**Nombre del QA aspirante:** Verónica Garcés

**Fecha:** jueves 18 de enero del 2024

**Requisitos de la evaluación:**

Git

Docker con Docker compose

**Tecnologías utilizadas**

Selenium (para pruebas automatizadas)

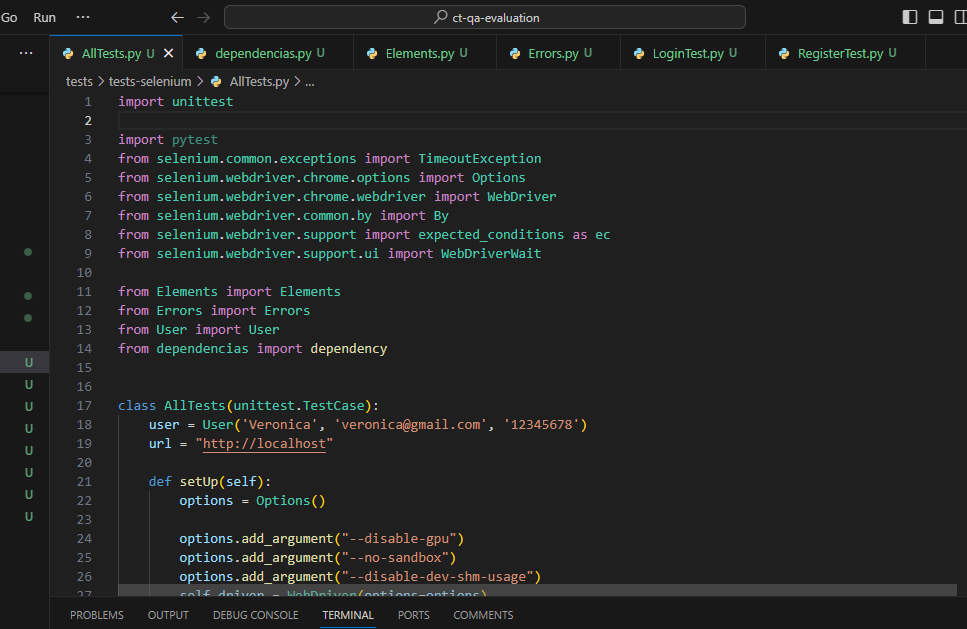
Visual

**Pasos para seguir:**

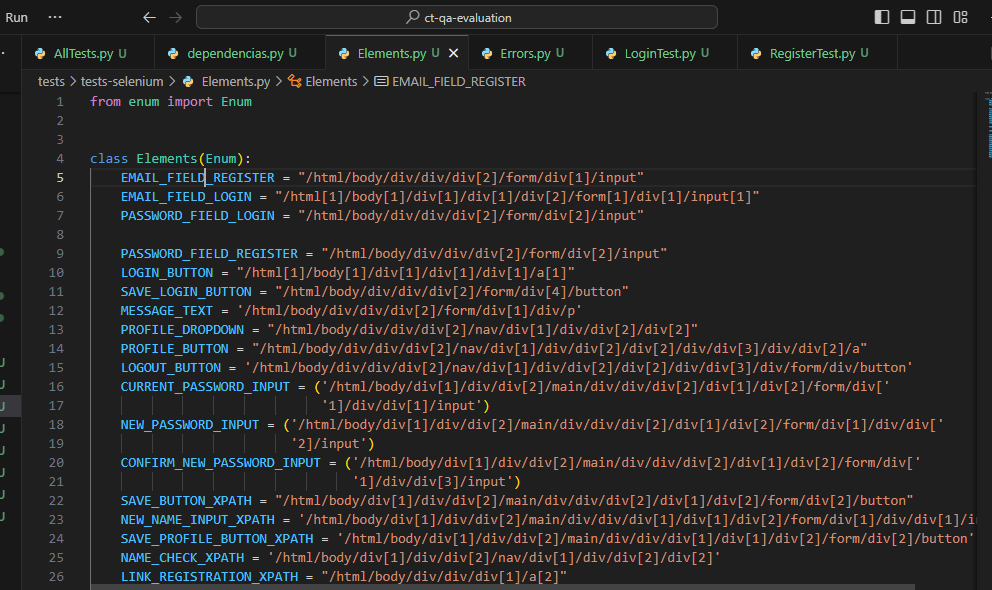
* Clonar el repositorio: el repositorio se clonará desde github por medio de gh repo clone para poder clonar el proyecto y poder tenerlo en Visual Studio Code
* Crear un archivo .env ajustándolo a cualquier variable deseable
* Iniciar el proyecto con Docker Compose Up -d : el cual hará que se inicie los servicios definidos en el archivo docker-compose.yml en segundo plano
* Acceder al contenedor de desarrollo (php.dev) utilizando Docker Compose Exec php.dev bash y ejecutar las siguientes líneas:
* Instalar las dependencias de php : composer install
* Generar una nueva clave de aplicación: php artisan key:generate
* Migrar la base de datos: php artisan migrate
* Correr el programa: php artisan service (para el Backend)
* Correr el programa: npm run dev (para el Frontend)

