Estrategia de pruebas

Aplicación bajo pruebas

Nombre aplicación

Modernización CCP

Versión

0.1

Descripción

Desarrollar una serie de aplicativos tanto web como móvil para modernizar la operación logística, compras y ventas de la empresa CCP. Esto con el fin de facilitar a la compañía su misión de crecer 3% anual por los siguientes 4 años, disminuir en un 10% en los años siguientes el costo de las inconformidades con clientes y reducir en un año un 60% las perdidas monetarias generadas por productos perecederos.

La modernización del software de CCP incluye crear un aplicativo web para el manejo de inventario, clientes, proveedores y despacho. Además, dos módulos (o dos aplicaciones) para aplicaciones móviles enfocado en la fuerza de ventas y el autoservicio para los clientes.

Contexto de la estrategia de pruebas

Este documento tiene como objetivo detallar las pruebas a realizar sobre el sistema, incluyendo los componentes que serán evaluados, el tipo de pruebas que se llevarán a cabo y el alcance de estas. Su propósito es asegurar que cada elemento del software cumpla con los estándares de calidad y funcione correctamente en diferentes escenarios. A través de este proceso, buscamos identificar posibles fallos, optimizar el rendimiento y garantizar una experiencia de usuario sin inconvenientes.

Tipos de pruebas

TNT (Técnicas, Niveles y Tipos) de pruebas:

En el proceso de pruebas de software, es fundamental estructurar las pruebas según su nivel, tipo y técnica utilizada. A continuación, se detallan los principales enfoques adoptados:

- **Pruebas Unitarias**: Se realizan a nivel de unidad de código y generalmente siguen una estrategia de **caja blanca**, donde se tiene visibilidad del código fuente. Para su ejecución, se emplea la técnica de **unit testing**, validando el correcto funcionamiento de componentes individuales.
- **Pruebas de Sistema**: En este nivel se evalúa el sistema en su conjunto, asegurando que cumpla con los requisitos esperados. Según la estrategia aplicada, pueden clasificarse en:
- **Pruebas de Caja Negra**: Se centran en la validación de la funcionalidad sin considerar la implementación interna. En este caso, pueden ejecutarse pruebas **E2E manuales** o **E2E automatizadas**, dependiendo de si la validación se realiza de forma manual o mediante scripts automatizados.
- **Pruebas Mixtas**: Combinan enfoques de caja negra y caja blanca para evaluar el sistema desde múltiples perspectivas. Un ejemplo de esto son las **pruebas manuales exploratorias**, en las que los testers interactúan con la aplicación sin un guion predefinido para identificar posibles problemas de usabilidad o comportamiento inesperado.

Nivel	**Tipo**	**Técnica**
Unitarias	Caja blanca	Unit testing
Sistema	Caja negra	- E2E manuales - E2E automatizadas
Sistema	Mixtas	Pruebas manuales exploratorioas

Módulos por evaluar en pruebas

Se definieron los siguientes componentes:

- Del lado del cliente:
    - Sistema web para gestión interna
    - Sistema móvil para fuerza de ventas
    - Sistema móvil para clientes
- Del lado del servidor:
    - Sistema de autenticación.
    - Sistema de gestión de usuarios.
    - Sistema de gestión de ventas.
    - Sistema de gestión de bodegas.
    - Sistema de gestión de productos.
    - Sistema de gestión de pedidos.
    - Sistema de gestión de clientes.
    - Sistema de gestión de recomendaciones.

Alcance de las pruebas

Para garantizar la calidad del software, se implementará una estrategia de pruebas basada en la metodología TDD (Test-Driven Development), lo que permitirá desarrollar pruebas unitarias a medida que se construye el código. La integración de nuevas funcionalidades en la rama principal del repositorio estará condicionada a un coverage mínimo del 75%, verificado mediante la librería pytest para el backend y Jest para el frontend. En el caso de las aplicaciones móviles, se usará la estructura de Android Studio recomendada.

Además de las pruebas unitarias, se llevarán a cabo pruebas de sistema que incluyen:

- Pruebas de Integración: Evaluarán la interacción entre múltiples componentes del sistema para verificar su correcto funcionamiento en conjunto.
- Pruebas E2E Automatizadas: Se implementarán pruebas de extremo a extremo para validar la integración completa del sistema, asegurando que los flujos críticos de la aplicación funcionen correctamente.

