**SISTEMA DE REGISTRO DE ASISTENCIA**

Bienvenido! Estoy encantado de guiarte a través de la documentación de nuestro sistema de registro de asistencia. El objetivo de esta guía es proporcionarte información detallada y útil sobre nuestro software.

Nuestro sistema utiliza la librería JavaScript de código abierto llamada Leaflet, la cual se integra con Google Maps para crear mapas interactivos. En resumen, esto nos permite realizar un seguimiento de la asistencia de los profesores de manera eficiente. Para lograr esto, el software utiliza la geolocalización del usuario, solicitando permisos para acceder a la latitud y longitud de su ubicación.

Cuando un usuario intenta marcar su asistencia, el sistema verifica si la ubicación actual se encuentra dentro de un radio de 50 metros del punto predefinido. Si es así, se registra la asistencia. En caso contrario, el sistema informa al usuario que no se encuentra dentro del rango especificado y, por lo tanto, la asistencia no se marca. El propósito fundamental de este sistema es registrar la asistencia de los profesores mediante la geolocalización proporcionada por Google Maps.

Esperamos que esta documentación te sea de gran ayuda para comprender las funciones y el alcance de este proyecto. ¡No dudes en contactarnos si tienes alguna pregunta o necesitas más información!

**CONTENIDO**

Pág.

Bienvenida\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

Índice \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2

Objetivo del Proyecto\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

Alternativas y Viabilidad\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

Organización y gestión del proyecto\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6

1. **OBJETIVO DEL PROYECTO**

Problema:

La necesidad de automatizar el proceso de registro de asistencia para hacerlo más rápido y eficiente, eliminando la dependencia de métodos manuales

Objetivo:

Crear un software que automatice el proceso de marcar asistencia y hacer el seguimiento de esta para lo cual usaremos la geolocalización de Google Maps para obtener la ubicación del usuario y verificar si está en el rango

Comenzaremos haciendo un análisis de los datos de usuario que necesitaremos. Para este sistema tendremos dos tipos de usuarios:

***Profesor:*** podrá usar el software para registrar su asistencia y ver el horario que le asigno el administrador.

***Administrador:*** podrá agregar, eliminar o editar ya sea los datos del profesor o el horario asignado a cada profesor.

Estos dos tipos de usuarios tendrán una relación de uno a muchos ya que el administrador podrá asignar un horario a uno o a muchos profesores, los datos que se usaran para ambos usuarios son: Nombre, Apellido, DNI, Teléfono, Email, Contraseña.

1. **ALTERNATIVAS Y VIALIDAD**

**API - Librería**

La primera alternativa que tuve fue usar la API de geolocalización de Google maps para obtener y mostrar la ubicación en la que se encuentra el usuario pero me di cuenta que es un poco imprecisa a la hora de obtener la longitud y latitud y haciendo una investigación por internet encontré una librería de JavaScript que hacia lo mismo que la API de Google maps y decidí ponerla a prueba… me di cuenta que era pon poco más precisa a la hora de mostrar y obtener la ubicación además de que tiene una sintaxis un poco mas entendible y fácil. Entonces decidí usar esta librería.

**Lenguaje de programación:**

El lenguaje que pensé usar de primera fue PHP (**Hypertext Preprocessor**) la cual es un lenguaje de programación del lado del servidor que es muy utilizado en el desarrollo de aplicación web, pero me di cuenta que podría usar un framework (**Laravel**), no estaría mal ya que te ayuda a mejorar la seguridad, además de que proporciona un conjunto de herramientas y funciones que facilitan el desarrollo de aplicaciones web.

Estos son algunos de los beneficios del framework **laravel** de PHP**:**

* Mejora la productividad
* Potente sistema de enrutamiento
* ORM Eloquent
* Migraciones
* Autenticación
* Middlewares

