

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
ÁREA DE CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA EN SISTEMAS INTELIGENTES
PROYECTOS COMPUTACIONALES III

PROYECTO

PSICOFI

MANUAL DE PROGRAMADOR

- > BAUTISTA GÓMEZ JUAN PABLO
- > HERNÁNDEZ ALONSO JESÚS ALEJANDRO
- > RAMÍREZ MARTÍNEZ FERNANDO ANTONIO
- > RAMÍREZ PADRÓN ERICK ENRIQUE
- > VEGA MÁRQUEZ ANDREA ISABEL

ASESORA

SANDRA EDITH NAVA MUÑOZ

DOCENTE

DR. EDGAR CASTILLO BARRERA

30 DE NOVIEMBRE 2024

Índice

Introducción	4
Laravel	4
Web Service.....	5
Estructura del proyecto.....	5
Rutas.....	6
Emails	30
Angular.....	30
Estructura del Proyecto	30
Archivos y carpetas principales	31
Carpeta src.....	33
Archivos Principales en src.....	33
Carpeta app.....	34
Carpeta components	34
calendario	34
footer	35
header.....	36
loader.....	37
servicios.....	37
Carpeta model	38
Modelo: Alumno	38
Modelo: Nota-Cita.....	38
Carpeta modules.....	38
agendar-cita.....	39
cambio-contrasena	40
cita-urgente.....	40
gestion-agenda.....	40

gestion-psicologos.....	40
historial-alumnos.....	40
login	40
pagina-inicio	40
reporte-citas.....	41
reportes	41
Carpeta servicios	41
csrfToken.....	41
Carpeta assets.....	41

Introducción

Este documento ha sido diseñado para servir como una guía completa y accesible para desarrolladores, con el objetivo de proporcionar una referencia acerca del funcionamiento de los diversos módulos de la plataforma PSICOFI.

PSICOFI es una plataforma web que implementa un sistema integral de gestión de pacientes para el Departamento de Atención Psicológica de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí.

Laravel

Laravel dentro de la plataforma PSICOFI funciona como una API, recibiendo las solicitudes que solicitadas por la página web desde Angular y retornando la información solicitada de la BD.

Cabe mencionar que la protección CSRF está habilitada, por lo que todos los enlaces, a partir de ahora mencionadas como rutas, que tengan como método POST necesitan que se envíe el token generado por Laravel.

Este token es generado y almacenado en una cookie de sesión usando la librería "Sanctum" y que Angular almacena mientras este activa la sesión.

Para una mejor organización del proyecto se decidió agrupar las rutas según el usuario o módulo que las utiliza. A continuación, se explican las rutas de la API, así como los argumentos que recibe y la forma en la que se retornan los datos.

NOTA: Las rutas pueden cambiar según la dirección que tenga asignado el servidor en el que se alojará la API.

A partir de ahora se ignorará el enlace de la ruta y solo se referenciará la ruta de laravel.

EJ. <http://localhost/PSICOFI-Api/public/login>

Web Service

Con el objetivo de una integración fácil con los servicios de la universidad se hace uso de un web service que permite la autenticación de alumnos y administradores con las credenciales ya existentes y usadas en otros servicios de la universidad (Clave única / RPE).

Adicionalmente el servicio permite obtener los datos de los alumnos utilizando la clave única del alumno, mismos datos que solo pueden ser solicitados y observados por los psicólogos registrados por el administrador.

Estructura del proyecto

La estructura del proyecto de laravel se basa en el uso de controladores. Cada uno con un nombre significativo que permite asumir la función de cada uno. A continuación, se listan los controladores usados.

- **AlumnoController**

Este controlador se encarga de obtener la información referente a los alumnos, como el historial de citas de un alumno, llamas al servicio para obtener información del alumno, así como dar de alta la información necesaria de los alumnos en la BD, usada para la generación de gráficas estadísticas (Reportes).

- **AuthController**

Este controlador se encarga de realizar la autenticación de los tres tipos de usuarios (Psicólogo, Alumno, Administrador) de la aplicación usando el web service.

- **DateController**

Se encarga de manejar y obtener toda la información referente a las citas, es decir, la agenda de citas, confirmación y cancelación de estas, generación de citas por parte de los psicólogos, así como funciones que permiten el envío de correos electrónicos para la notificación tanto para alumnos como psicólogos.

- **NotaCitaController**

Este controlador se encarga de manejar el almacenamiento de las notas creadas cuando se atiende una cita por parte de los psicólogos, así como la obtención de las mismas cuando se consulta una cita pasada.

