



Proyecto Crazy Vegan

c11-01-m-python

Plan de Pruebas

Versión: 1.0 Draft
Mayo 12, 2023

Contenido

| | |
|---|----------|
| INTRODUCCIÓN | 4 |
| 1. ESTRATEGIA DE TESTING | 5 |
| 1.1. ALCANCE | 5 |
| 1.2. ENFOQUES DE LAS PRUEBAS | 6 |
| 1.2.1. REQUERIMIENTOS FUNCIONALES | 6 |
| 1.2.2. REQUERIMIENTOS NO FUNCIONALES | 6 |
| 1.3. AMBIENTES DE TESTING | 6 |
| 1.3.1. GESTIÓN DE AMBIENTES | 6 |
| 1.3.2. HERRAMIENTAS DE TESTING | 6 |
| 2. RECURSOS | 7 |
| 3. TRACKING Y REPORTE | 7 |
| 3.1. HERRAMIENTAS DE TRACKING | 7 |
| 3.2. ESTADO DE LAS PRUEBAS | 7 |
| 4. NOMENCLATURA CASOS DE PRUEBAS | 8 |
| 5. PROCEDIMIENTO PARA REPORTE DE DEFECTOS..... | 8 |

Historial

| Versión | Descripción | Responsable | Fecha |
|----------------|---|------------------|----------|
| 1.0.0 Draft | Descripción de la estrategia de testing de la implementación del sitio web Crazy Vegan para No Country. | Llorente Gabriel | 12/05/23 |
| 1.0.1 Baseline | | | |
| 1.1 Aprobado | | | |



INTRODUCCIÓN

Este documento define la estrategia de las pruebas que se realizarán para asegurar la calidad de los diferentes releases a implementarse del sitio web Crazy Vegan para No Country, en referencia a los requerimientos especificados en la herramienta de Trello y Jira.

Este documento incluye material del proceso de desarrollo de las pruebas, recursos y los requerimientos acordados que serán probados en las diferentes etapas de implementación de manera incremental, y el plan de acción de las pruebas.

1. ESTRATEGIA DE TESTING

1.1. ALCANCE

Este proyecto está basado en la implementación del sitio web Crazy Vegan para No Country, el mismo será implementado de manera incremental en 4 Releases en las próximas semanas.

El **Release 1** a implementarse, incluye los siguientes módulos:

- **Perfil:** Registro básico
- **Compras:** Agregar carrito
- **Ventas:** Formulario de compra básico
- **Mis anuncios:** Ver anuncios que he publicado
- **Búsqueda:** Función de buscar artículo
- **Visualización:** Se observe productos en línea

El **Release 2** a implementarse, incluye los siguientes módulos:

- **Perfil:** Registro con e-mail
- **Compras:** Hacer compras en efectivo.
- **Ventas:** Formulario de venta detallada
- **Mis anuncios:** Editar anuncios
- **Búsqueda:** Ordenar artículo por categoría
- **Visualización:** Se observe productos en cuadrícula

El **Release 3** a implementarse, incluye los siguientes módulos:

- **Perfil:** Registro con ubicación
- **Compras:** Hacer compras con tarjetas de crédito
- **Ventas:** Historial de ventas
- **Mis anuncios:** agregar productos de venta
- **Búsqueda:** Ordenar artículo por precio
- **Visualización:** Página de factura y gracias por la compra.

El **Release 4** a implementarse, incluye los siguientes módulos:

- **Perfil:** Registro con foto
- **Compras:** Hacer compras con medios digitales
- **Ventas:** Edición de datos de dirección de envío
- **Mis anuncios:** Editar precios
- **Búsqueda:** buscar productos por más vendidos

1.2. ENFOQUES DE LAS PRUEBAS

A continuación se detallan los diferentes tipos y niveles de pruebas que se ejecutarán a lo largo de toda la etapa de pruebas del proyecto, dependiendo si son requerimientos funcionales o no funcionales:

1.2.1. REQUERIMIENTOS FUNCIONALES

- Testing Funcional: pruebas que se ejecutarán para probar los componentes del sistema definidos y aprobados en las especificaciones del diseño.
- Testing de Integración: pruebas que se ejecutarán para verificar que los módulos implementados funcionen correctamente al integrarse.
- Testing de Regresión: pruebas que se ejecutarán luego de implementando un release y se hayan detectado errores en el mismo y luego en el próximo release sean implementado dichos fixes, en tal caso, correremos casos de pruebas de regresión para verificar que lo que antes funcionaba bien continúe haciéndolo.

