**Segundo Informe | Equipo QA | Proyecto Reffindr**

**Responsable:** Carlos Daniel Jiménez Cálcena, Líder de QA

Equipo: Maria Alejandra Atanasios – Mónica Mondino – Carlos Daniel Jiménez Cálcena  
**Fecha:** 22-11-2024

En esta semana completamos el Sprint 2, en el cual nos abocamos a definir cuáles serían los modelos de pruebas a realizar de acuerdo a cada historia de usuario, así como también, preparar los entornos de prueba para garantizar la fluidez del trabajo y ya poder empezar a tomar las issues del GitHub Projects y empezar con la ejecución de pruebas primarias, tanto funcionales como no funcionales, de seguridad, de carga, de estrés y de respuesta del servidor.

Este documento contiene la redacción de casos de prueba funcionales basados en las historias de usuario y los criterios de aceptación definidos, así como casos de prueba adicionales diseñados para validar entradas incorrectas, valores límite y comportamientos inesperados. Se emplearon técnicas de diseño de pruebas de caja negra como partición de equivalencia y análisis de valores límite, de acuerdo a lo solicitado en el documento entregable a Igrowker.

Ejemplos de Casos de pruebas primarios a realizar:

**1. Casos de Prueba Funcionales**

| **ID** | **Título** | **Precondiciones** | **Pasos para Reproducir** | **Datos de Entrada** | **Resultado Esperado** |
| --- | --- | --- | --- | --- | --- |
| **TC-001** | Validar que un usuario puede registrarse | El sistema debe estar accesible y el endpoint /register debe estar disponible. | 1. Navegar al formulario de registro. 2. Completar los campos obligatorios con datos válidos. 3. Enviar el formulario. | Nombre: "Juan" Email: "juan@email.com" Contraseña: "Pass1234!" | El sistema responde con un código 201 y un mensaje: "Registro exitoso". |
| **TC-002** | Validar que un usuario puede iniciar sesión | El sistema debe tener un usuario registrado con email y contraseña válidos. | 1. Navegar al formulario de inicio de sesión. 2. Ingresar el email y contraseña registrados. 3. Hacer clic en "Iniciar sesión". | Email: "juan@email.com" Contraseña: "Pass1234!" | El usuario es autenticado y redirigido al panel principal. |
| **TC-003** | Validar publicación de una propiedad | El usuario debe estar autenticado como inquilino saliente. | 1. Acceder al formulario "Publicar Propiedad". 2. Completar los datos obligatorios. 3. Enviar la solicitud de publicación. | Dirección: "Calle Falsa 123" Precio: 500 Requisitos: "Ingresos >1000" | La propiedad aparece listada con los datos proporcionados. |
| **TC-004** | Validar postulación a una propiedad | El usuario debe estar autenticado como inquilino entrante y debe cumplir con los requisitos. | 1. Seleccionar una propiedad disponible. 2. Hacer clic en "Postularme". 3. Confirmar los datos de postulación. | ID Propiedad: 123 Usuario: "Pedro" | El sistema confirma la postulación y notifica al inquilino saliente. |

**2. Casos de Prueba para Entradas Inválidas y Valores Límite**

**2.1. Partición de Equivalencia**

| **ID** | **Título** | **Precondiciones** | **Datos de Entrada** | **Resultado Esperado** |
| --- | --- | --- | --- | --- |
| **TC-005** | Validar error por email inválido | El sistema debe validar el formato del email. | Email: "juan.com" Contraseña: "Pass1234" | El sistema responde con un mensaje de error: "Formato de email no válido". |
| **TC-006** | Validar error por contraseña débil | El sistema debe tener políticas de seguridad configuradas para contraseñas. | Email: "juan@email.com" Contraseña: "123" | El sistema muestra un mensaje de error: "Contraseña demasiado débil". |

**2.2. Análisis de Valores Límite**

| **ID** | **Título** | **Precondiciones** | **Datos de Entrada** | **Resultado Esperado** |
| --- | --- | --- | --- | --- |
| **TC-007** | Validar límite inferior en el precio de propiedad | El formulario debe aceptar precios >= 100. | Precio: 99 | El sistema muestra un mensaje de error: "El precio debe ser mayor o igual a 100". |
| **TC-008** | Validar límite superior en el precio de propiedad | El formulario debe aceptar precios <= 5000. | Precio: 5001 | El sistema muestra un mensaje de error: "El precio no debe superar los 5000". |

**3. Organización y Formato de Entrega**

**Estructura del Documento**

* **Formato:** Hoja de cálculo en Google Sheets o Excel.
* **Columnas Incluidas:**
  + ID
  + Título
  + Historia de Usuario
  + Tipo de Prueba (Funcional, Negativa, Límite, etc.)
  + Precondiciones
  + Pasos para Reproducir
  + Datos de Entrada
  + Resultado Esperado

**Ejemplo en Tabla:**

| **ID** | **Título** | **Historia de Usuario** | **Tipo de Prueba** | **Precondiciones** | **Pasos para Reproducir** | **Datos de Entrada** | **Resultado Esperado** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC-001 | Validar que un usuario puede registrarse | "Un usuario puede registrarse." | Funcional | El sistema debe estar accesible y el endpoint /register debe estar disponible. | 1. Navegar al formulario de registro. 2. Completar los campos obligatorios con datos válidos. 3. Enviar el formulario. | Nombre: "Juan" Email: "juan@email.com" Contraseña: "Pass1234!" | El sistema responde con un código 201 y un mensaje: "Registro exitoso". |

En conclusión, por el momento venimos avanzando según lo planeado en cuanto al diseño y planificación de pruebas. Ya tenemos preparadas las herramientas a utilizar, como navegadores y simuladores para pruebas manuales y herramientas de consultas de API como Postman, Jmeter, Cypress y Selenium para ejecutar algunos scripts de automatizaciones.

Por ahora lo único que necesitamos es una reunión con Back-End y Front-End, para que nos muestren el paso a paso para levantar los servidores y tener listo el ambiente de pruebas, entiendo que se está por utilizar Docker, y no se si es necesario que nosotros también lo configuremos para ejecutar las pruebas, sería la única “duda/bloqueante” por el momento.