

## Estrategia de pruebas

### .1. Aplicación Bajo Pruebas

#### .1.1. Nombre Aplicación: CPP

#### .1.2. Versión: 1.0.0

.1.3. **Descripción:** La plataforma de CCP es un sistema de aplicaciones web y móviles diseñado para optimizar la gestión de ventas, distribución y seguimiento de productos. Facilita las operaciones internas de la compañía en logística e inventarios, al tiempo que permite a la fuerza de ventas y a los clientes registrar información y realizar pedidos de manera eficiente.

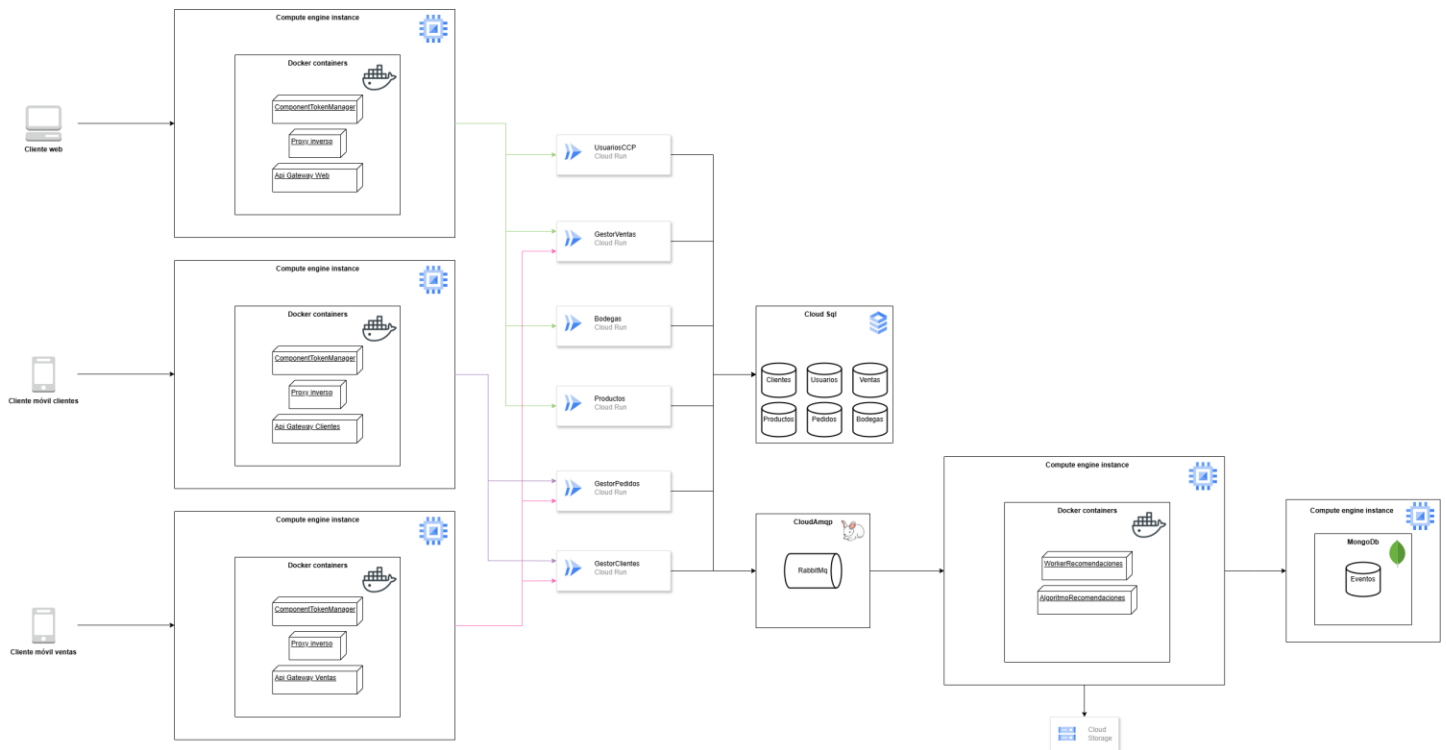
#### .1.4. Funcionalidades Core:

- Disponibilidad registro de fabricante
- Consulta fabricante
- Disponibilidad registro de vendedores
- Generación plan de ventas
- Acceso reporte de vendedores
- Carga de producto
- Carga masiva de producto
- Consultar información de un producto
- Gestión de fallas al ingresar producto
- Seguridad información de producto
- Gestión seguridad información geográfica de pedidos
- Ubicación de pedidos en bodegas
- Generación ruta de entrega
- Consulta ruta de entrega por área logística
- Gestión servicio de consulta de clientes
- Visualización de ruta de visitas de vendedor
- Registro de visitas de vendedor a cliente
- Consulta stock de producto
- Realización pedido por vendedor

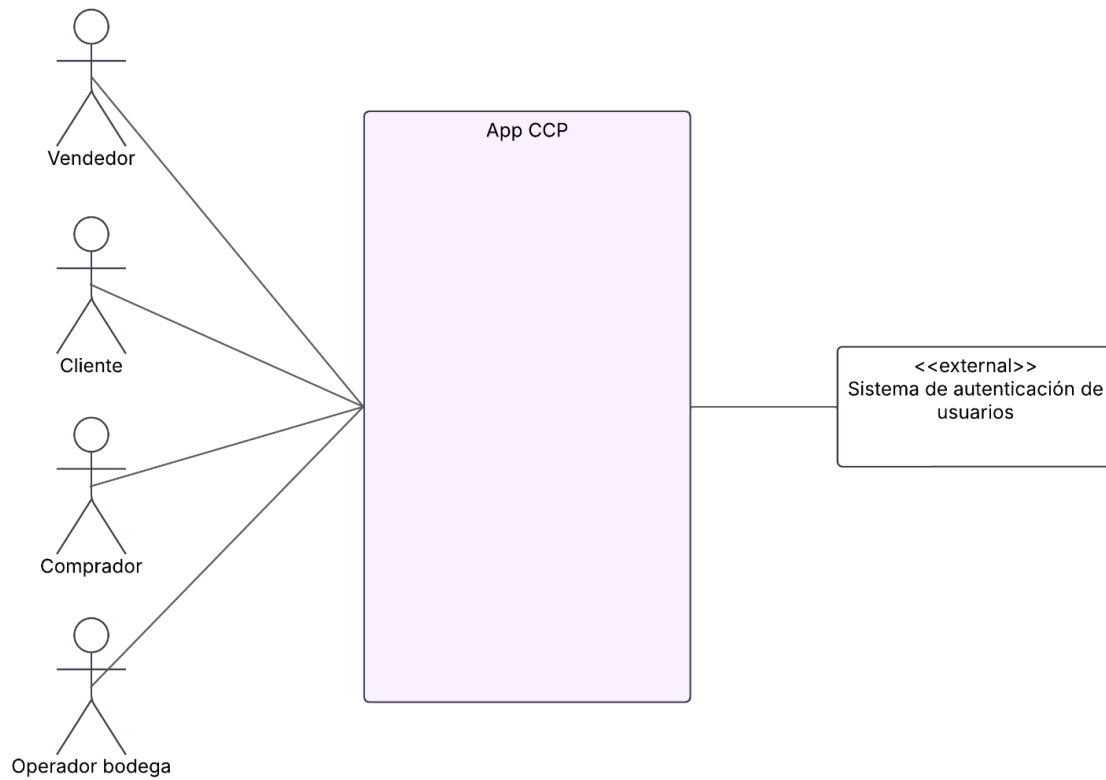
- Gestión carga de videos
- Solicitud de recomendaciones
- Gestión registro de clientes
- Creación de pedido cliente
- Consulta estado de pedido cliente

### .1.1. Diagrama de Arquitectura:

[https://github.com/imorangedev/proyecto\\_final\\_miso/blob/main/Docs/W3%20-%20Diagrama%20de%20arquitectura.png](https://github.com/imorangedev/proyecto_final_miso/blob/main/Docs/W3%20-%20Diagrama%20de%20arquitectura.png)

