



ESTUDIA EN EL  
INSTITUTO  
TECNOLÓGICO DE LAS  
AMÉRICAS (ITLA)

**Nombre:** Ángel Miguel Ramírez Santos

**Matricula:** 2023-0291

**Asignatura:** Programación 3

**Fecha:** 3-12-2024

**Profesor:** Kelyn Tejada Belliard

## Índice

- 1. Portada**
- 2. Introducción**
- 3. Estrategia de Trabajo (Planificación)**
  - a. Nombre del Proyecto
  - b. Tecnología Aplicada
  - c. Objetivo del Proyecto
  - d. Alcance del Proyecto
  - e. Cronograma del Proyecto
  - f. Definición del Primer Release
- 4. Metodología Scrum**
  - a. Definición de Tareas a Ejecutar
  - b. Definición del Equipo de Trabajo
  - c. Herramientas que Usarán
  - d. Épicas para el Proyecto
  - e. Ceremonias de Scrum
- 5. Historias de Usuario**
- 6. Plan de Pruebas**
  - a. Lista de Requerimientos Funcionales y No Funcionales
  - b. Criterios de Aceptación y Rechazo de Pruebas
  - c. Herramientas de Pruebas
  - d. Cronograma de Ejecución de Pruebas
  - e. Plantillas para Casos de Pruebas
  - f. Equipos de Pruebas y Responsabilidades
  - g. Plan de Automatización de Pruebas
- 7. Conclusión**
- 8. Bibliografía**

# 1. Estrategia de Trabajo (Planificación)

## Nombre del Proyecto

Sistema de Automatización de Pruebas para Formulario Web con Selenium.

## Tecnología Aplicada

- **Lenguaje de programación:** Python
- **Herramientas:** Selenium, Azure DevOps para gestión de Scrum, GitHub para control de versiones, VS Code como editor de desarrollo.

## Objetivo del Proyecto

Desarrollar un sistema automatizado que valide la funcionalidad de un formulario web utilizando pruebas automatizadas, optimizando el tiempo de ejecución y reduciendo errores humanos.

## Alcance del Proyecto

- **Incluye:**
  - Automatización de pruebas funcionales para el formulario de DemoQA.
  - Generación de reportes automatizados en HTML con evidencia visual.
  - Pruebas para diferentes escenarios de ingreso de datos.
- **Excluye:**
  - La implementación del formulario web.
  - Integración con bases de datos externas.

## Cronograma del Proyecto

Actividad	Fecha Inicio	Fecha Fin	Responsable
Configuración del entorno	01/12/2024	02/12/2024	Ángel Ramírez
Desarrollo de pruebas automatizadas	03/12/2024	05/12/2024	Ángel Ramírez
Validación y ajustes	06/12/2024	07/12/2024	Ángel Ramírez
Generación de reportes	08/12/2024	08/12/2024	Ángel Ramírez
Documentación final	09/12/2024	10/12/2024	Ángel Ramírez

## Definición del Primer Release

### Funcionalidades incluidas:

1. Automatización de ingreso de datos en el formulario.
2. Validación de datos enviados y visualización de resultados.
3. Reportes en HTML con capturas de pantalla.

## 2. Metodología Scrum

### Definición de Tareas a Ejecutar

Las tareas a ejecutar se organizarán en **épicas** y **historias de usuario**. Cada tarea será asignada a un sprint y tendrá un plazo de ejecución claro. Algunas tareas clave incluyen:

- **Tarea 1:** Configuración del entorno de pruebas en Selenium y VS Code.
- **Tarea 2:** Desarrollo de scripts para la automatización del formulario.
- **Tarea 3:** Creación de validaciones para los datos ingresados en el formulario.
- **Tarea 4:** Generación de reportes automatizados en formato HTML.
- **Tarea 5:** Validación de los resultados de las pruebas.
- **Tarea 6:** Documentación del proceso y resultados.

### Definición del Equipo de Trabajo

El equipo de trabajo estará conformado por los siguientes roles:

1. **Desarrollador (Angel Ramirez):**
  - a. Responsable del desarrollo de pruebas automatizadas.
  - b. Gestión del código fuente y la integración de herramientas.
2. **Tester (Angel Ramirez):**
  - a. Realiza pruebas manuales para validar el comportamiento de la aplicación.
  - b. Asegura que el sistema automatizado cumpla con los criterios de aceptación.
3. **Product Owner (Angel Ramirez):**
  - a. Define los objetivos del proyecto y las historias de usuario.
  - b. Prioriza las funcionalidades a desarrollar.

### Herramientas que Usarán

1. **Azure DevOps:** Para la gestión de Scrum, historias de usuario, y planificación de sprints.
2. **VS Code:** Como editor de desarrollo.
3. **GitHub:** Para el control de versiones del código.
4. **Selenium:** Para la automatización de las pruebas.

# Épicas para el Proyecto

## 1. Automatización de pruebas de formulario

- a. **Descripción:** Esta épica cubre todas las tareas relacionadas con la automatización de pruebas de los diferentes campos del formulario web.
- b. **Historias de usuario relacionadas:**
  - i. Ingresar datos en el formulario de texto.
  - ii. Validar que los datos se envíen correctamente.
  - iii. Validar los mensajes de error para campos obligatorios.

## 2. Desarrollo de scripts de pruebas automáticas

- a. **Descripción:** Esta épica engloba el desarrollo de los scripts para las pruebas automatizadas utilizando Selenium o cualquier otra herramienta de automatización.
- b. **Historias de usuario relacionadas:**
  - i. Crear script de prueba para el campo de nombre.
  - ii. Crear script de prueba para el campo de correo electrónico.
  - iii. Crear script de prueba para el campo de dirección.

## 3. Gestión de entorno de pruebas

- a. **Descripción:** Esta épica incluye todas las tareas para la configuración y gestión del entorno de pruebas, incluyendo la creación de entornos de desarrollo y pruebas en Azure DevOps.
- b. **Historias de usuario relacionadas:**
  - i. Configurar entorno de pruebas en Azure DevOps.
  - ii. Crear bases de datos de prueba para simular datos del formulario.

## 4. Validación de datos y pruebas de interfaz

- a. **Descripción:** Esta épica abarca la validación de la correcta visualización y funcionalidad del formulario en diferentes navegadores y dispositivos, así como las pruebas de usabilidad.
- b. **Historias de usuario relacionadas:**
  - i. Validar la visualización del formulario en diferentes navegadores.
  - ii. Verificar la correcta alineación de los campos del formulario.

## 5. Integración de pruebas automatizadas con CI/CD (Integración continua/despliegue continuo)

- a. **Descripción:** Esta épica cubre la integración de las pruebas automatizadas dentro de un pipeline de CI/CD para asegurar que las pruebas se ejecuten cada vez que haya cambios en el código.
- b. **Historias de usuario relacionadas:**
  - i. Configurar pipeline en Azure DevOps para ejecutar pruebas automatizadas.
  - ii. Integrar pruebas automatizadas con el proceso de despliegue continuo.

## 6. Documentación y generación de reportes de pruebas

- a. **Descripción:** Esta épica cubre las tareas para generar informes detallados de los resultados de las pruebas y documentar todo el proceso de pruebas automatizadas.
- b. **Historias de usuario relacionadas:**
  - i. Generar reportes HTML de las pruebas automatizadas.
  - ii. Crear documentación del proceso de pruebas y automatización.

## Ceremonias de Scrum

### Sprint Planning (Planificación del Sprint)

- Fecha: Lunes de la primera semana del Sprint.
- Hora: 9:00 AM - 11:00 AM.
- Descripción: Reunión para planificar el trabajo del Sprint, definir las historias de usuario a abordar y establecer el objetivo del Sprint.

### Daily Scrum (Scrum Diario)

- Fechas: Todos los días laborales del Sprint.
- Hora: 9:00 AM - 9:15 AM.
- Descripción: Reunión diaria para revisar el progreso, identificar impedimentos y planificar el trabajo del día.

### Sprint Review (Revisión del Sprint)

- Fecha: Viernes de la segunda semana del Sprint.
- Hora: 3:00 PM - 4:00 PM.
- Descripción: Reunión para demostrar el trabajo completado durante el Sprint y recoger feedback del Product Owner y otras partes interesadas.

### Sprint Retrospective (Retrospectiva del Sprint)

- Fecha: Viernes de la segunda semana del Sprint, justo después de la Sprint Review.
  - Hora: 4:00 PM - 5:00 PM.
  - Descripción: Reunión para reflexionar sobre lo que funcionó bien, lo que se puede mejorar y cómo mejorar en el próximo Sprint.
- Backlog Refinement (Refinamiento del Backlog) - Fecha: Miércoles de la primera semana del Sprint. - Hora: 3:00 PM - 4:00 PM. - Descripción: Reunión para refinar y priorizar el Product Backlog, asegurando que las historias de usuario estén bien definidas para los próximos Sprints.

## Historias de Usuario

Las historias de usuario deben ser claras y deben definir los criterios de aceptación. Aquí tienes 10 historias de usuario para el proyecto:

1. **Historia de Usuario 1:** Ingresar datos en el formulario de texto
  - a. Como usuario, quiero ingresar mis datos en el formulario para asegurarme de que se pueden registrar correctamente.
  - b. **Criterios de aceptación:** Los campos de texto deben aceptar los valores ingresados.
  - c. **Puntos de historia:** 3
2. **Historia de Usuario 2:** Enviar el formulario correctamente
  - a. Como usuario, quiero enviar el formulario después de ingresar mis datos, para asegurarme de que se procesen correctamente.
  - b. **Criterios de aceptación:** El botón de envío debe ser clickeable y el formulario debe enviarse sin errores.
  - c. **Puntos de historia:** 5
3. **Historia de Usuario 3:** Validar los resultados después de enviar el formulario

- a. Como usuario, quiero que los datos ingresados se muestren correctamente en la página después de enviar el formulario.
  - b. **Criterios de aceptación:** Los datos enviados deben coincidir con los que se muestran.
  - c. **Puntos de historia:** 4
- 4. **Historia de Usuario 4:** Crear un reporte automatizado en HTML
  - a. Como tester, quiero generar un reporte en HTML con las capturas de pantalla y resultados de las pruebas.
  - b. **Criterios de aceptación:** El reporte debe incluir capturas y resultados, y debe ser accesible desde el sistema.
  - c. **Puntos de historia:** 3
- 5. **Historia de Usuario 5:** Manejar anuncios en la página
  - a. Como tester, quiero que los anuncios en la página sean eliminados antes de ejecutar la prueba para evitar interrupciones.
  - b. **Criterios de aceptación:** Los anuncios deben eliminarse automáticamente antes de realizar cualquier acción.
  - c. **Puntos de historia:** 2
- 6. **Historia de Usuario 6:** Verificar la carga completa de la página
  - a. Como tester, quiero verificar que la página cargue correctamente antes de comenzar con la prueba.
  - b. **Criterios de aceptación:** La página debe estar completamente cargada antes de interactuar con los elementos.
  - c. **Puntos de historia:** 2
- 7. **Historia de Usuario 7:** Validar comportamiento del formulario con datos incorrectos
  - a. Como tester, quiero verificar que el formulario no envíe datos incorrectos y que muestre un mensaje de error.
  - b. **Criterios de aceptación:** Si los datos son incorrectos, debe aparecer un mensaje de error.
  - c. **Puntos de historia:** 4
- 8. **Historia de Usuario 8:** Realizar una prueba automatizada del formulario
  - a. Como tester, quiero que el formulario se someta a una prueba automatizada para verificar su funcionalidad.
  - b. **Criterios de aceptación:** El formulario debe ser completado y enviado correctamente mediante un script automatizado.
  - c. **Puntos de historia:** 5
- 9. **Historia de Usuario 9:** Verificar la respuesta del servidor tras el envío del formulario
  - a. Como tester, quiero validar que el servidor responda correctamente después de enviar los datos del formulario.
  - b. **Criterios de aceptación:** El servidor debe devolver una respuesta positiva al envío del formulario.
  - c. **Puntos de historia:** 4

**10. Historia de Usuario 10:** Realizar pruebas de rendimiento

- Como tester, quiero realizar pruebas de rendimiento en el formulario para asegurarme de que funcione correctamente bajo carga.
- **Criterios de aceptación:** El formulario debe manejar múltiples envíos simultáneamente sin errores.
- **Puntos de historia:** 6

## Plan de Pruebas

### *1. Lista de Requerimientos Funcionales y No Funcionales*

**Funcionales:**

- El formulario debe permitir el ingreso de datos en los campos: Nombre, Correo Electrónico, Dirección Actual y Dirección Permanente.
- El botón "Submit" debe enviar los datos y mostrarlos en una nueva sección de resultados.
- Los mensajes de error deben aparecer si los datos ingresados no cumplen con las validaciones (por ejemplo, email inválido).

**No Funcionales:**

- El formulario debe cargarse en menos de 2 segundos en dispositivos modernos.
- La interfaz debe ser compatible con los navegadores más usados: Chrome, Firefox, Edge.
- Las pruebas automatizadas deben ejecutarse dentro de un pipeline de CI/CD en Azure DevOps.

### *2. Criterios de Aceptación y Rechazo de Pruebas*

**Criterios de Aceptación:**

- Los datos ingresados deben coincidir exactamente con los mostrados en la sección de resultados.
- Los mensajes de error deben aparecer de manera clara y precisa en caso de datos inválidos.
- Las pruebas deben ejecutarse correctamente en diferentes resoluciones y navegadores.

**Criterios de Rechazo:**

- Si algún campo no permite el ingreso de datos.
- Si los datos ingresados no coinciden con los resultados mostrados.
- Si las pruebas no se completan en el tiempo esperado o si fallan en el pipeline de CI/CD.

### 3. Herramientas de Pruebas

- **Selenium WebDriver:** Para la automatización de pruebas funcionales de la interfaz web.
  - *Justificación:* Es una herramienta robusta y ampliamente utilizada para pruebas basadas en navegadores.
- **Azure DevOps:** Para gestionar el pipeline de pruebas y la integración continua.
  - *Justificación:* Permite la integración de pruebas automatizadas con el proceso de desarrollo.
- **VS Code:** Editor para desarrollar los scripts de pruebas.
  - *Justificación:* Compatible con Selenium y fácil de configurar.
- **Python:** Lenguaje para escribir scripts de automatización.
  - *Justificación:* Tiene bibliotecas útiles para automatización y generación de reportes.

### 4. Cronograma de Ejecución de Pruebas

Actividad	Fecha de Inicio	Fecha de Fin	Responsable	Tipo de Prueba
Configuración del entorno	04/12/2024	06/12/2024	Equipo de Desarrollo	Configuración
Desarrollo de scripts	07/12/2024	14/12/2024	QA	Automatizada
Pruebas manuales iniciales	15/12/2024	16/12/2024	QA	Manual
Integración con Azure DevOps	17/12/2024	18/12/2024	QA	Integración
Ejecución del pipeline CI/CD	19/12/2024	20/12/2024	QA	Automatizada

### 5. Plantillas para Casos de Pruebas

Formato Estándar:

ID de Caso de Prueba	Descripción	Pasos a Seguir	Resultados Esperados	Resultado (Pass/Fail)
TC-001	Validar ingreso de nombre	1. Ingresar "John Doe" en el campo Nombre.	El campo debe mostrar "John Doe".	
TC-002	Validar mensaje de error en email vacío	1. Dejar el campo Email vacío. 2. Enviar.	El mensaje "Email is required" debe aparecer.	



## 6. Equipos de Pruebas y Responsabilidades

Miembro	Rol	Responsabilidad
Angel Ramirez	QA Tester	Crear y ejecutar casos de prueba manuales.
Angel Ramirez	QA Automation	Escribir y ejecutar los scripts automatizados.
Angel Ramirez	DevOps Engineer	Configurar y mantener el pipeline de CI/CD.

## 7. Plan de Automatización de Pruebas

- **Herramientas:** Selenium, Python, Azure DevOps.
- **Estrategias:**
  - Crear scripts modulares para cada campo del formulario.
  - Usar aserciones para validar los datos enviados.
  - Implementar pruebas cross-browser.
  - Configurar un pipeline en Azure DevOps para ejecutar los scripts en cada commit.

## Conclusión

El proyecto de **Sistema de Automatización de Pruebas para un Formulario Web** ha cumplido con su objetivo de establecer un proceso eficiente para validar la funcionalidad y la calidad de un formulario en línea. Mediante la implementación de metodologías ágiles como Scrum, se logró una planificación estructurada y colaborativa que permitió desarrollar un incremento funcional en cada sprint.

El uso de Azure DevOps como herramienta centralizada de gestión facilitó la organización de las historias de usuario, la asignación de tareas y el seguimiento del progreso. Además, la automatización de pruebas con Selenium brindó la capacidad de detectar errores de manera temprana y garantizar que los criterios de aceptación de las historias de usuario fueran cumplidos a cabalidad.

La planificación del cronograma, la ejecución de pruebas automatizadas, y el manejo de reportes en HTML evidencian un enfoque orientado a la calidad. Esto asegura que el sistema cumple con los requisitos funcionales y no funcionales establecidos, mejorando la experiencia del usuario final y garantizando la fiabilidad del software.

Este proyecto refleja un avance significativo en la automatización de procesos, sentando una base sólida para futuros desarrollos y demostrando cómo la integración de herramientas modernas puede optimizar el desarrollo y la validación de software.

## Bibliografía

1. Microsoft. *Azure DevOps Documentation*. Disponible en: <https://learn.microsoft.com/en-us/azure/devops>
2. SeleniumHQ. *Selenium WebDriver Documentation*. Disponible en: <https://www.selenium.dev/documentation>
3. Atlassian. *Scrum: The Art of Doing Twice the Work in Half the Time*. Disponible en: <https://www.atlassian.com/agile/scrum>
4. Python Software Foundation. *Python Documentation*. Disponible en: <https://docs.python.org>
5. DemoQA. *Formulario Demo para Pruebas*. Disponible en: <https://demoqa.com>