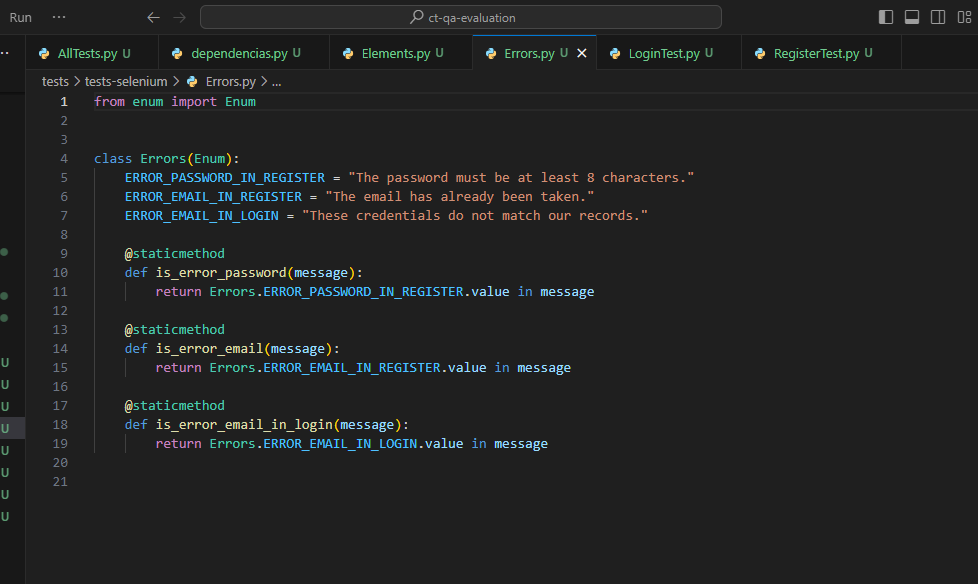
Luego de haber corrido el programa y poder observar detalladamente lo que contiene el sistema internamente, se procede a realizar los siguientes pasos solicitados para la respectiva evaluación:

Antes de realizar cada prueba, se creó un archivo donde se irán almacenando las pruebas correspondientes a realizar, teniendo dentro de este archivo las respectivas importaciones por parte de selenium con Python, creando una clase con funciones, donde cada función pertenece a los pasos que se piden al evaluar.