- **PsicoController**

Controlador destinado a manejar toda la información referente a los psicólogos, desde la obtención de su información, como el cambio de contraseña en el caso de psicólogos externos. Así como el alta de nuevo psicólogos en la BD por parte del administrador.

- **ReporteController**

Este controlador esta destinado a la obtención de la información necesaria para la generación de los reportes usando consultas SQL para realizar filtrados, conteos, etc.

Rutas

A continuación, se realiza un listado de las rutas (endpoints), en donde se menciona su función, el tipo de protocolo permitido, la información que recibe y las respuestas que se obtienen.

- **/login (POST)**

Permite verificar las credenciales de los alumnos y psicólogos, usando el web service. Para alumnos externos no se hace uso del web service, pero si de consultas en la BD para verificar la existencia del psicólogo y la confirmación de que su contraseña sea correcta.

- **Parámetros (Body)**

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única de un usuario o la CURP/RPE para administradores y/o psicólogos externos.
password	String	La contraseña del usuario

- **Respuesta (Usuario válido)**

Parámetro	Tipo de dato	Descripción
id	String / Int	La clave única de un usuario o la CURP/RPE para administradores y/o psicólogos externos.
nombre	String	El nombre completo del usuario
validación	String	Usuario válido
rol	String	Rol del Usuario

- **Respuesta (Usuario inválido)**

Parámetro	Tipo de dato	Descripción
validación	String	Usuario inválido

- **/loginAdmin (POST)**

Permite verificar las credenciales del administrador con el web service. Esta ruta consta de dos verificaciones, la primera por parte del web service y la segunda a través de una búsqueda del ID del administrador en la BD para corroborar que efectivamente se trata de un administrador.

- **Parámetros (Body)**

Parámetro	Tipo de dato	Descripción
password	String	La contraseña del usuario

- Respuesta (Usuario válido)

Parámetro	Tipo de dato	Descripción
id	String / Int	La clave única de un usuario o la CURP/RPE para administradores y/o psicólogos externos.
validación	String	Usuario válido
rol	String	Rol del usuario

- Respuesta (Usuario inválido)

Parámetro	Tipo de dato	Descripción
validación	String	Usuario inválido

- / alumno/getData (POST)

Permite obtener la clave única y nombre de un alumno.

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del alumno

- Respuesta (Alumno válido)

Parámetro	Tipo de dato	Descripción
clave única	String / Int	La clave única del alumno
nombres	String	El nombre o nombres del alumno
apellidoPaterno	String	El apellido paterno del alumno
apellidoMaterno	String	El apellido materno del alumno

- / alumno/getDataAlumno (POST)

Permite obtener los datos personales del alumno haciendo uso del web service.

Nota: La única información del alumno que se almacena en la BD es la clave única, semestre, carrera y área de estudio. Cuya información es necesaria para la generación de reportes de manera eficiente.

- **Parámetros (Body)**

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del alumno

- **Respuesta (Alumno válido)**

Parámetro	Tipo de dato	Descripción
clave única	String / Int	La clave única del alumno
nombres	String	El nombre o nombres del alumno
apellidoPaterno	String	El apellido paterno del alumno
apellidoMaterno	String	El apellido materno del alumno
área	String	El área al que pertenece la carrera del alumno
carrera	String	La carrera que cursa el alumno
semestre	Int	Semestre actual del alumno
asesor	String	Nombre del asesor del alumno
condicionAcademica	String	La condición académica del alumno
promedioGral	Float	Promedio general del alumno
creditosAprobados	Int	Total, de créditos aprobados por el alumno
creditosInscritos	Int	Total, de créditos inscritos por el alumno en el semestre actual
sexo	Char	El género del alumno
edad	Int	La edad del alumno

- **Respuesta (Alumno inválido)**

Parámetro	Tipo de dato	Descripción
Error	String	Alumno NO encontrado

- / alumno/getDate (POST)

Consulta si el alumno tiene una cita activa, si es así, la retorna.

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del alumno

- Respuesta (Cita encontrada)

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD
hora	String	La hora de la cita agendada
fecha	String	La fecha de la cita agendada
estado	String	El estado de la cita (Agendada o confirmada)
claveUnica	Int	La clave única del alumno que agendo la cita
nombres	String	El nombre o nombres del psicólogo
apellidoPaterno	String	El apellido paterno del psicólogo
apellidoMaterno	String	El apellido materno del psicólogo

- Respuesta (Cita no encontrada)

Parámetro	Tipo de dato	Descripción
Error	String	Cita NO encontrada

- / alumno/getRecord (POST)

Consulta el historial del alumno y devuelve las citas a la que ha asistido

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	Int	La clave única del alumno

