



QA CANDIDATE

Michael Morán
Michaeleduardo1204@gmail.com

Contenido

Breve Descripción del Proyecto.....	2
Propósito de la Documentación	2
Resumen de Requisitos y Tecnologías Utilizadas:	2
• Requisitos:	2
• Tecnologías Utilizadas:	2
Pasos de Instalación:	2
Escenarios de Evaluación:	4
1. Registro de Usuario:	4
2. Inicio de Sesión:.....	4
3. Carga de Perfil de Usuario:	4
4. Cambio de Contraseña:	4
5. Revalidación de Sesión:	4
6. Actualización del Nombre de Usuario:	4
7. Inicio de Sesión con Cuenta No Registrada:	4
8. Nombre en el Perfil no Coincide:	5
Requisitos del Sistema:.....	5
Carpetas y tests creados:.....	5
Conclusión del proyecto:	6

Breve Descripción del Proyecto

El proyecto se basa en la aplicación de inicio [laravel/jestream](#). Proporciona una estructura y funcionalidades iniciales para el desarrollo web con Laravel. El objetivo principal del proyecto es facilitar el desarrollo y las pruebas de aplicaciones web utilizando Laravel y Jestream.

Propósito de la Documentación:

La documentación tiene como objetivo proporcionar instrucciones claras y detalladas para la instalación y configuración del proyecto. Además, establece requisitos, tecnologías utilizadas y proporciona escenarios de prueba para evaluar el correcto funcionamiento del sistema. La sección de evaluación también destaca las herramientas de prueba recomendadas, como [Cypress](#) y [Selenium](#), así como escenarios específicos para probar con el marco de prueba de Laravel (PHPUnit).

Resumen de Requisitos y Tecnologías Utilizadas:

- Requisitos:
 - Git
 - Docker con el complemento docker-compose
 - Dominio de Git y Docker
- Tecnologías Utilizadas:
 - Laravel (basado en laravel/jetstream)
 - PHPUnit (para pruebas de Laravel)
 - Cypress o Selenium (para pruebas de extremo a extremo)
 - Vite

Pasos de Instalación:

1. Clonar el repositorio.
2. Crear un archivo .env utilizando el ejemplo proporcionado y ajustar las variables según sea necesario.
3. Iniciar el proyecto con docker compose up -d.
4. Ingresar al contenedor de desarrollo con docker compose exec php.dev bash.
5. Ejecutar los siguientes comandos:
 - composer install para instalar las dependencias de PHP.
 - php artisan key:generate para generar una nueva clave de la aplicación.
 - php artisan migrate para migrar la base de datos.
 - Npm install
 - Npm run dev

6. En caso de no ejecutar los comandos en Docker, usar el entorno normal desde cmd usar los siguientes comandos desde la ruta del proyecto

- Docker-compose up -d para levantar el contenedor
- composer install para instalar las dependencias de PHP.
- php artisan key:generate para generar una nueva clave de la aplicación.
- php artisan migrate para migrar la base de datos.
- Npm install
- Npm run dev

7. Para ejecutar las pruebas en selenium

- Cd tests/tests-selenium ir a la siguiente ruta abriendo el cmd
- Python install selenium para instalar selenium para Python
- Python install pytest instalar pytest para ejecutar las pruebas en selenium
- Python -m unittest para ejecutar los respectivos tests

8. Para ejecutar las pruebas unitarias de php

- docker exec -it ct-qa-evaluation-php.dev-1 bash ejecutar el commando en la ruta del proyecto
- vendor/bin/phpunit --filter CodeApiTest una vez dentro del contenedor para ejecutar el primer test
- vendor/bin/phpunit --filter ApiTest ejecutar el segundo test
- vendor/bin/phpunit --filter ArraySortTest ejecutar el tercer test

Pruebas de ejecución de Prueba en selenium:

```
=====
ERROR: test_update_profile_fail_process (test_8_updateProfileFileTest.UpdateProfileFailTest.test_update_profile_fail_p
rocess)
=====
Traceback (most recent call last):
  File "C:\Users\micha\OneDrive\Escritorio\ct-qa-evaluation\tests\tests-selenium\test_8_updateProfileFileTest.py", line
  38, in test_update_profile_fail_process
    self.perform_login(login_page)
  File "C:\Users\micha\OneDrive\Escritorio\ct-qa-evaluation\tests\tests-selenium\test_8_updateProfileFileTest.py", line
  39, in perform_login
    login_page.login(email, password)
AttributeError: 'LoginPage' object has no attribute 'login'
=====
FAIL: test_login_fail_process (test_7_loginFailTest.LoginFailTest.test_login_fail_process)
=====
Traceback (most recent call last):
  File "C:\Users\micha\OneDrive\Escritorio\ct-qa-evaluation\tests\tests-selenium\test_7_loginFailTest.py", line 33, in test_login_fail_process
    self.perform_login(login_page, email, password)
  File "C:\Users\micha\OneDrive\Escritorio\ct-qa-evaluation\tests\tests-selenium\test_7_loginFailTest.py", line 44, in perform_login
    self.assertNotIn(error_message, login_page.message(), f'Fallo en la prueba: {error_message}')
AssertionError: 'These credentials do not match our records.' unexpectedly found in 'These credentials do not match our records.' : Fallo en la pr
ueba: These credentials do not match our records.
=====
Ran 8 tests in 88.442s

FAILED (failures=1, errors=1)
PS C:\Users\micha\OneDrive\Escritorio\ct-qa-evaluation\tests\tests-selenium> |
```

Proyecto corriendo en Docker:

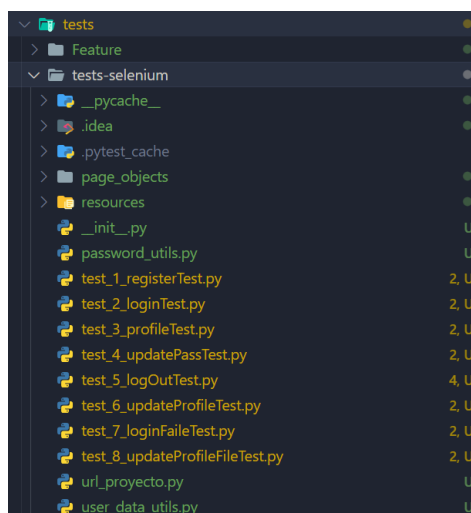
- **Descripción:** Intenta iniciar sesión con credenciales inválidas y valida el manejo de errores.
8. Nombre en el Perfil no Coincide:
- **Script:** `test_8_updateProfileFileTest.py`
 - **Descripción:** Valida si el nombre cambiado es el mismo que está en la barra de herramientas.

Requisitos del Sistema:

- Lista de requisitos para desarrollar en el proyecto.
 - Git
 - Docker y Docker Compose

Carpetas y tests creados:

- Tests/: carpeta raíz de los tests
- Feature/: pruebas unitarias creadas en php
 - ApiTest
 - ArraySortTest
 - CodeApiTest
- Tests-selenium/: Carpeta creada para alojar los test específicos de selenium
- tests-selenium/page_objects/: Clases para Page Object Model.
- tests-selenium/resources/: Datos de prueba incluyendo user_data.json.
- tests-selenium/: Scripts de pruebas.
- Feature/: Pruebas unitarias en php



Conclusión del proyecto:

El proyecto utiliza Laravel Jetstream como kit inicial de aplicación, que proporciona una estructura sólida y características integradas para el desarrollo de aplicaciones web en Laravel. La aplicación utiliza Docker para el entorno de desarrollo, lo que facilita la gestión de dependencias y la ejecución de la aplicación en diferentes entornos.

El flujo de instalación incluye clonar el repositorio, configurar un archivo **.env**, y utilizar Docker Compose para iniciar los contenedores. Se utiliza la herramienta **composer** para instalar las dependencias PHP y se ejecutan comandos de Artisan para la generación de claves y migración de la base de datos.

La documentación proporciona instrucciones claras sobre la instalación y también establece escenarios para pruebas de extremo a extremo utilizando Cypress o Selenium, así como pruebas de Laravel PHPUnit. Además, se mencionan requisitos específicos para la ejecución de las pruebas, enfatizando que solo se necesitan Git y Docker en la máquina del evaluador.

Se han definido escenarios de prueba que incluyen el registro de una nueva cuenta, inicio de sesión, carga del perfil de usuario, cambio de contraseña, cierre de sesión y otras acciones. La selección de herramientas como Cypress y Selenium para pruebas de extremo a extremo y PHPUnit para pruebas de Laravel indica un enfoque integral de pruebas.

Se han identificado algunos problemas en la ejecución de pruebas, como errores de conexión cURL y problemas en el entorno Docker. Estos problemas pueden deberse a configuraciones incorrectas, puertos bloqueados o servicios no iniciados.