Debido a estos beneficios usare el framework de PHP Laravel

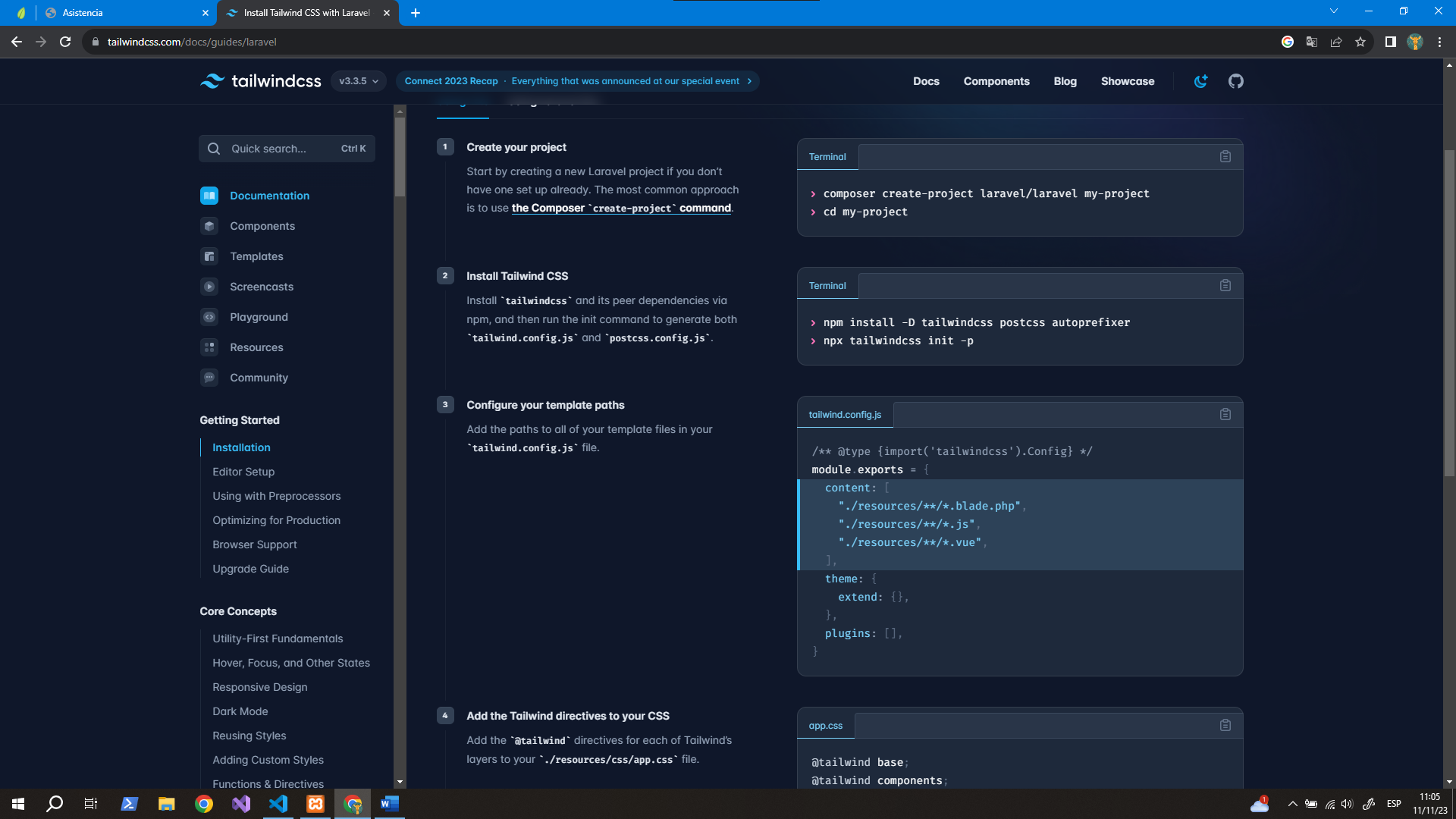
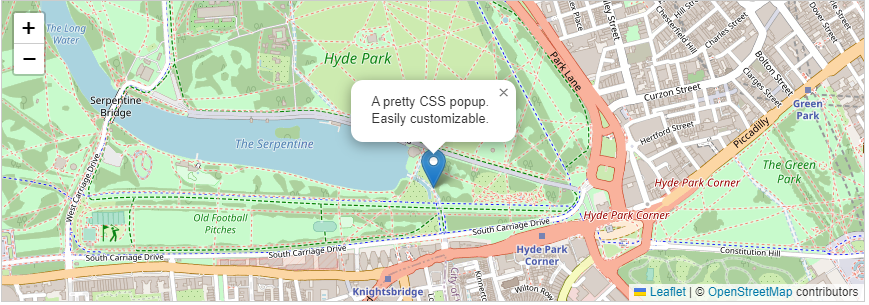
**Estilos css:**

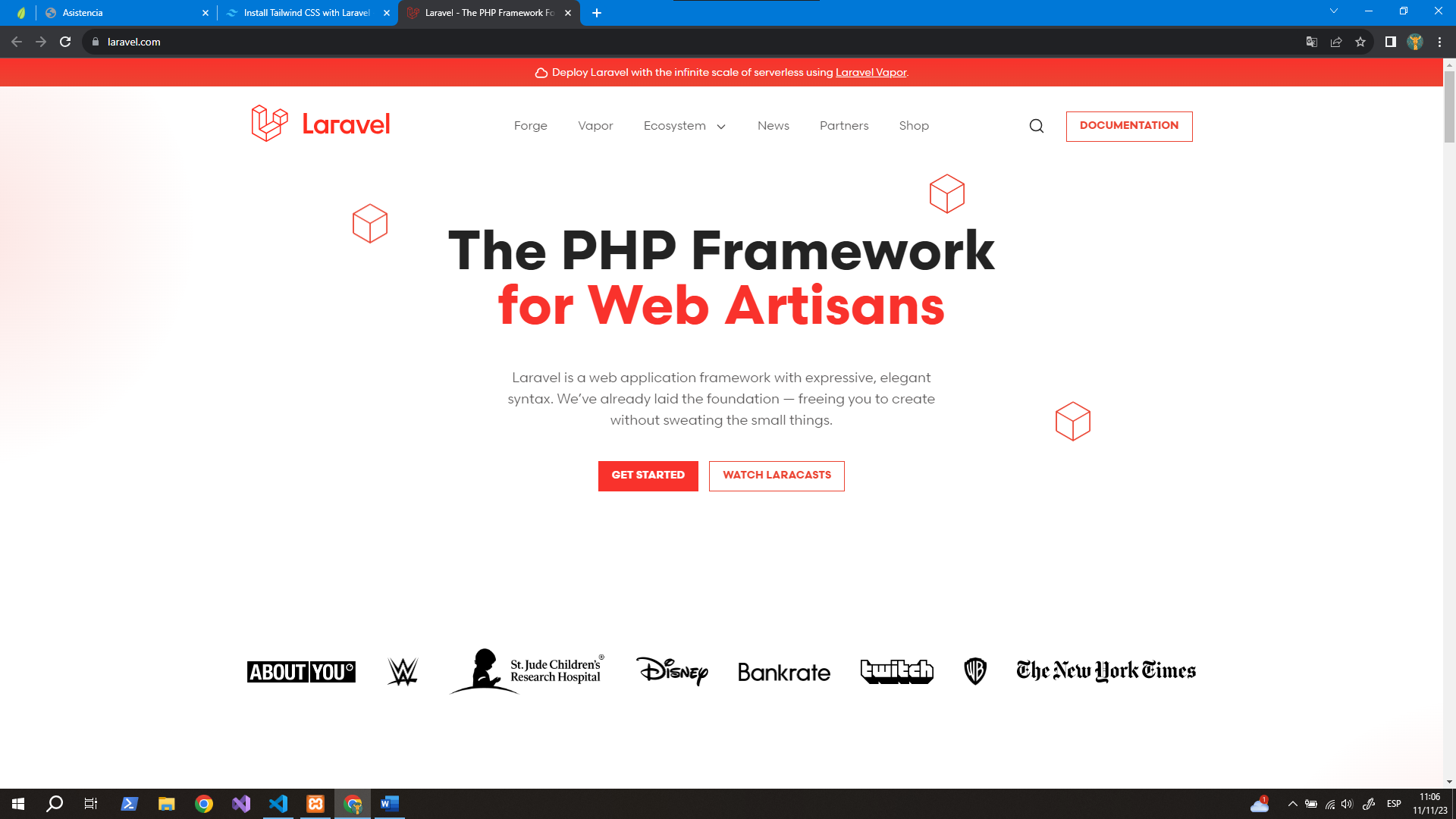
Para los estilos usare la librería de Tailwindcss ya que Tailwind implementa los estilos mediante clases directamente en el HTML agilizando el proceso de desarrollo, ya no es necesario navegar por hojas de estilo separadas para realizar ajustes ya que las clases están directamente relacionadas con las propiedades de estilo específicas.

Leaflet: <https://leafletjs.com/index.html>

Tailwindcss: <https://tailwindcss.com/>

Laravel: <https://laravel.com/>

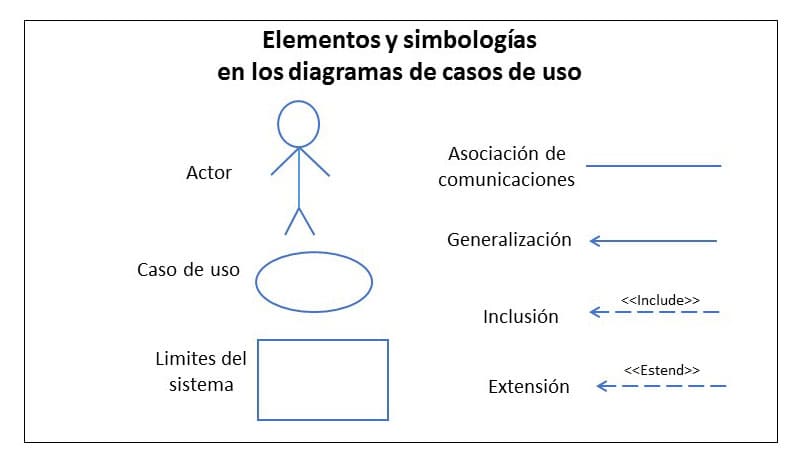




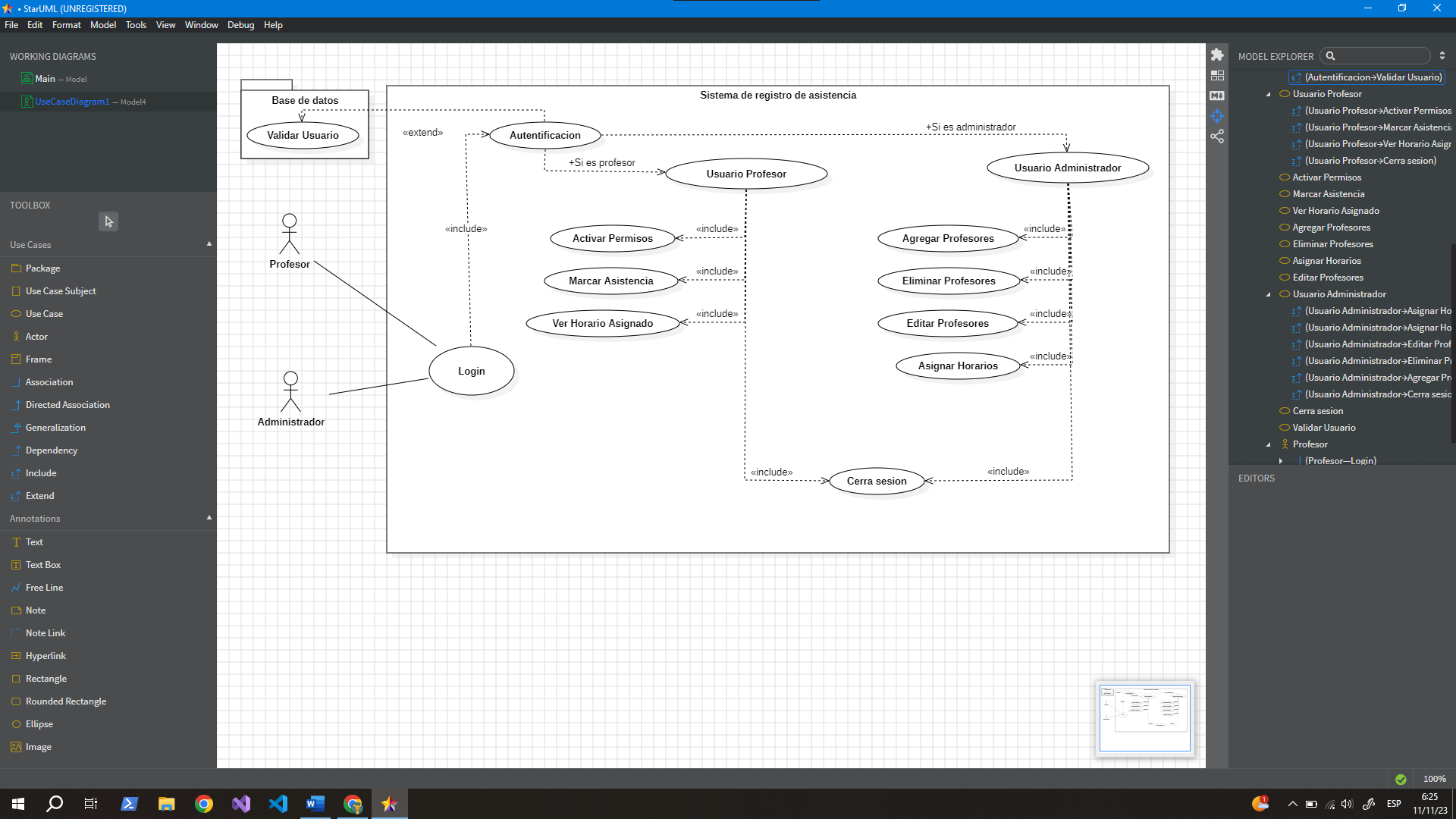
1. **ORGANIZACIÓN Y GESTION DEL PROYECTO**

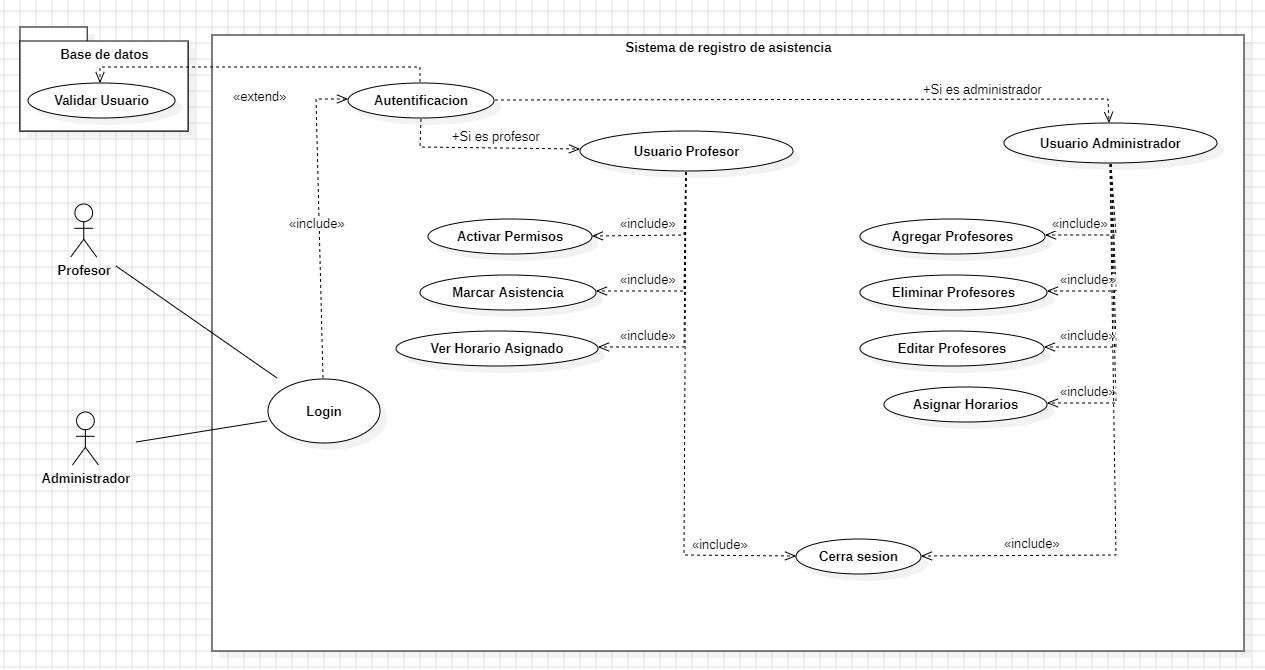
* **Pasos:**

Ahora que sé qué herramientas usaré comenzare diseñando las funciones que tendrá mi software. Para lo cual utilizaré UML (Lenguaje Unificado de Modelado), que es un conjunto de diagramas que me ayudaran a representar visualmente como interactúan las partes de mi software y funciones cumplirán. El tipo de diagrama que usare será Diagrama de casos de uso la cual describe como los usuarios interactúan con un sistema y las funcionalidades que ofrece.



* + CREACION DEL DIAGRAM DE CASOS DE USO





En este diagrama de casos de uso tenemos los dos tipos de usuario quienes podrán ingresar al Login Donde tendrán que ingresar su correo y contraseña y al dar clic en el botón de ingresar lo que hará el programa es hacer una consulta en la base de datos para verificar si las credenciales ingresadas existan, si no existe ningún usuario con esas credenciales lo que hará el programa es negarle el acceso de caso contrario lo dejara ingresar al siguiente nivel que es verificar si es un usuario administrador o un usuario profesor, dependiendo de esta verificación se les mostraran las funciones o opciones que tiene cada usuario.

**Funciones Usuario Profesor:**

* Activar Permisos: Al ingresar el sistema automáticamente solicitará permisos de ubicación la cual el usuario deberá activar.
* Marcar Asistencia: Una vez activada los permisos el usuario podrá perseguir con el registro de su asistencia dando clic en botón “marcar asistencia” y este botón tendrá la función de verificar si la ubicación en la que se encuentra el usuario esta fuera o dentro del rango requerido para poder marcar su asistencia.
* Ver Horario asignado: En esta parte solo se mostrará el horario asignado por el administrador a cada profesor.

**Funciones Usuario Administrador:**

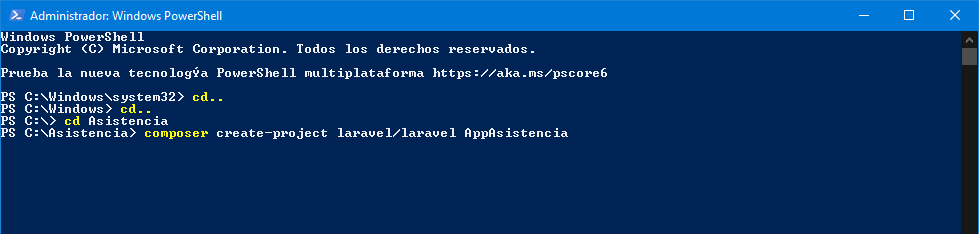
* Agregar Profesores: En esta parte el usuario administrador podrá agregar un profesor.
* Eliminar Profesores: el administrador también podrá eliminar un profesor.
* Editar Profesores: el administrador podrá editar los datos del profesor que desea solo usando el id que está en la base de datos
* Asignar Horarios: Una vez agregado un profesor el administrador podrá asignarle un horario a ese profesor.

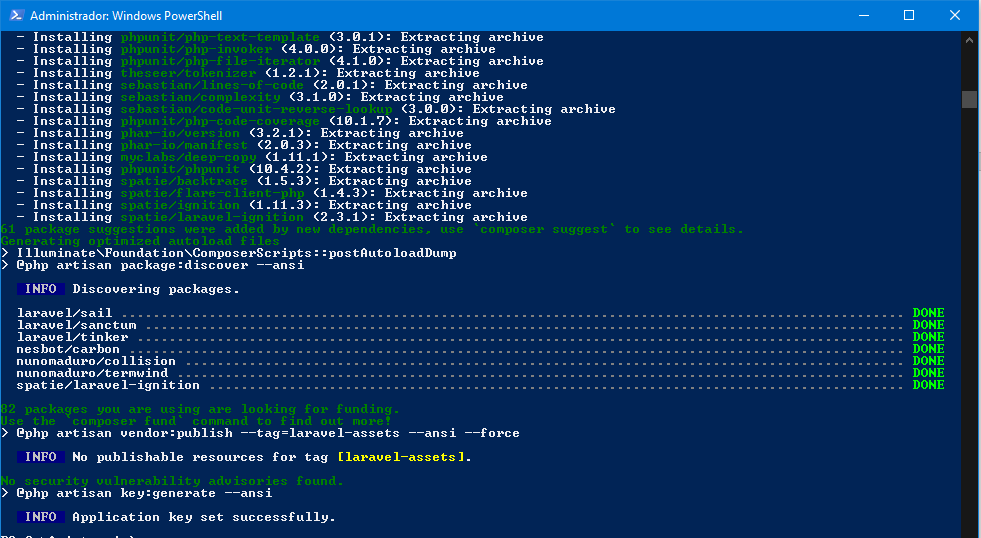
**Funciones para los dos usuarios:**

* Cerra sesión: Esta función estará activa para ambos usuarios ya que ambos tendrán la opción de cerrar su cuenta.

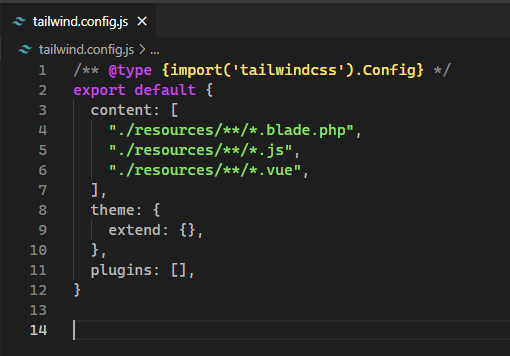
Comenzaremos con la creación del proyecto con el framework Laravel, para lo cual primero debemos tener la última versión de PHP instalada, también Composer y NodeJs. Ya que estas herramientas nos ayudaran a crear y ejecutar el proyecto:

* Abrimos el terminal que deseamos y navegamos hasta la carpeta donde queremos que cree nuestro proyecto y agregamos el comando para crear el proyecto laravel: composer créate-project laravel/laravel NombreDeProyecto y damos enter para que comience a crear nuestro proyecto:





* Una vez terminada la creación ya podemos abrirla con editor que deseas yo en este caso usare Visual Estudio Code, Comenzaremos instalando las dependencias de Tailwindcss para los estilos:
  + Instalamos las dependencias de Tailwind usando el comando **npm install -D tailwindcss postcss autoprefixer**
  + Luego inicializamos las dependencias instaladas usamos el comando **npx tailwindcss init -p** esto creara un archivo llamado tailwind.config.js.
  + En el archivo que se creó (tailwind.config.js) editamos y le decimos que se encerte en todo archivo que tenga .blade.php

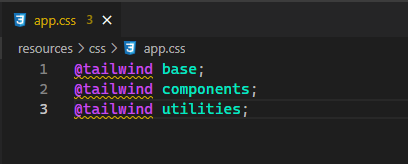


* + Luego editamos el archivo app.css que está en resources/css/app.css agregamos:

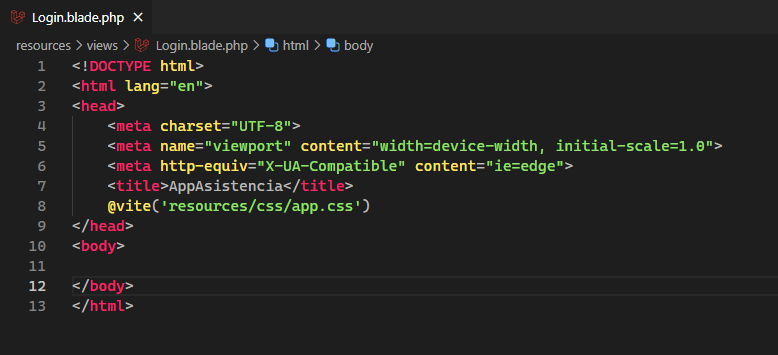
**@tailwind base;**

**@tailwind components;**

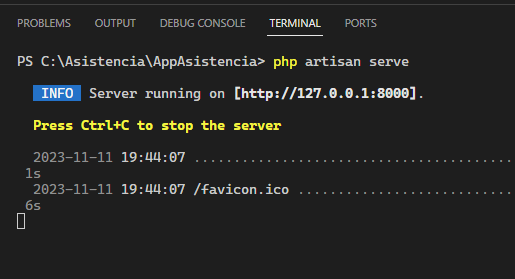
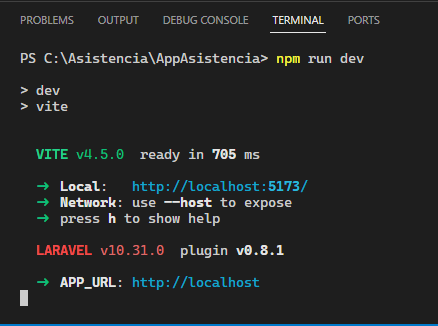
**@tailwind utilities;**