- **Respuesta (Historial encontrado)**

La respuesta es un arreglo de citas, todas con el siguiente formato

Parámetro	Tipo de dato	Descripción
idCita	Int	Id de la cita
hora	String	La hora de la cita agendada
fecha	String	La fecha de la cita agendada
estado	String	El estado de la cita (Agendada o confirmada)
claveUnica	Int	La clave única del alumno que agendo la cita
nombres	String	El nombre o nombres del psicólogo
apellidoPaterno	String	El apellido paterno del psicólogo
apellidoMaterno	String	El apellido materno del psicólogo

- **Respuesta (Historial NO encontrado)**

Parámetro	Tipo de dato	Descripción
Error	String	Sin historial

- **/ psicologo/searchPsicologo (POST)**

Consulta la información personal del psicólogo

- **Parámetros (Body)**

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo

- Respuesta (Psicólogo encontrado)

Parámetro	Tipo de dato	Descripción
clave única / CURP	String / Int	La clave única del psicólogo si es interno, la CURP del psicólogo si es externo
nombres	String	El nombre o nombres del psicólogo
apellidoPaterno	String	El apellido paterno del psicólogo
apellidoMaterno	String	El apellido materno del psicólogo
semestre	Int	Semestre actual del psicólogo
correo	String	El correo electrónico del psicólogo
activo	String	La condición del psicólogo
carrera	String	La carrera del psicólogo

- Respuesta (Psicólogo NO encontrado)

Parámetro	Tipo de dato	Descripción
Error	String	Psicólogo NO encontrado

- / psicologo/ registerPsicologo (POST)

Permite registrar nuevos psicólogos

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Clave	Int	La clave única del psicólogo interno, si existe CURP es nulo
CURP	String	La CURP del psicólogo externo, si existe Clave es nulo

nombres	String	El nombre o nombres del psicólogo
apellidoPaterno	String	El apellido paterno del psicólogo
apellidoMaterno	String	El apellido materno del psicólogo
carrera	String	La carrera del psicólogo
semestre	Int	Semestre actual del psicólogo
activo	String	La condición del psicólogo
correo	String	El correo electrónico del psicólogo
contrasena	String	La contraseña del psicólogo

- Respuesta (Registro exitoso)

Parámetro	Tipo de dato	Descripción
Confirmación	Bool	True

- Respuesta (Registro incorrecto)

Parámetro	Tipo de dato	Descripción
Confirmación	Bool	False

- / psicologo/ getPsicologos (POST)

Permite registrar nuevos psicólogos

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
activo	Int	1 si la ruta debe retornar solo los psicólogos activos, 0 o nulo si debe retornar todos los psicólogos

- Respuesta (Psicólogos encontrados)

La respuesta es un arreglo de psicólogos, cada uno con la siguiente información

Parámetro	Tipo de dato	Descripción
identificador	String / Int	La clave única del psicólogo si es interno, la CURP del psicólogo si es externo
nombres	String	El nombre o nombres del psicólogo
apellidoPaterno	String	El apellido paterno del psicólogo
apellidoMaterno	String	El apellido materno del psicólogo
activo	String	La condición del psicólogo
created_at	String	Fecha de registro del psicólogo

- Respuesta (Psicólogos NO encontrados)

Parámetro	Tipo de dato	Descripción
Arreglo	Array	Arreglo sin datos

- / psicologo/ getPatients (POST)

Permite consultar los alumnos atendidos por el psicólogo

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo

- Respuesta (Pacientes encontrados)

La respuesta es un arreglo de claves únicas de alumnos que han sido atendidos por el psicólogo con el id de la consulta.

Parámetro	Tipo de dato	Descripción
claveUnica	Int	La clave única del alumno

- Respuesta (Pacientes NO encontrados)

Parámetro	Tipo de dato	Descripción
Error	String	Psicólogo sin alumnos atendidos

- / psicologo/ updatePsicologo (PUT)

Permite actualizar la información de un psicólogo

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Clave	Int	La clave única del psicólogo interno, si existe el otro atributo es nulo
CURP	String	La CURP del psicólogo externo, si existe el otro atributo es nulo
nombres	String	El nombre o nombres del psicólogo
apellidoPaterno	String	El apellido paterno del psicólogo
apellidoMaterno	String	El apellido materno del psicólogo
activo	String	La condición del psicólogo

- Respuesta (Actualización exitosa)

Parámetro	Tipo de dato	Descripción
Confirmación	Bool	True

- Respuesta (Actualización fallida)

Parámetro	Tipo de dato	Descripción
Confirmación	Bool	False

- / psicologo/ updatePsicologo (PUT)

