**Prueba QA**

***Johaynes Caceres***

**HISTORIA DE USUARIO: CITAS VIRTUALES PARA ESTUDIANTES**

***Parte 1 Diseño de casos de prueba***

**Casos de prueba**

1 Validar inicio de sesión estudiante correcto

2 Validar inicio de sesión estudiante incorrecto por ingresar correo inválido

3 Validar inicio de sesión estudiante incorrecto por ingresar contraseña invalida

4 Verificar botón tipo calendario

5 Verificar botón tipo hora

6 Validar que exista botón agendar cita

7 Validar funcionalidad botón agendar cita

8 Comprobar que la cita tiene un tiempo fijo de 20 minutos

9 Validar que días fuera del horario aparecen deshabilitadas

10 Validar que horas fuera del horario aparecen deshabilitadas

11 Validar que las horas vayan de 20 en 20, es decir, 8 - 8:20am, 8:20am - 8:40am

12 Validar que se deshabilite un horario agendado, es decir, si el estudiante reservó de 8 a 8:20 luego de confirmar la cita se debe deshabilitar ese horario

13 Confirmar que el correo electrónico se recibe correctamente como confirmación

14 Evaluar ortografía

15 Evaluar tipo de letra

16 Evaluar tamaño de letra

17 Evaluar colores de los botones

***Parte 2 Análisis de un bug reportado***

Bug 1: Validación incorrecta de superposición de citas

Se valida que el sistema no realiza correctamente la comprobación de solapamiento de las citas en la base de datos antes de confirmar una nueva reserva. Esto permite que varias citas se agenden en el mismo horario, a pesar de que cada cita dura 20 minutos y debería ser exclusiva en ese intervalo.

Pasos para reproducir:

1. Accede a la plataforma como estudiante y abre la sección de agendamiento.
2. Intenta crear una cita en un horario específico, por ejemplo, 10:00 AM a 10:20 AM, y confirma.
3. Cierre sesión e ingresa con nuevo estudiante, intenta agendar otra cita en el mismo horario (10:00 AM a 10:20 AM).
4. Observa que la segunda cita se confirma y aparece en el calendario, solapándose con la primera.

Causa probable:

La función que evalúa si el horario está disponible no está considerando correctamente los rangos de tiempo al verificar la existencia de citas en la base de datos. O bien, la condición lógica para impedir solapamientos está mal implementada.

Soporte:

Se deja pantallazos /videos del funcionamiento incorrecto y el paso a paso.

***Nota:*** No sería necesario que el Student Manager reporte el bug ya que para eso son las pruebas exhaustivas y se debe encontrar el bug ejecutando el caso de prueba número 12 "Validar que se deshabilite un horario agendado, es decir, si el estudiante reservó de 8 a 8:20 luego de confirmar la cita se debe deshabilitar ese horario", pero en caso que haya sucedido de tal manera que no existió un caso de prueba que probara el escenario del bug encontrado lo manejaría de la misma forma, replicaría el bug y reportaría a desarrollo para ser corregido.

***Parte 3 Propuesta de mejoras de procesos de QA***

**Historias de usuario claras**: Para asegurar la calidad del desarrollo es de suma importancia que las necesidades queden bien documentadas para tener la información lo más detallada y clara posible.

**Pruebas de regresión**: Asegurarse de que los cambios no afecten la funcionalidad que se ejecutaba correctamente.

**Implementar un sistema de validación automática de conflictos horarios**: Esto garantizará que las citas no puedan ser agendadas en horarios ya ocupados, evitando solapamientos y reduciendo errores manuales. Mejorará la confiabilidad y usabilidad del sistema, ahorrando tiempo a gestores y estudiantes.

**Agregar mensajes de advertencia y confirmación en el proceso de agendamiento: I**nformar en tiempo real si un horario ya está reservado ayuda a los usuarios a evitar intentos fallidos. Esto mejora la experiencia del usuario, haciendo la plataforma más intuitiva y transparente.

**Identificar siempre causa raíz de los errores:** Analizar a fondo que causó una falla en el desarrollo y evitar cometer mismos errores para próximas historia.

***Parte 4 Redacción y atención al detalle***

***Correo:***

***Destinatario:*** *Product Owner*

***Asunto:*** *Pruebas terminadas*

***Descripción:*** Se realizaron las pruebas correspondientes y se encontró un bug que permite que dos estudiantes agenden una cita en el mismo horario.  
Propuesta siguiente paso: Agendar una llamada con el desarrollador para darle más detalle sobre el bug, a pesar de las evidencias si es necesario replicarlo juntos y él lo pueda tener claro para chequear posibles fallas.

***Parte 5 Automatización conceptual***

Aunque existen muchas herramientas utilizaría Selenium y Java, ya que tienen mucho tiempo en el mercado y es más accesible tener acceso a información para implementar pruebas automatizadas. A diferencia de algunas herramientas Selenium permite pruebas en Java, JavaScript, C#, Ruby, Python, lo cual sería más conveniente si el equipo de desarrollo trabaja con distintos lenguajes.

***Parte 6 Uso de herramientas de inteligencia artificial de QA***

Realmente no he utilizado ninguna herramienta de inteligencia artificial para QA, sin embargo debido a esta pregunta me dediqué a estudiar un poco el tema. Puedo deducir que la IA puede analizar requerimientos escritos en lenguaje natural o código para generar casos de prueba automáticamente.

Ejemplo 1: Herramientas como Functionize o incluso ChatGPT pueden transformar una historia de usuario en un conjunto de pruebas funcionales, de lo cual nos podemos apoyar (siempre utilizando nuestro análisis crítico como humano real que usaría una plataforma) para ampliar tal vez nuestros posibles escenarios.

Ejemplo 2: La IA puede detectar errores de diseño o visuales de una manera más efectiva que nosotros mismos.

Puedo aportar respecto a este tema que la IA no reemplaza el rol de un QA pero puede ser una herramienta de gran ayuda para crear casos de prueba de una manera más amplia lo cual llevaría a mejorar nuestro objetivo que es garantizar que una plataforma esté apta para ser utilizada perfectamente.