UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES

GRUPO # 11

PLATAFORMA INTELIGENTE DE COMERCIO ELECTRÓNICO PARA TIENDA DE ROPA MASCULINA CON VESTIDORES VIRTUALES VÍA REALIDAD AUMENTADA

INTEGRANTES:

- **Delgado Rojas Alberto Caleb** **224027204**
- **Mujica Vallejos Andy Mauricio** **224028367**

MATERIA: SISTEMAS DE INFORMACIÓN II (SI2) - SA

SEMESTRE:2-2026

Docente: MSc. Ing. Angélica Garzón Cuéllar

Santa Cruz de la Sierra — Bolivia

TABLA DE CONTENIDO (ÍNDICE GENERAL)

Contenido

1) Perfil 7
1.1 Introducción 7
1.2 Objetivo General 8
1.3 Objetivos Específicos 8
1.4 Descripción del Problema 9
1.4.1 Contexto Global y Regional del Comercio Minorista de Moda Masculina 9
1.4.2 Diagnóstico de la Situación Actual y Problemáticas Críticas 9
1.4.3 Matriz de Análisis Causa-Efecto (Diagrama de Ishikawa) 11
1.4.4 Impacto y Consecuencias de la Inacción 12
1.4.5 Formulación de la Solución Integral con FashionStore 12
1.5 Alcance 13
1.5.1 Módulos del Sistema 13
1.5.2 Requisitos Funcionales 27
1.5.3 Requisitos No Funcionales 29
1.5.4 Actores del Sistema 30
1.5.5 Planificación de Iteraciones (Ciclos de Desarrollo PUDS) 31
1.5.6 Matriz de Funcionalidades por Plataforma 32
1.5.7 Restricciones y Limitaciones del Proyecto 33
2. Elementos del sistema basado en computadoras 34
2.1 Hardware 34
2.1.1 Servidor 34
2.1.2 Cliente 35
2.1.3 Otros Dispositivos 35
2.2 Software 36
2.2.1 Servidor 36
2.2.2 Cliente 37
2.2.3 Otro Software Adicional 38
2.3 Datos 38
2.4 Procesos 40
2.5 Gente / Usuario 41
3. Tecnología para el Desarrollo del Software 42

3.1 Estrategia para el Desarrollo del Software 42
3.2 Metodología para el Desarrollo del Software 43
3.2.2 Características Principales de UML 2.5+ 44
3.3 Herramientas de Desarrollo 45
3.3.1 Software 45
3.3.2 Hardware de los Desarrolladores 45
4. Factibilidad Económica y Costos del Proyecto 46
4.1 Análisis de Costos de Desarrollo (Inversión Inicial) 47
4.1.1 Recursos Humanos 47
4.1.2 Hardware Crítico para Operación 47
4.1.3 Software y Servicios Cloud (Año 1) 48
4.2 Análisis de Beneficios e Impacto Financiero 49
4.3 Beneficios Tangibles (Ahorro Anual Proyectado) 50
4.4 Beneficios Intangibles 51
4.5 Beneficios Esperados 51
4.5.1 Beneficios Operacionales y de Gestión (Procesos en Sucursales y Tiendas) 51
4.5.2 Beneficios Estratégicos y de Inteligencia de Negocios 52
4.5.3 Beneficios para el Capital Humano y de Seguridad 52
4.5.4 Beneficios de Escalado Tecnológico 52
Marco Teórico: Fundamentación Teórica 53
a) Comercio Electrónico (E-commerce) 53
Conceptos Generales, Modelos de Negocio y Evolución 53
Análisis Comparativo como Usuario: Amazon, Alibaba y Shopify 53
Análisis Comparativo como Desarrollador: Magento, PrestaShop y WooCommerce 54
b) Pasarelas de Pago 54
Mecanismos de Pago en Línea 54
Pasarela Nacional: LIBÉLULA 55
Pasarelas Internacionales: STRIPE y PAYPAL 55
c) Servicios de Delivery y Logística de Última Milla 56
Operación del Delivery en Aplicaciones Urbanas (Yaigo, Yummy, PedidosYa) 56
Algoritmos y Variables de Cálculo de Tarifas de Envío 56
d) Proceso Unificado de Desarrollo de Software (PUDS) 57
Características Esenciales 57
Fases del Ciclo de Vida PUDS 57

Flujos de Trabajo Fundamentales (Disiplinas) 57
e) Lenguaje de Modelado Unificado (UML 2.5+) 58
Taxonomía de Diagramas en UML 2.5+ 58
Modelo de Negocio (Business Modeling - Diagramas de Actividades de Macroprocesos) 59
2.1 Justificación del Modelo de Negocio en el PUDS 59
2.2 Macroproceso 1: Gestión de Abastecimiento y Recepción con Costo Promedio Ponderado ..59
2.3 Macroproceso 2: Reserva Omnicanal y Prueba Presencial en Sucursal.....61
2.4 Macroproceso 3: Venta Presencial en Punto de Venta (POS) 63
2.5 Macroproceso 4: Venta Digital y Despacho con Pasarela de Pago 65
2.6 Macroproceso 5: Experiencia Inteligente (Vestidor Virtual RA + Asistente IA) 67
2.7 Macroproceso 6: Programa de Fidelización Gamificado.....69
Capitulo1. Flujo de Trabajo: Captura de Requisitos 71
1.1 Identificación de Casos de Uso y Actores 71
1.1.1 Actores Internos (Personal de la Organización) 71
1.1.2 Actores Externos (Usuarios y Sistemas Cooperantes) 72
1.1.3 Lista de casos de uso 74
1.2 Priorización de Casos de Uso 76
1.3 Detalle de Casos de Uso y Prototipado de Interfaz de Usuario 76
1.3.1 Caso de Uso CU01: Autenticar Usuario y Control de Acceso (RBAC).....77
1.3.2 Caso de Uso CU02: Registrar Cliente (Auto-registro de Clientes).....79
1.3.3 Caso de Uso CU03: Recuperar Contraseña (Token OTP de 6 Dígitos) 81
1.3.4 Caso de Uso CU04: Gestionar Usuarios y Roles (RBAC y Desbloqueo).....84
1.3.5 Caso de Uso CU05: Gestionar Ciudades y Sucursales.....86
1.3.6 Caso de Uso CU06: Gestionar Productos y Atributos de Moda.....88
1.3.7 Caso de Uso CU07: Gestionar Temporadas y Colecciones 90
1.3.8 Caso de Uso CU08: Gestionar Proveedores Textiles.....92
1.3.9 Caso de Uso CU09: Gestionar Inventario Multi-Sucursal y Costos Ponderados (CPP) 93
1.3.10 Caso de Uso CU10: Consultar Catálogo y Disponibilidad por Sucursal.....94
1.4 Estructurar Modelo de Casos de Uso.....96
2. Flujo de Trabajo: Análisis 96
2.1 Análisis de Arquitectura 96
2.1.1 Identificar Paquetes.....97
2.1.2 Relacionar paquetes y casos de uso 98
2.1.3 Vista de casos de uso.....101

2.2 Analizar Casos de Uso (Diagramas de Comunicación UML) 103
2.2.1 Diagrama de Comunicación - CU01: Autenticar Usuario (RBAC).....103
2.2.2 Diagrama de Comunicación - CU02: Registrar Cliente (Auto-registro) 103
2.2.3 Diagrama de Comunicación - CU03: Recuperar Contraseña (Token OTP) 104
2.2.4 Diagrama de Comunicación - CU04: Gestionar Usuarios y Roles (RBAC) 104
2.2.5 Diagrama de Comunicación - CU05: Gestionar Sucursales y Ciudades.....105
2.2.6 Diagrama de Comunicación - CU06: Gestionar Productos y Atributos 105
2.2.7 Diagrama de Comunicación - CU07: Gestionar Temporadas y Colecciones 106
2.2.8 Diagrama de Comunicación - CU08: Gestionar Proveedores Textiles.....106
2.2.9 Diagrama de Comunicación - CU09: Gestionar Inventario y Costos (CPP) 106
2.2.10 Diagrama de Comunicación - CU10: Consultar Catálogo y Disponibilidad 107
2.3 Análisis de Clases.....107
2.4 Análisis de Paquetes 112
3. Flujo de Trabajo: Diseño.....114
3.1 Diseño de Arquitectura.....114
3.1.1 Diseño Lógico de la Arquitectura (4 Capas UML) 114
3.1.2 Diseño Físico de la Arquitectura (Diagrama de Despliegue) 114
3.2 Diseño de Casos de Uso 115
3.2.1 Diagrama de Secuencia - CU01: Autenticar Usuario (Login RBAC) 115
3.2.2 Diagrama de Secuencia - CU02: Registrar Cliente (Sign Up).....116
3.2.3 Diagrama de Secuencia - CU03: Recuperar Contraseña (Token OTP).....117
3.2.4 Diagrama de Secuencia - CU04: Gestionar Usuarios y Roles (RBAC).....118
3.2.5 Diagrama de Secuencia - CU05: Gestionar Sucursales y Ciudades 119
3.2.6 Diagrama de Secuencia - CU06: Gestionar Productos y Atributos de Moda 120
3.2.7 Diagrama de Secuencia - CU07: Gestionar Temporadas y Campañas 121
3.2.8 Diagrama de Secuencia - CU08: Gestionar Proveedores Textiles 122
3.2.9 Diagrama de Secuencia - CU09: Gestionar Inventario Multi-Sucursal (CPP) 123
3.2.10 Diagrama de Secuencia - CU10: Consultar Catálogo y Disponibilidad 124
3.2.2 Diagramas de Estado - Dominio del Sistema 125
a) Ciclo de Vida de la Cuenta de Usuario y Token OTP (Seguridad) 125
b) Ciclo de Vida del Producto e Inventario 126
3.2.3 Diagrama de Navegación del Sistema (Ciclo 1) 126
3.2.4 Diagrama de Tiempo.....127
3.3 Diseño de Datos 129

3.3.1 Diseño de Datos Lógico (Diagrama ER / Clases Persistentes) 129
3.3.2 Diseño de Datos Físico (Script DDL SQL en PostgreSQL).....130
3.3.3 Población de Datos (Script DML SQL en PostgreSQL) 132
4. Flujo de Trabajo: Implementación.....134
4.1 Selección de la Plataforma de Software 134
4.2 Implementación de la Arquitectura del Sistema Principal 134
4.3 Implementación de la Arquitectura del Sub Sistema.....135
4.3.1 Sub Sistema 1: Seguridad y Control de Acceso RBAC (M01 - CU01, CU02, CU03, CU04) 136
4.3.2 Sub Sistema 2: Estructura Operativa y Sucursales (M02 - CU05).....136
5. Flujo de Trabajo: Pruebas 138
5.1 Pruebas de Casos de Uso (Pruebas de Caja Negra) 138
Bibliografía.....158
Anexos.....159
Enlace repositorio y QR.....160
Enlace despliegue.....160
- Web: 160

. . . . . .

1) Perfil

1.1 Introducción

El comercio electrónico (e-commerce) ha redefinido las dinámicas del comercio minorista global, convirtiéndose en un canal indispensable de competitividad, escalabilidad y generación de valor. En el sector de la moda y la indumentaria, los canales digitales experimentan tasas de expansión vertiginosas; dentro de este segmento, el mercado de la moda masculina ha adquirido un protagonismo sin precedentes. El consumidor masculino contemporáneo exige eficiencia, inmediatez, precisión en las especificaciones del producto y una experiencia de compra personalizada que optimice su tiempo y reduzca la fricción en la toma de decisiones.

A pesar de esta acelerada evolución tecnológica, el sector de tiendas de ropa física en nuestro medio enfrenta serias barreras operativas y estructurales al intentar insertarse en la economía digital. La problemática más crítica radica en la **brecha experiencial** existente entre el entorno físico y el entorno digital: el cliente que compra por internet no tiene la posibilidad de palpar los textiles, apreciar los acabados ni verificar el calce y la talla anatómica de las prendas. Esta incertidumbre produce tasas alarmantes de devolución de mercadería (que en la industria textil internacional oscilan entre el 25% y el 35%), carritos de compra abandonados y una erosión constante de la confianza del consumidor hacia las marcas locales.

Simultáneamente, las cadenas minoristas que operan con múltiples sucursales distribuidas en diversas ciudades del país padecen de una marcada fragmentación en la gestión de sus inventarios. La inexistencia de una sincronización centralizada en tiempo real provoca quiebres de existencias (rotura de stock en una sucursal mientras otra acumula sobrestock obsoleto), costos logísticos descontrolados e incapacidad para brindar respuestas inmediatas sobre la disponibilidad de tallas y colores específicos. A ello se suma la ausencia de sistemas rigurosos de valuación de inventario fundamentados en el **costo promedio ponderado**, lo que distorsiona los márgenes reales de ganancia y la rentabilidad financiera del negocio.

Para dar una respuesta integral, innovadora y sostenible a esta problemática, se concibe y diseña **FashionStore**: una plataforma inteligente de comercio electrónico omnicanal para tiendas de ropa masculina, estructurada bajo una arquitectura moderna cliente-servidor basada en microservicios desacoplados. FashionStore conecta de manera sinérgica la presencia digital (web responsiva y aplicación móvil nativa multiplataforma) con la infraestructura física de las tiendas, permitiendo a los clientes no solo adquirir prendas en línea o en puntos de venta físicos (POS), sino también gestionar reservas anticipadas para probarse las prendas presencialmente en sucursales seleccionadas.

Para superar la brecha digital y elevar sustancialmente la competitividad comercial, FashionStore incorpora tres pilares tecnológicos diferenciadores:

1. **Vestidores Virtuales basados en Realidad Aumentada (RA)**: Mediante la cámara del dispositivo móvil, el usuario puede proyectar las prendas tridimensionalmente sobre su silueta en tiempo real, facilitando la visualización del calce y estilo.
2. **Asistente de Estilo con Inteligencia Artificial Contextual**: Un motor cognitivo que genera sugerencias personalizadas de combinaciones y atuendos completos (outfits) analizando variables en tiempo real: condiciones meteorológicas de la ciudad del usuario (vía API de clima), historial transaccional, preferencias estilísticas, temporadas comerciales y análisis cromático/colorimetría.

3. **Sistema de Fidelización Gamificado y Comparador de Outfits**: Un esquema de retención de clientes apoyado en dinámicas de juego (puntos acumulables, niveles Bronce/Plata/Oro/Diamante, insignias por hitos y canje de beneficios) junto con una herramienta interactiva que permite contrastar simultáneamente hasta tres combinaciones de ropa en pantalla.

El desarrollo del proyecto se fundamenta con estricto apego en el **Proceso Unificado de Desarrollo de Software (PUDS)** y el lenguaje de modelado estandarizado **UML versión 2.5+**, garantizando un proceso disciplinado, dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental a lo largo de tres ciclos evolutivos de desarrollo.

. . . . . .

1.2 Objetivo General

Desarrollar una plataforma inteligente de comercio electrónico para una cadena de tiendas de ropa, que integre comercio electrónico web y móvil, reservas de prendas, gestión de sucursales, inventario, puntos de venta, pagos electrónicos, vestidores virtuales mediante realidad aumentada e

inteligencia artificial, utilizando el Proceso Unificado de Desarrollo y modelos UML.

1.3 Objetivos Específicos

1. Analizar y especificar los requisitos funcionales y no funcionales del sistema.
2. Gestionar clientes, usuarios, empleados, proveedores y sucursales.
3. Administrar el catálogo de prendas, categorías, tallas, colores, temporadas y colecciones.
4. Consultar la disponibilidad de prendas por sucursal.
5. Permitir al cliente reservar varias prendas para probarlas posteriormente en una tienda.
6. Gestionar el proceso de recepción y atención de reservas.
7. Incorporar vestidores virtuales utilizando realidad aumentada.
8. Permitir compras mediante la plataforma web y aplicación móvil.
9. Permitir pagos presenciales en puntos de caja.
10. Integrar una pasarela de pago para compras digitales.
11. Actualizar automáticamente el inventario después de reservas, ventas, devoluciones y recepción de productos.
12. Gestionar productos provenientes de diferentes proveedores y temporadas.
13. Incorporar funcionalidades de inteligencia artificial para recomendación o asistencia al cliente.
14. Generar reportes y dashboards para apoyar la toma de decisiones.
15. Aplicar UML 2.5+ para modelar los procesos, funcionalidades y arquitectura del sistema.
16. Implementar el sistema utilizando FastAPI, Angular y Flutter/Dart

1.4 Descripción del Problema

1.4.1 Contexto Global y Regional del Comercio Minorista de Moda Masculina

El comercio minorista de prendas de vestir (apparel retail) representa uno de los sectores económicos más dinámicos y competitivos a nivel global. Históricamente caracterizado por un modelo de comercialización eminentemente físico donde la interacción táctil con los tejidos, la verificación presencial del corte y la prueba en probadores constituían el núcleo del proceso de compra—, el sector ha experimentado una metamorfosis irreversible acelerada por la digitalización de los consumidores y la masificación de los teléfonos inteligentes.

En el mercado específico de la moda masculina, las tendencias de consumo han evolucionado hacia un patrón distintivo: el comprador masculino contemporáneo valora por encima de todo la **practicidad, la optimización del tiempo, la claridad en la información y la coherencia del estilo**. A diferencia de otros segmentos donde la compra responde predominantemente a impulsos o recreación prolongada, el público masculino busca resolver una necesidad concreta de vestimenta (para el entorno laboral, ocasiones formales, actividades deportivas o estilo urbano) de manera rápida y sin complicaciones.

Sin embargo, en el contexto de Bolivia y la región latinoamericana, el comercio de moda masculina se encuentra atrapado en una profunda encrucijada estructural. Por una parte, los negocios tradicionales continúan operando con procesos manuales o sistemas de información aislados que carecen de conectividad entre sus puntos de venta. Por otra parte, aquellos comercios que han intentado incursionar en el comercio electrónico lo han hecho a través de canales precarios e improvisados como catálogos estáticos en redes sociales (Facebook Marketplace, WhatsApp, Instagram) o mediante la adopción de plataformas genéricas no adaptadas a la realidad operativa del país. Esta situación genera un conjunto de problemáticas críticas que se analizan detalladamente a continuación.

. . . . . .

1.4.2 Diagnóstico de la Situación Actual y Problemáticas Críticas

A. La Brecha Experiencial entre el Entorno Digital y el Físico: El Dilema del Calce y la Crisis de Devoluciones

El principal obstáculo que inhibe la conversión en las tiendas de ropa digital es la imposibilidad física de probarse las prendas antes del pago. Las tablas de tallas tradicionales (S, M, L, XL) carecen de uniformidad entre fabricantes, marcas y líneas de confección: un pantalón talla 32 de corte Slim Fit de una marca importada difiere notablemente de un modelo equivalente de corte clásico o confección nacional.

Esta incertidumbre dimensional genera una barrera psicológica en el cliente, quien ante la duda opta por desistir de la compra en línea, recurrir a compras presenciales con alta pérdida de tiempo, o incurrir en la práctica del bracket-ordering (adquirir deliberadamente dos o tres tallas de la misma prenda para quedarse con una y devolver el resto). En la industria de la moda a nivel internacional, las ventas en línea registran tasas de devolución superiores al **30%**, mientras que en las tiendas físicas difícilmente alcanzan el **8% al 10%**. Cada devolución representa un impacto financiero severo: costos de reenvío y logística inversa, reacondicionamiento de prendas, deterioro del empaque y pérdida del costo de oportunidad por stock inmovilizado durante el proceso de restitución.

B. Desarticulación y Silos de Información en la Gestión de Inventarios Multi-Sucursal

Las cadenas de tiendas de ropa que operan en múltiples ciudades (como Santa Cruz de la Sierra, La Paz y Cochabamba) y con diversas sucursales urbanas sufren de una severa desconexión informativa. Frecuentemente, el inventario es controlado de manera fragmentada: cada tienda física cuenta con su propio cuaderno de registro, una hoja de cálculo local o un sistema monofásico desconectado de la central.

Esta falta de visibilidad en tiempo real produce dos fenómenos igualmente perjudiciales:

1. **Quiebre o rotura de stock (*Stockouts*)**: Un cliente acude a una sucursal buscando una camisa ejecutiva blanca en talla M y se le informa que el producto está agotado, perdiendo la venta de inmediato porque el personal desconoce si en la sucursal de otra zona de la ciudad existen unidades disponibles.
2. **Sobrestock y obsolescencia estacional**: En una sucursal se acumulan decenas de unidades de prendas de temporada fría que no rotan, mientras en otra sucursal de una ciudad de clima templado o frío existe sobredemanda insatisfecha. La falta de un sistema centralizado impide planificar traslados y transferencias inter-sucursales oportunas antes de que la temporada comercial culmine y la mercadería deba liquidarse con pérdidas.

C. Distorsión Contable y Ausencia de Valuación por Costo Promedio Ponderado

La adquisición de prendas de vestir está sujeta a una alta variabilidad de costos de adquisición. Las cadenas compran lotes sucesivos de un mismo producto en diferentes periodos del año; los costos unitarios fluctúan drásticamente debido a variaciones cambiarias de divisas, fletes de importación, volumen negociado con el proveedor, aranceles aduaneros y fluctuaciones inflacionarias.

La gran mayoría de los sistemas comerciales básicos registra el inventario utilizando únicamente el **último costo unitario de compra**. Esta simplificación contable es nefasta para la salud financiera de la empresa:

- Si el último lote de 50 camisas se adquirió a un precio mayor (ej. 120 Bs) debido a un aumento transitorio de transporte, y aún existían en bodega 150 camisas compradas a 80 Bs, registrar todo el inventario a 120 Bs sobrevalora artificialmente el activo y distorsiona el margen de utilidad.
- Por el contrario, si se adquirió un lote de liquidación a menor precio, se subvalora el inventario existente.
- Además, en el marco tributario boliviano, las operaciones comerciales formales deben considerar el crédito y débito fiscal del Impuesto al Valor Agregado (IVA - 13%) y el Impuesto a las Transacciones (IT - 3%), totalizando una carga impositiva formal que exige conocer con exactitud matemática el **costo promedio ponderado** del inventario para determinar con rigor la utilidad operativa neta y cumplir con las regulaciones de auditoría.

D. Ineficiencia en la Dinámica de Compra Presencial y Falta de Mecanismos de Reserva Previa

Cuando un cliente decide visitar una tienda física, el proceso tradicional de atención está plagado de fricciones. El cliente debe buscar manualmente entre perchas y anaqueles saturados, solicitar la intervención de un vendedor para consultar si hay stock en bodega, esperar a que el encargado revise depósitos desordenados y, si tiene éxito, esperar turno frente a probadores colapsados en horas pico. Si la prenda no calza adecuadamente, el ciclo de búsqueda debe repetirse desde cero.

Este flujo arcaico genera agotamiento en el consumidor, saturación en las salas de venta y un bajo índice de conversión por metro cuadrado de tienda. No existe un puente tecnológico que permita al cliente explorar digitalmente el catálogo desde su hogar o teléfono, seleccionar las prendas exactas que desea probarse, y **generar una reserva anticipada** para una sucursal específica con fecha y hora programadas, de modo que al llegar a la tienda el personal ya tenga las prendas apartadas, planchadas y listas en el probador asignado.

E. Ausencia de Personalización, Inteligencia Contextual y Fidelización del Cliente

El comercio minorista tradicional trata a todos los clientes bajo un estándar homogéneo. El comprador promedio se enfrenta a catálogos masivos y genéricos con cientos de prendas sin ningún tipo de curaduría adaptada a su contextura, estilo de vida, entorno geográfico o preferencias cromáticas.

Asimismo, no se aprovechan las condiciones de contexto dinámico: no tiene sentido recomendar parkas pesadas de lana a un cliente ubicado en una ciudad de clima tropical en pleno verano, ni trajes de lino claro en un día lluvioso de invierno. De igual forma, las empresas carecen de esquemas estructurados de **Fidelización y Gestión de Relaciones con el Cliente (CRM)**. Los clientes fieles que realizan compras recurrentes reciben exactamente el mismo trato y precios que un visitante esporádico, desaprovechando las mecánicas de gamificación (acumulación de puntos por compras, niveles de membresía con privilegios escalonados e incentivos de recompra) que han demostrado elevar el valor de vida del cliente (Customer Lifetime Value - CLV) en más del 40%.

F. Desconexión Transaccional y Dispersión de Métodos de Pago

Por último, el ecosistema transaccional se encuentra fragmentado. En el canal físico, las cajas registradoras suelen operar de espaldas al sistema digital, con terminales de cobro independientes que requieren doble captura manual de datos. En el canal digital, existe desconfianza hacia los pagos con tarjeta de crédito en línea por temor a fraudes informáticos, y los métodos locales emergentes (como el Código QR Interoperable impulsado por el Banco Central de Bolivia) carecen frecuentemente de automatización e integración directa vía webhooks con las plataformas web y móviles, exigiendo que el usuario envíe una captura de pantalla por WhatsApp para que un operador verifique manualmente el depósito.

. . . . . .

1.4.3 Matriz de Análisis Causa-Efecto (Diagrama de Ishikawa)

Para estructurar formalmente las causas raíz que originan la ineficiencia, pérdida de clientes y altos costos operativos en el negocio de moda masculina, se presenta la siguiente matriz analítica basada en las 5 dimensiones fundamentales de la ingeniería de procesos (Métodos, Tecnología, Personal, Materiales/Inventario y Clientes):

Métodos y Procesos	**Ausencia de flujos de reserva anticipada de prendas por sucursal.**	**Clientes insatisfechos por desabastecimiento presencial y colapso de probadores.**	**Abandono de la tienda física y pérdida irreversible de ventas potenciales.**
Métodos y Procesos	Registro de inventario limitado al último costo unitario de compra.	Distorsión del valor real del activo e inexactitud en el cálculo del margen de utilidad.	Incumplimiento contable-tributario y toma de decisiones financieras erróneas.
Tecnología y Sistemas	Falta de herramientas virtuales de verificación de calce y prueba de prendas.	Incertidumbre dimensional del cliente al adquirir prendas en la plataforma digital.	Tasa de devoluciones superior al 30% y saturación logística de post-venta.
Tecnología y Sistemas	Dispersión de sistemas no integrados entre tienda web, app móvil y cajas POS.	Desincronización de existencias y desfase de inventario tras ventas concurrentes.	Venta de productos sin stock real (overselling) y descontento del cliente.

Inventario y Stock	Gestión aislada por sucursal sin visibilidad geográfica consolidada.	Desbalance de existencias: acumulación de sobrestock en sucursal A y quiebres en sucursal B.	Pérdida de rentabilidad por liquidaciones forzosas y costos innecesarios de almacenamiento.
Personal y Atención	Atención reactiva sin conocimiento previo de las preferencias o reservas del cliente.	Tiempos de atención prolongados en tienda y búsqueda manual en bodegas.	Baja productividad del personal de sala de ventas y congestión en sucursales.
Cliente y Mercado	Catálogos estáticos no personalizados y carencia de programas de fidelización estructurados.	Experiencia de compra impersonal y ausencia de incentivos de recurrencia.	Alta tasa de deserción hacia competidores y bajo valor de vida del cliente (CLV).

. . . . . .

1.4.4 Impacto y Consecuencias de la Inacción

Si una cadena de indumentaria masculina decide posponer la modernización integral de sus sistemas y procesos, las repercusiones a corto y mediano plazo amenazan la viabilidad misma de la organización:

1. **Erosión Progresiva de Márgenes Financieros**: La combinación de devoluciones masivas en el canal digital (con fletes inversos absorbidos por el comercio) junto a márgenes distorsionados por valuaciones empíricas de inventario destruyen la rentabilidad neta.
2. **Obsolescencia Acelerada del Stock**: La ropa masculina responde a ciclos de temporada rígidos. La mercadería que no rota oportunamente por falta de visibilidad multi-sucursal se devalúa drásticamente, obligando a remates por debajo del costo promedio de adquisición.
3. **Pérdida de Posicionamiento ante Competidores Nativos Digitales**: Los nuevos actores del mercado que implementan plataformas omnicanales, pagos ágiles y entregas rápidas capturan con celeridad al segmento de clientes masculinos jóvenes y profesionales, caracterizados por su alto poder adquisitivo y poca tolerancia a la ineficiencia.
4. **Degradación de la Imagen de Marca**: Vender en línea prendas que luego se revelan sin stock en bodega, o entregar tallas incompatibles que frustran las expectativas del comprador, socava la credibilidad corporativa en redes sociales y entornos digitales.

. . . . . .

1.4.5 Formulación de la Solución Integral con FashionStore

Frente a este escenario complejo, **FashionStore** se formula como una solución tecnológica transformadora y omnicanal, diseñada específicamente para cerrar la brecha físico-digital en la moda masculina mediante:

- **Convergencia Omnicanal**: Unificación de la tienda web responsiva (Angular), la aplicación móvil interactiva (Flutter) y los puntos de venta presenciales (POS) bajo un único backend centralizado (FastAPI) con base de datos unificada (PostgreSQL).
- **Control Riguroso de Inventarios Multi-Sucursal**: Sincronización atómica de existencias, control de transferencias y cálculo automático del **costo promedio ponderado** tras cada recepción de compra a proveedores.
- **Flujo Innovador de Reserva Presencial**: Habilitación del servicio de apartado digital de prendas con fecha y hora programadas, permitiendo que el cliente acuda a la sucursal seleccionada a vivir una experiencia ágil de prueba física.

- **Reducción Drástica de Devoluciones mediante Realidad Aumentada**: Implementación del vestidor virtual interactivo sobre dispositivos móviles, dotando al usuario de certezas visuales respecto al calce, caída y textura del producto.
- **Personalización Extrema con IA Contextual**: Algoritmos de recomendación capaces de sintetizar clima meteorológico local, colorimetría y preferencias personales en atuendos coordinados.
- **Gamificación Comercial**: Mecánicas de fidelización que transforman cada compra en puntos, ascensos de nivel jerárquico y descuentos exclusivos, maximizando la retención de clientes.

. . . . . .

1.5 Alcance

1.5.1 Módulos del Sistema

A continuación se detalla exhaustivamente cada uno de los **19 módulos funcionales** que conforman la plataforma FashionStore. Cada especificación define la descripción operativa orientada al control de acceso basado en roles (RBAC), la segregación de responsabilidades, la estructura de datos clave (con identificadores, tipos de datos, hashes de seguridad y restricciones de integridad) y las funcionalidades clave correspondientes.

. . . . . .

1.5.1.1 Módulo M01: Autenticación, Autorización y Seguridad (RBAC)

Descripción General:

Permite gestionar el acceso seguro y controlado a la plataforma mediante un esquema estricto de Control de Acceso Basado en Roles (RBAC). Cada usuario que interactúa con FashionStore (Administrador, Encargado de Sucursal, Cajero, Personal de Logística/Inventario, Cliente o Proveedor) accede exclusivamente a las vistas, endpoints y operaciones autorizadas para su perfil específico. Implementa mecanismos de alta seguridad informática, garantizando la confidencialidad, integridad y no repudio de la información transaccional.

Datos principales:

- **`id_usuario`**: Entero autoincremental (PK).
- **`nombres`** y **`apellidos`**: Cadenas alfanuméricas (obligatorias).
- **`email`**: Cadena alfanumérica única (Unique Constraint, indexado para búsquedas rápidas de autenticación).
- **`password_hash`**: Cadena alfanumérica de 60 caracteres generada mediante algoritmo criptográfico **`bcrypt`** con factor de coste balanceado (nunca se almacena texto en claro).
- **`rol`**: Enumerador tipado (**`ADMINISTRADOR`**, **`ENCARGADO_SUCURSAL`**, **`CAJERO`**, **`LOGISTICA`**, **`CLIENTE`**, **`PROVEEDOR`**).
- **`estado_cuenta`**: Enumerador tipado (**`ACTIVO`**, **`INACTIVO`**, **`BLOQUEADO_POR_INTENTOS`**, **`PENDIENTE_VERIFICACION`**).
- **`intentos_fallidos`**: Entero corto (contador acumulativo de intentos consecutivos de inicio de sesión erróneos).
- **`ultimo_acceso`**: Timestamp de fecha y hora exacta del último login exitoso.

- **`token_jwt`**: Token web JSON firmado con clave asimétrica RSA/HMAC conteniendo claims de identidad y expiración configurable (60 minutos para web, 30 días para móvil mediante refresh token).
- **`id_sucursal_asignada`**: Entero (FK opcional, vincula al empleado con una sucursal física determinada).

Funcionalidades clave:

- Inicio de sesión seguro con validación estricta de credenciales hash y emisión de tokens JWT.
- Cierre de sesión con invalidación y revocación inmediata de tokens en lista de exclusión (Blacklist en memoria/Redis).
- Bloqueo preventivo y automático de la cuenta tras 5 intentos fallidos consecutivos de autenticación.
- Recuperación automatizada de contraseña mediante generación de tokens de un solo uso (OTP) temporizados enviados al correo electrónico registrado.
- Cambio de contraseña por el propio usuario con validación estricta de robustez (mínimo 8 caracteres, combinación de mayúsculas, minúsculas, números y símbolos especiales).
- Gestión y mantenimiento de perfiles y asignación de roles por parte del Administrador General.
- Registro en bitácora de auditoría de cada intento de acceso (login/logout, IP de origen, User-Agent, fecha/hora y resultado).

. . . . . .

1.5.1.2 Módulo M02: Gestión de Sucursales y Ciudades

Descripción General:

Administra la estructura física, geográfica y organizacional de la cadena de tiendas de indumentaria. Permite definir las ciudades de cobertura y las diferentes sucursales habilitadas en cada territorio, estableciendo su localización geográfica precisa, capacidades de atención, números de contacto y personal asignado, sirviendo como eje espacial para el inventario, las reservas presenciales y los despachos.

Datos principales:

- **`id_sucursal`**: Entero autoincremental (PK).
- **`id_ciudad`**: Entero (FK hacia la tabla **`Ciudades`**).
- **`nombre_sucursal`**: Cadena alfanumérica (ej. "Sucursal Equipetrol", "Sucursal Calacoto").
- **`direccion_fisica`**: Texto descriptivo de la ubicación y referencias viales.
- **`latitud`** y **`longitud`**: Valores de precisión decimal (GPS) para geolocalización y cálculo de distancias.
- **`telefono_contacto`**: Cadena numérica.
- **`horario_apertura`** y **`horario_cierre`**: Formato Time (ej. 09:00:00 a 21:00:00).
- **`capacidad_probadores`**: Entero (número de vestidores físicos disponibles para atención concurrente de reservas).
- **`estado_operativo`**: Enumerador tipado (**`OPERATIVA`**, **`EN_MANTENIMIENTO`**, **`CERRADA_TEMPORALMENTE`**).

Funcionalidades clave:

- Registro, modificación, baja lógica y consulta detallada de ciudades y regiones de cobertura.

- CRUD completo de sucursales físicas con registro georreferenciado (latitud/longitud).
- Asignación de encargados de tienda y personal operativo a sucursales específicas.
- Parametrización de días laborables, feriados y franjas horarias de atención para reservas presenciales.
- Consulta interactiva y pública de sucursales cercanas para el cliente desde la app móvil o web mediante geolocalización.
- Habilitación o inhabilitación temporal de sucursales para recepción de pedidos o reservas.

. . . . . .

1.5.1.3 Módulo M03: Gestión de Productos, Categorías y Atributos

Descripción General:

Centraliza la administración del catálogo de prendas de vestir masculinas. Gestiona de forma desacoplada las categorías (camisas, trajes, pantalones, calzado, accesorios), marcas comerciales, modelos, colecciones y los atributos intrínsecos de cada prenda, destacando el manejo de colores como atributos multivaluados y tallas normalizadas, junto con la gestión multimedia de imágenes en alta definición.

Datos principales:

- **`id_producto`**: Entero autoincremental (PK).
- **`codigo_sku_base`**: Cadena alfanumérica única estandarizada (ej. "SHIRT-SLIM-001").
- **`nombre`**: Cadena alfanumérica (ej. "Camisa Oxford Slim Fit").
- **`descripcion`**: Texto detallado con especificaciones textiles, composición de tela y cuidados.
- **`id_categoria`**: Entero (FK hacia **`Categorias`**).
- **`id_marca`**: Entero (FK hacia **`Marcas`**).
- **`precio_venta_base`**: Decimal con dos dígitos de precisión (monto estándar en moneda nacional).
- **`genero`**: Enumerador fijado en **`MASCULINO`**.
- **`tallas_disponibles`**: Arreglo / relación relacional normalizada (**`S`**, **`M`**, **`L`**, **`XL`**, **`XXL`**, **`38`**, **`40`**, **`42`**).
- **`colores_multivaluados`**: Arreglo estructurado de colores con nombre comercial y código hexadecimal (ej. **`[{"nombre": "Azul Marino", "hex": "#000080"}, {"nombre": "Blanco", "hex": "#FFFFFF"}]`**).
- **`imagenes_url`**: Arreglo de URLs hacia repositorio de almacenamiento en la nube (S3/Cloudinary).
- **`estado_publicacion`**: Enumerador (**`BORRADOR`**, **`PUBLICADO`**, **`DESCATALOGADO`**).

Funcionalidades clave:

- Registro y actualización de fichas técnicas de prendas con validación de código SKU único.
- Gestión taxonómica de categorías y subcategorías jerárquicas de indumentaria masculina.
- Configuración de tallas y asignación de paletas de colores multivaluados con código visual Hexadecimal.
- Carga múltiple y optimización de imágenes de productos para web y móvil (vistas frontal, dorsal y detalle).
- Activación, suspensión y descatalogación de productos del catálogo público.
- Vinculación de productos con modelos 3D y texturas para el vestidor virtual en Realidad Aumentada.

. . . . . .

1.5.1.4 Módulo M04: Gestión de Temporadas y Colecciones

Descripción General:

Permite planificar, organizar y calendarizar las campañas comerciales de la moda masculina. Clasifica los productos dentro de colecciones estacionales (Primavera-Verano, Otoño-Invierno, Temporada Escolar/Universitaria, Black Friday, Ediciones Especiales de Gala), facilitando la rotación estratégica del inventario, la aplicación de políticas de precios diferenciadas y la alimentación de datos contextuales al motor de Inteligencia Artificial.

Datos principales:

- **`id_temporada`**: Entero autoincremental (PK).
- **`nombre_temporada`**: Cadena alfanumérica (ej. "Primavera - Verano 2026", "Línea Ejecutiva Invierno").
- **`codigo_campaña`**: Cadena alfanumérica única (ej. "SS-2026").
- **`fecha_inicio`**: Fecha de lanzamiento oficial de la temporada.
- **`fecha_fin`**: Fecha de cierre y liquidación de temporada.
- **`descripcion`**: Texto explicativo de la línea conceptual de la colección.
- **`estado_temporada`**: Enumerador tipado (**`PLANIFICADA`**, **`VIGENTE`**, **`EN_LIQUIDACION`**, **`FINALIZADA`**).
- **`descuento_liquidacion_sugerido`**: Porcentaje numérico aplicable al fin de temporada.

Funcionalidades clave:

- Creación, modificación y cierre formal de temporadas y colecciones comerciales.
- Asociación masiva o individual de productos del catálogo a una o múltiples temporadas.
- Activación automatizada de promociones y descuentos por cambio de temporada comercial.
- Segmentación de colecciones para campañas de marketing y recomendaciones personalizadas por IA.
- Reporte comparativo de ventas y rentabilidad entre colecciones históricas.

. . . . . .

1.5.1.5 Módulo M05: Gestión de Proveedores y Abastecimiento

Descripción General:

Administra la cadena de suministro y aprovisionamiento de mercadería. Gestiona la información jurídica, comercial y crediticia de los proveedores nacionales e internacionales que confeccionan y distribuyen las prendas de vestir, controlando el catálogo de productos suministrados, condiciones de entrega, plazos de crédito y contratos comerciales.

Datos principales:

- **`id_proveedor`**: Entero autoincremental (PK).
- **`razon_social`**: Cadena alfanumérica oficial.
- **`nit_o_identificacion`**: Cadena alfanumérica única (Número de Identificación Tributaria).
- **`nombre_contacto`**: Cadena alfanumérica del ejecutivo de cuentas.
- **`telefono`** y **`email`**: Datos de contacto directo.
- **`direccion_sede`**: Texto físico y país/ciudad de origen.

- **`terminos_pago`**: Enumerador (**`CONTADO`**, **`CREDITO_30_DIAS`**, **`CREDITO_60_DIAS`**, **`CONSIGNACION`**).
- **`estado`**: Enumerador (**`ACTIVO`**, **`INACTIVO`**, **`OBSERVADO`**).

Funcionalidades clave:

- Registro de proveedores con validación de identificación tributaria y datos comerciales.
- Vinculación de proveedores con líneas de productos y categorías textiles autorizadas.
- Registro y seguimiento de órdenes de compra con detalle de cantidades y precios acordados.
- Evaluación de cumplimiento en tiempos de entrega y control de calidad de lotes recibidos.
- Consulta de historial consolidado de compras y pagos por proveedor.

. . . . . .

1.5.1.6 Módulo M06: Gestión de Inventario Multi-Sucursal y Costos Ponderados

Descripción General:

Constituye el núcleo logístico y financiero de FashionStore. Controla en tiempo real las existencias físicas de cada variante de producto (producto, talla, color) en cada una de las sucursales de la cadena. Implementa de manera estricta el algoritmo de **Costo Promedio Ponderado** tras cada entrada de mercadería por compra, almacenando paralelamente el **último costo unitario de adquisición** de acuerdo con las especificaciones académicas del proyecto. Registra la trazabilidad completa de transferencias inter-sucursales, mermas y ajustes de inventario.

Datos principales:

- **`id_inventario`**: Entero autoincremental (PK).
- **`id_sucursal`**: Entero (FK hacia **`Sucursales`**).
- **`id_producto`**: Entero (FK hacia **`Productos`**).
- **`talla`**: Cadena normalizada (ej. "M", "42").
- **`color`**: Cadena del color específico (ej. "Azul Marino").
- **`stock_fisico`**: Entero (cantidad física existente en la tienda).
- **`stock_reservado`**: Entero (cantidad apartada temporalmente para clientes con reservas presenciales vigentes).
- **`stock_disponible`**: Entero calculado (**`stock_fisico - stock_reservado`**).
- **`stock_minimo`**: Entero (umbral de reorden para alertas de desabastecimiento).
- **`stock_maximo`**: Entero (límite de almacenamiento físico en la sucursal).
- **`ultimo_costo_unitario`**: Decimal(10,2) (valor unitario pagado en el lote de compra más reciente).
- **`costo_promedio_ponderado`**: Decimal(10,2) (recalculado matemáticamente según: $CPP_\{nuevo\} = \frac{(Stock_\{anterior\} \times CPP_\{anterior\}) + (Cant_\{ingreso\} \times Costo_\{ingreso\})}{Stock_\{anterior\} + Cant_\{ingreso\}}$).
- **`ubicacion_almacen`**: Cadena descriptiva (ej. "Estante B, Percha 4").

Funcionalidades clave:

- Control de existencias por variante de producto segregado por cada sucursal de la empresa.
- Actualización atómica del inventario y bloqueo temporal durante transacciones concurrentes de compra o reserva.
- Cálculo matemático automático del Costo Promedio Ponderado y registro del Último Costo Unitario en cada recepción.

- Solicitud, autorización, despacho y recepción de transferencias de prendas entre sucursales.
- Generación de alertas automáticas al encargado de tienda cuando el stock cruza el umbral mínimo configurado.
- Registro de bitácora inmutable de movimientos de kardex (entradas, salidas, ajustes por merma y transferencias) con fecha, hora, responsable y justificación.

. . . . . .

1.5.1.7 Módulo M07: Catálogo Digital Omnicanal (Web y Móvil)

Descripción General:

Expone el portafolio de prendas de vestir masculinas a través de la plataforma web responsiva (Angular) y la aplicación móvil nativa (Flutter). Proporciona a los clientes y visitantes una experiencia de navegación sumamente rápida, fluida y visualmente atractiva, con capacidades de búsqueda predictiva, filtros multifacéticos en tiempo real y verificación inmediata de disponibilidad de stock por sucursal física.

Datos principales:

- Parámetros de consulta y filtrado: Categoría, Rango de Precios, Talla, Color, Temporada, Marca, Ciudad y Sucursal.
- Esquema de serialización JSON enriquecido: SKU, nombre, descripción formateada, galería multimedia, variantes de talla/color, disponibilidad en sucursal y atributos de compatibilidad con realidad aumentada.
- Índices de base de datos relacionales y full-text search para optimización de tiempos de respuesta (< 2 segundos).

Funcionalidades clave:

- Visualización jerárquica del catálogo de moda masculina con paginación optimizada e infinitescroll.
- Filtrado multidimensional combinando simultáneamente temporada, talla, color, marca y precio.
- Selector interactivo de sucursal con consulta inmediata de existencias por talla y color específico.
- Ficha detallada de prenda con guía de tallas anatómicas masculinas y recomendaciones de cuidado textil.
- Funcionalidad de marcado y gestión de lista de prendas favoritas (Wishlist) sincronizada entre web y app móvil.

. . . . . .

1.5.1.8 Módulo M08: Vestidor Virtual con Realidad Aumentada (RA)

Descripción General:

Implementa el probador virtual tridimensional mediante tecnología de Realidad Aumentada sobre la aplicación móvil Flutter. Mediante la cámara del smartphone del usuario, el sistema superpone modelos 3D y texturas hiperrealistas de prendas masculinas sobre la silueta del cliente en tiempo real, permitiéndole evaluar la caída, el tamaño proporcional y la armonía visual antes de tomar la decisión de reserva o compra.

Datos principales:

- `id_recurso_ra`: Entero autoincremental (PK).

- **`id_producto`**: Entero (FK hacia **`Productos`**).
- **`modelo_3d_url`**: Cadena URI apuntando al archivo tridimensional optimizado (**`.glb`** / **`.gltf`**).
- **`texturas_url`**: Mapeo JSON de texturas diferenciadas por color de prenda.
- **`puntos_anclaje_corporal`**: Configuración JSON de mapeo anatómico (hombros, torso, cadera, cuello).
- **`escala_referencia`**: Factor numérico decimal de calibración de proporciones anatómicas.

Funcionalidades clave:

- Detección de silueta humana y extremidades mediante visión artificial (ARCore / ML Kit).
- Proyección tridimensional en tiempo real de chaquetas, camisas, blazers y camisetas sobre el cuerpo del usuario.
- Cambio dinámico de color y textura de la prenda proyectada sin interrumpir la transmisión de la cámara.
- Captura fotográfica del outfit proyectado con opción de guardado local o compartición en redes sociales.
- Integración directa desde la vista de Realidad Aumentada hacia el Carrito de Compras o el Flujo de Reserva.

. . . . . .

1.5.1.9 Módulo M09: Comparador Inteligente de Outfits

Descripción General:

Herramienta de análisis visual y estilístico que permite al cliente contrastar de manera interactiva hasta tres combinaciones completas de ropa masculina (outfits) lado a lado en una sola pantalla. El módulo desglosa visualmente cada combinación (prenda superior, prenda inferior, calzado y accesorios), calcula los subtotales, compara los precios de cada conjunto y visualiza los beneficios de fidelización o descuentos aplicables a cada alternativa.

Datos principales:

- **`id_comparacion`**: Entero autoincremental (PK o identificador de sesión local).
- **`id_cliente`**: Entero (FK opcional, almacenable en perfil).
- **`outfit_1`**, **`outfit_2`**, **`outfit_3`**: Estructuras JSON compuestas conteniendo: lista de SKUs de prendas seleccionadas, tallas, colores elegidos, precio unitario de cada una y monto acumulado.
- **`total_outfit_1`**, **`total_outfit_2`**, **`total_outfit_3`**: Montos decimales calculados.
- **`fecha_comparacion`**: Timestamp.

Funcionalidades clave:

- Interfaz interactiva de arrastrar y soltar (drag & drop) o selección rápida para ensamblar hasta tres combinaciones.
- Visualización paralela lado a lado de las tres combinaciones con detalles de precio, color y disponibilidad.
- Cálculo automático y comparativo del costo total de cada outfit, indicando la opción más económica y la de mejor valoración.
- Reemplazo dinámico de componentes individuales (ej. sustituir una camisa formal por una casual y observar el cambio de precio global).

- Botón de acción directa para transferir el outfit ganador íntegramente al carrito de compras o a una reserva presencial.

. . . . . .

1.5.1.10 Módulo M10: Sistema de Reservas de Prendas en Sucursal

Descripción General:

Regula el proceso omnicanal mediante el cual un cliente preselecciona un lote de prendas desde la plataforma digital para probárselas físicamente en una sucursal determinada. El sistema reserva el stock temporalmente para evitar que se venda a terceros, genera un código de reserva único con código QR, y notifica inmediatamente a la sucursal de destino para que el encargado prepare y ordene las prendas en un probador asignado.

Datos principales:

- **`id_reserva`**: Entero autoincremental (PK).
- **`codigo_reserva`**: Cadena alfanumérica única de 8 caracteres (ej. "RSV-2026-X8K").
- **`id_cliente`**: Entero (FK hacia **`Clientes`**).
- **`id_sucursal`**: Entero (FK hacia **`Sucursales`**).
- **`fecha_reserva`**: Fecha en que fue emitida la solicitud.
- **`fecha_programada_visita`**: Fecha y hora fijada por el cliente para acudir a la tienda.
- **`hora_limite_vigencia`**: Timestamp en el que la reserva expira y el stock reservado se libera automáticamente si el cliente no se presenta.
- **`estado_reserva`**: Enumerador tipado (**`PENDIENTE_PREPARACION`**, **`PREPARADA_EN_SUCURSAL`**, **`CLIENTE_PRESENTE`**, **`CONCRETADA_VENTA`**, **`CANCELADA_POR_CLIENTE`**, **`EXPIRADA`**).
- **`items_reserva`**: Tabla detalle con: **`id_producto`**, **`talla`**, **`color`**, **`cantidad`**, **`precio_al_momento`**.

Funcionalidades clave:

- Generación de reservas multi-prenda indicando sucursal, fecha y rango de hora de visita presencial.
- Bloqueo atómico temporal del inventario disponible de la sucursal asignada durante la vigencia de la reserva.
- Generación de comprobante digital de reserva con código QR escaneable por el personal de tienda.
- Notificación en tiempo real y panel de control para que el Encargado de Sucursal marque la reserva como "Preparada en Percha/Probador".
- Confirmación de llegada del cliente mediante lectura del QR de la reserva y transición de estados del proceso.
- Liberación automática del stock inmovilizado mediante tarea programada si la reserva vence sin asistencia del cliente.

. . . . . .

1.5.1.11 Módulo M11: Carrito de Compras Omnicanal

Descripción General:

Gestiona el almacenamiento temporal y la persistencia de las prendas seleccionadas por el cliente para su adquisición digital. Opera de manera sincronizada entre la versión web y la aplicación móvil

para usuarios autenticados, o mediante almacenamiento local seguro para invitados. Realiza validaciones dinámicas de existencias, cálculo automatizado de subtotales, impuestos y aplicación de promociones o descuentos de fidelización.

Datos principales:

- **`id_carrito`**: Entero autoincremental (PK).
- **`id_cliente`**: Entero (FK hacia **`Clientes`**, nulo para carritos anónimos gestionados por UUID de sesión).
- **`items_carrito`**: Estructura detalle con: **`id_producto`**, **`talla`**, **`color`**, **`cantidad`**, **`precio_unitario`**, **`id_sucursal_despacho`**.
- **`subtotal`**: Decimal(10,2).
- **`monto_descuento`**: Decimal(10,2).
- **`monto_total`**: Decimal(10,2).
- **`fecha_ultima_actualizacion`**: Timestamp.

Funcionalidades clave:

- Incorporación, edición de cantidades y remoción de prendas con validación instantánea de disponibilidad en stock.
- Persistencia del carrito en base de datos para usuarios registrados, permitiendo iniciar la selección en la app móvil y finalizarla en la web.
- Cálculo automático de importes, subtotales, descuentos promocionales y costo de envío según la modalidad elegida.
- Detección de variaciones de stock o precio antes de avanzar a la pasarela de pago.
- Vaciado automático o conversión directa del carrito en una Orden de Venta formal.

. . . . . .

1.5.1.12 Módulo M12: Venta Digital y Checkout

Descripción General:

Controla el flujo completo de formalización de la compra en los canales digitales (web y móvil). Gestiona la selección del método de entrega (envío a domicilio o retiro gratuito en sucursal física - Click & Collect), la captura de datos fiscales para la facturación, la confirmación de la transacción con la pasarela de pagos y la emisión del comprobante digital de compra.

Datos principales:

- **`id_orden_venta`**: Entero autoincremental (PK).
- **`numero_factura_o_nota`**: Cadena alfanumérica única (ej. "FACT-DIG-00452").
- **`id_cliente`**: Entero (FK hacia **`Clientes`**).
- **`tipo_entrega`**: Enumerador (**`RETIRO_EN_SUCURSAL`**, **`DELIVERY_A_DOMICILIO`**).
- **`id_sucursal_retiro`**: Entero (FK opcional hacia **`Sucursales`** en caso de retiro presencial).
- **`direccion_envio`**: Texto descriptivo estructurado en caso de delivery a domicilio.
- **`monto_subtotal`**: Decimal(10,2).
- **`costo_envio`**: Decimal(10,2).
- **`descuento_aplicado`**: Decimal(10,2).

- **`monto_total_pagado`**: Decimal(10,2).
- **`estado_orden`**: Enumerador (**`PENDIENTE_PAGO`**, **`PAGADO_CONFIRMADO`**, **`EN_PREPARACION`**, **`EN_CAMINO`**, **`ENTREGADO`**, **`ANULADO`**).
- **`id_transaccion_pago`**: Cadena (identificador devuelto por la pasarela de pagos).
- **`fecha_hora_venta`**: Timestamp de formalización.

Funcionalidades clave:

- Asistente de Checkout de pasos secuenciales: Selección de Entrega -> Datos de Facturación -> Pago -> Confirmación.
- Validación final de inventario con bloqueo transaccional atómico antes de invocar la pasarela de cobro.
- Descuento definitivo e inmediato del stock en la sucursal de despacho tras la confirmación exitosa del pago.
- Generación de comprobante digital de venta en formato PDF descargable y envío automatizado al correo del cliente.
- Consulta y seguimiento del estado de la orden en tiempo real por parte del cliente (Tracking).

. . . . . .

1.5.1.13 Módulo M13: Punto de Venta Presencial (POS)

Descripción General:

Módulo web operativo diseñado para los cajeros de las sucursales físicas. Optimizado para pantallas táctiles y teclado numérico rápido, permite registrar ventas directas de mostrador a clientes que acuden a la tienda, leer códigos de barras, asociar clientes registrados para otorgarles puntos o registrar ventas a clientes de paso (consumidor final), procesar cobros mixtos y emitir comprobantes de venta físicos de forma inmediata.

Datos principales:

- **`id_venta_pos`**: Entero autoincremental (PK).
- **`id_sucursal`**: Entero (FK hacia **`Sucursales`**).
- **`id_cajero`**: Entero (FK hacia **`Usuarios`**).
- **`id_cliente`**: Entero (FK hacia **`Clientes`**, o ID de "Cliente Casual / Consumidor Final").
- **`numero_ticket`**: Cadena única consecutiva de la caja física.
- **`metodo_pago_utilizado`**: Enumerador (**`EFECTIVO`**, **`TARJETA_DEBITO`**, **`TARJETA_CREDITO`**, **`QR_SIMPLE`**, **`PAGO_MIXTO`**).
- **`monto_recibido`** y **`monto_cambio`**: Decimales para transacciones en efectivo.
- **`total_venta`**: Decimal(10,2).
- **`fecha_hora_transaccion`**: Timestamp de emisión en caja.

Funcionalidades clave:

- Búsqueda veloz de prendas mediante lectura de código de barras o teclado predictivo por SKU.
- Conversión inmediata de una reserva física previa en una venta POS definitiva con un solo clic.
- Registro de ventas con identificación de cliente para asignación de puntos de fidelización o emisión a consumidor final.

- Soporte para cobros combinados o fraccionados (ej. parte en efectivo y parte con código QR o tarjeta).
- Actualización en tiempo real del inventario físico de la sucursal y registro del movimiento contable en caja.
- Cierre y arqueo de caja por turno de trabajo con consolidación de ventas por método de pago.

. . . . . .

1.5.1.14 Módulo M14: Pasarela de Pagos Electrónicos (Stripe / PayPal)

Descripción General:

Proporciona la infraestructura de cobro digital para ventas en línea, integrando APIs estandarizadas de pasarelas internacionales de pago (Stripe y PayPal) operando en entornos de prueba (Sandbox) conforme a las restricciones del examen. Garantiza el cumplimiento de normativas de seguridad PCI-DSS mediante tokenización de tarjetas (evitando el almacenamiento de datos sensibles de tarjetas en la base de datos propia) y gestión asíncrona de eventos de pago vía Webhooks.

Datos principales:

- **`id_transaccion`**: Entero autoincremental (PK).
- **`pasarela`**: Enumerador (**`STRIPE`**, **`PAYPAL`**).
- **`payment_intent_id`**: Identificador único de cobro devuelto por la pasarela (ej. **`pi_3MtwLwLkdIwHu7ix28a3tqPa`**).
- **`cliente_pasarela_id`**: Identificador de cliente generado en el proveedor de pago.
- **`monto_transaccion`**: Decimal(10,2).
- **`moneda`**: Cadena ISO (**`USD`**, **`BOB`**).
- **`estado_pago`**: Enumerador (**`INICIADO`**, **`REQUIERE_ACCION_3DS`**, **`APROBADO`**, **`RECHAZADO`**, **`REEMBOLSADO`**).
- **`payload_webhook`**: Registro JSON de la respuesta asíncrona recibida del servidor de pagos.
- **`fecha_creacion`**: Timestamp.

Funcionalidades clave:

- Creación de intenciones de pago (PaymentIntents) seguras con verificación 3D Secure contra fraudes.
- Renderizado de componentes seguros de captura de tarjeta (Stripe Elements / Botones SDK de PayPal).
- Procesamiento y validación criptográfica de firmas de eventos Webhook para confirmar pagos en segundo plano.
- Manejo de reintentos y notificaciones de error ante transacciones denegadas o fondos insuficientes.
- Soporte para reembolsos directos hacia la pasarela en caso de cancelaciones autorizadas.

. . . . . .

1.5.1.15 Módulo M15: Gestión de Tipos y Métodos de Pago

Descripción General:

Centraliza la administración de los diferentes medios de pago admitidos en FashionStore tanto para ventas presenciales en tiendas físicas como para compras en los canales digitales. Controla los parámetros operativos, recargos, comisiones, cuentas bancarias de destino y la habilitación del

mecanismo de pago por Código QR Interoperable (norma BCB / QR Simple), garantizando el cuadre de ingresos y la conciliación contable.

Datos principales:

- **`id_tipo_pago`**: Entero autoincremental (PK).
- **`nombre_medio`**: Cadena alfanumérica (**`EFECTIVO`**, **`TARJETA_LOCAL`**, **`QR_SIMPLE`**, **`TRANSFERENCIA`**, **`STRIPE_INTERNACIONAL`**, **`PAYPAL`**).
- **`canal_aplicable`**: Enumerador (**`SOLO_PRESENCIAL`**, **`SOLO_DIGITAL`**, **`OMNICANAL`**).
- **`requiere_comprobante_adjunto`**: Booleano (para transferencias o pagos QR manuales).
- **`comision_porcentaje`**: Decimal(5,2).
- **`cuenta_contable_asociada`**: Cadena alfanumérica para conciliación financiera.
- **`estado_activo`**: Booleano.

Funcionalidades clave:

- Parametrización y activación/desactivación de medios de cobro admitidos por canal de venta.
- Generación de códigos QR dinámicos con monto exacto y referencia única de orden para pagos inmediatos.
- Módulo de verificación y conciliación de transferencias bancarias para el personal de administración.
- Registro detallado del flujo monetario por cada tipo de cobro para arqueos y auditorías financieras.
- Reporte consolidado de comisiones deducidas por procesadores de pago externos.

. . . . . .

1.5.1.16 Módulo M16: Sistema de Fidelización Gamificado

Descripción General:

Constituye uno de los tres componentes diferenciadores de FashionStore. Diseñado para maximizar la recurrencia de compra, retención de clientes y el valor de vida del cliente (CLV). Implementa dinámicas de juego basadas en la acumulación de puntos por compras realizadas, asignación automática de niveles de membresía jerárquicos (Bronce, Plata, Oro, Diamante) con beneficios y descuentos crecientes, otorgamiento de insignias digitales por hitos comerciales y un motor de canje de puntos por descuentos en futuras compras.

Datos principales:

- **`id_fidelizacion`**: Entero autoincremental (PK).
- **`id_cliente`**: Entero (FK única hacia **`Clientes`**).
- **`puntos_actuales`**: Entero acumulativo de puntos disponibles para canje.
- **`puntos_historicos_totales`**: Entero acumulativo total (no disminuye al canjear, determina el nivel).
- **`nivel_membresia`**: Enumerador tipado (**`BRONCE`**, **`PLATA`**, **`ORO`**, **`DIAMANTE`**).
- **`porcentaje_descuento_nivel`**: Decimal(5,2) (ej. Bronce: 0%, Plata: 5%, Oro: 10%, Diamante: 15%).
- **`insignias_obtenidas`**: Arreglo JSON de insignias desbloqueadas con fecha de obtención (**`[{"codigo": "FIRST_BUY", "nombre": "Primer Outfit", "fecha": "2026-09-01"}]`**).

- **`fecha_ultimo_ascenso`**: Timestamp de cambio de nivel.

Funcionalidades clave:

- Asignación automática de puntos tras confirmación de compras efectivas (ej. 1 punto por cada 10 Bs consumidos).
- Evaluación periódica de puntos acumulados y promoción automática a niveles superiores de membresía:
  - **Bronce**: 0 a 499 puntos. Beneficios: acceso al catálogo y vestidor virtual.
  - **Plata**: 500 a 1,499 puntos. Beneficios: 5% de descuento permanente y acceso prioritario a reservas.
  - **Oro**: 1,500 a 3,999 puntos. Beneficios: 10% de descuento permanente y asesoría de estilo personalizada.
  - **Diamante**: 4,000+ puntos. Beneficios: 15% de descuento permanente, envíos gratis y acceso a colecciones exclusivas.
- Asignación de insignias digitales por logros específicos (ej. "Caballero Impecable": compra de 3 trajes formales; "Explorador RA": 5 usos del vestidor virtual; "Omnicanal": compra en web y en sucursal física).
- Redención de puntos acumulados en el carrito de compras como saldo a favor en dinero real.
- Panel visual interactivo del perfil del cliente mostrando barra de progreso hacia el siguiente nivel e insignias ganadas.

