**Proyecto: Automatización de Pruebas con Selenium**

David Stiven Silva Velandia

U Compensar

Facultad de ingeniería

Pruebas de Software I

Profesor Julián David Gómez Pérez

**Contenido**

[Introducción 2](#_Toc486456546)

[Actividad 2](#_Toc1161917218)

[Objetivo 2](#_Toc373567600)

[Requisitos del proyecto 3](#_Toc1071465107)

[Registro de Usuario 3](#_Toc738637404)

[Carga de Archivos 3](#_Toc1360276695)

[Descarga de Archivos 3](#_Toc1678736280)

[Alertas 3](#_Toc1328891609)

[Modo Headless 3](#_Toc929555880)

[Entregables 3](#_Toc854361367)

[Documentación 4](#_Toc1600475084)

[Desarrollo 4](#_Toc546560906)

[Guía para la instalación y ejecución 5](#_Toc281206814)

[Windows 5](#_Toc569889843)

[Linux 5](#_Toc1709720515)

[Reporte HTML con resultados de pruebas 5](#_Toc961009766)

[Explicación de recargas y modo Headless 6](#_Toc1654289206)

[Video demostración 6](#_Toc952281456)

[Rutas para su verificación 7](#_Toc1178246292)

[Estructura gráfica del proyecto 7](#_Toc1693054126)

[Herramientas utilizadas para todo el proyecto 8](#_Toc1810295858)

[Fuentes de información y recursos adicionales 9](#_Toc1593042386)

[Conclusiones 10](#_Toc1732906875)

[Fuentes 11](#_Toc1283557278)

# **Introducción**

Dentro del mundo del desarrollo de software existen diferentes ramas de conocimiento que, con el pasar de los años, crecen más. Una de estas ramas es la de calidad, QA (Quality Assurance). Dentro de esta área, se ve todo el tema de revisión de esos aspectos de software que se pueden mejorar a través de procesos o pruebas que permitan medir esa calidad y hacer esas propuestas de cambio con consecuencias positivas. Por este motivo, nace este proyecto de pruebas automatizadas, como demostración de todo lo aprendido durante el segundo corte en la materia pruebas de software I.

# **Actividad**

Durante esta actividad, se realizarán diferentes automatizaciones de pruebas hechas a sitios web que se crearon estrictamente para hacer estos procesos de práctica. Se propone un objetivo y unos requisitos que abrirán paso al proceso hecho en Python utilizando Selenium para dichas automatizaciones. También se documentará toda la instalación y ejecución, así como el reporte del resultado de las pruebas.

## **Objetivo**

Se tiene como objetivo desarrollar pruebas automatizadas para [https://demoqa.com](https://demoqa.com/) , cubriendo:

* Formularios (registro de usuario).
* Carga/Descarga de archivos (en la sección "Upload and Download").
* Alertas (en la sección "Alerts").
* Modo Headless y generación de reportes.

## **Requisitos del proyecto**

Dentro de los requisitos del proyecto se tienen unos casos de pruebas obligatorias con sus respectivas descripciones.

### ***Registro de Usuario***

* Rellenar el formulario en <https://demoqa.com/automation-practice-form> .
* Validar que el mensaje de éxito contenga "Thanks for submitting the form".

### ***Carga de Archivos***

* Subir un archivo en <https://demoqa.com/upload-download>.
* Verificar que el nombre del archivo aparece en la página.

### ***Descarga de Archivos***

* Descargar el archivo "sampleFile.jpeg" y validar su existencia en una carpeta específica.

### ***Alertas***

* Interactuar con las alertas en <https://demoqa.com/alerts> :
* Aceptar una alerta de confirmación.
* Capturar texto de una alerta de prompt y validar respuesta.

### ***Modo Headless***

* Hacer las descargas automáticas en C:/downloads (Windows) o ~/downloads (Mac/Linux).

## **Entregables**

Código fuente:

* Repositorio en GitHub/GitLab con Scripts de prueba.

## **Documentación**

* Guía de instalación y ejecución: se pretende mostrar toda la guía de instalación y ejecución de las herramientas utilizadas para las pruebas (Python, Selenium, VSCODE). Esto se hará en dos sistemas operativos. Windows y Linux, el segundo se debe a que es el sistema operativo que actualmente estoy utilizando.
* Reporte HTML con resultados de las pruebas: se hará todo el reporte de resultados mediante un documento de HTML.
* Explicación de cómo se manejaron las descargas y el modo Headless: se explicará todo el proceso de cómo se hizo el manejo de las descargas y el modo Headless y cómo es su funcionamiento e importancia dentro de las pruebas automatizadas. Esto se hará a través del documento de Word.
* Video Demostración que muestre la ejecución de cada una de las pruebas y las validaciones de archivos: se hará un video con dos versiones de todo el proceso de ejecución de las pruebas. Las dos versiones se refieren a una en español y otra en inglés.