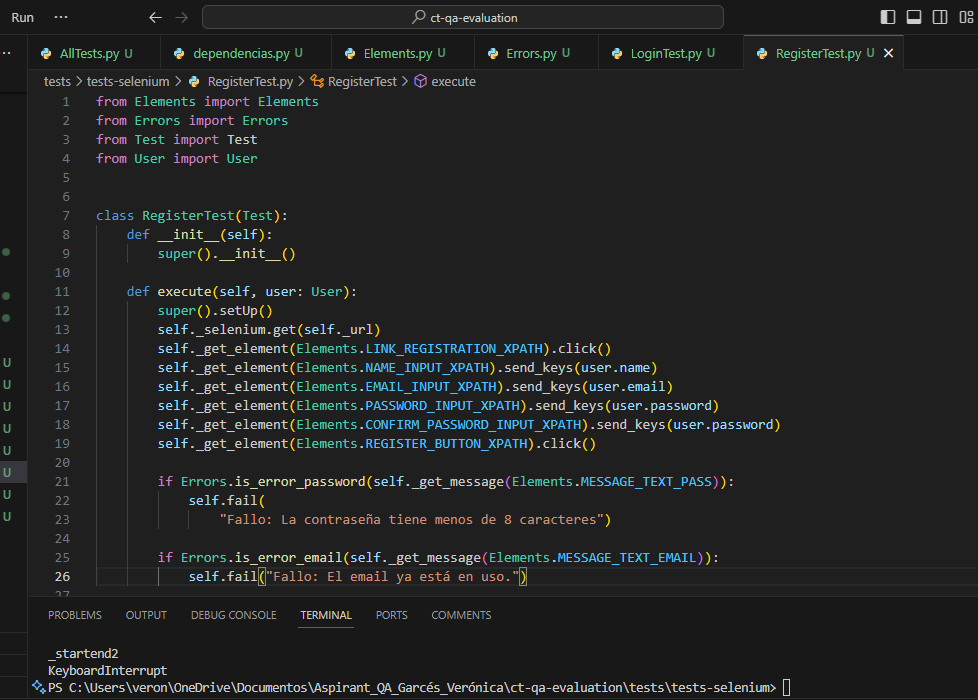
También se ha creado un archivo llamado Elements donde se almacena todos los Xpatch que se van a utilizar en cada prueba, estos Xpatch son localizadores que permiten buscar la dirección correcta de lo que se solicite al momento de hacer las respectivas pruebas. También se crea un archivo donde se almacenan los errores presentados en las deferentes pruebas solicitadas. Y finalmente se crear un archivo de Errores, donde se irán almacenando los mensajes de cada error que se presente al momento de hacer la prueba.





**Registrar nuevo usuario de prueba**

Se crea un archivo de Registro nuevo de usuario de prueba donde se declara las importaciones necesarias y se crea una clase donde tiene funciones que permite llamar a los localizadores (Xpatch), y muestra un mensaje si se comete un error previamente ya programado.

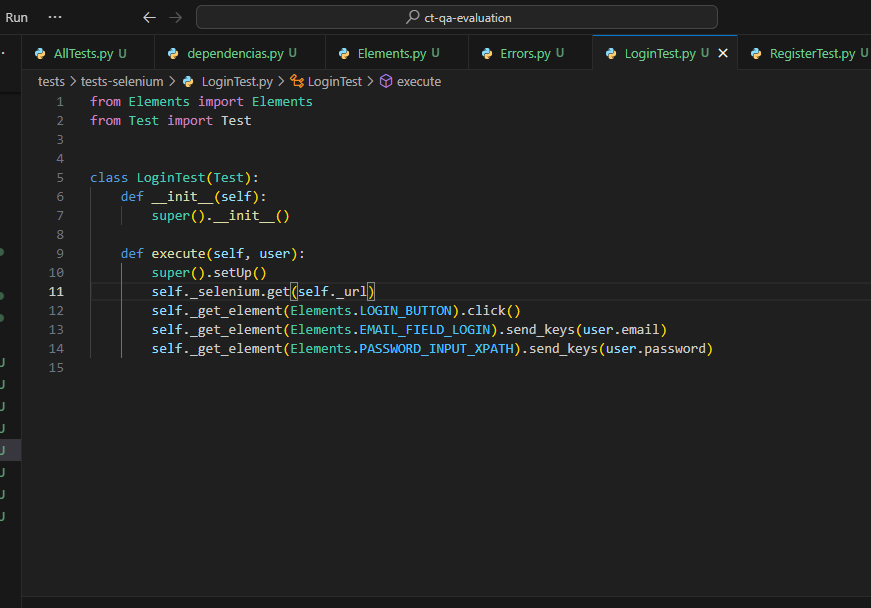


Captura de pantalla de computadora

Descripción generada automáticamente

Ingresar usuario de prueba creado

En este procedimiento, se crea un archivo donde almacena las importaciones correspondientes y también los Xpatch que son los localizadores que permitirá que al momento de hacer las pruebas con selenium vaya validando cada enlace

