**RETO 1**

**Historia de usuario**

Como usuario nuevo deseo poder registrarme en la web y poder loguearme de manera satisfactoria

**Criterios de aceptación**

Los criterios de aceptación para el registro son los siguientes:

- El correo ingresado debe tener un @ para ser válido.

- La contraseña debe tener como mínimo 5 caracteres.

- El correo y la contraseña son requeridos obligatoriamente.

- El sistema debe mostrar un mensaje de error al ingresar campos no válidos.

Los criterios de aceptación para el login son los siguientes:

- El correo ingresado debe tener un @ para ser válido.

- La contraseña debe tener como mínimo 5 caracteres.

- El correo y la contraseña son requeridos obligatoriamente.

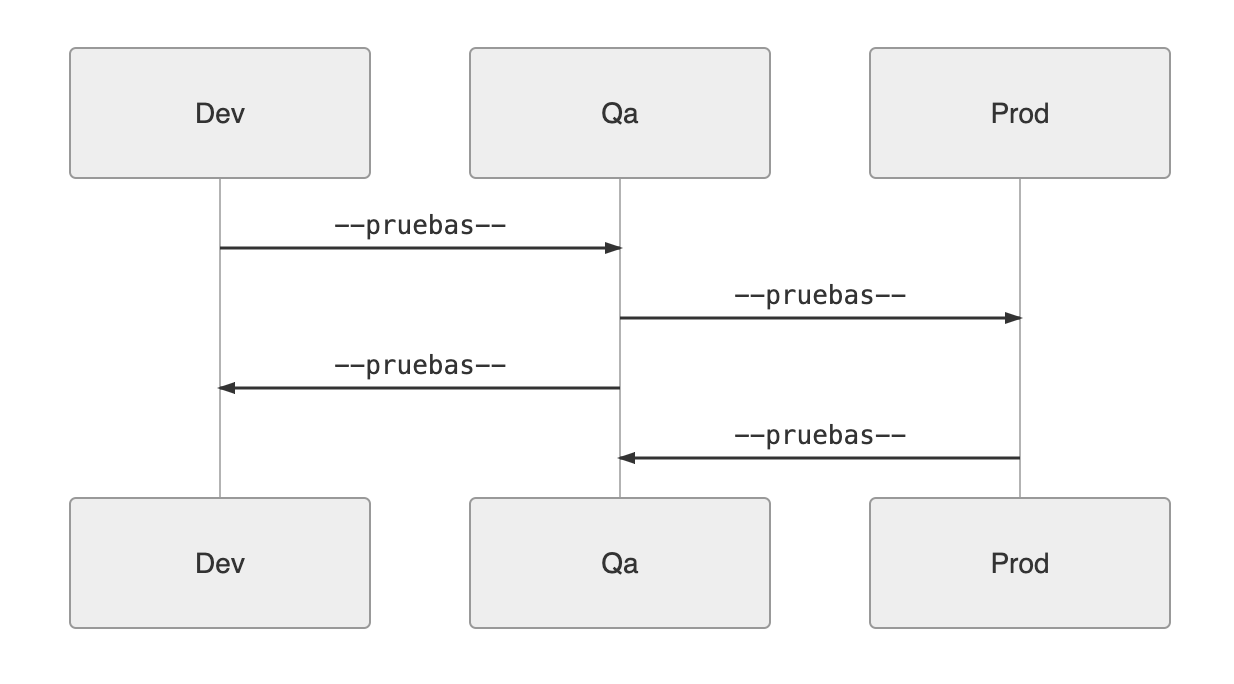
- El sistema debe mostrar un mensaje de error al ingresar campos no válidos.

El equipo de desarrollo realizo sus tareas y ahora te toca a ti certificar que dichas actividades se realizaron correctamente 🙌.

\*\* Todos los requerimientos deben ser considerados tanto para para parte front como para la parte back del sistema.

1. Identifica los tipos de casos de prueba que estarían enlazadas a las requerimientos propuestos.
   * Pruebas funcionales por cada criterio de aceptación
   * Pruebas relacionadas al cambio (retest y regresión)
2. Menciona que técnica de diseño de casos de prueba usarías para poder generar dichos casos
   * Ya que se tiene acceso al codigo, parte de la arquitectura y demás podría ir por tecnicas de caja blanca.
3. Identifica, registra y describe los bugs o mejoras que consideres necesarios .
   * El proyecto presentaba errores al momento de levantarlos, no hace la validación de campos.

**RETO 2**



1. Comenta en base a tu criterio que nivel(es) de pruebas aplicaría para cada ambiente (puedes sugerir otros ambientes u estructura si es que lo consideras) (Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx ) )
   * Pruebas unitarias (según mi criterio lo debería hacer el DEV, y QA debe asegurarse de que tenga cierta cobertura de pruebas a nivel unitario)
   * Pruebas de integración y APIs, de ser el caso que esté separado en servicios o microservicios o que algun middlewere o cliente esté interactuando con los sistemas internos.
   * Pruebas de interfaz, verificando que se hacen las validaciones respectivas por la parte front, filtro basico de seguridad en caso de input de datos, recomendaciones de usabilidad, diseño, etc.