Permite actualizar la contraseña de un psicólogo externo

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
CURP	String	La CURP del psicólogo externo
contrasena	String	La contraseña actual del psicólogo, se usa como método de seguridad
nuevaContrasena	String	La contraseña nueva del psicólogo

- Respuesta (Contraseña actualizada)

Parámetro	Tipo de dato	Descripción
Confirmación	Bool	True

- Respuesta (Contraseña NO actualizada)

Parámetro	Tipo de dato	Descripción
Confirmación	Bool	False

- / cita/getDates (GET)

Permite obtener las citas sin agendar de un psicólogo

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo

- Respuesta (Citas sin agendar disponibles)

La respuesta es un arreglo de citas, cada una con el siguiente formato

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD
clavePsicologo	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo al que corresponde esa cita
hora	String	La hora de la cita agendada
fecha	String	La fecha de la cita agendada
estado	String	El estado de la cita (Libre)

claveUnica	Int	La clave única del alumno que agendo la cita (NULL)
------------	-----	---

- Respuesta (Citas sin agendar no disponibles)

Parámetro	Tipo de dato	Descripción
Arreglo	Array	Arreglo vacío

- / cita/getAllDates (GET)

Permite obtener todas las citas de un psicólogo

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo

- Respuesta (Citas sin agendar disponibles)

La respuesta es un arreglo de citas, cada una con el siguiente formato

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD
clavePsicologo	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo al que corresponde esa cita
hora	String	La hora de la cita agendada
fecha	String	La fecha de la cita agendada
estado	String	El estado de la cita
claveUnica	Int	La clave única del alumno que agendo la cita

- Respuesta (Psicólogo sin citas)

Parámetro	Tipo de dato	Descripción
Arreglo	Array	Arreglo vacío

- / cita/createDates (POST)

Permite al psicólogo crear citas disponibles en una cierta fecha según las horas elegidas desde angular.

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno o CURP del psicólogo externo
fecha	String	La fecha en la que se van a crear las citas disponibles
hora	Array[String]	Arreglo que contiene las horas en las que el psicólogo desea agendar citas

- Respuesta (Citas creadas correctamente)

Parámetro	Tipo de dato	Descripción
Respuesta	String	Citas creadas correctamente

- Respuesta (Citas NO creadas)

Parámetro	Tipo de dato	Descripción
Error	String	Citas NO creadas

- / cita/cancelDate (POST)

Permite a un alumno o psicólogo cancelar una cita agendada

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno, alumno o CURP del psicólogo externo
idCita	Int	El identificador de la cita en la BD

- Respuesta (Cita cancelada)

Parámetro	Tipo de dato	Descripción
Respuesta	String	Cita cancelada correctamente

- Respuesta (Cita NO cancelada)

Parámetro	Tipo de dato	Descripción
Error	String	Cita NO cancelada

NOTA: Una cita cancelada se queda en estado de cancelado, por lo que no puede ser reagendada con un alumno diferente. Los alumnos que hacen una cancelación reciben una penalización de 7 días hasta poder volver a agendar una cita.

Tanto al psicólogo como al alumno se le notifica de la cancelación.

- **/ cita/confirmDate (POST)**

Permite a un alumno confirmar una cita agendada

- **Parámetros (Body)**

Parámetro	Tipo de dato	Descripción
Id	String / Int	La clave única del psicólogo interno, alumno o CURP del psicólogo externo
idCita	Int	El identificador de la cita en la BD

- **Respuesta (Cita confirmada)**

Parámetro	Tipo de dato	Descripción
Respuesta	String	Cita confirmada correctamente

- **Respuesta (Cita NO confirmada)**

Parámetro	Tipo de dato	Descripción
Error	String	Cita NO confirmada

NOTA: Se notifica tanto al psicólogo como al alumno que se realizó la confirmación de la cita. La cual solo puede ser realizada a partir de dos días antes de la fecha de la cita.

- **/ cita/deleteDate/{idCita} (DELETE)**

Permite eliminar una cita

- **Parámetros (URL)**

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD

- **Respuesta (Cita eliminada)**

Parámetro	Tipo de dato	Descripción
Message	String	Cita eliminada correctamente

- **Respuesta (Cita NO eliminada)**

Parámetro	Tipo de dato	Descripción
Error	String	Error al eliminar la cita

- **/ getCita/{idCita} (GET)**

Permite obtener una cita

- **Parámetros (URL)**

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD

- Respuesta (Citas existente)

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD
clavePsicologo	Int	La clave única del psicólogo interno
clavePsicologoExterno	String	CURP del psicólogo externo
hora	String	La hora de la cita agendada
fecha	String	La fecha de la cita agendada
estado	String	El estado de la cita
claveUnica	Int	La clave única del alumno que agendo la cita

- Respuesta (Cita NO existente)

Parámetro	Tipo de dato	Descripción
Error	String	Cita no encontrada

- /crear-cita (POST)

Permite crear un registro de cita en la BD.

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
clavePsicologo	Int	La clave única del psicólogo interno puede ser -1
clavePsicologoExterno	String	CURP del psicólogo externo, puede ser -1
hora	String	La hora de la cita
fecha	String	La fecha de la cita
estado	String	El estado de la cita
claveUnica	Int	La clave única del alumno que asistió a la cita

- Respuesta (Cita creada)

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD

- Respuesta (Cita NO creada)

Parámetro	Tipo de dato	Descripción
Error	String	Error al crear la cita

- /reporte-citas/{idCita} (GET)

Permite consultar la nota de una cita

- Parámetros (URL)

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD

- Respuesta (Reporte encontrado)

Parámetro	Tipo de dato	Descripción
idNotaCita	Int	El identificador de la nota en la BD
tipoIntervención	String	Tipo de intervención que se le dio al alumno (Académica, emocional, social)
notas	String	Notas del psicólogo
departamento	String	Departamento al que se canaliza al alumno, puede ser nulo

detalleCanalizacion	String	Motivos por los que se canaliza al alumno, puede ser nulo si departamento es nulo
idCita	Int	El identificador de la cita a la que pertenece la nota
foráneo	Bool	Valor booleano que indica si el alumno es foráneo

- Respuesta (Cita NO creada)

Parámetro	Tipo de dato	Descripción
Error	String	Error al crear la cita

- **/api/nota-cita/{id} (PUT)**

Permite actualizar los detalles de un reporte de cita

- Parámetros (URL)

Parámetro	Tipo de dato	Descripción
idCita	Int	El identificador de la cita en la BD

- Parámetros (Body)

Parámetro	Tipo de dato	Descripción
tipoIntervención	String	Tipo de intervención que se le dio al alumno (Académica, emocional, social)
notas	String	Notas del psicólogo
departamento	String	Departamento al que se canaliza al alumno, puede ser nulo
detalleCanalizacion	String	Motivos por los que se canaliza al alumno, puede ser nulo si departamento es nulo

foráneo	Bool	Valor booleano que indica si el alumno es foráneo
---------	------	---

- Respuesta (Reporte actualizado)

Parámetro	Tipo de dato	Descripción
Message	String	Nota de cita actualizada correctamente.

- Respuesta (Reporte NO actualizado)

Parámetro	Tipo de dato	Descripción
Error	String	Error al actualizar la nota de cita.

- /tipos-intervencion (GET)

Recupera todos los tipos de intervención por los que se puede atender a un alumno

- Parámetros

No requiere enviar parámetros

- Respuesta

La respuesta es un arreglo con los tipos de intervención

Parámetro	Tipo de dato	Descripción
idTipoIntervención	Int	Id del tipo de intervención en la BD
tipoIntervención	String	Tipo de intervención que se le dio al alumno (Académica, emocional, social)

- **/departamentos (GET)**

Recupera todos los departamentos a los que se puede canalizar un alumno

- **Parámetros**

No requiere enviar parámetros

- **Respuesta**

La respuesta es un arreglo con los departamentos

Parámetro	Tipo de dato	Descripción
idDepartamento	Int	Id del departamento en la BD
departamento	String	Nombre del departamento

- **/api/nota-cita (POST)**

Crea una nota de cita para la cita indicada

Parámetro	Tipo de dato	Descripción
tipoIntervención	String	Tipo de intervención que se le dio al alumno (Académica, emocional, social)
notas	String	Notas del psicólogo
departamento	String	Departamento al que se canaliza al alumno, puede ser nulo
detalleCanalizacion	String	Motivos por los que se canaliza al alumno, puede ser nulo si departamento es nulo
foráneo	Bool	Valor booleano que indica si el alumno es foráneo

- Respuesta (Nota creada)

Parámetro	Tipo de dato	Descripción
Message	String	Nota de cita creada correctamente

- Respuesta (Nota NO creada)

Parámetro	Tipo de dato	Descripción
Error	String	Nota de cita NO creada

- /reporte/getAreasCarreras (POST)

Obtiene un arreglo con las carreras o áreas que existen en la BD sin repetir.

Parámetro	Tipo de dato	Descripción
tipo	String	Determina si se extraen las carreras o áreas existentes en la BD (carrera o area).