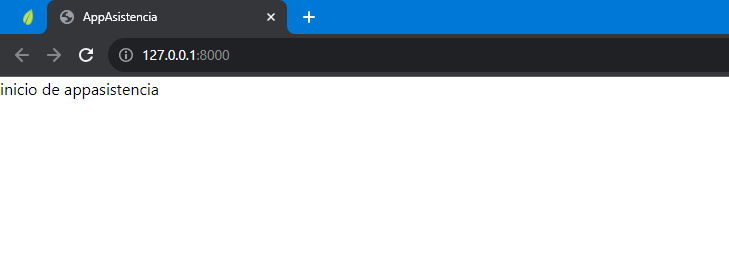
* + Creamos un archivo Blade Login.blade.php y agregamos **@vite('resources/css/app.css')** para cargar los estilos de la siguiente manera:

****

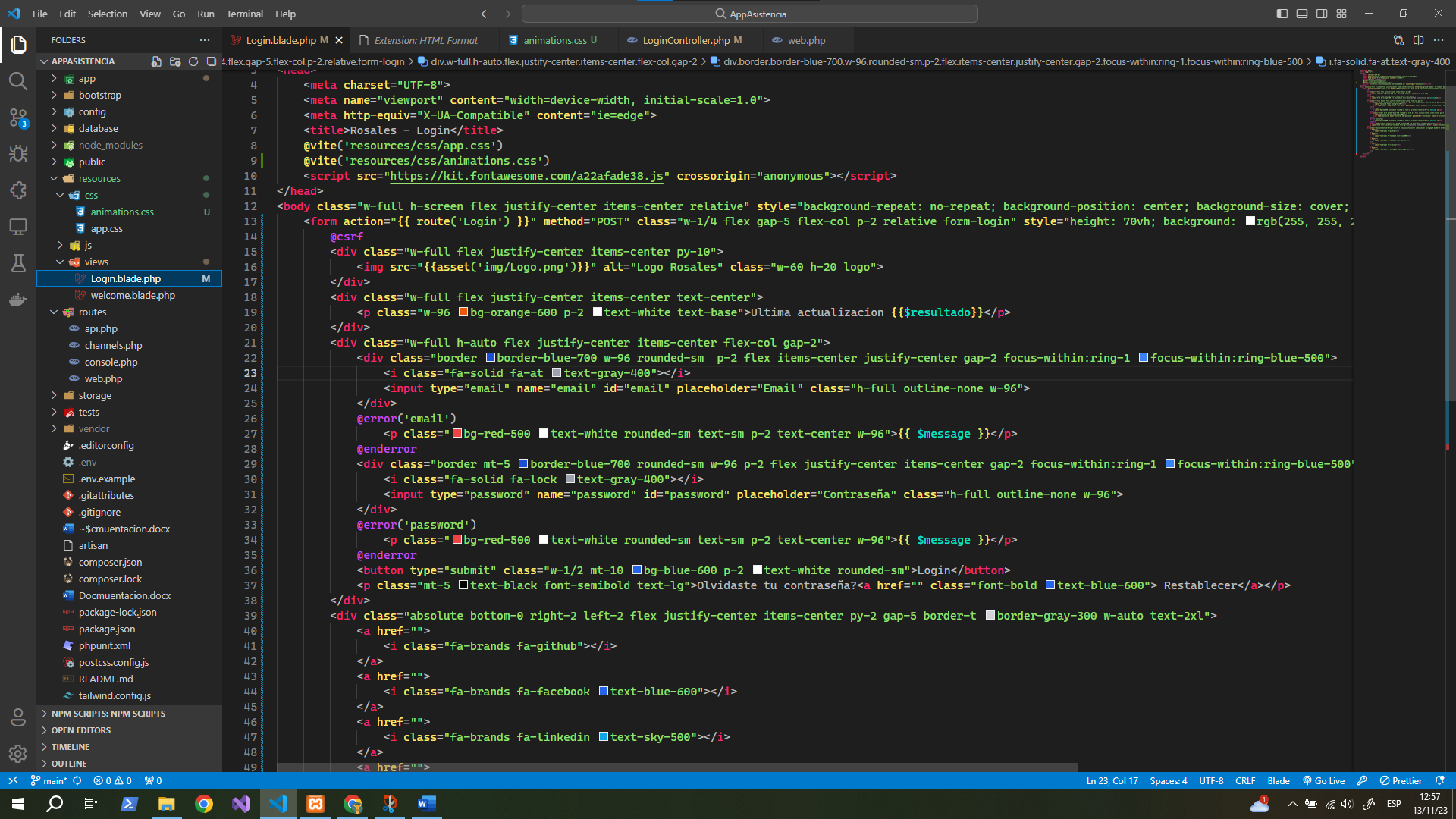
* + Seguido a esto ejecutamos los comandos **php artigan serve** para crear un puerto donde correrá nuestro programa y **npm run dev** para que se carguen los estilos que ponemos:

****

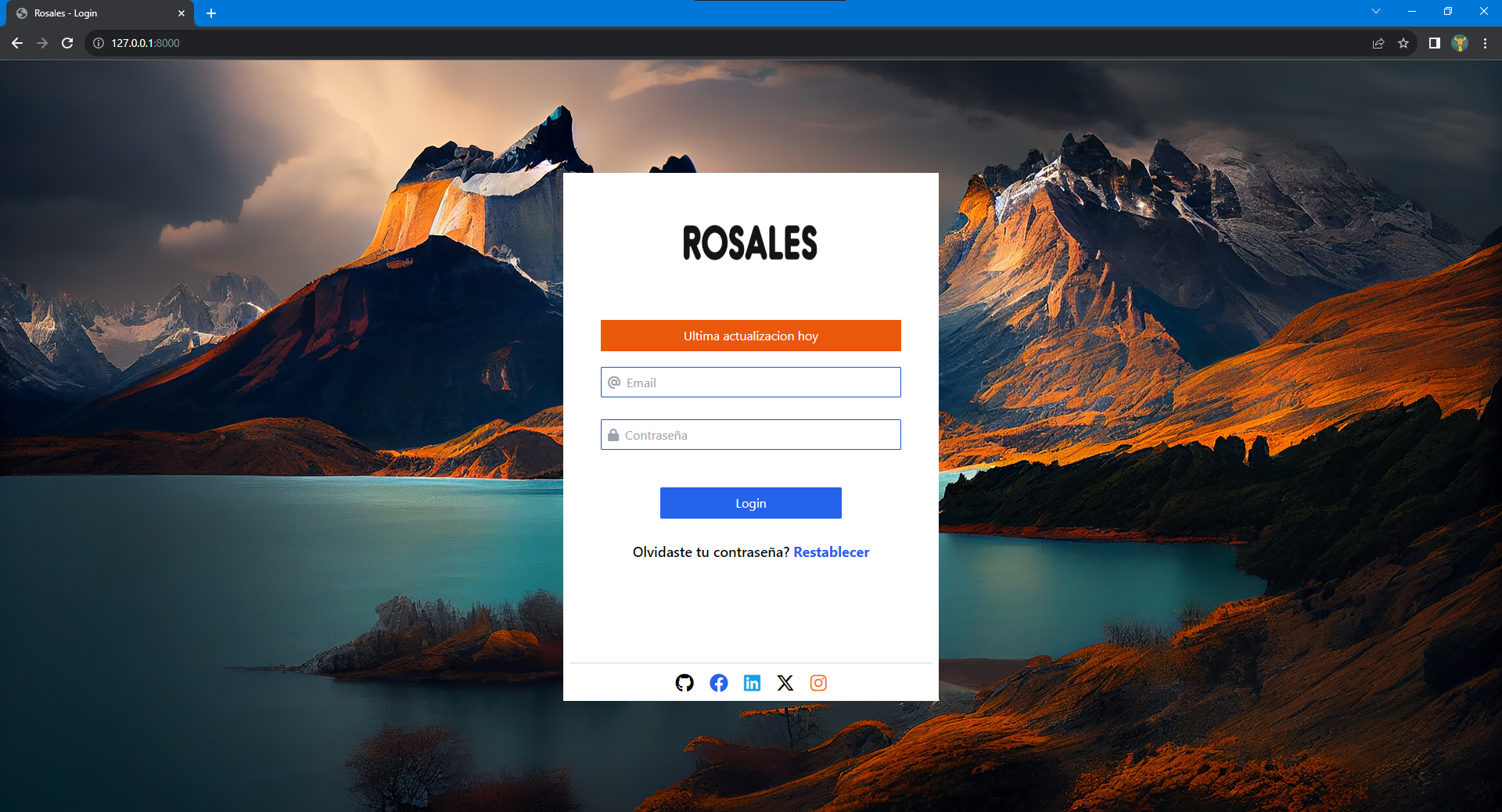
**Resultado:**

****

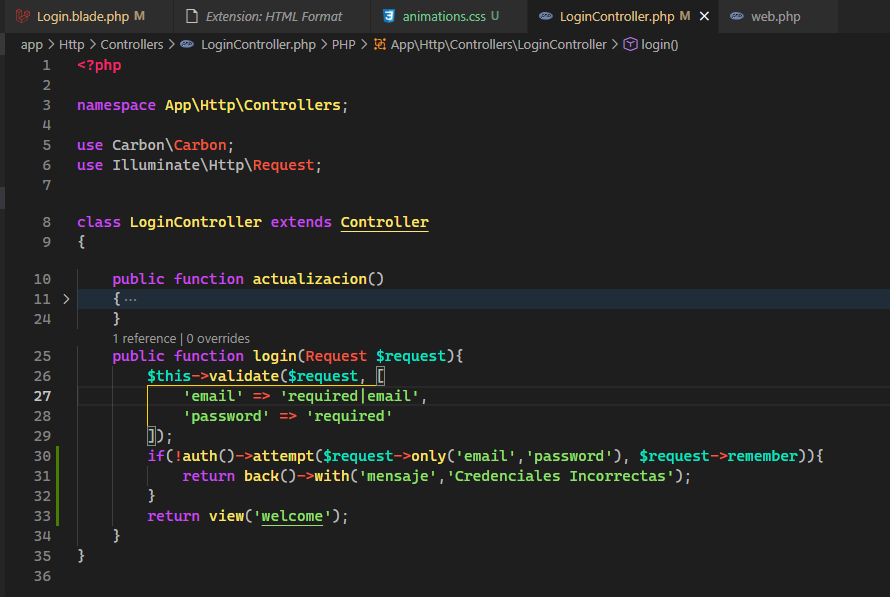
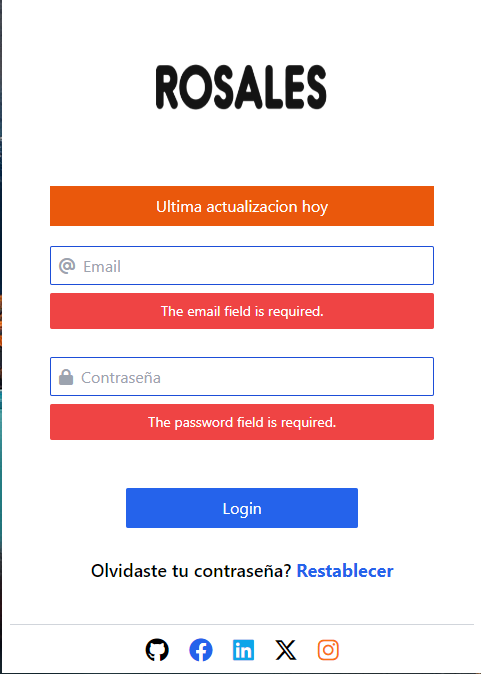
* + Ya que tengo mi proyecto corriendo procedí a crear la vista LOGIN y también añadí estilos
    - Código

****

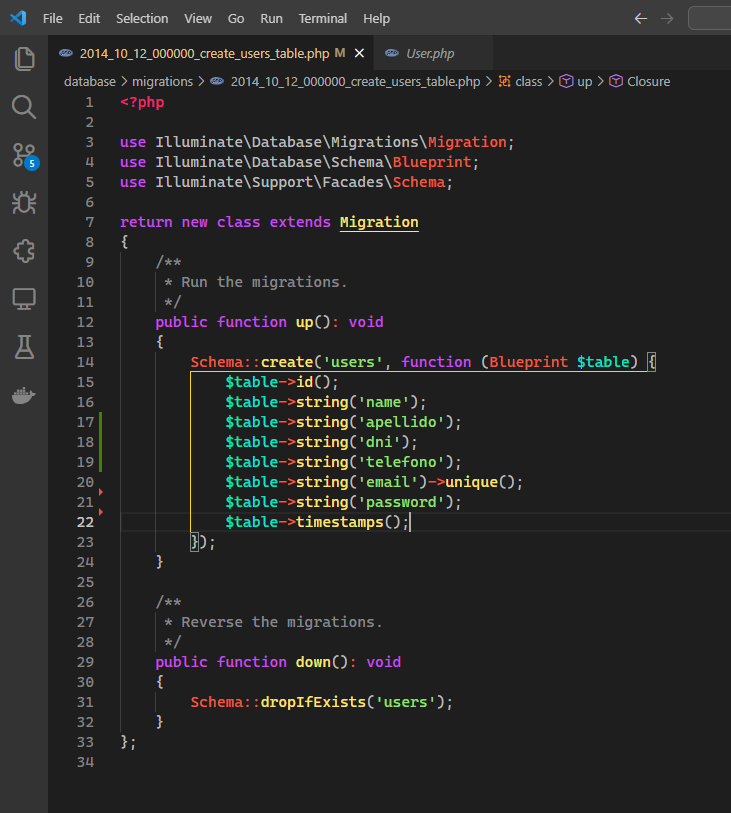
* + - **Resultado**

****

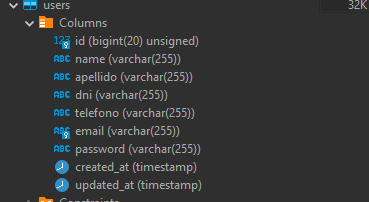
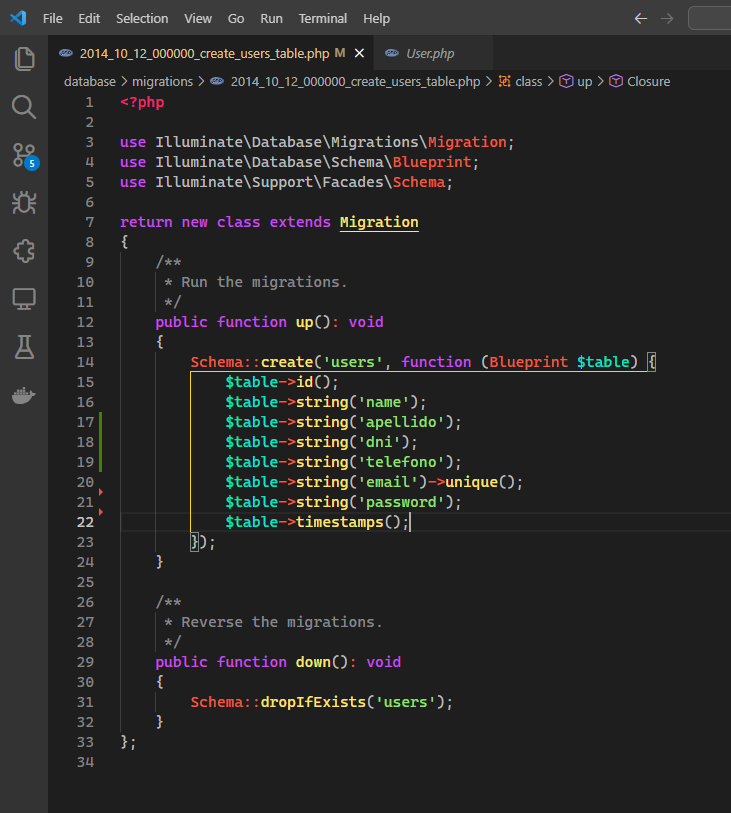
* + Ahora creare un controlador para verificar si los campos están vacíos y también verificar si esas credenciales existen en la base de datos
    - En este controlador, utilicé la clase **Request** para recuperar la información enviada a la solicitud **HTTP**. La clase Request nos permite acceder a datos de formularios y encabezados **HTTP**. Entonces el controlador primero verifica si los campos están vacíos o tienen alguna información, si todo esta correcto redirecciona al usuario a la vista **Home** donde se mostrarán las opciones.

****

* + Bueno ya que tengo creada mi LOGIN hare una migración para verificar que no haya ningún error en el controlador para eso usare el comando: **php artisan migrate**
    - Primero añadi los campos que usare:

****

* + - De la misma manera también añadí los campos en el modelo:

****