Nota: todas las pruebas del tipo funcionales que se ejecutarán serán del tipo manuales.

1.2.2. REQUERIMIENTOS NO FUNCIONALES

- Testing de Rendimiento/Performance: pruebas que se ejecutarán para evaluar el rendimiento/performance del sitio web.

1.3. AMBIENTES DE TESTING

1.3.1. GESTION DE AMBIENTES

El ambiente definido para la ejecución de los diferentes casos de pruebas es el siguiente:

- DEV: ambiente establecido para las pruebas unitarias de Desarrollo y Funcionales del equipo de QA.

1.3.2. HERRAMIENTAS DE TESTING

Las siguientes herramientas son las definidas para la creación y ejecución de los casos de pruebas manuales y de performance:

- Jira: herramienta para creación y ejecución de casos de pruebas
- GTmetrix: herramienta de analítica de rendimiento que permite aplicar test de velocidad web en poco tiempo.

2. RECURSOS

Los recursos identificados para las etapas de pruebas y su seguimiento son los siguientes:

- QA Manual: responsable de la gestión de pruebas, planificación y diseño de los casos de pruebas. Análisis de los errores detectados y derivar a los desarrolladores para su corrección.
- Desarrolladores: responsables de corregir los errores detectados en etapas de pruebas.
- Líder de Proyecto: responsable de seguimiento del proyecto.

3. TRACKING Y REPORTES

3.1. HERRAMIENTAS DE TRACKING

Las herramientas definidas para el seguimiento de los resultados de las ejecuciones de los casos de pruebas, los errores detectados durante dichas ejecuciones, las especificaciones de los requerimientos, y reportes son las siguientes:

- Trello y Jira: Gestión de Incidencias; como ser: historias de usuarios, mejoras, tareas y defectos.
- Jira: Gestión de Testing.

3.2. ESTADO DE LAS PRUEBAS

El resultado de las pruebas se deberá clasificar en la herramienta de casos de la siguiente manera:

- Paso: Resultado exitoso.
- Fallo: Resultado erróneo.
- Bloqueado: No se pudo ejecutar el caso de prueba por algún otro error en el sitio o por falta de alguna implementación.

4. NOMENCLATURA CASOS DE PRUEBAS

La Nomenclatura definida para la creación de los casos de pruebas se define a continuación:

- Nombre Identificador del Caso de Prueba, el mismo debe respetar la siguiente nomenclatura:
TC_[Módulo Funcional que se testeará]_[Título del Caso de Prueba, el mismo debe contener una breve descripción que identifique que se está probando]_[0000: numeración cronológica incremental]
Ejemplo: TC_Login_Registro de usuario_0001

5. PROCEDIMIENTO PARA REPORTE DE DEFECTOS

Todos los defectos detectados en la etapa de pruebas deberán ser registrados en Trello.

Los QA serán los responsables de crear los defectos en Trello y derivar el mismo a la gente de Desarrollo de c11-01-m-python.

La información a completarse en Trello al momento que se encuentre un defecto es la siguiente:

- ID del Defecto
- Título del Defecto: breve descripción del defecto encontrado
- Versión/Found In: especificar en qué versión fue detectado el defecto
- Evidencia: adjuntar en el ticket de Trello toda información que sea necesaria para el Desarrollador como por ejemplo capturas de imágenes o videos.
- Descripción del Defecto: agregar una explicación detallada del error encontrado, agregando la siguiente información:
 - * Descripción de la falla
 - * Pasos para reproducir el error
 - * Asociar el Caso de Prueba, con el cual se detectó el defecto



- * Ambiente en el cual fue detectado el mismo; ejemplo: DEV
- * Resultado Esperado
- * Resultado Obtenido