«Talento Tech»

Testing QA

Clase 03

Buenos Aires Ciudad

Clase N° 3 | Ciclo de Vida de QA y Gestión de Requerimientos

Temario:

- Criterios de Aceptación en las User Stories
    - Creación de los criterios de aceptación
        - Funcionales
        - No Funcionales
        - Error Handling (manejo del error)
- Ciclo de vida de QA en un proyecto
- Técnicas de Prueba
- Ejercicios.

Objetivos de clase

En esta clase, vamos a aprender a revisar y definir los Criterios de Aceptación (AC) en las User Stories, entendiendo su papel fundamental en la validación de requisitos y la calidad del software. Vamos a explorar la diferencia entre criterios funcionales y no funcionales, así como la importancia del manejo de errores (Error Handling) dentro de estos criterios. Además, vamos a analizar el ciclo de vida de QA en un proyecto y aplicar diferentes técnicas de prueba (funcionales, no funcionales y exploratorias)

Criterios de Aceptación en las User Stories

Los **Criterios de Aceptación (AC)** son una lista de condiciones que deben cumplirse para que una User Story se considere completa y satisfactoria. Actúan como una especie de "contrato" entre el equipo de desarrollo y el cliente, asegurando que todos entienden lo mismo sobre la funcionalidad.

Características de los Criterios de Aceptación:

- **Específicos:** Describen condiciones concretas y medibles.
- **Verificables:** Se pueden probar para determinar si se cumplen o no.
- **Comprensibles:** Son claros y fáciles de entender para todos los involucrados.

Tipos de Criterios de Aceptación:

- **Funcionales:** Describen el comportamiento del sistema.
    - Ejemplo: "El sistema debe permitir al usuario ingresar su nombre de usuario y contraseña para iniciar sesión."
- **No funcionales:** Describen aspectos como el rendimiento, la seguridad o la usabilidad.
    - Ejemplo: "El tiempo de respuesta del sistema no debe superar los 2 segundos."
- **Error Handling:** Describen cómo debe comportarse el sistema ante errores o excepciones.
    - Ejemplo: "Si el usuario ingresa una contraseña incorrecta, el sistema debe mostrar un mensaje de error claro."

User Stories (Historias de Usuario)

Tal como mencionamos la clase pasada las **User Stories** son descripciones concisas de una funcionalidad del software desde la perspectiva del usuario final. Se utilizan en metodologías ágiles como Scrum para capturar las necesidades del usuario de manera clara y sencilla.

Formato típico de una User Story:

"Como [tipo de usuario], quiero [acción que desea realizar] para [beneficio que obtiene]."

Ejemplos de User Stories:

- "Como cliente de un banco, quiero poder ver el saldo de mi cuenta online para controlar mis gastos."
- "Como administrador de una tienda online, quiero poder agregar nuevos productos al catálogo para mantenerlo actualizado."
- "Como usuario de una red social, quiero poder compartir fotos con mis amigos para mantenerme conectado con ellos."

Características de una buena User Story:

- **Centrada en el usuario:** Describe la necesidad desde la perspectiva del usuario, no del sistema.
- **Concisa:** Es breve y fácil de entender.
- **Valiosa:** Aporta valor al usuario o al negocio.
- **Estimable:** Se puede estimar el esfuerzo necesario para implementarla.
- **Probable:** Se puede probar para verificar que se cumple.

Ejemplo de User Story con Criterios de Aceptación:

User Story: "Como usuario de una app de transporte, quiero poder ver la ubicación del vehículo en tiempo real para saber cuándo llegará."

Criterios de Aceptación:

- **Funcionales:**
  - La app debe mostrar la ubicación del vehículo en un mapa.
  - La ubicación del vehículo debe actualizarse automáticamente cada 5 segundos.
  - La app debe mostrar la distancia y el tiempo estimado de llegada al destino.
- **No funcionales:**
  - La app debe consumir poca batería del dispositivo.

    - La app debe funcionar correctamente con diferentes conexiones a Internet (Wi-Fi, datos móviles).
- **Error Handling:**
    - Si no hay conexión a Internet, la app debe mostrar un mensaje informativo.
    - Si hay un error al obtener la ubicación del vehículo, la app debe mostrar un mensaje de error y ofrecer la opción de reintentar.

Relación entre User Stories y Criterios de Aceptación

Los Criterios de Aceptación son la "prueba de fuego" de una User Story. Definen cuándo la historia está realmente completa y cumple con las expectativas del usuario. Sin AC, una User Story puede ser interpretada de diferentes maneras, lo que puede llevar a funcionalidades incompletas o incorrectas.

En resumen:

- Las User Stories describen "qué" quiere el usuario.
- Los Criterios de Aceptación definen "cómo" se debe cumplir ese "qué".

Ejemplos con diferentes funcionalidades en una app de reservas de hoteles:

Funcionalidad 1: "Búsqueda de hoteles"

User Story: "Como usuario, quiero poder buscar hoteles por ciudad, fechas y cantidad de huéspedes para encontrar el alojamiento ideal para mis vacaciones."

Criterios de Aceptación:

- **Funcionales:**
    - El usuario debe poder ingresar la ciudad de destino.
    - El usuario debe poder seleccionar las fechas de check-in y check-out.
    - El usuario debe poder especificar la cantidad de huéspedes (adultos y niños).
    - La app debe mostrar una lista de hoteles disponibles que coincidan con los criterios de búsqueda.
    - La lista de hoteles debe incluir información relevante como nombre, foto, precio por noche y puntuación de los huéspedes.
- **No Funcionales:**
    - El tiempo de respuesta de la búsqueda no debe superar los 5 segundos.
    - La app debe ser compatible con diferentes dispositivos (celulares, tablets, computadoras).
    - La app debe ser fácil de usar e intuitiva.
- **Error Handling:**
    - Si no se encuentran hoteles disponibles para los criterios de búsqueda ingresados, la app debe mostrar un mensaje informativo y sugerir destinos alternativos o fechas flexibles.

    - Si la conexión a Internet se interrumpe durante la búsqueda, la app debe mostrar un mensaje de error y permitir al usuario reintentar la búsqueda más tarde.

Funcionalidad 2: "Reserva de habitación"

User Story: "Como usuario, quiero poder reservar una habitación en el hotel seleccionado para las fechas y cantidad de huéspedes especificadas."

Criterios de Aceptación:

- **Funcionales:**
    - El usuario debe poder seleccionar el tipo de habitación (simple, doble, suite, etc.).
    - El usuario debe poder ver el precio total de la reserva antes de confirmarla.
    - El usuario debe poder ingresar la información de los huéspedes (nombres, edades, etc.).
    - El usuario debe poder seleccionar un método de pago (tarjeta de crédito, transferencia bancaria, etc.).
    - La app debe enviar una confirmación de reserva al usuario por correo electrónico o SMS.
- **No Funcionales:**
    - El proceso de reserva no debe superar los 2 minutos.
    - La app debe garantizar la seguridad de la información personal y de pago del usuario.
    - La app debe ser accesible para personas con discapacidad visual o auditiva.
- **Error Handling:**
    - Si la reserva no se puede completar por algún motivo (falta de disponibilidad, error en el pago, etc.), la app debe mostrar un mensaje de error claro y ofrecer al usuario la opción de intentarlo nuevamente o contactar al servicio de atención al cliente.
    - Si el usuario ingresa información incorrecta o incompleta, la app debe mostrar mensajes de error específicos y guiar al usuario para que corrija la información.

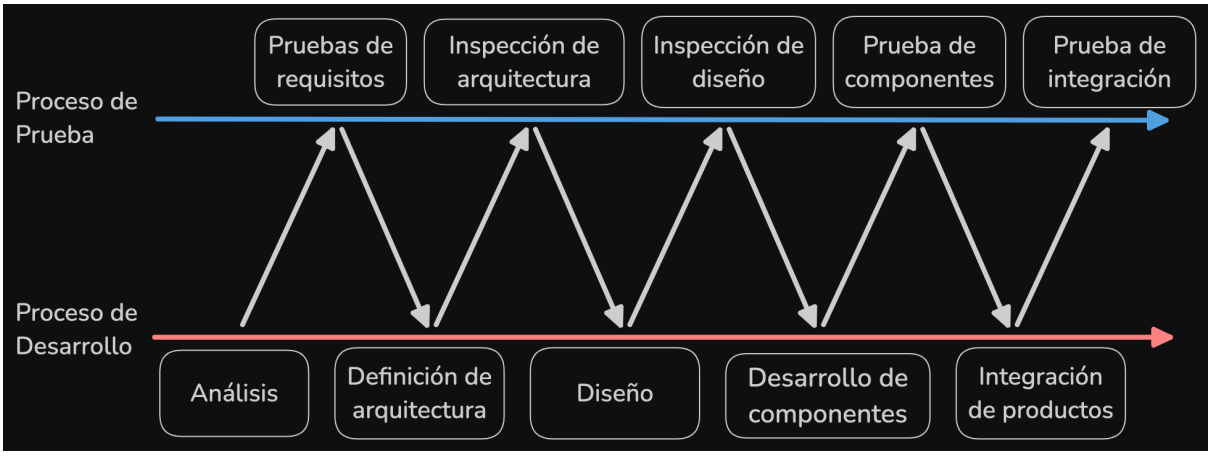
Ciclo de vida de QA en un proyecto

El ciclo de vida de QA en un proyecto se refiere a las diferentes etapas que atraviesa el equipo de QA desde el inicio hasta el final del proyecto.

Etapas:

1. Planificación: Se definen los objetivos de calidad, las estrategias de prueba y los recursos necesarios. Se establece el "qué" y el "cómo" del testing.

2. Diseño de pruebas: Se crean los casos de prueba que cubren los diferentes escenarios y funcionalidades del software. Se define "qué" se va a probar específicamente.
3. Ejecución de pruebas: Se ponen a prueba el software y se registran los resultados. Se lleva a la práctica el "cómo" se va a probar.
4. Gestión de defectos: Se comunican los errores encontrados a los desarrolladores para que los corrijan. Se realiza el seguimiento de los errores hasta su resolución.
5. Verificación de correcciones: Se asegura que los errores han sido corregidos correctamente. Se vuelve a probar el software para confirmar que los errores ya no están presentes.
6. Cierre: Se elabora un informe final con los resultados de las pruebas y las lecciones aprendidas. Se resume el trabajo realizado y se identifican áreas de mejora.



Ejemplo de Ciclo de vida del testing en relación con el Ciclo de vida del software.

Técnicas de Prueba

Existen diferentes técnicas de prueba que se pueden utilizar para evaluar la calidad del software.

Algunas de las más comunes son:

- Pruebas funcionales: Verifican que el software hace lo que se supone que debe hacer. Se centran en las funcionalidades del software.
- Pruebas no funcionales: Evalúan aspectos como el rendimiento, la seguridad y la usabilidad del software. Se centran en cómo funciona el software.
- Pruebas exploratorias: El tester explora el software de manera libre para encontrar errores inesperados. Se deja al tester descubrir errores a través de la exploración.
- Pruebas unitarias: Pruebas que realiza el desarrollador para verificar el correcto funcionamiento de pequeñas porciones de código. Se centran en el código del software.
- Pruebas de integración: Pruebas que se realizan para verificar el correcto funcionamiento de las interacciones entre los diferentes módulos del software. Se centran en la interacción entre los módulos.
- Pruebas de sistema: Pruebas que se realizan sobre el sistema completo para verificar que funciona correctamente en su totalidad. Se centran en el sistema como un todo.
- Pruebas de aceptación: Pruebas que realiza el usuario final para verificar que el software cumple con sus expectativas. Se centran en la satisfacción del usuario.

Iremos desarrollando y hablando de estas pruebas a lo largo del curso.

Ejercicio Práctico

¡Trabajando en Talento Lab!

Hoy vamos a trabajar con el desglose de requerimientos, usando metodologías ágiles. Dividiremos los proyectos en Épicas (los objetivos), Features (las funcionalidades) y User Stories (la interacción del usuario).

Matías nos va a ayudar con los criterios de aceptación para asegurar la calidad del producto.

Los criterios de aceptación (funcionales, no funcionales y de errores) son fundamentales para la calidad. Deben ser claros, concisos y verificables. No duden en preguntar si tienen alguna duda.

Ejemplo: Navegación

- **Épica**: Experiencia del Usuario
- **Feature**: Navegación Intuitiva
- **User Story**: Acceder a la página principal desde cualquier página.
- **Criterios**: Enlace visible, redirección correcta, carga en menos de 3 segundos.

Principios – Documentación de Entrada

Sitio Web TechLab: https://talentolab-test.netlify.app/

- **Instrucciones:**
    1. Según las dos funcionalidades elegidas del proyecto previamente en la clase anterior (ej.: Cargar CV en la página web).
    2. Creación de los criterios de aceptación:
        - Funcionales
        - No Funcionales
        - Error Handling (manejo del error)
    3. Documenta las Épicas, Features y User Stories manualmente en una hoja de cálculo (Excel)
- **Objetivo:** Desarrollar una descomposición detallada de los requerimientos del sistema, identificando criterios de aceptación, funcionalidades clave, aspectos no funcionales y estrategias de manejo de errores para garantizar la calidad del producto en un entorno ágil.

Ejemplo de: Documentación de requerimientos, épicas, features, user stories y criterios de aceptación de un sitio web de reservas

---

Preguntas para Reflexionar

- ¿Qué importancia tienen los Criterios de Aceptación en el proceso de desarrollo de software?
- ¿Por qué es importante considerar tanto los criterios funcionales como los no funcionales al definir los Criterios de Aceptación?

Próximos Pasos

En la próxima clase, vamos a ver los ambientes de uso normales en un proyecto de desarrollo de software y el uso de cada uno, lo que se entiende por backlog y sprint y las diferencias entre fallas y defectos.


Buenos Aires
aprende
Agencia de Habilidades para el Futuro
Buenos
Aires
Ciudad