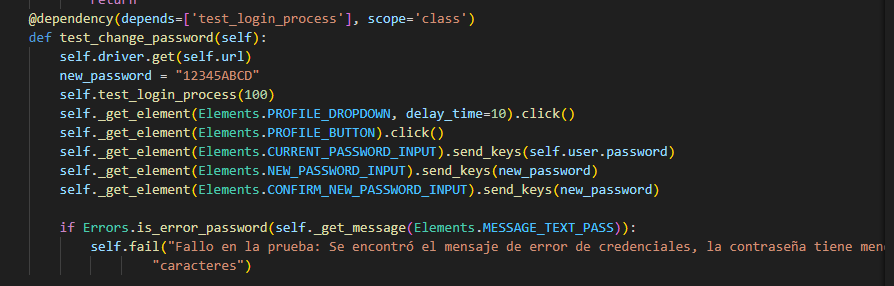


Texto

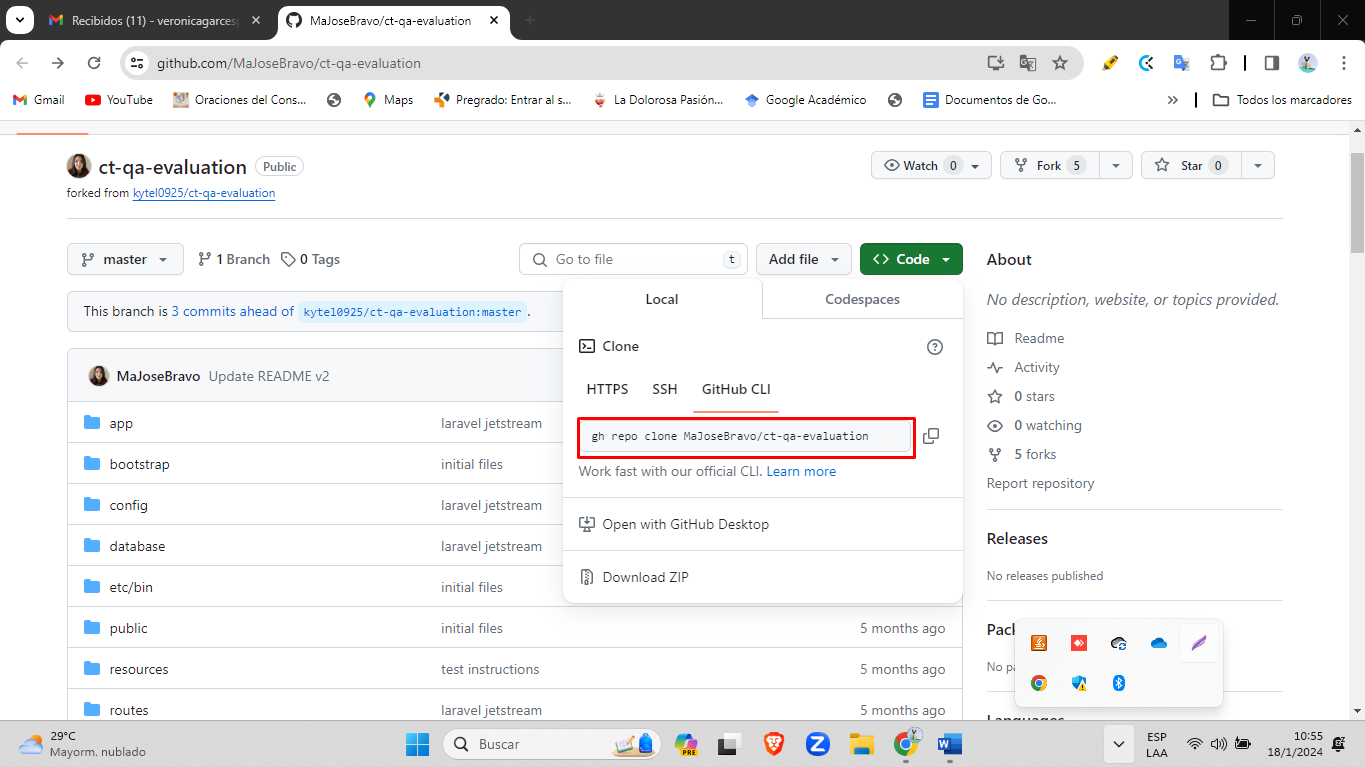
Descripción generada automáticamente

Cambiar la contraseña

Dentro del archivo donde se encuentran todas las pruebas, se crea una función donde se llaman a los xpatch que van a ser recorridos para poder realizar las pruebas



