



# PROYECTO FINAL - TESTING QA

Análisis de calidad a eCommerce de **Peppos**

Alumno: **Camilo Ferreira**  
Comisión: 23320

Docente: **Cristian Ciarallo**  
Tutor: **Romina Oliva**

# Índice

<b>1. Descripción general del proyecto</b>	<b>2</b>
<b>2. Objetivo y alcance del proyecto</b>	<b>2</b>
<b>3. Casos de prueba</b>	<b>2</b>
3.1 Documentación de casos de prueba	2
3.2 Parámetros y características de casos de prueba	2
3.3 Técnicas empleadas para creación de casos de prueba	2
<b>4. Defectos</b>	<b>3</b>
4.1 Documentación defectos	3
<b>5. Jira</b>	<b>3</b>
<b>6. Postman</b>	<b>3</b>
<b>7. Selenium</b>	<b>3</b>
<b>8. Enlaces generales</b>	<b>4</b>

# 1. Descripción general del proyecto

Se crea un proyecto de testing para cumplir con el proyecto final del curso de Testing de Coderhouse. Se seleccionó como web a probar el eCommerce de Peppos, <https://www.peppos.com.uy/>

## 2. Objetivo y alcance del proyecto

Primeramente se busca definir casos de prueba a través de la exploración de la página así conociendo sus funcionalidades más representativas.

El proyecto incluirá testing funcional manual, automatizado incluyendo dos variantes (postman y selenium) y como herramienta adicional de gerencia de proyecto Jira (cobertura parcial).

Cabe destacar que las pruebas realizadas no cubren la finalización de ordenes de compra, el contenido interno referente a preguntas frecuentes, algunas funcionalidades de usuarios logueados y registro de usuario ligado a cuentas de redes sociales. Así mismo cubrirá únicamente la versión de la web estilo escritorio, es decir quedando afuera mobile app y web versión mobile.

## 3. Casos de prueba

### 3.1 Documentación de casos de prueba

Se ha elegido como medio para almacenar casos de prueba una planilla de Google. Así mismo la misma cuenta con todas las columnas necesarias para el registro y ejecución de los mismos. Cabe destacar que la planilla cuenta con una pestaña de carácter general para almacenar comentarios de su versionado, glosario para términos propios implementados, clases de equivalencia

### 3.2 Parámetros y características de casos de prueba

Se podrá ver en la planilla que se cuenta en la pestaña de “Casos y Ejecuciones” con columnas para caracterizar a los casos de prueba. Se mencionan de los más importantes.

- Prioridad: Define la importancia que tiene en la suite de casos de pruebas
- Tipo de caso: Define el tipo de flujo al que corresponde el resultado esperado de la prueba
- Versión Web: Se refiere a que versión corresponde el caso de prueba. Ya que la página en cuestión al ser accedida desde un dispositivo móvil se modifican algunos elementos, a excepción de activar la opción de ver como en escritorio.

### 3.3 Técnicas empleadas para creación de casos de prueba

Cabe resaltar entre otras técnicas empleadas para la creación de casos de prueba la creación de clases de equivalencias y posteriormente la aplicación de Pairwise.

De esta manera se asegura la cobertura óptima de las variantes para el formulario de registro de cuentas en la web.

## 4. Defectos

### 4.1 Documentación defectos

Se ha elegido como medio para almacenar los defectos encontrados una planilla de Google. Así mismo la misma cuenta con todas las columnas necesarias para el registro de los mismos.

## 5. Jira

Se crea un proyecto de práctica de Jira incluyendo

- Creación de sprints
- Creación de issue types customizados (test case y test run)
- Creación de tareas en el backlog de tipo test case issue y task issue
- Creación de Epics en base a funcionalidades presentadas en la planilla de Google

Se incluye video de demostración breve de lo realizado en Jira en los siguientes enlaces

- ESPAÑOL -> <https://youtu.be/akY0fK5jdFg>
- INGLÉS -> <https://youtu.be/aOzGVAyobmY>

## 6. Postman

Se implementa Postman para cubrir algunos casos de pruebas y algunos escenarios extras de manera de optimizar futuras ejecuciones de pruebas. Para acceder a lo implementado en postman acceder al [enlace workspace](#), en éste se encuentra descripción adicional de lo realizado.

A resaltar encontramos los siguientes puntos:

- Creación de colecciones en base a funcionalidad
- Creación de request y test para las mismas
- Creación de monitor
- Creación de ambientes a modo didáctico

## 7. Selenium

En referencia a lo realizado en Selenium (en código) encontramos la aplicación de lo siguiente

- Patrón Page Object Model
- TestNG
- PageFactory
- JAVA

Para acceder al mismo ir al repositorio [PepposCoderHouse](#)

## 8. Enlaces generales

A continuación se presentan de manera conjunta los enlaces más importantes del proyecto

- Evidencia de ejecución de prueba
  - [https://drive.google.com/drive/folders/16RfRu\\_no9cwV9Qa9LJ8JfNtkgmKlpVlw?usp=sharing](https://drive.google.com/drive/folders/16RfRu_no9cwV9Qa9LJ8JfNtkgmKlpVlw?usp=sharing)
- Evidencia de defectos encontrados
  - [https://drive.google.com/drive/folders/1m6q-0\\_bR5KqgSyu8MXr21K-glzi5LyIP?usp=sharing](https://drive.google.com/drive/folders/1m6q-0_bR5KqgSyu8MXr21K-glzi5LyIP?usp=sharing)
- Test cases, reporte de defectos y relacionado
  - [https://drive.google.com/drive/folders/1S0z5dPz\\_p7URFeSpnf7k\\_7kbH1TRtTy?usp=sharing](https://drive.google.com/drive/folders/1S0z5dPz_p7URFeSpnf7k_7kbH1TRtTy?usp=sharing)
- Repositorio de solución creada de Selenium
  - <https://github.com/ferreiracamilo/PepposCoderhouse>
- Solución creada de Postman
  - <https://www.postman.com/cferreira89/workspace/pepposcoderhouseapi-tests/overview>
- Implementación de JIRA (video de demostración)
  - ESPAÑOL -> <https://youtu.be/akY0fK5jdFg>
  - INGLÉS -> <https://youtu.be/aOzGVAyobmY>