

## **Caso práctico QA evolutivo zara.com**

### **Producto**

Formas parte de un equipo de producto que desarrolla nuevas funcionalidades para un sistema de e-commerce del sector textil.

Dicha plataforma de venta online permite la venta en 10 mercados diferentes, repartidos entre EEUU,

Europa, Asia y América del Sur.

### **Equipo**

Formas parte de un equipo de uno de los equipos de producto, a los que llamamos verticales de negocio, como qa evolutivo, el cuál trabaja con metodología Scrum con sprint de una semana.

Esta vertical está compuesta por

- 1 product owner
- 1 proxy product owner: project manager
- 1 delivery manager
- 5 desarrolladores IOS
- 5 desarrolladores Android
- 4 desarrolladores backend
- 4 desarrolladores android
- 5 QAs evolutivos, entre los que estás tú

### **Tus funciones como QA evolutivo**

- Definición de los criterios de aceptación de esa HU
- Escritura de esos criterios de aceptación en BDD
- Ejecución manual

### **HU**

Se ha definido una nueva HU cuyo enunciado es el siguiente:

“Modificar el proceso de login para que además de usuario y contraseña sea obligatorio aceptar un

## **check de política de privacidad”**

**En base a la información aportada, se pide:**

### **1. Definición de los criterios de aceptación de esa HU**

1. El formulario de login debe incluir un nuevo campo de verificación de política de privacidad.
2. El campo de verificación de política de privacidad debe estar marcado por defecto.
3. El usuario debe poder desmarcar el campo de verificación de política de privacidad.
4. Al intentar realizar el login sin marcar el campo de verificación de política de privacidad, se debe mostrar un mensaje de error indicando que la aceptación de la política es obligatoria.
5. Al hacer clic en el enlace de la política de privacidad, se debe abrir una nueva ventana o pestaña con el contenido completo de la política.
6. La casilla de verificación de aceptación de la política de privacidad no debe interferir con los campos de usuario y contraseña, ni afectar su funcionalidad.
7. El sistema debe registrar la aceptación de la política de privacidad por parte del usuario para fines de auditoría y cumplimiento normativo.
8. Si el usuario olvida marcar la casilla de verificación de aceptación de la política de privacidad y hace clic en el botón de inicio de sesión, se le mostrará un mensaje de error que le recuerde que debe aceptar la política antes de poder continuar.
9. Si la política de privacidad cambia en el futuro, el sistema debe reflejar los cambios y requerir que los usuarios vuelvan a aceptar la nueva versión antes de poder iniciar sesión.
10. Al marcar el campo de verificación de política de privacidad y proporcionar el usuario y contraseña correctos, el usuario debe poder iniciar sesión correctamente.
11. Al iniciar sesión correctamente, el usuario debe ser redirigido a la página principal o a la última página visitada antes de iniciar sesión.
12. Al marcar el campo de verificación de política de privacidad y proporcionar un usuario o contraseña incorrectos, se debe mostrar un mensaje de error indicando que las credenciales son inválidas.
13. La funcionalidad de aceptar la política de privacidad debe ser compatible con los navegadores y dispositivos principales utilizados por los usuarios.
14. La funcionalidad de aceptar la política de privacidad debe ser probada en los diferentes mercados (EEUU, Europa, Asia y América del Sur) para asegurar su correcto funcionamiento en cada uno de ellos.

## 2. Escritura en BDD Gherkin de los casos de test para validar esa HU

Adjunto los casos en inglés porque es como los defino yo en mi puesto de trabajo actual:

Casos de prueba para validar la HU "Modify the login process to make acceptance of a privacy policy check mandatory":

Feature: Modify login process with mandatory privacy policy acceptance

Scenario: Display error message if privacy policy is not accepted

Given I am on the login page

When I enter my valid username and password

And I do not check the privacy policy acceptance checkbox

And I click the login button

Then I should see an error message stating that I must accept the privacy policy

Scenario: Successfully log in with privacy policy acceptance

Given I am on the login page

When I enter my valid username and password

And I check the privacy policy acceptance checkbox

And I click the login button

Then I should successfully log in and be redirected to the main page

Scenario: Display error message if credentials are invalid

Given I am on the login page

When I enter an invalid username or password

And I check the privacy policy acceptance checkbox

And I click the login button

Then I should see an error message stating that the credentials are invalid

Scenario: Verify compatibility across different browsers and devices

Given I am on the login page

When I enter my valid username and password

And I check the privacy policy acceptance checkbox

And I click the login button

Then I should see the main page or the last visited page before logging in

And the privacy policy acceptance functionality works correctly across different browsers and devices

Scenario Outline: Verify functionality in different markets

Given I am on the login page

When I enter my valid username and password

And I check the privacy policy acceptance checkbox

And I click the login button

Then I should see the main page or the last visited page before logging in

And the privacy policy acceptance functionality works correctly in the "<market>" market

Examples:

| market |

| USA |

| Europe |

| Asia |

| South America |

**3. Teniendo en cuenta que esta HU tiene implementación tanto de Web, como de iOS, Android y backend, comenta qué herramientas utilizarías para validar esta HU en cada disciplina.**

Para validar la HU de forma manual en cada disciplina, utilizaría las siguientes herramientas:

1. Validación Web:

- Navegadores web: Chrome, Firefox, Safari y Edge, para probar la funcionalidad en diferentes entornos de navegación.

- Herramientas de desarrollo integradas en los navegadores, como las "DevTools" de Chrome y las herramientas de desarrollador de Firefox, para inspeccionar elementos, verificar la consola de errores y realizar pruebas de rendimiento.