Anexos



Texto

Descripción generada automáticamente

Ilustración : Clonando el repositorio

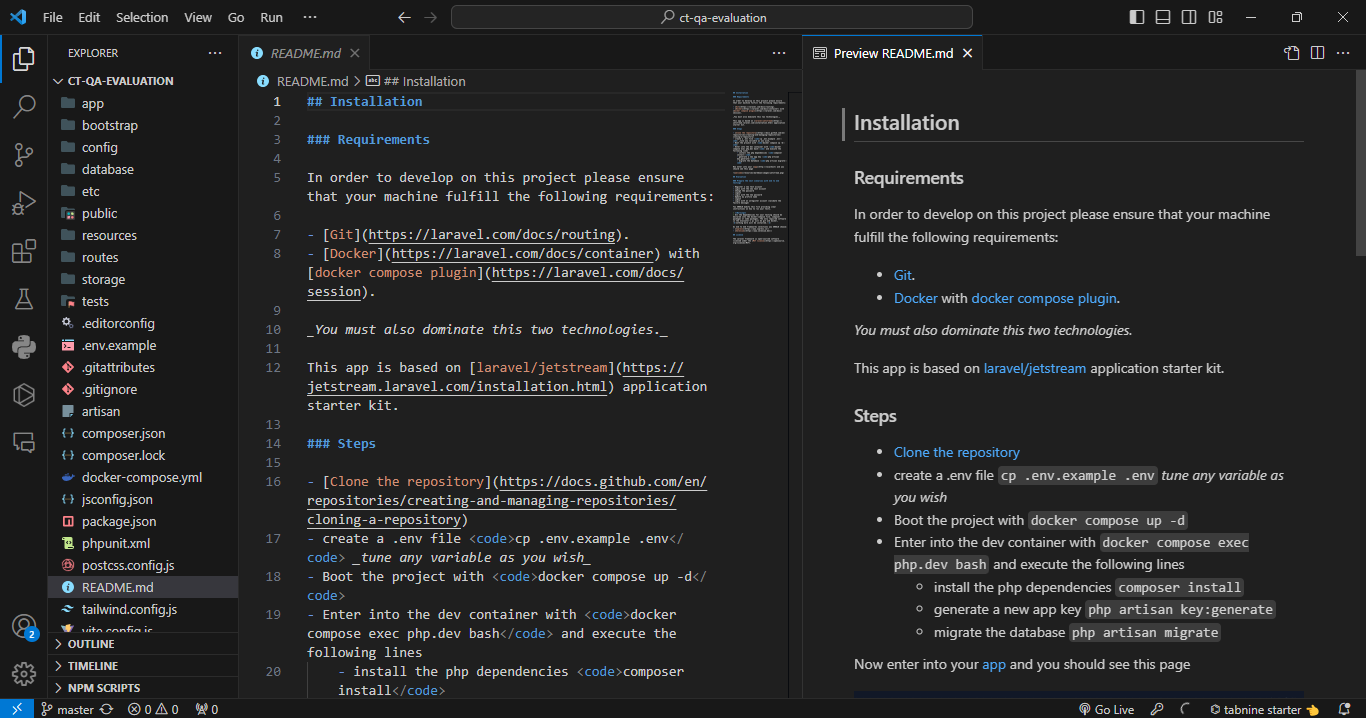


