**Caso práctico QA**

1. *Definición de los Criterios de Aceptación de la HU.*

Criterio de aceptación: no permitir el login sin antes aceptar un check de política de privacidad.

1. *Escritura en BDD Gherkin de los casos de prueba que validen la HU.*

**Caso 1:**

Dado que un usuario se encuentra en el login

Cuando ingresa su usuario correctamente

E ingresa su contraseña correctamente

Y acepta el check de política de privacidad

Entonces ingresa al sitio web

**Caso 2:**

Dado que un usuario se encuentra en el login

Cuando ingresa su usuario correctamente

E ingresa su contraseña correctamente

Y no acepta el check de política de privacidad

Entonces no ingresa al sitio web

**Caso 3:**

Dado que un usuario se encuentra en el login

Cuando ingresa su usuario

Y acepta el check de política de privacidad

Entonces no ingresa al sitio web

**Caso 4:**

Dado que un usuario se encuentra en el login

Cuando ingresa su contraseña

Y acepta el check de política de privacidad

Entonces no ingresa al sitio web

**Caso 5:**

Dado que un usuario se encuentra en el login

Cuando acepta el check de política de privacidad

Entonces no ingresa al sitio web

1. *Teniendo en cuenta que esta HU tiene implementación tanto WEB como de IOS, Android y Backend, comenta que herramientas utilizarías para validar esta HU en cada disciplina.*

Para validar la WEB: pruebas manuales y automatizadas, además de pruebas de rendimiento.

Para validar IOS y Android: Simuladores y dispositivos reales para poder llevar a cabo la prueba. Utilizaría la herramienta Andoid Accessibility Scanner para probar la accesibilidad de la aplicación en Android.

Para validar el Backend: pruebas unitarias (utilizando Junit), pruebas de integración (utilizando postman o RestAssured)

1. *Como QA Engineer dentro de un equipo, compártenos que ceremonias y reuniones crees que existirían durante la semana que dura el Sprint.*

Las ceremonias dentro de un sprint:

1. Daily: todos los días del sprint, para pasar un status respecto a lo que estamos trabajando o lo que tenemos dudas/trabas.
2. Refinamiento: al final del sprint, para trabajar sobre las nuevas funcionalidades del sprint siguiente
3. Planning: El primer día del sprint para organizar como vamos y que vamos a trabajar en el sprint
4. Retro: al final del sprint, para evaluar que hicimos mal, que hicimos bien y que mejorar en el próximo sprint.
5. Demo: al final del sprint para mostrar que se estuvo trabajando en el sprint
6. *Como último paso, nos gustaría incorporar la validación de este caso de uso a nuestro plan de regresión automática.*

*a. ¿Qué herramientas usarías para automatizar la parte Web, IOS, Android y la parte de Backend?*

* Para automatizar web: utilizaría selenium o cypress.
* Para automatizar iOS y Android: utilizaría como herramienta appium.
* Para automatizar back: herramientas como Junit – postman – restAssured.

*b. ¿Qué herramientas de CI/CD crees que se podría usar para programar el lanzamiento de la regresión de forma planificada?*

Se pueden usar teamCity, Jenkins, Bitbucket, AzureDevOps*.*

1. *El equipo de Desarrollo nos entrega un día tarde una HU prioritaria para el Producto.*

*a. ¿Qué estrategia tomarías en caso de que veas que no puedes llegar a probarla?*

Para estas situaciones es importante previamente haber realizado un análisis de prioridad de los casos de prueba, por lo que, llegado este momento ejecutaría aquellos con casos con una prioridad mayor para asegurarnos probar aquellos casos que nos permitan asegurarnos que la funcionalidad principal esta correcta.

Adicionalmente sumaria un smoke test para poder llevar una prueba más rápida e informal que permita abarcar un poco mas todo el producto.

**Prueba exploratoria y reporte de bugs**

La primera tarea es encontrar un bug, nos da igual en que web de internet lo encuentres, lo importante es poder encontrarlo y reportarlo. No te preocupes porque internet está lleno de ellos.

Haz el reporte al nivel de detalle que consideres necesario para que el equipo de desarrollo pueda encontrarlo, debuggearlo y solucionarlo. Proporciona evidencias del fallo y resultado esperado. Clasifícalo en cuanto a prioridad, impacto y probabilidad de ocurrencia.

**REPORTE DE BUG**

Título: El sitio no loguea correctamente

Severidad: Alta (ya que, si el sitio no permite loguearme, no puedo acceder a mas nada dentro de él)

Prioridad: Alta (necesitamos que el bug se revuelva lo antes posible)

Ambiente en el que se detectó: productivo

Pasos para replicar el error:

1. Ir al Loguin
2. Ingresa usuario y contraseña correcta
3. Seleccionar el check

Resultado esperado: ingresar al sitio

Resultado observado: visualiza un mensaje de error “El usuario es incorrecto” y no logra ingresar

Datos de prueba: [user@gmail.com](mailto:user@gmail.com) – pass: 12345

Evidencia:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente