

# Aplicación web de base de datos de computadora

---

Plan de pruebas funcionales

## Indice

Historial de revisiones .....	2
Introducción .....	2
Propósito.....	2
Configuración del Entorno .....	4
Proceso de Pruebas Automatizadas .....	5
Resultados y Observaciones .....	8
Complicaciones.....	9
Evidencias.....	10

## Historial de revisiones

Rev.	Fecha	Autor	Motivo de los cambios
1	2023-11-12	Hector Arismendi	Primer borrador

## Introducción

Este documento tiene como finalidad describir el plan de pruebas para el proyecto. Su objetivo es definir las estrategias y los enfoques que se adoptarán para garantizar que el sistema cumple con los requisitos establecidos y funciona según lo previsto.

El documento está estructurado para proporcionar una guía clara y detallada sobre cómo se llevarán a cabo las pruebas, los recursos necesarios para su ejecución, los criterios de aceptación para cada fase de prueba, y los procedimientos para la documentación y resolución de cualquier incidencia encontrada.

## Propósito

El propósito de este documento de pruebas es establecer un plan comprensivo y coherente que guíe las actividades de pruebas para el proyecto. Este documento tiene como objetivos:

- Definir el alcance y los enfoques de las pruebas a realizar para asegurar que el sistema cumple con los requisitos especificados.
- Identificar los recursos necesarios, incluyendo personal, herramientas y entornos, para llevar a cabo las pruebas de manera efectiva.

- Describir los casos de prueba, incluyendo los criterios de entrada y salida, para garantizar una cobertura de prueba adecuada.
- Establecer un cronograma de pruebas y asignar responsabilidades a los miembros del equipo de pruebas.
- Documentar el proceso de seguimiento de defectos y la gestión de incidencias para asegurar la resolución oportuna de los problemas encontrados.
- Proporcionar un registro claro y detallado de todas las actividades de pruebas, incluyendo los resultados de las pruebas y las evidencias de ejecución.

## Documentación del Plan de Pruebas Automatizadas

### Configuración del Entorno

- Entorno Local: Laragon.
- Entorno Alternativo: Docker (actualizar la URL de la aplicación en los scripts de prueba si es necesario).
- Estructura del Proyecto:
  - test\_codigo/: Carpeta raíz.
  - Env/: Entorno virtual Python.
  - helpers/: Funciones de ayuda.
  - page\_objects/: Clases para Page Object Model.
  - reports/: Informes de pruebas.
  - resources/: Datos de prueba incluyendo user\_data.json.
  - test/: Scripts de pruebas.
  - utils/: Herramientas y utilidades.

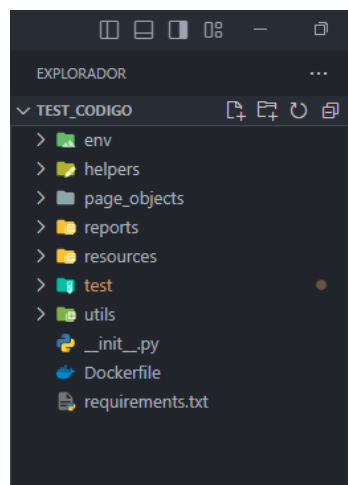


Ilustración 1

## Proceso de Pruebas Automatizadas

### 1. Registro de Usuario

- Script: registerTest.py
- Descripción: Registra un usuario y guarda las credenciales en resources/user\_data.json.
- Validaciones: Unicidad del email y requisitos de la contraseña.
- Pasos:
  - a. Abrir página de registro.
  - b. Completar y enviar formulario.
  - c. Confirmar registro y redirección.

### 2. Inicio de Sesión

- Script: loginTest.py
- Descripción: Inicia sesión con credenciales guardadas, verifica autenticidad.
- Pasos:
  - a. Abrir inicio de sesión.
  - b. Introducir credenciales.
  - c. Validar acceso al perfil.

### 3. Carga de Perfil de Usuario

- Script: profileTest.py
- Descripción: Verifica carga correcta de datos de perfil.
- Pasos:
  - a. Navegar al perfil.
  - b. Confirmar datos del perfil.

#### 4. Cambio de Contraseña

- Script: updatePassTest.py
- Descripción: Actualiza contraseña del usuario y actualiza JSON.
- Pasos:
  - a. Cambiar contraseña en perfil.
  - b. Confirmar actualización.

#### 5. Revalidación de Sesión

- Script: logOutTest.py
- Descripción: Cierra sesión y revalida cambio de contraseña.
- Pasos:
  - a. Iniciar sesión con nueva contraseña.
  - b. Confirmar acceso y cerrar sesión.

#### 6. Actualización del Nombre de Usuario

- Script: updateProfileTest.py
- Descripción: Actualiza nombre de usuario, verifica actualización en UI.
- Pasos:
  - a. Modificar nombre en perfil.
  - b. Comprobar cambio en barra de herramientas.

## 7. Inicio de Sesión con Cuenta No Registrada

- Script: loginFaileTest.py
- Descripción: Intenta inicio de sesión con credenciales inválidas, valida manejo de errores.
- Pasos:
  - a. Intentar inicio de sesión con credenciales no registradas.
  - b. Verificar mensaje de error.

## 8. Nombre en el Perfil no coincide

- Script: updateProfileFileTest.py
- Descripción: Valida si el nombre cambiado es el mismo que esta en el toolbar.
- Pasos:
  - a. Modificar nombre en perfil.
  - b. Comprobar cambio en barra de herramientas.

## Resultados y Observaciones

Las pruebas automatizadas se han ejecutado con éxito en el entorno local provisto por Laragon, lo que demuestra la funcionalidad completa y el rendimiento esperado de la aplicación web. La automatización ha abarcado el flujo de vida completo del usuario, desde el registro hasta las operaciones de inicio y cierre de sesión, incluyendo las interacciones críticas dentro del perfil del usuario.

Las validaciones implementadas para verificar la unicidad del correo electrónico y la conformidad de la contraseña con los requisitos establecidos han demostrado ser efectivas, con todos los casos de prueba pasando satisfactoriamente. El archivo `user_data.json` ha servido como un medio centralizado para gestionar las credenciales del usuario, asegurando la integridad de los datos a través de las distintas fases de prueba.

El cambio de contraseña y la actualización del nombre de usuario se han probado meticulosamente, validando no solo el cambio en la base de datos sino también su reflejo inmediato en la interfaz de usuario, lo que es esencial para una experiencia de usuario coherente. La capacidad de la suite de pruebas para manejar contraseñas aleatorias y cambios dinámicos de datos de usuario resalta su robustez y adaptabilidad.

El intento de inicio de sesión con una cuenta no registrada proporcionó una excelente oportunidad para probar la gestión de errores de la aplicación, y la respuesta adecuada del sistema a las entradas inválidas fue confirmada.

A pesar de las complicaciones iniciales con Docker, el cambio a Laragon no ha afectado la capacidad del conjunto de pruebas para ejecutar y validar los flujos de usuario. Este



cambio demuestra la portabilidad y flexibilidad de las pruebas, que pueden adaptarse a diferentes entornos sin comprometer la efectividad.

Es importante destacar que, aunque las pruebas se han completado sin errores y han cumplido con los objetivos establecidos, se recomienda mantener una supervisión continua y realizar pruebas de regresión periódicas. Estas prácticas asegurarán que cualquier cambio futuro en la aplicación mantenga la alta calidad y la funcionalidad que los usuarios esperan.

En resumen, este ciclo de pruebas ha reforzado la confianza en la calidad del software y ha establecido una base sólida para futuras iteraciones y expansiones de las capacidades de prueba automatizadas.

## **Complicaciones**

Se levanto el proyecto con un entorno en Laragon, ya que se obtuvo complicaciones con docker, todas las pruebas se desarrollo satisfactoriamente en ese entorno, en tal caso de hacerlo con docker sera de la misma manera solo cambiando el link de la ruta.

# Evidencias

```
cmd C:\Windows\System32\cmd.exe
DevTools listening on ws://127.0.0.1:54403/devtools/browser/77c94b84-b46e-4000-80a6-86a286239685
[1112/190258.370:INFO:CONSOLE(229)] "[vite] connecting...", source: http://[::1]:5173/@vite/client (229)
[1112/190258.377:INFO:CONSOLE(324)] "[vite] connected.", source: http://[::1]:5173/@vite/client (324)
[1112/190258.691:INFO:CONSOLE(0)] "Autofocus processing was blocked because a document already has a focused element.", source: http://localhost:8091/login (0)
.
DevTools listening on ws://127.0.0.1:54436/devtools/browser/26055146-5e68-46e5-ae9c-fa6ab5372e38
[1112/190309.920:INFO:CONSOLE(229)] "[vite] connecting...", source: http://[::1]:5173/@vite/client (229)
[1112/190309.924:INFO:CONSOLE(324)] "[vite] connected.", source: http://[::1]:5173/@vite/client (324)
[1112/190310.243:INFO:CONSOLE(0)] "Autofocus processing was blocked because a document already has a focused element.", source: http://localhost:8091/login (0)
ExampleNewName
.
DevTools listening on ws://127.0.0.1:54482/devtools/browser/9d30698e-14ce-41fe-aaed-142604b578ac
[1112/190327.894:INFO:CONSOLE(229)] "[vite] connecting...", source: http://[::1]:5173/@vite/client (229)
[1112/190327.901:INFO:CONSOLE(324)] "[vite] connected.", source: http://[::1]:5173/@vite/client (324)
[1112/190328.228:INFO:CONSOLE(0)] "Autofocus processing was blocked because a document already has a focused element.", source: http://localhost:8091/login (0)
F
DevTools listening on ws://127.0.0.1:54512/devtools/browser/d8a7d3bf-ee28-472e-8dc2-c620fa0617a3
[1112/190332.251:INFO:CONSOLE(229)] "[vite] connecting...", source: http://[::1]:5173/@vite/client (229)
[1112/190332.258:INFO:CONSOLE(324)] "[vite] connected.", source: http://[::1]:5173/@vite/client (324)
[1112/190332.571:INFO:CONSOLE(0)] "Autofocus processing was blocked because a document already has a focused element.", source: http://localhost:8091/login (0)
F
=====
FAIL: test_login_faile_process (test.test_7_loginFaileTest.LoginFaileTest.test_login_faile_process)
-----
Traceback (most recent call last):
  File "C:\Users\HRams\OneDrive\Escritorio\testAssignment\test_codigo\test\test_7_loginFaileTest.py", line 35, in test_login_faile_process
    self.fail("Fallo en la prueba: Se encontró el mensaje de error de credenciales")
AssertionError: Fallo en la prueba: Se encontró el mensaje de error de credenciales
=====
FAIL: test_profile_process (test.test_8_updateProfileFileTest.ProfileTest.test_profile_process)
-----
Traceback (most recent call last):
  File "C:\Users\HRams\OneDrive\Escritorio\testAssignment\test_codigo\test\test_8_updateProfileFileTest.py", line 48, in test_profile_process
    self.fail("Fallo en la prueba: No se encontró el nombre de usuario en Perfil ")
AssertionError: Fallo en la prueba: No se encontró el nombre de usuario en Perfil
-----
Ran 8 tests in 104.433s

FAILED (failures=2)

C:\Users\HRams\OneDrive\Escritorio\testAssignment\test_codigo>
```

Evidencia 1