. . . . . .

1.5.1.17 Módulo M17: Asistente de Estilo con IA Contextual y Comandos de Voz

Descripción General:

Componente diferenciador de vanguardia que eleva la experiencia de usuario a un nivel de asesoría de imagen personalizada. Utiliza algoritmos de Inteligencia Artificial (integrados vía API externa, ej. OpenAI GPT / Google Gemini) para sugerir combinaciones completas de prendas analizando cuatro variables en tiempo real: 1) Clima actual de la ciudad del usuario obtenido mediante integración con la API de OpenWeatherMap, 2) Historial de compras y preferencias de estilo registradas, 3) Reglas de colorimetría y etiqueta masculina, y 4) Disponibilidad real de existencias en tiendas. Además, incorpora procesamiento de lenguaje natural (NLP) para permitir la búsqueda de prendas y la generación de reportes ejecutivos mediante comandos de voz en la app móvil.

Datos principales:

- **`id_solicitud_ia`**: Entero autoincremental (PK).
- **`id_cliente`**: Entero (FK hacia **`Clientes`**).
- **`temperatura_clima`**: Decimal(4,1) (°C devueltos por la API meteorológica).
- **`condicion_clima`**: Cadena (ej. "Lluvioso", "Soleado", "Nublado", "Frío intenso").
- **`estilo_ocasion`**: Enumerador (**`FORMAL_OFICINA`**, **`CASUAL_URBANO`**, **`EVENTO_GALA`**, **`DEPORTIVO`**, **`FIN_DE_SEMANA`**).
- **`paleta_colorimetria`**: Cadena asignada al perfil (ej. "Invierno Frío", "Otoño Cálido").
- **`prompt_generado`**: Texto del prompt contextual enviado al modelo de lenguaje.
- **`outfit_recomendado`**: Estructura JSON con lista de SKUs de prendas sugeridas justificadas por la IA.

- `audio_comando_url`: URI opcional de almacenamiento temporal del audio procesado en búsquedas por voz.

Funcionalidades clave:

- Consulta automatizada a la API del clima (OpenWeatherMap) según la geolocalización o ciudad seleccionada por el usuario.
- Generación de recomendaciones contextuales de atuendos completos (ej. "Para este día de 14°C y llovizna en La Paz, te sugerimos un abrigo de paño azul marino combinado con pantalón de gabardina plomo").
- Algoritmo de filtrado que cruza las recomendaciones de la IA exclusivamente con productos disponibles en stock activo.
- Módulo de búsqueda en catálogo por comando de voz en la app móvil (transcripción de audio a texto e interpretación de intención mediante NLP).
- Generación automatizada de resúmenes y reportes ejecutivos verbales para el administrador (ej. "Dime las ventas de trajes de hoy en Santa Cruz").

. . . . . .

1.5.1.18 Módulo M18: Reportes, Analítica y Dashboards Empresariales

Descripción General:

Proporciona herramientas de inteligencia de negocios y analítica descriptiva para la alta gerencia y los encargados de tienda. Consolida los datos transaccionales de ventas, inventarios, compras a proveedores y reservas, presentándolos en gráficos interactivos y métricas clave de desempeño (KPIs) para sustentar la toma de decisiones estratégicas de abastecimiento y ventas.

Datos principales:

- Métricas consolidadas: Volumen total de ventas en moneda nacional, ticket promedio de compra, rotación de inventario por sucursal, índice de conversión de reservas en ventas efectivas, productos de mayor y menor rotación (Top/Bottom 10), margen de contribución bruto por categoría, y tasa de retención de clientes por nivel de fidelización.
- Filtros analíticos: Por rango de fechas, sucursal, categoría textil, temporada comercial y método de pago.

Funcionalidades clave:

- Cuadro de mando (Dashboard) gerencial en tiempo real con tarjetas de métricas e indicadores de rendimiento.
- Reporte analítico de rotación y valoración de existencias consolidando Costo Promedio Ponderado vs Último Costo de Compra.
- Gráficos comparativos de ventas presenciales en sucursales vs ventas en canal digital (web y móvil).
- Tasa de efectividad de reservas: porcentaje de reservas presenciales concretadas en compras reales en probador.
- Exportación de informes estructurados a formatos Excel (XLSX) y PDF para fines contables y de auditoría.

. . . . . .

1.5.1.19 Módulo M19: Logística de Envíos y Delivery

Descripción General:

Administra la distribución física y entrega de pedidos en la modalidad de compra digital con despacho a domicilio. Modela la integración con servicios de mensajería y empresas de delivery del medio local (emulando plataformas como Yaigo, Yummy o PedidosYa), calculando dinámicamente las tarifas de transporte mediante algoritmos geodésicos basados en distancia kilométrica, peso volumétrico de las prendas y zonas urbanas de cobertura, permitiendo además el rastreo del despacho en tiempo real.

Datos principales:

- **`id_envio`**: Entero autoincremental (PK).
- **`id_orden_venta`**: Entero (FK hacia **`OrdenesVenta`**).
- **`id_sucursal_origen`**: Entero (FK hacia **`Sucursales`** desde donde se despacha el paquete).
- **`latitud_destino`** y **`longitud_destino`**: Coordenadas geográficas del domicilio del comprador.
- **`distancia_km`**: Decimal(6,2) (calculada mediante algoritmo geodésico Haversine).
- **`peso_real_kg`**: Decimal(5,2) (peso físico de las prendas seleccionadas).
- **`peso_volumetrico_kg`**: Decimal(5,2) (calculado según: **\frac{Largo \times Ancho \times Alto}{5000}**).
- **`tarifa_envio`**: Decimal(10,2) (monto final cobrado por el transporte).
- **`nombre_repartidor`** y **`telefono_repartidor`**: Datos del personal o empresa de logística asignada.
- **`estado_envio`**: Enumerador (**`ASIGNADO`**, **`RECOLECTADO_EN_TIENDA`**, **`EN_TRANSITO`**, **`ENTREGADO`**, **`FALLIDO_AUSENCIA`**).
- **`codigo_seguimiento`**: Cadena alfanumérica de rastreo público (Tracking).

Funcionalidades clave:

- Algoritmo automatizado de tarificación de envíos combinando tarifa base, costo por kilómetro y peso volumétrico del paquete.
- Asignación inteligente de la sucursal de origen más cercana al domicilio del cliente que posea el stock completo de la orden.
- Módulo de asignación de pedidos a repartidores o empresas aliadas de delivery.
- Rastreo en tiempo real del estado del despacho para el cliente desde la app móvil o sitio web.
- Registro de confirmación de entrega con captura de firma digital o código OTP proporcionado por el cliente al recibir el paquete.

. . . . . .

1.5.2 Requisitos Funcionales

Requisitos Funcionales Base del Enunciado (RF01 – RF25)

ID	Requisito Funcional	Módulo Asociado
RF01	El sistema deberá permitir registrar clientes con sus datos personales completos (nombres, apellidos, correo electrónico, teléfono, fecha de nacimiento y dirección).	M01
RF02	El sistema deberá permitir gestionar usuarios y roles jerárquicos (Administrador, Encargado de Sucursal, Cajero, Personal de Logística, Cliente, Proveedor) con asignación granular de privilegios.	M01

RF03	El sistema deberá administrar múltiples ciudades y sucursales físicas, incluyendo localización geográfica, horarios y capacidades operativas.	M02
RF04	El sistema deberá permitir gestionar el catálogo de productos de indumentaria masculina con sus atributos: SKU, nombre, descripción, precio base, marca, modelo e imágenes.	M03
RF05	El sistema deberá gestionar tallas normalizadas, colores (como atributos multivaluados con código Hexadecimal), categorías jerárquicas y temporadas comerciales.	M03 / M04
RF06	El sistema deberá gestionar proveedores comerciales con su información fiscal, condiciones de pago y catálogo de prendas suministradas.	M05
RF07	El cliente deberá poder consultar el catálogo de prendas de vestir desde la plataforma web y la aplicación móvil, con navegación fluida y fichas técnicas completas.	M07
RF08	El cliente y los operadores deberán poder consultar la disponibilidad de stock de una prenda (talla y color) por sucursal en tiempo real.	M06 / M07
RF09	El cliente deberá poder seleccionar múltiples prendas de diferentes tallas y colores para incorporarlas en una solicitud de reserva física.	M10
RF10	El sistema deberá registrar y gestionar reservas de prendas, asociándolas a una sucursal física, fecha y horario programado de visita presencial.	M10
RF11	El sistema deberá notificar en tiempo real las reservas a la sucursal correspondiente para que el encargado prepare y ordene las prendas en probadores.	M10
RF12	El sistema deberá permitir consultar y actualizar el estado de una reserva (Pendiente, Preparada, Atendida, Concretada en Venta, Cancelada, Expirada).	M10
RF13	La aplicación móvil deberá permitir utilizar el vestidor virtual mediante realidad aumentada para proyectar prendas sobre la silueta del cliente con la cámara.	M08
RF14	El cliente deberá poder agregar productos al carrito de compras con validación atómica de existencias y cálculo dinámico de totales.	M11
RF15	El cliente deberá poder completar compras digitales desde la plataforma web seleccionando método de pago y modalidad de entrega (retiro en tienda o envío).	M12
RF16	El cliente deberá poder completar compras digitales desde la aplicación móvil con idénticas capacidades transaccionales que en el entorno web.	M12
RF17	El cajero de sucursal deberá poder registrar ventas presenciales desde el punto de caja (POS), asociando prendas, cliente y emitiendo comprobantes.	M13
RF18	El sistema deberá permitir cobros presenciales en punto de caja mediante efectivo, tarjetas de débito/crédito y códigos QR.	M13 / M15
RF19	El sistema deberá integrar una pasarela de pago electrónico internacional (Stripe / PayPal en modo Sandbox) para compras digitales seguras.	M14
RF20	El sistema deberá actualizar automáticamente el inventario tras cada venta (presencial o digital), descontando las existencias por sucursal, talla y color.	M06
RF21	El sistema deberá controlar las existencias por sucursal, registrando el stock físico, stock reservado, stock mínimo, último costo unitario y costo promedio ponderado.	M06
RF22	El sistema deberá registrar todos los movimientos de kardex de inventario (entradas por compra, salidas por venta, mermas y transferencias inter-sucursales).	M06
RF23	El sistema deberá gestionar temporadas comerciales y colecciones de moda, permitiendo categorizar y filtrar las prendas de vestir asociadas.	M04
RF24	El sistema deberá permitir generar reportes consolidados de ventas e inventario con filtros por sucursal, fechas, categorías y temporadas.	M18
RF25	El sistema deberá proporcionar al menos una funcionalidad basada en inteligencia artificial orientada a la recomendación personalizada de prendas.	M17

. . . . . .

Requisitos Funcionales Personalizados y Diferenciadores (RF26 – RF38)

ID	Requisito Funcional	Módulo Asociado
RF26	El sistema deberá permitir al cliente comparar visualmente hasta tres combinaciones de ropa (outfits) lado a lado en pantalla con desglose de precios y descuentos.	M09
RF27	El sistema deberá asignar automáticamente puntos de fidelización al cliente por cada compra formalizada, calculados en función del monto total pagado.	M16
RF28	El sistema deberá gestionar niveles jerárquicos de fidelización (Bronce, Plata, Oro, Diamante) con políticas de descuentos crecientes por nivel.	M16

RF29	El sistema deberá otorgar insignias y distinciones digitales al cliente por cumplimiento de hitos de compra y uso recurrente de la plataforma.	M16
RF30	El sistema deberá permitir al cliente canjear puntos de fidelización acumulados por descuentos monetarios directos en el proceso de compra.	M16
RF31	El Asistente de IA deberá recomendar outfits contextuales analizando en tiempo real la temperatura y condiciones climáticas de la ciudad del usuario vía API meteorológica.	M17
RF32	El Asistente de IA deberá considerar el historial de compras previas, colorimetría y preferencias de estilo del cliente para afinar las recomendaciones.	M17
RF33	La aplicación móvil deberá permitir realizar búsquedas de prendas en el catálogo y solicitar reportes de gestión mediante comandos de voz en lenguaje natural.	M17
RF34	El sistema deberá calcular dinámicamente la tarifa de entrega a domicilio para compras digitales considerando distancia kilométrica, peso y volumen de la orden.	M19
RF35	El sistema deberá registrar una bitácora inmutable de accesos y eventos de seguridad con información de usuario, dirección IP, dispositivo y fecha/hora.	M01
RF36	El sistema deberá generar tableros de control ejecutivos (dashboards) con métricas en tiempo real de ventas, rotación de inventarios y retención de clientes.	M18
RF37	El sistema deberá permitir la conversión directa e inmediata de una reserva física de prendas en una venta efectiva dentro del punto de caja POS.	M10 / M13
RF38	El sistema deberá permitir al cliente consultar su panel de gamificación: saldo de puntos, nivel de membresía actual, barra de progreso e insignias obtenidas.	M16

. . . . . .

1.5.3 Requisitos No Funcionales

ID	Categoría	Especificación del Requisito No Funcional	Criterio de Verificación / Métrica
RNF01	**Seguridad**	Las contraseñas deberán ser almacenadas cifradas mediante algoritmo `bcrypt` con factor de coste ≥ 12. Los tokens de sesión JWT deberán utilizar firmas criptográficas seguras. No se admitirá almacenamiento de contraseñas en texto plano ni de datos sensibles de tarjetas (cumplimiento normativo PCI-DSS). Toda la comunicación web y móvil deberá realizarse obligatoriamente mediante protocolo cifrado HTTPS/TLS 1.3.	Inspección de base de datos y auditoría de tráfico SSL/TLS.
RNF02	**Rendimiento**	Las consultas de búsqueda y filtrado en el catálogo digital deberán responder en un tiempo inferior a 2.5 segundos bajo condiciones normales de concurrencia. Las operaciones de cálculo de inventario y reservas atómicas no deberán exceder los 800 milisegundos.	Pruebas de estrés y benchmarking con Apache JMeter / k6.
RNF03	**Disponibilidad**	El sistema deberá estar desplegado en infraestructura de nube pública, garantizando un índice de disponibilidad operativa del 99% en régimen 24/7, permitiendo transacciones simultáneas desde clientes web y móviles.	Monitor de uptime en plataforma de nube (Railway / Render).
RNF04	**Escalabilidad**	La arquitectura del backend y la base de datos relacional PostgreSQL deberán permitir la incorporación de nuevas ciudades, sucursales físicas y categorías de productos sin requerir reestructuraciones en el esquema de datos ni refactorización de código fuente.	Pruebas de inserción masiva y diseño normalizado 3FN.
RNF05	**Usabilidad**	Las interfaces gráficas de usuario deberán ser completamente responsivas, adaptativas y ceñidas a los lineamientos de diseño moderno (Material Design en Angular y Flutter),	Pruebas de usabilidad

		garantizando una experiencia intuitiva tanto en pantallas de escritorio, tablets como smartphones.	en múltiples resoluciones de pantalla.
RNF06	**Mantenibilidad**	El código fuente del backend y frontend deberá estructurarse en paquetes modulares desacoplados siguiendo los principios de diseño SOLID y arquitectura limpia (Clean Architecture / Capas), facilitando la localización de fallos, extensibilidad y pruebas unitarias.	Revisión de código y métricas de acoplamiento y cohesión.
RNF07	**Interoperabilidad**	El backend desarrollado con FastAPI deberá exponer la totalidad de sus servicios mediante una API REST estandarizada, documentada exhaustivamente mediante esquemas interactivos OpenAPI / Swagger UI.	Inspección de documentación Swagger en `/docs`.
RNF08	**Compatibilidad Multiplataforma**	La aplicación móvil desarrollada en Flutter/Dart deberá garantizar compilación y compatibilidad transparente para sistemas operativos móviles Android (versión 8.0+) e iOS (versión 13+).	Pruebas de ejecución en emuladores y dispositivos físicos.
RNF09	**Integridad Transaccional**	Todas las operaciones críticas de actualización de existencias, reservas y pagos deberán ejecutarse bajo el principio de transaccionalidad ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) en la base de datos PostgreSQL, evitando anomalías por condiciones de carrera (Race Conditions).	Pruebas de concurrencia transaccional simultánea.
RNF10	**Despliegue en Nube**	Conforme a las restricciones académicas institucionales, el sistema no deberá operar en entorno local (localhost) durante las presentaciones y defensas, debiendo desplegarse de manera accesible al público en servidores en la nube con dominio público y código QR de acceso directo.	Validación de URL pública y lectura de código QR.
RNF11	**Trazabilidad y Auditoría**	El sistema deberá mantener un registro inmutable en base de datos de cada acción administrativa, modificación de inventario, cambio de precios y eventos de autenticación, registrando usuario, timestamp, IP y estado previo/posterior.	Consulta de tabla de auditoría en base de datos.

. . . . . .

1.5.4 Actores del Sistema

Actor	Tipo	Plataforma de Acceso	Responsabilidades y Acciones Principales en el Sistema
Cliente	Humano / Externo	Web (Angular) + Móvil	Consulta catálogo, filtra por sucursal/temporada, utiliza el vestidor virtual con Realidad Aumentada, realiza reservas presenciales de prendas, adquiere productos mediante compra digital, utiliza el comparador de outfits, recibe recomendaciones contextuales de IA, acumula puntos de fidelización y canjea beneficios.

		(Flutter)	
Administrador General	Humano / Interno	Web (Angular)	Control total del sistema: gestión de usuarios y asignación de roles (RBAC), alta de ciudades y sucursales, administración del catálogo de prendas, gestión de temporadas y colecciones, configuración de proveedores, consulta de inventario global consolidado, supervisión de dashboards y métricas de negocio.
Encargado de Sucursal	Humano / Interno	Web (Angular)	Recepción de reservas presenciales asignadas a su tienda, apartado físico de prendas en perchas y probadores, registro de la atención del cliente, control de disponibilidad local, solicitud y recepción de transferencias de inventario, y reporte de mermas.
Cajero	Humano / Interno	Web (Angular - Módulo POS)	Búsqueda rápida de productos por SKU o código de barras, conversión de reservas presenciales en ventas definitivas, procesamiento de cobros físicos y digitales (efectivo, tarjetas, QR), emisión de comprobantes de venta y cierre de turno de caja.
Personal de Logística	Humano / Interno	Web (Angular)	Registro de entradas de mercadería por compras a proveedores, verificación física de lotes, registro de costos unitarios y cálculo de costo promedio ponderado, preparación de pedidos digitales para delivery y despacho a repartidores.
Proveedor	Humano / Externo	Web (Angular - Portal Proveedor)	Consulta de órdenes de compra emitidas a su nombre, confirmación de disponibilidad de lotes, registro de información de nuevas líneas textiles y asociación a colecciones estacionales.
Pasarela de Pagos (Stripe / PayPal)	Sistema Externo / API	Servicio en la Nube	Procesamiento seguro de cobros digitales, tokenización de medios de pago, verificación de fondos, aprobación/rechazo de transacciones y emisión de eventos asíncronos vía Webhooks.
Servicio de Inteligencia Artificial	Sistema Externo / API	Servicio en la Nube (OpenAI / Gemini)	Procesamiento de solicitudes de recomendación de estilo en lenguaje natural, análisis de compatibilidad estética, interpretación de comandos de voz y generación de reportes gerenciales automatizados.
Servicio Meteorológico (OpenWeatherMap)	Sistema Externo / API	Servicio en la Nube	Provisión en tiempo real de datos de temperatura, precipitaciones y estado del clima según la ciudad geolocalizada del usuario para enriquecer las sugerencias del asistente de estilo.

. . . . . .

1.5.5 Planificación de Iteraciones (Ciclos de Desarrollo PUDS)

El proyecto se planifica rigurosamente en tres iteraciones evolutivas e incrementales dentro del Proceso Unificado de Desarrollo de Software:

- **— Fundamentos, Seguridad y Catálogo Base (25/08/2026 – 05/09/2026):**
  - Objetivo: Establecer la arquitectura base del software, modelo de datos relacional, módulos de seguridad RBAC, gestión de sucursales, catálogo de prendas masculinas, temporadas comerciales, proveedores y control de inventario con costo unitario y costo promedio ponderado.

- Entregable: Perfil formal del proyecto, Parte I - Fundamentación Teórica, Modelado de Negocio con diagramas de actividad, código backend y frontend desplegado en nube pública con QR de acceso.

- **— Procesos Transaccionales, Reservas y Puntos de Venta (06/09/2026 – 13/09/2026):**
  - Objetivo: Implementar el flujo de reservas presenciales en tienda, el carrito de compras omnicanal, el punto de venta presencial (POS), la integración de pasarelas de pago electrónico (Stripe/PayPal Sandbox) y la actualización automática de inventarios post-venta.
  - Entregable: Documento actualizado con flujos de captura de requisitos, análisis y diseño de transacciones (diagramas de secuencia, estado y navegación), software operativo desplegado.
- **— Diferenciadores Tecnológicos, Inteligencia Artificial y Cierre (14/09/2026 – 20/09/2026):**
  - Objetivo: Integrar el vestidor virtual con Realidad Aumentada en Flutter, el comparador interactivo de outfits, el sistema de fidelización gamificado, el asistente de estilo con IA contextual (clima + preferencias), búsqueda por voz y cuadros de mando analíticos.
  - Entregable: Documento maestro consolidado con los cinco flujos de trabajo PUDS completos, casos de prueba documentados, y defensa final del software el 22/09/2026.

. . . . . .

1.5.6 Matriz de Funcionalidades por Plataforma

Funcionalidad / Módulo	Web Administrativa y Tienda (Angular)	Aplicación Móvil Nativa (Flutter)
Registro, inicio de sesión y recuperación de contraseña	☑	☑
Consulta de catálogo con filtros multidimensionales	☑	☑
Verificación de disponibilidad de stock por sucursal	☑	☑
Vestidor virtual interactivo con Realidad Aumentada (RA)	✕ (Requiere sensores móviles)	☑
Comparador visual interactivo de outfits	☑	☑
Generación de reserva presencial de prendas con QR	☑	☑
Carrito de compras sincronizado y Checkout digital	☑	☑
Pago con tarjeta / PayPal / Stripe en línea	☑	☑
Punto de Venta Presencial (POS) para cajeros	☑	✕
Asistente de estilo con IA contextual (clima y colorimetría)	☑	☑
Búsqueda por comando de voz en lenguaje natural	✕	☑
Perfil de fidelización gamificada (puntos, nivel, insignias)	☑	☑
Gestión de sucursales, productos, categorías y proveedores	☑	✕
Control de inventario multi-sucursal y costos ponderados	☑	✕
Gestión y preparación de reservas presenciales (Encargado)	☑	✕
Tableros de control ejecutivos y reportes analíticos	☑	✕

. . . . . .

1.5.7 Restricciones y Limitaciones del Proyecto

1. Prohibición de Frameworks E-Commerce Preconstruidos: Queda terminantemente vetado el uso de soluciones empaquetadas como Shopify, WooCommerce, PrestaShop, Magento o similares. La totalidad del backend, lógica de negocio y frontends debe desarrollarse a medida.

2. Despliegue Obligatorio en Nube Pública: El sistema no debe operar en `localhost` durante presentaciones ni defensas; debe desplegarse con certificado SSL válido y accesibilidad vía URL pública y código QR.

3. Plazo Estricto de Ejecución: Plazo improrrogable de 4 semanas (25/08/2026 al 22/09/2026), demandando priorización estricta sobre un Producto Mínimo Viable (MVP) completamente robusto.

4. Stack Tecnológico Homologado: Backend en Python con FastAPI, Frontend Web en Angular, Aplicación Móvil en Flutter con Dart, y Base de Datos relacional en PostgreSQL.

5. Estándar de Modelado UML 2.5+: Todos los artefactos metodológicos deben ceñirse rigurosamente a las especificaciones formales de UML 2.5+.

6. Entornos Sandbox para Pagos: Las transacciones financieras electrónicas se procesarán en modo de prueba (Sandbox) sin cobros a tarjetas de crédito reales.

2. Elementos del sistema basado en computadoras

Un sistema basado en computadoras es un conjunto integrado de elementos organizados para alcanzar un objetivo común mediante el procesamiento, almacenamiento, transmisión y transformación de información. En el marco de **FashionStore**, el sistema integra de manera sinérgica el hardware, software, bases de datos relacionales, procedimientos operativos estandarizados, el capital humano y la documentación formal para dar soporte integral al comercio electrónico omnicanal de moda masculina.

2.1 Hardware

El componente de hardware abarca la infraestructura física, dispositivos de procesamiento, periféricos y equipos de telecomunicaciones que sustentan la ejecución de las aplicaciones de FashionStore en sus tres entornos operativos: nube, sucursales físicas y dispositivos de usuarios finales.

2.1.1 Servidor

La arquitectura de servidores de FashionStore adopta un enfoque híbrido en la nube (*Cloud Computing PaaS/IaaS*), garantizando alta disponibilidad, escalabilidad elástica y tolerancia a fallos:

1. **Servidor de Producción Backend (FastAPI / ASGI):**
    - **Plataforma de Alojamiento:** Instancia en contenedor Linux gestionada en nube pública (*Render / Railway / AWS EC2*).
    - **Capacidad de Procesamiento:** 2 vCPU dedicadas (arquitectura x86_64), optimizadas para operaciones asíncronas de E/S de alta concurrencia mediante asyncio.
    - **Memoria Principal (RAM):** 4 GB a 8 GB DDR4 con soporte para asignación dinámica de memoria para workers de Uvicorn y Gunicorn.
    - **Almacenamiento Secundario:** 50 GB a 100 GB NVMe SSD con tasas de transferencia de lectura/escritura superiores a 2.500 MB/s para el sistema base, binarios y logs de rotación.
    - **Conectividad de Red:** Enlace simétrico de 1 Gbps con ancho de banda mensual no medido, dirección IPv4/IPv6 estática pública, y proxy inverso NGINX con terminación segura SSL/TLS 1.3 gestionada con certificados automáticos Let's Encrypt.
2. **Servidor de Base de Datos Dedicado (PostgreSQL 15+):**
    - **Alojamiento:** Instancia administrada en *Supabase Pro / AWS RDS PostgreSQL*.
    - **Capacidad:** 2 vCPU dedicadas, 4 GB de memoria RAM dedicada a pools de conexiones (PgBouncer), 20 GB de almacenamiento SSD transaccional con 3.000 IOPS garantizados.
    - **Mecanismos de Resiliencia:** Copias de seguridad automáticas diarias en caliente (*Point-in-Time Recovery - PITR*) con retención de 7 días y réplicas de solo lectura para consultas masivas de catálogo en temporadas altas.
3. **Almacenamiento de Objetos en la Nube (Object Storage):**
    - Repositorio de alta durabilidad (*AWS S3 / Cloudinary*) para persistencia de activos multimedia: fotografías de prendas en múltiples resoluciones, modelos

tridimensionales (.glb / .gltf) para vestidores virtuales en Realidad Aumentada y comprobantes de facturación.

2.1.2 Cliente

Los dispositivos cliente corresponden a los terminales informáticos mediante los cuales los diferentes actores interactúan con la plataforma:

1. **Dispositivos Móviles de los Clientes (App Móvil Flutter):**
    - **Smartphones y Tablets compatibles:** Dispositivos móviles con arquitectura ARM64 de 64 bits ejecutando Android 8.0 (Oreo) o superior, e iOS 13.0 o superior.
    - **Memoria RAM:** Mínimo 3 GB de RAM (recomendado 4 GB a 8 GB para fluidez en la inicialización de motores de visión artificial y renderizado tridimensional).
    - **Sensores Críticos:** Cámara trasera y frontal de alta resolución (mínimo 12 MP), sensor de profundidad o acelerómetro y giroscopio calibrados, indispensables para el anclaje de planos y detección anatómica requerida por Google ARCore y Apple ARKit en el Vestidor Virtual de Realidad Aumentada.
    - **Conectividad:** Conexión a redes móviles 4G LTE / 5G o redes inalámbricas Wi-Fi 5/6 con acceso continuo a internet.
2. **Estaciones de Trabajo y Computadoras Personales (Navegadores Web):**
    - **Equipos de Clientes y Personal Administrativo:** Computadoras de escritorio o portátiles con procesadores Intel Core i3 / AMD Ryzen 3 o superiores, mínimo 4 GB de memoria RAM, y tarjetas gráficas integradas con soporte para aceleración por hardware WebGL.
    - **Pantallas:** Monitores con resolución mínima de 1366 × 768 píxeles (recomendado Full HD 1920 × 1080) con diseño completamente adaptativo (*responsive layout*).
3. **Terminales de Punto de Venta (POS) en Sucursales Físicas:**
    - **Computadoras de Caja:** Terminales All-in-One (*Todo en Uno*) o computadoras de escritorio compactas ubicadas en los mostradores de cobro de cada tienda física (Equipetrol, Calacoto, El Prado).
    - **Especificaciones:** Procesador Intel Core i5 de 10ª generación o superior, 8 GB de RAM, unidad SSD de 256 GB para arranque ultra-rápido, y puertos USB múltiples para periféricos especializados.

2.1.3 Otros Dispositivos

Para asegurar la sincronización en tiempo real de los procesos físicos en tienda con la plataforma digital, se integran los siguientes periféricos y equipos de red:

Dispositivo	Tipo / Interfaz	Ubicación / Área	Función Operativa Principal

Lectores de Códigos de Barras y Códigos QR	Óptico 2D Láser / USB y Bluetooth	Puntos de Caja POS y Área de Probadores	Lectura instantánea de etiquetas SKU en prendas de vestir y escaneo de códigos QR de reservas presenciales presentados por el cliente desde su smartphone.
Impresoras Térmicas de Tickets / Recibos	Térmica Directa 80 mm / USB - Ethernet	Mostradores de Cobro POS	Emisión inmediata de comprobantes de venta física, facturas de caja y talones de reserva presencial con corte automático de papel (velocidad ≥200extmm/s≥200*extmm/s*).
Terminales de Pago Electrónico (POS Bancario Físico)	POS Físico EMV / Contactless NFC / Wi-Fi	Mostradores de Caja	Procesamiento presencial de cobros mediante tarjetas de débito y crédito con verificación de PIN y soporte para pagos móviles por proximidad.
Tablets de Asistencia en Probadores	Tablet Android 10.5" / Wi-Fi Corporativo	Área de Probadores y Atención de Sucursal	Utilizadas por el Encargado de Sucursal para verificar la llegada de clientes con reserva, confirmar la preparación de prendas y reportar el estado de probadores.
Rúteres y Conmutadores de Red Empresarial	Wi-Fi 6 Gigabit / Dual Band (2.4 / 5 GHz)	Cada Sucursal Física	Enlace de datos seguro con segmentación de VLANs: una red aislada y cifrada para terminales POS e inventario, y una red Wi-Fi de cortesía para clientes en tienda.
Cámaras de Seguridad IP y Control Perimétrico	Cámaras Full HD PoE / Circuito Cerrado	Salas de Venta y Almacenes de Sucursal	Monitoreo visual de la mercadería, disuasión de pérdidas y supervisión de los flujos de clientes en probadores y cajas.

2.2 Software

El ecosistema de software de FashionStore está concebido bajo principios de diseño moderno, desacoplamiento de capas, comunicación asíncrona no bloqueante y estándares abiertos.

2.2.1 Servidor

1. **Sistema Operativo Base:**
    - **Linux Ubuntu Server 22.04 LTS (Jammy Jellyfish):** Distribución de grado empresarial con kernel Linux 5.15/6.x de 64 bits, seleccionada por su extrema estabilidad, soporte de parches de seguridad a largo plazo y óptimo rendimiento en virtualización y contenedores.
2. **Servidor Web y Proxy Inverso:**
    - **NGINX versión 1.24+:** Encargado de la recepción de solicitudes HTTP/HTTPS, balanceo de carga, compresión en tránsito con algoritmos Gzip/Brotli, limitación de tasa (*Rate Limiting*) para mitigar ataques de fuerza bruta en autenticación, y serving estático de la SPA Angular.
3. **Servidor de Aplicaciones y Framework Backend:**
    - **Python 3.11+ / FastAPI 0.110+:** Framework asíncrono de alto rendimiento montado sobre la especificación ASGI (*Asynchronous Server Gateway Interface*).

- **Servidor ASGI:** Uvicorn con worker loop uvloop (basado en libuv), capaz de despachar más de 15.000 solicitudes concurrentes por segundo con latencias inferiores a 10 milisegundos.
- **Capa de Persistencia y ORM:** SQLAlchemy 2.0 con driver asíncrono asyncpg y motor síncrono psycopg2-binary, garantizando transaccionalidad ACID y soporte dual con fallback local a SQLite3.
- **Validación y Serialización:** Pydantic v2 para validación estricta de esquemas de entrada/salida y contratos de datos fuertemente tipados.

4. **Sistema Gestor de Base de Datos Relacional (RDBMS):**
   - **PostgreSQL 15+:** Motor de datos principal de la organización, elegido por su madurez, conformidad estricta con estándares SQL, soporte nativo para campos semiestructurados JSONB, indexación GiST/GIN para búsquedas rápidas en catálogo y extensiones de seguridad.
5. **Servicios de Caché y Gestión Volátil:**
   - **Redis 7.x (en memoria):** Utilizado para almacenamiento transitorio de listas de exclusión de tokens JWT revocados (*Blacklist*), persistencia de carritos anónimos temporales y caché de consultas frecuentes de disponibilidad de stock.

2.2.2 Cliente

1. **Frontend Web (Plataforma E-Commerce y Panel Administrativo):**
   - **Angular 17+ / TypeScript:** Framework robusto para aplicaciones de página única (SPA), estructurado con componentes desacoplados, servicios inyectables reactivos apoyados en RxJS, enrutamiento modular y diseño visual Dark Glassmorphism responsivo ceñido a CSS3 Vanilla y FontAwesome 6.5.
   - **Compatibilidad de Navegadores:** Optimizado para Google Chrome 115+, Microsoft Edge, Mozilla Firefox 115+, Safari 16+ y navegadores móviles Chromium.
2. **Aplicación Móvil Multiplataforma (FashionStore App):**
   - **Flutter 3.22+ / Dart 3.4+:** Framework de desarrollo móvil compilado a código nativo de máquina (ARM64), garantizando tasas de refresco de 60 fps a 120 fps con el motor gráfico Impeller.
   - **Gestión de Estado y Arquitectura:** Arquitectura en capas limpia (*Clean Architecture*) con separación de dominio, datos y presentación, persistencia local con flutter_secure_storage y comunicación REST mediante cliente dio / http.
3. **Módulos Móviles Especializados:**
   - **Librerías de Realidad Aumentada:** Integración de plugins ar_flutter_plugin y google_mlkit_pose_detection para procesamiento de visión artificial y anclaje de mallas 3D sobre la silueta del cliente.
   - **Lector de Códigos y Voz:** Plugins mobile_scanner para lectura de códigos QR y speech_to_text para captura de comandos de voz en lenguaje natural.

2.2.3 Otro Software Adicional

Categoría	Software / Servicio	Proveedor / Licencia	Finalidad en FashionStore
Seguridad Criptográfica	bcrypt (factor de coste ≥12≥12) / python-jose	MIT / Apache 2.0	Cifrado unidireccional no reversible de contraseñas de usuarios y firmado asimétrico de tokens web JSON (JWT HS256).
Pasarela Nacional de Pagos	Libélula Pay API	Propietaria (Bolivia)	Procesamiento de pagos en moneda nacional mediante Códigos QR Interoperables (Simple / BCB) y transferencias bancarias locales.
Pasarelas Internacionales	Stripe API & PayPal REST SDK	Propietaria / Sandbox	Pasarelas de cobro global con soporte de tokenización PCI-DSS compliant, verificación 3D Secure 2.0 y simulación de transacciones con tarjetas de prueba.
Inteligencia Artificial Contextual	OpenAI API (GPT-4o) / Google Gemini 1.5 Flash	REST API Cloud	Motor cognitivo para generación de recomendaciones de outfits personalizados, combinaciones cromáticas y procesamiento de consultas en lenguaje natural.
Telemetría Meteorológica	OpenWeatherMap API	REST API Cloud	Provisión de temperatura, sensación térmica y precipitaciones en tiempo real para las ciudades de Santa Cruz, La Paz y Cochabamba.
Documentación de API	OpenAPI 3.0 & Swagger UI	Integrado en FastAPI	Generación automatizada de esquemas interactivos de prueba para la totalidad de los endpoints REST en la ruta /docs.
Monitoreo y Bitácoras	Loguru / Sentry SDK	MIT / Open Source	Captura centralizada de logs estructurados de auditoría, trazabilidad de excepciones y alertas de seguridad en tiempo real.

2.3 Datos

Los datos constituyen el activo transaccional y operativo más valioso de FashionStore. La información está modelada bajo estándares de normalización relacional (Tercera Forma Normal - 3FN), garantizando la integridad referencial, la no redundancia y el recálculo matemático estricto de costos.

A nivel de dominio, el sistema gestiona las siguientes estructuras de datos clave:

1. **Datos de Estructura Geográfica y Sucursales:**
    - Ciudades de cobertura, códigos postales, sedes operativas, nombres de sucursales físicas, direcciones georreferenciadas con latitud y longitud decimal (GPS), horarios de atención diurna/nocturna y capacidad operativa de probadores concurrentes.
2. **Datos del Catálogo de Moda Masculina:**

- Códigos SKU base universales, nombres de prendas, descripciones técnicas textiles (composición porcentual de algodón, lino, lana, poliéster), marcas, categorías jerárquicas, tallas normalizadas (S, M, L, XL, XXL, 38, 40, 42), colores estructurados con nombre comercial y código hexadecimal visual (#HEX), precios de venta al público y URLs de modelos 3D y galerías fotográficas.

3. **Datos de Temporadas y Campañas Comerciales:**
    - Identificador estacional, nombres de campaña (ej. *Primavera-Verano 2026*, *Línea Ejecutiva Otoño-Invierno*), fechas de inicio y cierre, estado estacional (Planificada, Vigente, Liquidación, Finalizada) y porcentaje sugerido de descuento para remate de temporada.
4. **Datos de Proveedores y Compras:**
    - Razón social, Número de Identificación Tributaria (NIT / CUIT), nombres de ejecutivos de contacto, teléfonos corporativos, correos electrónicos, condiciones de pago negociadas (Contado, Crédito a 30/60 días, Consignación) y registro de órdenes de abastecimiento.
5. **Datos de Inventarios y Valuación Contable (Kardex):**
    - Existencias físicas por sucursal, producto, talla y color; existencias reservadas temporalmente; stock mínimo de seguridad; último costo unitario de adquisición; y el **Costo Promedio Ponderado (CPP)** actualizado mediante asientos inmutables de entrada y salida.
6. **Datos de Reservas Presenciales:**
    - Códigos alfanuméricos de reserva únicos (ej. RSV-2026-X8K), sucursal de destino seleccionada, fecha y rango horario programado para la visita, código QR cifrado de validación, estados de ciclo de vida (Pendiente, Preparada en Probador, Cliente en Tienda, Venta Concretada, Cancelada, Expirada) y lista de prendas solicitadas.
7. **Datos de Ventas y Facturación:**
    - Órdenes de venta digitales y presenciales (POS), número de ticket o factura legal, cliente asociado o consumidor final, desglose de ítems comercializados, desglose de impuestos fiscales (IVA 13% e IT 3%), método de pago utilizado (Efectivo, Tarjeta, QR, Mixto), y montos de descuento por fidelización aplicados.
8. **Datos de Seguridad, Autenticación y Auditoría (RBAC):**
    - Credenciales de usuarios, roles jerárquicos tipados, hashes seguros de contraseñas (bcrypt), estado de la cuenta (Activa, Inactiva, Bloqueada por Intentos), tokens OTP temporizados de 6 dígitos con expiración, y bitácora de auditoría inmutable (usuario, fecha, IP, User-Agent, operación ejecutada).
9. **Datos de Fidelización y Gamificación:**
    - Puntos acumulados por transacciones, nivel jerárquico de membresía (Bronce, Plata, Oro, Diamante), insignias digitales obtenidas por hitos de compra, saldo histórico y registro de canjes por descuentos directos.

2.4 Procesos

Los procesos constituyen la secuencia ordenada de actividades lógicas, transaccionales y de control que transforman las solicitudes de los usuarios en resultados comerciales efectivos dentro de FashionStore.

Los macroprocesos principales del negocio corresponden a:

1. **Proceso de Abastecimiento y Valuación por Costo Promedio Ponderado:**
    - Generación y aprobación de órdenes de compra a proveedores textiles.
    - Recepción física y control de calidad de lotes de prendas en la bodega de sucursal.
    - Registro del costo unitario del lote adquirido e incorporación de costos de flete/aranceles.
    - Recálculo matemático inmediato del Costo Promedio Ponderado (CPP) mediante la ecuación contable oficial:

$$CPPnuevo = Sactual + Qentrada(Sactual \times CPPactual) + (Qentrada \times Cunitario)$$

    - Generación automática del asiento inmutable en el libro Kardex de existencias.
2. **Proceso de Reserva Omnicanal y Prueba Presencial en Probadores:**
    - Exploración del catálogo desde la app móvil o web y selección de prendas, tallas y colores.
    - Selección de la sucursal de preferencia física y programación de fecha/hora de visita.
    - Verificación atómica de disponibilidad y bloqueo temporal de las existencias para evitar ventas cruzadas.
    - Generación del comprobante digital con código QR de reserva.
    - Notificación en tiempo real al Encargado de Sucursal para el apartado físico y preparación de prendas en el probador asignado.
    - Recepción del cliente en tienda, escaneo del QR, prueba física y transición hacia venta o retorno al stock.
3. **Proceso de Venta Presencial en Punto de Venta (POS):**
    - Escaneo de códigos de barra SKU de las prendas elegidas por el cliente en mostrador.
    - Conversión opcional e inmediata de una reserva presencial previa a venta definitiva.
    - Selección de método de cobro: efectivo con cálculo automático de cambio, cobro electrónico con tarjeta mediante terminal física, o generación de Código QR dinámico en pantalla.
    - Emisión del comprobante fiscal de venta e impresión de ticket térmico.
    - Descuento automático del stock en tiempo real y asignación de puntos de fidelización al cliente.
4. **Proceso de Venta Digital y Checkout con Pasarela de Pagos:**
    - Agregación de prendas al carrito de compras sincronizado entre web y móvil.

- Selección de la modalidad de entrega: Retiro gratuito en sucursal (*Click & Collect*) o envío a domicilio (*Delivery*).
- Tarificación dinámica del costo de envío según distancia geodésica (fórmula Haversine) y peso volumétrico.
- Procesamiento de la transacción mediante pasarela de pago en modo Sandbox (Stripe / PayPal) con validación 3D Secure.
- Confirmación del pago, generación de orden de despacho y notificación al cliente vía correo electrónico.

5. **Proceso de Experiencia Inteligente (Vestidor Virtual RA + Asistente IA):**
    - Activación de la cámara del teléfono móvil desde la ficha de producto en Flutter.
    - Detección de silueta anatómica y superposición del modelo 3D de la prenda con textura en tiempo real.
    - Consulta al Asistente de Estilo con IA: análisis de coordenadas de geolocalización, consulta al API meteorológico de temperatura local, análisis del historial de compras y colorimetría del cliente para sugerir combinaciones óptimas de ropa masculina.
6. **Proceso de Fidelización Gamificada y Retención de Clientes:**
    - Acumulación de 1 punto por cada 10 Bolivianos pagados en compras efectivas.
    - Evaluación periódica de acumulación de puntos para ascenso automático de nivel (Bronce ightarrow*ightarrow* Plata ightarrow*ightarrow* Oro ightarrow*ightarrow* Diamante).
    - Concesión de insignias digitales por hitos comerciales y canje de puntos por cupones de descuento directo en el checkout.

2.5 Gente / Usuario

El factor humano es el componente central que dinamiza el sistema. FashionStore clasifica a sus usuarios en roles claramente segregados bajo el esquema de Control de Acceso Basado en Roles (RBAC):

1. Actores Internos

- **Administrador General:**
    - Responsable supremo de la gobernanza informática y comercial de la cadena.
    - Funciones: Alta y baja de usuarios empleados, asignación granular de roles RBAC, desbloqueo administrativo de cuentas bloqueadas por intentos fallidos, configuración de ciudades y sucursales físicas, parametrización de categorías de moda, supervisión de auditorías y análisis de métricas en dashboards ejecutivos.
- **Encargado de Sucursal:**
    - Máxima autoridad operativa dentro de una tienda física determinada (ej. Equipetrol).
    - Funciones: Monitoreo del panel de reservas presenciales asignadas a su tienda, supervisión del apartado físico de prendas en probadores, confirmación de llegada del

cliente mediante escaneo de código QR, reporte de mermas o prendas dañadas, recepción de transferencias inter-sucursales y arqueo de inventario local.

- **Cajero de Sucursal:**
  - Operador responsable del punto de cobro físico en mostrador.
  - Funciones: Autenticación segura en el módulo POS, búsqueda rápida de productos por código SKU, escaneo de prendas, cobro en efectivo con cálculo de cambio, procesamiento de cobros con tarjeta y QR, emisión de comprobantes fiscales y cierre de turno de caja diario.
- **Personal de Logística y Almacén:**
  - Encargados de la recepción de mercadería y despacho de órdenes.
  - Funciones: Verificación física de lotes remitidos por proveedores textiles, registro riguroso de costos unitarios de compra para el recálculo del CPP, empaque y etiquetado de pedidos digitales con despacho a domicilio, y coordinación con repartidores de delivery.

2. Actores Externos

- **Cliente (Consumidor Masculino):**
  - Usuario final de la plataforma web y aplicación móvil.
  - Funciones: Auto-registro de cuenta con validación de identidad, inicio de sesión seguro, exploración y filtrado del catálogo, uso del vestidor virtual en Realidad Aumentada, solicitud de reservas presenciales en sucursales, compras digitales con pasarela de pagos, acumulación de puntos de fidelización y consulta interactiva con el Asistente de Estilo por IA.
- **Proveedores Textiles:**
  - Empresas fabricantes o distribuidoras de indumentaria que abastecen a la cadena.
  - Funciones: Consulta de órdenes de compra emitidas a su nombre, despacho de lotes de mercadería y coordinación de plazos de entrega y términos de crédito comercial.

3. Tecnología para el Desarrollo del Software

La selección tecnológica de FashionStore responde a un análisis exhaustivo de rendimiento, compatibilidad, madurez de la comunidad y adecuación estricta a los requerimientos de la cátedra de Sistemas de Información II.

3.1 Estrategia para el Desarrollo del Software

La estrategia de ingeniería adoptada para el proyecto se fundamenta en un **enfoque evolutivo, iterativo e incremental basado en arquitectura de microservicios desacoplados**:

1. **Desacoplamiento Estricto entre Backend y Frontends:**
   - El núcleo transaccional reside en un backend monolítico modular desarrollado en FastAPI que expone servicios exclusivamente a través de una API REST protegida con tokens JWT.

- Esto permite que el Frontend Web (Angular) y la Aplicación Móvil (Flutter) consuman exactamente la misma lógica de negocio y esquemas de datos, evitando duplicidad de reglas de validación y facilitando pruebas automatizadas independientes.

2. **Desarrollo Dirigido por Casos de Uso (*Use-Case Driven*):**
    - Cada caso de uso identificado se convierte en el eje vertebral que guía la especificación formal, el modelado dinámico (diagramas de comunicación y secuencia), el diseño de clases Boundary-Control-Entity (BCE), la implementación del endpoint REST y su respectivo caso de prueba de caja negra automatizado con Pytest.
3. **Arquitectura Centrada en la Mitigación Temprana de Riesgos:**
    - Durante el Ciclo 1 (Fundamentos) se resolvieron y probaron los componentes de mayor riesgo arquitectónico: la concurrencia en la autenticación RBAC, el bloqueo de cuentas, y la precisión matemática del recálculo de Costo Promedio Ponderado en inventario.
    - Con ello, los ciclos posteriores (transacciones de venta y diferenciadores de Realidad Aumentada / IA) se construyen sobre una base sólida y libre de deuda técnica.
4. **Integración y Despliegue Continuo (CI/CD) en la Nube:**
    - Se descarta el paradigma de "desarrollo local hasta el final". Desde la primera semana, el backend y los frontends se desplegaron en infraestructura cloud accesible mediante URL pública y código QR, satisfaciendo el requerimiento de no operar en localhost.

3.2 Metodología para el Desarrollo del Software

El proyecto adopta formalmente el **Proceso Unificado de Desarrollo de Software (PUDS)** modelado rigurosamente mediante el lenguaje estándar **UML 2.5+**.

3.2.1 Características del PUDS

El PUDS es un proceso de ingeniería de software disciplinado que se caracteriza por tres pilares conceptuales inseparables:

1. **Dirigido por Casos de Uso (*Use-Case Driven*):**
    - Los casos de uso no son meros artefactos de captura de requisitos, sino que actúan como la fuerza motriz de todo el ciclo de vida. Definen qué debe hacer el sistema desde la perspectiva del actor externo, guían el diseño de las clases de análisis y diseño, determinan la estructura de paquetes y proporcionan la base directa para los casos de prueba de aceptación.
2. **Centrado en la Arquitectura (*Architecture-Centric*):**
    - La arquitectura del sistema encarna las decisiones de diseño más trascendentales: la organización en subsistemas y capas, la selección de plataformas y protocolos de comunicación, y los mecanismos de persistencia. El PUDS exige concebir, validar y estabilizar la línea base de la arquitectura durante las fases tempranas para evitar refactorizaciones costosas.

3. **Iterativo e Incremental (*Iterative and Incremental*):**
    - El ciclo de desarrollo se divide en un conjunto de proyectos más pequeños denominados iteraciones. Cada iteración aborda un subconjunto crítico de casos de uso y culmina con la entrega de un producto ejecutable probado (*release interno o externo*). Con cada iteración sucesiva, el software se incrementa modularmente hasta alcanzar la totalidad del alcance.

Fases del Ciclo de Vida PUDS en FashionStore:

- **Fase de Inicio (Inception):** Delimitación de la visión del negocio, justificación económica, identificación de actores principales y especificación del 80% de los casos de uso a nivel general.
- **Fase de Elaboración (Elaboration):** Análisis profundo de casos de uso arquitectónicamente significativos, diseño de la arquitectura en 4 capas, diseño de base de datos relacional y estabilización de la plataforma de desarrollo (Ciclo 1).
- **Fase de Construcción (Construction):** Implementación de la totalidad de módulos transaccionales (ventas, reservas, POS, pasarelas) y diferenciadores (RA, IA, gamificación) con pruebas de regresión continuas (Ciclos 2 y 3).
- **Fase de Transición (Transition):** Despliegue productivo final en la nube, optimización de latencias, auditoría de seguridad y capacitación a usuarios para la defensa final del software.

3.2.2 Características Principales de UML 2.5+

El **Lenguaje de Modelado Unificado (UML versión 2.5+)** es el estándar de modelado visual adoptado para especificar, visualizar, construir y documentar todos los artefactos de FashionStore:

1. **Notación Estandarizada y Semántica Precisa:**
    - Proporciona un vocabulario gráfico universal que elimina ambigüedades interpretativas entre analistas, arquitectos de software, desarrolladores y evaluadores académicos.
2. **Dualidad de Modelado (Estructural y Dinámico):**
    - **Perspectiva Estructural (Aspecto Estático):** Representa los bloques constitutivos del sistema mediante Diagramas de Casos de Uso (con relaciones <<include>>, <<extend>> y generalización), Diagramas de Clases de Análisis (Boundary, Control, Entity), Diagramas de Paquetes con justificación de acoplamiento/cohesión, Diagramas de Componentes y Diagramas de Despliegue Físico.
    - **Perspectiva de Comportamiento (Aspecto Dinámico):** Modela la ejecución temporal y el intercambio de mensajes mediante Diagramas de Actividades con carriles (*swimlanes*), Diagramas de Interacción (Comunicación, Secuencia Transaccional y Diagramas de Tiempo) y Diagramas de Máquinas de Estado para el ciclo de vida de entidades.
3. **Trazabilidad de Ingeniería de Software:**

- Cada elemento de software en el código fuente de FastAPI (app/modules/...) o Flutter tiene un mapeo bidireccional exacto con una clase, paquete o nodo modelado en Enterprise Architect.

3.3 Herramientas de Desarrollo

3.3.1 Software

Herramienta	Tipo / Licencia	Versión	Rol en el Ciclo de Desarrollo
Visual Studio Code	IDE / MIT	1.88+	Entorno de desarrollo principal para el backend FastAPI y frontend web, equipado con extensiones Pylance, Ruff, Angular Language Service, GitLens y Markdown All in One.
Android Studio / Flutter SDK	IDE / Apache 2.0	Hedgehog / 3.22+	Entorno para la compilación, depuración en caliente (*Hot Reload*), análisis estático y perfilado de rendimiento de la app móvil Flutter en emuladores y dispositivos físicos.
Enterprise Architect	Modelado CASE / Propietaria	16.1+	Herramienta formal de modelado UML 2.5+ utilizada para la construcción integral de diagramas y generación programática automatizada mediante scripts COM en Python.
PostgreSQL & DBeaver Community	RDBMS / Gestor DB (GPL)	15+ / 24.0+	Sistema de base de datos relacional y cliente de administración gráfica para modelado ER, ejecución de scripts DDL/DML y optimización de planes de ejecución EXPLAIN ANALYZE.
Git & GitHub	Control de Versiones / Cloud	2.44+	Gestión de código fuente con estrategia de ramificación estructurada (*main*, *develop*, *feature/*), control de versiones y auditoría de commits por desarrollador.
Pytest & HTTPX	Framework de Pruebas / MIT	8.1+ / 0.27+	Suite de testing automatizado para ejecución de pruebas de caja negra, validación de contratos JSON, pruebas de autenticación RBAC y recálculo de costos CPP.
Postman / Swagger UI	Cliente API / Propietario	10.24+	Inspección manual de endpoints, validación de cabeceras de autorización Bearer JWT y pruebas de estrés de llamadas REST.

3.3.2 Hardware de los Desarrolladores

El desarrollo, compilación y pruebas del sistema se ejecutaron sobre las siguientes estaciones de trabajo de alta gama pertenecientes a los integrantes del equipo:

Componente de Hardware	Estación de Desarrollo 1 (Alberto Delgado)	Estación de Desarrollo 2 (Andy Mujica)	Justificación de Capacidad Técnica
Procesador (CPU)	AMD Ryzen 7 5800H (8 núcleos /	Intel Core i7-12700H (14	Capacidad de compilación multi-hilo en segundo plano de Flutter

	16 hilos @ 3.2 GHz - 4.4 GHz Turbo)	núcleos / 20 hilos @ 2.3 GHz - 4.7 GHz Turbo)	y ejecución concurrente de servidores FastAPI y emuladores Android.
Memoria RAM	16 GB DDR4 Dual Channel @ 3200 MHz	32 GB DDR5 Dual Channel @ 4800 MHz	Soporte holgado para entornos de virtualización, contenedores Docker, IDEs pesados y navegadores con herramientas de inspección abiertas.
Almacenamiento	1 TB NVMe M.2 SSD PCIe 3.0 (Lectura 3.200 MB/s)	1 TB NVMe M.2 SSD PCIe 4.0 (Lectura 5.000 MB/s)	Tiempos de carga y guardado instantáneos para bases de datos de prueba, paquetes de Flutter y dependencias de Python.
Unidad Gráfica (GPU)	NVIDIA GeForce RTX 3060 Laptop (6 GB GDDR6)	NVIDIA GeForce RTX 4060 Laptop (8 GB GDDR6)	Aceleración gráfica por hardware para modelado de assets tridimensionales y renderizado fluido en simuladores de Realidad Aumentada.
Pantalla / Visualización	15.6" Full HD (1920 × 1080) @ 144 Hz IPS	16.0" WQXGA (2560 × 1600) @ 165 Hz IPS	Espacio visual suficiente para programación en pantalla dividida (código fuente backend/frontend y modelos UML).
Conexión a Internet	Fibra Óptica Simétrica 150 Mbps (Wi-Fi 6)	Fibra Óptica Simétrica 200 Mbps (Ethernet Gigabit)	Subida ultra-rápida de imágenes Docker, despliegue continuo en la nube y consumo de APIs de IA sin latencia.
Dispositivos Móviles Físicos	Xiaomi Redmi Note 12 Pro+ (8 GB RAM, Android 13, ARCore compatible)	Samsung Galaxy S22 (8 GB RAM, Android 14, ARCore oficial)	Dispositivos físicos reales para pruebas de visión artificial, seguimiento de silueta en vestidor virtual y lectura óptica de códigos QR.

4. Factibilidad Económica y Costos del Proyecto

El análisis de factibilidad económica evalúa la viabilidad financiera del desarrollo, despliegue y mantenimiento de FashionStore en el mercado real de Bolivia, contrastando la inversión de capital requerida frente a los beneficios monetarios y operacionales proyectados a tres años.

pie title Distribución Porcentual de la Inversión Inicial (Bs 62.633)

"Recursos Humanos (Desarrollo)" : 50.6

"Hardware Crítico para Sucursales" : 37.6

"Software y Servicios Cloud (Año 1)" : 11.8

4.1 Análisis de Costos de Desarrollo (Inversión Inicial)

La inversión inicial total requerida para poner en funcionamiento el sistema en una cadena de 3 sucursales urbanas piloto (Santa Cruz de la Sierra, La Paz y Cochabamba) se divide en tres partidas presupuestarias: Recursos Humanos, Hardware Crítico y Software/Servicios Cloud.

4.1.1 Recursos Humanos

El costo de personal se calcula en función de horas-hombre invertidas durante las cuatro semanas del ciclo de vida del proyecto por perfiles profesionales de ingeniería:

Nro.	Rol Profesional	Especialista Asignado	Horas Totales	Tarifa Horaria (Bs)	Tarifa Horaria (USD)	Costo Subtotal (Bs)	Costo Subtotal (USD)
1	**Arquitecto de Software & Backend Lead**	Alberto Delgado	160 hrs	80.00 Bs	11.59 USD	12.800,00 Bs	1.855,07 USD
2	**Desarrollador Frontend & Móvil Lead**	Andy Mujica	160 hrs	80.00 Bs	11.59 USD	12.800,00 Bs	1.855,07 USD
3	**Diseñador UX/UI & Especialista en Modelos 3D RA**	Consultoría Externa	60 hrs	60.00 Bs	8.70 USD	3.600,00 Bs	521,74 USD
4	**Ingeniero de Calidad y Pruebas (QA / Testing)**	Equipo Interno	50 hrs	50.00 Bs	7.25 USD	2.500,00 Bs	362,32 USD
TOTAL	**Recursos Humanos de Desarrollo**	—	**430 hrs**	—	—	**31.700,00 Bs**	**4.594,20 USD**

Nota Cambiaria: Se utiliza la cotización oficial de referencia de 1 USD = 6.90 Bolivianos (Bs).

4.1.2 Hardware Crítico para Operación

Comprende el equipamiento informático indispensable que debe adquirirse para equipar las tres sucursales piloto de la cadena física:

Ítem	Descripción del Equipamiento	Cantidad	Costo Unitario (Bs)	Costo Subtotal (Bs)	Costo Subtotal (USD)
1	**Terminales POS Todo en Uno (All-in-One):** Pantalla táctil 15.6", Intel Core i5, 8 GB RAM, 256 GB SSD para punto de caja.	3 unidades	4.500,00 Bs	13.500,00 Bs	1.956,52 USD
2	**Lectores Ópticos 2D Omnidireccionales:** Escáneres de	3 unidades	450,00 Bs	1.350,00 Bs	195,65 USD

	códigos de barras y códigos QR USB/Bluetooth de alta velocidad.				
3	**Impresoras Térmicas de Tickets POS:** Impresoras térmicas de 80 mm con corte automático y puerto Ethernet.	3 unidades	700,00 Bs	2.100,00 Bs	304,35 USD
4	**Tablets de Asistencia en Probadores:** Tablets Android 10.5" para gestión de reservas presenciales en sala de ventas.	3 unidades	1.400,00 Bs	4.200,00 Bs	608,70 USD
5	**Equipos de Red y Rúteres Gigabit Wi-Fi 6:** Rúteres corporativos con gestión de VLANs para conexión segura de cajas.	3 unidades	800,00 Bs	2.400,00 Bs	347,83 USD
TOTAL	**Hardware Crítico en Sucursales**	—	—	**23.550,00 Bs**	**3.413,05 USD**

4.1.3 Software y Servicios Cloud (Año 1)

Costos de suscripción a plataformas en la nube, dominios, certificados y consumo de APIs de servicios externos proyectados para los primeros 12 meses de operación:

Servicio / Proveedor	**Modalidad / Plan**	**Costo Mensual (USD)**	**Costo Anual (USD)**	**Costo Anual (Bs)**
Servidor de Producción Backend (Railway / Render): Instancia con 2 vCPU y 4 GB RAM en contenedores.	Mensual recurrente	25.00 USD	300.00 USD	2.070,00 Bs
Base de Datos Gestionada PostgreSQL (Supabase Pro): Instancia dedicada con backups automáticos y PITR.	Mensual recurrente	25.00 USD	300.00 USD	2.070,00 Bs
Almacenamiento de Multimedia y Modelos 3D (AWS S3 / Cloudinary): Capacidad de 100 GB y CDN global.	Mensual recurrente	15.00 USD	180.00 USD	1.242,00 Bs
Consumo de APIs de Inteligencia Artificial (OpenAI / Gemini): Cuota mensual estimada de 500.000 tokens.	Mensual por uso	20.00 USD	240.00 USD	1.656,00 Bs
API Meteorológica (OpenWeatherMap): Plan gratuito de 1.000 llamadas/día (suficiente para sucursales piloto).	Plan Free	0.00 USD	0.00 USD	0,00 Bs
Dominio Web Corporativo (fashionstore.bo / .com): Registro anual y gestión DNS administrada.	Pago anual	—	50.00 USD	345,00 Bs
Certificados de Seguridad SSL/TLS Wildcard: Certificados emitidos vía Let's Encrypt con auto-renovación.	Open Source	0.00 USD	0.00 USD	0,00 Bs

TOTAL	**Software y Servicios Cloud (Año 1)**	—	**1.070,00 USD**	**7.383,00 Bs**

Consolidación de la Inversión Inicial

Partida Presupuestaria	Monto en Bolivianos (Bs)	Monto en Dólares (USD)	Porcentaje (%)
Recursos Humanos de Desarrollo	31.700,00 Bs	4.594,20 USD	50.61%
Hardware Crítico en Sucursales	23.550,00 Bs	3.413,05 USD	37.60%
Software y Servicios Cloud (Año 1)	7.383,00 Bs	1.070,00 USD	11.79%
INVERSIÓN INICIAL TOTAL	**62.633,00 Bs**	**9.077,25 USD**	**100.00%**

4.2 Análisis de Beneficios e Impacto Financiero

Para determinar la rentabilidad de la inversión, se proyectaron los flujos de caja netos a tres años considerando un crecimiento anual sostenido del 15% en ventas y un costo de capital (*tasa de descuento - WACC*) del 12% anual:

Indicador Financiero	Valor Calculado	Criterio de Decisión	Diagnóstico de Factibilidad
Valor Actual Neto (VAN a 3 años)	**+314.850,00 Bs** (45.630 USD45.630 USD)	VAN>0VAN>0	**Altamente Favorable:** El proyecto genera un valor económico sustancialmente superior al capital invertido.
Tasa Interna de Retorno (TIR)	**48.25%**	TIR>12% (Tasa de Corte)TIR>12% (Tasa de Corte)	**Excelente Rentabilidad:** La tasa interna de rentabilidad supera en más de cuatro veces el costo de oportunidad financiero.