## 2. Validación iOS:

- Dispositivo iOS: realizar pruebas directamente en un dispositivo físico, utilizando Xcode, o algún simulador.

## 3. Validación Android:

- Dispositivo Android: Realizar pruebas directamente en un dispositivo físico o mediante algún emulador.

## 4. Validación Backend:

- API testing tools: Utilizaría herramientas como Postman para enviar solicitudes HTTP directamente a la API backend y verificar las respuestas.

- Pruebas de integración: Realizaría pruebas de integración donde se envían solicitudes al backend y se validan las respuestas, asegurándome de que los cambios en la funcionalidad del login no hayan afectado otras partes del sistema.

## **4. Como QA engineer dentro de un equipo, compártenos que ceremonias y reuniones crees que existirán durante la semana que dura el sprint.**

Como ingeniero de control de calidad (QA) en el equipo, durante la semana de sprint, participaremos en varias ceremonias y reuniones importantes para asegurar la calidad del producto. Aquí hay algunas de ellas:

1. Reunión de planificación del sprint: Al comienzo del sprint, nos reuniremos con el equipo para revisar y discutir las historias de usuario que se incluirán en el sprint. Analizaremos los requisitos y estableceremos las metas y prioridades para el sprint.

2. Reuniones diarias de seguimiento (stand-ups): Cada día, nos reuniremos brevemente con el equipo para compartir nuestro progreso, discutir los desafíos o problemas que enfrentamos y coordinar nuestras actividades. Estas reuniones son rápidas y están diseñadas para mantenernos en sintonía como equipo.

3. Sesiones de revisión de historias de usuario: Durante el sprint, colaboraremos con el equipo de desarrollo, el propietario del producto y otros miembros relevantes para revisar las historias de

usuario completadas. Verificaremos que se cumplan los criterios de aceptación y discutiremos cualquier problema o mejora necesaria.

4. Sesiones de retrospectiva del sprint: Al finalizar el sprint, nos reuniremos como equipo para reflexionar sobre el proceso y discutir lo que salió bien y lo que se puede mejorar. Compartiremos nuestros aprendizajes y trabajaremos juntos para implementar mejoras en futuros sprints.

5. Reuniones de demostración: Al final del sprint, participaremos en una reunión de demostración donde se presentarán las nuevas funcionalidades y mejoras implementadas durante el sprint. Esto nos permite mostrar nuestro trabajo y recibir comentarios valiosos de los stakeholders.

Además de estas ceremonias principales, también es posible que se realicen otras reuniones y actividades adicionales según las necesidades del equipo y del proyecto, como:

- Reuniones de refinamiento de historias de usuario: Estas reuniones se realizan para revisar y refinar las historias de usuario futuras antes de que se agreguen a un sprint.
- Reuniones ad hoc con el Product Owner: Para aclarar dudas sobre los requisitos, discutir cambios o prioridades, y recibir retroalimentación sobre la funcionalidad desarrollada.
- Reuniones de revisión de calidad: Para analizar métricas de calidad, revisar los resultados de pruebas y evaluar el estado general de la calidad del producto.
- Sesiones de colaboración y coordinación con otros equipos: Para garantizar una integración fluida y colaborar en la resolución de problemas o dependencias entre equipos.

**5. Como último paso, nos gustaría incorporar la validación de este nuevo caso de uso a nuestro plan de regresión automática.**

**• ¿Qué herramientas usarías para automatizar la parte de Web, iOS, Android y la parte backend?**

1. Automatización de pruebas web:

- Selenium WebDriver

2. Automatización de pruebas en iOS:

- XCTest o Appium.

3. Automatización de pruebas en Android:

- Appium

4. Automatización de pruebas en el backend:

- Postman

• **¿Qué herramientas de CI/CD crees que se podrían usar para programar el lanzamiento de la regresión de forma planificada y manual?**

- Jenkins o GitLab.

**6. El equipo de Desarrollo nos entrega un día tarde una HU prioritaria para el Producto. ¿Que estrategia tomarías en caso de que veas que no puedes llegar a probarla?**

Realizaría las siguientes acciones:

1. Comunicación: Lo primero que haría sería comunicar de inmediato la situación al Product Owner y al equipo de Desarrollo. Explicaría claramente que no hay suficiente tiempo para realizar pruebas exhaustivas en la Historia de Usuario debido a la entrega tardía.

2. Priorización de pruebas: En conjunto con el equipo, evaluaríamos qué pruebas son las más críticas para la Historia de Usuario. Identificaríamos los casos de prueba más importantes y los enfoques de prueba que nos permitan obtener una cobertura mínima pero relevante.

3. Enfoque de pruebas exploratorias: En lugar de seguir un enfoque de pruebas estructurado y detallado, optaría por realizar pruebas exploratorias. Esta técnica me permitiría explorar la funcionalidad de la Historia de Usuario de manera más libre y flexible, centrándome en los aspectos clave y descubriendo posibles problemas o defectos.

4. Pruebas básicas de aceptación: Me centraría en realizar pruebas básicas de aceptación para asegurarme de que la Historia de Usuario cumple con los requisitos mínimos y no presenta errores obvios. Esto incluiría verificar que los elementos de la interfaz de usuario sean accesibles y funcionen correctamente, y que se cumplan los flujos principales de la funcionalidad.

5. Comunicación transparente: Durante todo el proceso, mantendría una comunicación transparente con el equipo y el Product Owner para informarles sobre el alcance limitado de las pruebas y los posibles riesgos asociados. Sería honesto acerca de las limitaciones de tiempo y esfuerzo, y estaría dispuesto a colaborar para encontrar soluciones alternativas.