- Respuesta (Consulta válida)

Parámetro	Tipo de dato	Descripción
Datos	Array	Arreglo con las carreras o áreas existentes en la BD sin repetir.

- Respuesta (Consulta inválida)

Parámetro	Tipo de dato	Descripción
Error	String	Consulta incorrecta.

- /reporte/getReporte (POST)

Obtiene la información necesaria para la generación de los reportes.

Parámetro	Tipo de dato	Descripción
Tipo	String	Determina el tipo de reporte (carrera, facultad o area).
Nombre	String	Nombre del área o carrera sobre el cuál se va a crear el reporte.
Fecha inicio	String	Representa la fecha a partir de la cual comienza el periodo que abarca el reporte. Formato YYYY-MM-DD
Fecha final	String	Representa la fecha que marca el final del periodo abarcado por el reporte. Formato YYYY-MM-DD

- Respuesta (Consulta válida)

Parámetro	Tipo de dato	Descripción
Consulta	Array	Arreglo con la información estadística necesaria para la generación del reporte.

- Respuesta (Consulta inválida)

Parámetro	Tipo de dato	Descripción
Error	String	Sin datos.

Emails

El envío de emails se hace utilizando el servidor de Gmail. El envío de estos correos esta programado para ser asíncrono, es decir, que el error en el envío de alguno de los correos no hace que la aplicación deje de funcionar, sin embargo, es necesario que se ejecute el siguiente comando:

php artisan queue:work

El cual permite que los correos se ejecuten en segundo plano. **La no ejecución de este comando si puede ocasionar que la aplicación falle.**

Dentro de la carpeta App de la raíz principal existe una subcarpeta llamada Mail. La cual contiene los diferentes tipos de correos que se envían por la aplicación. Cuyo diseño individual puede encontrarse en la siguiente ruta

public -> resources -> views -> email

Angular

Angular se utiliza en la plataforma PSICOFI como el framework frontend encargado de la interacción directa con el usuario. A través de esta capa, se gestionan las vistas y se realiza la comunicación con el backend mediante las rutas de la API descritas en la sección anterior. A continuación, se detalla la estructura del proyecto Angular y las principales características de este.

Estructura del Proyecto

El proyecto Angular está organizado dentro de la carpeta PSICOFI-Web de manera estructurada, siguiendo las prácticas estándar para facilitar su mantenimiento y desarrollo. En el directorio raíz se encuentran los archivos de configuración y dependencias necesarios para el funcionamiento del proyecto.

Archivos y carpetas principales

Carpetas

- **.angular**
Carpeta generada automáticamente por Angular para almacenar datos temporales del framework y del servidor de desarrollo. No es necesario modificarla directamente.
- **.venv**
Carpeta que contiene el entorno virtual, generalmente utilizado para propósitos relacionados con Python, posiblemente para scripts adicionales o configuraciones específicas del proyecto.
- **.vscode**
Contiene configuraciones específicas del editor Visual Studio Code, como reglas de linting, configuración de depuración, y extensiones recomendadas.
- **environments**
Almacena archivos de configuración para los diferentes entornos (desarrollo, producción, etc.). Los valores específicos del entorno, como URLs de APIs, se gestionan aquí.
- **node_modules**
Carpeta generada automáticamente por NPM que contiene todas las dependencias del proyecto. No se debe modificar manualmente, ya que se gestiona mediante el archivo package.json.
- **src**
Contiene el código fuente de la aplicación. Dentro de esta carpeta se encuentra la implementación de componentes, servicios, modelos y módulos de Angular. (Más detalles se incluyen en la sección).

Archivos

- **.editorconfig**
Archivo de configuración para mantener un estilo de codificación uniforme entre diferentes editores y desarrolladores.
- **.gitignore**
Define los archivos y carpetas que deben ser ignorados por Git, como `node_modules` y archivos temporales generados por Angular o el sistema operativo.
- **angular.json**
Archivo principal de configuración del proyecto Angular. Define parámetros como las rutas de entrada, los estilos globales, los scripts y las configuraciones para la compilación en diferentes entornos.
- **package.json**
Describe las dependencias del proyecto, los scripts de desarrollo y las configuraciones básicas de NPM. Contiene los comandos utilizados `ng serve`, `ng build`, etc.
- **package-lock.json**
Archivo generado automáticamente que asegura que las versiones específicas de las dependencias se mantengan consistentes entre instalaciones.
- **README.md**
Archivo de documentación inicial del proyecto. Contiene información general sobre la aplicación y cómo ejecutarla. Es útil para nuevos desarrolladores o colaboradores.

- **tsconfig.json**

Archivo de configuración de TypeScript. Define opciones como el nivel de compilación de JavaScript y las rutas de los archivos fuente.

