

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO
FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y
TELECOMUNICACIONES



Sistema Inteligente de Gestión Comercial y Reportes Dinámicos

Materia: Sistemas de Información II

Sigla: INF 412

Docente: MSc. Ing. Angelica Garzon Cuellar

Integrantes:

- Chumacero Miranda Nelson
- Veizaga Ramirez Deivy Brahian



Semestre II/2025

Santa Cruz de la Sierra – Bolivia

Contenido

PERFIL.....	5
1.1 Introducción.....	5
1.2 Objetivo General.....	5
1.3 Objetivos Específicos.....	5
1.4 Descripción del problema.....	7
1.5 Alcance.....	7
1.5.1 Módulo de Usuarios y Roles.....	7
1.5.2 Catálogo de Productos.....	8
1.5.3 Clientes.....	8
1.5.4 Carrito, Ventas y Pagos (Sandbox).....	8
1.5.5 Comprobantes y Histórico.....	8
1.5.6 Reportes Dinámicos por Texto/Voz.....	9
1.5.7 Analítica y Predicción de Ventas.....	9
1.5.8 Visualización Web (React PWA).....	9
1.5.9 Aplicación Móvil (Flutter).....	9
1.5.10 Notificaciones y Alertas.....	9
1.5.11 Seguridad, Auditoría y Cumplimiento.....	10
1.5.12 Datos y Base de Datos (PostgreSQL).....	10
1.5.13 Despliegue en la Nube y Operación.....	10
Marco Teórico.....	10
2 PROCESO DE DESARROLLO.....	11
a) Seguir los pasos descritos por el PUDS para el desarrollo de la Aplicación.....	11
b) Generar los modelos pertinentes utilizando UML.....	12
2.1 Flujo de Trabajo: Captura de Requisitos.....	13
2.1.1 Identificar Actores y Casos de Uso.....	13
2.1.1.1 Actores.....	13
2.1.1.2 Casos de uso.....	13
2.1.2 Priorizar Casos de Uso.....	14
Ciclo #1.....	15
Ciclo #2.....	16
Ciclo #3.....	16
2.1.3 Detalles de un Caso de Uso.....	17
Ciclo #1.....	17
CU01 Iniciar sesión.....	17
CU02 Cerrar sesión.....	18
CU03 Gestionar roles y permisos.....	19
CU04 Registrar producto.....	19
CU05 Editar producto electrodoméstico.....	20
CU06 Registrar cliente.....	21
CU07 Editar cliente.....	22
Ciclo #2.....	22
CU08 Listar productos.....	22
CU09 Listar clientes.....	23

CU10 Bitácora.....	24
CU11 Registrar venta simple.....	25
CU12 Generar comprobante PDF.....	25
CU13 Ver dashboard con predicción.....	26
Ciclo #3.....	27
CU14 Registrar garantía.....	27
CU15 Registrar reclamo de garantía.....	28
CU16 Consultar estado de garantía/reclamo.....	29
CU17 Gestionar ofertas.....	29
CU18 Notificaciones push.....	30
CU19 Búsqueda por voz (producto/cliente).....	31
CU20 Reportes por voz (prompts).....	32
2.1.4 Prototipo de Interfaz de Usuario.....	33
Ciclo #1.....	33
Ciclo #2.....	36
Ciclo #3.....	39
2.1.5 Estructurar el Modelo de Caso de Uso.....	46
Ciclo #1.....	46
Ciclo #2.....	47
Ciclo #3.....	48
2.2 Flujo de Trabajo: Análisis.....	49
2.2.1 Análisis de Arquitectura.....	49
2.2.1.1 Identificar Paquetes.....	49
2.2.1.2 Relacionar Paquetes y Casos de Uso.....	50
2.2.1.3 Vista de Paquetes.....	51
2.2.2 Análisis de Caso de Uso.....	53
Ciclo #1.....	53
Ciclo #2.....	54
Ciclo #3.....	56
2.2.3 Análisis de una Clase.....	58
Ciclo #1.....	58
CU1 Iniciar sesión.....	58
CU2 Cerrar sesión.....	58
CU3 Gestionar roles y permisos.....	58
Ciclo #2.....	59
Ciclo #3.....	60
2.2.4 Analizar paquete.....	62
2.3 Flujo de Trabajo: Diseño.....	63
2.3.1 Diseño de Arquitectura.....	63
2.3.1.1 Diseño físico (Diagrama de despliegue).....	63
2.3.1.2 Diseño Lógico (Diagrama de Paquete).....	64
2.3.2 Diagrama de Secuencia.....	64
Ciclo #1.....	64
Ciclo #2.....	68