**RETO 3**

**Priorización de errores**

**Caso:**

Imagina una herramienta web, desarrollada con el objetivo de proporcionar una versión de ancho de banda ligero (se requiere un uso óptimo alrededor de 50 MB) de una herramienta principalmente utilizada por personas de todo el mundo para comprar productos de segunda mano.

Defectos:

1. Los precios no varían cuando una transacción se realiza en diferentes monedas.
2. Las transacciones entre usuarios no se reflejan en un informe de back-end.
3. Hay algunos errores ortográficos en la página de inicio.
4. Las credenciales de usuario no están cifradas en la base de datos.
5. Los foros oficiales de soporte no siempre registran los comentarios del usuario.
6. El ancho de banda necesario para una experiencia de usuario óptima es una velocidad de descarga de 100 MB.

**Objetivos**

Para el siguiente conjunto de defectos, prioriza del 1 al 6, siendo el primero el más crítico y explique brevemente por qué (puedes incluir alguna suposición). Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx )

Defectos priorizados por criticidad sugerida:

1. Las credenciales de usuario no están cifradas en la base de datos. – En algun momento X puede llegar una auditoría, se podría filtrar información y podría llegar una demanda grave, es un asunto de seguridad basico encriptar los credenciales de los usuarios.
2. Los precios no varían cuando una transacción se realiza en diferentes monedas. – Esto afecta economicamente a la empresa sin embargo, podría no ser un criterio definido previamente, PERO debe hacerse notar la alerta para que se tomen consideraciones al respecto.
3. Hay algunos errores ortográficos en la página de inicio. – No considero critico pero si medianamente importante, ya que es muy poco usual y de rapida corrección.
4. Las transacciones entre usuarios no se reflejan en un informe de back-end. – Es de suma importancia tener un registro tipo log del recorrido que hace el usuario para detectar mas rapido errores y solucionarlos o considerarlos como funcionalidad de ser el caso.
5. El ancho de banda necesario para una experiencia de usuario óptima es una velocidad de descarga de 100 MB. – En cuanto a performance y experiencia de usuario es importante, mas el sistema podría andar sin problemas ya que es un incidente que puede resolverse en el camino.
6. Los foros oficiales de soporte no siempre registran los comentarios del usuario. – Es importante recibir los comentarios de los usuarios pero si todo lo anterior está fallando lo mas probable es que los comentarios sean basicamente sobre ello, por eso creería que estabilizar el producto sería un principal criterio para solucionar y luego ya habilitar esta sección.

**RETO 4**

**Agile testing**

**Caso:**

La aplicación web a probar es una plataforma como Mercado Libre. A continuación, podrás encontrar una lista de técnicas y procedimientos utilizados en las pruebas

**Objetivos**

Evalúa si son los procedimientos de la lista son los más adecuados. Brinda una alternativa / tipo de prueba o mejora para las que no son adecuadas desde tu punto de vista. (Coloca tus respuestas en el documento ( qa-challenge-reto[nombre].docx )

El equipo de pruebas :

1. Utiliza el mismo entorno que los usuarios finales para realizar las pruebas perfomance.
   * Sugiero homologar un ambiente de pruebas con las características que tiene el que usan los usuarios finales, hasta que las pruebas estén OK, una vez hecho esto, verificar en qué horario y fecha hay menor impacto o nulo de interacción con los usuarios finales, para así no bloquear el uso regular en caso de una posible caida de servicios.
2. Las pruebas de humo se ejecutan para cubrir todas las posibles opciones del usuario y se realizan en cada implementación de una nueva versión, todas las pruebas se realizan manualmente para reflejar la misma experiencia del usuario.
   * Las pruebas de humo deben ser casi como exploratorias estas si deberían ser manuales para simular el comportamiento comun de un usuario.
3. Las pruebas de regresión solo cubren los módulos que probablemente se vean afectados por los cambios realizados en la última versión.
   * Se debe tener en cuenta cuáles son los flujos mas criticos para sobre ellos asegurar que no se rompa ninguna funcionalidad critica, comunmente estos flujos deben ser automatizados, no unicamente probar aquello que ha sido implementado a menos que sean sistemas isolados y se tenga la certeza de no impacto.
4. Los casos de prueba se crean en Excel para que sean fáciles de editar. Los casos de prueba se eliminan después de la implementación de esa versión, por lo que se crean nuevos completos en cada iteración.
   * Si no se tiene un repositorio de casos de prueba, podría trabajarse en un excel sin embargo no es lo recomendable, existen aplicaciones free de escritorio donde se podría gestionar estos, por otro lado el tiempo que se invierte creando estos casos de prueba son importantes por ello se busca tener un repositorio para reusarlos cada vez, a la par sirve de una posible documentación ante la llegada de un nuevo al equipo.
5. Las pruebas comienzan a estar involucradas en el proceso de desarrollo desde la etapa inicial.
   * Es importante separar el concepto de testing y aseguramiento de la calidad, asi como tener en cuenta el objetivo del area en general y lo que se espera de calidad, prevención o detección de errores unicamente, si es prevención de errores mientras mas temprano se involucre el equipo de calidad, se podría tener menor impacto de errores al final de la cadena, por ello es recomendable incluirlo desde la llegada del requerimiento.