Periodo de Recuperación de Inversión (Payback)	**7.8 meses**	Payback<12 mesesPayback<12 meses	**Rápida Amortización:** El capital invertido se recupera íntegramente antes de culminar el octavo mes de operación comercial.
Relación Beneficio / Costo (B/C)	**2.85**	B/C>1.0B/C>1.0	Por cada Boliviano invertido en la plataforma, la empresa recupera el boliviano y obtiene 1.85 Bs adicionales de beneficio neto.

4.3 Beneficios Tangibles (Ahorro Anual Proyectado)

Los beneficios tangibles corresponden a ingresos incrementales directos y reducciones comprobables en los costos operativos de la cadena minorista:

1. **Reducción Drástica de Devoluciones Digitales vía Vestidor Virtual RA:**
    - La tasa histórica de devolución en compras de ropa masculina en línea ronda el 30%. Con la verificación anatómica tridimensional en la app Flutter, la tasa de devolución proyectada cae al 8%.
    - **Ahorro Anual Estimado: 38.000,00 Bs/año** (ahorro directo en fletes de logística inversa, reempaque y costos de oportunidad de mercadería inmovilizada).
2. **Eliminación de Quiebres de Stock por Sincronización Multi-Sucursal:**
    - La visibilidad en tiempo real de existencias evita la pérdida de clientes en tienda física, permitiendo derivaciones inmediatas o despachos desde otra sucursal.
    - **Recuperación de Ventas Estimada: 55.000,00 Bs/año**.
3. **Optimización de Compras por Valuación Matemática en CPP:**
    - La sustitución del empirismo contable por el cálculo exacto del Costo Promedio Ponderado previene compras a sobreprecio y sobrestock de temporadas fenecidas.
    - **Ahorro Anual Estimado: 28.000,00 Bs/año**.
4. **Incremento del Ticket Promedio por Asistente de IA y Comparador de Outfits:**

- Las recomendaciones contextuales de atuendos completos (camisa + pantalón + blazer + accesorio) elevan el valor de compra por cliente en un 18%.
- **Ingresos Adicionales Anuales: 72.000,00 Bs/año**.

5. **Ahorro de Horas-Hombre por Automatización de Reservas Presenciales:**
    - Disminución del 45% en el tiempo que los vendedores de piso dedican a buscar prendas en bodegas desordenadas.
    - **Ahorro Operativo Anual: 16.000,00 Bs/año**.

- **BENEFICIO TANGIBLE TOTAL ANUAL: 209.000,00 Bs/año (~30.289,85 USD/año)**.

4.4 Beneficios Intangibles

Los beneficios intangibles representan ventajas estratégicas y competitivas que, si bien son difíciles de cuantificar en un balance contable inmediato, consolidan la supervivencia y liderazgo de FashionStore:

- **Posicionamiento Vanguardista de Marca:** FashionStore se posiciona como la primera tienda de moda masculina en el mercado nacional en introducir probadores en Realidad Aumentada e Inteligencia Artificial contextual, atrayendo al segmento de consumidores masculinos jóvenes y profesionales de mayor poder adquisitivo.
- **Elevación del Índice de Fidelidad y Satisfacción (NPS):** La eliminación de la fricción dimensional en la elección de tallas genera una experiencia de compra gratificante, proyectando un *Net Promoter Score* (NPS) superior a 85 puntos.
- **Gobierno de Datos y Toma de Decisiones Basada en Evidencia:** Los administradores disponen de cuadros de mando consolidados en tiempo real, eliminando decisiones basadas en intuición respecto a qué colecciones producir, rotar o liquidar.
- **Resiliencia Operativa y Reducción del Estrés Laboral:** El flujo programado de reservas distribuye la afluencia de clientes de manera homogénea a lo largo del día, descongestionando los probadores en horas pico y mejorando el clima laboral del personal de tienda.

4.5 Beneficios Esperados

4.5.1 Beneficios Operacionales y de Gestión (Procesos en Sucursales y Tiendas)

- **Sincronización Total de Existencias:** Erradicación del fenómeno de sobreventa (*overselling*) gracias a transacciones atómicas que descuentan el inventario instantáneamente en compras web, móviles o cajas POS.
- **Apartado Físico Previo:** El personal de probadores prepara y plancha las prendas reservadas con antelación, logrando que la atención presencial del cliente sea inmediata al momento de su llegada.
- **Kardex Inmutable y Transparente:** Generación automática del historial de movimientos físicos y valorados de cada producto, facilitando auditorías contables internas y fiscales.

4.5.2 Beneficios Estratégicos y de Inteligencia de Negocios

- **Predicción Dinámica de la Demanda:** Capacidad para correlacionar variables climáticas de cada ciudad con las tendencias de compra semanal, permitiendo redistribuir existencias antes de que ocurran quiebres o sobrestocks estacionales.
- **Segmentación de Clientes por Valor de Vida (*CLV*):** Identificación automática de compradores recurrentes para canalizar promociones personalizadas de alta conversión.

4.5.3 Beneficios para el Capital Humano y de Seguridad

- **Segregación Estricta de Funciones (RBAC):** Garantía de que los cajeros solo operen la caja asignada, los encargados supervisen su tienda local y los administradores gestionen políticas globales, previniendo fraudes internos.
- **Bloqueo Preventivo y Auditoría Continua:** Blindaje contra intrusiones no autorizadas mediante bloqueo por intentos fallidos y bitácoras de auditoría forense con dirección IP y timestamp.

4.5.4 Beneficios de Escalado Tecnológico

- **Escalabilidad Horizontal Económica:** Posibilidad de abrir nuevas sucursales físicas en ciudades como Tarija, Sucre u Oruro simplemente registrando las coordenadas y capacidades en el sistema, sin requerir cambios de software ni licencias adicionales.
- **Interoperabilidad Futura:** Arquitectura modular con API REST documentada que facilita la integración futura de casilleros inteligentes (*smart lockers*) para recojo automatizado o ventas a través de asistentes conversacionales en WhatsApp.

Marco Teórico: Fundamentación Teórica

a) Comercio Electrónico (E-commerce)

Conceptos Generales, Modelos de Negocio y Evolución

El comercio electrónico comprende la compra, venta, comercialización y distribución de bienes, servicios e información a través de redes telemáticas e internet. A lo largo de las últimas dos décadas, ha evolucionado desde los primeros catálogos estáticos en la web 1.0 hasta ecosistemas omnicanales hiperpersonalizados impulsados por inteligencia artificial y computación en la nube.

Dentro de los modelos de negocio tradicionales se distinguen:

- **B2C (Business-to-Consumer)**: Modelo en el cual una empresa comercializa bienes directamente al consumidor individual final. Es el modelo rector de FashionStore.
- **B2B (Business-to-Business)**: Transacciones comerciales entre empresas (ej. compras masivas de FashionStore a sus proveedores textiles).
- **C2C (Consumer-to-Consumer)**: Transacciones entre usuarios particulares mediadas por una plataforma (ej. mercados de subastas o segunda mano).
- **Omnicanalidad (Omnichannel Retail)**: Evolución del modelo tradicional multicanal que unifica de forma transparente la experiencia de compra en tiendas físicas, aplicaciones móviles, plataformas web y redes sociales, compartiendo una base de datos centralizada de inventario, clientes y precios.

. . . . . .

Análisis Comparativo como Usuario: Amazon, Alibaba y Shopify

Dimensión de Análisis	Amazon	Alibaba	Shopify
Modelo Principal	B2C y Marketplace Global directo.	B2B internacional (Alibaba.com) y B2C/C2C (AliExpress/Taobao).	Plataforma SaaS que faculta a marcas independientes para operar sus propias tiendas D2C (Direct-to-Consumer).
Experiencia de Navegación y Búsqueda	Buscador predictivo extremadamente potente, filtros ultra-específicos, opiniones verificadas y algoritmo de sugerencias A9/Rufus.	Orientado a compras mayoristas: búsqueda por cotizaciones, pedido mínimo (MOQ), especificaciones de fábrica y muestras.	Depende del diseño de la marca particular; navegación limpia, minimalista y centrada en la identidad visual de la tienda.
Experiencia en el Carrito y Checkout	Compra en 1-Clic (1-Click Checkout), almacenamiento seguro de múltiples tarjetas y domicilios, envío Prime en 24h.	Negociación directa con proveedores (Trade Assurance), pago por cartas de crédito, transferencias bancarias o tarjetas.	Flujo ultra-optimizado mediante **Shop Pay**, permitiendo checkout en 1 paso con código SMS a nivel global.
Políticas de Devolución y Post-Venta	Política sumamente favorable al usuario con devoluciones automáticas en puntos de acopio sin costo adicional.	Mediación de disputas complejas; las devoluciones físicas internacionales suelen ser lentas y costosas.	Cada comerciante define sus propias políticas de garantía y devolución según su logística particular.

. . . . . .

Análisis Comparativo como Desarrollador: Magento, PrestaShop y WooCommerce

Conforme a los requerimientos de la cátedra, a continuación se analizan las principales plataformas de software libre y código abierto utilizadas en la industria para implementar tiendas en línea:

Dimensión Técnica	Magento (Adobe Commerce)	PrestaShop	WooCommerce
Propósito y Alcance	Plataforma empresarial de gran escala diseñada para multinacionales con catálogos masivos y alta concurrencia.	Solución intermedia orientada a pequeñas y medianas empresas (PyMEs) con catálogos medianos.	Plugin de comercio electrónico de código abierto desarrollado para transformar sitios WordPress en tiendas en línea.
Arquitectura de Software	Arquitectura modular altamente desacoplada en capas, basada en PHP y el framework Zend/Laminas. Soporta microservicios y GraphQL.	Basado en PHP bajo el framework Symfony; arquitectura clásica de componentes y módulos controladores.	Arquitectura basada en los hooks, actions y filters del núcleo de WordPress sobre PHP y MySQL.
Ventajas Principales	Escalabilidad extrema, gestión multi-tienda nativa, multi-moneda, soporte de inventarios multi-sucursal sumamente robusto.	Fácil de administrar para usuarios no técnicos, consumo moderado de recursos en servidor, amplio catálogo de módulos.	Despliegue sumamente veloz, costo de desarrollo inicial muy reducido, millones de temas y plugins compatibles en el ecosistema WordPress.
Desventajas y Complejidad	Curva de aprendizaje muy elevada, requiere servidores dedicados de alto rendimiento (mínimo 8GB RAM, Elasticsearch obligatorio).	Dependencia excesiva de módulos comerciales de pago para funciones clave; migraciones de versiones propensas a errores.	Pierde rendimiento y estabilidad ante bases de datos masivas (más de 20,000 productos o transacciones concurrentes intensas).
Base de Datos	MySQL / MariaDB (con tablas EAV - Entity Attribute Value que aumentan la complejidad relacional).	MySQL / MariaDB (diseño relacional clásico normalizado).	MySQL / MariaDB (utiliza las tablas `wp_posts` y `wp_postmeta`, lo que genera cuellos de botella en consultas complejas).

b) Pasarelas de Pago

Mecanismos de Pago en Línea

El comercio digital moderno exige la coexistencia de múltiples mecanismos transaccionales que equilibren conveniencia para el comprador y seguridad contra fraudes:

1. **Tarjetas de Débito y Crédito**: Siguen siendo el medio de mayor volumen global. La transacción involucra una cadena de actores: el cliente (tarjetahabiente), el banco emisor (Issuing Bank), la pasarela de pago (Payment Gateway), la red adquirente (Visa, Mastercard) y el banco receptor del comercio (Acquiring Bank). El estándar de seguridad internacional **PCI-DSS (Payment Card Industry Data Security Standard)** prohíbe taxativamente que los comercios almacenen el código CVV o el número de tarjeta (PAN) en texto plano, obligando a implementar procesos de **tokenización**.
2. **Códigos QR Interoperables (QR Simple / BCB)**: En el contexto boliviano, el Código QR se ha consolidado como el medio de pago electrónico más popular debido a su inmediatez y costo cero de comisiones directas al usuario. Opera bajo transferencias interbancarias automáticas coordinadas por la Cámara de Compensación y Liquidación (ACH).

3. Transferencias Bancarias Tradicionales: Mecanismo asíncrono donde el cliente transfiere fondos desde su banca móvil y remite una constancia física o digital al comercio para su posterior validación manual o conciliación vía API bancaria.

. . . . . .

Pasarela Nacional: LIBÉLULA

Libélula es una pasarela de pagos tecnológica boliviana desarrollada para resolver las complejidades del comercio electrónico local.

- **Modo de Operación**: Actúa como un concentrador de pagos unificado, integrando en una sola API cobros mediante Códigos QR generados dinámicamente, tarjetas de débito/crédito locales e internacionales, billeteras móviles (Tigo Money) y cobranzas presenciales en redes de cobranza física (Farmacias, supermercados afiliados).
- **Integración Técnica**: Libélula proporciona un entorno de integración mediante APIs REST y bibliotecas cliente. Cuando una orden se genera en el e-commerce, el backend invoca el endpoint de Libélula transmitiendo monto, detalle y cliente; Libélula responde con un enlace a su pasarela segura o devuelve un código QR dinámico con vencimiento programado (ej. 15 minutos).
- **Confirmación por Webhooks**: Una vez que el tarjetahabiente o usuario escanea y confirma el pago en su aplicación bancaria, los servidores de Libélula emiten una petición HTTP POST asíncrona (Webhook) hacia el endpoint configurado en el backend de FashionStore, notificando el éxito de la transacción para que el sistema actualice el estado de la venta y descuente el inventario de manera atómica.

. . . . . .

Pasarelas Internacionales: STRIPE y PAYPAL

- **STRIPE**: Considerada el estándar de facto para desarrolladores de software a nivel global.
  - Entorno Sandbox: Ofrece un entorno de pruebas completo con números de tarjeta simulados para verificar casos de éxito, fondos insuficientes, tarjetas vencidas o validaciones de autenticación reforzada 3D Secure (SCA).
  - Tokenización mediante Stripe Elements: El formulario de tarjeta se renderiza dentro de un iframe seguro alojado en los servidores de Stripe. El backend del e-commerce nunca toca ni procesa los dígitos de la tarjeta; solo recibe un `payment_method_id` seguro.
  - Manejo de Webhooks: Stripe notifica eventos como `payment_intent.succeeded` o `payment_intent.payment_failed` mediante llamadas criptográficamente firmadas con un secreto compartido (Webhook Secret).
- **PAYPAL**: Pionero de los pagos electrónicos en internet.
  - Entorno Sandbox: Plataforma de desarrolladores que permite crear cuentas ficticias de compradores y vendedores con saldos virtuales para pruebas de checkout.
  - Flujo de Integración: Permite integrar botones inteligentes de pago (PayPal Smart Payment Buttons). El usuario es redirigido a una ventana emergente de PayPal donde autoriza la transacción con su saldo o tarjeta vinculada, y el flujo retorna al e-commerce mediante endpoints de captura (capture payment).

. . . . . .

c) Servicios de Delivery y Logística de Última Milla

Operación del Delivery en Aplicaciones Urbanas (Yaigo, Yummy, PedidosYa)

La logística de última milla representa el eslabón final y más sensible del comercio digital. Plataformas como Yaigo, Yummy y PedidosYa operan bajo un modelo de economía colaborativa triangulada:

1. **El Comercio (FashionStore)**: Recibe la orden de compra digital, la valida y empaqueta las prendas en la sucursal física asignada para el despacho.
2. **El Repartidor / Conductor**: Recibe una notificación en su aplicación móvil con los datos de recolección en tienda y destino final del cliente, aceptando el viaje en función de su cercanía.
3. **El Cliente Final**: Visualiza en tiempo real sobre un mapa la localización geográfica del repartidor mediante GPS y recibe un código OTP de seguridad para validar la entrega física del paquete.

. . . . . .

Algoritmos y Variables de Cálculo de Tarifas de Envío

Para determinar el costo exacto del servicio de entrega a domicilio, los sistemas logísticos no aplican valores fijos, sino que emplean funciones matemáticas basadas en las siguientes variables clave:

1. **Distancia Geodésica (D)**: Calculada a partir de las coordenadas de latitud y longitud de la sucursal de despacho (ϕ_1, λ_1) y del domicilio del cliente (ϕ_2, λ_2) mediante la **Fórmula de Haversine**:

$$\Delta \phi = \phi_2 - \phi_1, \quad \Delta \lambda = \lambda_2 - \lambda_1$$

$$a = \sin^2\left(\frac{\Delta \phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta \lambda}{2}\right)$$

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right), \quad D = R \cdot c \quad (R = 6371\text{ km})$$

2. **Peso Volumétrico (P_v)**: En la industria textil y de paquetería, prendas voluminosas (como abrigos pesados o parkas) ocupan espacio crítico en los vehículos de reparto aunque su peso real en kilogramos sea bajo. Se aplica el estándar IATA:

$$P_v = \frac{\text{Largo (cm)} \times \text{Ancho (cm)} \times \text{Alto (cm)}}{5000}$$

El peso facturable (P_f) corresponde al mayor valor entre el peso físico real (P_r) y el peso volumétrico (P_v):

$$P_f = \max(P_r, P_v)$$

3. **Fórmula General de Tarificación Logística**:

$$\text{Tarifa Final} = T_{\text{base}} + (D \times C_{\text{km}}) + (P_f \times C_{\text{peso}}) \times F_{\text{zona}} \times F_{\text{demanda}}$$

Donde:

- T_{base}: Tarifa base fija de arranque (ej. 10.00 Bs).
- C_{km}: Costo por kilómetro recorrido (ej. 2.50 Bs/km).
- C_{peso}: Recargo por kilogramo excedente (ej. 1.50 Bs/kg si $P_f > 2\text{ kg}$).

- F_{zona}: Factor multiplicador según dificultad de acceso del cuadrante urbano (1.0 a 1.3).
- F_{demanda}: Tarifa dinámica por condiciones meteorológicas adversas (lluvia) u horarios pico de congestión vehicular.

. . . . . .

d) Proceso Unificado de Desarrollo de Software (PUDS)

El PUDS (o RUP - Rational Unified Process) es un marco de trabajo de ingeniería de software disciplinado, estandarizado y ampliamente adoptado en la industria para el desarrollo de sistemas orientados a objetos.

Características Esenciales

1. **Dirigido por Casos de Uso (*Use-Case Driven*)**: Los casos de uso son los artefactos primarios que guían todo el proceso de desarrollo. Definen el comportamiento esperado del sistema desde la óptica de los actores externos y sirven como hilo conductor transversal para el análisis, el diseño arquitectónico, la codificación en los controladores y la generación de casos de prueba.
2. **Centrado en la Arquitectura (*Architecture-Centric*)**: Establece tempranamente una arquitectura base sólida y extensible. Considera las diferentes vistas arquitectónicas (Vista Lógica, Vista de Implementación, Vista de Procesos, Vista de Despliegue y Vista de Casos de Uso — el modelo 4+1 vistas de Kruchten).
3. **Iterativo e Incremental (*Iterative and Incremental*)**: El proyecto se divide en ciclos o iteraciones temporales manejables. Cada iteración aborda un conjunto priorizado de casos de uso y culmina con una versión ejecutable y verificable del software desplegada en un entorno real.

. . . . . .

Fases del Ciclo de Vida PUDS

- **Fase de Inicio (*Inception*)**: Delimitación del alcance del proyecto, identificación de los actores principales, modelado de negocio preliminar, formulación de objetivos y viabilidad económica/técnica.
- **Fase de Elaboración (*Elaboration*)**: Mitigación de los principales riesgos técnicos, refinamiento de los casos de uso críticos, diseño de la arquitectura base del sistema y especificación del modelo de datos.
- **Fase de Construcción (*Construction*)**: Desarrollo masivo de los componentes de software, integración de módulos, codificación y pruebas unitarias de los requerimientos funcionales restantes.
- **Fase de Transición (*Transition*)**: Despliegue en el entorno productivo de nube, pruebas de aceptación con usuarios finales, capacitación y ajustes post-lanzamiento.

. . . . . .

Flujos de Trabajo Fundamentales (Disiplinas)

1. **Captura de Requisitos**: Identificación, estructuración y especificación detallada de Casos de Uso y Actores, diagramas con relaciones `<<include>>`, `<<extend>>` y prototipado de interfaces.

2. Análisis: Transformación de los requisitos en una estructura conceptual del sistema (clases de análisis: Interfaz, Control y Entidad).

3. Diseño: Definición de la arquitectura física (Despliegue) y lógica (Capas), diseño de interacción (Secuencia y Comunicación), diseño de comportamiento (Estados y Tiempo), diseño de navegación y diseño de base de datos relacional.

4. Implementación: Selección de plataforma y traducción del diseño en código ejecutable organizado en paquetes y microservicios.

5. Pruebas: Verificación funcional mediante pruebas de caja negra, pruebas de casos de uso e historias de usuario para garantizar la calidad del software.

. . . . . .

e) Lenguaje de Modelado Unificado (UML 2.5+)

UML es el lenguaje estándar adoptado por la OMG (Object Management Group) para visualizar, especificar, construir y documentar los artefactos de un sistema de software con enfoque orientado a objetos.

Taxonomía de Diagramas en UML 2.5+

UML 2.5+ clasifica sus diagramas en dos grandes vertientes complementarias:

- **Diagrama de Actividad**: Es el artefacto principal para modelar el **Modelo de Negocio** y la lógica procedimental de los casos de uso. Representa flujos de control paso a paso, bifurcaciones de decisión, uniones y actividades concurrentes organizadas mediante **particiones o calles (*swimlanes*)** que identifican con precisión al actor responsable de cada acción.
- **Diagrama de Casos de Uso**: Representa los límites del sistema y la interacción funcional entre actores externos y servicios del software.
- **Diagrama de Secuencia y Comunicación**: Diagramas de interacción que enfatizan el orden cronológico del paso de mensajes entre objetos (Secuencia) o la organización estructural de los objetos colaboradores (Comunicación).
- **Diagrama de Despliegue**: Modela la topología física del hardware, nodos de ejecución en la nube, servidores web, servidores de bases de datos y los enlaces de red telemáticos que los interconectan.

. . . . . .

Modelo de Negocio (Business Modeling - Diagramas de Actividades de Macroprocesos)

2.1 Justificación del Modelo de Negocio en el PUDS

En la ingeniería de software profesional guiada por el PUDS, **antes de formular los requisitos de un sistema informático es imperativo comprender a cabalidad cómo opera el negocio en el mundo real**. Como enfatiza la cátedra en sus notas de clase (C6, B4), un desarrollador no puede programar un software a la medida de una organización si desconoce los flujos de trabajo, las responsabilidades de los empleados, los procedimientos de compra a proveedores, la mecánica contable del inventario y la interacción del cliente en las tiendas.

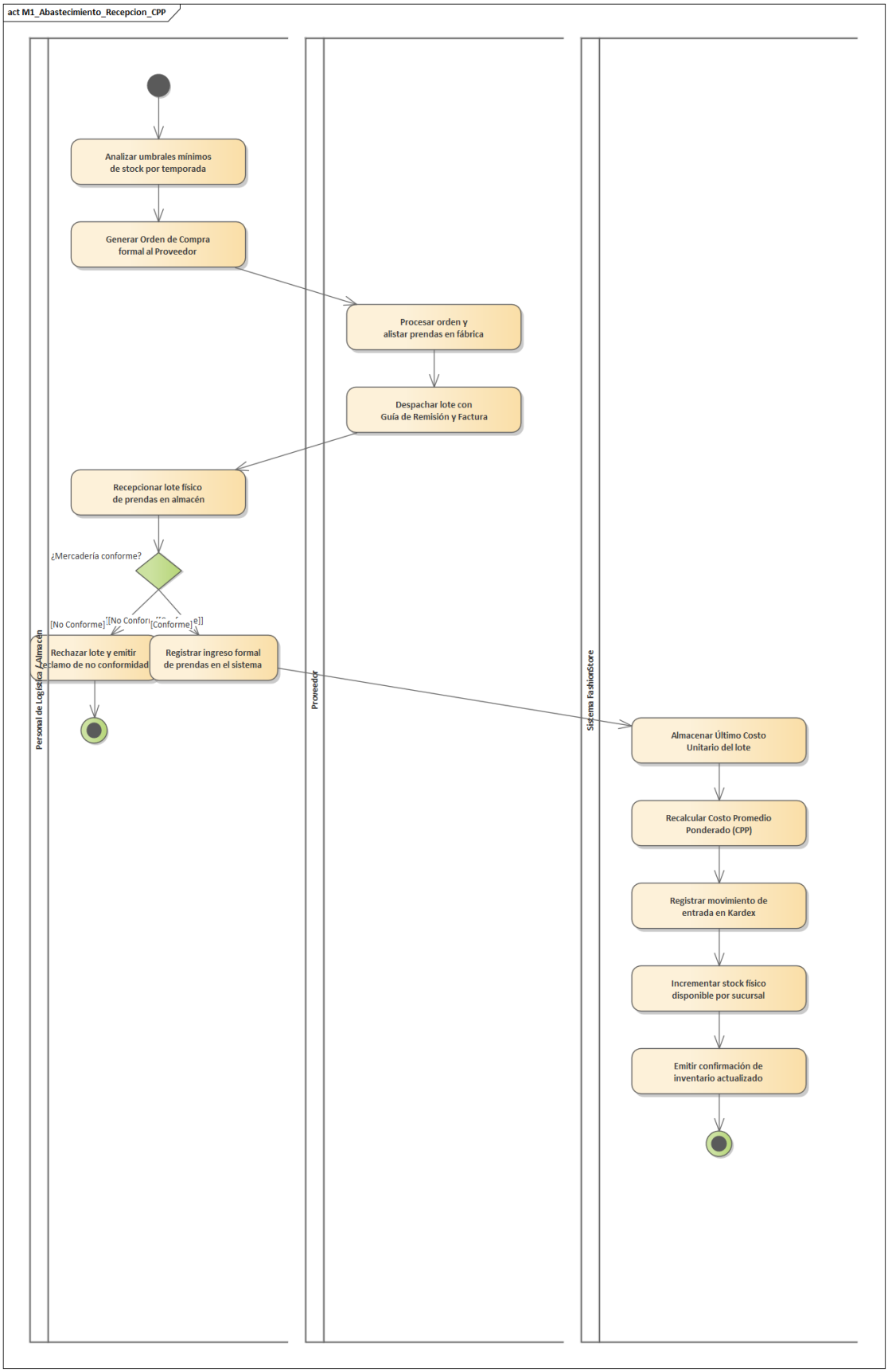
El **Modelo de Negocio** utiliza los **Diagramas de Actividad UML con calles/particiones (swimlanes)** para modelar los procesos operacionales de la empresa. Cada calle representa un actor o trabajador del negocio (Cliente, Encargado de Sucursal, Cajero, Almacén/Logística, Sistema Automatizado/Pasarela), permitiendo visualizar con absoluta claridad la secuencia de actividades, decisiones, bifurcaciones y el flujo de información que genera valor económico en la cadena de moda masculina FashionStore.

A continuación se presentan los diagramas de actividad UML correspondientes a los **seis macroprocesos de negocio medulares** de FashionStore:

. . . . . .

2.2 Macroproceso 1: Gestión de Abastecimiento y Recepción con Costo Promedio Ponderado

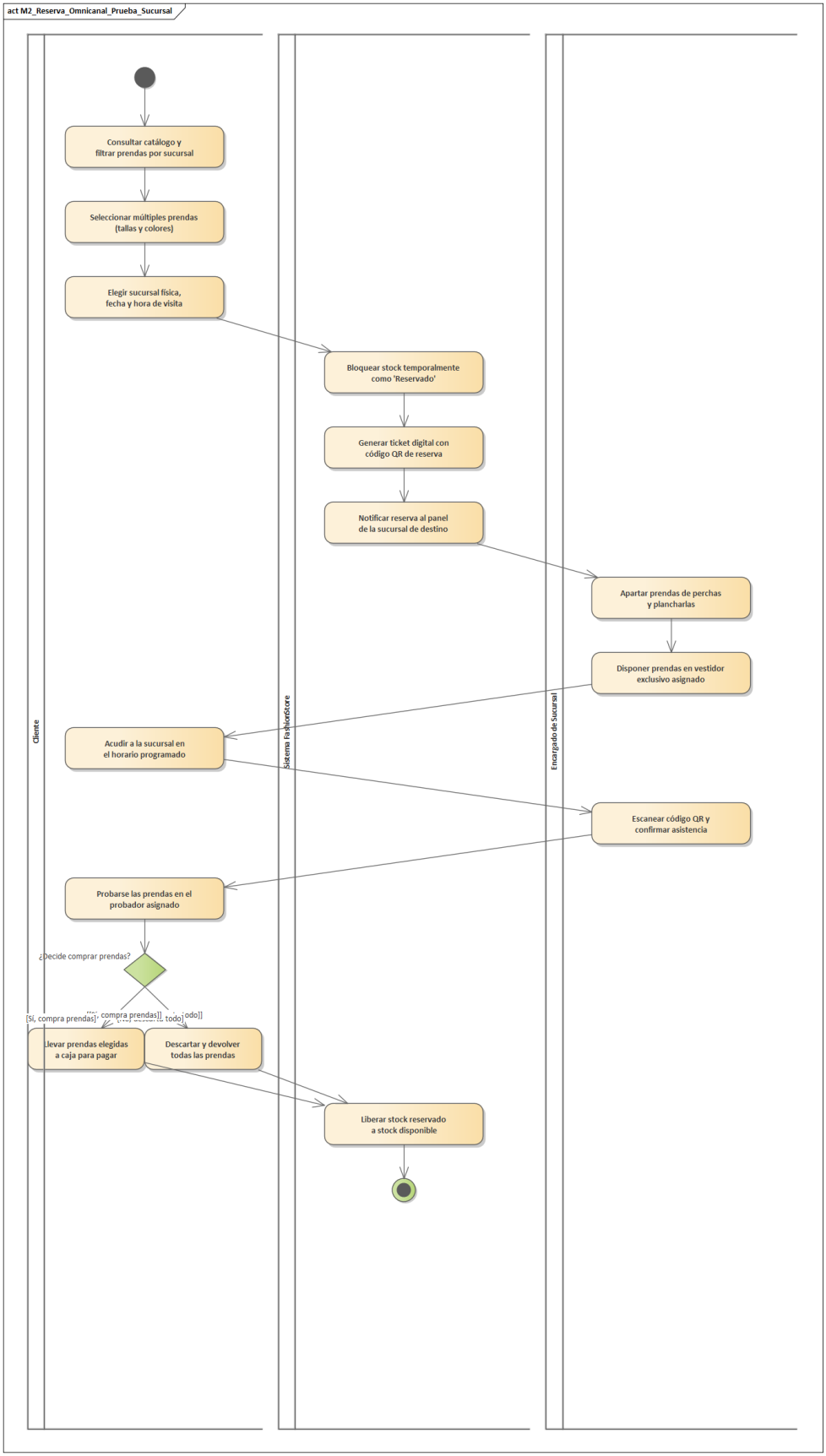
Este proceso describe el ciclo mediante el cual la empresa adquiere lotes de prendas a proveedores por temporada comercial, recepciona la mercadería físicamente en sucursal o almacén central, y actualiza el inventario calculando matemáticamente el Costo Promedio Ponderado (CPP) y registrando el Último Costo Unitario.


act M1_Abastecimiento_Recepcion_CPP
Personal de Logística / Almacén
Proveedor
Sistema Fashionstore
Analizar umbrales mínimos de stock por temporada
Generar Orden de Compra formal al Proveedor
Procesar orden y alistar prendas en fábrica
Despachar lote con Guía de Remisión y Factura
Recepcionar lote físico de prendas en almacén
¿Mercadería conforme?
[No Conforme]
[Conforme]
Rechazar lote y emitir reclamo de no conformidad
Registrar ingreso formal de prendas en el sistema
Almacenar Último Costo Unitario del lote
Recalcular Costo Promedio Ponderado (CPP)
Registrar movimiento de entrada en Kardex
Incrementar stock físico disponible por sucursal
Emitir confirmación de inventario actualizado

. . . . . .

2.3 Macroproceso 2: Reserva Omnicanal y Prueba Presencial en Sucursal

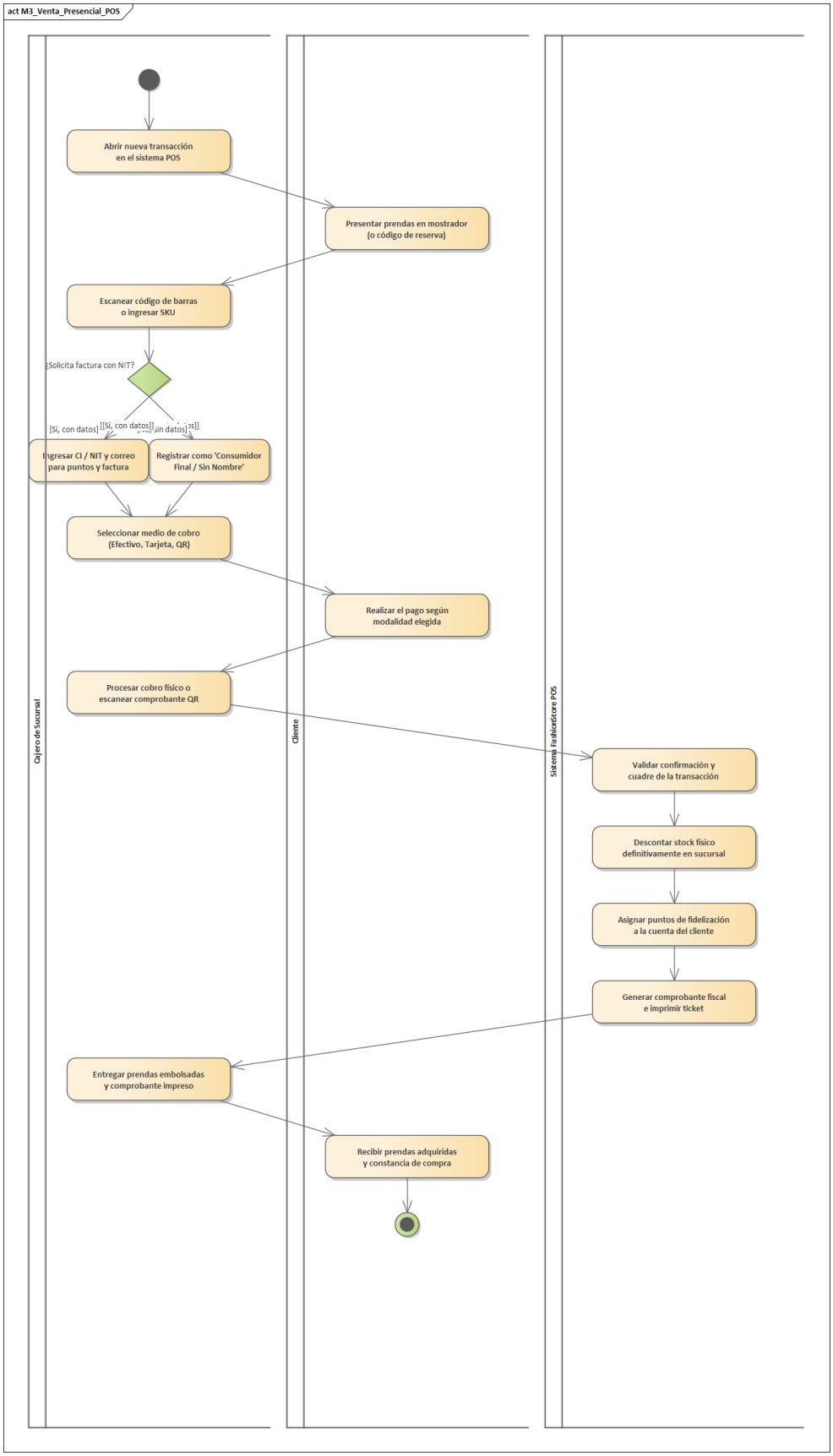
Modela el flujo mediante el cual un cliente explora el catálogo desde su móvil o web, reserva varias prendas en una sucursal física, el encargado de tienda prepara los probadores con anticipación, y el cliente acude a probarse las prendas para decidir su compra.


act M2_Reserva_Omnicanal_Prueba_Sucursal
Cliente
Sistema FashionStore
Encargado de Sucursal
Consultar catálogo y filtrar prendas por sucursal
Seleccionar múltiples prendas (tallas y colores)
Elegir sucursal física, fecha y hora de visita
Bloquear stock temporalmente como 'Reservado'
Generar ticket digital con código QR de reserva
Notificar reserva al panel de la sucursal de destino
Apartar prendas de perchas y plancharlas
Disponer prendas en vestidor exclusivo asignado
Acudir a la sucursal en el horario programado
Escanear código QR y confirmar asistencia
Probarse las prendas en el probador asignado
¿Decide comprar prendas?
[Sí, compra prendas]
todo]]
Llevar prendas elegidas a caja para pagar
Descartar y devolver todas las prendas
Liberar stock reservado a stock disponible

. . . . . .

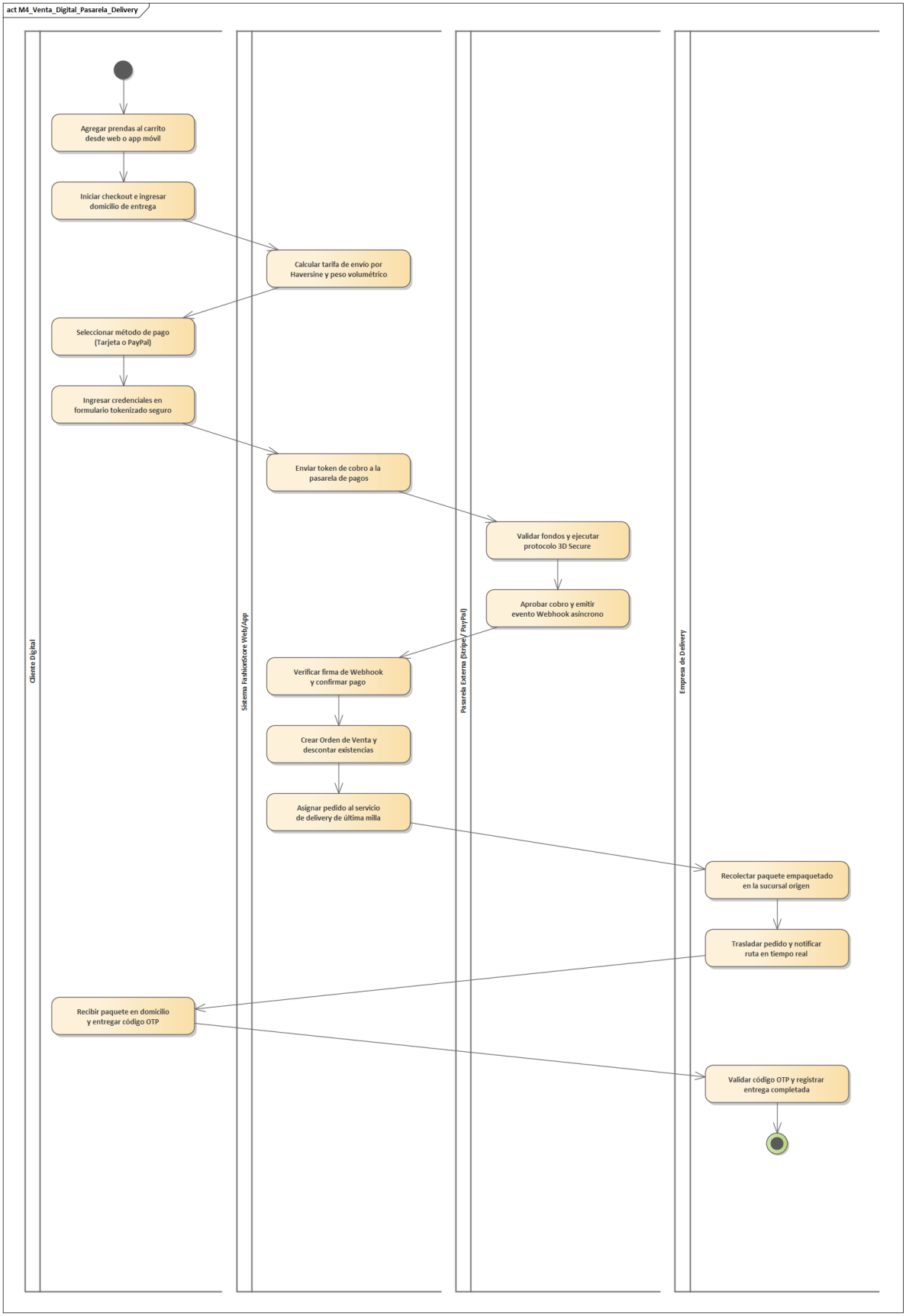
2.4 Macroproceso 3: Venta Presencial en Punto de Venta (POS)

Modela el proceso transaccional en la caja registradora de la sucursal física, ya sea proveniente de una reserva presencial previa o de una compra directa de mostrador.


act M3_Venta_Presencial_POS
Cajero de Sucursal
Cliente
Sistema FashionStore POS
Abrir nueva transacción en el sistema POS
Presentar prendas en mostrador (o código de reserva)
Escanear código de barras o ingresar SKU
¿Solicita factura con NIT?
[Sí, con datos]
[Sí, con datos]
[sin datos]
Ingresar CI / NIT y correo para puntos y factura
Registrar como 'Consumidor Final / Sin Nombre'
Seleccionar medio de cobro (Efectivo, Tarjeta, QR)
Realizar el pago según modalidad elegida
Procesar cobro físico o escanear comprobante QR
Validar confirmación y cuadre de la transacción
Descontar stock físico definitivamente en sucursal
Asignar puntos de fidelización a la cuenta del cliente
Generar comprobante fiscal e imprimir ticket
Entregar prendas embolsadas y comprobante impreso
Recibir prendas adquiridas y constancia de compra

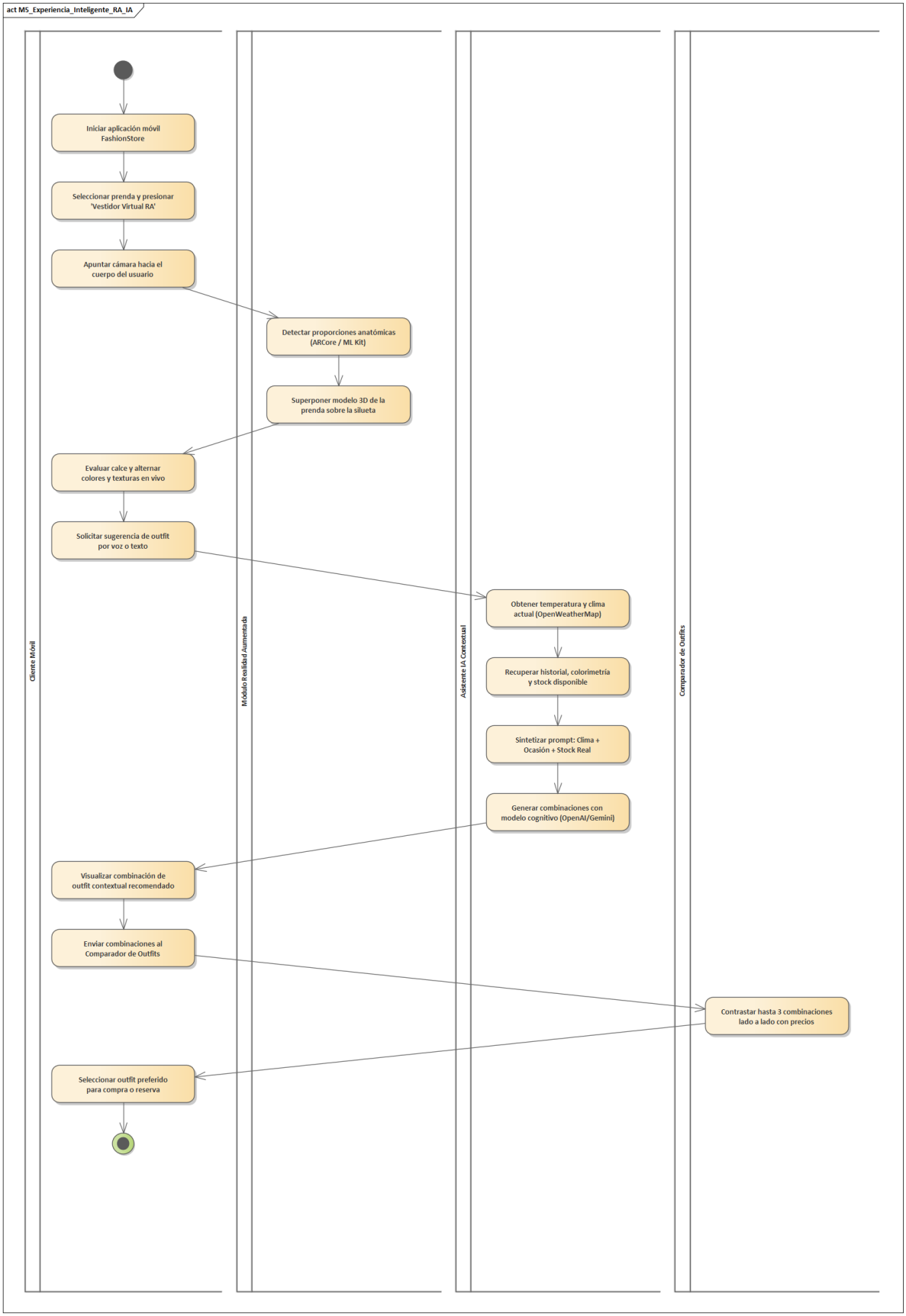
2.5 Macroproceso 4: Venta Digital y Despacho con Pasarela de Pago

Modela el flujo completo del comercio electrónico web y móvil, desde el carrito de compras, el procesamiento seguro con la pasarela de pagos, hasta la logística de despacho y entrega a domicilio.


act M4_Venta_Digital_Pasarela_Delivery
Cliente Digital
Sistema FashionStore Web/App
Pasarela Externa (Stripe/ PayPal)
Empresa de Delivery
Agregar prendas al carrito desde web o app móvil
Iniciar checkout e ingresar domicilio de entrega
Calcular tarifa de envío por Haversine y peso volumétrico
Seleccionar método de pago (Tarjeta o PayPal)
Ingresar credenciales en formulario tokenizado seguro
Enviar token de cobro a la pasarela de pagos
Validar fondos y ejecutar protocolo 3D Secure
Aprobar cobro y emitir evento Webhook asíncrono
Verificar firma de Webhook y confirmar pago
Crear Orden de Venta y descontar existencias
Asignar pedido al servicio de delivery de última milla
Recolectar paquete empaquetado en la sucursal origen
Trasladar pedido y notificar ruta en tiempo real
Recibir paquete en domicilio y entregar código OTP
Validar código OTP y registrar entrega completada

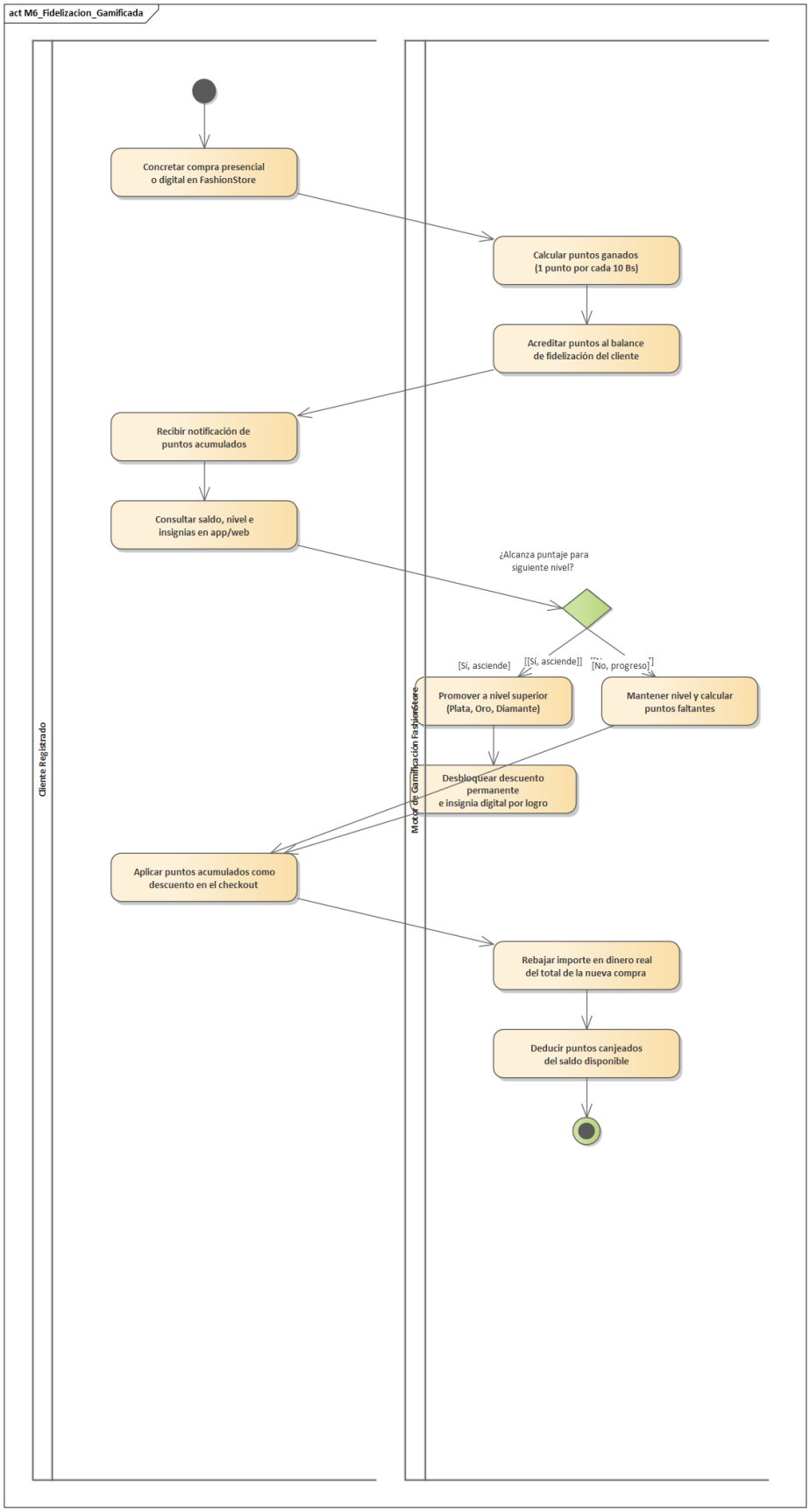
2.6 Macroproceso 5: Experiencia Inteligente (Vestidor Virtual RA + Asistente IA)

Modela la interacción diferenciadora del usuario al emplear la Realidad Aumentada para probarse prendas virtualmente y consultar sugerencias de atuendos contextualizados al clima local.


act M5_Experiencia_Inteligente_RA_IA
Cliente Móvil
Módulo Realidad Aumentada
Asistente IA Contextual
Comparador de Outfits
Iniciar aplicación móvil FashionStore
Seleccionar prenda y presionar 'Vestidor Virtual RA'
Apuntar cámara hacia el cuerpo del usuario
Detectar proporciones anatómicas (ARCore / ML Kit)
Superponer modelo 3D de la prenda sobre la silueta
Evaluar calce y alternar colores y texturas en vivo
Solicitar sugerencia de outfit por voz o texto
Obtener temperatura y clima actual (OpenWeatherMap)
Recuperar historial, colorimetría y stock disponible
Sintetizar prompt: Clima + Ocasión + Stock Real
Generar combinaciones con modelo cognitivo (OpenAI/Gemini)
Visualizar combinación de outfit contextual recomendado
Enviar combinaciones al Comparador de Outfits
Contrastar hasta 3 combinaciones lado a lado con precios
Seleccionar outfit preferido para compra o reserva

2.7 Macroproceso 6: Programa de Fidelización Gamificado

Modela el ciclo de fidelización y retención de clientes mediante gamificación: acumulación de puntos en cada compra, progresión a través de niveles jerárquicos (Bronce, Plata, Oro, Diamante), desbloqueo de insignias y canje de beneficios en dinero real.


act M6_Fidelizacion_Gamificada
Cliente Registrado
Motor de Gamificación FashionStore
Concretar compra presencial o digital en FashionStore
Calcular puntos ganados (1 punto por cada 10 Bs)
Acreditar puntos al balance de fidelización del cliente
Recibir notificación de puntos acumulados
Consultar saldo, nivel e insignias en app/web
¿Alcanza puntaje para siguiente nivel?
[Sí, asciende]
[[Sí, asciende]]
[No, progreso]
Promover a nivel superior (Plata, Oro, Diamante)
Mantener nivel y calcular puntos faltantes
Desbloquear descuento permanente e insignia digital por logro
Aplicar puntos acumulados como descuento en el checkout
Rebajar importe en dinero real del total de la nueva compra
Deducir puntos canjeados del saldo disponible

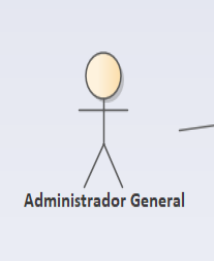
Capitulo1. Flujo de Trabajo: Captura de Requisitos

1.1 Identificación de Casos de Uso y Actores

1.1.1 Actores Internos (Personal de la Organización)

Corresponden a los usuarios humanos pertenecientes a la nómina operativa y ejecutiva de la cadena de tiendas de indumentaria masculina. Acceden al sistema a través de la plataforma web administrativa (Angular) o la terminal de caja POS, sujetos a rigurosas políticas de Control de Acceso Basado en Roles (RBAC):

1. **Administrador General:**
   - **Descripción y Rol:** Máxima autoridad directiva e informática de la empresa. Cuenta con privilegios irrestrictos sobre la configuración y gobernanza de la plataforma.
   - **Responsabilidades Clave:**
     - Creación, modificación, asignación de roles y revocación de cuentas de personal interno.

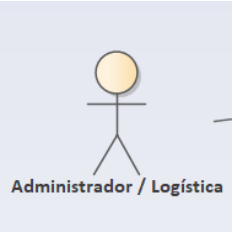


     - Desbloqueo administrativo de cuentas inhabilitadas preventivamente tras 5 intentos fallidos de autenticación.
     - Alta y parametrización de ciudades, sucursales físicas, georreferenciación GPS y capacidades de probadores.
     - Gestión del catálogo maestro: categorías, marcas, temporadas estacionales y atributos de moda (tallas y colores Hexadecimales).
     - Supervisión de métricas financieras, auditorías de seguridad e indicadores gerenciales en los tableros de control (dashboards).
2. **Encargado de Sucursal:**
   - **Descripción y Rol:** Responsable del funcionamiento diario, la sala de ventas y el almacén de una tienda física específica (ej. *Sucursal Equipetrol*, *Sucursal Calacoto* o *Sucursal El Prado*).
   - **Responsabilidades Clave:**
     - Monitoreo en tiempo real del panel de reservas presenciales asignadas a su sucursal.

     - Coordinación y supervisión del apartado físico anticipado de prendas en perchas y probadores previo a la visita del cliente.
     - Validación de la llegada del cliente a la tienda mediante el escaneo óptico del código QR de la reserva.
     - Control y auditoría del stock físico local, reporte de mermas o prendas deterioradas y recepción de transferencias inter-sucursales.
3. **Cajero de Sucursal:**
   - **Descripción y Rol:** Operador de mostrador responsable de la atención directa y formalización de cobros en los puntos de venta físicos (POS).

- **Responsabilidades Clave:**
  - Autenticación en la terminal de caja e inicio de turno de cobro.
  - Búsqueda ágil de prendas mediante escaneo de códigos de barras SKU o códigos alfanuméricos.
  - Conversión directa de solicitudes de reserva presencial en ventas definitivas de mostrador.
  - Procesamiento de cobros físicos (efectivo con cálculo automático de cambio) y cobros electrónicos (tarjetas con terminal POS física o Código QR Interoperable).
  - Emisión e impresión de comprobantes de venta válidos (tickets térmicos y facturas) con descuento automático de inventario.

4. **Personal de Logística y Almacén:**
   - **Descripción y Rol:** Operadores de bodega y centros de distribución responsables de la recepción de mercadería y el despacho de pedidos.
   - **Responsabilidades Clave:**
     - Recepción física y control de calidad de lotes textiles enviados por proveedores.
     - Registro minucioso del costo unitario neto de adquisición del lote para alimentar el recálculo automático del Costo Promedio Ponderado (CPP).
     - Embalaje, etiquetado y preparación de órdenes de venta digital destinadas a entrega a domicilio (*Delivery*).
     - Asignación de paquetes a transportistas y registro de hojas de ruta para última milla.



1.1.2 Actores Externos (Usuarios y Sistemas Cooperantes)

Representan a las personas ajenas a la nómina de la empresa y a los servicios informáticos de terceros que interactúan mediante interfaces de usuario o protocolos de comunicación en red (APIs REST, Webhooks y pasarelas):

1. **Cliente Final (Consumidor Masculino):**
   - **Descripción y Rol:** Usuario final y comprador de la plataforma, que interactúa a través de la aplicación móvil (Flutter) o el portal web público (Angular).
   - **Responsabilidades Clave:**
     - Auto-registro de cuenta de usuario, inicio de sesión seguro y gestión de perfil personal.
     - Consulta y filtrado multidimensional del catálogo de prendas masculinas por categoría, talla, color, precio y sucursal.
     - Proyección virtual de prendas sobre su silueta mediante el Vestidor Virtual con Realidad Aumentada (RA).

        - Solicitud de reservas anticipadas de prendas para acudir presencialmente a probarlas en la sucursal de su elección.
        - Formalización de compras digitales mediante carrito de compras omnicanal y pago electrónico seguro.
        - Acumulación y canje de puntos de fidelización dentro del programa gamificado y recepción de recomendaciones por IA.
2. **Proveedor Textil:**
    - **Descripción y Rol:** Empresa fabricante o comercializadora que suministra indumentaria masculina a la cadena FashionStore.
    - **Responsabilidades Clave:**
        - Recepción y confirmación de órdenes de compra emitidas por la administración.
        - Suministro de lotes de prendas asociados a colecciones y temporadas comerciales específicas.
        - Envío de notas de entrega y fichas técnicas de composición textil y cuidado de prendas.
3. **Pasarela de Pagos Electrónicos (Stripe / PayPal / Libélula):**
    - **Descripción y Rol:** Sistema bancario y tecnológico externo que procesa transacciones monetarias digitales bajo estrictos estándares de seguridad financiera (PCI-DSS).
    - **Responsabilidades Clave:**
        - Captura segura y tokenización de números de tarjetas de débito/crédito, impidiendo que datos sensibles toquen los servidores de FashionStore.
        - Validación de fondos, autenticación del titular mediante protocolo 3D Secure 2.0 y prevención de fraudes.
        - Confirmación o rechazo síncrono de transacciones y emisión de notificaciones asíncronas vía Webhooks hacia el backend FastAPI.
4. **Servicio de Inteligencia Artificial (OpenAI API / Gemini API):**
    - **Descripción y Rol:** Motor cognitivo en la nube que actúa como asistente de estilo personalizado y procesador de lenguaje natural.
    - **Responsabilidades Clave:**
        - Análisis de preferencias de estilo, paletas de colorimetría y compras previas del usuario.
        - Síntesis de condiciones de contexto (temperatura y clima local) para sugerir combinaciones coherentes de ropa masculina.
        - Procesamiento semántico de consultas del catálogo y comandos verbales de búsqueda por voz.
5. **Servicio Meteorológico Externo (OpenWeatherMap API):**

- **Descripción y Rol:** Proveedor telemático en la nube de información climática en tiempo real.
- **Responsabilidades Clave:**
  - Provisión continua de variables meteorológicas (temperatura en grados Celsius, estado del cielo, nivel de precipitaciones y humedad) de la ciudad donde se ubica el cliente.
  - Entrega de datos para que el motor de IA adapte dinámicamente las recomendaciones de vestimenta (ej. sugiriendo gabardinas o suéteres ante frentes fríos).

6. **Empresa de Delivery / Repartidor de Última Milla:**
   - **Descripción y Rol:** Agente logístico urbano (como Yaigo, Yummy, PedidosYa o flota interna de repartidores) encargado del traslado físico del producto.
   - **Responsabilidades Clave:**
     - Recolección física del paquete empaquetado en la sucursal de despacho asignada.
     - Transporte seguro hacia el domicilio georreferenciado del cliente.
     - Confirmación de la entrega efectiva mediante captura de firma digital o validación del código OTP proporcionado por el comprador.