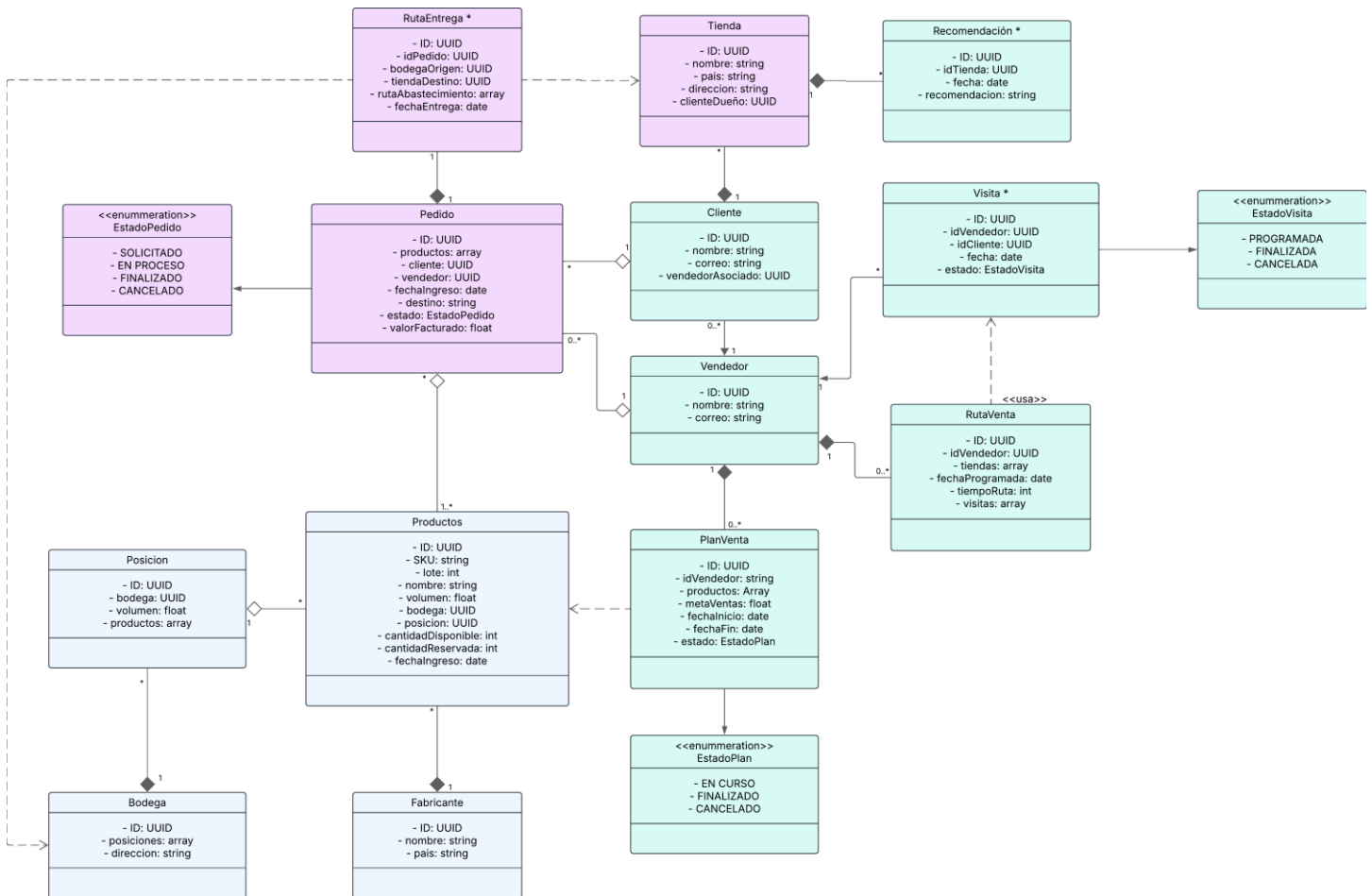


### .1.2. Diagrama de Contexto:



### .1.3. Modelo de Datos:

[https://lucid.app/lucidchart/9a39cda8-d88d-4d8c-8c5b-eb723382f772/edit?viewport\\_loc=-3197%2C-1729%2C6801%2C4329%2Cz9UQ20RIPJU8&invitationId=inv\\_caf52fcf-2a07-4c1c-97eb-4ffcb4f41b87](https://lucid.app/lucidchart/9a39cda8-d88d-4d8c-8c5b-eb723382f772/edit?viewport_loc=-3197%2C-1729%2C6801%2C4329%2Cz9UQ20RIPJU8&invitationId=inv_caf52fcf-2a07-4c1c-97eb-4ffcb4f41b87)