- **tsconfig.app.json**

Archivo de configuración específico para la aplicación Angular. Extiende tsconfig.json y aplica configuraciones adicionales relevantes solo para la compilación de la aplicación.

- **tsconfig.spec.json**

Archivo de configuración para las pruebas unitarias de la aplicación. Define cómo se deben compilar y ejecutar los archivos de prueba.

Carpeta src

La carpeta src es el núcleo del proyecto Angular. Contiene todos los recursos necesarios para construir la aplicación, incluyendo el código fuente, estilos, configuraciones y archivos de inicialización. Aquí se desglosa en detalle su estructura y contenido.

Archivos Principales en src

- **index.html**

Es el archivo HTML principal donde se carga la aplicación Angular. Incluye la etiqueta <app-root> que Angular utiliza como punto de entrada para renderizar los componentes. Este archivo puede ser modificado para añadir metaetiquetas o scripts globales requeridos por la aplicación.

- **main.ts**

Es el punto de entrada de la aplicación. Este archivo inicializa el módulo principal (AppModule) y arranca la aplicación en el navegador.

- **styles.css**

Archivo de estilos globales para la aplicación. Aquí se definen estilos que se aplican a toda la interfaz, complementando los estilos específicos de los componentes.

Carpeta app

La carpeta app contiene el código principal de la aplicación. Su estructura modular permite que cada funcionalidad esté encapsulada y sea fácilmente gestionable.

Subcarpetas de app:

Carpeta components

La subcarpeta components dentro del directorio src/app es una de las partes fundamentales de la arquitectura del proyecto Angular. En esta carpeta se encuentran todos los componentes reutilizables que aportan funcionalidades específicas a la interfaz de usuario, así como el archivo shared.module.ts, encargado de gestionar componentes y módulos compartidos.

A continuación, se describen las subcarpetas principales:

calendario

Descripción:

El componente calendario es esencial para la funcionalidad de agendamiento y visualización de citas dentro de la plataforma PSICOFI. Este componente permite a los usuarios interactuar con un calendario visual para seleccionar fechas y horarios disponibles, así como para visualizar citas ya programadas.

Características:

- **Interactividad:**

Los usuarios pueden seleccionar días específicos y visualizar citas en un formato intuitivo.

- **Integración con la API:**

El componente utiliza servicios para obtener datos de las citas almacenadas en el backend y mostrarlas en el calendario.

Archivos:

- `calendario.component.ts`: Contiene la lógica y los eventos relacionados con la interacción del calendario.
- `calendario.component.html`: Define la estructura visual del calendario.
- `calendario.component.css`: Estilos específicos para el diseño del calendario.

footer

Descripción:

El componente footer define el pie de página que se muestra en toda la plataforma. Este componente se encarga de proporcionar información institucional, enlaces relevantes y otros elementos que deben estar accesibles en la parte inferior de cada página como el aviso de privacidad.

Archivos:

- `footer.component.ts`: Archivo vacío que podría contener la lógica para actualizar dinámicamente el contenido del pie de página si es necesario en un futuro.
- `footer.component.html`: Plantilla con el diseño del pie de página.

- `footer.component.css`: Estilos personalizados para el diseño visual del pie de página.

header

Descripción:

El componente header representa la cabecera de la plataforma. Es una parte central de la navegación, permitiendo a los usuarios acceder rápidamente a las secciones principales de la aplicación.

Características:

- **Menú de Navegación:**
Proporciona enlaces personalizados para cada tipo de usuario hacia las principales funcionalidades de la plataforma, como inicio, agenda de citas, etc.
- **Estado del Usuario:**
Muestra información relevante como la clave del usuario y permite acceder a opciones como cerrar sesión.

Archivos:

- `header.component.ts`: Implementa la lógica del menú de navegación y las acciones del usuario.
- `header.component.html`: Define la estructura visual del encabezado.
- `header.component.css`: Contiene los estilos específicos para garantizar un diseño profesional y consistente.

loader

Descripción:

El componente loader proporciona una animación de carga que se muestra mientras la aplicación está procesando solicitudes o cargando datos desde el backend.

Características:

- **Minimalista** y **Reutilizable:**
El diseño es simple y puede ser invocado en cualquier módulo que requiera un indicador de carga.

Archivos:

- loader.component.ts: Lógica del componente.
- loader.component.html: Plantilla de la animación.
- loader.component.css: Estilos para la animación de carga.

servicios

Descripción:

Este componente proporciona la lógica para listar y describir algunos de los servicios utilizados para hacer peticiones al backend en la plataforma.