Ciclo #3.....	71
2.3.3 Diseño de Datos.....	74
2.3.3.1 Diseño de Datos Lógico.....	75
● Diagrama de Clase.....	75
● Mapeo.....	75
● Normalización.....	76
2.3.3.2) Diseño de Datos Físico.....	77
● Tabla de Volumen.....	77
● Diagrama Relacional.....	86
● Scrip.....	86
2.4 Flujo de Trabajo: Implementación.....	96
2.4.1 Elección de plataforma de desarrollo de Software.....	96
2.4.1.1 Implementación de la arquitectura del Sistema.....	96
2.4.1.1.1 Diagrama de componente principal del sistema.....	96
2.4.1.2 Implementación de la arquitectura del subsistema.....	97
2.4.1.2.1 Diagrama de componente de cada paquete.....	97
2.5 Flujo de Trabajo: Pruebas.....	102
Manual de Usuario.....	104
CONCLUSIÓN.....	108
RECOMENDACIÓN.....	109
BIBLIOGRAFÍA.....	109
Anexos.....	110
- GitHub:.....	110
- Azure (nube):.....	110

PERFIL

1.1 Introducción

SmartSales365 es un sistema de gestión comercial orientado a optimizar la toma de decisiones diarias en ventas. Integra funcionalidades core (gestión de productos con stock, clientes, ventas y emisión de comprobantes) con capacidades de análisis y reporte a demanda mediante prompts de texto o voz, reduciendo tiempos de obtención de información y mejorando la visibilidad del negocio.

Como diferenciador, incorpora un dashboard de analítica predictiva basado en RandomForestRegressor (scikit-learn) entrenado con históricos de ventas para estimar tendencias por período o categoría. El frontend web se implementa como PWA en React; el backend expone una API REST en Django REST Framework sobre PostgreSQL, responsable de interpretar prompts, construir consultas dinámicas y generar reportes (pantalla, PDF, Excel). La app móvil Flutter prioriza flujos de alto impacto (consultas rápidas y notificaciones push) aprovechando capacidades nativas.

El proyecto se desarrolla bajo el Proceso Unificado (UML 2.5+), con entregables de arquitectura, modelos, descripción del componente de IA y manual de usuario. El despliegue objetivo es en la nube (Azure/AWS/GCP) mediante contenedores, facilitando escalabilidad horizontal, versionado y CI/CD.

1.2 Objetivo General

Diseñar, construir, desplegar y documentar SmartSales365, una plataforma full-stack para gestión comercial y generación de reportes bajo demanda (texto/voz), con dashboard predictivo basado en RandomForestRegressor, que reduzca $\geq 50\%$ el tiempo de obtención de reportes operativos y alcance MAPE $\leq 20\%$ en predicción mensual de ventas, siguiendo PUDS + UML 2.5+ y publicando una demo en nube (Azure/AWS/GCP) con contenedores.

1.3 Objetivos Específicos

- Gestión Comercial Base**

Implementar CRUD de productos (categorías, precios, stock con alertas), clientes y

ventas, comprobantes y un histórico con filtros por fecha/cliente/producto; carrito con texto o voz; pagos sandbox (Stripe/PayPal).

Criterio de aceptación: flujos E2E operativos, manejo de stock en tiempo real y comprobante descargable.

- **Motor de Reportes por Texto/Voz**

Construir un intérprete de prompts (reglas/regex/parser simple) que genere consultas dinámicas (ORM/SQL) y exporte pantalla/PDF/Excel.

Criterio: entregar en ≤5 s reportes como “ventas de septiembre por producto (PDF)” y “01/10/2024–01/01/2025 por cliente (Excel)”.

- **Analítica Predictiva (ML)**

Desarrollar pipeline con RandomForestRegressor (scikit-learn): features temporales, entrenamiento periódico, serialización (joblib) y endpoint de predicción para el dashboard.

Criterio: MAPE ≤20% con datos sintéticos iniciales; re-entrenos programados; versión del modelo y fecha de último entrenamiento visibles.

- **Frontend Web PWA (React)**

Implementar PWA responsive con Recharts/Chart.js para históricos y predicciones, caché básico offline para vistas de reportes, y consumo de API con Swagger como contrato.

Criterio: Lighthouse ≥85 en Performance/Best-Practices/Accessibility.

- **Aplicación Móvil (Flutter)**

Entregar app con reportes resumidos, acciones rápidas (p. ej., re-generar último reporte) y notificaciones push (firebase_messaging).