## .2. Contexto de la estrategia de pruebas

### .2.1. Objetivos:

- .2.1.1. Garantizar la estabilidad y correcto funcionamiento de los componentes individuales a través de pruebas unitarias, alcanzando una cobertura mínima de pruebas del 75%.
- .2.1.2. Identificar defectos y problemas de usabilidad mediante pruebas exploratorias, evaluando la aplicación desde una perspectiva mixta (caja blanca y caja negra).
- .2.1.3. Asegurar la integración efectiva de los módulos y flujos de trabajo por medio de pruebas E2E manuales y automatizadas, verificando la coherencia y comunicación entre los diferentes componentes del sistema.
- .2.1.4. Detectar errores en la traducción de palabras y segmentos de texto por medio de pruebas unitarias y pruebas semiautomatizadas de GUI Ripping y exploración manual de los resultados.
- .2.1.5. Garantizar que la aplicación sea accesible para todos los usuarios mediante la realización de pruebas de accesibilidad. Esto incluye evaluar la conformidad con las pautas WCAG (Web Content Accessibility Guidelines) y verificar la funcionalidad con tecnologías de asistencia.

- .2.1.6. Validar la respuesta entre componentes que requieren una comunicación síncrona por medio de pruebas de integración, con el fin de detectar respuestas o interacciones no esperadas

## **.2.2. Duración de la iteración de pruebas:**

Se espera que las pruebas se puedan cumplir en los siguientes periodos de tiempo:

- 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 semana para el desarrollo y ejecución de pruebas exploratorias automatizadas
- 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.

## **.2.3. Presupuesto de pruebas:**

### **.2.3.1. 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.

### **.2.3.2. Recursos Computacionales:**

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

### **.2.3.3. Recursos Económicos para la contratación de servicios/personal:**

Los recursos económicos destinados a la contratación de servicios, personal y la realización de pruebas provendrán del presupuesto asignado a los salarios de los cuatro desarrolladores encargados de la aplicación. Es decir, no se contempla un presupuesto adicional para estas actividades, ya que serán cubiertas dentro de los honorarios del equipo de desarrollo.

## **.2.4. TNT (Técnicas, Niveles y Tipos) de pruebas:**

- .2.4.1. **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.
- .2.4.2. **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.
- .2.4.3. **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	Objetivo
Unitarias	Caja blanca	Unit testing	1, 5
Integración	Caja gris	Integration testing	6
Sistema	Caja negra	Pruebas E2E (APIs)	3
Sistema	Mixtas	Pruebas manuales exploratorias	2, 5
Sistema	Caja negra	GUI Ripping	4

## .2.5. Distribución de 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 y de integración:** 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.

- **GUI Ripping:** El ripper será escrito por 2 desarrolladores enfocados a cubrir la parte automática de las pruebas de internacionalización. Los otros 2, estarán enfocados en pruebas unitarias sobre los componentes de texto más importantes de las aplicaciones
- **Pruebas de accesibilidad:** Cada desarrollador usará linters para verificar su código a medida que escribe. Sin embargo, 1 desarrollador se encargará de hacer la configuración del pipeline para ejecutar pruebas sobre la totalidad del código

© - **Derechos Reservados:** la presente obra, y en general todos sus contenidos, se encuentran protegidos por las normas internacionales y nacionales vigentes sobre propiedad intelectual, por lo tanto su utilización parcial o total, reproducción, comunicación pública, transformación, distribución, alquiler, préstamo público e importación, total o parcial, en todo o en parte, en formato impreso o digital y en cualquier formato conocido o por conocer, se encuentran prohibidos, y solo serán lícitos en la medida en que se cuente con la autorización previa y expresa por escrito de la Universidad de los Andes.

De igual manera, la utilización de la imagen de las personas, docentes o estudiantes, sin su previa autorización está expresamente prohibida. En caso de incumplirse con lo mencionado, se procederá de conformidad con los reglamentos y políticas de la universidad, sin perjuicio de las demás acciones legales aplicables.

---