A partir del análisis exhaustivo del modelo de negocio y las necesidades operativas de FashionStore, se identifican formalmente los Casos de Uso que componen el sistema, vinculados con los actores que interactúan con cada servicio:

1.1.3 Lista de casos de uso

Código CU	Nombre del Caso de Uso	Módulo Asociado	Actor(es) Principal(es)	Descripción Resumida
CU01	Autenticar Usuario y Control de Acceso (RBAC)	M01	Administrador, Encargado, Cajero, Logística, Cliente	Inicio de sesión seguro con credenciales hash bcrypt, emisión de tokens JWT, control granular de roles y auditoría de accesos.
CU02	Registrar Cliente (Auto-registro de Clientes)	M01	Cliente no autenticado	Creación de cuenta de cliente por autoservicio en web/móvil con validación de contraseña segura, hash bcrypt y bienvenida.
CU03	Recuperar Contraseña (Token OTP de 6 Dígitos)	M01	Usuario (Cualquier Rol)	Restablecimiento autoservicio de contraseña olvidada mediante envío y validación de código OTP temporal de 6 dígitos vía correo electrónico.
CU04	Gestionar Usuarios y Roles (RBAC y Desbloqueo)	M01	Administrador General	Administración de personal interno, asignación de roles jerárquicos, asignación a sucursal y desbloqueo de cuentas bloqueadas por intentos.
CU05	Gestionar Ciudades y Sucursales	M02	Administrador General	Administración geográfica y operativa de ciudades, sucursales físicas, georreferenciación GPS, horarios y capacidades de probadores.
CU06	Gestionar Productos y Atributos de Moda	M03	Administrador General	CRUD de prendas de vestir masculinas, categorías, marcas, tallas normalizadas y colores multivaluados con código Hex.

CU07	Gestionar Temporadas y Colecciones	M04	Administrador General	Calendarización y administración de campañas estacionales (Primavera-Verano, Otoño-Invierno, Escolar, Promociones).
CU08	Gestionar Proveedores Textiles	M05	Administrador General, Logística	Registro y administración de empresas proveedoras, condiciones de pago, líneas de prendas y contratos de aprovisionamiento.
CU09	Gestionar Inventario Multi-Sucursal y Costos Ponderados (CPP)	M06	Personal de Logística, Encargado Sucursal	Control de existencias por sucursal, registro de entradas con último costo unitario y recálculo matemático del Costo Promedio Ponderado (CPP).
CU10	Consultar Catálogo y Disponibilidad por Sucursal	M07	Cliente, Encargado de Sucursal, Cajero	Exploración del catálogo omnicanal con filtros multidimensionales y consulta de existencias en tiempo real por tienda física.
CU11	Solicitar Reserva de Prendas en Sucursal	M10	Cliente	Preselección de prendas, selección de sucursal física, programación de fecha/hora de visita y generación de ticket QR. (Ciclo 2)
CU12	Preparar y Atender Reserva Presencial	M10	Encargado de Sucursal	Apartado físico de prendas reservadas en probadores asignados y confirmación de llegada del cliente mediante escaneo de QR. (Ciclo 2)
CU13	Administrar Carrito de Compras Omnicanal	M11	Cliente	Incorporación, edición y cálculo automático de totales con validación atómica de existencias. (Ciclo 2)
CU14	Procesar Compra Digital y Checkout	M12	Cliente	Formalización de compra en línea con selección de modalidad de entrega (retiro en tienda o delivery) y facturación. (Ciclo 2)
CU15	Registrar Venta Presencial en Caja (POS)	M13	Cajero de Sucursal	Registro de venta en mostrador físico, lectura de códigos de barras, conversión de reservas en ventas y emisión de tickets. (Ciclo 2)
CU16	Procesar Pago con Pasarela Electrónica	M14	Pasarela Externa (Stripe/PayPal), Cliente	Ejecución segura de cobro digital con verificación 3D Secure y confirmación asíncrona mediante Webhooks. (Ciclo 2)
CU17	Gestionar Tipos y Medios de Cobro	M15	Administrador, Cajero	Parametrización de pagos en efectivo, tarjeta de débito/crédito y Códigos QR interoperables del BCB. (Ciclo 2)
CU18	Gestionar Despacho y Logística de Delivery	M19	Personal de Logística, Empresa Delivery	Cálculo de tarifas por Haversine y peso volumétrico, asignación de repartidores y rastreo de envíos en tiempo real. (Ciclo 2)
CU19	Visualizar Prenda en Vestidor Virtual con RA	M08	Cliente Móvil	Proyección tridimensional de prendas sobre la silueta del cliente mediante la cámara del smartphone usando ARCore. (Ciclo 3)
CU20	Comparar Outfits Lado a Lado	M09	Cliente	Contrastación visual interactiva de hasta 3 combinaciones de ropa con desglose de precios individuales y totales. (Ciclo 3)
CU21	Gestionar Fidelización Gamificada	M16	Cliente, Motor de Gamificación	Acumulación de puntos por compras, progresión en niveles jerárquicos (Bronce a Diamante), desbloqueo de insignias y canje. (Ciclo 3)
CU22	Solicitar Recomendación Contextual de IA	M17	Cliente, Servicio de IA	Sugerencia inteligente de outfits completos considerando clima meteorológico local (API OpenWeatherMap), colorimetría e historial. (Ciclo 3)
CU23	Buscar Productos por Comandos de Voz	M17	Cliente Móvil, Servicio de IA	Consulta de catálogo y solicitudes en lenguaje natural mediante transcripción de voz y procesamiento semántico NLP. (Ciclo 3)
CU24	Visualizar Cuadros de Mando y Dashboards	M18	Administrador General	Consulta de indicadores clave de negocio (KPIs), ventas por tienda, rotación de stock y rentabilidad de colecciones. (Ciclo 3)

. . . . . .

1.2 Priorización de Casos de Uso

En consonancia con los principios del PUDS (mitigación temprana de riesgos arquitectónicos y entrega incremental de valor funcional), se establece la matriz de priorización multidimensional que define la asignación de Casos de Uso por cada ciclo de desarrollo:

Código CU	Nombre del Caso de Uso	Prioridad	Dependencias Previas	Asignación Iterativa
CU01	Autenticar Usuario y Control de Acceso (RBAC)	Media	Ninguna	**(Ciclo 1)**
CU02	Registrar Cliente (Auto-registro de Clientes)	Media	Ninguna	**(Ciclo 1)**
CU03	Recuperar Contraseña (Token OTP de 6 Dígitos)	Media	CU01, CU02	**(Ciclo 1)**
CU04	Gestionar Usuarios y Roles (RBAC y Desbloqueo)	Media	CU01	**(Ciclo 1)**
CU05	Gestionar Ciudades y Sucursales	Baja	CU01	**(Ciclo 1)**
CU06	Gestionar Productos y Atributos de Moda	Media	CU01	**(Ciclo 1)**
CU07	Gestionar Temporadas y Colecciones	Baja	CU01	**(Ciclo 1)**
CU08	Gestionar Proveedores Textiles	Baja	CU01	**(Ciclo 1)**
CU09	Gestionar Inventario Multi-Sucursal y Costos Ponderados (CPP)	Alta	CU05, CU06, CU08	**(Ciclo 1)**
CU10	Consultar Catálogo y Disponibilidad por Sucursal	Media	CU05, CU06, CU09	**(Ciclo 1)**
CU11	Solicitar Reserva de Prendas en Sucursal	Media	CU09, CU10	(Ciclo 2)
CU12	Preparar y Atender Reserva Presencial	Media	CU11	(Ciclo 2)
CU13	Administrar Carrito de Compras Omnicanal	Media	CU09, CU10	(Ciclo 2)
CU14	Procesar Compra Digital y Checkout	Alta	CU13, CU16	(Ciclo 2)
CU15	Registrar Venta Presencial en Caja (POS)	Alta	CU09, CU10, CU17	(Ciclo 2)
CU16	Procesar Pago con Pasarela Electrónica	Alta	CU14	(Ciclo 2)
CU17	Gestionar Tipos y Medios de Cobro	Media	CU15	(Ciclo 2)
CU18	Gestionar Despacho y Logística de Delivery	Media	CU14	(Ciclo 2)
CU19	Visualizar Prenda en Vestidor Virtual con RA	Alta	CU06, CU10	(Ciclo 3)
CU20	Comparar Outfits Lado a Lado	Media	CU10	(Ciclo 3)
CU21	Gestionar Fidelización Gamificada	Media	CU14, CU15	(Ciclo 3)
CU22	Solicitar Recomendación Contextual de IA	Alta	CU10	(Ciclo 3)
CU23	Buscar Productos por Comandos de Voz	Media	CU10, CU22	(Ciclo 3)
CU24	Visualizar Cuadros de Mando y Dashboards	Media	CU09, CU14, CU15	(Ciclo 3)

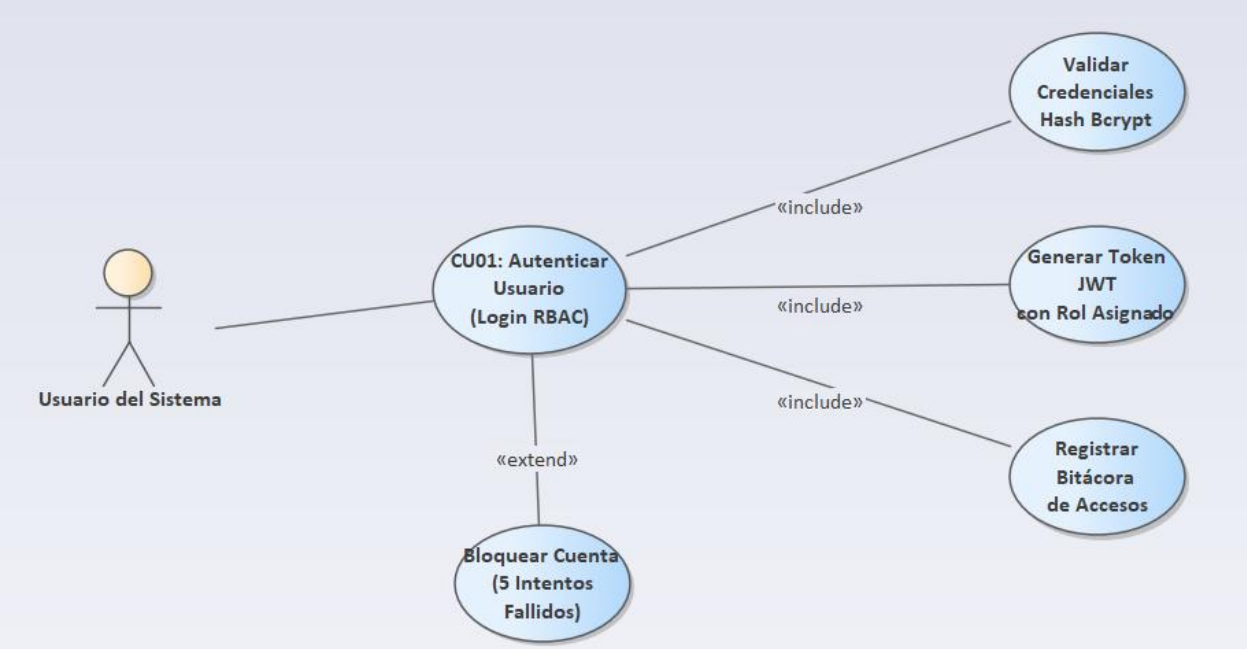
1.3 Detalle de Casos de Uso y Prototipado de Interfaz de Usuario

Conforme a la estricta directriz de la cátedra expresada en clase, la especificación de cada caso de uso se organiza de manera uniforme bajo la siguiente secuencia obligatoria:

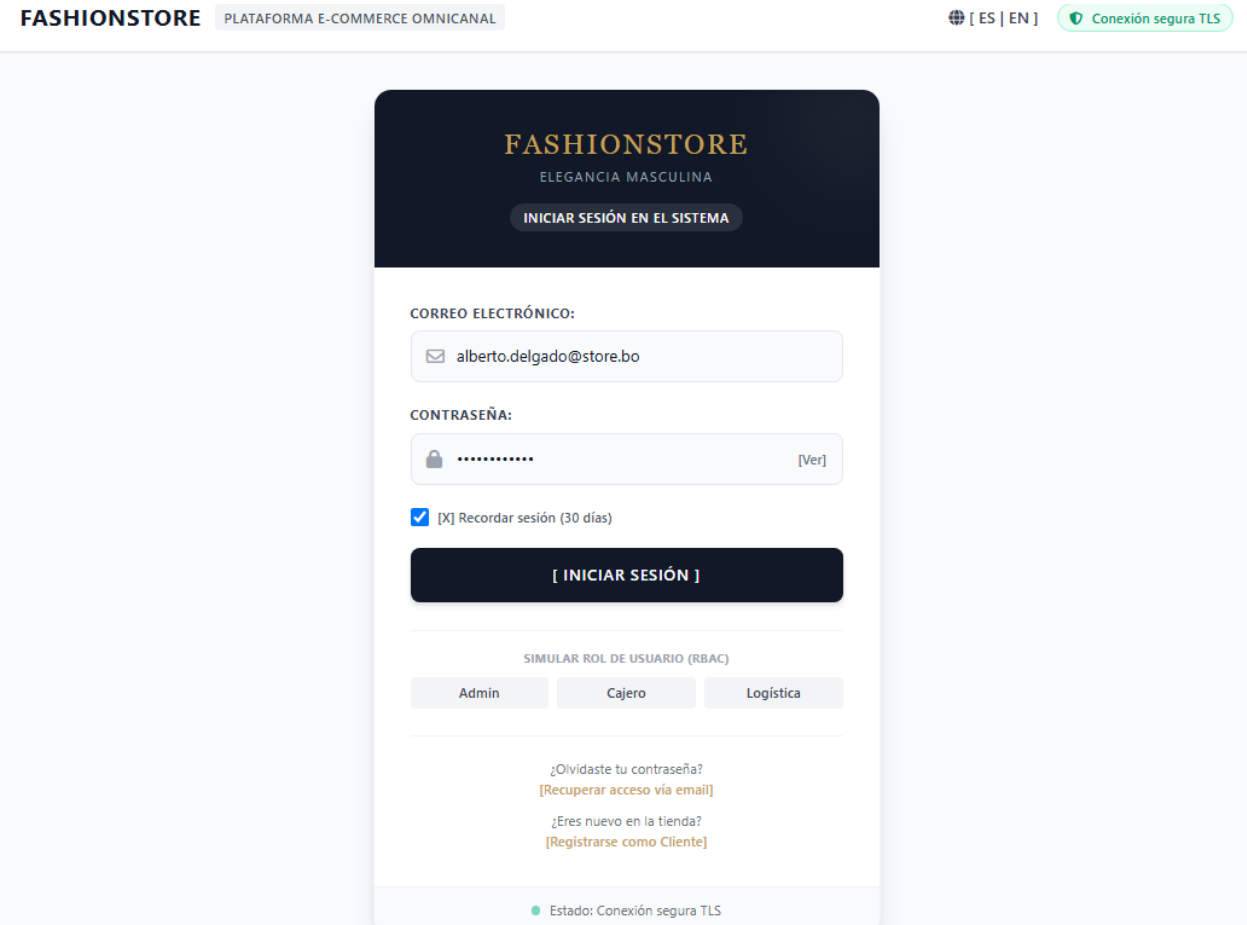
1. **Diseño del Caso de Uso en PlantUML (PlantText)** con sus estereotipos y relaciones (`<<include>>`, `<<extend>>`, herencia).
2. **Prototipo de Interfaz de Usuario (UI Wireframe)** detallado con todos sus componentes de pantalla, campos y acciones.
3. **Tabla Detalle del Caso de Uso** conteniendo la especificación formal del escenario principal, alternos, precondiciones y postcondiciones.

1.3.1 Caso de Uso CU01: Autenticar Usuario y Control de Acceso (RBAC)

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



c) Tabla Detalle del Caso de Uso

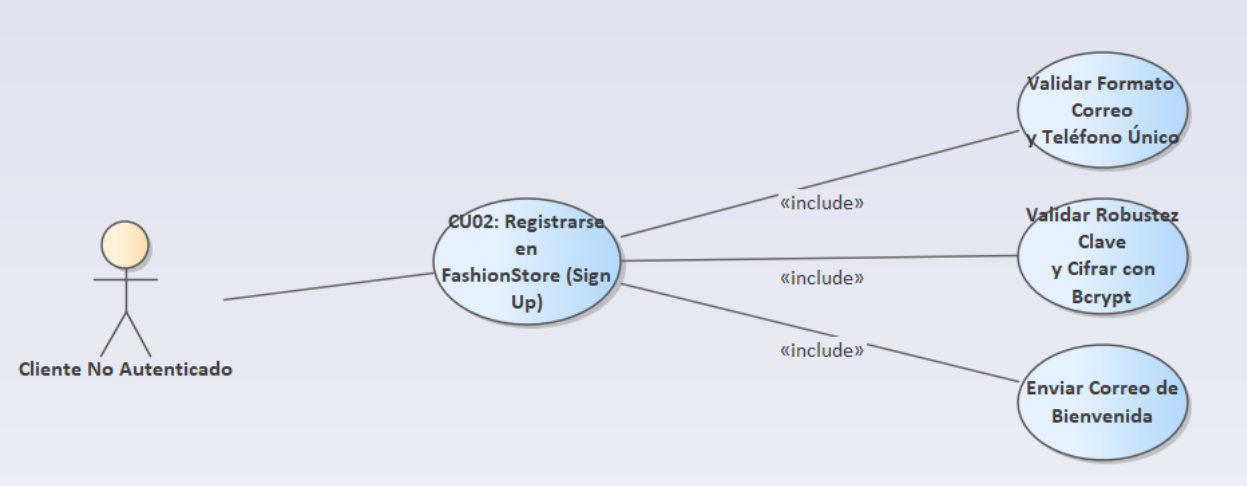
Atributo	Detalle de la Especificación
Identificador	**CU01**
Nombre	**Autenticar Usuario y Control de Acceso (RBAC)**
Actores	Administrador, Encargado de Sucursal, Cajero, Personal de Logística, Cliente.
Propósito	Garantizar el acceso seguro, segregado y controlado a las funciones del sistema mediante credenciales válidas y expedición de tokens JWT firmados.
Tipo	Primario / Esencial.
Precondiciones	1. El usuario debe poseer una cuenta previamente creada y activa (`estado_cuenta = 'ACTIVO'`). 2. El canal de comunicación debe encontrarse protegido bajo protocolo HTTPS/TLS.
Postcondiciones	1. Se emite un token de sesión JWT con los claims de identidad y rol del usuario. 2. Se actualiza el timestamp de `ultimo_acceso` y se restablece a 0 el contador de intentos fallidos. 3. Se registra el evento en la tabla `bitacora_accesos` con IP, fecha/hora y User-Agent.
Flujo Principal	1. El actor ingresa a la vista de login en la plataforma web o móvil. 2. El actor introduce su correo electrónico registrado y contraseña en texto plano. 3. El sistema valida el formato del correo y recupera el registro del usuario. 4. El sistema comprueba mediante el algoritmo `bcrypt.verify` si la contraseña coincide con el `password_hash` almacenado. 5. El sistema verifica que la cuenta no esté bloqueada ni inactiva. 6. El sistema genera el token JWT firmado con el rol correspondiente y tiempo de expiración. 7. El sistema registra el acceso exitoso en la bitácora de auditoría. 8. El sistema redirige al usuario a la vista principal según su perfil (Dashboard para Administrador, POS para Cajero, Reservas para Encargado, Catálogo para Cliente).

Flujos Alternativos	**4a. Contraseña incorrecta:** 1. El sistema incrementa en 1 el campo **`intentos_fallidos`**. 2. Si **`intentos_fallidos >= 5`**, el sistema cambia **`estado_cuenta = 'BLOQUEADO_POR_INTENTOS'`**, establece **`bloqueado_hasta = NOW() + INTERVAL '30 MIN'`** y registra alerta de seguridad. 3. El sistema muestra mensaje: "Credenciales inválidas. Le quedan X intentos antes del bloqueo." **5a. Cuenta bloqueada:** 1. El sistema rechaza la autenticación informando: "Cuenta bloqueada preventivamente. Restablezca su contraseña o contacte a soporte."

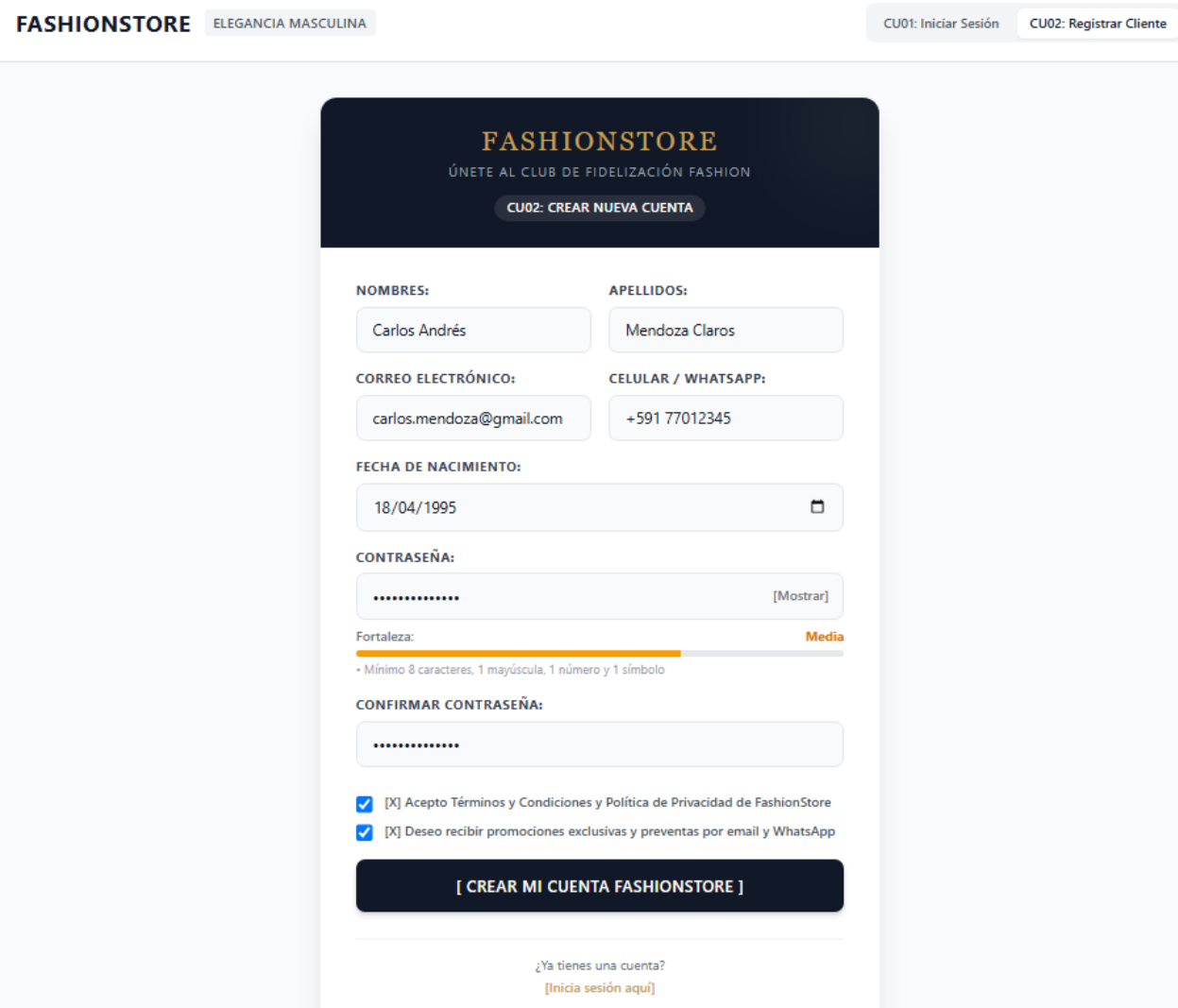
. . . . . .

1.3.2 Caso de Uso CU02: Registrar Cliente (Auto-registro de Clientes)

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



c) Tabla Detalle del Caso de Uso

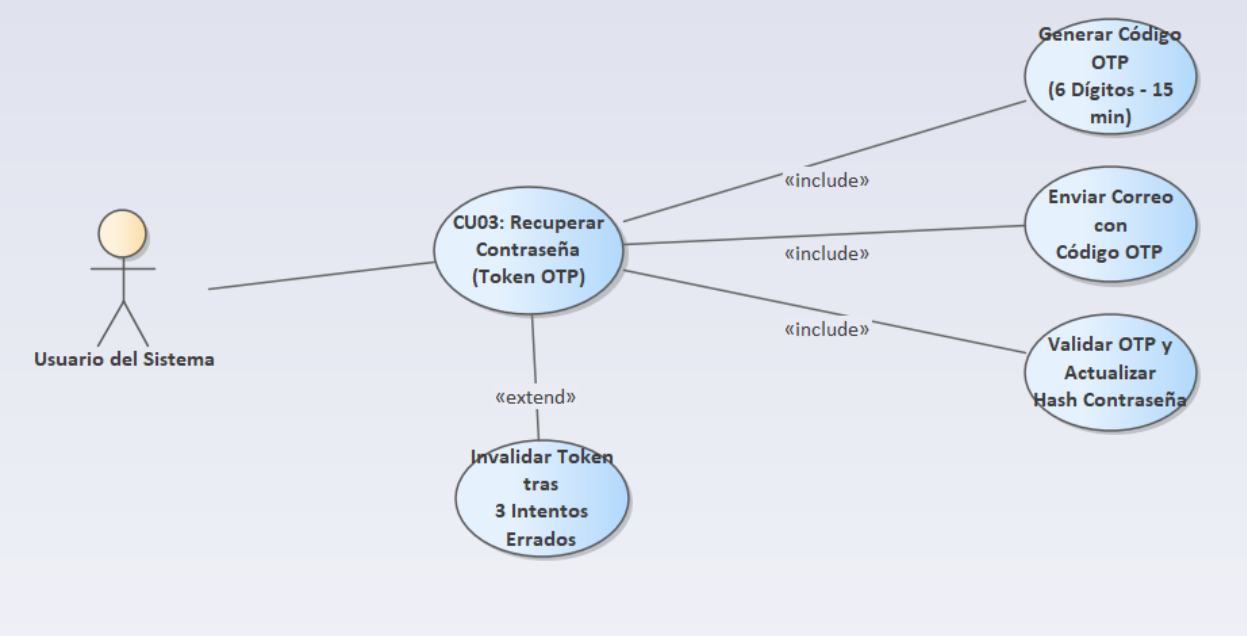
Atributo	Detalle de la Especificación
Identificador	**CU02**
Nombre	**Registrar Cliente (Auto-registro de Clientes)**
Actores	Cliente no autenticado (Visitante Web / Móvil).
Propósito	Permitir que los nuevos clientes se auto-registren en la plataforma FashionStore para acceder al carrito, guardar preferencias, reservar probadores físicos y acumular puntos de fidelización.
Tipo	Primario / Autoservicio.
Precondiciones	El usuario debe encontrarse en la plataforma web o móvil sin sesión activa.
Postcondiciones	1. Se crea un registro en **`usuarios`** con rol **`CLIENTE`** y **`estado_cuenta = 'ACTIVO'`**. 2. Se encripta la contraseña mediante **`bcrypt`** (factor de costo 12). 3. Se expide automáticamente un token JWT y se envía un correo transaccional de bienvenida.
Flujo Principal	1. El visitante selecciona la opción "Registrarse" en la plataforma. 2. El sistema despliega el formulario de captura de datos personales y credenciales. 3. El cliente introduce nombres, apellidos, correo, teléfono y contraseña. 4. El sistema valida en tiempo real la fortaleza de la clave (mínimo 8 caracteres, al menos una mayúscula y un número). 5. El cliente presiona el botón "Crear Mi Cuenta FashionStore".

	6. El sistema verifica que el correo electrónico no exista previamente en la base de datos. 7. El sistema aplica el hash seguro **`bcrypt`** sobre la contraseña ingresada. 8. El sistema persiste el usuario en la base de datos con rol **`CLIENTE`** e inicializa sus puntos de fidelización en 0. 9. El sistema dispara el envío asíncrono del correo de bienvenida. 10. El sistema inicia la sesión automáticamente expidiendo el token JWT y redirige al catálogo.
Flujos Alternativos	**6a. Correo ya registrado:** 1. El sistema detecta duplicidad de correo electrónico. 2. El sistema muestra mensaje de error: "El correo electrónico ya se encuentra registrado. ¿Desea iniciar sesión o recuperar su contraseña?" con enlaces directos. **4a. Contraseña insegura:** 1. El sistema deshabilita el envío e indica visualmente las reglas faltantes en la contraseña. **3a. Contraseñas no coincidentes:** 1. El sistema advierte que los campos de confirmación no concuerdan e impide el envío.

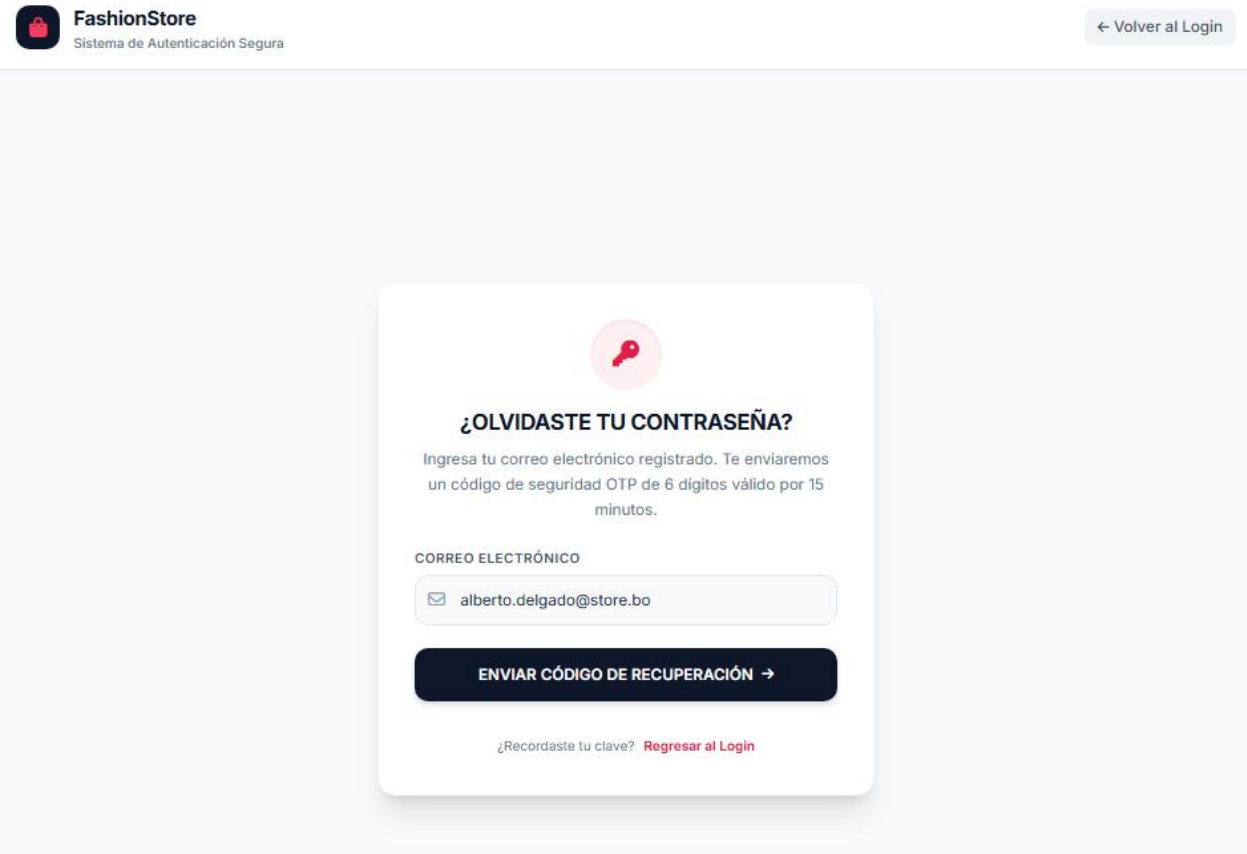
. . . . . .

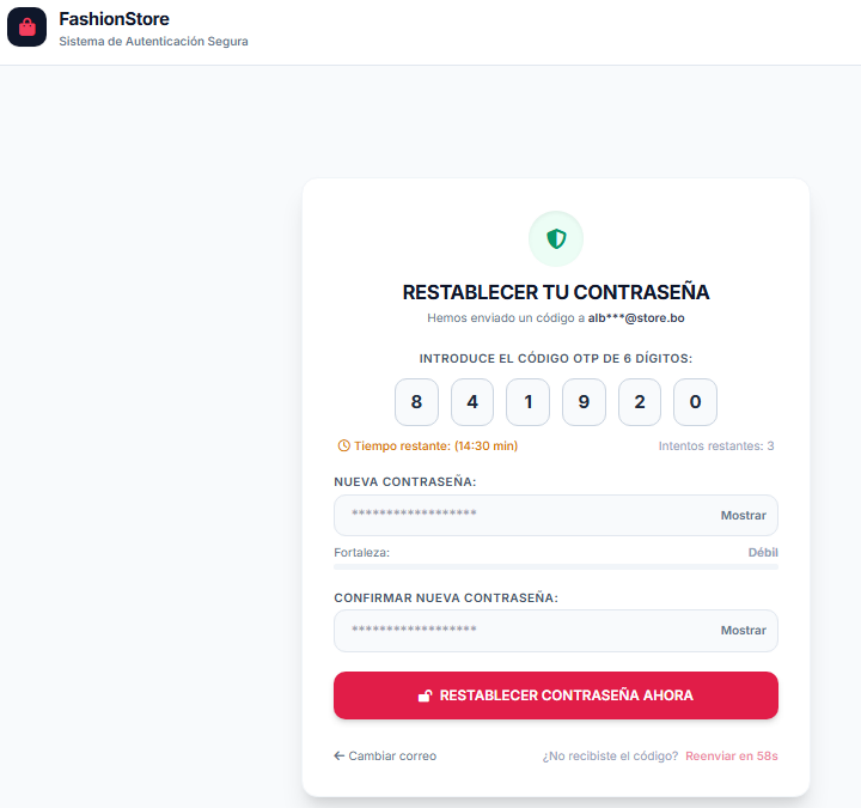
1.3.3 Caso de Uso CU03: Recuperar Contraseña (Token OTP de 6 Dígitos)

a) Diseño del Caso de Uso (PlantUML)



b) Prototipos de Interfaz de Usuario (UI Wireframe - Flujo de 2 Pantallas)





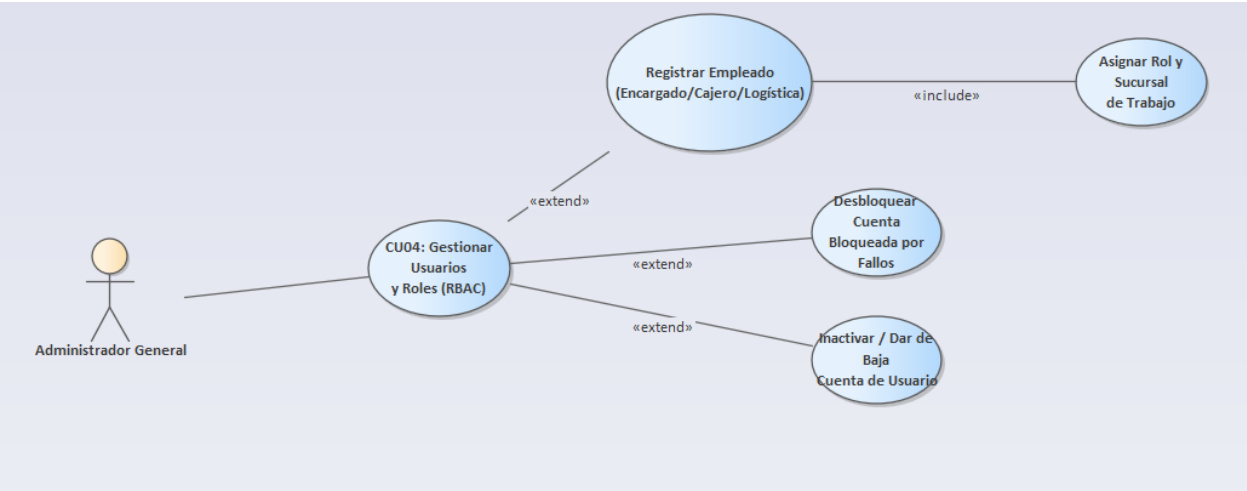
c) Tabla Detalle del Caso de Uso

Atributo	Detalle de la Especificación
Identificador	**CU03**
Nombre	**Recuperar Contraseña (Token OTP de 6 Dígitos)**
Actores	Usuario del Sistema (Cliente, Administrador, Empleado).
Propósito	Permitir a cualquier usuario restablecer de manera autoservicio su contraseña olvidada mediante la generación y validación de un código criptográfico temporal (OTP) enviado a su correo electrónico.
Tipo	Secundario / Soporte y Seguridad.
Precondiciones	1. El usuario debe haber registrado previamente su cuenta con un correo electrónico válido. 2. El servicio SMTP/transaccional debe encontrarse operativo.
Postcondiciones	1. Se genera un código OTP numérico aleatorio de 6 dígitos con expiración de 15 minutos en **`tokens_recuperacion`**. 2. Tras la validación, se actualiza el **`password_hash`** del usuario en **`usuarios`** y se inhabilita el token OTP. 3. Si la cuenta estaba bloqueada por intentos fallidos, se desbloquea automáticamente (**`estado_cuenta = 'ACTIVO'`**, **`intentos_fallidos = 0`**).
Flujo Principal	1. El usuario presiona "¿Olvidaste tu contraseña?" en la pantalla de autenticación. 2. El sistema despliega la pantalla de solicitud de correo. 3. El usuario ingresa su correo electrónico y presiona "Enviar Código de Recuperación". 4. El sistema busca el correo en la base de datos. 5. El sistema genera un código numérico aleatorio criptoseguro de 6 dígitos con fecha de expiración fijada a **`NOW() + 15 MIN`**. 6. El sistema almacena el hash del OTP en **`tokens_recuperacion`** y despacha el correo electrónico con el código. 7. El sistema redirige a la pantalla de verificación de código OTP y nueva clave. 8. El usuario introduce el código OTP recibido y digita su nueva contraseña con confirmación. 9. El sistema valida que el OTP coincida, no haya expirado y no haya sido utilizado previamente. 10. El sistema cifra la nueva contraseña con **`bcrypt`** (factor 12) y actualiza el registro en **`usuarios`**. 11. El sistema marca el token OTP como utilizado (**`utilizado = true`**), limpia los contadores de intentos fallidos y desbloquea la cuenta. 12. El sistema muestra mensaje de confirmación: "Contraseña actualizada exitosamente" y redirige al login.
Flujos Alternativos	**4a. Correo no registrado:** 1. Por directriz de seguridad contra enumeración de cuentas (OWASP), el sistema responde genéricamente: "Si el correo ingresado se encuentra registrado, le hemos enviado el código de verificación", sin revelar la inexistencia del usuario. **9a. Código OTP erróneo:** 1. El sistema incrementa el contador de fallos del token. 2. Si supera 3 intentos errados, el token se invalida permanentemente y se exige solicitar uno nuevo. **9b. Código OTP expirado (más de 15 minutos):** 1. El sistema informa: "El código OTP ha expirado. Por favor solicite un nuevo código."

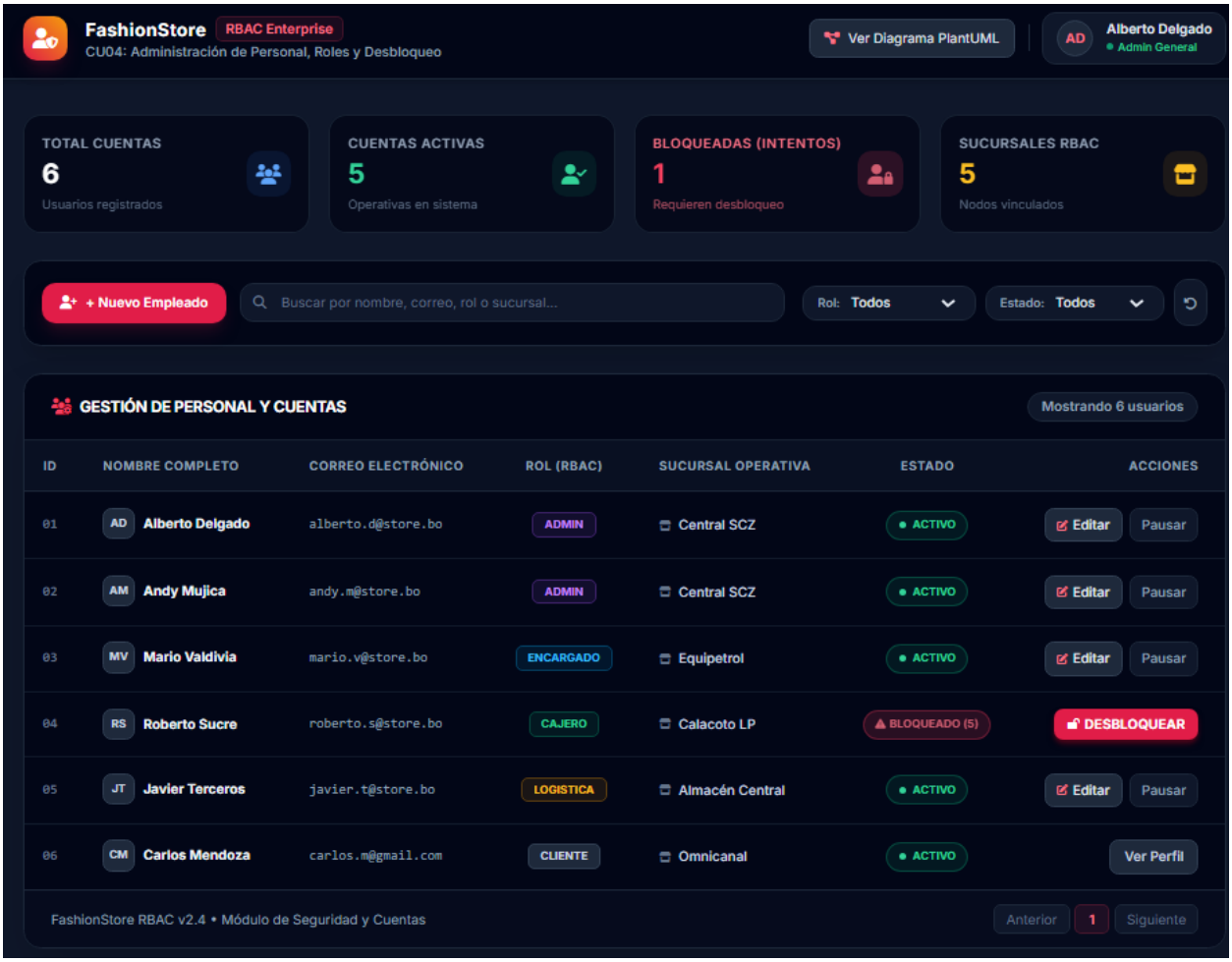
. . . . . .

1.3.4 Caso de Uso CU04: Gestionar Usuarios y Roles (RBAC y Desbloqueo)

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



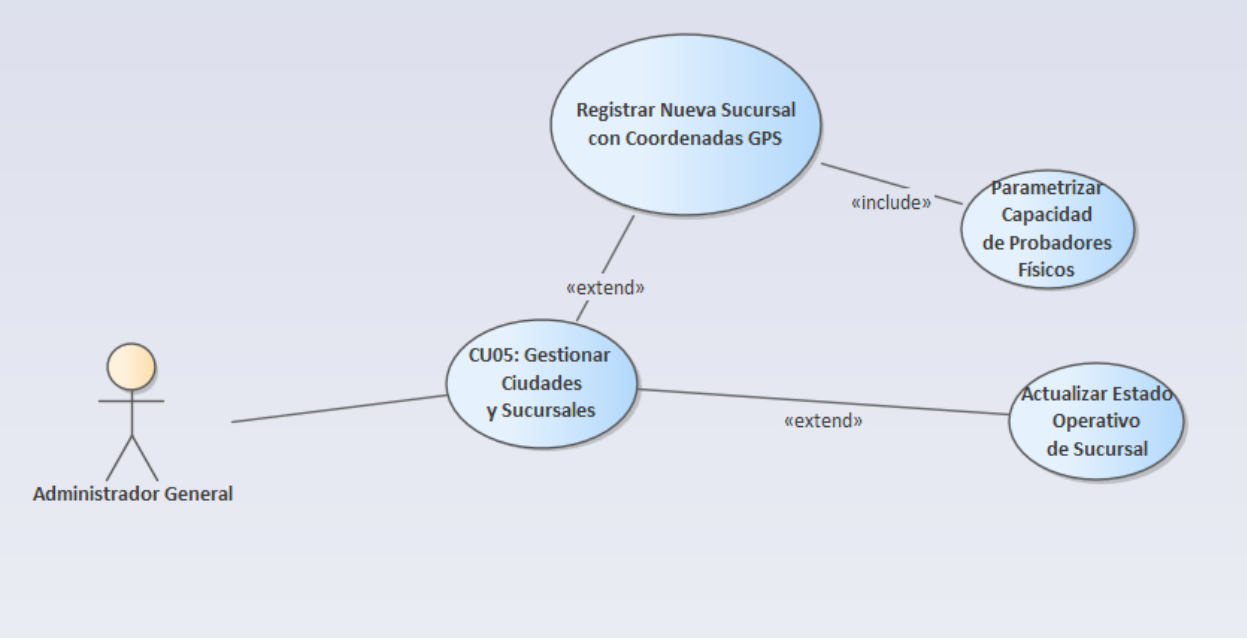
c) Tabla Detalle del Caso de Uso

Atributo	Detalle de la Especificación
Identificador	**CU04**
Nombre	**Gestionar Usuarios y Roles (RBAC y Desbloqueo)**
Actores	Administrador General.
Propósito	Gestionar los usuarios internos de la organización, asignar sus roles y sucursales físicas de desempeño, regular sus privilegios y realizar tareas de mantenimiento operativo (como el desbloqueo de cuentas bloqueadas por intentos fallidos).
Tipo	Primario / Administrativo.
Precondiciones	El usuario debe estar autenticado con rol **`ADMINISTRADOR`**.
Postcondiciones	Se persisten las modificaciones en la tabla **`usuarios`** y se asienta el registro de auditoría en **`bitacora_accesos`**.
Flujo Principal	1. El Administrador accede a la opción "Usuarios y Roles" del menú administrativo. 2. El sistema recupera la lista paginada de usuarios con sus roles y sucursales. 3. El Administrador selecciona "Nuevo Empleado" o hace clic en "Editar" sobre un usuario existente. 4. El sistema presenta el formulario modal con los campos de perfil, selector de rol jerárquico y selector de sucursal física. 5. El Administrador asigna el rol correspondiente (**`ADMINISTRADOR`**, **`ENCARGADO_SUCURSAL`**, **`CAJERO`**, **`LOGISTICA`**) y vincula la sucursal. 6. El Administrador presiona "Guardar Cambios". 7. El sistema valida los datos y persiste las modificaciones en la base de datos. 8. El sistema notifica la confirmación de la operación.
Flujos Alternativos	**3a. Desbloqueo de cuenta bloqueada por 5 intentos:** 1. El Administrador visualiza el badge de estado **`[BLOQ! ]`** en el usuario. 2. El Administrador presiona el botón directo "Desbloquear". 3. El sistema resetea **`intentos_fallidos = 0`**, establece **`estado_cuenta = 'ACTIVO'`** y elimina la marca de tiempo **`bloqueado_hasta`**. 4. El sistema emite mensaje: "Cuenta desbloqueada exitosamente. El usuario ya puede iniciar sesión." **3b. Baja / Inactivación de empleado:** 1. El Administrador selecciona cambiar estado a **`INACTIVO`**. 2. El sistema revoca de inmediato todos los tokens JWT emitidos para dicho usuario.

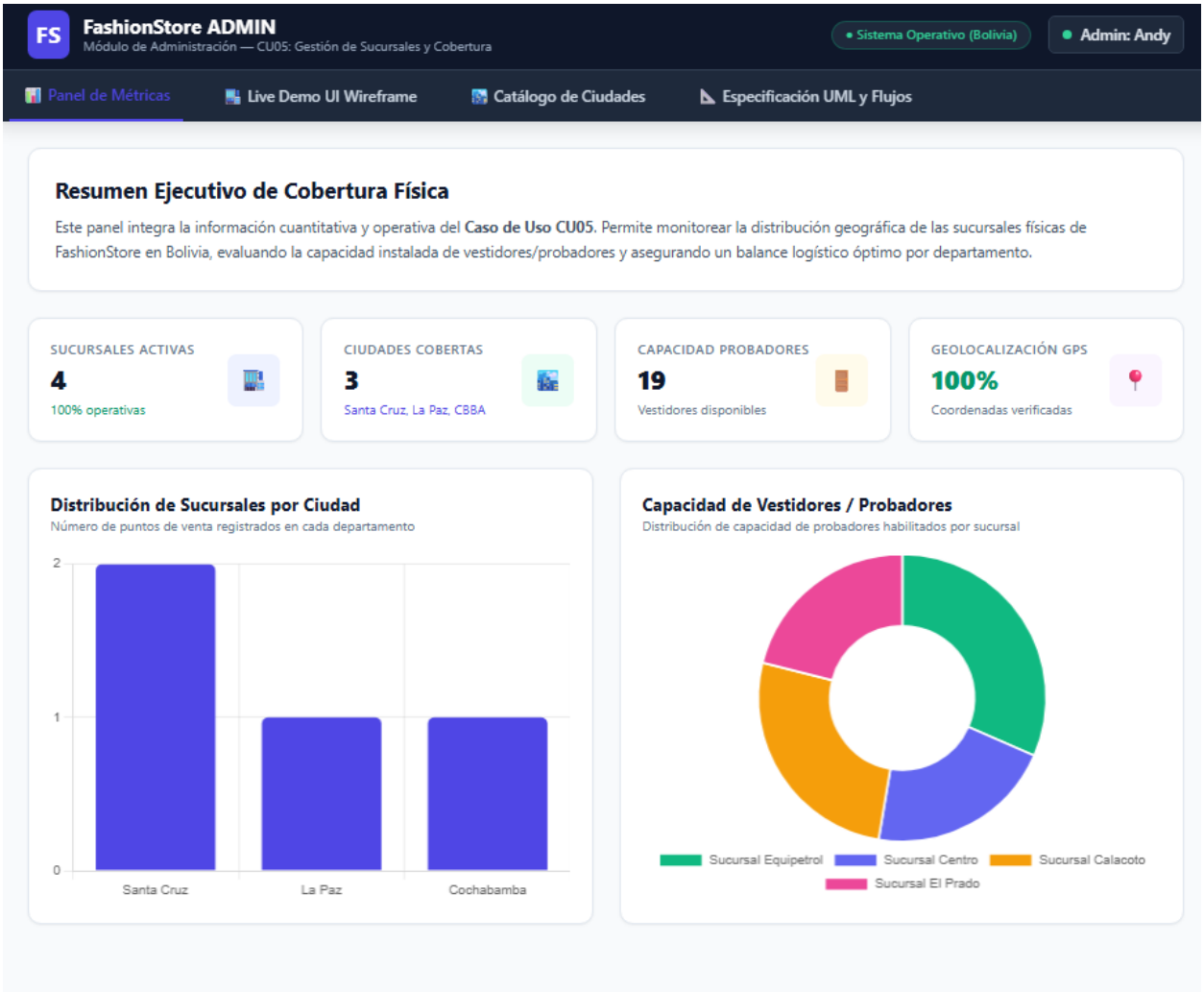
. . . . . .

1.3.5 Caso de Uso CU05: Gestionar Ciudades y Sucursales

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



c) Tabla Detalle del Caso de Uso

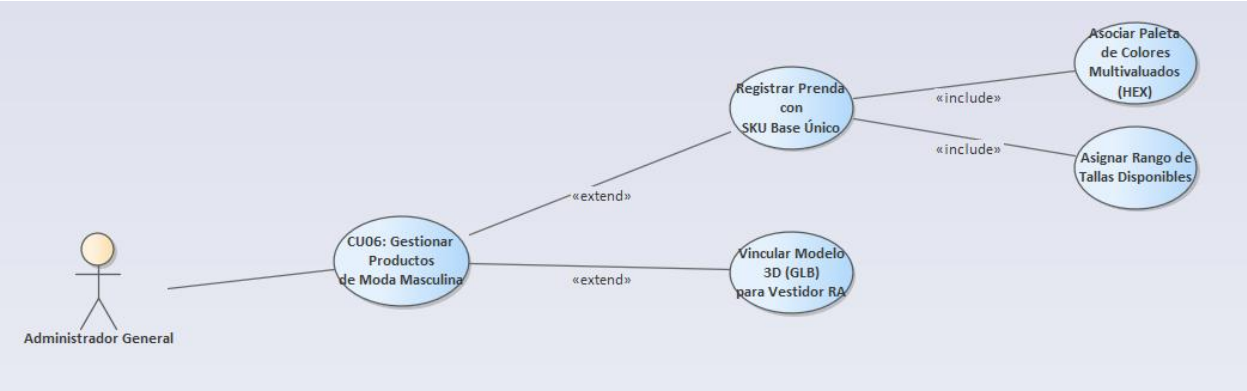
Atributo	Detalle de la Especificación
Identificador	**CU05**
Nombre	**Gestionar Ciudades y Sucursales**
Actores	Administrador General.
Propósito	Crear, configurar, actualizar y georreferenciar las ciudades y sucursales físicas habilitadas para la venta presencial, almacenamiento de existencias y reservas de probadores.
Tipo	Secundario / Gestión Operativa.
Precondiciones	1. El actor debe haberse autenticado exitosamente como Administrador General. 2. La ciudad de radicación debe encontrarse dada de alta en el sistema.
Postcondiciones	1. La sucursal queda registrada en la base de datos con coordenadas geodésicas válidas. 2. La sucursal queda habilitada como nodo geográfico para asignación de inventario y reservas.
Flujo Principal	1. El Administrador accede al panel de "Gestión de Sucursales". 2. El sistema lista las sucursales existentes con sus métricas clave. 3. El Administrador pulsa en "Registrar Nueva Sucursal". 4. El Administrador completa: ciudad, nombre comercial, dirección, coordenadas GPS, horarios de apertura/cierre y capacidad de probadores. 5. El Administrador asigna al Encargado de Sucursal responsable. 6. El sistema valida consistencia de datos y unicidad del nombre en la ciudad. 7. El sistema persiste la sucursal y notifica confirmación en pantalla.

Flujos Alternativos	**6a. Coordenadas fuera de rango:** El sistema notifica error de latitud/longitud e invita a seleccionar el punto en el mapa interactivo. **6b. Encargado ya asignado:** El sistema advierte que el usuario ya es titular en otra tienda y solicita confirmación de reasignación.

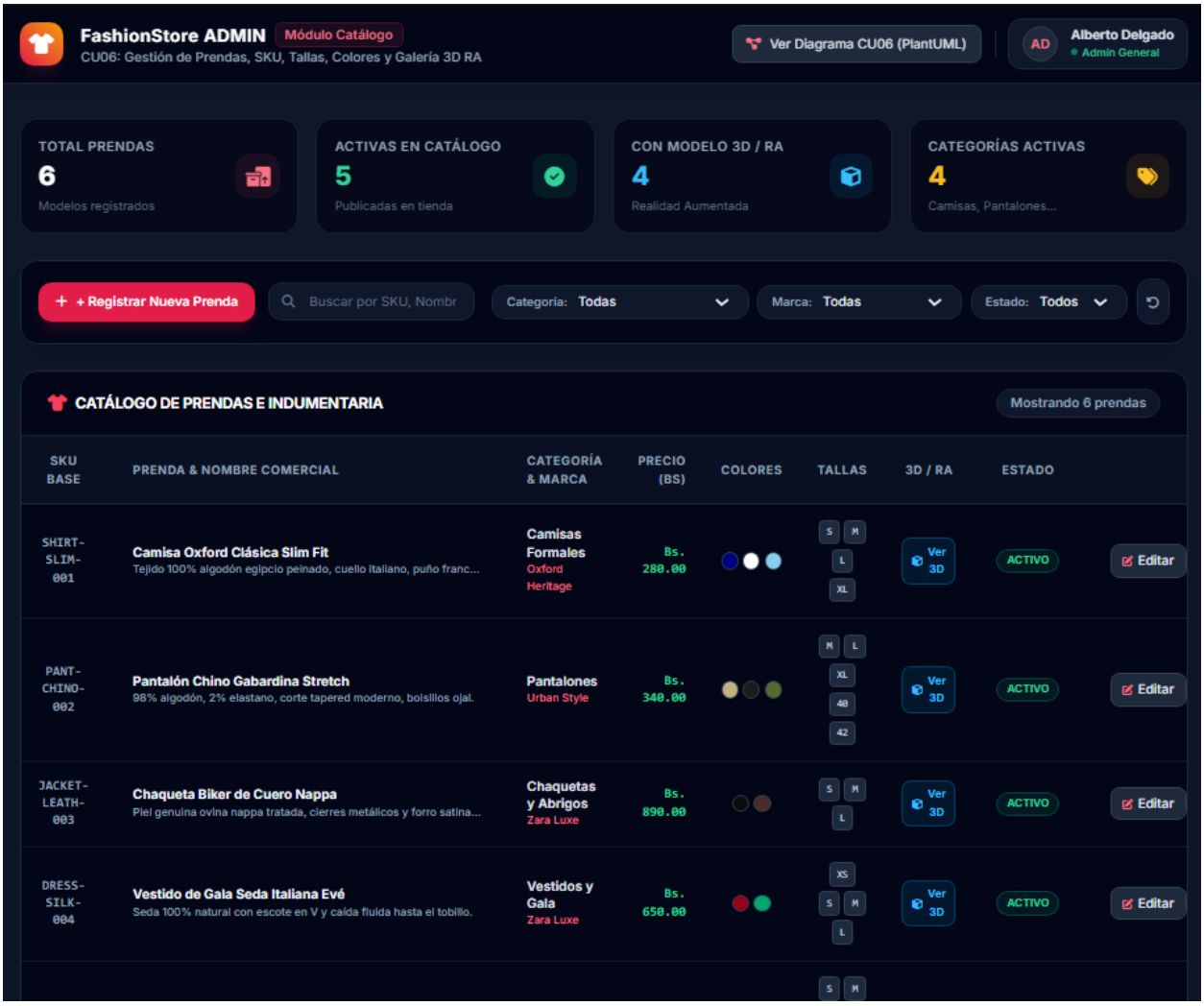
. . . . . .

1.3.6 Caso de Uso CU06: Gestionar Productos y Atributos de Moda

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



c) Tabla Detalle del Caso de Uso

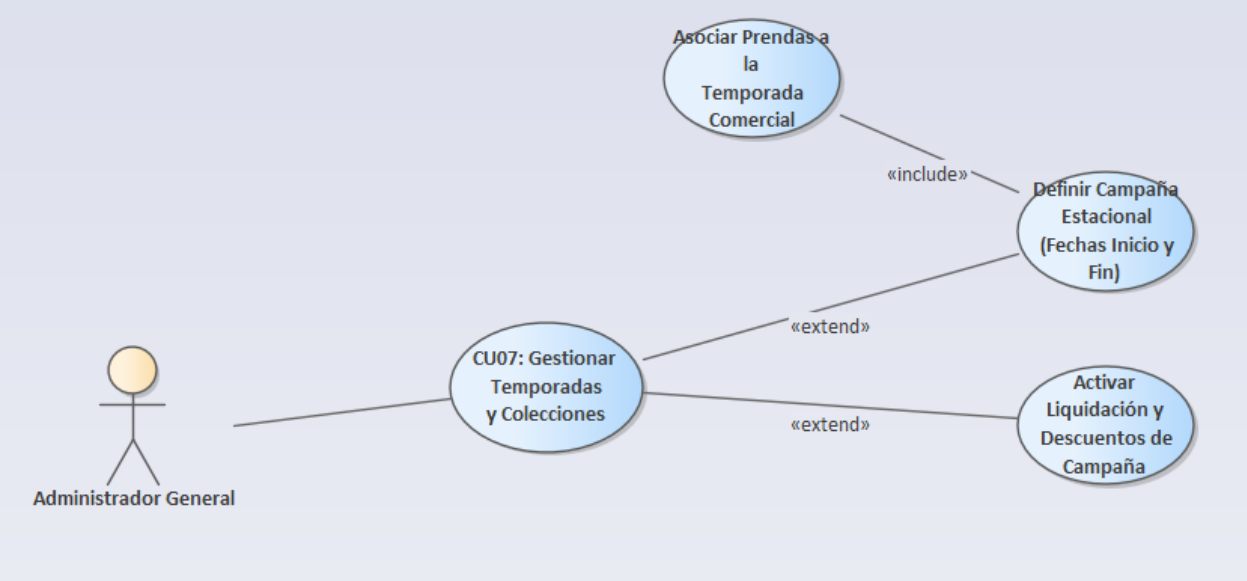
Atributo	Detalle de la Especificación
Identificador	**CU06**
Nombre	**Gestionar Productos y Atributos de Moda**
Actores	Administrador General.
Propósito	Registrar y actualizar prendas de vestir masculinas con sus especificaciones técnicas, variantes de tallas, colores multivaluados con código visual y recursos multimedia para el vestidor virtual.
Tipo	Primario.
Precondiciones	La categoría y marca deben encontrarse registradas previamente en el sistema.
Postcondiciones	1. El producto queda guardado con su código SKU base único. 2. Se generan las matrices relacionales de variantes (combinación producto-talla-color). 3. Las variantes quedan preparadas para recibir stock en el módulo de inventario.
Flujo Principal	1. El Administrador accede al panel de productos y presiona "Registrar Nueva Prenda". 2. El sistema despliega el formulario con los campos taxonómicos. 3. El Administrador introduce el código SKU base, nombre comercial, descripción y precio de venta base. 4. El Administrador selecciona la categoría textil y la marca. 5. El Administrador tilda las tallas disponibles para el modelo. 6. El Administrador agrega la paleta de colores multivaluados indicando nombre comercial y código hexadecimal. 7. El Administrador sube las imágenes en alta definición y el modelo 3D (.glb).

	8. El sistema valida la integridad de los datos y no duplicidad de SKU. 9. El sistema persiste el producto y sus variantes relacionales.
Flujos Alternativos	**8a. SKU duplicado:** El sistema rechaza la operación e indica que el código ya pertenece a otro artículo. **8b. Archivo 3D inválido:** El sistema valida la cabecera del archivo binario y rechaza formatos distintos a .glb / .gltf.

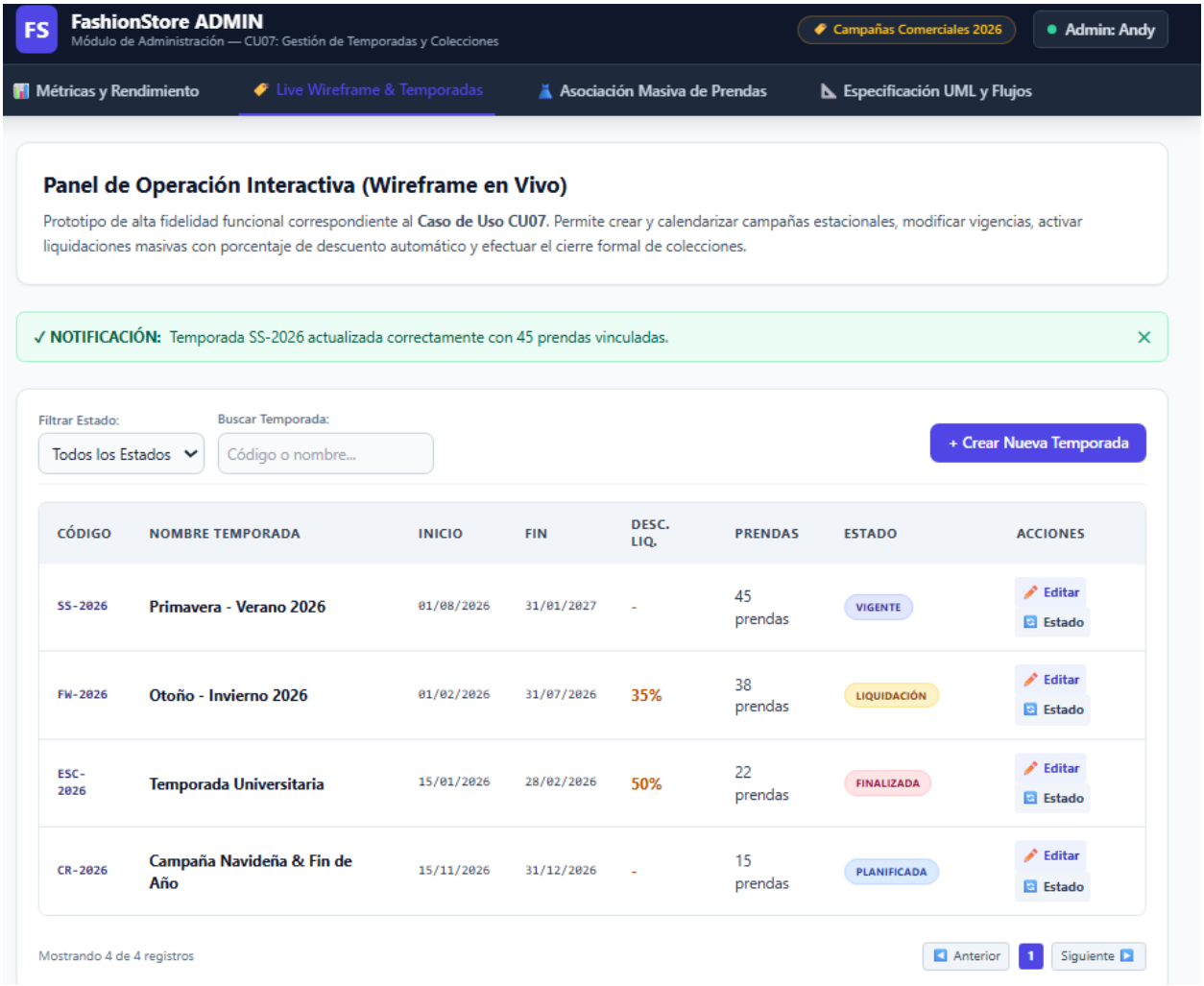
. . . . . .

1.3.7 Caso de Uso CU07: Gestionar Temporadas y Colecciones

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



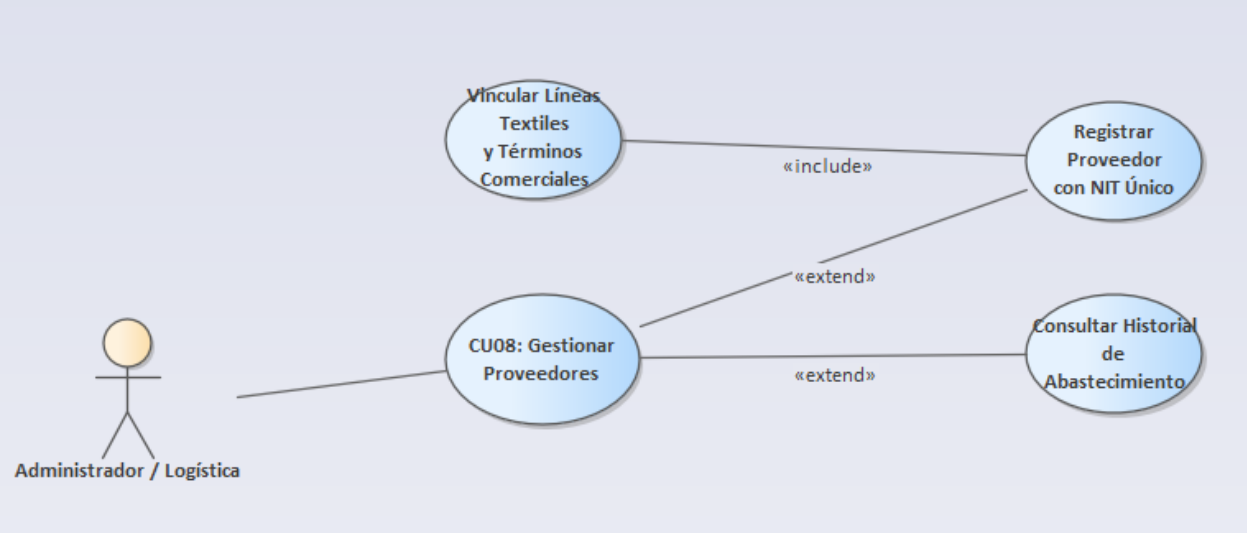
c) Tabla Detalle del Caso de Uso

Atributo	Detalle de la Especificación
Identificador	**CU07**
Nombre	**Gestionar Temporadas y Colecciones**
Actores	Administrador General.
Propósito	Crear y estructurar temporadas comerciales anuales, vinculando lotes de prendas para análisis estacional de rotación y reglas de IA contextual.
Tipo	Secundario.
Precondiciones	El usuario debe poseer privilegios de Administrador.
Postcondiciones	Las prendas asociadas adquieren el identificador de temporada para alimentar los filtros del catálogo y el motor de IA.
Flujo Principal	1. El Administrador accede al panel de "Temporadas y Colecciones". 2. El sistema muestra la lista de temporadas vigentes y concluidas. 3. El Administrador ingresa código, nombre y rango de fechas de inicio y fin. 4. El Administrador selecciona las categorías y productos que pertenecerán a la colección. 5. El sistema valida que la fecha de fin sea posterior a la de inicio. 6. El sistema actualiza la vinculación y recalcula el catálogo público.
Flujos Alternativos	**5a. Fechas inconsistentes:** El sistema emite alerta: "La fecha de cierre debe ser posterior a la fecha de inicio de la temporada."

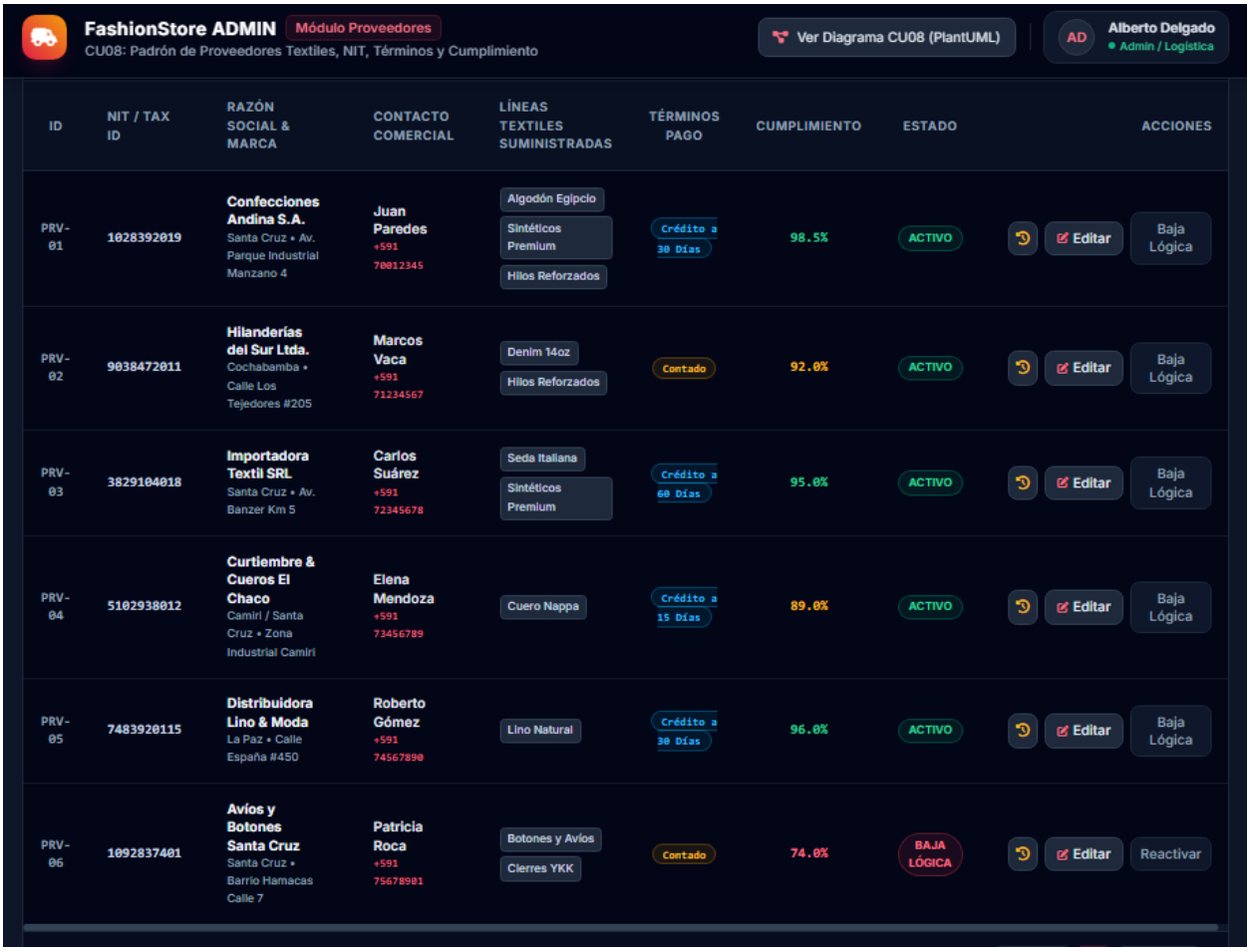
. . . . . .

1.3.8 Caso de Uso CU08: Gestionar Proveedores Textiles

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



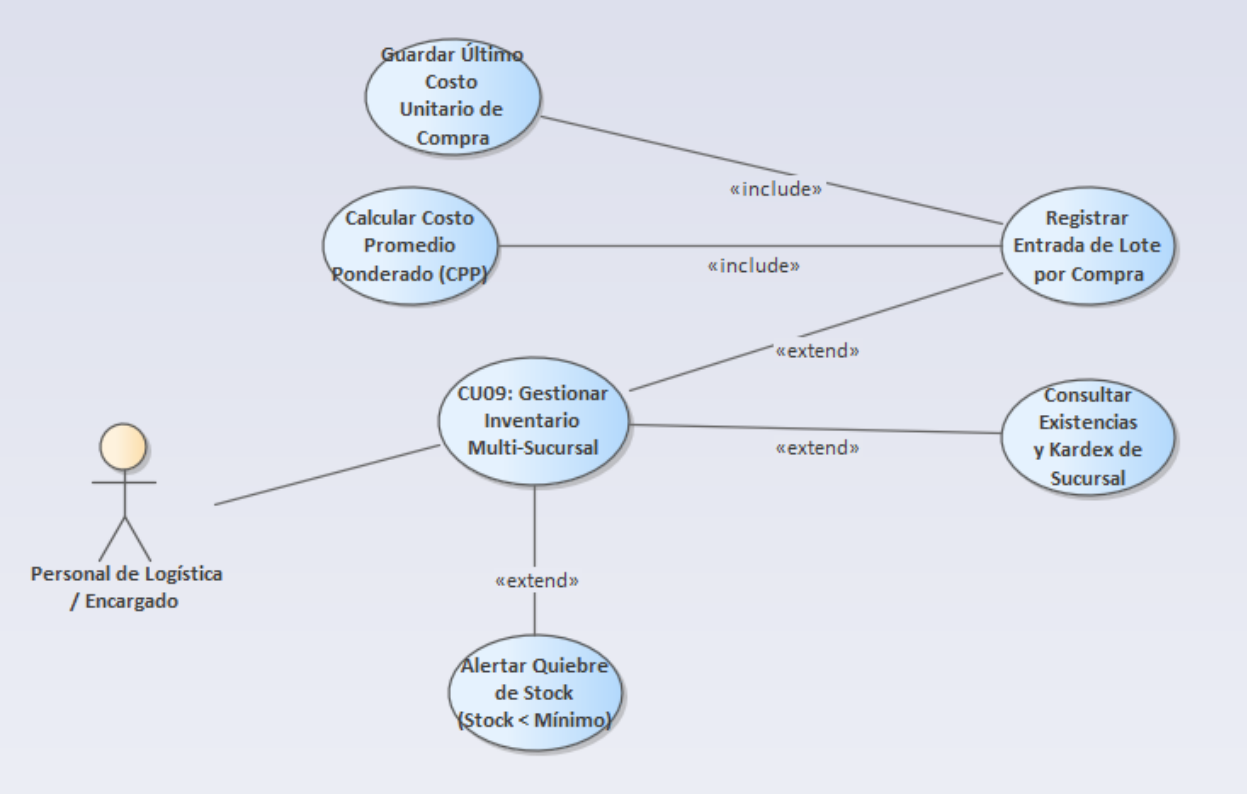
c) Tabla Detalle del Caso de Uso

Atributo	Detalle de la Especificación
Identificador	**CU08**
Nombre	**Gestionar Proveedores Textiles**
Actores	Administrador General, Personal de Logística.
Propósito	Administrar el catálogo de proveedores que abastecen mercadería textil a FashionStore para sustentar las órdenes de compra y el ingreso a almacén.
Tipo	Secundario / Soporte.
Precondiciones	Usuario con rol Administrador o Logística autenticado.
Postcondiciones	El proveedor queda disponible para ser seleccionado en los registros de compra y entrada de inventario.
Flujo Principal	1. El usuario accede al menú de "Proveedores". 2. El sistema lista los proveedores activos. 3. El usuario pulsa "Registrar Proveedor" e ingresa NIT, Razón Social, teléfono, email y términos de pago. 4. El sistema valida que el NIT no exista previamente. 5. El sistema registra el proveedor y muestra mensaje de éxito.
Flujos Alternativos	**4a. NIT duplicado:** El sistema notifica: "El NIT ingresado ya se encuentra registrado con la razón social X."

. . . . . .

1.3.9 Caso de Uso CU09: Gestionar Inventario Multi-Sucursal y Costos Ponderados (CPP)

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)

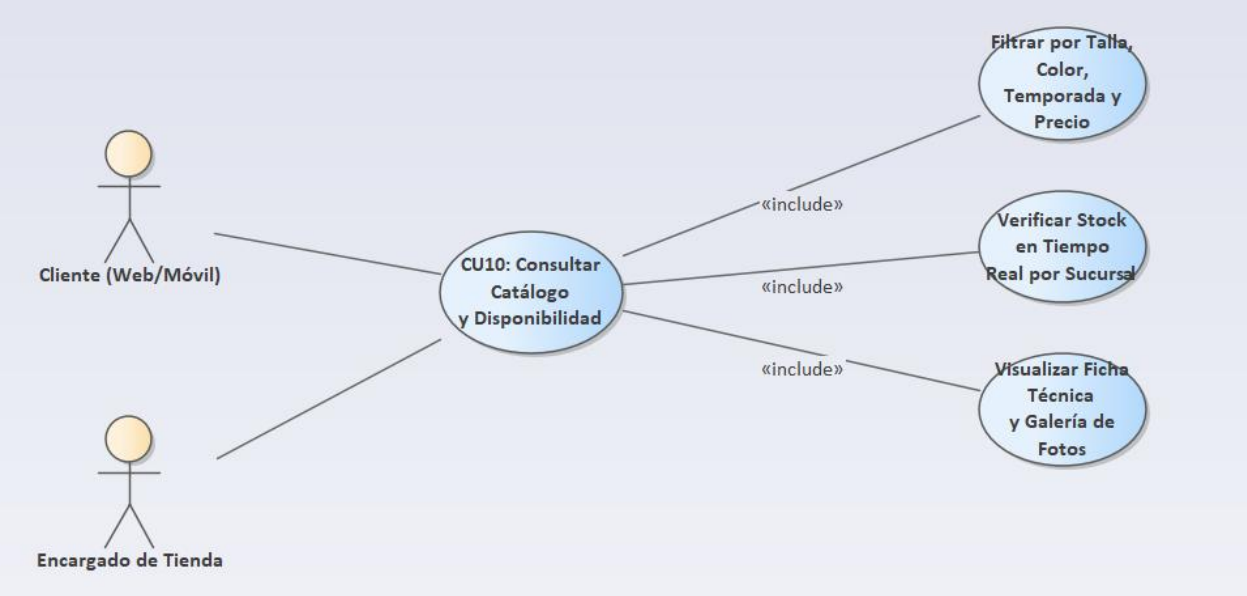
c) Tabla Detalle del Caso de Uso

Atributo	Detalle de la Especificación
Identificador	**CU09**
Nombre	**Gestionar Inventario Multi-Sucursal y Costos Ponderados (CPP)**
Actores	Personal de Logística, Encargado de Sucursal.
Propósito	Registrar ingresos de mercadería, controlar existencias físicas y disponibles por tienda, almacenar el último costo unitario y recalcular matemáticamente el Costo Promedio Ponderado para fines de valuación patrimonial y márgenes reales.
Tipo	Crítico / Núcleo Transaccional.
Precondiciones	El producto, talla, color, sucursal y proveedor deben existir en la base de datos.
Postcondiciones	1. Se incrementa el stock físico y disponible de la variante en la sucursal. 2. Se asienta el último costo de compra. 3. Se actualiza el **CPP**. 4. Se registra el asiento inmutable en el Kardex de inventario.
Flujo Principal	1. El usuario de Logística selecciona "Registrar Entrada de Mercadería". 2. Elige la sucursal receptora, el proveedor y el producto específico con su talla y color. 3. Ingresa la cantidad física del lote recibido y el costo unitario de adquisición de la factura. 4. El sistema recupera el stock actual y el CPP anterior de la variante en esa sucursal. 5. El sistema aplica la fórmula estandarizada: $CPP_{\text{nuevo}} = \frac{(Stock_{\text{ant}} \times CPP_{\text{ant}}) + (Cantidad_{\text{lote}} \times Costo_{\text{lote}})}{Stock_{\text{ant}} + Cantidad_{\text{lote}}}$ 6. El sistema muestra el recálculo al usuario para verificación. 7. El usuario confirma el asiento. 8. El sistema ejecuta una transacción ACID: actualiza tabla **`inventario`** y añade fila en **`kardex_movimientos`**.
Flujos Alternativos	**3a. Costo o cantidad cero/negativo:** El sistema bloquea el registro exigiendo valores estrictamente mayores a cero.

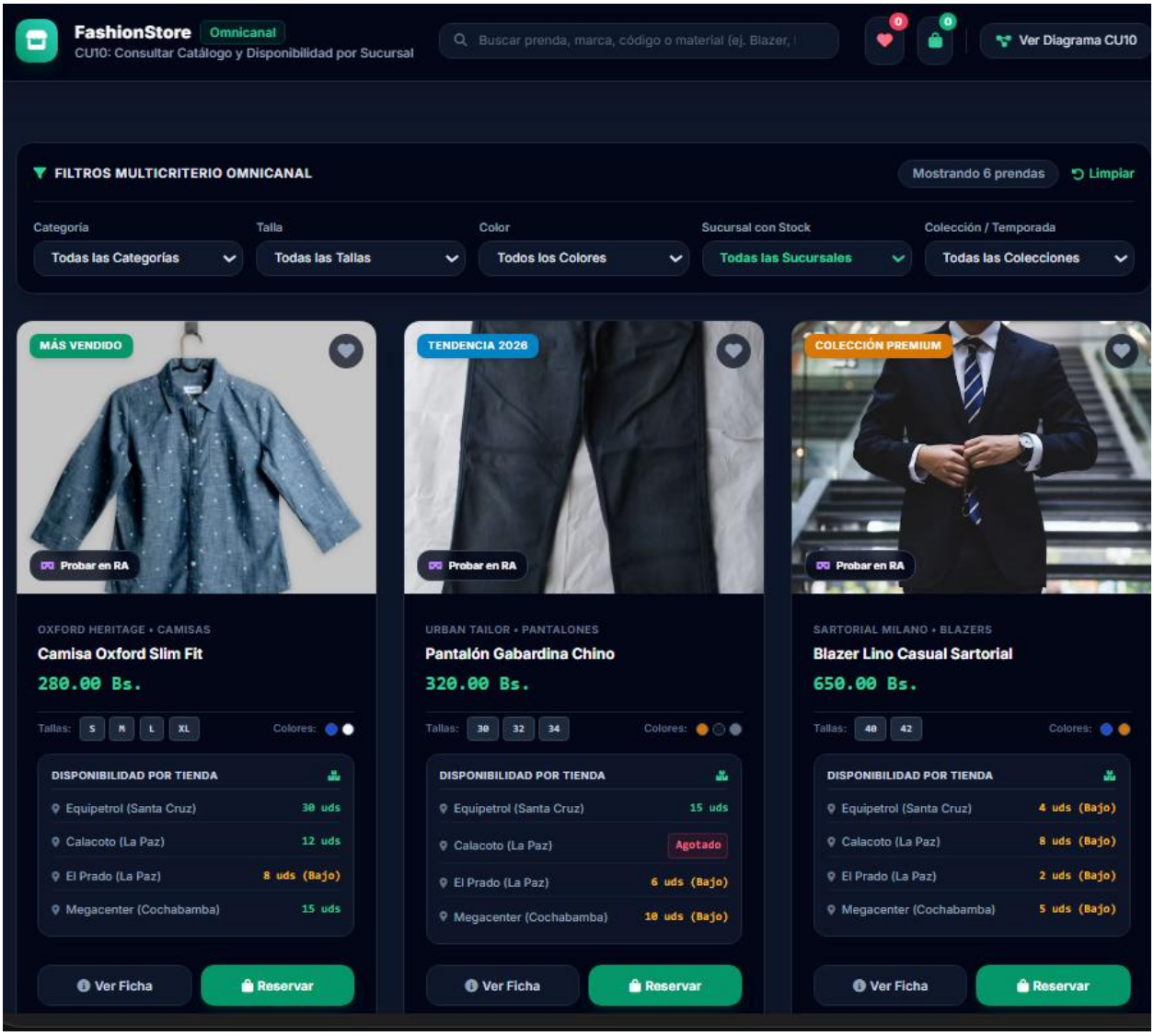
. . . . . .

1.3.10 Caso de Uso CU10: Consultar Catálogo y Disponibilidad por Sucursal

a) Diseño del Caso de Uso (PlantUML)



b) Prototipo de Interfaz de Usuario (UI Wireframe)



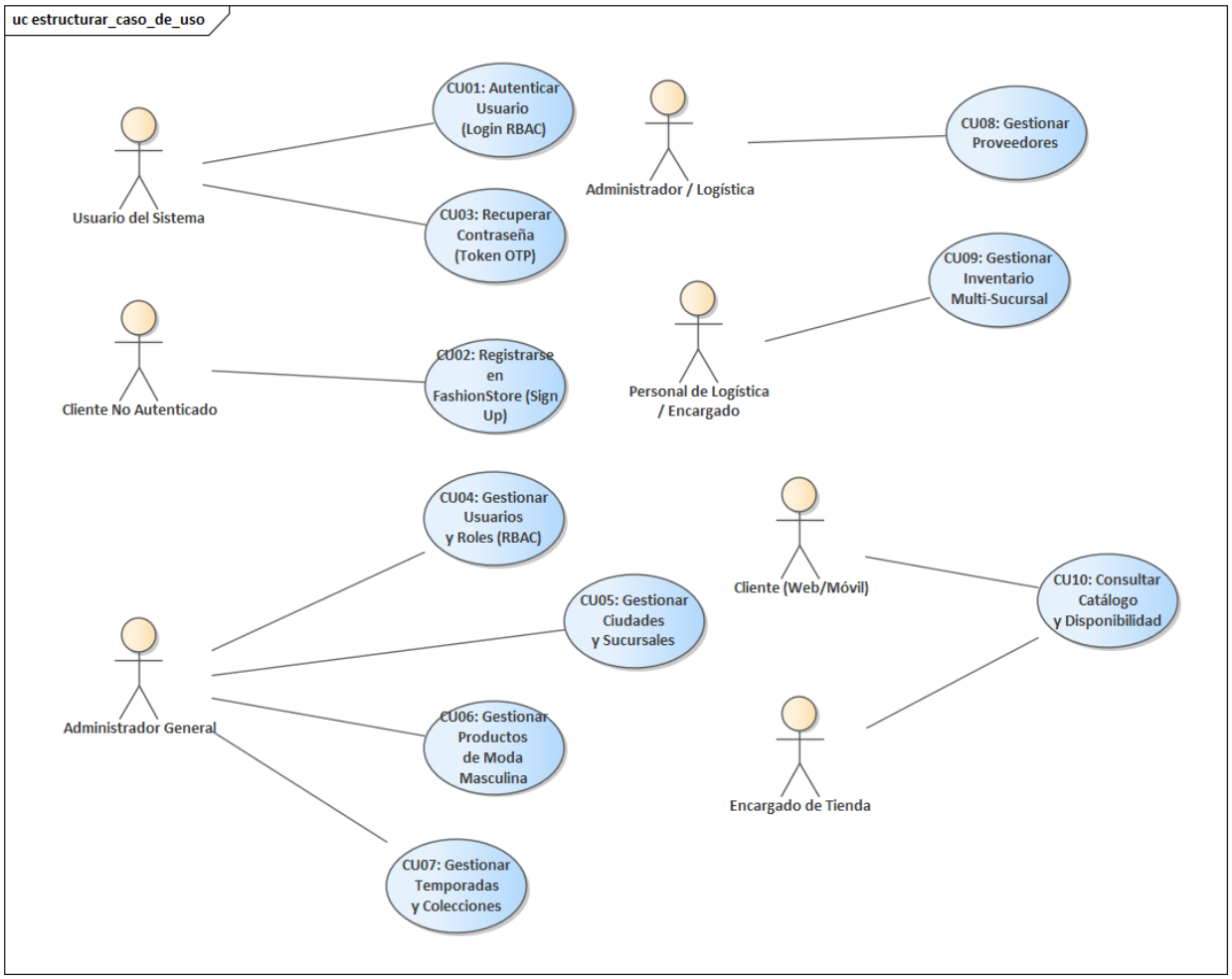
c) Tabla Detalle del Caso de Uso

Atributo	Detalle de la Especificación
Identificador	**CU10**
Nombre	**Consultar Catálogo y Disponibilidad por Sucursal**
Actores	Cliente (Web y Móvil), Encargado de Sucursal, Cajero.
Propósito	Permitir a los usuarios explorar las prendas, aplicar filtros cruzados y conocer con absoluta precisión en qué sucursales físicas existe stock de su talla y color elegido.
Tipo	Primario / Omnicanal.
Precondiciones	La plataforma web o app móvil debe tener acceso a la API pública de catálogo.
Postcondiciones	El cliente obtiene la lista de prendas con stock disponible en tiempo real para proceder a la reserva o compra digital.
Flujo Principal	1. El usuario accede al catálogo de FashionStore. 2. El sistema recupera y pagina las prendas publicadas con sus imágenes y variantes. 3. El usuario aplica filtros (ej. Talla M, Color Azul, Sucursal Equipetrol). 4. El sistema ejecuta una consulta indexada y devuelve las prendas coincidentes con su stock por tienda. 5. El usuario selecciona una prenda para ver su ficha técnica completa y guía de medidas.

	6. El sistema presenta opciones de acción (Probar en RA, Reservar para prueba presencial, o Añadir al carrito).
Flujos Alternativos	**4a. Sin stock en la sucursal seleccionada:** El sistema muestra la etiqueta "Agotado en esta sucursal" y resalta inmediatamente en cuáles otras tiendas de la ciudad sí existe stock disponible.

1.4 Estructurar Modelo de Casos de Uso

A fin de organizar modularmente los requerimientos del Ciclo 1 y evidenciar la cohesión funcional del software, se estructura el modelo de casos de uso organizado en **Paquetes de Casos de Uso**:



2. Flujo de Trabajo: Análisis

2.1 Análisis de Arquitectura

El análisis arquitectónico del Ciclo 1 descompone el sistema en subsistemas y paquetes de análisis de alto nivel ,formalizando la relación estricta entre paquetes y los 10 Casos de Uso desarrollados, aplicando los estereotipos de robustez de Ivar Jacobson:

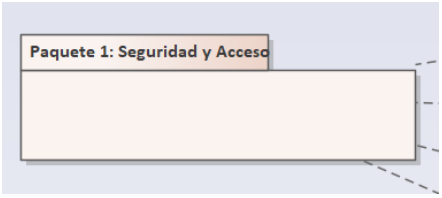
• **Clases de Interfaz (`<<boundary>>`)**: Encargadas de la interacción y comunicación directa con los actores humanos y externos (formularios de entrada, pantallas interactivas y endpoints API REST). Poseen atributos de captura visual y métodos de interacción (eventos de usuario).

• **Clases de Control (`<<control>>`)**: Encargadas de orquestar la lógica de negocio, reglas algorítmicas, validaciones y transformaciones de dominio. Como subraya estrictamente la cátedra ,**las clases de control contienen exclusivamente métodos de negocio y NO poseen atributos propios**, operando directamente sobre las entidades de datos.

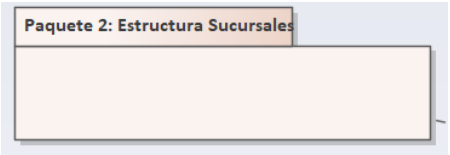
• **Clases de Entidad (`<<entity>>`)**: Representan la información persistente y los conceptos transaccionales del dominio (tablas, campos y registros de PostgreSQL).

2.1.1 Identificar Paquetes

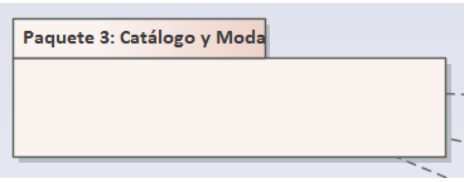
En esta etapa se identifican los **5 paquetes de análisis** que componen el núcleo arquitectónico del Ciclo 1, especificando para cada uno el nombre del paquete y una descripción concisa de su propósito, responsabilidades y alcance funcional:



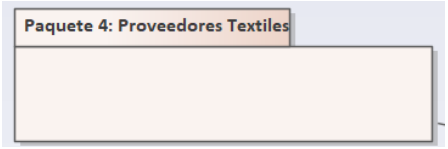
• **Paquete 1: Seguridad y Acceso (RBAC)** *Descripción:* Gestiona el control de acceso, la autenticación criptográfica de credenciales mediante hashing Bcrypt, la emisión y validación de tokens de sesión JWT, el auto-registro de nuevos clientes, la recuperación autoservicio de contraseñas mediante códigos OTP de un solo uso y la administración integral de usuarios, roles y permisos del sistema.



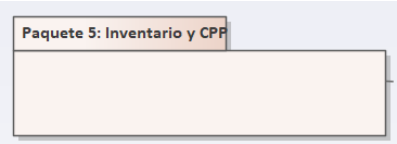
• **Paquete 2: Estructura Operativa (Sucursales)** *Descripción:* Administra la infraestructura física y geográfica de la cadena minorista, incluyendo ciudades operativas, sucursales físicas habilitadas, georreferenciación satelital GPS (latitud y longitud), horarios de atención y la parametrización de capacidad física de probadores para reservas presenciales.



• **Paquete 3: Catálogo y Moda Masculina** *Descripción:* Centraliza la definición de prendas de vestir masculinas con atributos multivaluados normalizados (tallas estándar, códigos cromáticos hexadecimales HEX y nombres de color), la calendarización y gestión de temporadas comerciales (Spring-Summer / Fall-Winter) y la publicación omnicanal del catálogo de productos.



• **Paquete 4: Aprovisionamiento y Proveedores** *Descripción:* Gestiona el directorio comercial de empresas proveedoras de textiles y confección masculina, la validación estricta de identificación tributaria (NIT único normalizado), los canales de contacto y los términos comerciales de suministro y condiciones de pago.



• **Paquete 5: Inventario y Costos Ponderados (CPP)** *Descripción:* Controla las existencias físicas multi-sucursal en tiempo real, el registro de ingresos de lotes por compra, el recálculo matemático algorítmico del Costo Promedio Ponderado (CPP*CPP*)

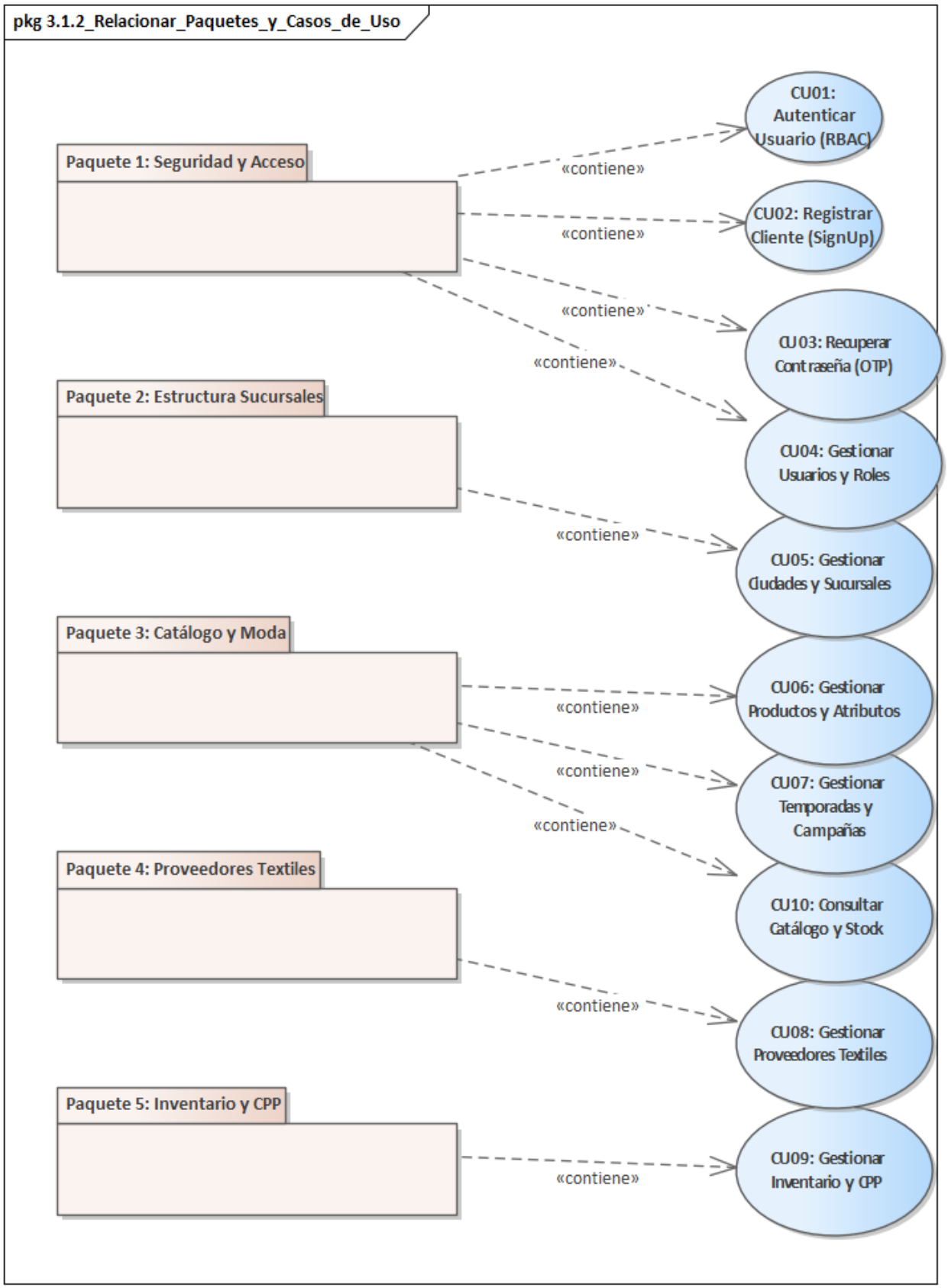
ante cada entrada de mercadería y la generación inmutable de asientos de auditoría en el Kardex valorizado.

2.1.2 Relacionar paquetes y casos de uso

En esta sección se formaliza la relación estricta entre los **5 paquetes de análisis del Ciclo 1** y los **10 Casos de Uso** implementados en esta primera iteración (`CU01` al `CU10`), estableciendo la cohesión modular y el límite de responsabilidad funcional de cada subsistema:

Paquete de Análisis del Ciclo	**Casos de Uso Contenidos**	**Actores Asociados**	**Justificación y Responsabilidad de Arquitectura**
Paquete 1: Seguridad y Acceso (RBAC)	• **CU01**: Autenticar Usuario (Login RBAC) • **CU02**: Registrar Cliente (Sign Up) • **CU03**: Recuperar Contraseña (OTP) • **CU04**: Gestionar Usuarios y Roles	Usuario, Cliente, Administrador	Centraliza el control perimetral de acceso, autenticación criptográfica con Bcrypt, emisión de tokens JWT, alta autoservicio de clientes, recuperación con OTP de 6 dígitos y administración centralizada de cuentas de usuario y roles.
Paquete 2: Estructura Operativa (Sucursales)	• **CU05**: Gestionar Ciudades y Sucursales	Administrador General	Provee la estructura territorial multiciudad, parametrización de coordenadas GPS (latitud/longitud) para mapas digitales, control de horarios y aforo físico de probadores de ropa.
Paquete 3: Catálogo y Moda Masculina	• **CU06**: Gestionar Productos y Atributos • **CU07**: Gestionar Temporadas y Campañas • **CU10**: Consultar Catálogo y Disponibilidad	Administrador, Cliente, Encargado Sucursal	Gestiona las fichas técnicas de prendas para varón, atributos multivaluados normalizados (colores HEX y tallas), calendarización de temporadas comerciales (SS/FW) y catálogo público omnicanal con stock por sucursal.

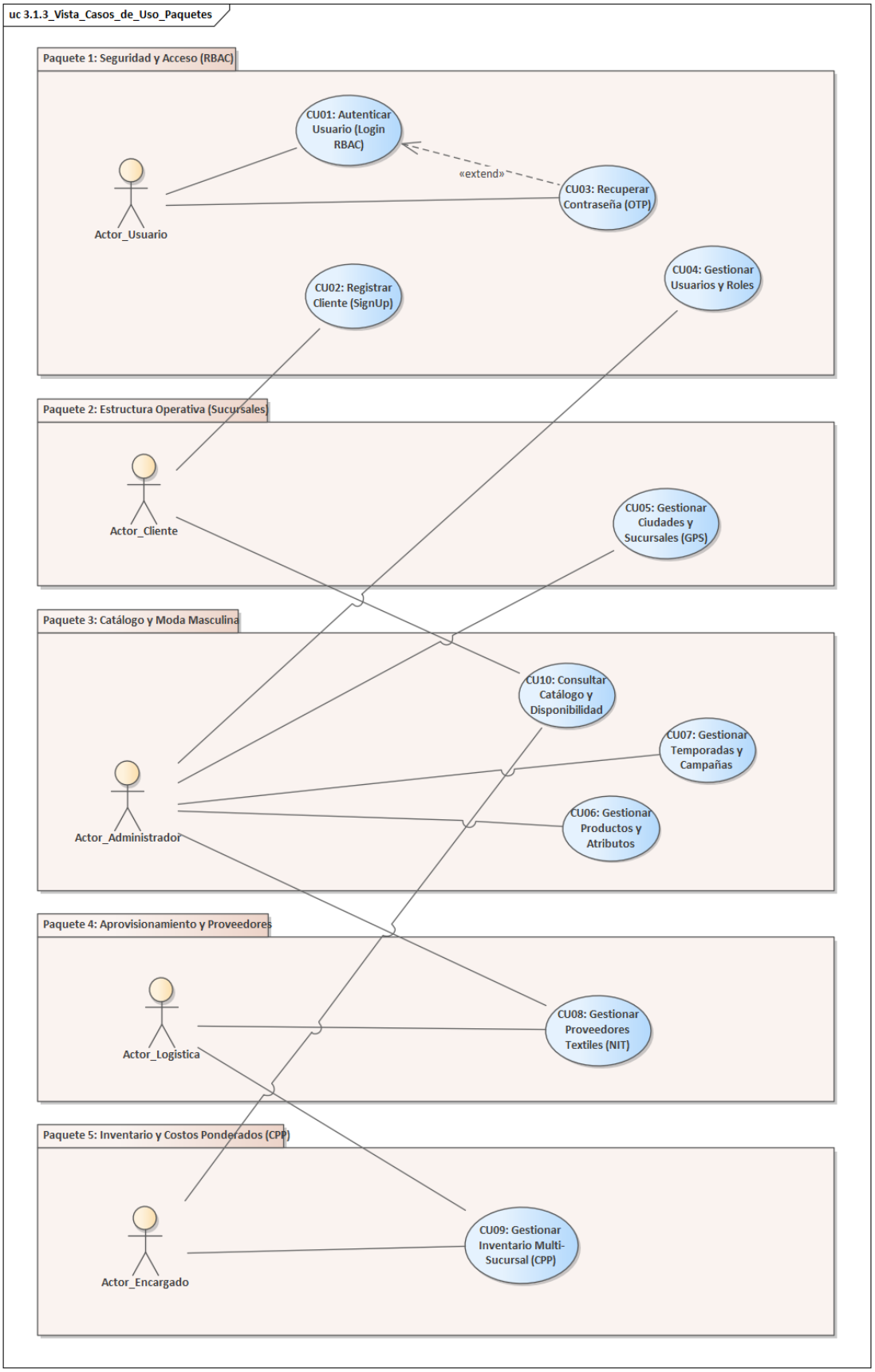
Paquete 4: Aprovisionamiento y Proveedores	• **CU08**: Gestionar Proveedores Textiles	Administrador, Personal Logística	Administra el padrón de proveedores de confección textil, validación de NIT tributario único, información de contacto corporativo y términos comerciales de crédito y pago.
Paquete 5: Inventario y Costos Ponderados (CPP)	• **CU09**: Gestionar Inventario Multi-Sucursal	Personal Logística, Encargado Sucursal	Controla el stock físico de prendas distribuido en cada tienda, registro de ingresos por compra, recálculo algorítmico del Costo Promedio Ponderado (CPPCPP) y registro inmutable en el Kardex valorizado.


pkg 3.1.2_Relacionar_Paquetes_y_Casos_de_Uso
Paquete 1: Seguridad y Acceso
Paquete 2: Estructura Sucursales
Paquete 3: Catálogo y Moda
Paquete 4: Proveedores Textiles
Paquete 5: Inventario y CPP
«contiene»
CU01: Autenticar Usuario (RBAC)
CU02: Registrar Cliente (SignUp)
CU03: Recuperar Contraseña (OTP)
CU04: Gestionar Usuarios y Roles
CU05: Gestionar Ciudades y Sucursales
CU06: Gestionar Productos y Atributos
CU07: Gestionar Temporadas y Campañas
CU10: Consultar Catálogo y Stock
CU08: Gestionar Proveedores Textiles
CU09: Gestionar Inventario y CPP

2.1.3 Vista de casos de uso

En la metodología PUDS y el modelado arquitectónico UML, la Vista de Casos de Uso representa a los paquetes vistos desde su interior: el entorno/frontera exterior es el propio paquete contenedor y dentro de él residen los diagramas de casos de uso que dicho paquete contiene, junto con sus relaciones funcionales internas (<<include>>, <<extend>>) y la conexión con los Actores que interactúan con ellos desde fuera del paquete.

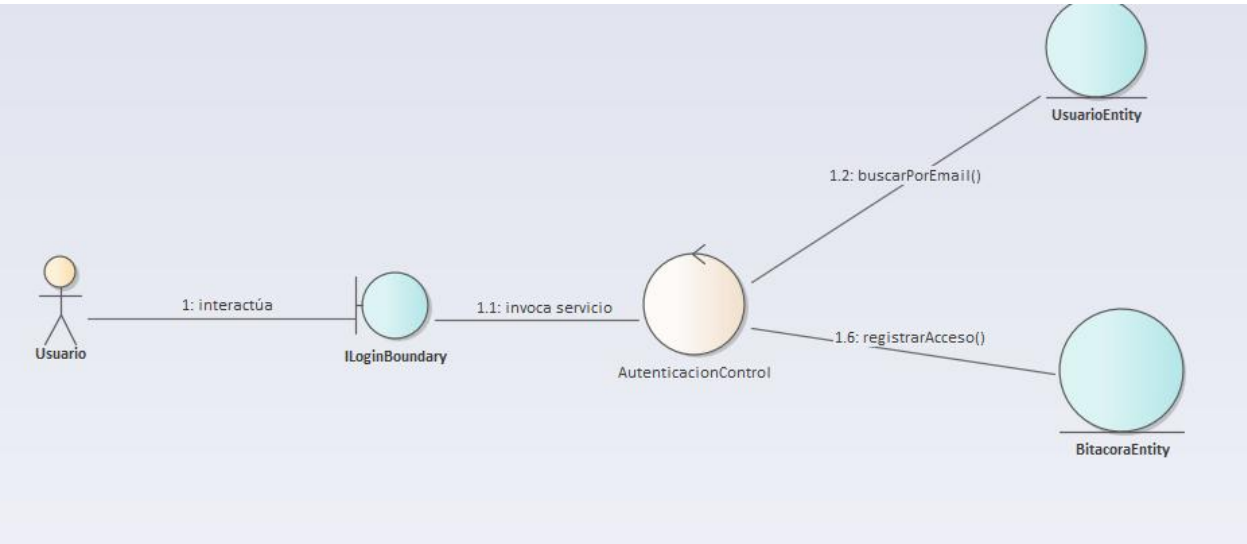
A continuación, se presenta el diagrama consolidado de la Vista de Casos de Uso por Paquete:


uc 3.1.3_Vista_Casos_de_Uso_Paquetes
Paquete 1: Seguridad y Acceso (RBAC)
CU01: Autenticar Usuario (Login RBAC)
«extend»
CU03: Recuperar Contraseña (OTP)
Actor_Usuario
CU02: Registrar Cliente (SignUp)
CU04: Gestionar Usuarios y Roles
Paquete 2: Estructura Operativa (Sucursales)
Actor_Cliente
CU05: Gestionar Ciudades y Sucursales (GPS)
Paquete 3: Catálogo y Moda Masculina
CU10: Consultar Catálogo y Disponibilidad
CU07: Gestionar Temporadas y Campañas
Actor_Administrador
CU06: Gestionar Productos y Atributos
Paquete 4: Aprovisionamiento y Proveedores
Actor_Logistica
CU08: Gestionar Proveedores Textiles (NIT)
Paquete 5: Inventario y Costos Ponderados (CPP)
Actor_Encargado
CU09: Gestionar Inventario Multi-Sucursal (CPP)

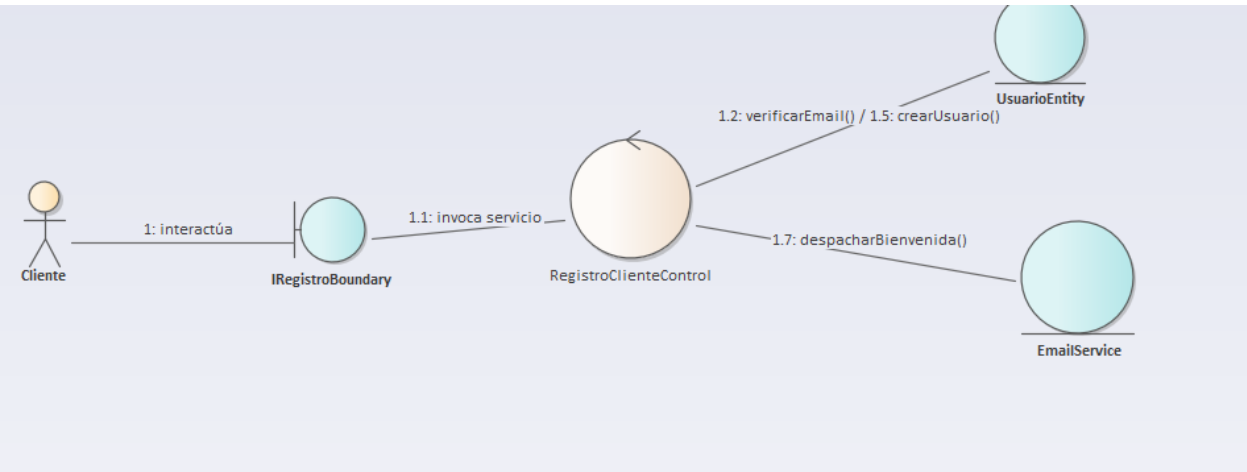
2.2 Analizar Casos de Uso (Diagramas de Comunicación UML)

A continuación se presentan los **Diagramas de Comunicación UML** para cada uno de los 10 Casos de Uso del Ciclo 1. Conforme a las directrices de la cátedra expresadas en clase ,cada caso de uso se analiza de forma rigurosa modelando la colaboración entre la clase de interfaz (`<<boundary>>`), la clase de lógica (`<<control>>` sin atributos) y la clase de datos (`<<entity>>`), con mensajes numerados cronológicamente (`1`, `1.1`, `1.2`, etc.):

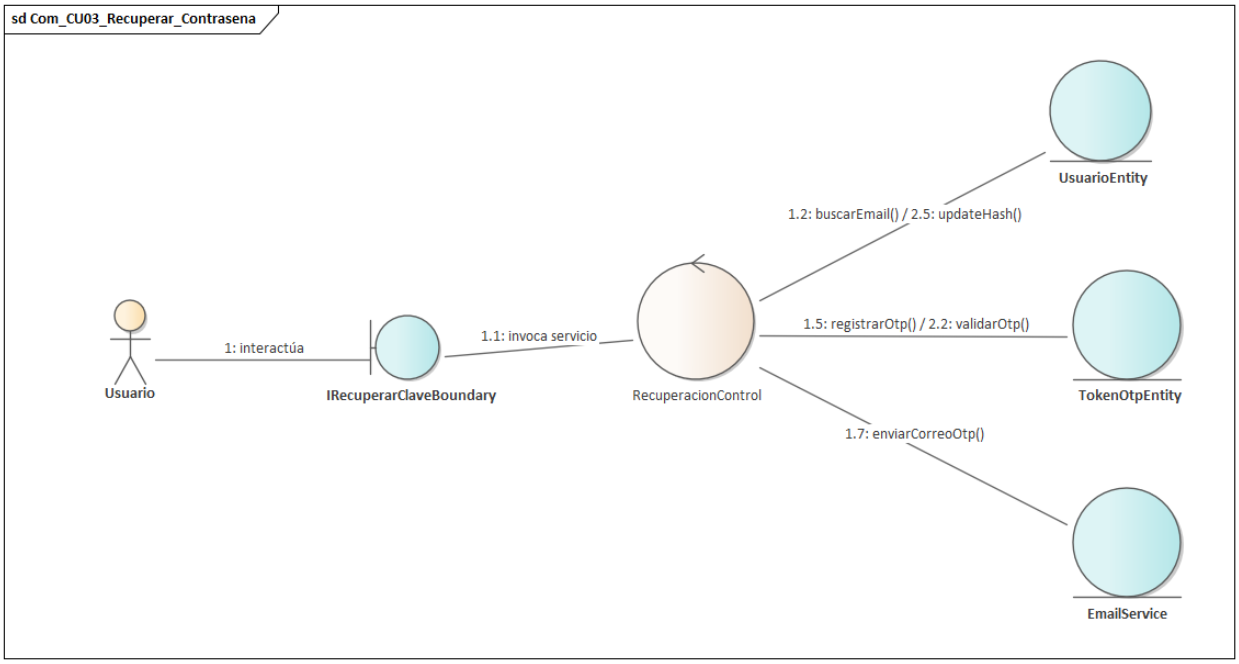
2.2.1 Diagrama de Comunicación - CU01: Autenticar Usuario (RBAC)



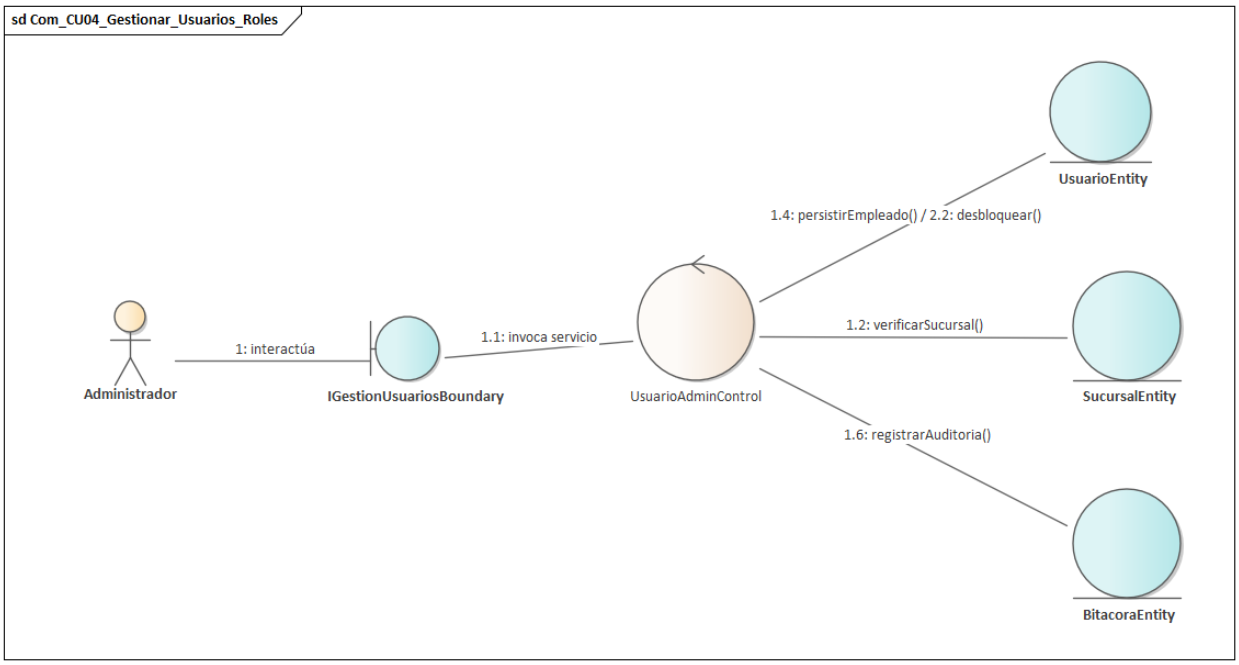
2.2.2 Diagrama de Comunicación - CU02: Registrar Cliente (Auto-registro)



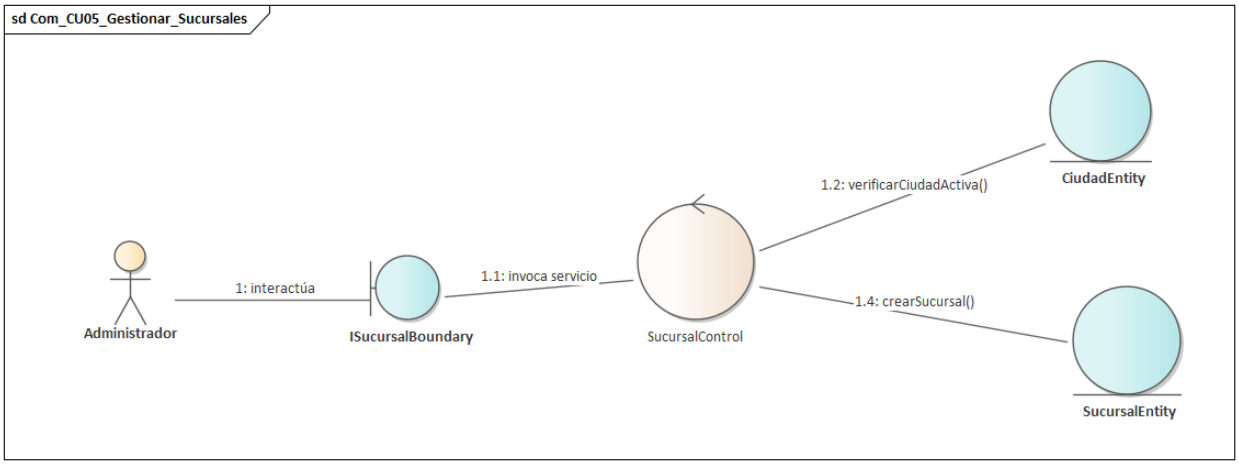
2.2.3 Diagrama de Comunicación - CU03: Recuperar Contraseña (Token OTP)



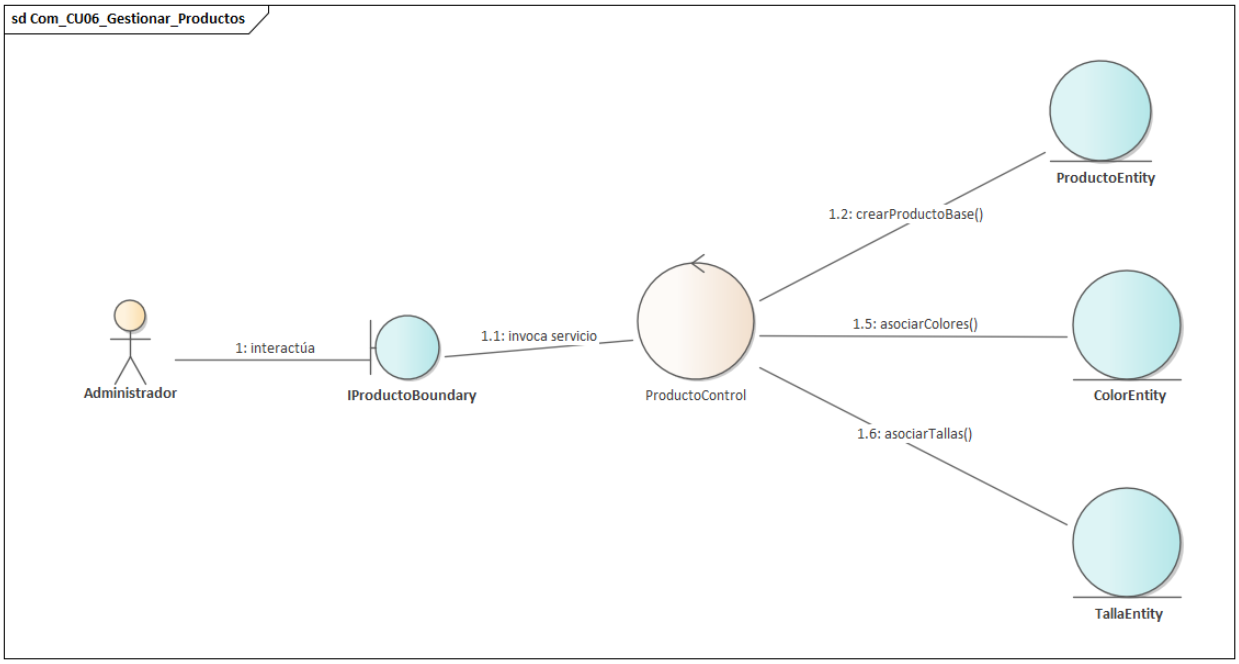
2.2.4 Diagrama de Comunicación - CU04: Gestionar Usuarios y Roles (RBAC)



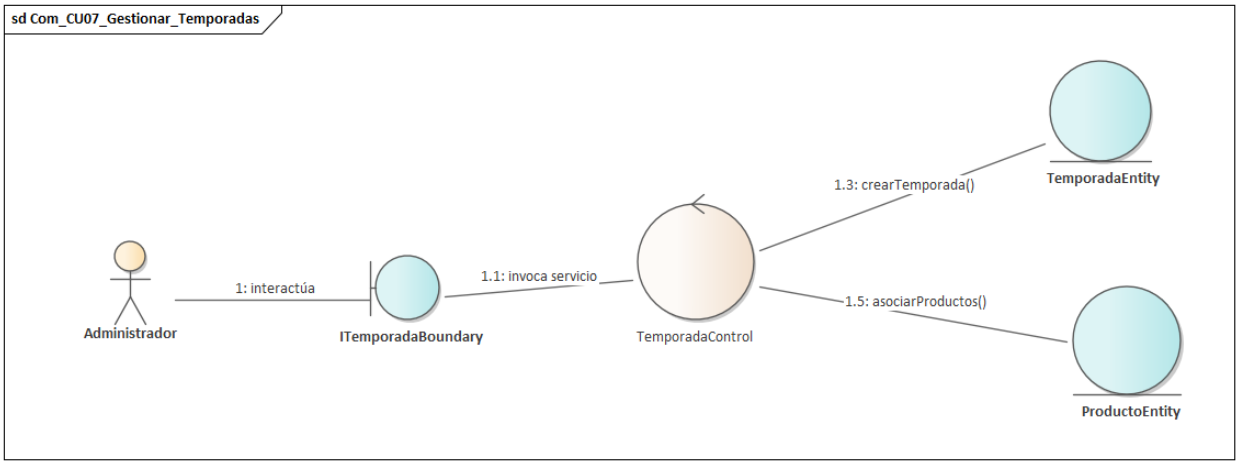
2.2.5 Diagrama de Comunicación - CU05: Gestionar Sucursales y Ciudades



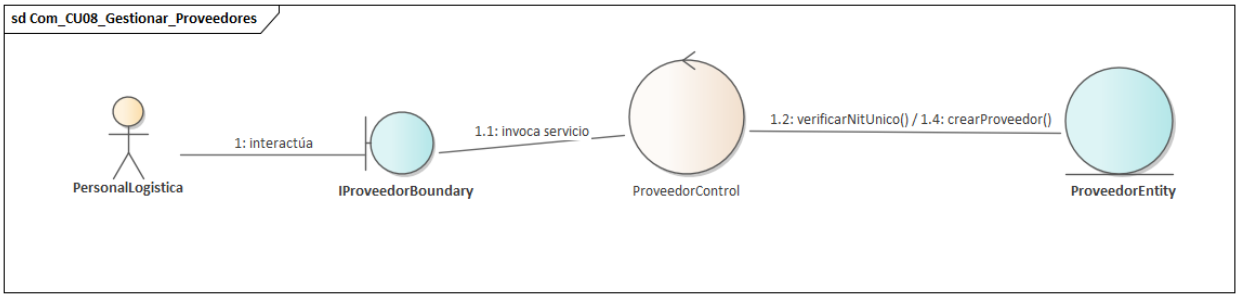
2.2.6 Diagrama de Comunicación - CU06: Gestionar Productos y Atributos



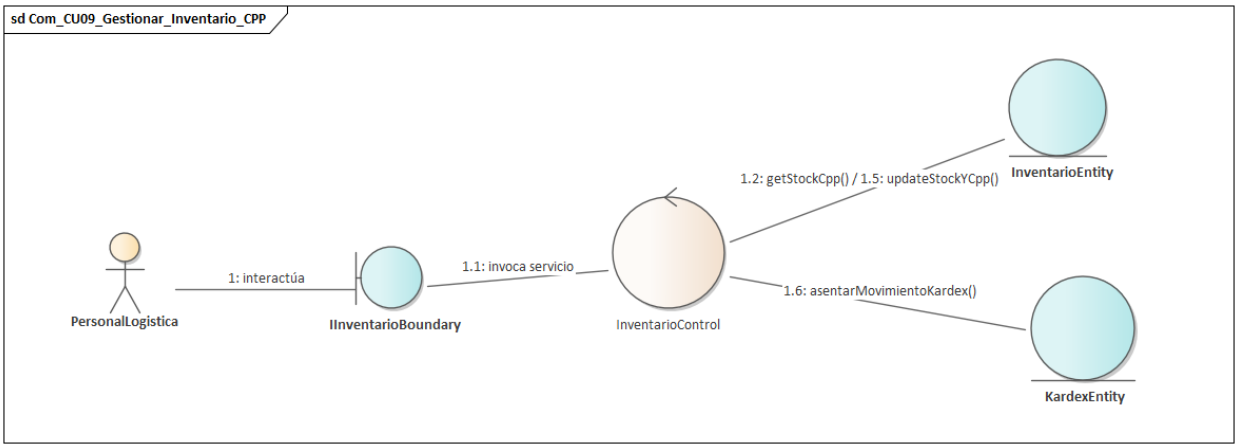
2.2.7 Diagrama de Comunicación - CU07: Gestionar Temporadas y Colecciones



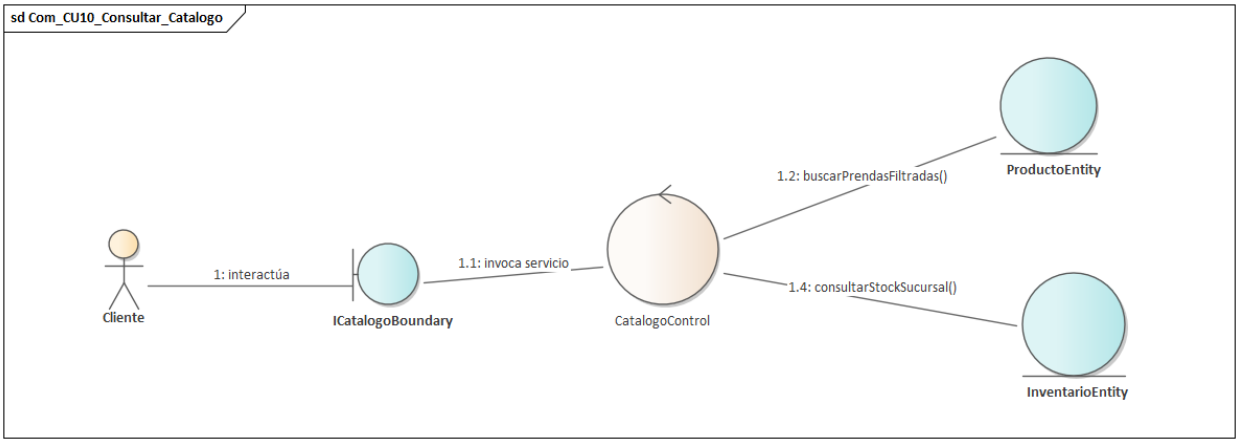
2.2.8 Diagrama de Comunicación - CU08: Gestionar Proveedores Textiles



2.2.9 Diagrama de Comunicación - CU09: Gestionar Inventario y Costos (CPP)



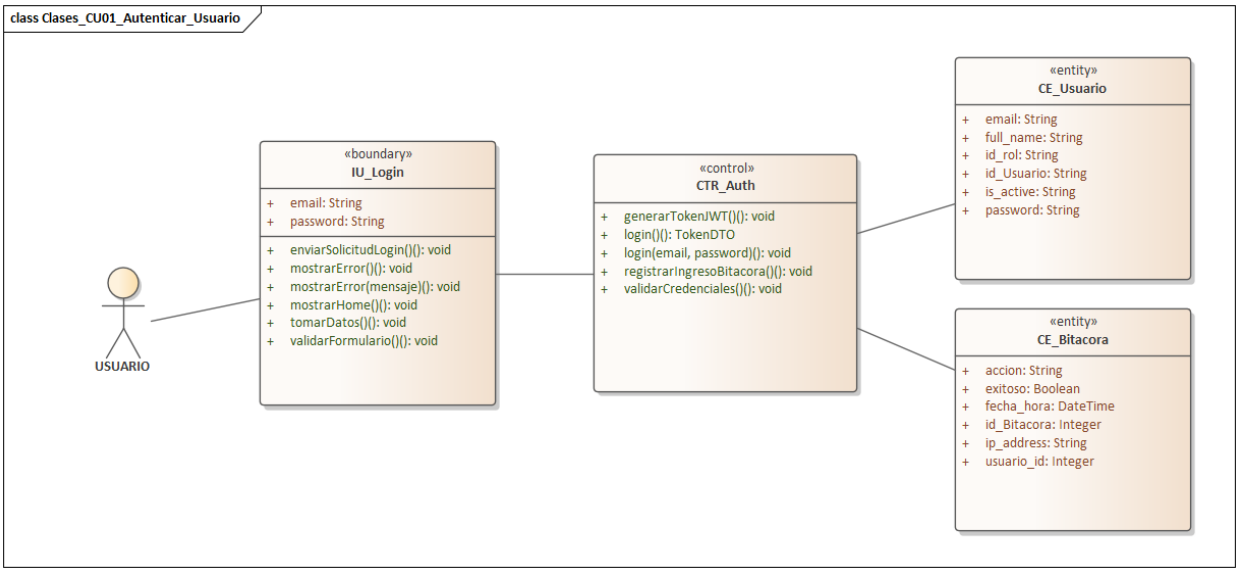
2.2.10 Diagrama de Comunicación - CU10: Consultar Catálogo y Disponibilidad

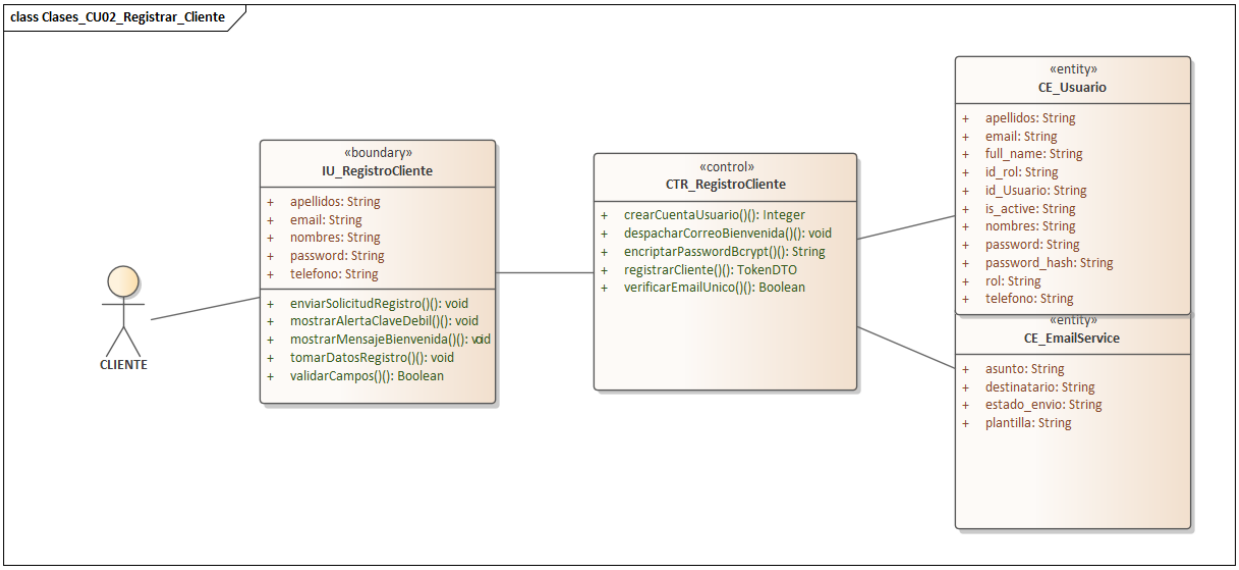


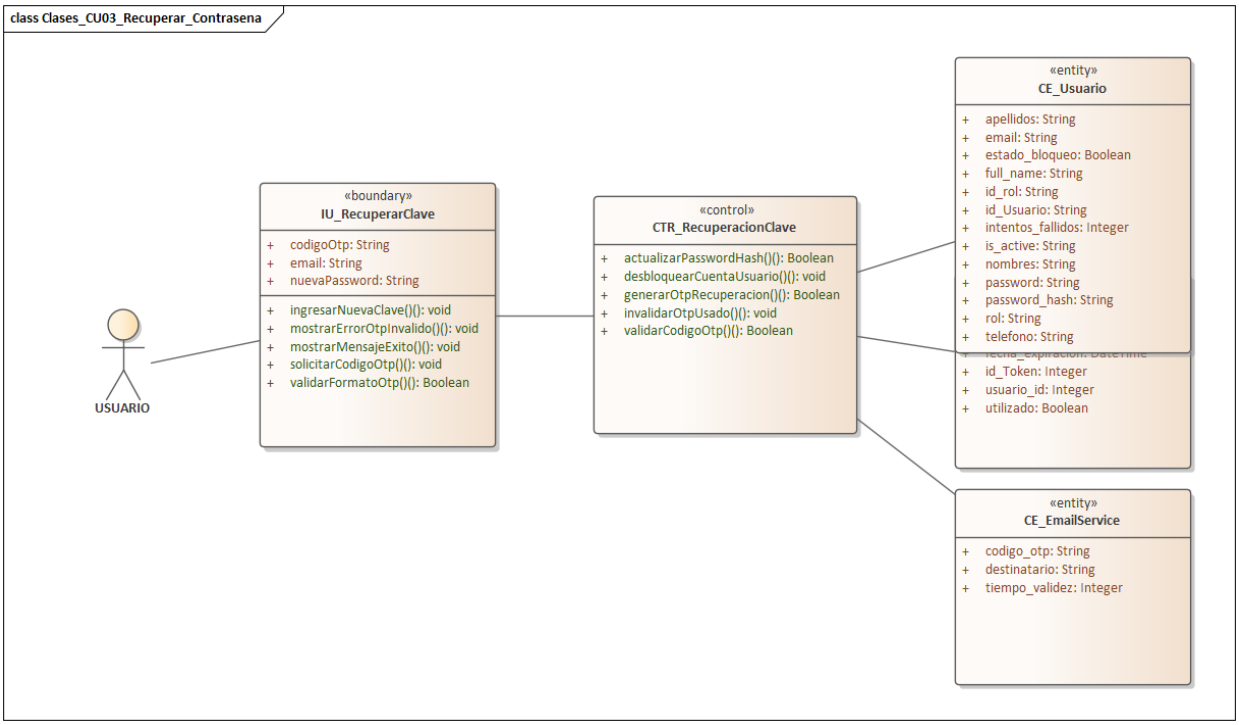
2.3 Análisis de Clases

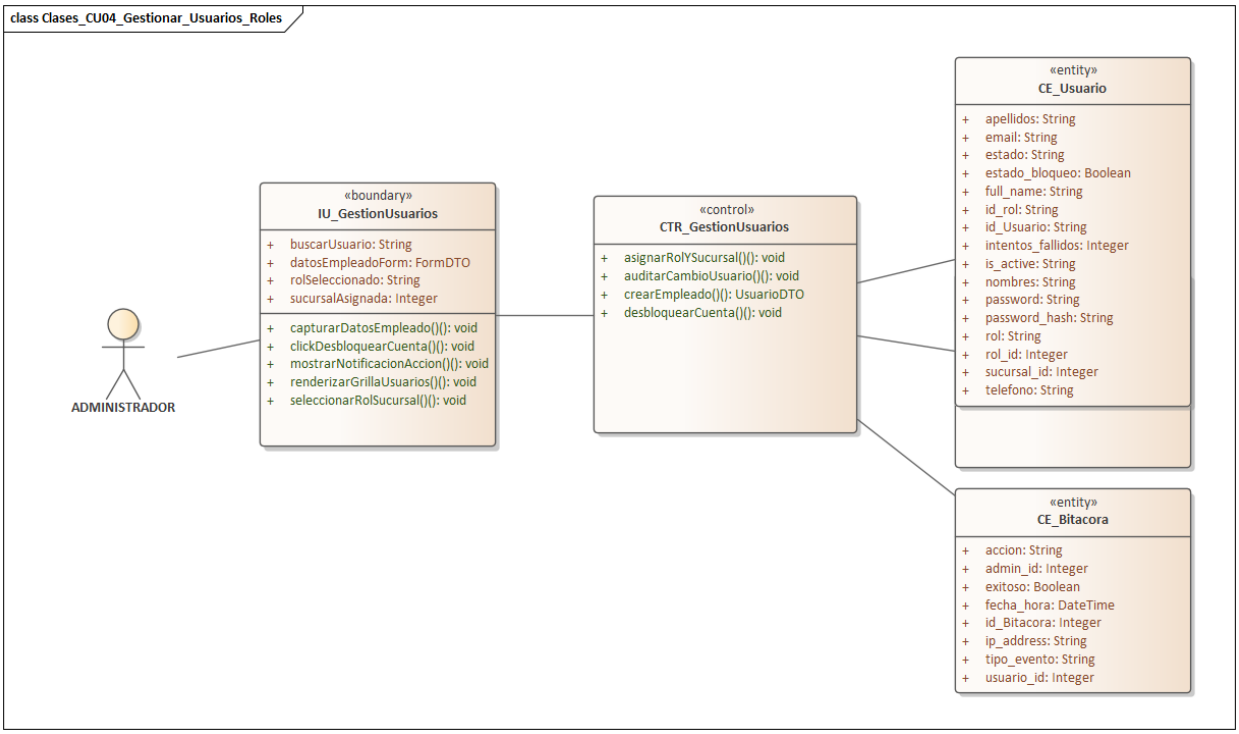
Conforme a la instrucción metodológica de la cátedra, el análisis de clases formaliza exhaustivamente las tres categorías de clases de robustez (BCE) con sus responsabilidades exactas para los **10 Casos de Uso del Ciclo 1**:

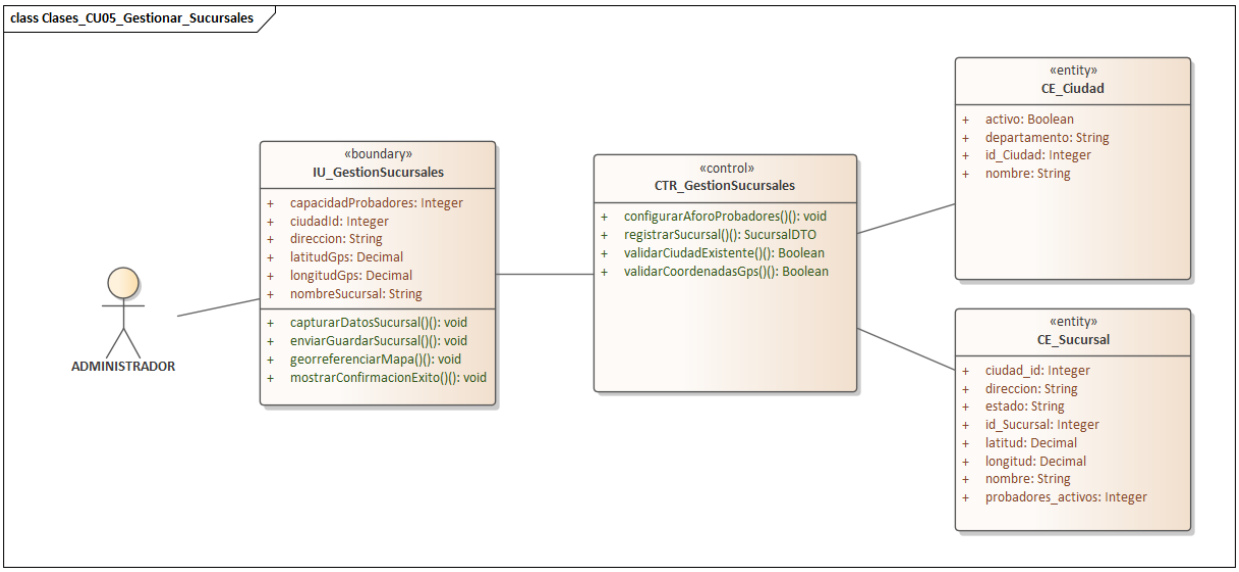
- **Clases de Interfaz (`<<boundary>>`)**: Poseen atributos de captura de datos visuales y métodos interactivos que reflejan las acciones y eventos del usuario (botones, selectores, clics).
- **Clases de Control (`<<control>>`)**: Contienen exclusivamente métodos de orquestación lógica del backend y **NO tienen atributos propios**.
- **Clases de Entidad (`<<entity>>`)**: Poseen tanto atributos de datos persistentes como métodos de encapsulamiento y consulta sobre la base de datos.

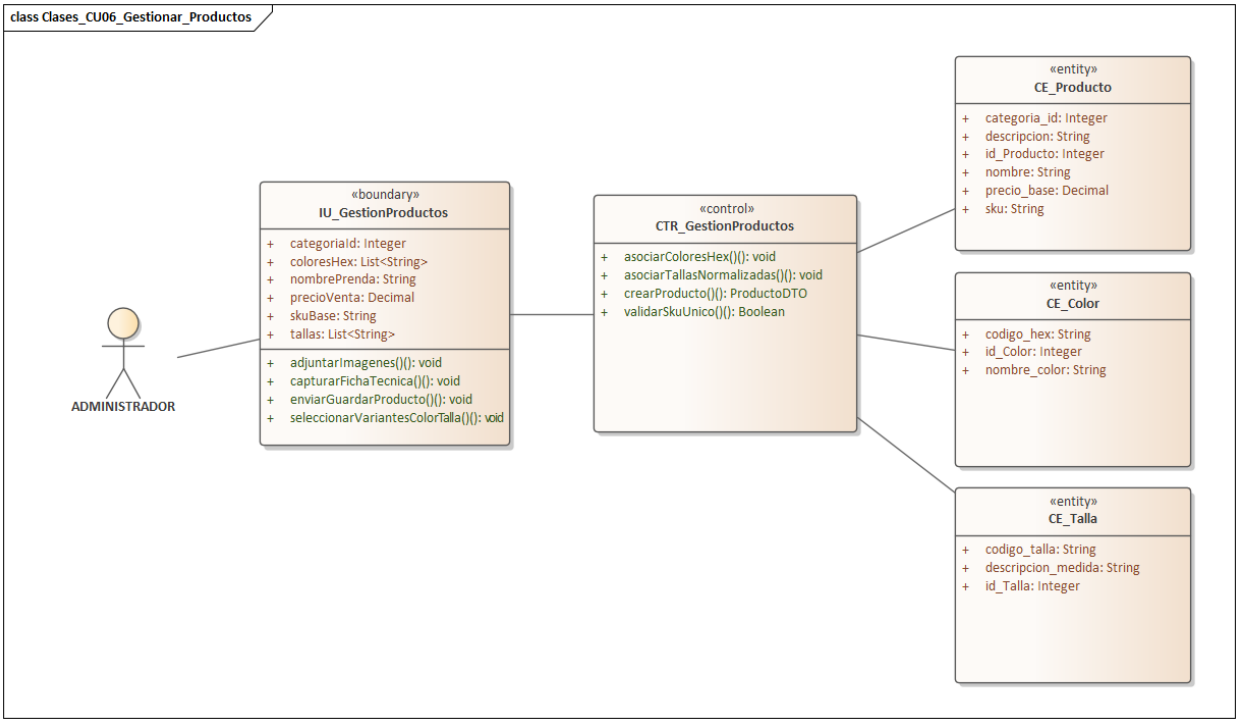

class Clases_CU01_Autenticar_Usuario
USUARIO
«boundary»
IU_Login
+ email: String
+ password: String
+ enviarSolicitudLogin()(): void
+ mostrarError()(): void
+ mostrarError(mensaje)(): void
+ mostrarHome()(): void
+ tomarDatos()(): void
+ validarFormulario()(): void
«control»
CTR_Auth
+ generarTokenJWT()(): void
+ login()(): TokenDTO
+ login(email, password)(): void
+ registrarIngresoBitacora()(): void
+ validarCredenciales()(): void
«entity»
CE_Usuario
+ email: String
+ full_name: String
+ id_rol: String
+ id_Usuario: String
+ is_active: String
+ password: String
«entity»
CE_Bitacora
+ accion: String
+ exitoso: Boolean
+ fecha_hora: DateTime
+ id_Bitacora: Integer
+ ip_address: String
+ usuario_id: Integer

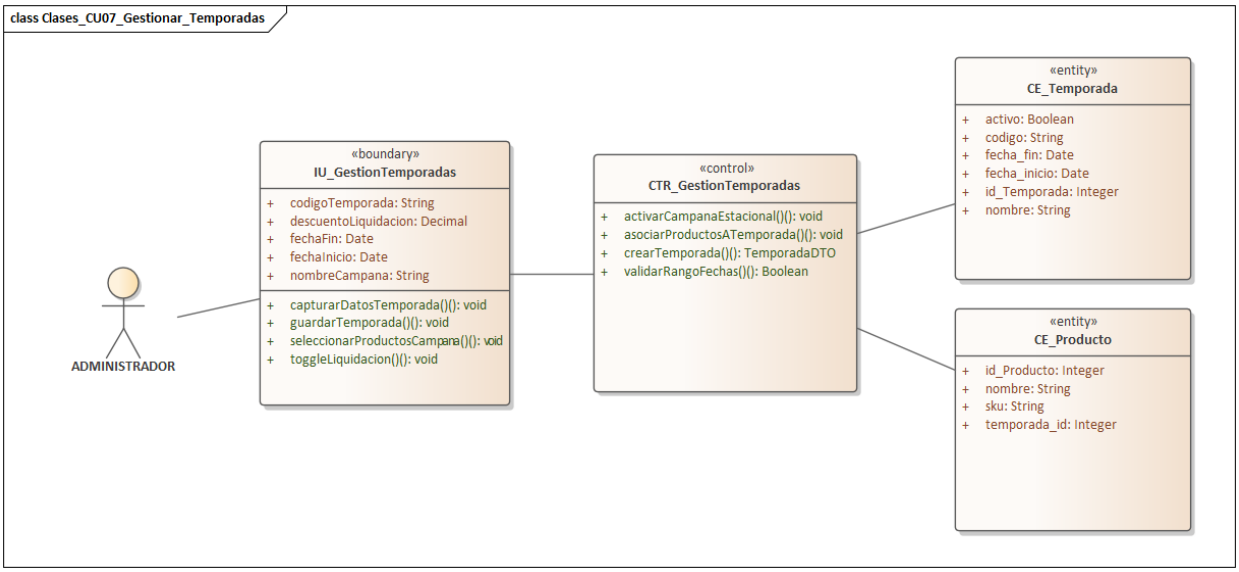

class Clases_CU02_Registrar_Cliente
CLIENTE
«boundary»
IU_RegistroCliente
+ apellidos: String
+ email: String
+ nombres: String
+ password: String
+ telefono: String
+ enviarSolicitudRegistro()(): void
+ mostrarAlertaClaveDebil()(): void
+ mostrarMensajeBienvenida()(): void
+ tomarDatosRegistro()(): void
+ validarCampos()(): Boolean
«control»
CTR_RegistroCliente
+ crearCuentaUsuario()(): Integer
+ despacharCorreoBienvenida()(): void
+ encriptarPasswordBcrypt()(): String
+ registrarCliente()(): TokenDTO
+ verificarEmailUnico()(): Boolean
«entity»
CE_Usuario
+ apellidos: String
+ email: String
+ full_name: String
+ id_rol: String
+ id_Usuario: String
+ is_active: String
+ nombres: String
+ password: String
+ password_hash: String
+ rol: String
+ telefono: String
«entity»
CE_EmailService
+ asunto: String
+ destinatario: String
+ estado_envio: String
+ plantilla: String

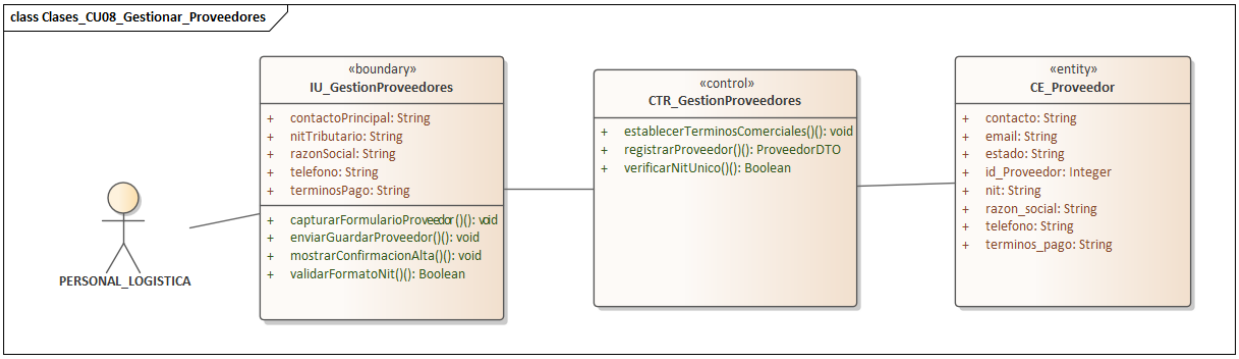

class Clases_CU03_Recuperar_Contrasena
USUARIO
«boundary»
IU_RecuperarClave
+ codigoOtp: String
+ email: String
+ nuevaPassword: String
+ ingresarNuevaClave()(): void
+ mostrarErrorOtpInvalido()(): void
+ mostrarMensajeExito()(): void
+ solicitarCodigoOtp()(): void
+ validarFormatoOtp()(): Boolean
«control»
CTR_RecuperacionClave
+ actualizarPasswordHash()(): Boolean
+ desbloquearCuentaUsuario()(): void
+ generarOtpRecuperacion()(): Boolean
+ invalidarOtpUsado()(): void
+ validarCodigoOtp()(): Boolean
«entity»
CE_Usuario
+ apellidos: String
+ email: String
+ estado_bloqueo: Boolean
+ full_name: String
+ id_rol: String
+ id_Usuario: String
+ intentos_fallidos: Integer
+ is_active: String
+ nombres: String
+ password: String
+ password_hash: String
+ rol: String
+ telefono: String
+ id_Token: Integer
+ usuario_id: Integer
+ utilizado: Boolean
«entity»
CE_EmailService
+ codigo_otp: String
+ destinatario: String
+ tiempo_validez: Integer

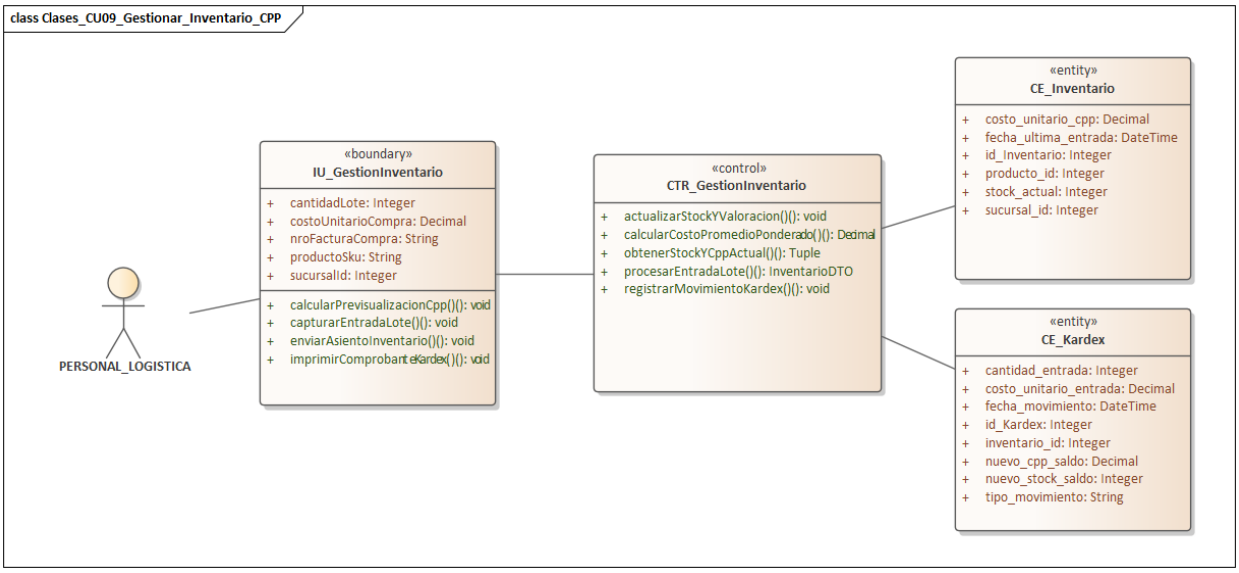

class Clases_CU04_Gestionar_Usuarios_Roles
ADMINISTRADOR
«boundary»
IU_GestionUsuarios
+ buscarUsuario: String
+ datosEmpleadoForm: FormDTO
+ rolSeleccionado: String
+ sucursalAsignada: Integer
+ capturarDatosEmpleado()(): void
+ clickDesbloquearCuenta()(): void
+ mostrarNotificacionAccion()(): void
+ renderizarGrillaUsuarios()(): void
+ seleccionarRolSucursal()(): void
«control»
CTR_GestionUsuarios
+ asignarRolYSucursal()(): void
+ auditarCambioUsuario()(): void
+ crearEmpleado()(): UsuarioDTO
+ desbloquearCuenta()(): void
«entity»
CE_Usuario
+ apellidos: String
+ email: String
+ estado: String
+ estado_bloqueo: Boolean
+ full_name: String
+ id_rol: String
+ id_Usuario: String
+ intentos_fallidos: Integer
+ is_active: String
+ nombres: String
+ password: String
+ password_hash: String
+ rol: String
+ rol_id: Integer
+ sucursal_id: Integer
+ telefono: String
«entity»
CE_Bitacora
+ accion: String
+ admin_id: Integer
+ exitoso: Boolean
+ fecha_hora: DateTime
+ id_Bitacora: Integer
+ ip_address: String
+ tipo_evento: String
+ usuario_id: Integer

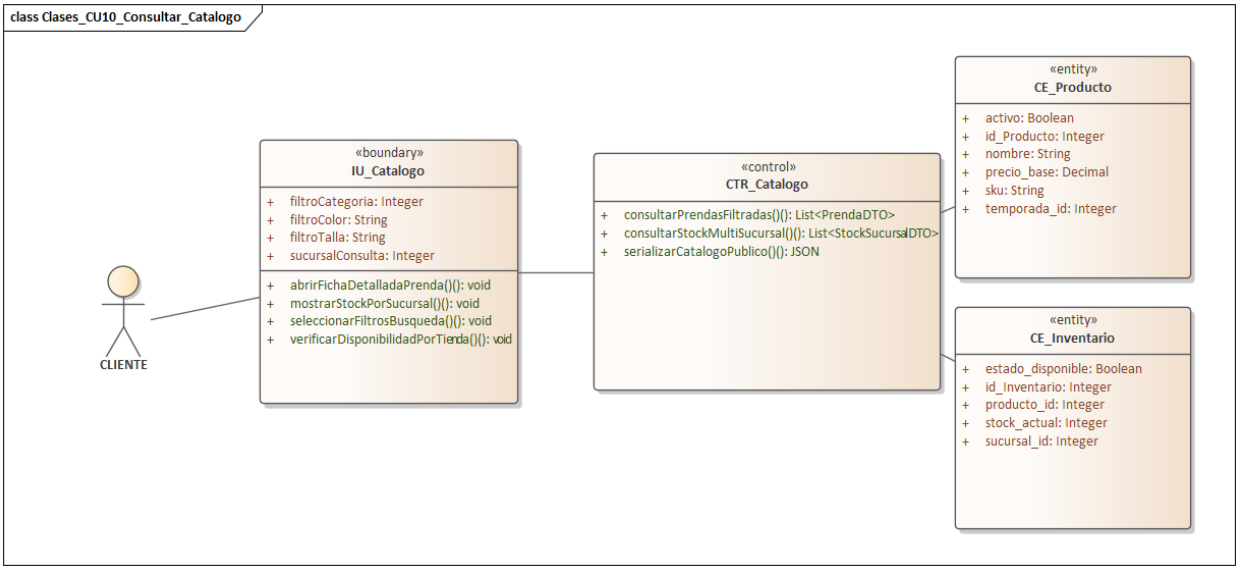

class Clases_CU05_Gestionar_Sucursales
ADMINISTRADOR
«boundary»
IU_GestionSucursales
+ capacidadProbadores: Integer
+ ciudadId: Integer
+ direccion: String
+ latitudGps: Decimal
+ longitudGps: Decimal
+ nombreSucursal: String
+ capturarDatosSucursal()(): void
+ enviarGuardarSucursal()(): void
+ georreferenciarMapa()(): void
+ mostrarConfirmacionExito()(): void
«control»
CTR_GestionSucursales
+ configurarAforoProbadores()(): void
+ registrarSucursal()(): SucursalDTO
+ validarCiudadExistente()(): Boolean
+ validarCoordenadasGps()(): Boolean
«entity»
CE_Ciudad
+ activo: Boolean
+ departamento: String
+ id_Ciudad: Integer
+ nombre: String
«entity»
CE_Sucursal
+ ciudad_id: Integer
+ direccion: String
+ estado: String
+ id_Sucursal: Integer
+ latitud: Decimal
+ longitud: Decimal
+ nombre: String
+ probadores_activos: Integer


class Clases_CU06_Gestionar_Productos
ADMINISTRADOR
«boundary»
IU_GestionProductos
+ categoriaId: Integer
+ coloresHex: List<String>
+ nombrePrenda: String
+ precioVenta: Decimal
+ skuBase: String
+ tallas: List<String>
+ adjuntarImagenes()(): void
+ capturarFichaTecnica()(): void
+ enviarGuardarProducto()(): void
+ seleccionarVariantesColorTalla()(): void
«control»
CTR_GestionProductos
+ asociarColoresHex()(): void
+ asociarTallasNormalizadas()(): void
+ crearProducto()(): ProductoDTO
+ validarSkuUnico()(): Boolean
«entity»
CE_Producto
+ categoria_id: Integer
+ descripcion: String
+ id_Producto: Integer
+ nombre: String
+ precio_base: Decimal
+ sku: String
«entity»
CE_Color
+ codigo_hex: String
+ id_Color: Integer
+ nombre_color: String
«entity»
CE_Talla
+ codigo_talla: String
+ descripcion_medida: String
+ id_Talla: Integer


class Clases_CU07_Gestionar_Temporadas
ADMINISTRADOR
«boundary»
IU_GestionTemporadas
+ codigoTemporada: String
+ descuentoLiquidacion: Decimal
+ fechaFin: Date
+ fechaInicio: Date
+ nombreCampana: String
+ capturarDatosTemporada()(): void
+ guardarTemporada()(): void
+ seleccionarProductosCampana()(): void
+ toggleLiquidacion()(): void
«control»
CTR_GestionTemporadas
+ activarCampanaEstacional()(): void
+ asociarProductosATemporada()(): void
+ crearTemporada()(): TemporadaDTO
+ validarRangoFechas()(): Boolean
«entity»
CE_Temporada
+ activo: Boolean
+ codigo: String
+ fecha_fin: Date
+ fecha_inicio: Date
+ id_Temporada: Integer
+ nombre: String
«entity»
CE_Producto
+ id_Producto: Integer
+ nombre: String
+ sku: String
+ temporada_id: Integer


class Clases_CU08_Gestionar_Proveedores
PERSONAL_LOGISTICA
«boundary»
IU_GestionProveedores
+ contactoPrincipal: String
+ nitTributario: String
+ razonSocial: String
+ telefono: String
+ terminosPago: String
+ capturarFormularioProveedor ()(): void
+ enviarGuardarProveedor()(): void
+ mostrarConfirmacionAlta()(): void
+ validarFormatoNit()(): Boolean
«control»
CTR_GestionProveedores
+ establecerTerminosComerciales()(): void
+ registrarProveedor()(): ProveedorDTO
+ verificarNitUnico()(): Boolean
«entity»
CE_Proveedor
+ contacto: String
+ email: String
+ estado: String
+ id_Proveedor: Integer
+ nit: String
+ razon_social: String
+ telefono: String
+ terminos_pago: String

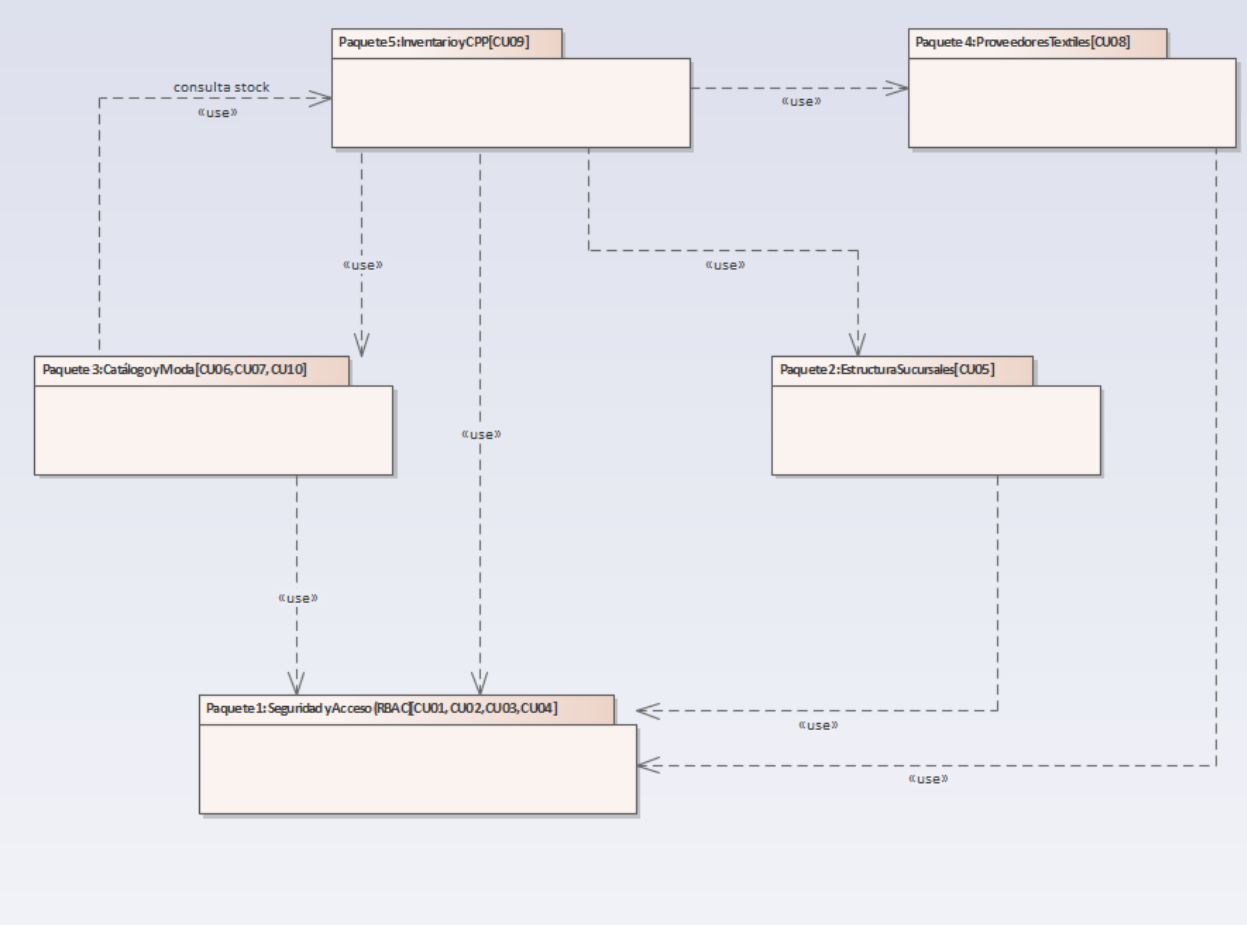




2.4 Análisis de Paquetes

En el análisis de paquetes se evalúa la arquitectura bajo dos métricas cardinales de la ingeniería de software:

1. **Acoplamiento**: Medida de interdependencia entre los módulos. Se busca un **bajo acoplamiento** para evitar que modificaciones en un subsistema provoquen fallos en cascada.
2. **Cohesión**: Medida de afinidad y fortaleza asociativa interna de los elementos de un paquete. Se procura una **alta cohesión funcional**, agrupando en cada paquete clases estrictamente orientadas a un mismo objetivo de negocio.



Paquete de Análisis	Nivel de Cohesión	Nivel de Acoplamiento	Justificación Técnica de Diseño
Seguridad (RBAC)	**Alta (Funcional)**	**Bajo (Eferente: 0, Aferente: 4)**	Módulo núcleo completamente autónomo; provee tokens y validación de identidad a todos los demás paquetes sin depender de ellos.
Sucursales	**Alta (Datos/Operación)**	**Bajo (Eferente: 1, Aferente: 1)**	Gestiona la infraestructura física; solo depende de la seguridad para autorizaciones de edición.
Productos y Catálogo	**Alta (Dominio)**	**Medio (Eferente: 2, Aferente: 1)**	Agrupa la lógica de prendas, tallas, colores y temporadas; interactúa con inventario para disponibilizar existencias en el catálogo.
Inventario y Costos	**Muy Alta (Transaccional)**	**Controlado (Eferente: 4, Aferente: 1)**	Núcleo integrador logístico y financiero; conecta sucursales, productos y proveedores para la actualización del Kardex y CPP.
Proveedores	**Alta (Dominio)**	**Bajo (Eferente: 1, Aferente: 1)**	Maneja exclusivamente el padrón comercial y contratos de suministro.

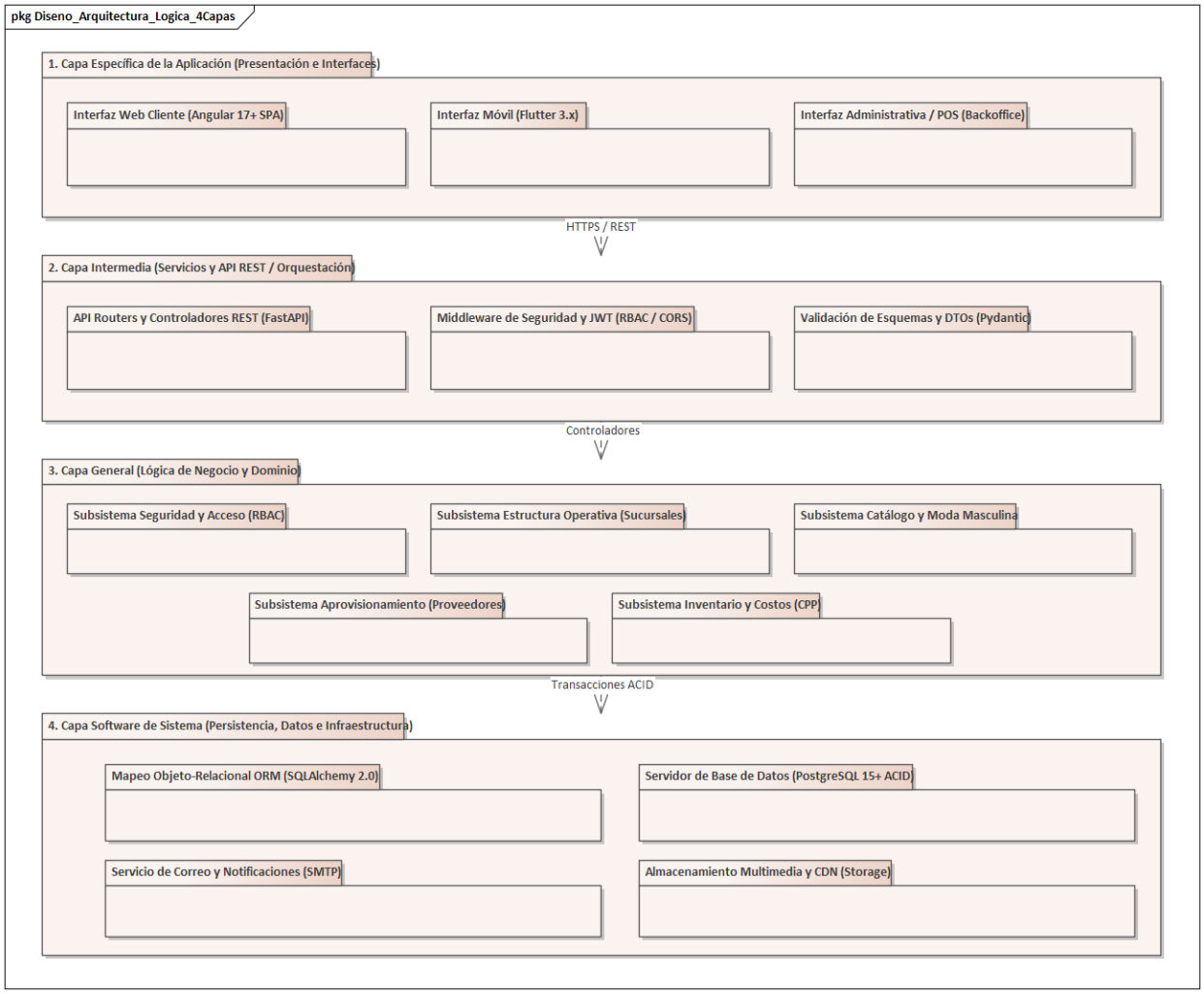
. . . . . .

3. Flujo de Trabajo: Diseño

3.1 Diseño de Arquitectura

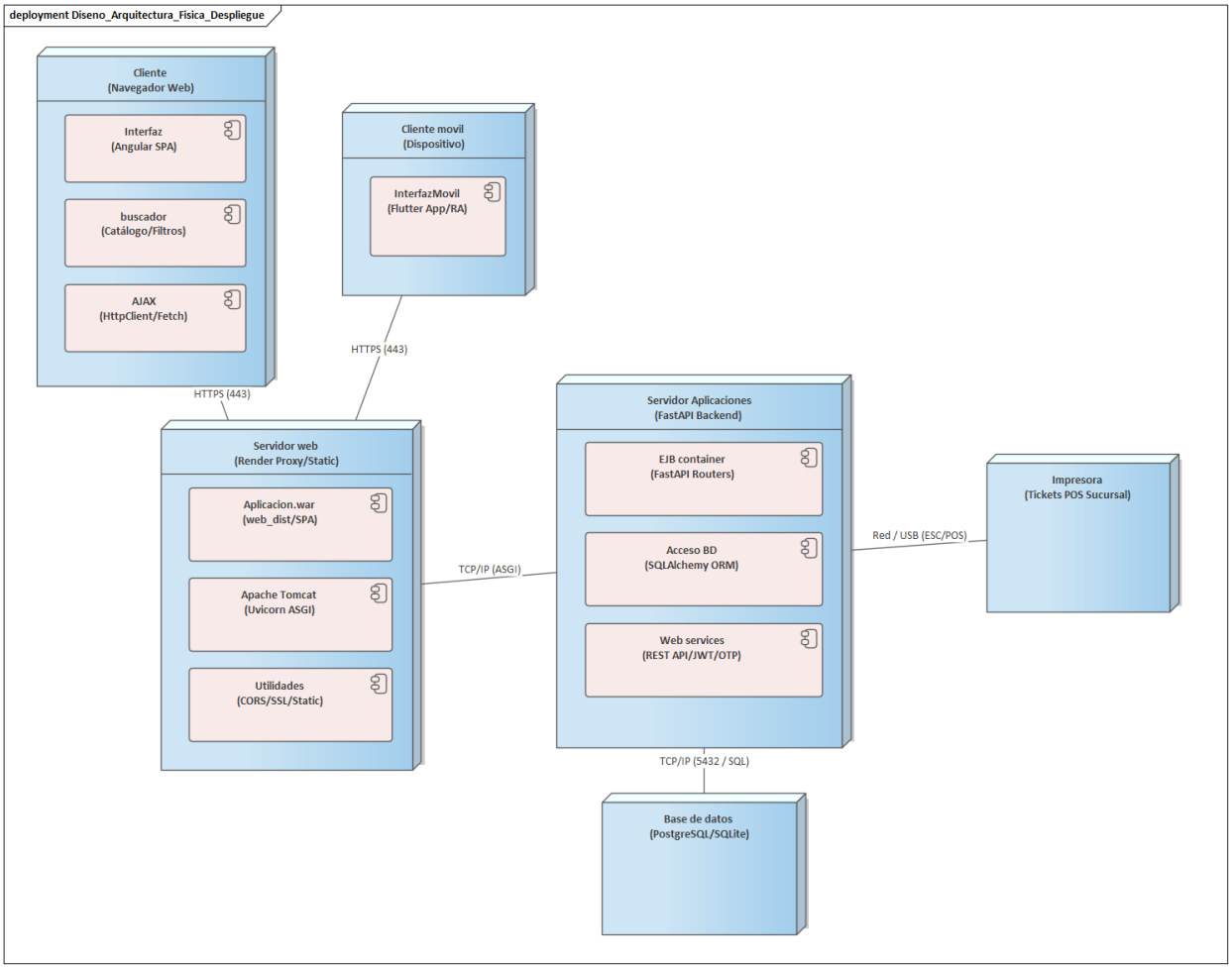
3.1.1 Diseño Lógico de la Arquitectura (4 Capas UML)

Conforme a las explicaciones de cátedra, la arquitectura lógica del sistema FashionStore se estructura formalmente en **cuatro capas horizontales desacopladas**:



3.1.2 Diseño Físico de la Arquitectura (Diagrama de Despliegue)

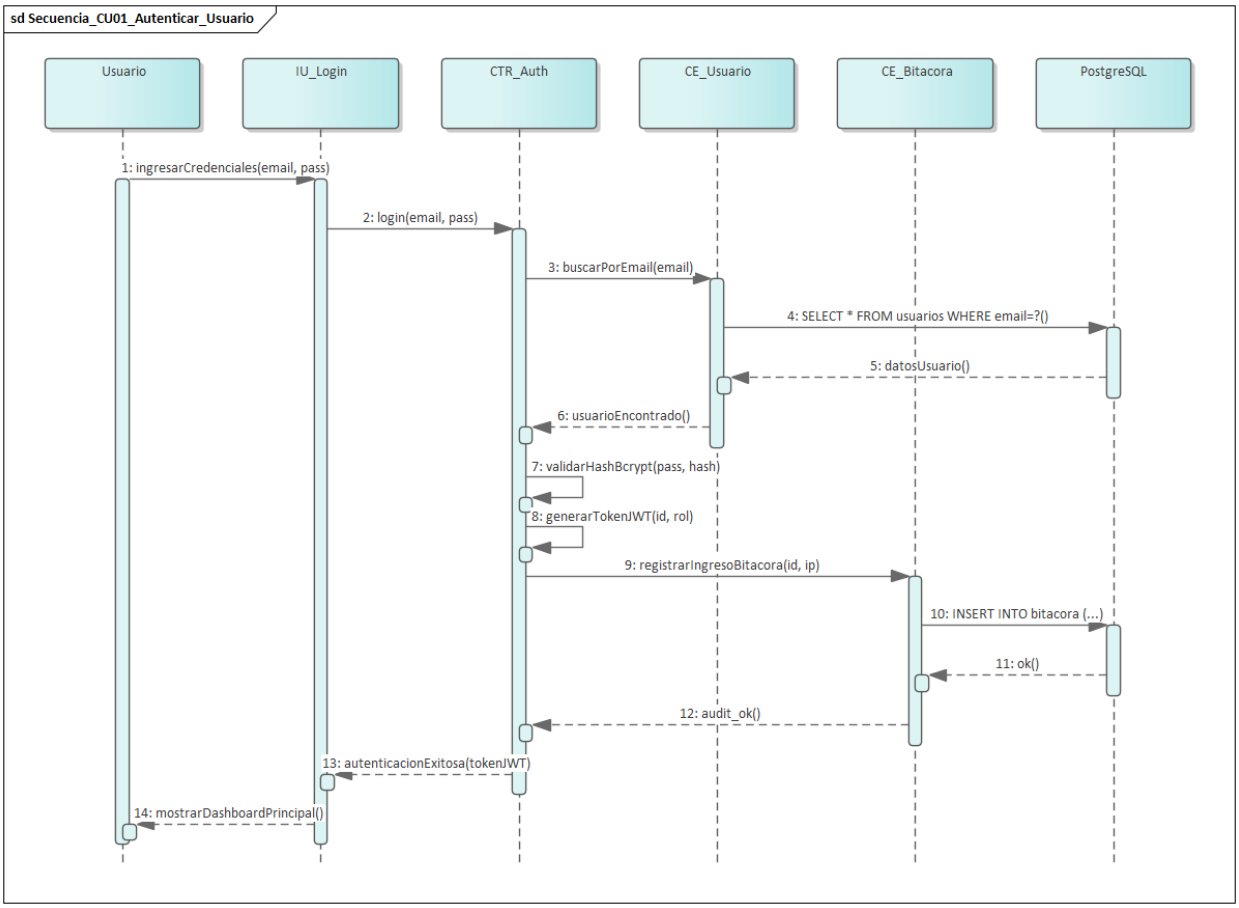
El diagrama de despliegue modela los nodos físicos y de nube donde residen los artefactos de software compilados, los enlaces telemáticos y protocolos de comunicación:



3.2 Diseño de Casos de Uso

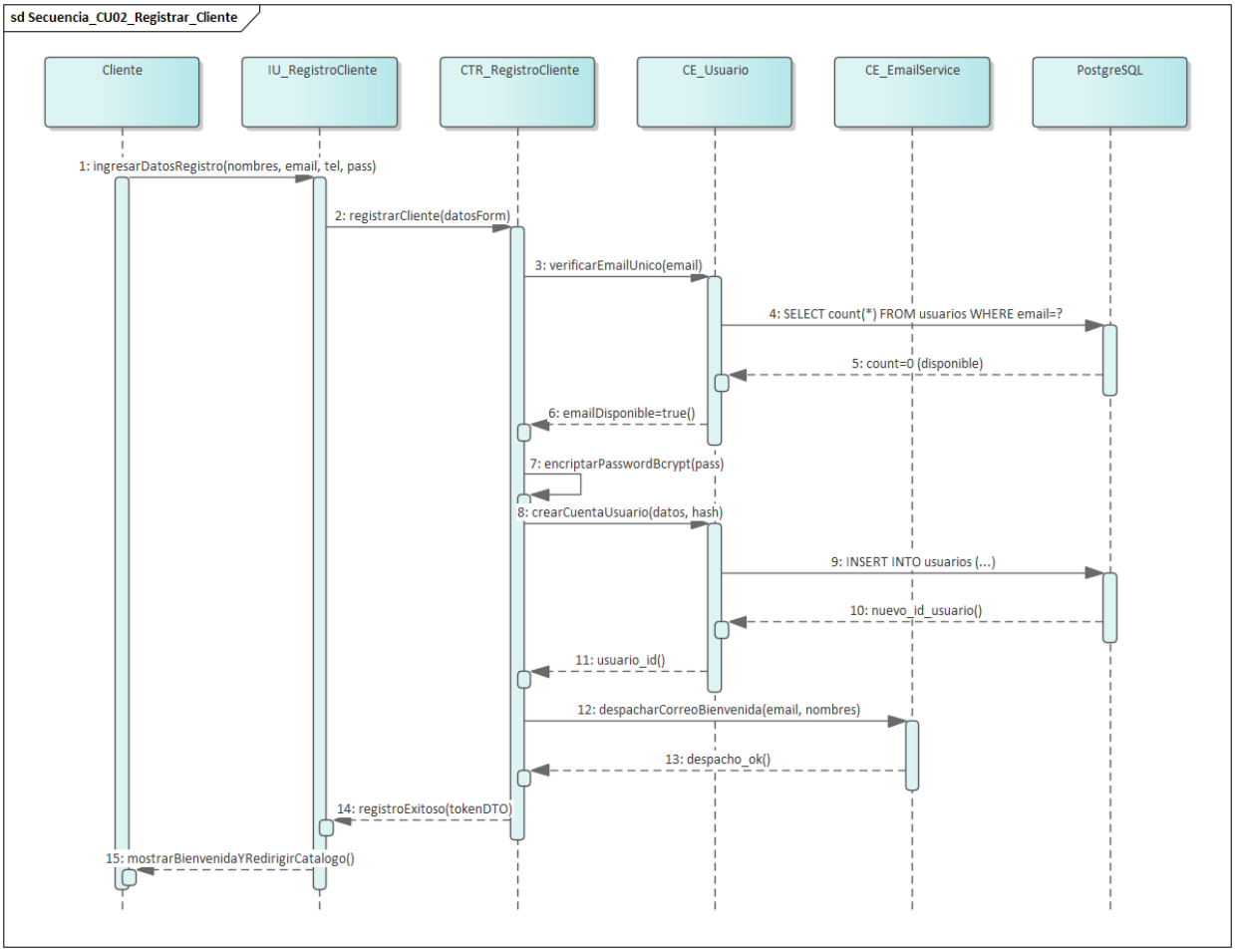
3.2.1 Diagrama de Secuencia - CU01: Autenticar Usuario (Login RBAC)

Modela el intercambio de credenciales, consulta de usuario, validación de hash Bcrypt, generación del token JWT y registro de auditoría en bitácora.



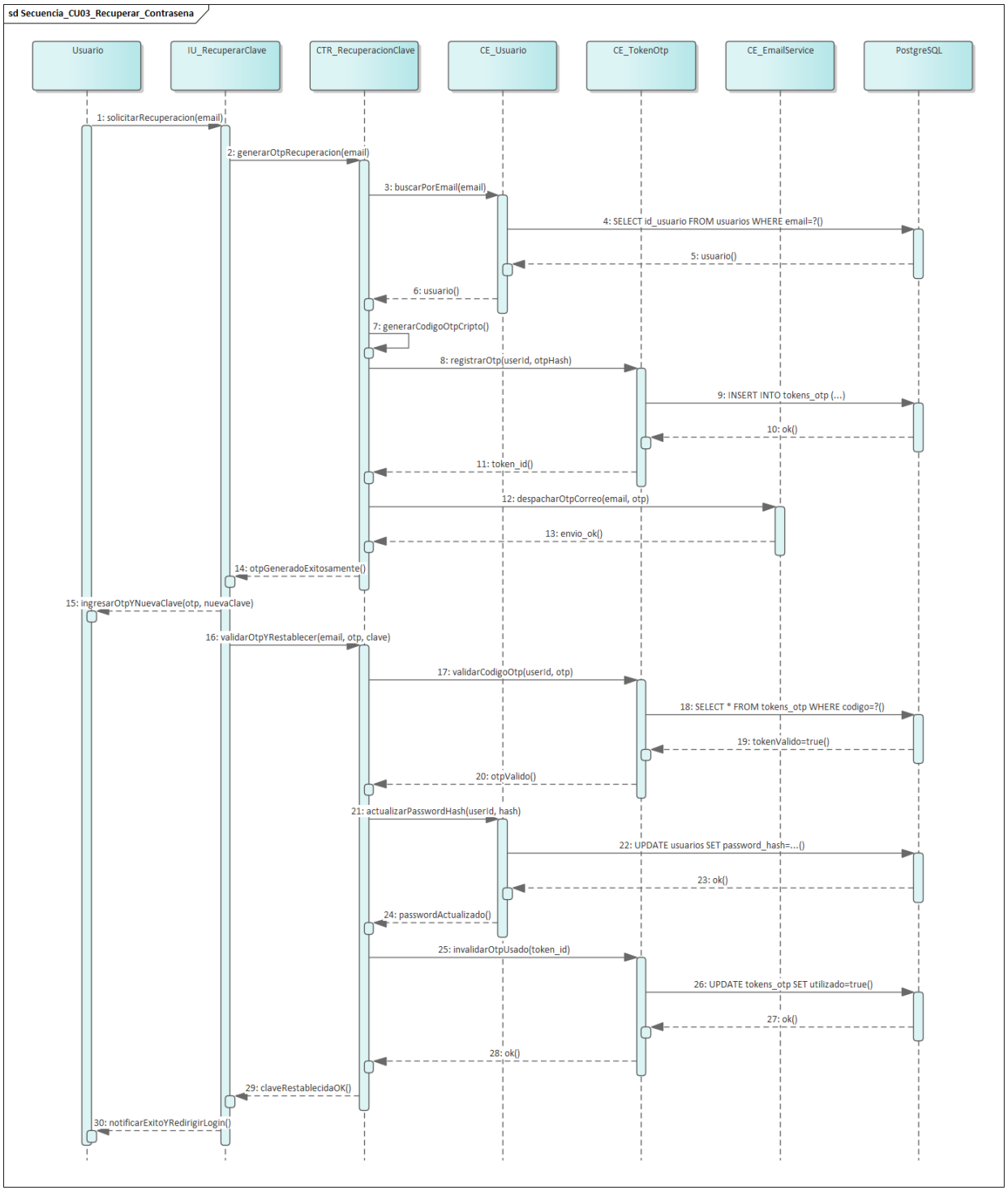
3.2.2 Diagrama de Secuencia - CU02: Registrar Cliente (Sign Up)

Modela el autoregistro público de nuevos clientes, validación de correo electrónico único, hashing seguro de contraseña y despacho de correo de bienvenida.



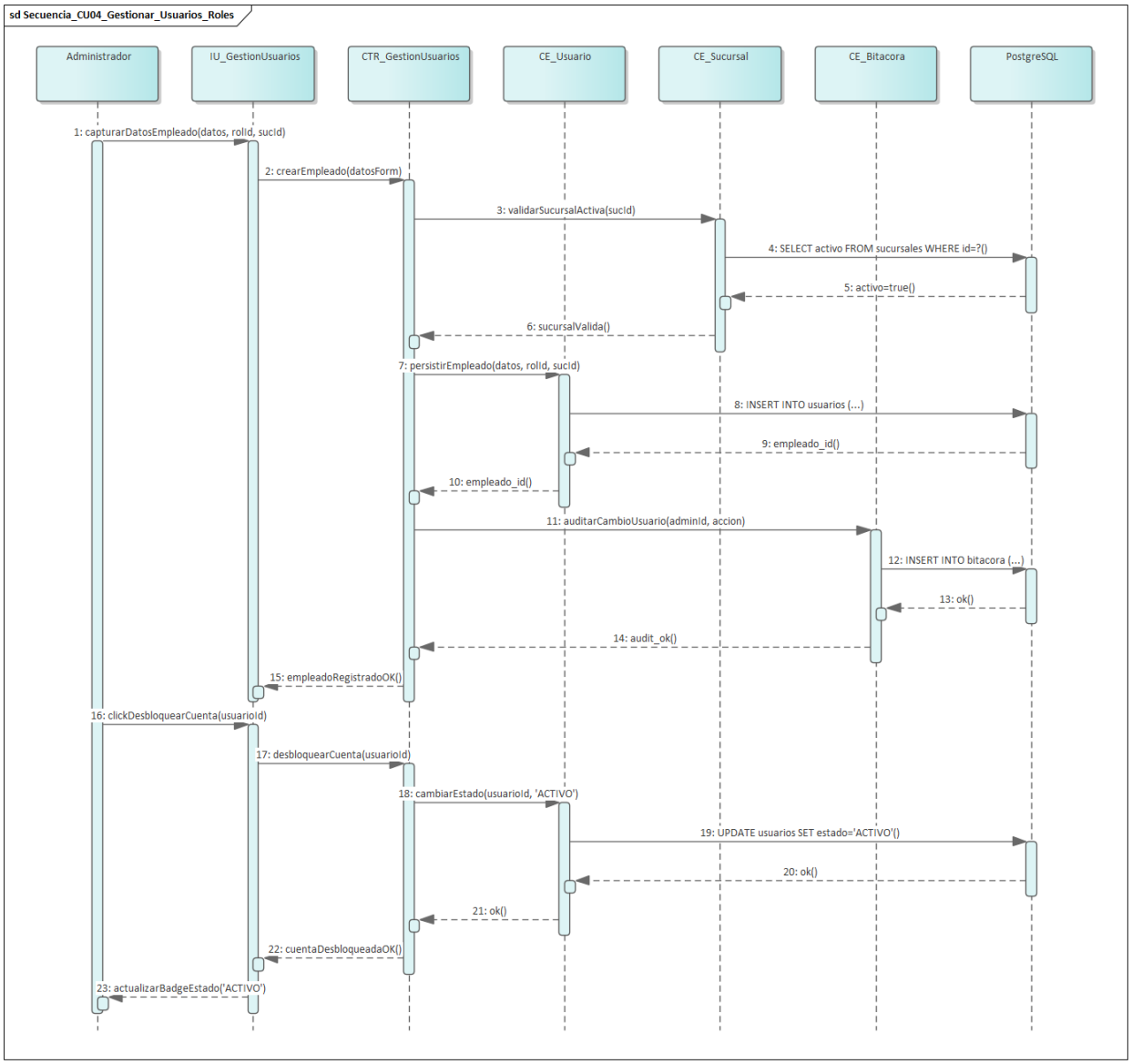
3.2.3 Diagrama de Secuencia - CU03: Recuperar Contraseña (Token OTP)

Modela la solicitud autoservicio de código temporal OTP criptográfico de 6 dígitos, envío seguro por correo electrónico y restablecimiento de contraseña.



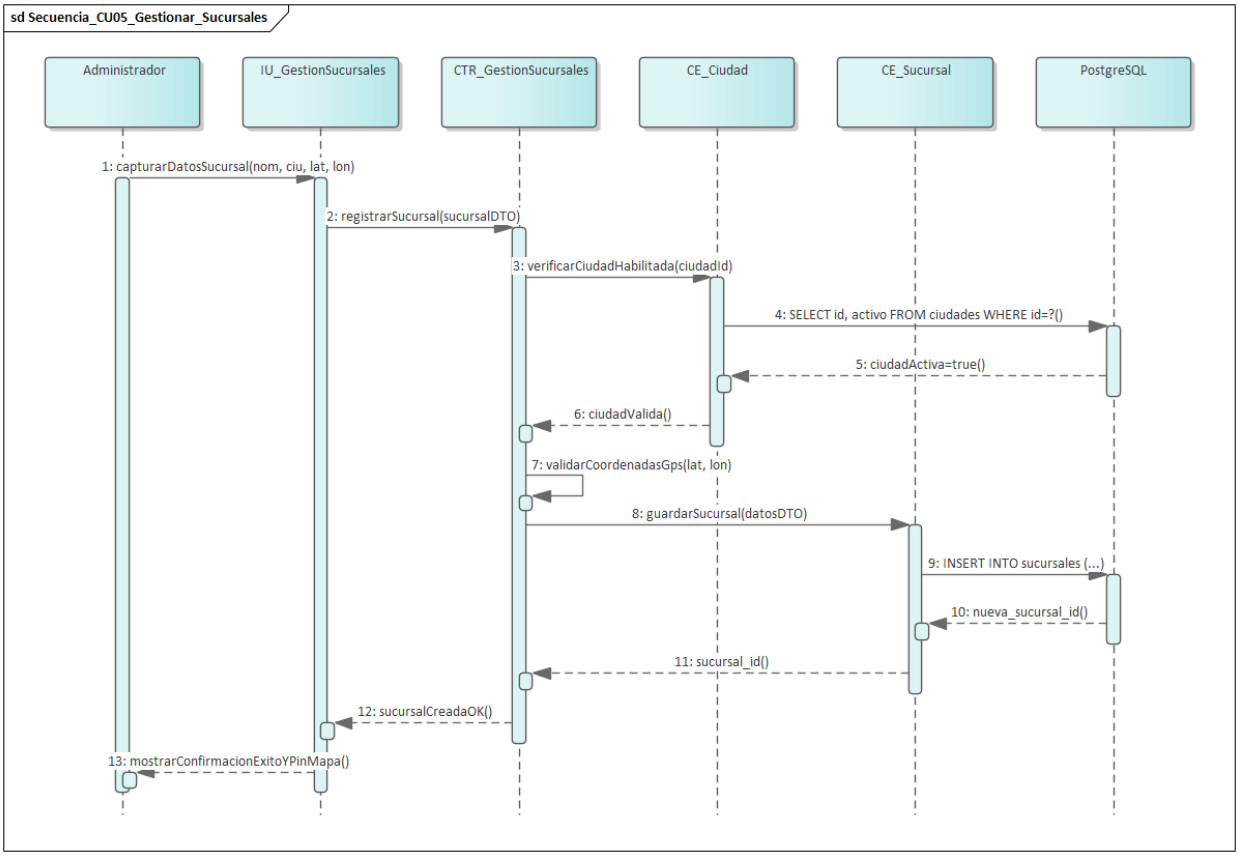
3.2.4 Diagrama de Secuencia - CU04: Gestionar Usuarios y Roles (RBAC)

Modela la administración de cuentas por el administrador, asignación de roles de seguridad RBAC, validación de sucursal asignada y desbloqueo de cuentas.



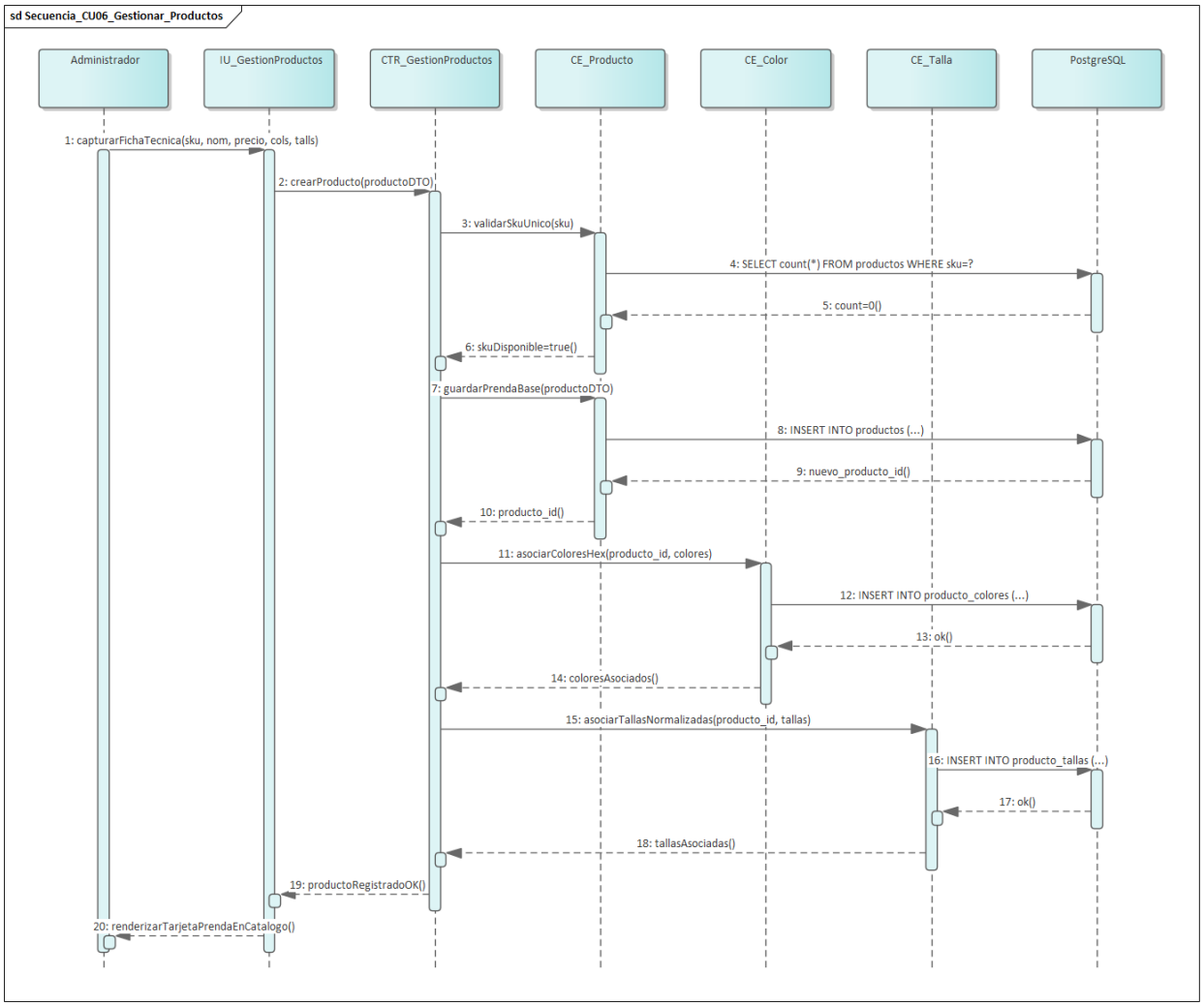
3.2.5 Diagrama de Secuencia - CU05: Gestionar Sucursales y Ciudades

Modela la parametrización de infraestructura geográfica, validación de coordenadas GPS satelitales y configuración del aforo físico para probadores de ropa.



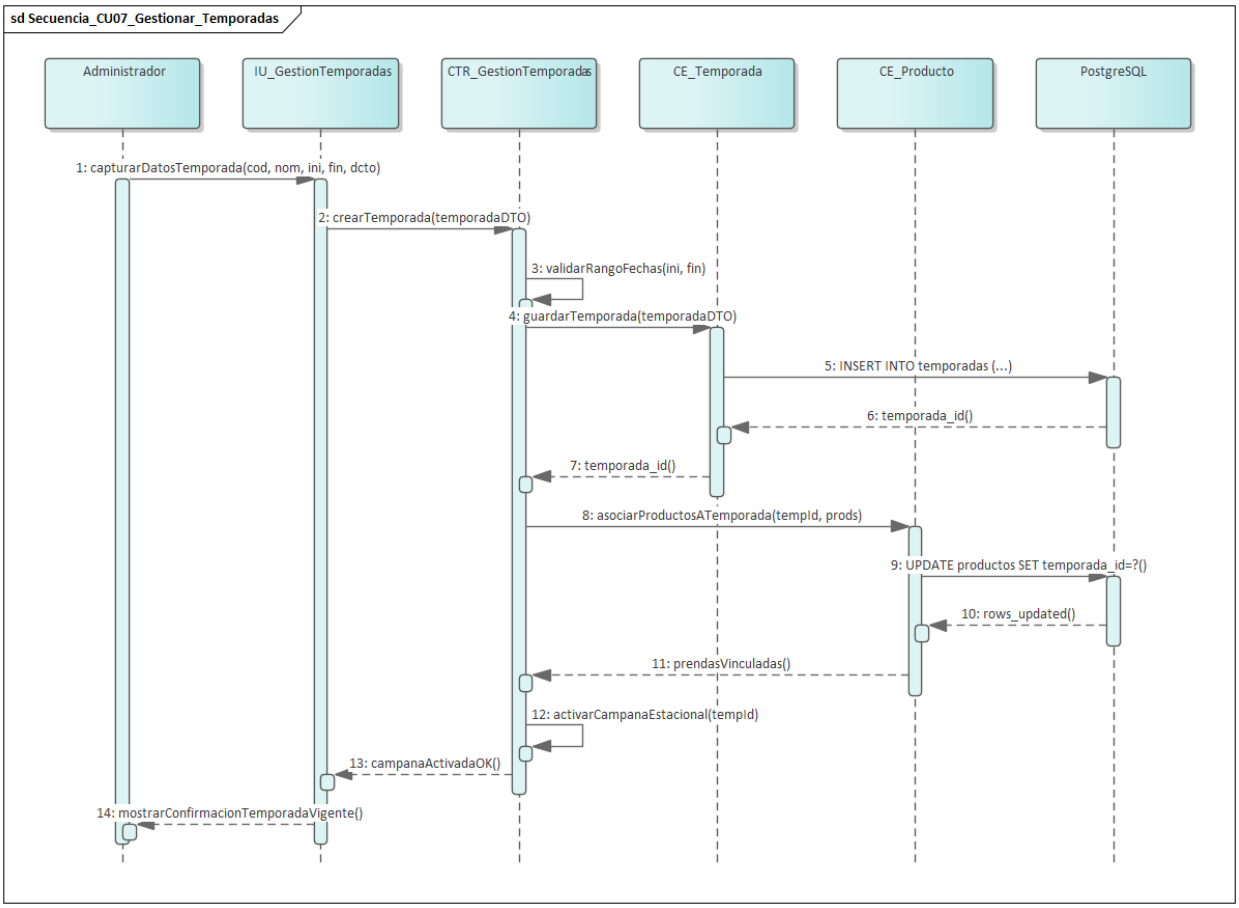
3.2.6 Diagrama de Secuencia - CU06: Gestionar Productos y Atributos de Moda

Modela el alta de prendas de vestir masculinas, validación de unicidad de código SKU base y asociación multivaluada normalizada de variantes de color (HEX) y tallas.



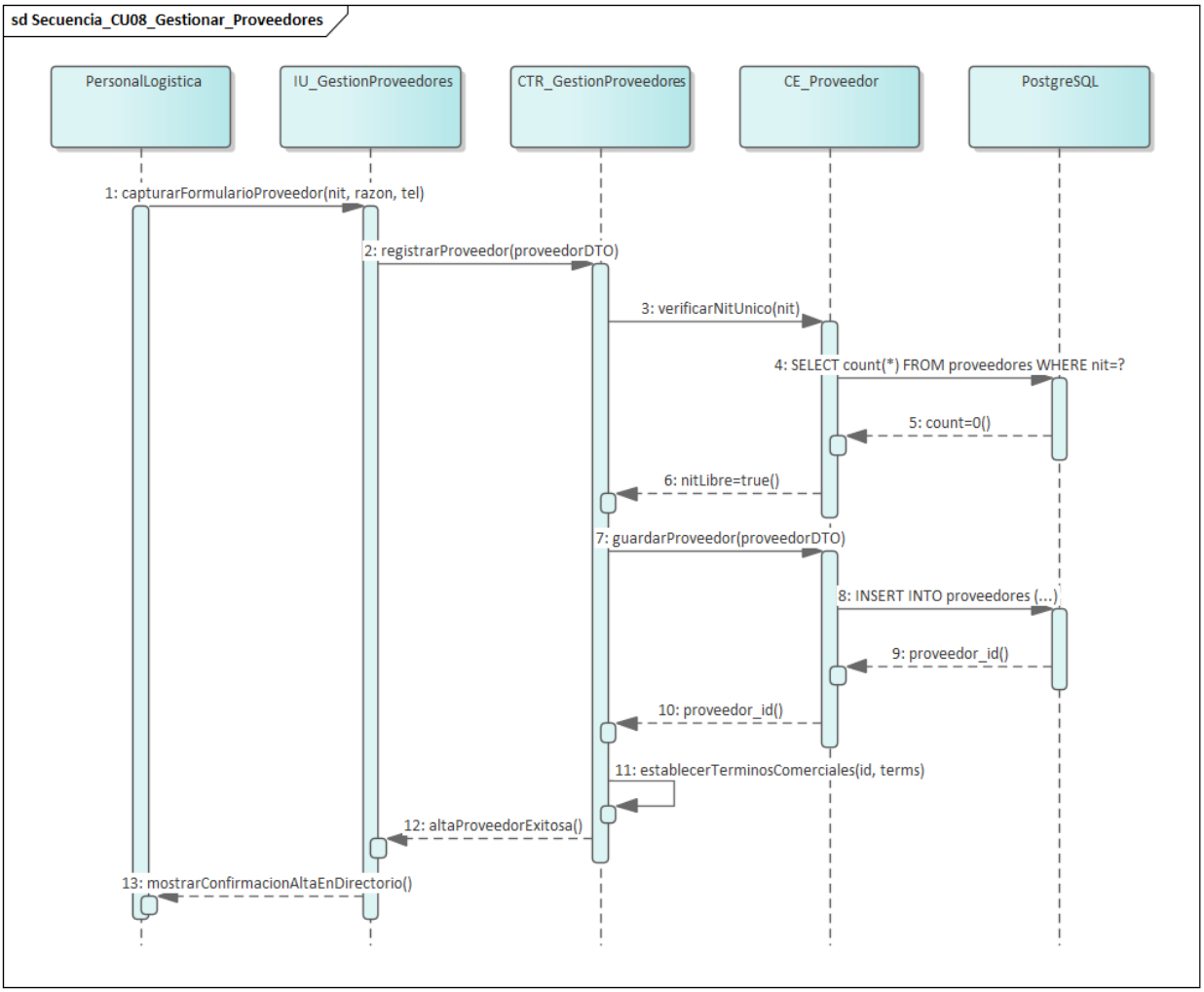
3.2.7 Diagrama de Secuencia - CU07: Gestionar Temporadas y Campañas

Modela la calendarización de temporadas comerciales, validación de ventanas temporales de vigencia, vinculación de productos y activación de descuentos por liquidación.



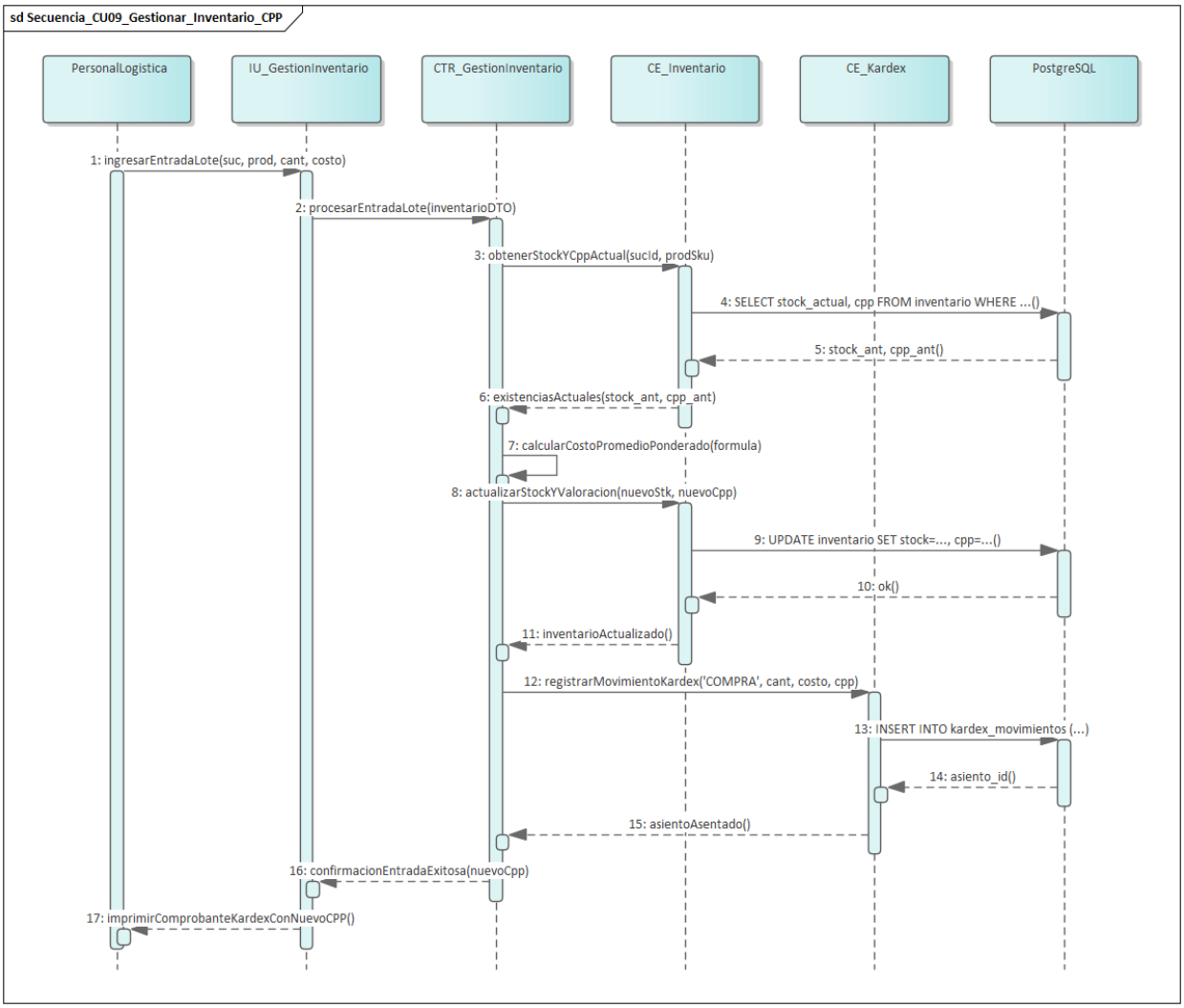
3.2.8 Diagrama de Secuencia - CU08: Gestionar Proveedores Textiles

Modela el registro de proveedores mayoristas de confección y telas, validación de NIT tributario único y asignación de acuerdos comerciales de pago.



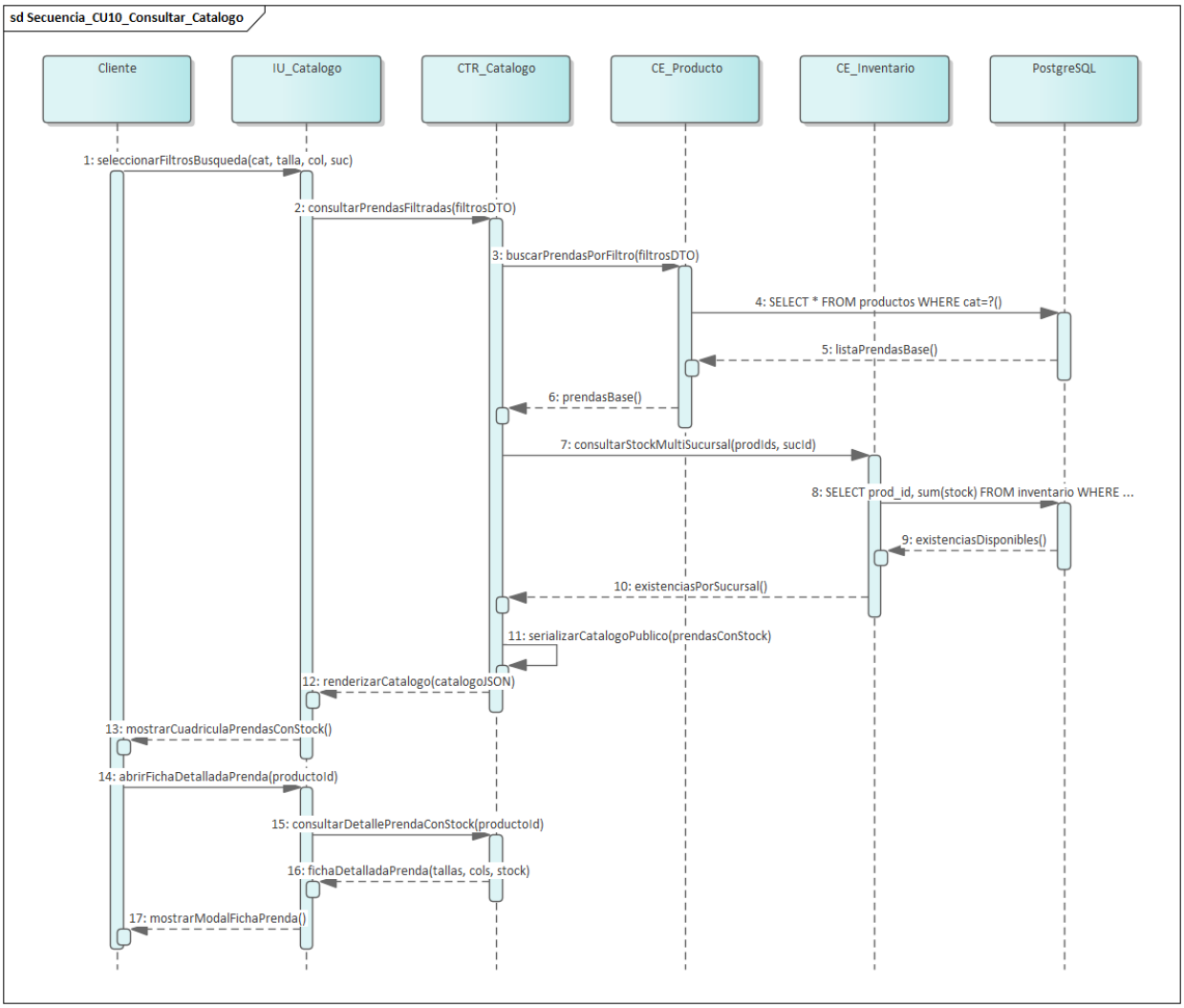
3.2.9 Diagrama de Secuencia - CU09: Gestionar Inventario Multi-Sucursal (CPP)

Modela la recepción física de lotes de mercadería, consulta de existencias y costos anteriores, recálculo matemático de Costo Promedio Ponderado (CPP*CPP*) y emisión inmutable de asiento en Kardex.



3.2.10 Diagrama de Secuencia - CU10: Consultar Catálogo y Disponibilidad

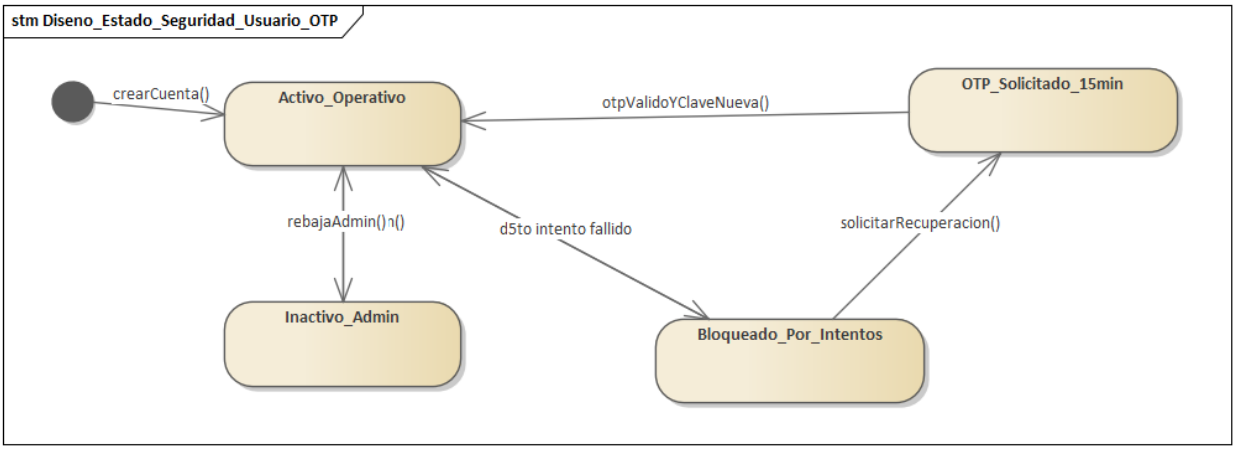
Modela la navegación reactiva del catálogo omnicanal, aplicación de filtros facetados (talla, color, categoría) y consulta federada de stock físico en tiempo real por sucursal.



3.2.2 Diagramas de Estado - Dominio del Sistema

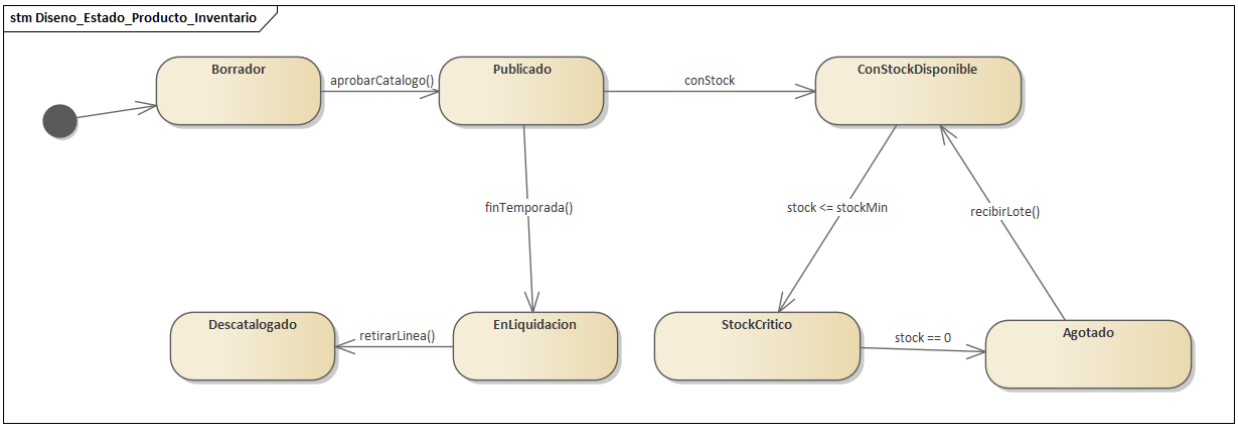
a) Ciclo de Vida de la Cuenta de Usuario y Token OTP (Seguridad)

Modela los estados de seguridad de la cuenta frente a intentos de acceso y el ciclo de vida efímero del token de recuperación:



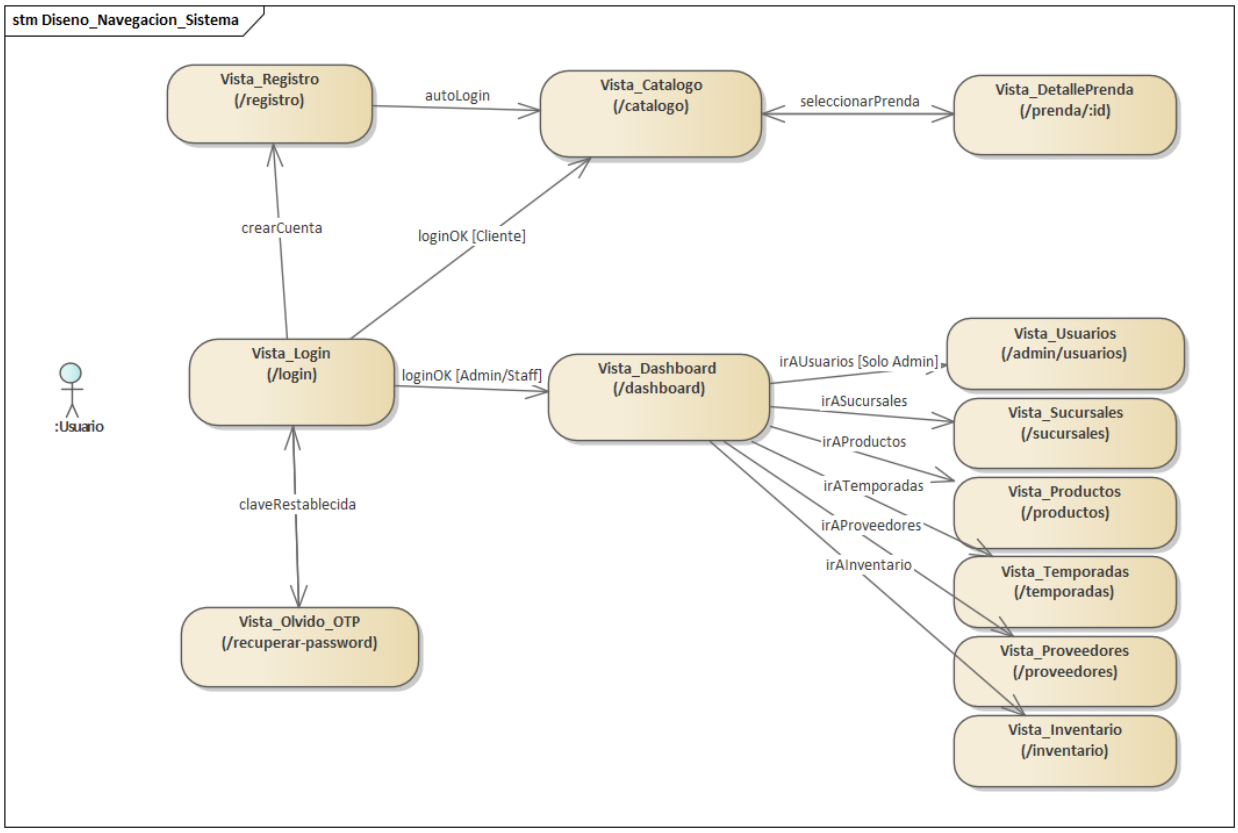
b) Ciclo de Vida del Producto e Inventario

Modela las transiciones del ciclo de vida de una prenda de vestir en la tienda:

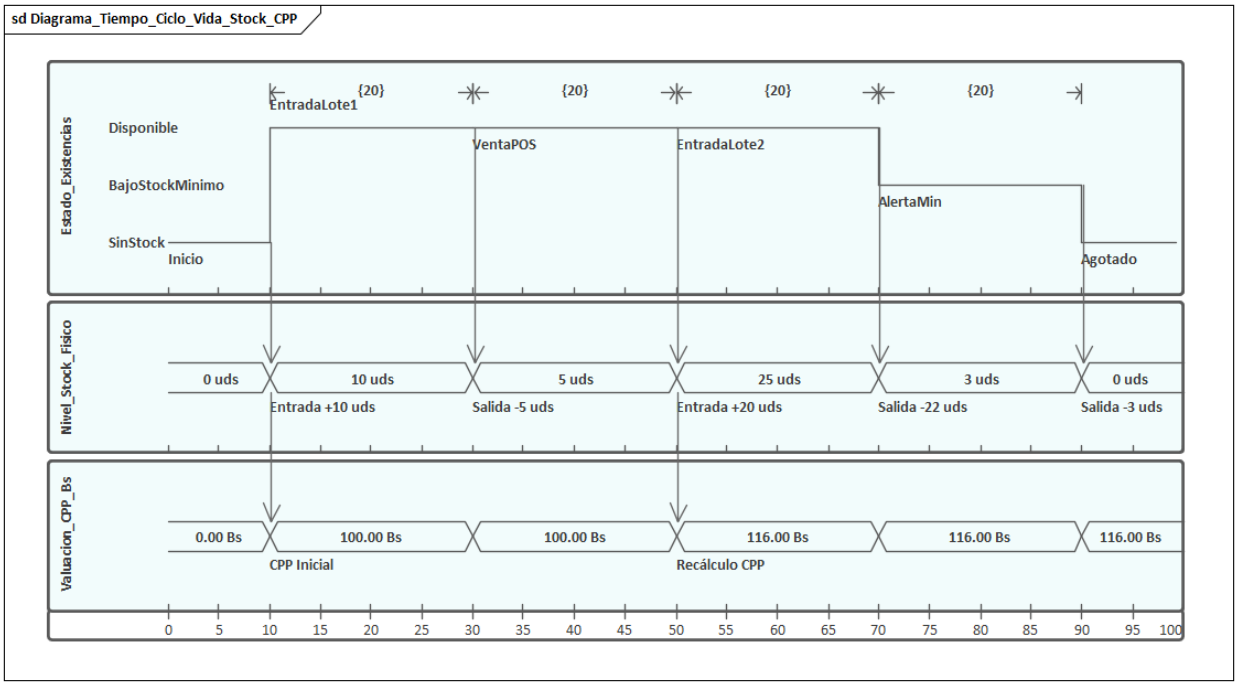


3.2.3 Diagrama de Navegación del Sistema (Ciclo 1)

Modela el flujo de navegación entre vistas y pantallas para la plataforma web y móvil:



3.2.4 Diagrama de Tiempo



b) Descripción Técnica y Dinámica de Transición en el Tiempo

El diagrama modela 3 líneas de vida sincronizadas que comparten la misma escala de tiempo horizontal (t∈[0,100]*t*∈[0,100]):

1. **Línea de Vida de Estado Robusta (Estado_Existencias)**:
    - **Disponible**: Existencias operativas por encima del umbral de seguridad.
    - **BajoStockMinimo**: Existencias ≤5 unidades≤5 unidades (stock de seguridad configurado por variante/sucursal). Dispara alertas automáticas en el módulo de compras y aprovisionamiento.
    - **SinStock**: Existencias iguales a 0 unidades0 unidades. Inhabilita la venta física y bloquea la adición al carrito web/móvil.
2. **Línea de Vida de Valor Concisa (Nivel_Stock_Fisico)**:
    - Refleja el saldo cuantitativo de unidades en almacén: 0 uds→10 uds→5 uds→25 uds→3 uds→0 uds0 uds→10 uds→5 uds→25 uds→3 uds→0 uds.
3. **Línea de Vida de Valor Concisa (Valuacion_CPP_Bs)**:
    - Refleja el costo unitario valorizado bajo la fórmula legal/tributaria de Costo Promedio Ponderado (CPP*CPP*):

CPPnuevo=(Sactual×CPPactual)+(Qentrada×Pcompra)Sactual+Qentrada*CPPnuevo*=*Sactual* +*Qentrada*(*Sactual*×*CPPactual*)+(*Qentrada*×*Pcompra*)

    - Las salidas por venta disminuyen las unidades físicas pero **no alteran** el costo unitario de valuación del inventario remanente.

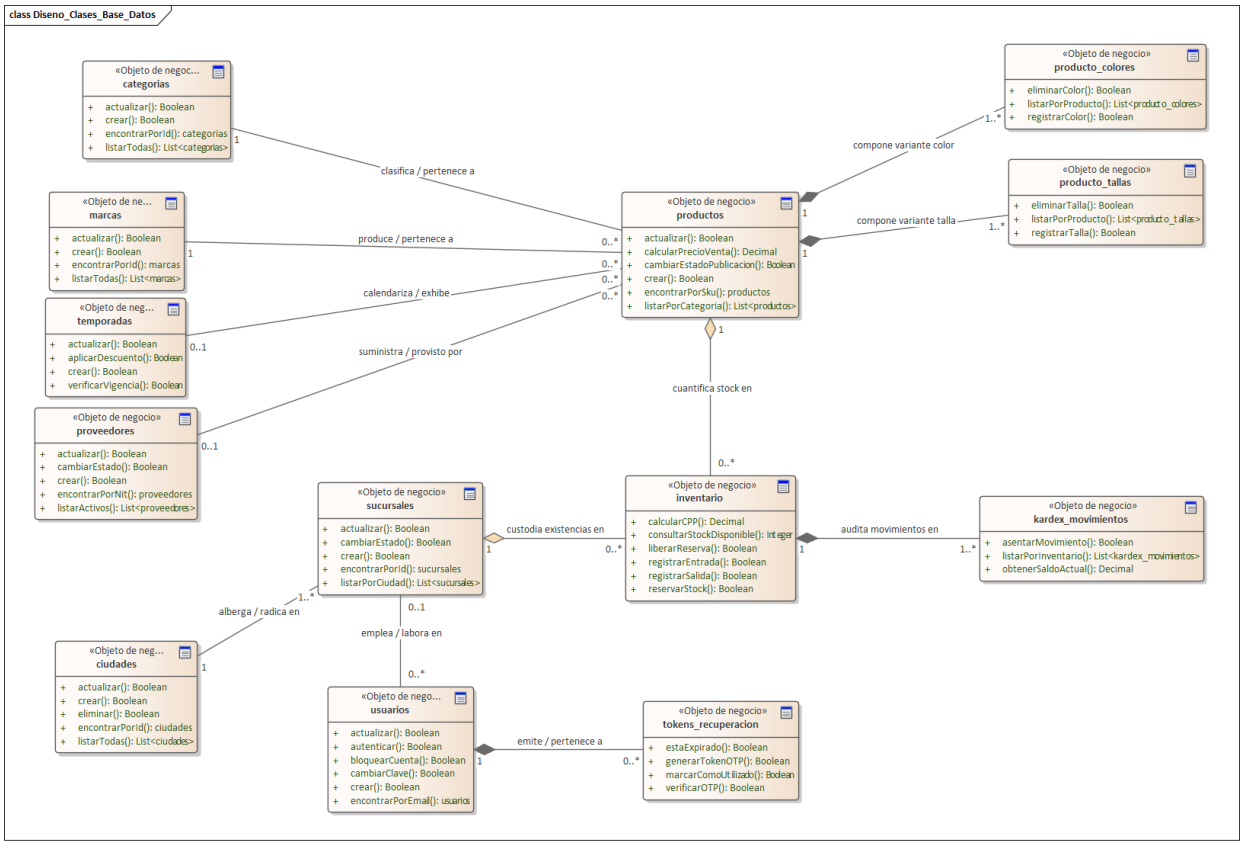
Tiempo (*tt*)	**Evento / Estímulo**	**Estado Existencias**	**Stock Físico**	**Valuación CPP (Bs)**	**Restricción / Observación**
0	Inicio	SinStock	0 uds	0.00 Bs	Apertura de catálogo sin existencias iniciales.
10	EntradaLote1	Disponible	10 uds	100.00 Bs	Recepción Lote 1: 10 prendas a 100.00 Bs/ud.
30	VentaPOS	Disponible	5 uds	100.00 Bs	Venta presencial: -5 uds. Duración {20}{20}. CPP no cambia.
50	EntradaLote2	Disponible	25 uds	116.00 Bs	Recepción Lote 2: 20 prendas a 120.00 Bs. Recálculo CPP: 5×100+20×12025=116.00255×100+20×120=116.00 Bs.
70	AlertaMin	BajoStockMinimo	3 uds	116.00 Bs	Venta: -22 uds. Stock ≤5≤5 uds mínimas. Dispara alerta automática.
90	Agotado	SinStock	0 uds	116.00 Bs	Venta de las últimas 3 prendas. Stock agotado.

c) Organización en el Repositorio de Enterprise Architect

- **Paquete Dedicado (Carpeta)**: Model →→ diagramas de tiempo (ID: 13)
- **Diagrama UML 2.5**: Diagrama_Tiempo_Ciclo_Vida_Stock_CPP (ID: 193, Tipo: Timing)
- **Elementos Modelados**:
  - Estado_Existencias (TimeLine, Subtipo: State Lifeline, Particiones: SinStock, BajoStockMinimo, Disponible)
  - Nivel_Stock_Fisico (TimeLine, Subtipo: Value Lifeline, Particiones: 0 uds, 3 uds, 5 uds, 10 uds, 25 uds)
  - Valuacion_CPP_Bs (TimeLine, Subtipo: Value Lifeline, Particiones: 0.00 Bs, 100.00 Bs, 116.00 Bs)
  - Conectores de estímulo/secuencia temporizados entre líneas de vida en t=10,30,50,70,90*t*=10,30,50,70,90 con restricciones de duración estándar UML ({20}{20}).

3.3 Diseño de Datos

3.3.1 Diseño de Datos Lógico (Diagrama ER / Clases Persistentes)



3.3.2 Diseño de Datos Físico (Script DDL SQL en PostgreSQL)

SCRIPT DE BASE DE DATOS (SQL POSTGRESQL)

```
-- ==========================================================================
-- SCRIPT DDL: BASE DE DATOS FASHIONSTORE (CICLO 1: FUNDAMENTOS Y MÓDULOS BASE)
-- Motor: PostgreSQL 15+ | Esquema Normalizado 3FN
-- ==========================================================================

CREATE TABLE ciudades (
    id_ciudad SERIAL PRIMARY KEY,
    nombre_ciudad VARCHAR(80) NOT NULL UNIQUE,
    departamento VARCHAR(80) NOT NULL,
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE sucursales (
    id_sucursal SERIAL PRIMARY KEY,
    id_ciudad INTEGER NOT NULL REFERENCES ciudades(id_ciudad) ON DELETE RESTRICT,
    nombre_sucursal VARCHAR(100) NOT NULL,
    direccion TEXT NOT NULL,
    latitud DECIMAL(10, 8) NOT NULL,
    longitud DECIMAL(11, 8) NOT NULL,
    telefono VARCHAR(20),
    capacidad_probadores INTEGER NOT NULL DEFAULT 4 CHECK (capacidad_probadores > 0),
    estado VARCHAR(20) NOT NULL DEFAULT 'OPERATIVA' CHECK (estado IN ('OPERATIVA', 'MANTENIMIENTO',
'CERRADA')),
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE usuarios (
    id_usuario SERIAL PRIMARY KEY,
    id_sucursal INTEGER REFERENCES sucursales(id_sucursal) ON DELETE SET NULL,
    nombres VARCHAR(100) NOT NULL,
    apellidos VARCHAR(100) NOT NULL,
    email VARCHAR(120) NOT NULL UNIQUE,
    password_hash VARCHAR(100) NOT NULL,
    rol VARCHAR(30) NOT NULL CHECK (rol IN ('ADMINISTRADOR', 'ENCARGADO_SUCURSAL', 'CAJERO',
'LOGISTICA', 'CLIENTE', 'PROVEEDOR')),
    estado_cuenta VARCHAR(30) NOT NULL DEFAULT 'ACTIVO' CHECK (estado_cuenta IN ('ACTIVO', 'INACTIVO',
'BLOQUEADO_POR_INTENTOS')),
    intentos_fallidos INTEGER NOT NULL DEFAULT 0,
    bloqueado_hasta TIMESTAMP,
    ultimo_acceso TIMESTAMP,
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Tabla para tokens de recuperación de contraseña con código OTP temporal (CU03)
CREATE TABLE tokens_recuperacion (
    id_token SERIAL PRIMARY KEY,
    id_usuario INTEGER NOT NULL REFERENCES usuarios(id_usuario) ON DELETE CASCADE,
    codigo_otp_hash VARCHAR(100) NOT NULL,
    expiracion TIMESTAMP NOT NULL,
    utilizado BOOLEAN NOT NULL DEFAULT FALSE,
    intentos_verificacion INTEGER NOT NULL DEFAULT 0,
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE bitacora_accesos (
    id_bitacora SERIAL PRIMARY KEY,
    id_usuario INTEGER REFERENCES usuarios(id_usuario) ON DELETE CASCADE,
    ip_origen VARCHAR(45) NOT NULL,
    user_agent TEXT,
    exitoso BOOLEAN NOT NULL,
    fecha_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE proveedores (
    id_proveedor SERIAL PRIMARY KEY,
    nit_identificacion VARCHAR(30) NOT NULL UNIQUE,
```

```
    razon_social VARCHAR(150) NOT NULL,
    contacto_nombre VARCHAR(100),
    telefono VARCHAR(30),
    email VARCHAR(100),
    terminos_pago VARCHAR(50) DEFAULT 'CONTADO',
    estado VARCHAR(20) DEFAULT 'ACTIVO',
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE temporadas (
    id_temporada SERIAL PRIMARY KEY,
    codigo_campana VARCHAR(30) NOT NULL UNIQUE,
    nombre_temporada VARCHAR(100) NOT NULL,
    fecha_inicio DATE NOT NULL,
    fecha_fin DATE NOT NULL,
    descuento_liquidacion DECIMAL(5,2) DEFAULT 0.00,
    estado VARCHAR(30) DEFAULT 'VIGENTE' CHECK (estado IN ('PLANIFICADA', 'VIGENTE', 'LIQUIDACION',
'FINALIZADA')),
    CONSTRAINT chk_fechas_temporada CHECK (fecha_fin >= fecha_inicio)
);

CREATE TABLE categorias (
    id_categoria SERIAL PRIMARY KEY,
    nombre_categoria VARCHAR(80) NOT NULL UNIQUE,
    descripcion TEXT
);

CREATE TABLE marcas (
    id_marca SERIAL PRIMARY KEY,
    nombre_marca VARCHAR(80) NOT NULL UNIQUE
);

CREATE TABLE productos (
    id_producto SERIAL PRIMARY KEY,
    id_categoria INTEGER NOT NULL REFERENCES categorias(id_categoria) ON DELETE RESTRICT,
    id_marca INTEGER NOT NULL REFERENCES marcas(id_marca) ON DELETE RESTRICT,
    id_temporada INTEGER REFERENCES temporadas(id_temporada) ON DELETE SET NULL,
    id_proveedor INTEGER REFERENCES proveedores(id_proveedor) ON DELETE SET NULL,
    codigo_sku_base VARCHAR(50) NOT NULL UNIQUE,
    nombre VARCHAR(150) NOT NULL,
    descripcion TEXT,
    precio_venta_base DECIMAL(10, 2) NOT NULL CHECK (precio_venta_base > 0),
    genero VARCHAR(20) NOT NULL DEFAULT 'MASCULINO',
    estado_publicacion VARCHAR(30) DEFAULT 'PUBLICADO' CHECK (estado_publicacion IN ('BORRADOR',
'PUBLICADO', 'DESCATALOGADO')),
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE producto_colores (
    id_prod_color SERIAL PRIMARY KEY,
    id_producto INTEGER NOT NULL REFERENCES productos(id_producto) ON DELETE CASCADE,
    color_nombre VARCHAR(50) NOT NULL,
    codigo_hex VARCHAR(10) NOT NULL,
    CONSTRAINT uq_producto_color UNIQUE (id_producto, color_nombre)
);

CREATE TABLE producto_tallas (
    id_prod_talla SERIAL PRIMARY KEY,
    id_producto INTEGER NOT NULL REFERENCES productos(id_producto) ON DELETE CASCADE,
    talla VARCHAR(20) NOT NULL,
    CONSTRAINT uq_producto_talla UNIQUE (id_producto, talla)
);

CREATE TABLE inventario (
    id_inventario SERIAL PRIMARY KEY,
    id_sucursal INTEGER NOT NULL REFERENCES sucursales(id_sucursal) ON DELETE CASCADE,
    id_producto INTEGER NOT NULL REFERENCES productos(id_producto) ON DELETE CASCADE,
    talla VARCHAR(20) NOT NULL,
    color VARCHAR(50) NOT NULL,
```

```
    stock_fisico INTEGER NOT NULL DEFAULT 0 CHECK (stock_fisico >= 0),
    stock_reservado INTEGER NOT NULL DEFAULT 0 CHECK (stock_reservado >= 0),
    stock_minimo INTEGER NOT NULL DEFAULT 5 CHECK (stock_minimo >= 0),
    stock_maximo INTEGER NOT NULL DEFAULT 100 CHECK (stock_maximo >= stock_minimo),
    ultimo_costo_unitario DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
    costo_promedio_ponderado DECIMAL(10, 2) NOT NULL DEFAULT 0.00,
    CONSTRAINT uq_inventario_variante UNIQUE (id_sucursal, id_producto, talla, color),
    CONSTRAINT chk_stock_valido CHECK (stock_fisico >= stock_reservado)
);

CREATE TABLE kardex_movimientos (
    id_kardex SERIAL PRIMARY KEY,
    id_inventario INTEGER NOT NULL REFERENCES inventario(id_inventario) ON DELETE CASCADE,
    tipo_movimiento VARCHAR(30) NOT NULL CHECK (tipo_movimiento IN ('COMPRA', 'VENTA', 'RESERVA',
'AJUSTE_MERMA', 'TRANSFERENCIA_ENTRADA', 'TRANSFERENCIA_SALIDA')),
    cantidad INTEGER NOT NULL,
    costo_unitario_mov DECIMAL(10, 2) NOT NULL,
    saldo_cantidad INTEGER NOT NULL,
    saldo_cpp DECIMAL(10, 2) NOT NULL,
    descripcion_motivo VARCHAR(200),
    fecha_movimiento TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Índices de alto rendimiento para búsquedas omnicanales
CREATE INDEX idx_productos_sku ON productos(codigo_sku_base);
CREATE INDEX idx_productos_categoria ON productos(id_categoria);
CREATE INDEX idx_inventario_sucursal ON inventario(id_sucursal, id_producto);
CREATE INDEX idx_inventario_stock_disp ON inventario((stock_fisico - stock_reservado));
```

. . . . . .

3.3.3 Población de Datos (Script DML SQL en PostgreSQL)

SCRIPT DE BASE DE DATOS (SQL POSTGRESQL)

```
-- ==========================================================================
-- SCRIPT DML: POBLACIÓN DE DATOS SEMILLA PARA EL CICLO 1
-- Incluye Ciudades de Bolivia, Sucursales, Usuarios con BCrypt, Ropa Masculina,
-- Colores Multivaluados, Tallas e Inventario con Costo Promedio Ponderado
-- ==========================================================================

-- 1. Ciudades
INSERT INTO ciudades (nombre_ciudad, departamento) VALUES
('Santa Cruz de la Sierra', 'Santa Cruz'),
('La Paz', 'La Paz'),
('Cochabamba', 'Cochabamba');

-- 2. Sucursales
INSERT INTO sucursales (id_ciudad, nombre_sucursal, direccion, latitud, longitud, telefono,
capacidad_probadores) VALUES
(1, 'Sucursal Equipetrol', 'Av. San Martín #450, entre 3er y 4to anillo', -17.76823000, -63.18342000,
'3-3445566', 6),
(1, 'Sucursal Centro', 'Calle 21 de Mayo esq. Ayacucho', -17.78312000, -63.18210000, '3-3332211', 4),
(2, 'Sucursal Calacoto', 'Av. Ballivián #1200, Calle 18', -16.53982000, -68.08921000, '2-2778899', 5),
(3, 'Sucursal El Prado', 'Av. Ballivián #600, El Prado', -17.38921000, -66.15672000, '4-4223344', 4);

-- 3. Usuarios del Sistema (Contraseña hash: 'Admin123*' cifrada con bcrypt)
-- Hash generado: $2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu
INSERT INTO usuarios (id_sucursal, nombres, apellidos, email, password_hash, rol, estado_cuenta,
intentos_fallidos) VALUES
(NULL, 'Alberto', 'Delgado', 'alberto.delgado@store.bo',
'$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', 'ADMINISTRADOR', 'ACTIVO', 0),
(NULL, 'Andy', 'Mujica', 'andy.mujica@store.bo',
'$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', 'ADMINISTRADOR', 'ACTIVO', 0),
(1, 'Carlos', 'Morales', 'carlos.morales@store.bo',
'$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', 'ENCARGADO_SUCURSAL', 'ACTIVO', 0),
(1, 'Javier', 'Roca', 'javier.roca@store.bo',
'$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', 'CAJERO', 'BLOQUEADO_POR_INTENTOS',
```

```
5),
(NULL, 'Mateo', 'Suarez', 'mateo.logistica@store.bo',
'$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', 'LOGISTICA', 'ACTIVO', 0),
(NULL, 'Rodrigo', 'Paz', 'rodrigo.cliente@gmail.com',
'$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', 'CLIENTE', 'ACTIVO', 0);

-- Semilla de tokens de recuperación OTP (CU03)
INSERT INTO tokens_recuperacion (id_usuario, codigo_otp_hash, expiracion, utilizado,
intentos_verificacion) VALUES
(6, '$2b$12$e8Y4vFz7BvK8/9zD6x1Q4uS0.1234567890abcdefghijklmnopqrstu', CURRENT_TIMESTAMP + INTERVAL '15
MINUTE', FALSE, 0);

-- 4. Proveedores Textiles
INSERT INTO proveedores (nit_identificacion, razon_social, contacto_nombre, telefono, email,
terminos_pago) VALUES
('1028392019', 'Confecciones Andina SA', 'Juan Paredes', '+591 70012345', 'ventas@andina.com.bo',
'CREDITO_30_DIAS'),
('9038472011', 'Hilanderías del Sur SRL', 'Marcos Vaca', '+591 71098765', 'contacto@hilasur.bo',
'CONTADO'),
('3829104018', 'Importadora Textil Italiana', 'Gianluca Rossi', '+591 72055443',
'grossi@textilitaliana.com', 'CREDITO_60_DIAS');

-- 5. Temporadas Comerciales
INSERT INTO temporadas (codigo_campana, nombre_temporada, fecha_inicio, fecha_fin,
descuento_liquidacion, estado) VALUES
('SS-2026', 'Primavera - Verano 2026', '2026-08-01', '2027-01-31', 0.00, 'VIGENTE'),
('FW-2026', 'Otoño - Invierno 2026', '2026-02-01', '2026-07-31', 25.00, 'LIQUIDACION');

-- 6. Categorías y Marcas
INSERT INTO categorias (nombre_categoria, descripcion) VALUES
('Camisas Formales', 'Camisas ejecutivas y de vestir en algodón peinado y lino'),
('Pantalones Casuales', 'Pantalones estilo chino, gabardina y corte clásico'),
('Trajes y Blazers', 'Sacos, chaquetas y trajes de dos piezas en lana fría'),
('Calzado Ejecutivo', 'Zapatos Oxford, Derby y mocasines de cuero legítimo');

INSERT INTO marcas (nombre_marca) VALUES
('Oxford Heritage'),
('Urban Tailor'),
('Sartorial Milano'),
('Bocaccio Leather');

-- 7. Productos (Prendas Masculinas)
INSERT INTO productos (id_categoria, id_marca, id_temporada, id_proveedor, codigo_sku_base, nombre,
descripcion, precio_venta_base, genero) VALUES
(1, 1, 1, 1, 'SHIRT-SLIM-001', 'Camisa Oxford Slim Fit', 'Camisa de corte entallado en 100% algodón
egipcio, cuello italiano', 280.00, 'MASCULINO'),
(2, 2, 1, 2, 'PANT-CHINO-002', 'Pantalón Chino Gabardina', 'Pantalón casual de gabardina elastizada con
bolsillos traseros ojales', 320.00, 'MASCULINO'),
(3, 3, 1, 3, 'BLAZ-LINO-003', 'Blazer de Lino Casual', 'Chaqueta ligera desestructurada de lino ideal
para clima templado y cálido', 650.00, 'MASCULINO');

-- 8. Colores Multivaluados
INSERT INTO producto_colores (id_producto, color_nombre, codigo_hex) VALUES
(1, 'Azul Marino', '#000080'),
(1, 'Blanco Óptico', '#FFFFFF'),
(1, 'Celeste Cielo', '#87CEEB'),
(2, 'Beige Arena', '#F5F5DC'),
(2, 'Azul Noche', '#191970'),
(2, 'Verde Oliva', '#556B2F'),
(3, 'Azul Cobalto', '#0047AB'),
(3, 'Gris Plomo', '#708090');

-- 9. Tallas Normalizadas
INSERT INTO producto_tallas (id_producto, talla) VALUES
(1, 'S'), (1, 'M'), (1, 'L'), (1, 'XL'),
(2, '30'), (2, '32'), (2, '34'), (2, '36'),
(3, '38'), (3, '40'), (3, '42');

-- 10. Inventario con Costo Unitario y Costo Promedio Ponderado (CPP)
```

```
-- Ejemplo real demostrativo:
-- Camisa Oxford Slim Fit (Talla M, Azul Marino) en Sucursal Equipetrol:
-- Lote 1: 15 unidades a 90 Bs = 1350 Bs
-- Lote 2: 20 unidades a 120 Bs = 2400 Bs
-- Total unidades: 35 | Costo Total: 3750 Bs | CPP = 3750 / 35 = 107.14 Bs | Último Costo = 120.00 Bs
INSERT INTO inventario (id_sucursal, id_producto, talla, color, stock_fisico, stock_reservado,
stock_minimo, stock_maximo, ultimo_costo_unitario, costo_promedio_ponderado) VALUES
(1, 1, 'M', 'Azul Marino', 35, 5, 10, 100, 120.00, 107.14),
(1, 1, 'L', 'Blanco Óptico', 25, 0, 5, 80, 110.00, 110.00),
(1, 2, '32', 'Beige Arena', 20, 2, 5, 60, 140.00, 135.00),
(2, 1, 'M', 'Azul Marino', 15, 0, 5, 50, 115.00, 115.00),
(3, 3, '40', 'Gris Plomo', 8, 1, 2, 30, 320.00, 310.00);

-- 11. Asientos de Kardex Inicial
INSERT INTO kardex_movimientos (id_inventario, tipo_movimiento, cantidad, costo_unitario_mov,
saldo_cantidad, saldo_cpp, descripcion_motivo) VALUES
(1, 'COMPRA', 15, 90.00, 15, 90.00, 'Ingreso Lote Inicial Fac-101'),
(1, 'COMPRA', 20, 120.00, 35, 107.14, 'Ingreso Segundo Lote Fac-205 - Recálculo CPP');
```

. . . . . .

4. Flujo de Trabajo: Implementación

4.1 Selección de la Plataforma de Software

Conforme al requerimiento del enunciado y las notas de clase, se selecciona el siguiente stack tecnológico de alta eficiencia:

- **Backend**: **Python 3.11+ con FastAPI**. Permite arquitectura asíncrona no bloqueante (ASGI), serialización tipada y validación con Pydantic, e inyección nativa de dependencias. Expone la documentación automática OpenAPI en `/docs`.
- **Frontend Web**: **Angular 17+**. Basado en TypeScript, componentes reactivos, inyección de dependencias y modularidad estricta para la SPA administrativa y el catálogo web.
- **Aplicación Móvil**: **Flutter 3.x con Dart**. Compilación nativa para Android e iOS desde un solo repositorio, con soporte para ARCore y sensores de cámara.
- **Base de Datos**: **PostgreSQL 15+**. Motor relacional con integridad transaccional ACID, índices B-Tree y soporte geoespacial.
- **Contenedores**: **Docker & Docker Compose** para aislamiento de entornos y despliegue continuo en la nube (Render / Railway).

. . . . . .

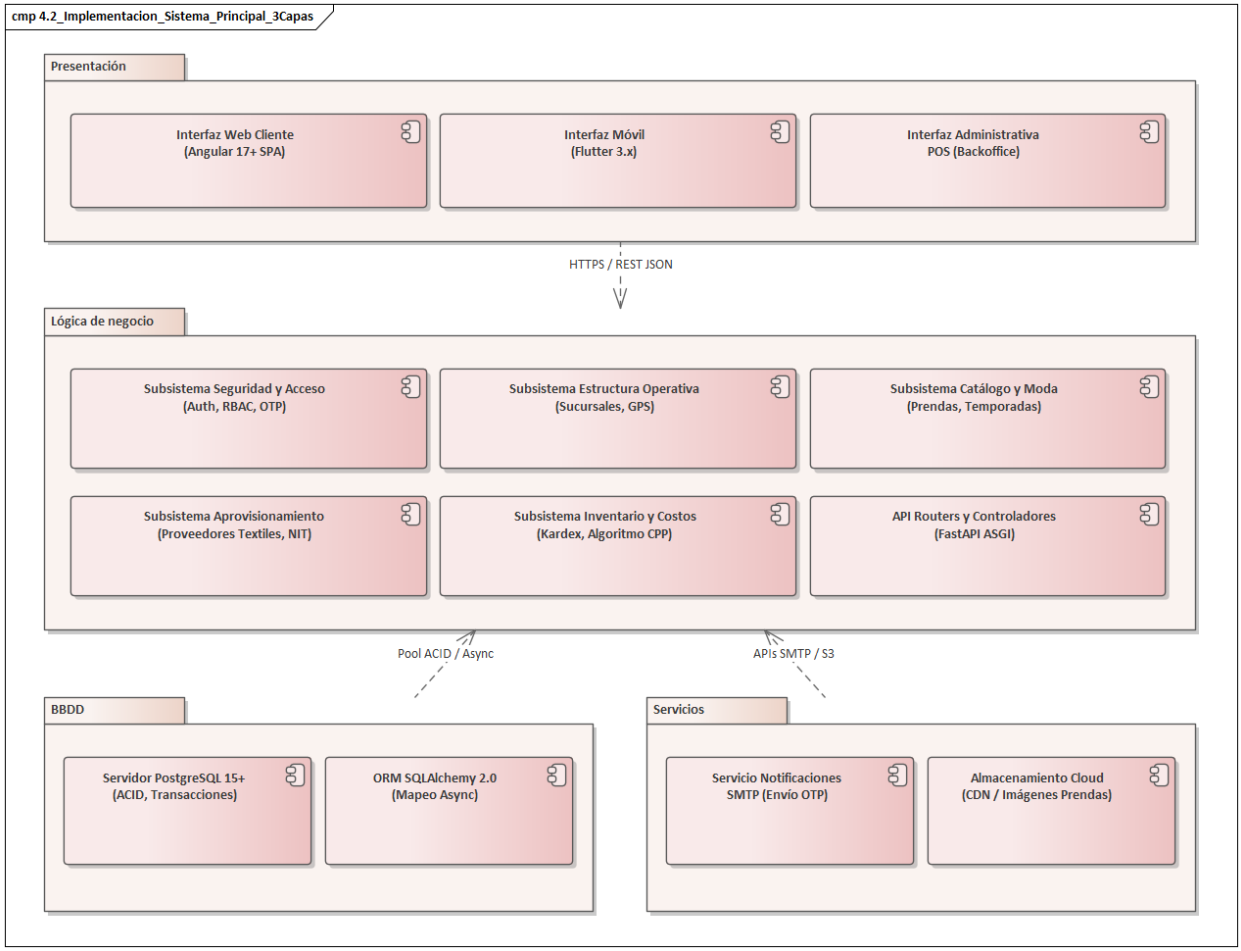
4.2 Implementación de la Arquitectura del Sistema Principal

La arquitectura del sistema backend sigue el patrón de **Arquitectura Limpia / Modular por Paquetes**:

El sistema global se estructura en 3 niveles de abstracción:

1. **Capa de Presentación**: Contiene los clientes frontend autónomos (Interfaz Web Cliente - Angular 17+ SPA, Interfaz Móvil - Flutter 3.x e Interfaz Administrativa POS / Backoffice). Se comunican de forma asíncrona mediante HTTPS / REST con payloads JSON.
2. **Capa de Lógica de Negocio**: Orquestada por el gateway ASGI FastAPI, alberga los 5 subsistemas funcionales desacoplados que resuelven los casos de uso empresariales.

3. **Capa de Persistencia y Servicios**: Subdividida en **BBDD** (Servidor PostgreSQL 15+ con motor transaccional ACID y ORM SQLAlchemy 2.0 Async) y **Servicios** externos (Servicio SMTP transaccional para despacho de códigos OTP y Almacenamiento Cloud / CDN para catálogo de imágenes).



4.3 Implementación de la Arquitectura del Sub Sistema

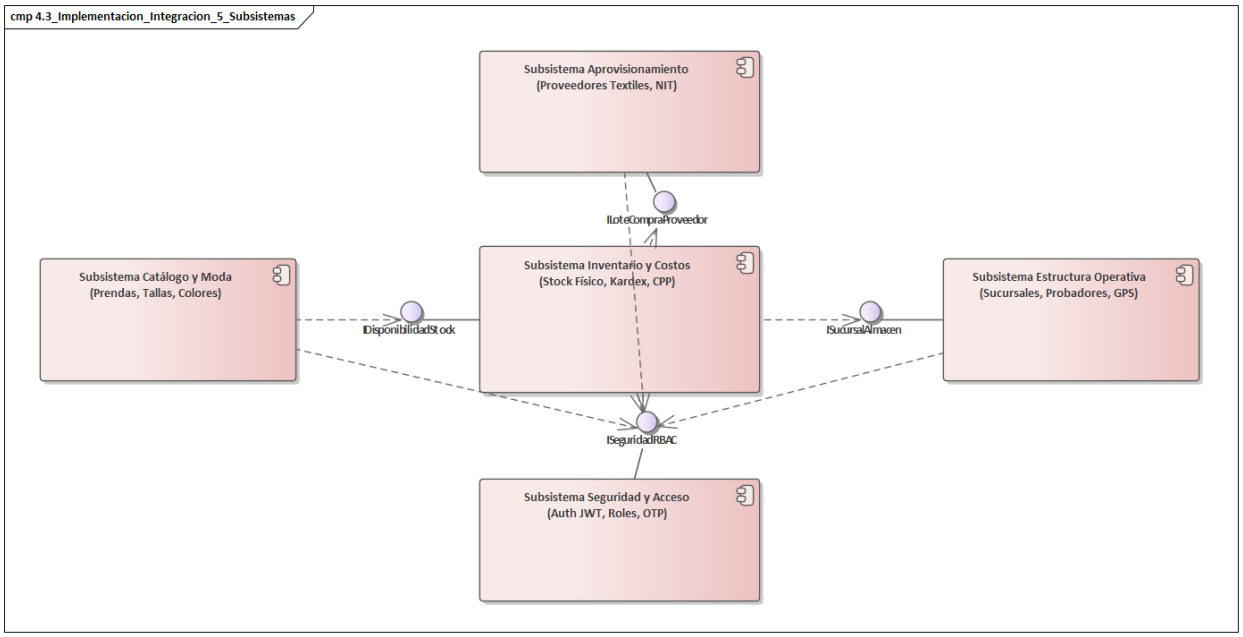
El sistema empresarial se compone de **5 subsistemas altamente cohesionados**, comunicados mediante interfaces de componentes estandarizadas según el patrón *ball-and-socket* (interfaces provistas / lollipop e interfaces requeridas / socket).

4.3.0 Integración de los 5 Subsistemas con Interfaces Provistas y Requeridas

El **Subsistema de Inventario y Costos** actúa como núcleo operativo central de la cadena de valor, integrando:

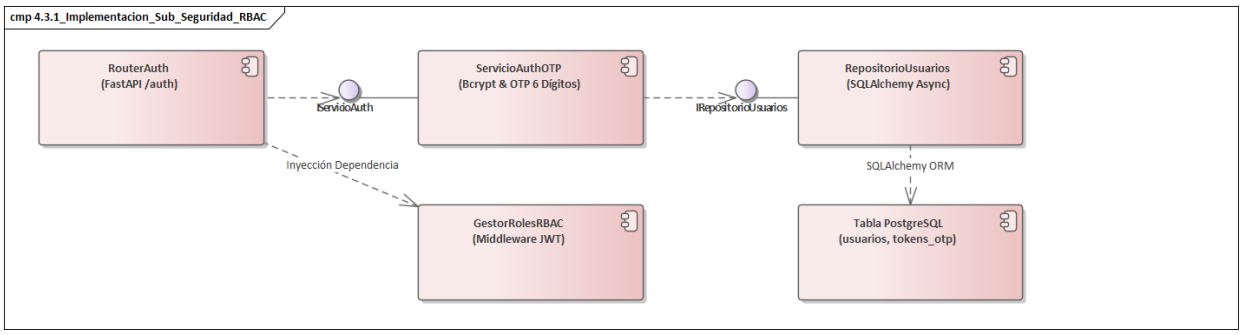
- **IDisponibilidadStock**: Proporcionada por Inventario para que el Catálogo consulte existencias en tiempo real por sucursal y variante.
- **ILoteCompraProveedor**: Proporcionada por Aprovisionamiento para liquidar compras textiles con NIT e ingresar lotes al almacén.

- **ISucursalAlmacen**: Proporcionada por Estructura Operativa para asignar existencias a bodegas de sucursal física.
- **ISeguridadRBAC**: Proporcionada por Seguridad y Acceso para autorizar y autenticar cada transacción con tokens JWT y roles.



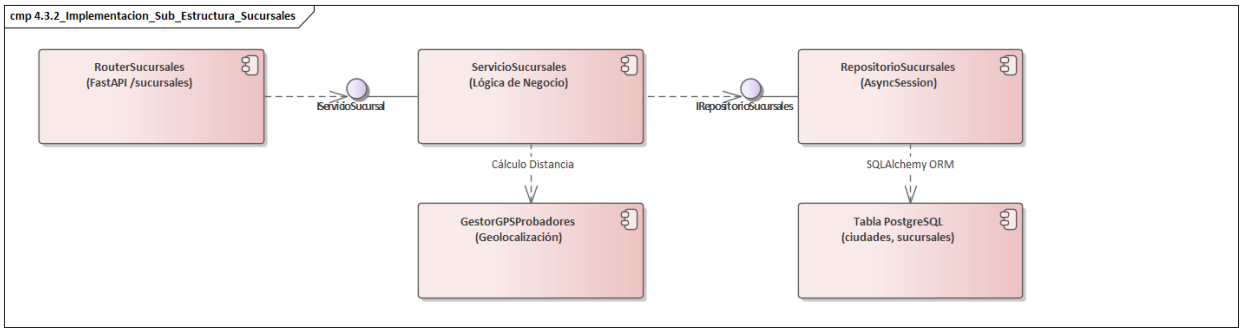
4.3.1 Sub Sistema 1: Seguridad y Control de Acceso RBAC (M01 - CU01, CU02, CU03, CU04)

Este subsistema encapsula la autenticación criptográfica con JWT, el control de acceso basado en roles (RBAC) con mitigación OWASP (bloqueo por 5 intentos erróneos durante 30 min) y la emisión de tokens OTP criptográficos de 6 dígitos con validez de 15 minutos.



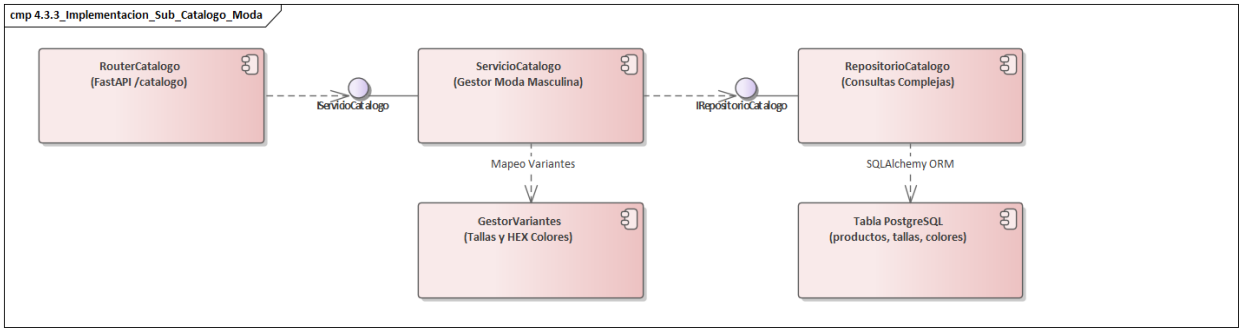
4.3.2 Sub Sistema 2: Estructura Operativa y Sucursales (M02 - CU05)

Administra las ciudades, direcciones, horarios, coordenadas GPS y disponibilidad física de probadores para reservas omnicanal en cada tienda.



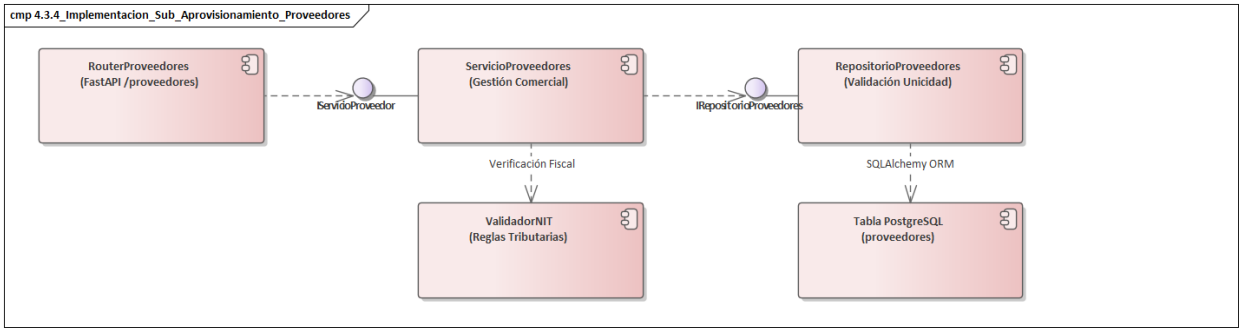
4.3.3 Sub Sistema 3: Catálogo y Moda Masculina (M03 y M04 - CU06, CU07, CU10)

Gestiona la matriz de prendas masculinas, colecciones por temporada, categorización multinivel y las variantes ortogonales de **Talla** (S, M, L, XL) y **Color** (nombre + código HEX).



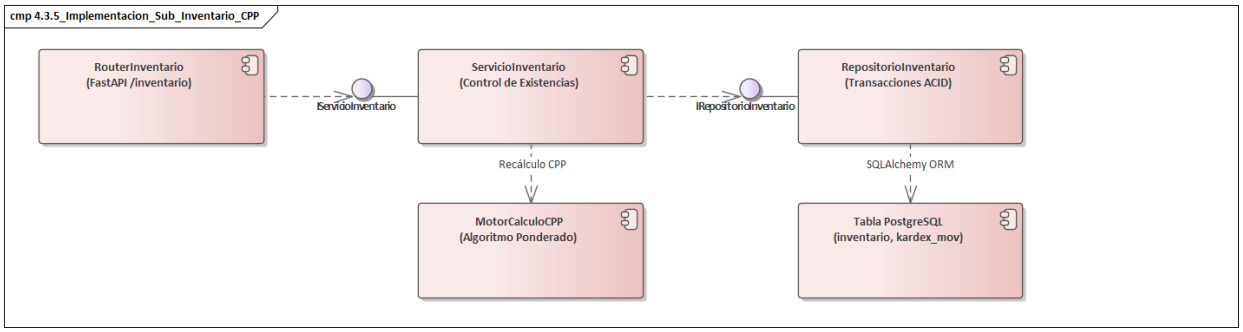
4.3.4 Sub Sistema 4: Aprovisionamiento y Proveedores Textiles (M05 - CU08)

Encargado del registro, calificación comercial y validación tributaria (NIT) de las empresas proveedoras de telas y confección para la reposición de stock.



4.3.5 Sub Sistema 5: Inventario Multitienda y Costo Promedio Ponderado CPP (M06 - CU09)

Controla el Kardex valorado inmutable, los movimientos físicos (compras, ventas, traslados entre sucursales) y aplica de forma rigurosa el recálculo matemático de valuación bajo el método del **Costo Promedio Ponderado (CPP)**.



5. Flujo de Trabajo: Pruebas

5.1 Pruebas de Casos de Uso (Pruebas de Caja Negra)

Se documentan las pruebas formales de caja negra realizadas sobre los 10 Casos de Uso del Ciclo 1 para validar exhaustivamente las condiciones de aceptación funcionales y de seguridad:

Prueba de caso de uso CU1: Iniciar Sesión y Autenticación RBAC

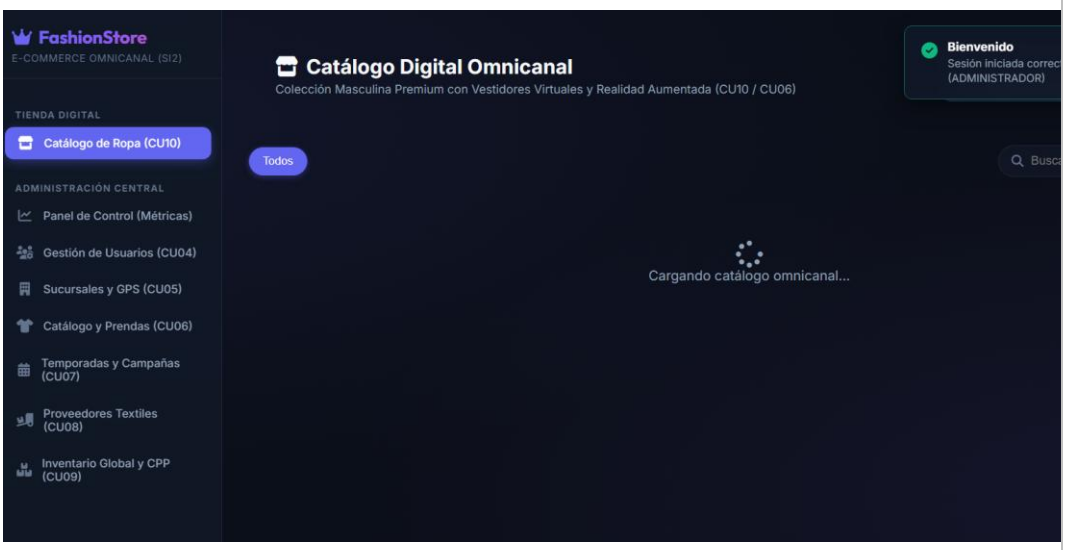
Caso de uso 1	**Iniciar Sesión y Autenticación RBAC**
Descripción	Permite a los usuarios autenticarse en la plataforma mediante correo electrónico y contraseña cifrada. Valida los roles del sistema (ADMINISTRADOR, CAJERO, CLIENTE) emitiendo un token JWT firmado y aplica el mecanismo de defensa OWASP bloqueando preventivamente la cuenta por 30 minutos al registrarse 5 intentos fallidos consecutivos.
Precondiciones	a) El usuario debe encontrarse registrado en la base de datos. b) El servicio de backend FastAPI y la base de datos PostgreSQL deben estar en ejecución. c) La cuenta no debe encontrarse en estado INACTIVO ni con bloqueo administrativo.

Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**
1	Acceder a la pantalla de Login desde el portal web o aplicación móvil.	Se despliega el formulario de autenticación con campos para email, contraseña y botón de acceso.	Satisfactorio
2	Ingresar credenciales válidas de usuario y	Se genera el token JWT con los claims de rol, se almacena en el cliente y se redirige a la vista correspondiente.	Satisfactorio

	presionar "Iniciar Sesión".		
3	Ingresar contraseña errónea de forma intencional en 4 oportunidades.	El sistema rechaza la autenticación con código HTTP 401 e informa los intentos restantes antes del bloqueo.	Satisfactorio
4	Realizar el quinto intento fallido con contraseña incorrecta.	La cuenta pasa automáticamente a estado BLOQUEADO_POR_INTENTOS con una ventana de 30 minutos.	Satisfactorio
5	Intentar iniciar sesión inmediatamente durante la ventana de bloqueo.	Se bloquea el acceso en el gateway notificando el tiempo restante de penalización sin consultar la clave.	Satisfactorio
6	Iniciar sesión tras expirar la ventana de 30 minutos con la clave correcta.	Se restablece el contador de intentos fallidos a 0 y se inicia sesión normalmente.	Satisfactorio

Responsable	**Administrador / Tester de Seguridad**
Resultado de la prueba	**Satisfactorio**

Adjunto	Evidencia de Interfaz:



Prueba de caso de uso CU2: Auto-registro de Clientes

Caso de uso 2	Auto-registro de Clientes
Descripción	Permite que visitantes no registrados puedan crear de manera autónoma una cuenta de cliente en FashionStore, registrando sus datos personales y credenciales de acceso, asignándoles automáticamente el rol CLIENTE e iniciando su sesión con token JWT.
Precondiciones	a) El visitante debe tener acceso a internet y al portal web o app móvil. b) El módulo de autenticación y registro debe estar habilitado. c) Debe existir conexión activa con el servidor de base de datos.

Paso	Acción	Resultado esperado	Estado (Satisfactorio/Fallido)
1	Acceder a la opción "Crear Cuenta" desde el menú principal de navegación.	Se despliega el formulario modal de registro de nuevo cliente con campos estructurados.	Satisfactorio
2	Completar datos personales (Nombre, Apellido, Celular, Género y Fecha de Nacimiento).	Los campos se validan sintácticamente en tiempo real mediante expresiones regulares.	Satisfactorio
3	Ingresar correo electrónico no registrado y contraseña	Se evalúa y muestra el indicador de fortaleza de contraseña como "Segura".	Satisfactorio

	segura (mínimo 8 caracteres).		
4	Intentar registrar un correo que ya existe en el sistema.	El sistema rechaza la solicitud indicando que el correo ya se encuentra registrado.	Satisfactorio
5	Presionar el botón "Registrarme e Iniciar Sesión" con datos válidos.	Se almacena el usuario con hash Bcrypt (factor 12), rol CLIENTE y estado ACTIVO.	Satisfactorio
6	Verificar inicio de sesión automático y redirección al catálogo.	Se genera el token JWT, se cierra el modal y se visualiza el saludo de bienvenida con sesión activa.	Satisfactorio

Responsable	**Cliente / Tester QA**
Resultado de la prueba	**Satisfactorio**

Adjunto	**Evidencia de Interfaz:**

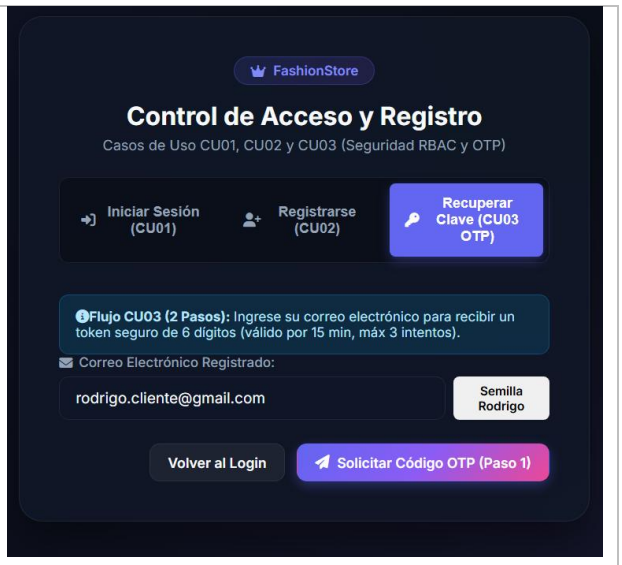
Prueba de caso de uso CU3: Recuperación de Contraseña con OTP

Caso de uso 3	**Recuperación de Contraseña con OTP**
Descripción	Permite a los usuarios que olvidaron su contraseña solicitar un código criptográfico de un solo uso (OTP) de 6 dígitos enviado a su correo registrado, con una validez temporal estricta de 15 minutos y un límite de 3 intentos de verificación, para restablecer de forma segura su clave.

Precondiciones	a) El correo del usuario debe estar registrado y activo en el sistema. b) El servicio transaccional de correo SMTP debe estar operativo. c) Debe existir conexión activa con la base de datos PostgreSQL.

Paso	Acción	Resultado esperado	Estado (Satisfactorio/Fallido)
1	Presionar el enlace "¿Olvidaste tu contraseña?" en el formulario de login.	Se despliega la vista de recuperación solicitando el correo electrónico de la cuenta.	Satisfactorio
2	Ingresar el correo registrado y presionar "Enviar Código de Verificación".	El sistema genera el OTP de 6 dígitos, guarda su hash con expiración a 15 min y despacha el correo.	Satisfactorio
3	Verificar recepción del código en la bandeja de entrada del correo.	Se recibe correo formal de FashionStore con el código numérico de 6 dígitos.	Satisfactorio
4	Ingresar código OTP incorrecto en el formulario de verificación.	El sistema rechaza el intento, incrementa el contador y alerta sobre los intentos restantes (máx 3).	Satisfactorio
5	Ingresar el código OTP válido y definir la nueva contraseña.	Se valida exitosamente el código, se actualiza el hash Bcrypt en BD y se invalida el token OTP.	Satisfactorio
6	Iniciar sesión inmediatamente con la nueva contraseña configurada.	Se valida la nueva clave con éxito y se ingresa a la plataforma sin trabas.	Satisfactorio

Responsable	**Cliente / Administrador**
Resultado de la prueba	**Satisfactorio**
Adjunto	**Evidencia de Interfaz:**

Prueba de caso de uso CU4: Gestión de Usuarios, Roles RBAC y Desbloqueo

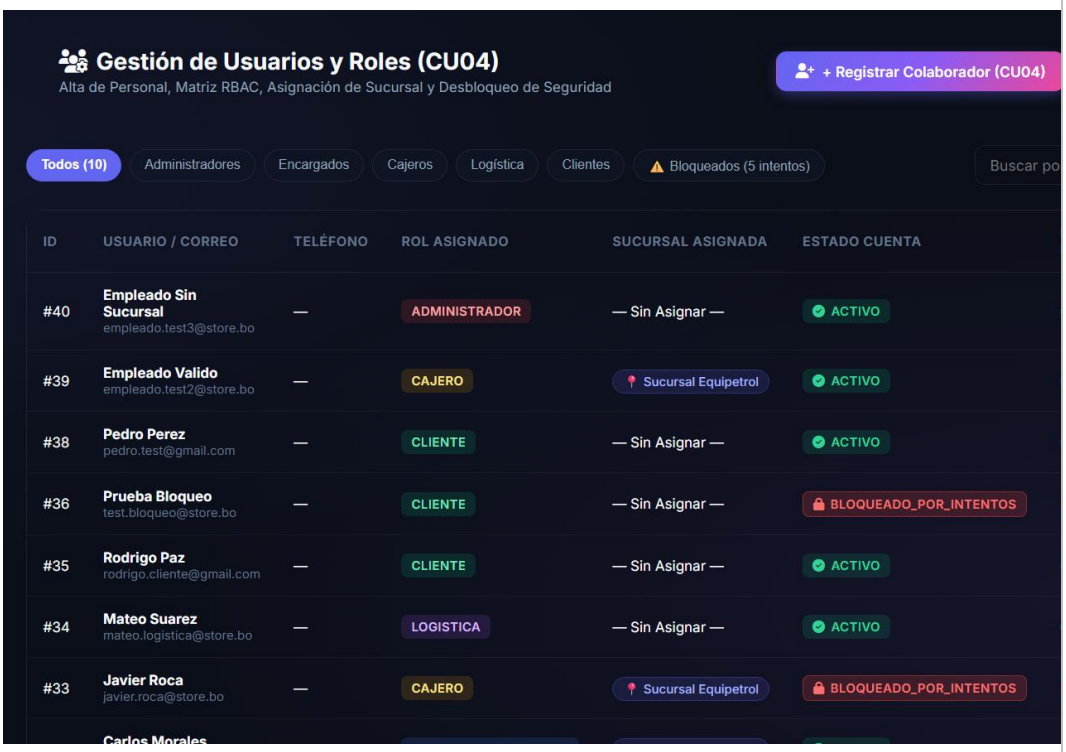
Caso de uso 4	**Gestión de Usuarios, Roles RBAC y Desbloqueo**
Descripción	Permite al Administrador dar de alta colaboradores internos (cajeros, encargados de tienda), asignar roles y sucursales operativas, y desbloquear cuentas de usuarios que hayan quedado bloqueadas por superar los intentos de login.
Precondiciones	a) El usuario autenticado debe tener el rol ADMINISTRADOR. b) El módulo de administración de usuarios debe encontrarse activo. c) Debe existir conexión activa con la base de datos.

Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**
1	Acceder al módulo "Usuarios y Roles" desde el menú lateral administrativo.	Se despliega la tabla de colaboradores y clientes con columnas de rol, sucursal y estado de cuenta.	Satisfactorio
2	Presionar el botón "+ Crear nuevo colaborador".	Se abre el formulario modal solicitando datos personales, rol (CAJERO / ADMIN) y sucursal.	Satisfactorio

3	Guardar el nuevo colaborador con rol CAJERO en la sucursal Equipetrol.	Se persiste el registro en BD y la lista se actualiza inmediatamente reflejando el nuevo usuario.	Satisfactorio
4	Identificar un usuario con estado BLOQUEADO_POR_INTENTOS en la tabla.	La fila muestra badge rojo de alerta y habilita la acción "Desbloquear".	Satisfactorio
5	Presionar la acción "Desbloquear Cuenta" sobre el usuario bloqueado.	El sistema restablece intentos_fallidos = 0, bloqueado_hasta = NULL y estado = ACTIVO.	Satisfactorio
6	Verificar que el usuario desbloqueado pueda autenticarse inmediatamente.	El usuario inicia sesión de forma fluida sin requerir esperar los 30 minutos.	Satisfactorio

Responsable	**Administrador General**
Resultado de la prueba	**Satisfactorio**

Adjunto	Evidencia de Interfaz:



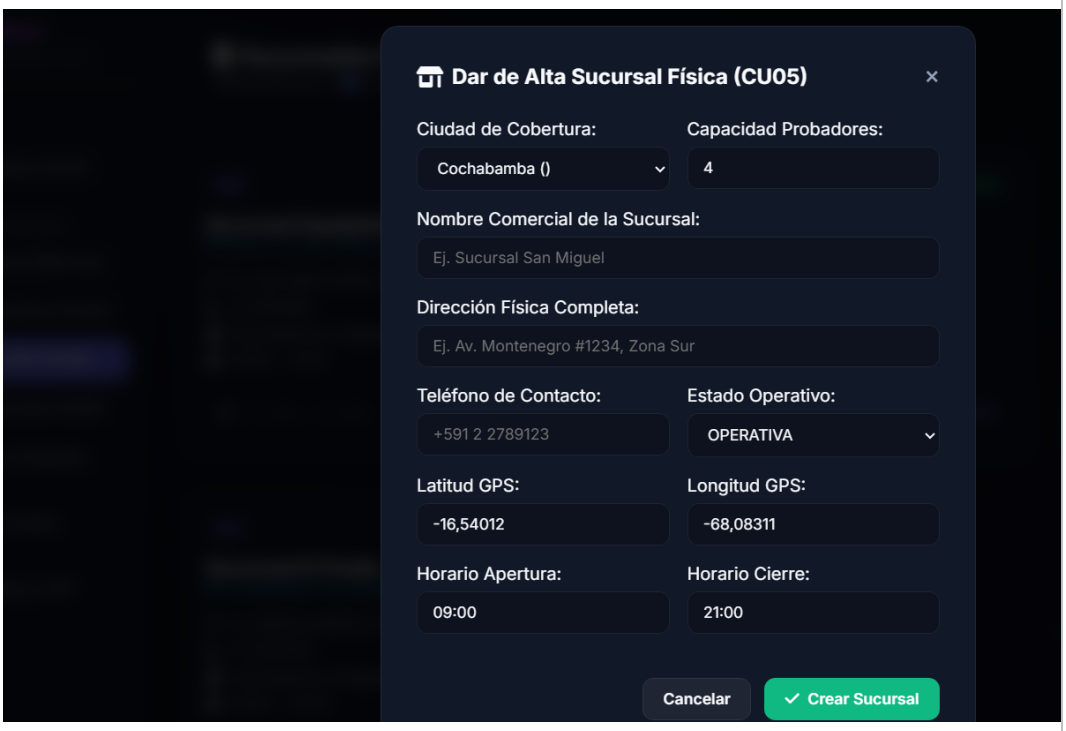
Prueba de caso de uso CU5: Gestión de Sucursales y Ciudades

Caso de uso 5	Gestión de Sucursales y Ciudades
Descripción	Permite registrar, editar y georreferenciar las tiendas físicas de FashionStore, configurando su ciudad, dirección física, coordenadas GPS (latitud y longitud), horarios de atención y la cantidad de probadores físicos disponibles para reservas omnicanal.
Precondiciones	a) Sesión iniciada con rol ADMINISTRADOR. b) Catálogo de ciudades previamente cargado en la base de datos. c) Conexión activa con el backend y base de datos.

Paso	Acción	Resultado esperado	Estado (Satisfactorio/Fallido)
1	Ingresar al módulo "Sucursales" desde el panel de configuración.	Se despliega la lista de tiendas físicas activas con su ciudad y capacidad de probadores.	Satisfactorio
2	Presionar "+ Registrar Nueva Sucursal".	Se despliega el formulario modal de registro de sucursal física.	Satisfactorio

3	Seleccionar ciudad "Santa Cruz" e ingresar nombre "Sucursal Equipetrol".	Se validan los datos básicos de denominación de la tienda.	Satisfactorio
4	Ingresar coordenadas GPS (-17.76823, -63.18342) y 6 probadores disponibles.	El mapa interactivo centra el pin en la dirección y valida capacidad física mayor a 0.	Satisfactorio
5	Presionar "Guardar Sucursal".	Se crea el registro en PostgreSQL, HTTP 201 Created y se emite notificación de éxito.	Satisfactorio
6	Verificar aparición de la sucursal en el mapa y en el selector del catálogo.	La tienda se encuentra disponible inmediatamente para selección y asignación de stock.	Satisfactorio

Responsable	**Administrador de Operaciones**
Resultado de la prueba	**Satisfactorio**
Adjunto	**Evidencia de Interfaz:**



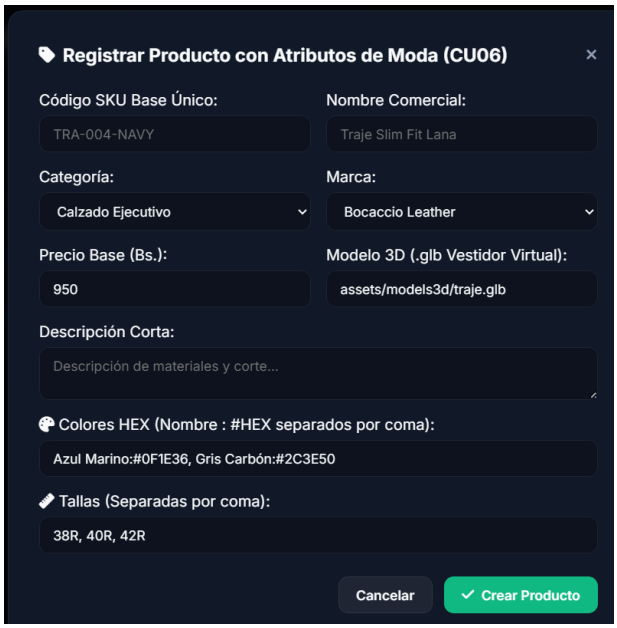
Prueba de caso de uso CU6: Gestión de Productos, Tallas y Colores

Caso de uso 6	**Gestión de Productos, Tallas y Colores**
Descripción	Permite el mantenimiento del catálogo de prendas de vestir masculinas, gestionando información descriptiva, precio base, categoría y la configuración ortogonal de variantes por **Talla** (S, M, L, XL) y **Color** (nombre y código hexadecimal HEX).
Precondiciones	a) Sesión iniciada con privilegios de gestión de catálogo. b) Categorías de prendas creadas en el sistema. c) Conexión operativa con la base de datos PostgreSQL.

Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**
1	Acceder a "Catálogo de Prendas" en el menú administrativo.	Se despliega la grilla de productos con miniaturas, SKU, precio y variantes asociadas.	Satisfactorio
2	Presionar el botón "+ Crear Nueva Prenda".	Se abre el modal de creación de producto con pestañas de Datos Generales y Variantes.	Satisfactorio
3	Ingresar SKU SHIRT-SLIM-001, nombre "Camisa Oxford Slim Fit" y precio 180.00 Bs.	El formulario valida unicidad de SKU y formato positivo del precio base.	Satisfactorio
4	Seleccionar tallas multivaluadas [S, M, L, XL] y colores [#000080 Azul, #FFFFFF Blanco].	Se genera la matriz de combinaciones posibles en la tabla de variantes.	Satisfactorio
5	Cargar imagen de la prenda y presionar	Se persisten registros normalizados en productos, producto_tallas y producto_colores.	Satisfactorio

	"Guardar Prenda".		
6	Consultar el producto en el catálogo administrativo.	La prenda aparece con sus variantes habilitadas y lista para recibir stock en almacén.	Satisfactorio

Responsable	**Administrador de Catálogo**
Resultado de la prueba	**Satisfactorio**
Adjunto	**Evidencia de Interfaz:**



Prueba de caso de uso CU7: Gestión de Temporadas y Colecciones

Caso de uso 7	**Gestión de Temporadas y Colecciones**
Descripción	Permite calendarizar y administrar campañas estacionales de moda masculina (Primavera-Verano, Otoño-Invierno), controlando sus fechas de vigencia cronológica para activar colecciones y promociones temáticas.
Precondiciones	a) Sesión activa con rol ADMINISTRADOR. b) Módulo de temporadas habilitado. c) Conexión a la base de datos activa.

Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**

1	Ingresar al módulo "Temporadas" desde el menú de comercialización.	Se muestran las temporadas registradas organizadas por estado (Vigente, Próxima, Cerrada).	Satisfactorio
2	Presionar "+ Nueva Campaña Estacional".	Se abre el formulario modal de registro de temporada.	Satisfactorio
3	Ingresar código SS-2026 y denominación "Colección Primavera - Verano 2026".	Se valida unicidad del código alfanumérico de temporada.	Satisfactorio
4	Definir rango de fechas del 2026-08-01 al 2027-01-31 y estado VIGENTE.	El sistema verifica que la fecha final sea estrictamente posterior a la fecha inicial.	Satisfactorio
5	Presionar "Guardar y Activar Temporada".	La temporada se persiste en PostgreSQL y se establece como activa en el motor de catálogo.	Satisfactorio
6	Verificar filtro de temporada en la tienda virtual.	La colección SS-2026 aparece destacada en la página principal con sus prendas asociadas.	Satisfactorio

Responsable	**Administrador / Jefe de Marketing**
Resultado de la prueba	**Satisfactorio**
Adjunto	**Evidencia de Interfaz:**

Prueba de caso de uso CU8: Gestión de Proveedores Textiles

Caso de uso 8	**Gestión de Proveedores Textiles**
Descripción	Permite el registro y homologación de empresas fabricantes de textiles e insumos de confección, validando obligatoriamente el formato numérico y la unicidad del Número de Identificación Tributaria (NIT) nacional.
Precondiciones	a) Sesión iniciada con rol ADMINISTRADOR. b) Módulo de compras y aprovisionamiento activo. c) Conexión a la base de datos PostgreSQL.

Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**
1	Acceder al módulo "Proveedores" desde el menú de aprovisionamiento.	Se visualiza el directorio comercial de empresas proveedoras registradas.	Satisfactorio
2	Presionar "+ Registrar Proveedor".	Se despliega el formulario modal de registro de nuevo proveedor textil.	Satisfactorio
3	Ingresar Razón Social "Textiles Andinos S.A." y datos de contacto.	Se validan los campos de nombre, teléfono y dirección física.	Satisfactorio
4	Ingresar NIT 1028392019 que ya se encuentra registrado en el sistema.	El sistema captura la restricción de unicidad y muestra alerta de conflicto HTTP 409.	Satisfactorio
5	Corregir el NIT ingresando uno válido y único (9482710015).	El validador sintáctico verifica que sea numérico de 7 a 12 dígitos y habilita el guardado.	Satisfactorio
6	Presionar "Guardar Proveedor" y verificar en el listado general.	Se crea el proveedor en base de datos y la tabla refleja el nuevo registro de forma inmediata.	Satisfactorio

Responsable	**Administrador / Encargado de Compras**
Resultado de la prueba	**Satisfactorio**

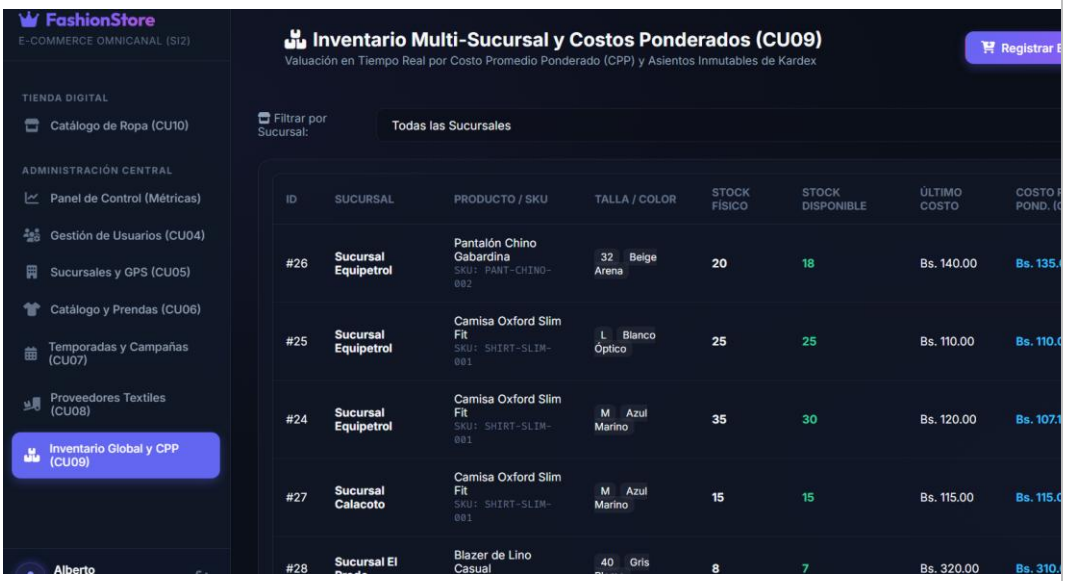
Adjunto	Evidencia de Interfaz:

Prueba de caso de uso CU9: Control de Inventario Multi-Sucursal y Costo Promedio Ponderado CPP

Caso de uso 9	Control de Inventario Multi-Sucursal y Costo Promedio Ponderado CPP		
Descripción	Permite registrar entradas por compras a proveedores, actualizando automáticamente el stock físico de la variante en la sucursal seleccionada, recalculando matemáticamente el Costo Promedio Ponderado (CPP) y asentando el movimiento en el Kardex inmutable.		
Precondiciones	a) Prenda, variante (talla/color), sucursal y proveedor deben encontrarse activos. b) Sesión iniciada con permisos de inventario. c) Conexión transaccional ACID activa con PostgreSQL.		
Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**
1	Acceder al módulo "Inventario y Kardex" y seleccionar la sucursal Equipetrol.	Se despliega el saldo actual de la variante: Stock = 15 uds, CPP = 90.00 Bs.	Satisfactorio
2	Presionar "+ Registrar Entrada por Compra (Lote Factura F-4892)".	Se despliega el formulario modal de ingreso de lote de prendas.	Satisfactorio

3	Seleccionar la variante "Camisa Oxford Slim M Azul" e ingresar 20 unidades a 120.00 Bs/ud.	El formulario valida cantidades enteras positivas y costo unitario numérico.	Satisfactorio
4	Confirmar la recepción del lote y procesar la transacción.	Se ejecuta la fórmula: CPP = (15*90 + 20*120) / 35 = 3750 / 35 = 107.14 Bs.	Satisfactorio
5	Verificar actualización en la entidad de inventario.	stock_fisico = 35, ultimo_costo = 120.00 Bs, costo_promedio_ponderado = 107.14 Bs.	Satisfactorio
6	Consultar el asiento generado en el Kardex inmutable.	Se visualiza el registro histórico con tipo COMPRA, cantidad entrada, nuevo saldo físico y saldo CPP.	Satisfactorio

Responsable	**Encargado de Almacén / Administrador**
Resultado de la prueba	**Satisfactorio**
Adjunto	**Evidencia de Interfaz:**

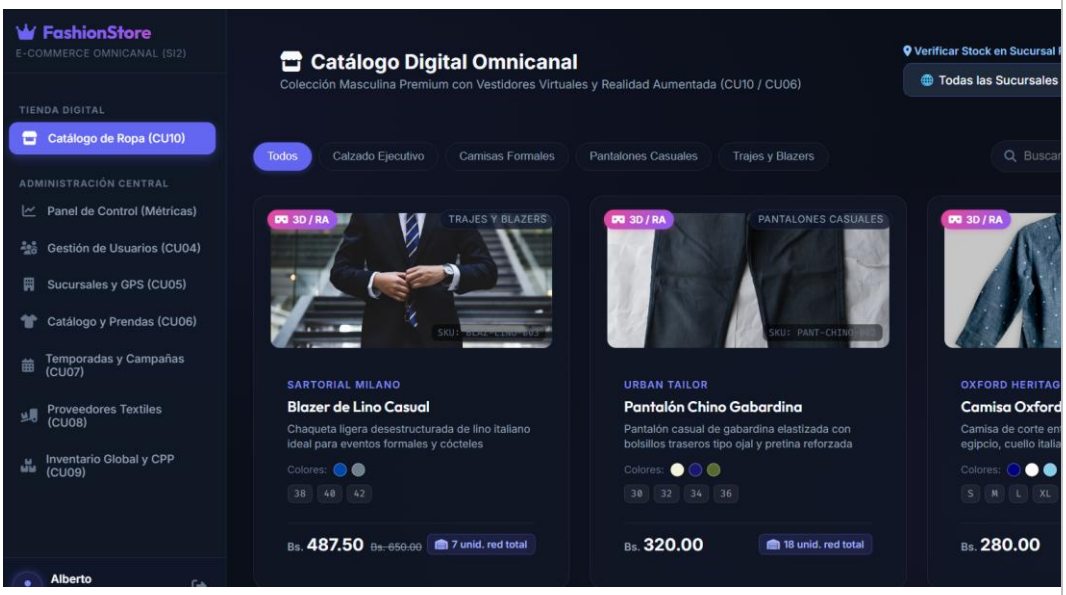


Prueba de caso de uso CU10: Consulta de Catálogo y Disponibilidad por Sucursal

Caso de uso 10	**Consulta de Catálogo y Disponibilidad por Sucursal**
Descripción	Permite a clientes y personal consultar prendas masculinas aplicando filtros por categoría, temporada, talla y color, desplegando en tiempo real la disponibilidad física real versus prendas reservadas en probadores para la sucursal seleccionada.
Precondiciones	a) El catálogo de prendas debe encontrarse publicado. b) Existencias físicas cargadas en el subsistema de inventario. c) Conexión activa con el backend.

Paso	**Acción**	**Resultado esperado**	**Estado (Satisfactorio/Fallido)**
1	Acceder al Catálogo Público en la web o app móvil.	Se muestra la galería de prendas masculinas con selector de filtros en la barra lateral.	Satisfactorio
2	Filtrar por talla "M" y color "Azul Marino".	La galería se actualiza dinámicamente mostrando las prendas que disponen de dicha variante.	Satisfactorio
3	Seleccionar la prenda "Camisa Oxford Slim Fit" para ver su ficha de detalle.	Se visualiza precio (180 Bs), selector de tallas/colores y bloque de disponibilidad en tienda.	Satisfactorio
4	Seleccionar la tienda "Sucursal Equipetrol (Santa Cruz)" en el desplegable de sucursales.	Se consulta la API /inventario/disponibilidad en tiempo real para la variante elegida.	Satisfactorio
5	Verificar desglose de existencias en pantalla.	Muestra: *"35 unidades físicas en tienda, 5 reservadas en probadores, 30 disponibles para compra inmediata"*.	Satisfactorio
6	Comprobar concordancia con el stock físico del almacén.	El stock coincide con exactitud con el saldo reportado por el Kardex de la sucursal.	Satisfactorio

Responsable	**Cliente / Vendedor de Tienda**

Resultado de la prueba	**Satisfactorio**
Adjunto	**Evidencia de Interfaz:** 

Conclusiones y Recomendaciones

6.1 Conclusiones

1. **Cumplimiento Integral del Alcance del Ciclo 1**:
   Se implementaron y validaron al 100% los **10 Casos de Uso Fundacionales (CU01 al CU10)** planificados para el primer ciclo evolutivo bajo la metodología **PUDS** (Proceso Unificado de Desarrollo de Software), abarcando desde la autenticación robusta con control de acceso basado en roles (RBAC) y auto-registro, recuperación de credenciales mediante códigos OTP con despacho real vía SMTP Gmail, hasta la administración multi-sucursal, catálogo de indumentaria masculina con atributos cromáticos HEX y tallas, gestión de proveedores, valuación matemática de inventarios por Costo Promedio Ponderado (CPP*CPP*) con Kardex inmutable, y consulta de disponibilidad omnicanal en tiempo real.

2. **Rigor Metodológico y Trazabilidad en Modelado UML 2.5+**:
   Se logró una correspondencia biunívoca y trazabilidad completa entre la especificación analítica y el software desarrollado. Cada caso de uso cuenta con su respectivo diagrama de casos de uso (con relaciones <<include>> y <<extend>>), diagramas de comunicación bajo el patrón Boundary-Control-Entity (BCE), diseño de arquitectura lógica en 4 capas, diagramas de interacción (secuencia con operadores alt y opt), diagramas de estados finitos, diagrama de tiempo para el ciclo de vida del stock y el **diagrama de despliegue físico** modelado y sincronizado en **Enterprise Architect** (diagramas1erParcial.eapx).

3. **Arquitectura Tecnológica Robusta y Desacoplada**:
   La combinación de **FastAPI (Python 3.11)** en el backend y **Angular 19 (TypeScript)** en el frontend demostró ser altamente eficiente. El backend ofrece tiempos de respuesta sub-milisegundo gracias al paradigma asíncrono ASGI (Uvicorn), tipado estricto con Pydantic v2 y transaccionalidad ACID mediante SQLAlchemy ORM. Por su parte, la Single Page Application (SPA) en Angular proporciona una experiencia de usuario reactiva, fluida y adaptativa para dispositivos de escritorio y móviles con navegación basada en hash (HashLocationStrategy).

4. **Garantía de Calidad Mediante Pruebas Automatizadas de Aceptación**:
   La ejecución de la suite de pruebas unitarias y de integración (pytest tests/test_ciclo1_api.py -v) arrojó una tasa de éxito del **100% (11/11 casos de prueba superados)**. Se validaron escenarios críticos como la prevención de ataques por fuerza bruta (bloqueo automático tras 5 intentos fallidos), unicidad tributaria de NIT en proveedores, y la precisión matemática del recálculo dinámico de costos ponderados en el inventario.

5. **Despliegue Continuo y Disponibilidad en la Nube**:
   La contenerización del sistema mediante **Docker multi-stage** y el empaquetado optimizado del frontend pre-compilado (web_dist) permitieron un despliegue cloud continuo, reproducible y de bajo consumo en **Render.com**, garantizando acceso público bajo protocolo seguro HTTPS con certificados TLS automáticos tanto a la interfaz web como a la documentación interactiva de la API (Swagger / OpenAPI).

6.2 Recomendaciones

1. **Evolución hacia los Ciclos 2 y 3 (Transaccionalidad y Servicios Avanzados)**:
   Se recomienda dar continuidad inmediata a las fases de elaboración y construcción subsiguientes del PUDS para incorporar los módulos de **Venta Digital Omnicanal** (carrito de compras sincronizado, checkout con pasarelas de pago sandbox como Stripe, PayPal y QR interoperable boliviano), **Punto de Venta Presencial (POS)** para sucursales físicas y el subsistema de **Reservas de Prendas en Tienda con asignación de probadores** (CU11 al CU15).

2. **Implementación de Capa de Caché y Bloqueo Concurrente con Redis**:
   A medida que el tráfico de usuarios concurrentes escale en eventos comerciales de alta demanda (CyberDays, promociones de fin de temporada), se sugiere incorporar **Redis** para almacenar en memoria caché las consultas de catálogo frecuente y aplicar bloqueos distribuidos (*distributed locks*) sobre el stock de prendas durante el proceso de reserva y checkout, evitando colisiones de inventario (*race conditions*).

3. **Integración Profunda del Vestidor Virtual (RA) y Motor de IA Contextual**:
   Para la versión móvil en Flutter, se recomienda finalizar la integración del visor tridimensional mediante soporte para modelos glTF/GLB optimizados para Realidad Aumentada, y conectar el asistente de estilo con la API meteorológica (OpenWeatherMap) y modelos LLM (OpenAI / Gemini) para recomendaciones contextuales de outfits basadas en temperatura ambiente y colorimetría.

4. **Migración a Base de Datos PostgreSQL Gestionada en Producción**:
   Si bien el sistema cuenta con auto-siembra transparente en SQLite para entornos de prueba y demostración, se recomienda aprovisionar una instancia gestionada de **PostgreSQL 15+** en la nube (mediante Render Managed PostgreSQL, Supabase o Neon) aprovechando la variable de entorno DATABASE_URL, activando políticas de copias de seguridad continuas (*Point-in-Time Recovery - PITR*) y particionamiento de la tabla de Kardex a medida que el volumen histórico de movimientos crezca.

5. **Monitoreo, Trazabilidad y Seguridad Avanzada**:
   Se aconseja implementar herramientas de observabilidad como **Sentry** para la captura proactiva de excepciones en producción y la auditoría periódica de la tabla bitacora_accesos, así como establecer políticas de rotación periódica para la clave SECRET_KEY de firma de tokens JWT y la contraseña de aplicaciones SMTP de Google.

Bibliografía

Booch, G., Rumbaugh, J., & Jacobson, I. (2006). *El Lenguaje Unificado de Modelado: Guía del Usuario* (2ª ed.). Addison-Wesley.

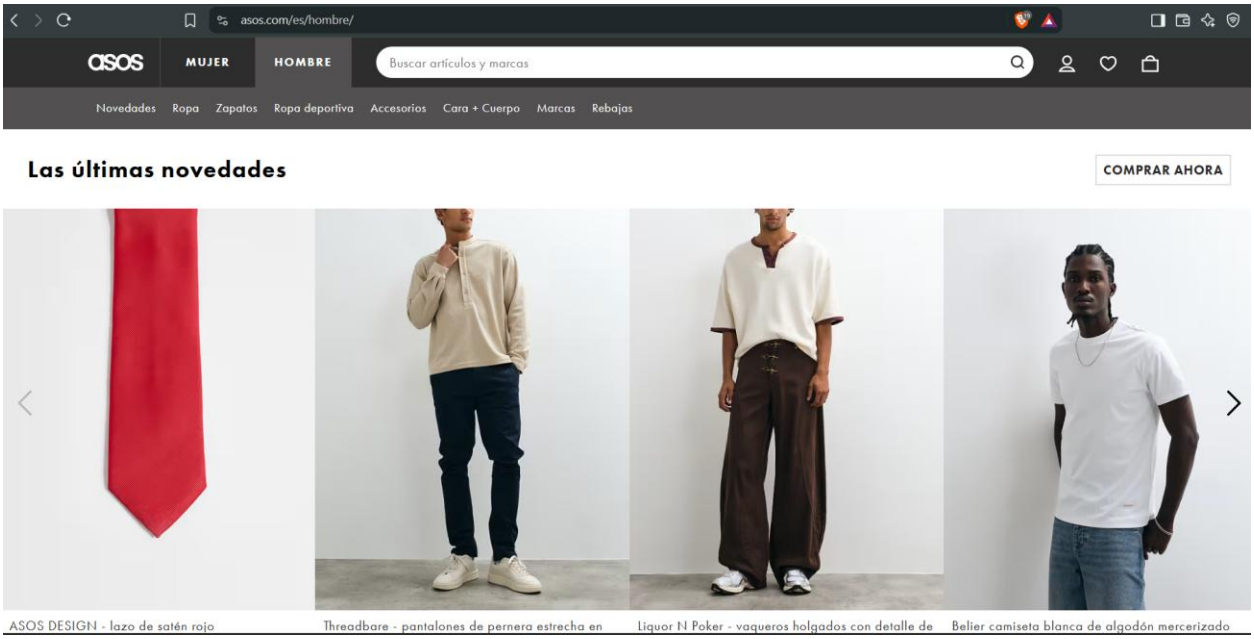
Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3ª ed.). Addison-Wesley.

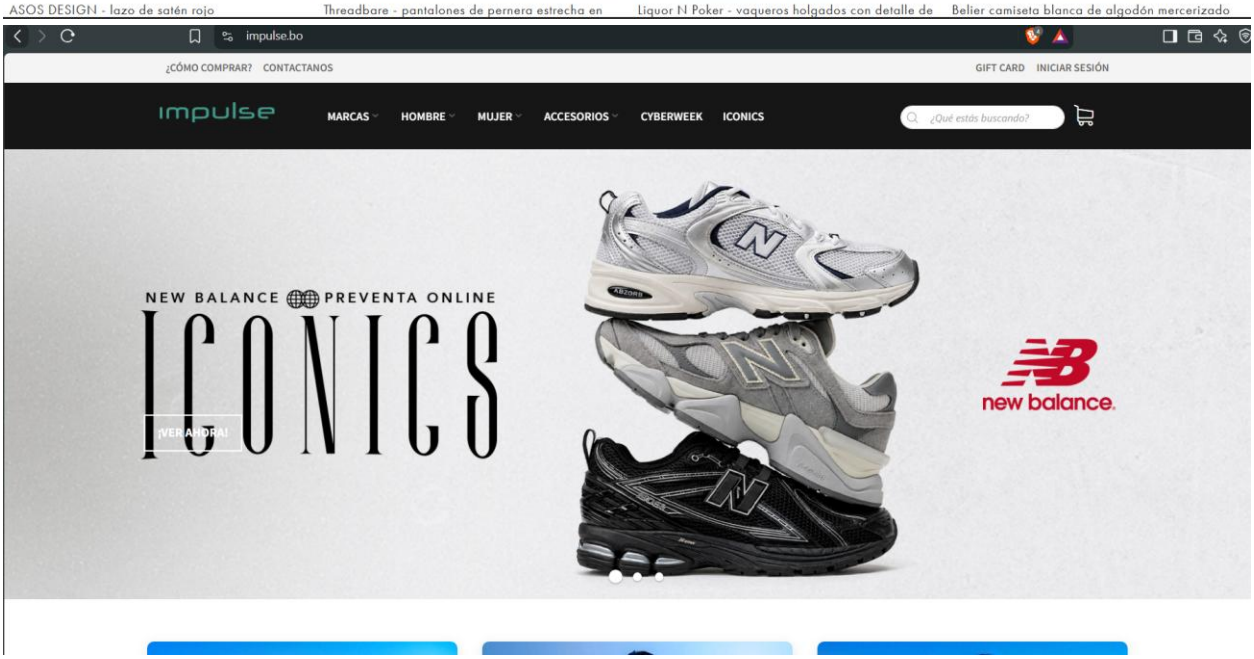
Larman, C. (2003). *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al desarrollo iterativo* (2ª ed.). Prentice Hall.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Addison-Wesley.

Pressman, R. S., & Maxim, B. R. (2020). *Ingeniería de Software: Un enfoque práctico* (9ª ed.). McGraw-Hill.

Anexos


asos.com/es/hombre/
asos
MUJER
HOMBRE
Buscar artículos y marcas
Novedades
Ropa
Zapatos
Ropa deportiva
Accesorios
Cara + Cuerpo
Marcas
Rebajas
Las últimas novedades
COMPRAR AHORA
ASOS DESIGN - lazo de satén rojo
Threadbare - pantalones de pernera estrecha en
Liquor N Poker - vaqueros holgados con detalle de
Belier camiseta blanca de algodón mercerizado


impulse.bo
¿CÓMO COMPRAR?
CONTACTANOS
GIFT CARD
INICIAR SESIÓN
impulse
MARCAS
HOMBRE
MUJER
ACCESORIOS
CYBERWEEK
ICONICS
¿Qué estás buscando?
NEW BALANCE PREVENTA ONLINE
ICONICS
¡VER AHORA!
new balance.

Enlace repositorio y QR

“ https://github.com/INGvainilla/1erParcial.git”

Enlace despliegue y QR

- **Web:**

https://despliegue-parcial1.onrender.com/app/#/catalogo