# **Desarrollo**

Para el desarrollo del proyecto se tiene como documento principal de guía y formateado en normas APA un PDF con toda la información pertinente. El documento contará con todos los enlaces externos a otras herramientas y se aclara que irá dentro de una carpeta comprimida, que también contendrá un archivo HTML para el reporte y un documento de apoyo para la explicación de las recargas y el modo Headless. De igual manera, dos últimos archivos en video donde se explicará todo el proceso y su funcionalidad. Cada uno de los videos tendrá el mismo contenido lo único que cambiará será el idioma en que se explique, ya que uno estará en inglés y el otro en español. Al final todos, los archivos estarán en una carpeta raíz y se encontrarán en un repositorio de GitHub.

## **Guía para la instalación y ejecución**

Para este proceso guía para la instalación y ejecución tendrá dos partes fundamentales. La primera será el proceso hecho en Windows, que es el más común. La segunda será Linux ya que es el sistema operativo que actualmente se está utilizando. Para tener una guía más explícita y en un formato de mejor calidad, se utilizará la herramienta de documentación Notion. Para cada uno de los procesos habrá un link directo al documento específico de Notion.

### ***Windows***

[Enlace de documentación notion](https://fair-fish-419.notion.site/Gu-a-de-instalaci-n-de-entorno-de-pruebas-automatizadas-Selenium-Python-para-Windows-Spanish-1d7cfa5b0c8480888e80c882493e58d7)

### ***Linux***

[Enlace de documentación notion](https://fair-fish-419.notion.site/Gu-a-de-instalaci-n-de-entorno-de-pruebas-automatizadas-Selenium-Python-para-Linux-Spanish-1d7cfa5b0c84807ea73ef55ea3fdbb32)

## **Reporte HTML con resultados de pruebas**

Dentro del reporte HTML se tiene un archivo HTML específico dentro de la carpeta de del proyecto donde estarán también adjuntos las capturas de cada proceso. Serán en su totalidad 23 capturas que mostrarán desde las páginas principales hasta cada parte del proceso hecho para comprobar la funcionad de cada parte de la página.

También se mostrarán los resultados de las pruebas, teniendo en cuenta la evidencia, para concluir si todas las funcionalidades trabajan correctamente. Este tendrá unas conclusiones, solamente del proceso de pruebas, ya que el trabajo en general también tendrá unas conclusiones que se verán más adelante en el presente documento.

## **Explicación de recargas y modo Headless**

La explicación de modo Headless y de recargas, se dará dentro del video que se explicará en el siguiente punto. Esto incluirá, también, toda la explicación del proyecto.

## **Video demostración**

Se grabará un video con la herramienta OBS-Studio, en el cual se pretende mostrar la estructura del proyecto en cuanto a carpetas, archivos y dependencias (dependiendo el caso). El video explicará de forma general y concisa todo el proceso de desarrollo del proyecto, las pruebas y al final se ejecutará, probando así, que el código es funcional. Este video estará disponible en YouTube y se podrá consultar dentro del documento presente en la sección de Estructura del proyecto.

**Estructura del proyecto**

* Carpeta madre

Incluye todos los archivos del proyecto.

* Carpeta archivo de entrega

Incluye el archivo presente que corresponde al entregable del proyecto.

* Carpeta capturas\_prueba

Incluye las capturas hechas por el programa como parte de las pruebas automatizadas.

* Carpeta descargas

Incluye el archivo descargado por el programa como parte de las pruebas automatizadas.

* Carpeta utils

Incluye un archivo genérico para hacer la carga en el sitio web.

* Carpeta venv

Esta carpeta es opcional dependiendo el sistema operativo. En este caso, el sistema operativo es Linux, por ende, es necesaria. Contiene todos los archivos de un entorno virtual para ejecutar el código con Python y Selenium.

* Archivo main.py

Contiene todo el código de todas las pruebas que se realizaron.

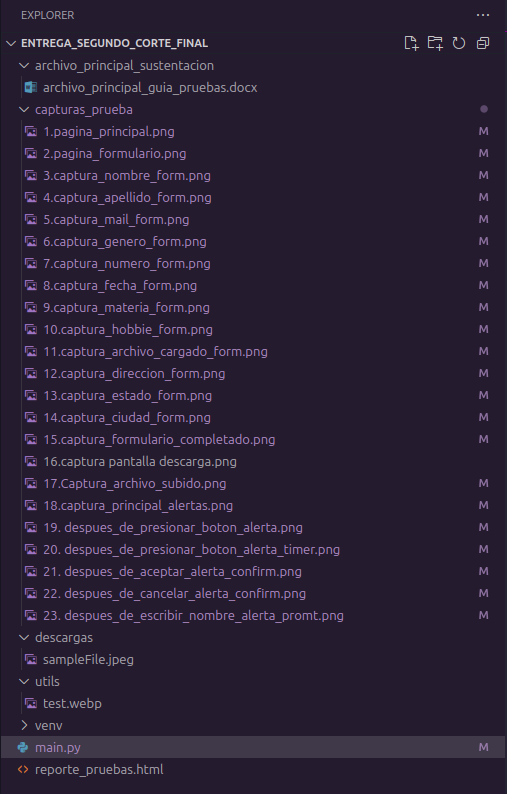
* Archivo reporte\_pruebas.html

Contiene el informe HTML de las pruebas.

## **Rutas para su verificación**

* [Repositorio de GitHub](https://github.com/KaonyKampus/proyecto_final_2docorte_pruebas_de_software)
* [Video de explicación subido a YouTube](https://youtu.be/xzgbf5ejwHY)
* [Documentación de proceso para Linux](https://fair-fish-419.notion.site/Gu-a-de-instalaci-n-de-entorno-de-pruebas-automatizadas-Selenium-Python-para-Linux-Spanish-1d7cfa5b0c84807ea73ef55ea3fdbb32)
* [Documentación de proceso para Windows](https://fair-fish-419.notion.site/Gu-a-de-instalaci-n-de-entorno-de-pruebas-automatizadas-Selenium-Python-para-Windows-Spanish-1d7cfa5b0c8480888e80c882493e58d7)

## **Estructura gráfica del proyecto**



# **Herramientas utilizadas para todo el proyecto**

* Python

Se utilizó Python como lenguaje de programación para hacer todo el proceso de codificación.

* Selenium

Se utilizó Selenium como librería de Python para hacer las pruebas automatizadas.

* Word-web

Se utilizó Word desde el paquete de 365 office, licencia proporcionada por la U Compensar, para hacer la estructura del documento entregable.

* OBS-Studio

Se utilizó OBS-Studio para realizar la grabación de pantalla y realizar el video de sustentación del proyecto.

* Notion

Se utilizó Notion como herramienta de documentación para el proceso de instalación y ejecución de Python y Selenium con un ejemplo básico de pruebas automatizadas. Este proceso se hizo como guía para realizarlo en dos sistemas operativos; Windows y Linux.

* VSCODE

Se utilizó VSCODE como editor de texto por su versatilidad. Toda la parte de codificación fue escrita y editada con esta herramienta.

## **Fuentes de información y recursos adicionales**

* YouTube.com

Se consultaron varios videos de YouTube para conocer y reforzar detalles sobre las pruebas automatizadas con Python y Selenium.

* Ket-Code.com

Se consultaron los Blogs proporcionados por el docente durante las clases, para reforzar información y procesos.

* Roadmap.sh

Se consultó Roadmap.sh, página de consulta que también funciona como herramienta, para tener en cuenta cuál es la ruta de un QA y que otros procesos existen dentro de las pruebas de software.

* Readthedocs.io

Se consultó Readthedocs.io como sitio de documentación no oficial de Selenium por recomendación en internet, ya que explica de forma sencilla el uso de Selenium y sus principales funciones.

* Browserstack.com

Se consultó Browserstack.com como fuente de información para conocer la importancia de Selenium en las pruebas automatizadas, y así mismo conocer porque en general son importantes las pruebas automatizadas.

# **Conclusiones**

A continuación, se presentan las conclusiones generales de todo el proyecto de pruebas automatizadas, que se da gracias a la clase de Pruebas de Software I. Estas conclusiones presentan una visión general enfocada a la utilidad de este tipo de proyectos y en que benefician a los estudiantes dentro del aula.

* Este proyecto reforzó los conocimientos adquiridos en el aula, ya que, al ser un proyecto tipo laboratorio, permitió tener una interacción con las pruebas de software orientadas a como se harían en la vida real, por lo cual permitió hacer la transición del conocimiento que se había adquirido anteriormente.
* A través del proyecto, que se enfoca en pruebas automatizadas, no solo se reforzaron conocimientos en este campo, sino que también involucró el uso de herramientas adicionales y así la rama de conocimiento se expandió considerablemente.
* La experiencia de trabajar en dos sistemas operativos (Windows y Linux) permitió comprender las particularidades de cada sistema en cuanto a rutas, permisos, configuraciones, y demostró la capacidad multiplataforma de Selenium y Python.
* Se reforzó también la habilidad en programación con un lenguaje de programación nuevo enfocado, en este caso, a las pruebas automatizadas.
* Se confirma que este tipo de proyectos son importantes en la formación académica, especialmente en programas arraigados a la tecnología donde la práctica es fundamental.

# **Fuentes**

* [https://browserstack.com/selenium](https://www.browserstack.com/selenium)
* <https://ket-code.blogspot.com/>
* <https://roadmap.sh/>
* <https://selenium-python.readthedocs.io/>
* [https://selenium.dev/documentation/](https://www.selenium.dev/documentation/)
* [https://youtube.com/watch?v=r7XJ1sxRNuA&t=465s](https://www.youtube.com/watch?v=r7XJ1sxRNuA&t=465s)