Para mantener la calidad del código, se aplicarán herramientas de análisis estático:

- Linting: Se utilizarán linters específicos para cada entorno, con flake8 en el backend (Python) y ESLint en el front-end (TypeScript), asegurando la correcta aplicación de convenciones de nomenclatura, espaciado y orden de importaciones.
- Análisis de Código: Se integrará SonarQube Cloud para evaluar métricas de calidad, detectar posibles vulnerabilidades y mejorar la mantenibilidad del código.

Finalmente, para medir el rendimiento del back-end, se ejecutarán pruebas de carga y estrés utilizando la librería Locust, permitiendo identificar cuellos de botella y optimizar la respuesta del sistema bajo diferentes condiciones de uso.

Duración de la iteración de pruebas

Debido a la extensión del proyecto y los tipos de prueba que se desean realizar, gran parte del desarrollo ira de la mano con el desarrollo de pruebas. Se espera entonces:

- 4 – 6 semanas de desarrollo de pruebas unitarias y e2e automatizadas. Desarrollo de pruebas en paralelo con desarrollo de la solución.
- 1 semana de pruebas manuales e2e para evaluar tanto el desarrollo como la UI y UX.
- 1 – 2 días para documentación de hallazgos y reporte.

Cada día de trabajo se espera 1 – 2 horas de dedicación exclusiva a pruebas por parte de cada desarrollador. En la etapa final, se espera que los desarrolladores dupliquen el tiempo dedicado a pruebas para ejecutar de forma extensa las pruebas manuales e2e.

Presupuesto de pruebas

Recursos computacionales

Dada la naturaleza de contratación que dispone el documento de arquitectura, se tiene en cuenta:

- No hay un presupuesto definido para recursos computacionales para cada desarrollador pues ellos cuentan con su propio dispositivo para este proyecto.
- De ser necesario, se facilitará un presupuesto de 50 USD por mes para recursos en la nube con el propósito de ejecutar pruebas. Este presupuesto lo puede usar el equipo de desarrollo en el servicio que desee y puede ser acumulativo. Es decir, cuentan con 100 USD para consumir en servicios web para el desarrollo de pruebas y libertad de ejecución.

Recursos Humanos

Disponibilidad de 4 desarrolladores tiempo parcial por las 8 semanas de duración de desarrollo del proyecto. Al ser un proyecto integral, no se contempla un presupuesto específico para el pago a los desarrolladores. No obstante, Del presupuesto que se tiene para el desarrollo definido en el documento de arquitectura, se estima que un 15% de este valor se asigne al presupuesto de pruebas. Esto equivale a 300 USD por mes por cada desarrollador.

Distribución del Esfuerzo

Para garantizar la calidad del software, se utilizarán diferentes tipos de pruebas alineadas con los objetivos definidos. La estrategia de pruebas incluye:

- **Pruebas Unitarias:** Se realizarán mediante **unit testing** en cada componente desarrollado.
- **Pruebas de Sistema:** Se ejecutarán pruebas **E2E manuales y automatizadas** para evaluar el sistema en su conjunto.

- **Pruebas Exploratorias:** Se llevarán a cabo pruebas manuales para identificar fallos no detectados en las pruebas estructuradas.

La asignación de esfuerzo y responsabilidades se definió aleatoriamente entre los cuatro desarrolladores, asegurando que todos participen en las pruebas del sistema.

- **Pruebas Unitarias:** Cada desarrollador es responsable de realizar pruebas unitarias en los componentes que implemente, siguiendo la metodología TDD (Test-Driven Development).
- **Pruebas E2E Manuales y Automatizadas:** Se distribuirán entre los cuatro desarrolladores, asegurando cobertura en los diferentes módulos.
- **Pruebas Exploratorias:** Se realizarán en conjunto antes de la entrega final del sistema.

Momentos de Ejecución

Las pruebas se ejecutarán en distintas fases del desarrollo:

- **Pruebas Unitarias:** Se realizarán constantemente durante la implementación de cada componente.
- **Pruebas E2E Manuales:** Se ejecutarán después del desarrollo de un componente.
- **Pruebas E2E Automatizadas:** Se ejecutarán en cada iteración y al finalizar componentes, con el fin de probar funcionalidades completas, de forma automatizada, como la creación de usuarios o la simulación de un pedido.
- **Pruebas Exploratorias:** Se realizarán en la fase final del desarrollo para detectar errores no previstos.