Ilustración : Repositorio Clonado

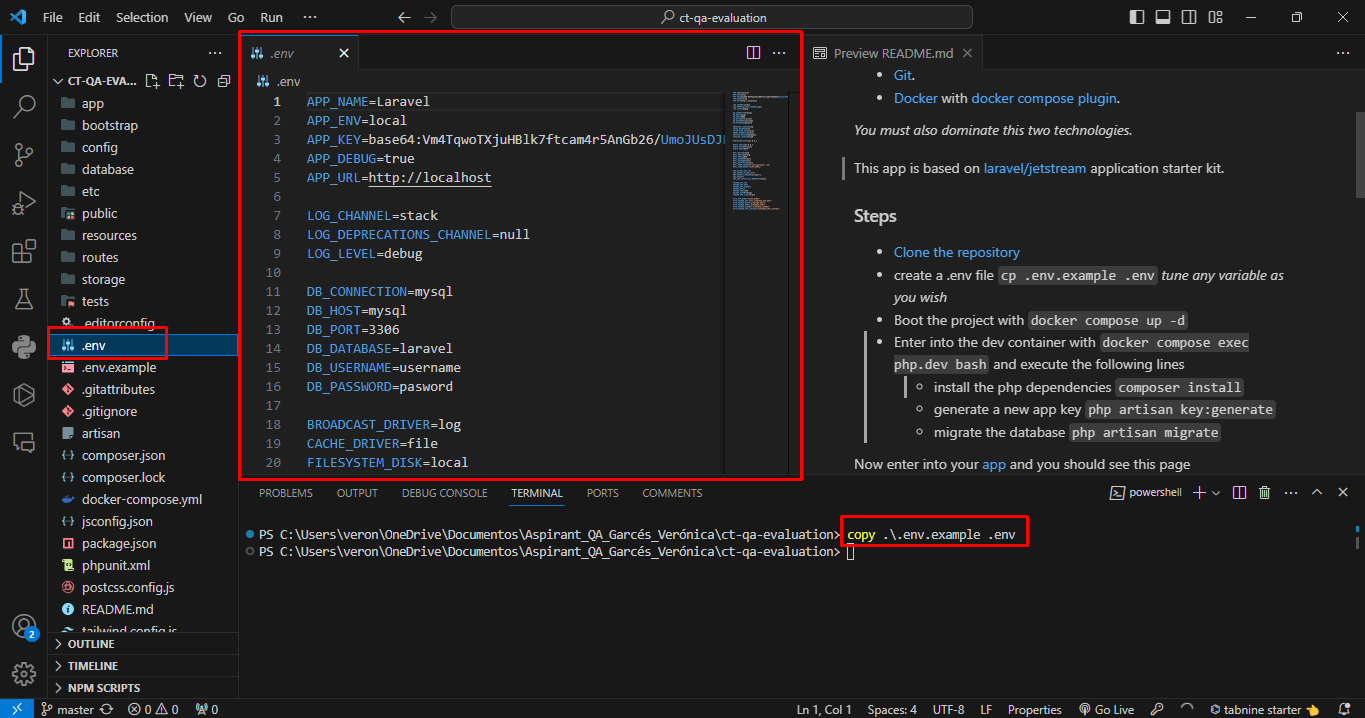
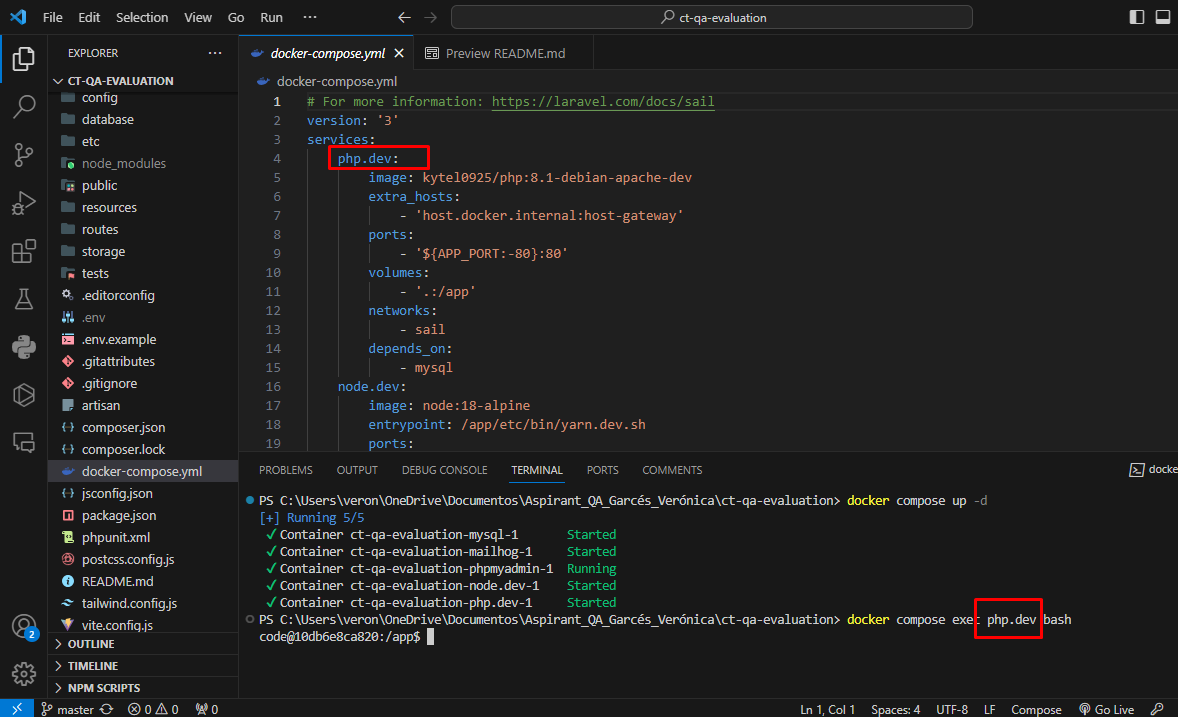


Ilustración : creación de archivo .env



Texto

Descripción generada automáticamente

Ilustración : Corriendo en el Back End

Texto

Descripción generada automáticamente

Ilustración : Corriendo en el Front End

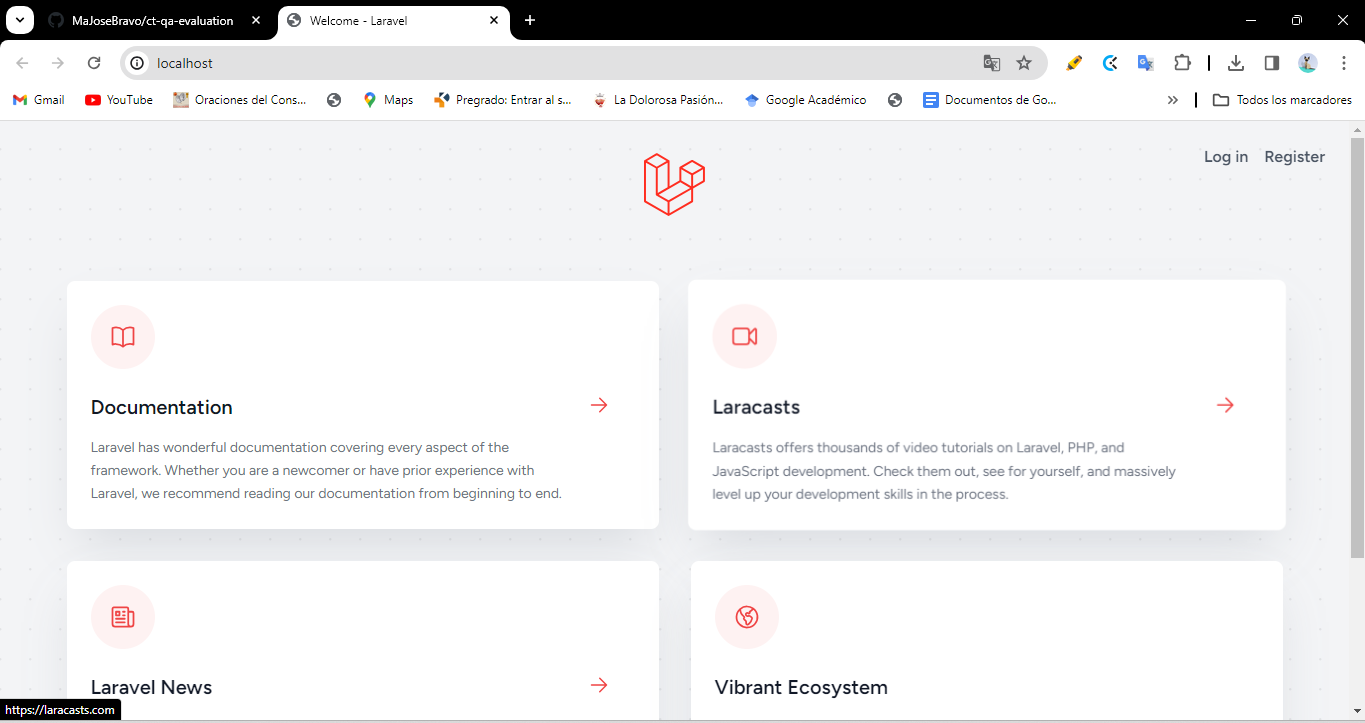


Ilustración : Página principal del sistema