PHPStorm o Visual Studio Code, según te acomodes con cada programa. Para la primera opción es necesario contar con una licencia, en caso de no querer pagar por el programa, se puede utilizar Vsual Studio Code sin problema.

Descripción de funciones

Migraciones

User_table

En este archivo, se establece la migración para poder crear la tabla de usuarios, los atributos de la tabla son

- **Id:** identificador único para cada usuario de tipo int
- **Nombre:** nombre o nombres del usuario de tipo string
- **Apellido_paterno:** primer apellido del usuario de tipo string
- **Apellido_materno:** segundo apellido del usuario de tipo string
- **Fecha_nacimiento:** día, mes y año de nacimiento del usuario de tipo date
- **Sexo:** genero del usuario de tipo string
- **CURP:** clave única curp del usuario de tipo string
- **Calle:** Domicilio donde reside el usuario de tipo string
- **Num_ext:** número del domicilio del usuario de tipo string
- **Num_int:** número interior del domicilio del usuario de tipo string, puede ser null

- **Colonia:** colonia del usuario de tipo string
- **Código_postal:** código postal del usuario de tipo string
- **Municipio:** municipio donde se encuentra el usuario, de tipo string
- **Estado:** estado donde se encuentra el usuario de tipo string
- **Email:** correo electrónico con el que el usuario podrá usar el sistema, de tipo string y único
- **Teléfono:** número de contacto del usuario de tipo string
- **Password:** contraseña para que el usuario pueda acceder al sistema
- **Tipo_usuario:** rol del sistema

Documentos

Tabla donde se almacenara la ruta de acceso para los documentos del usuario

- **Id:** identificador único de tipo int
- **Solicitud:** solicitud de visa
- **Foto:** foto del usuario
- **Comprobante_pago:** ticket o recibo del comprobante de pago del tramite de visa
- **Identificación:** identificación oficial con fotografía del usuario
- **P_economia:** prueba económica del usuario
- **P_edu_empleo:** prueba de educación o empleo (según sea el caso) del usuario
- **Id_usuario:** id de la cuenta del usuario proveniente de la tabla User.

Citas

Tabla que guardara la información de la cita agendada del usuario

- **Id:** identificador único de tipo int
- **Fecha:** fecha de la cita agendada
- **Hora:** hora de la cita agendada
- **Motivo:** asunto por el cual se agendo la cita
- **Tipo_visa:** tipo de visa que desea tramitar
- **Id_usuario** id de la cuenta del usuario proveniente de la tabla User.
- **Nombre_clte:** nombre del usuario
- **Status:** estado de la cita (pendiente/terminado)

Cuestionario

Tabla que guardara las respuestas del usuario, son 5 columnas de tipo string

Respuestas_x_usuario

Tabla pivote que ayudara a asociar las respuestas del usuario con la cuenta registrada en el sistema.

- **Id:** identificador único de tipo int

- **Id_cuestionario:** id del cuestionario proveniente de la tabla cuestionario
- **Id_usuario:** id de la cuenta del usuario proveniente de la tabla User.

HomeController

Controlador general del todo sistema, que incluye las funciones para cargar paginas e información extraída de la bd para mostrar en pantalla. También encargado de mandar correo, exportar .xlsx y .pdf

Index()

Regresa la vista “home”

Dashboard()

Obtiene la información de las citas en el sistema si el usuario es de tipo administrador y la regresa a la vista “dashboard”

verPerfil()

Función de tipo POST que recibe el id de una cuenta para buscarla en la base de datos y así poder mandar la información a la vista “perfilUsuario” para poder visualizarlo.

Documentos()

Función de tipo GET que recibe el id de un usuario para poder buscar los documentos en la base de datos, de la tabla documentos, así como la información de la cuenta del usuario propietario de los documentos para regresar la información a la vista “documentos”.

Respuestas()

Función de tipo GET que recibe el id del usuario a visualizar para buscar en la tabla respuesta_x_usuario y poder asociar al usuario con un cuestionario.

Ver()

Función de tipo POST que recibe el id, nombre y apellido paterno del usuario para poder formar la ruta de almacenamiento de los documentos y así poder crear un url temporal y regresarlo a la vista “documentos” para poder visualizarlos.

Descargar()

Función de tipo POST que recibe el id, nombre y apellido paterno del usuario para poder formar la ruta de almacenamiento de los documentos para poder descargarlos en formato pdf o jpg según sea el caso.

Contacto()

Regresa la vista “contacto”

enviarCorreo()

Función de tipo POST que recibe el correo, asunto y mensaje para poder generar un correo con los datos ingresados y se mande a la dirección ingresada con una

copia oculta al correo app.moviles.01.uaslp@gmail.com que es la cuenta que se usa para mandar los correos de contacto. Una vez enviado el mensaje se regresa a la vista “contacto” con un mensaje de confirmación

`exportUser()`

Función que se encarga de exportar en un archivo de tipo .xlsx los usuarios que hay dentro del sistema

`exportCitas()`

Función que se encarga de exportar en un archivo de tipo .xlsx las citas que hay dentro del sistema

Citas

Controlador encargado de manejar todo lo referente a el agendado y almacenamiento de las citas y documentos del usuario.

`citas()`

Función que se encarga de verificar si el usuario que esta autenticado tiene una cita ya en el sistema y manda la información para mostrarla en la vista “cita”. En caso de no tener cita agendada, se manda sin información a la vista “cita”.

`crearCita()`

Función de tipo POST que recibe la fecha, hora, motivo, tipo_visa, id_usuario, nombre_clte, y los archivos solicitados para poder generar una cita.

Como primer paso, se genera un objeto de tipo cita, el cual ayudara a almacenar la información de la cita que es

- Hora
- Fecha
- Motivo
- Tipo de visa
- Id del usuario
- Nombre del cliente

Una vez capturados los datos, se crea la ruta de almacenamiento compuesta por el id del usuario, nombre y apellido paterno. Cada archivo se guarda con un nombre compuesto, que es el nombre del archivo (solicitud, id, foto, etc.), id del usuario, nombre y apellido paterno. Una vez guardados los documentos se regresa a la vista “cuestionario”.

`comprobante()`

Función encargada de generar un archivo PDF basándose en una vista con los datos de la cita y del usuario para que se pueda descargar.

[Cuestionario\(\)](#)

Regresa a la vista cuestionario

[crearCuestionario\(\)](#)

Función de tipo POST que recibe las respuestas del cuestionario para almacenarlas en la base de datos. Una vez almacenados, se redirige a la vista “citas”.

[terminarCita\(\)](#)

Función de tipo POST que recibe el id de la cita para poder marcarla como terminada y regresar a la vista dashboard

[cancelarCita\(\)](#)

Función de tipo POST que recibe el id de la cita para poder marcarla como cancelada y regresar a la vista dashboard

[Laravel Breeze](#)

Esta biblioteca es la encargada de realizar el registro de los usuarios, para mayor información puede consultar la documentación de Laravel.