Características:

- **Listados** y **Dinámicos:**
Integra datos obtenidos del backend para generar listas de servicios disponibles.

Archivos:

- citas.service.ts: Lógica para hacer peticiones de citas directas del calendario.

- `psicologo.service.ts`: Lógica para traer información del psicólogo seleccionado en el calendario.

Carpeta model

La carpeta `model`, ubicada dentro del directorio `src/app`, contiene los modelos de datos que representan las entidades centrales utilizadas en la plataforma PSICOFI. Los modelos actúan como interfaces o clases que definen la estructura y el comportamiento de los datos que se manejan en la aplicación, facilitando la validación, manipulación y transferencia de información entre el frontend y el backend.

En esta carpeta se encuentran los modelos `alumno` y `nota-cita`, los cuales desempeñan un papel fundamental en la representación de la información dentro de la plataforma. A continuación, se detalla la funcionalidad y estructura de cada uno:

Modelo: Alumno

Descripción:

El modelo `alumno` define la estructura de datos de los estudiantes registrados en la plataforma. Esta entidad es clave para el manejo de la información personal y académica de los alumnos que reciben atención psicológica.

Modelo: Nota-Cita

Descripción:

El modelo `nota-cita` se utiliza para capturar la información relacionada con las notas de una cita.

Carpeta modules

La carpeta `modules`, ubicada dentro del directorio `src/app`, organiza la funcionalidad de la plataforma PSICOFI en unidades independientes y reutilizables, conocidas como módulos. Cada módulo encapsula las

vistas, componentes, servicios y lógica necesarios para implementar una funcionalidad específica, promoviendo la modularidad y escalabilidad del sistema.

Estructura de un Módulo

Cada módulo sigue una estructura consistente que incluye los siguientes elementos:

- **Archivos principales:**
 - **HTML:** Define la vista principal del módulo.
 - **TS:** Implementa la lógica y funcionalidad principal del módulo.
 - **CSS:** Aplica los estilos específicos para el módulo.
- **Subcarpeta components:** Contiene los componentes internos del módulo, que encapsulan funcionalidades específicas de su interfaz o lógica.
- **Subcarpeta services:** Incluye los servicios utilizados por los componentes del módulo para realizar peticiones a la API y manejar datos.

Módulos:

[agendar-cita](#)

Permite a los usuarios programar citas psicológicas a través de una interfaz intuitiva, gestionando la disponibilidad de horarios y asegurando la correcta asignación de citas.

[cambio-contrasena](#)

Proporciona una funcionalidad segura y fácil de usar para que los psicólogos externos puedan actualizar sus credenciales de acceso cuando lo necesiten.

[cita-urgente](#)

Gestiona de manera eficiente la creación y asignación de citas urgentes, priorizando a los usuarios que requieren atención inmediata.

[gestion-agenda](#)

Ofrece herramientas avanzadas para la administración de agendas, permitiendo que tanto psicólogos internos como psicólogos externos mantengan un control detallado sobre sus horarios y compromisos.

[gestion-psicologos](#)

Facilita la administración de la información de los psicólogos registrados, incluyendo datos personales y estatus. Además, ofrece la posibilidad de registrar nuevos psicólogos al sistema.

[historial-alumnos](#)

Muestra una tabla de alumnos atendidos, información específica de un alumno y su historial de citas y notas asociadas a cada alumno.

[login](#)

Implementa el sistema de autenticación de usuarios, garantizando un acceso seguro y controlado a la plataforma.

[pagina-inicio](#)

Presenta una página inicial de la plataforma y se ofrece acceso rápido a los módulos más relevantes para cada tipo de usuario.

reporte-citas

Muestra un desglose de los datos guardados de una cita específica, ayudando en el seguimiento y documentación de los casos atendidos.

reportes

Permite la generación de gráficas estadísticas relacionadas con el uso del sistema, permitiendo al administrador evaluar el desempeño del Departamento de Atención Psicológica.

Carpeta servicios

Dentro de esta carpeta se encuentra una subcarpeta `csrfService` que contiene un archivo TypeScript diseñado para manejar las operaciones relacionadas con los tokens CSRF, asegurando la protección contra ataques de falsificación de solicitudes en el lado del cliente.

csrfToken

El servicio `csrfToken.ts` centraliza la obtención y manejo del token CSRF necesario para realizar solicitudes seguras a las rutas de la API de Laravel. Este token es requerido para todas las solicitudes POST, PUT, y DELETE que interactúan con el backend.

Carpeta assets

La carpeta `assets`, ubicada en el directorio `src`, contiene todos los recursos estáticos que son utilizados por la aplicación Angular.