Criterio: recepción de push ante nuevos reportes y visualización en ≤3 toques.

- **API REST + Documentación**

Exponer endpoints en Django REST Framework con Swagger/OpenAPI autogenerado y ejemplos de uso para texto/voz, reportes y predicciones.

Criterio: catálogo completo de endpoints con ejemplos reproducibles.

- **Base de Datos y Datos Semilla**

Diseñar esquema en PostgreSQL (normalización básica + índices en campos de consulta) y script de datos sintéticos para pruebas/ML.

Criterio: consultas clave en ≤ 300 ms con 50k registros de ventas (entorno de demo).

- **Seguridad y Control de Acceso**

Autenticación JWT, CORS y CSRF (cuando aplique), validación de entradas, roles/permisos mínimos (admin/analista/vendedor) y registro de eventos (bitácora). Criterio: endpoints críticos protegidos y pruebas de autorización básicas aprobadas.

- **Calidad y Pruebas**

Pruebas unitarias e integración en backend; pruebas de carga a 100 RPS para reportes simples; linters en frontend/backend.

Criterio: cobertura $\geq 60\%$ backend; tiempos ≤ 5 s para reportes estándar bajo 100 RPS en entorno de demo.

- **Despliegue en la Nube**

Contenerización (Docker), compose o manifiestos equivalentes, y despliegue de demo en Azure/AWS/GCP.

Criterio: URL pública operativa, variables de entorno seguras y logs centralizados básicos.

1.4 Descripción del problema

En las pymes, la información comercial suele estar fragmentada entre múltiples sistemas (ERP básicos, planillas y pasarelas de pago), lo que genera reporteo rígido, demoras para responder preguntas ad-hoc y baja predictibilidad de la demanda. Las herramientas tradicionales obligan a usar reportes predefinidos y consultas técnicas (SQL), de modo que las decisiones sobre precios, stock y campañas se toman con datos tardíos o incompletos.

SmartSales365 aborda este problema al: (a) permitir prompts en lenguaje natural (texto/voz) para consultar, filtrar y exportar reportes sin conocimiento técnico; (b) incorporar predicción de ventas (RandomForestRegressor) para planificación de inventario y marketing; y (c) ofrecer app móvil para insights rápidos y notificaciones operativas (p. ej., quiebre de stock, desvíos de metas).

1.5 Alcance

1.5.1 Módulo de Usuarios y Roles

Objetivo. Controlar el acceso y operaciones según rol.

Datos principales. ID Usuario; Nombre; Email; Teléfono; Contraseña (hash); Rol (Admin/Analista/Vendedor); Estado; Fecha de alta/último acceso.

Criterios de aceptación. Autenticación JWT, autorización por rol en endpoints críticos, bitácora de login/logout/operaciones sensibles, recuperación de contraseña.

1.5.2 Catálogo de Productos

Objetivo. Administrar el portafolio de venta.

Datos principales. ID Producto; Nombre; Categoría; Precio; Stock; SKU; Descripción; Estado; Imágenes.

Criterios. CRUD completo; validación de stock; búsqueda y filtros por categoría/estado; alerta por umbral bajo de stock.

1.5.3 Clientes

Objetivo. Gestionar la base de clientes y su historial.

Datos principales. ID Cliente; Nombre/Razón social; CI/NIT; Email; Teléfono; Dirección; Fecha de registro; Estado.

Criterios. CRUD; historial de compras; segmentación básica (frecuencia, ticket promedio).

1.5.4 Carrito, Ventas y Pagos (Sandbox)

Objetivo. Registrar ventas con carrito y pagos en línea en modo sandbox (Stripe/PayPal).

Datos principales. ID Carrito; Ítems (Producto, Cantidad, Precio); Subtotal; Impuestos; Total; Método de pago; Referencia de pago; Estado (Pendiente/Pagado/Fallido).

Criterios. Flujo E2E: agregar ítems → confirmar → pago sandbox → actualización de stock → emisión de comprobante.

1.5.5 Comprobantes y Histórico

Objetivo. Emitir y conservar comprobantes y consultas históricas.

Datos principales. ID Venta; Cliente; Ítems; Totales; Fecha; Vendedor; Medio de pago; Enlace a comprobante (PDF).

Criterios. Generación de **PDF** descargable; filtros por fecha/cliente/producto; exportación a **Excel**.

1.5.6 Reportes Dinámicos por Texto/Voz

Objetivo. Generar reportes a demanda vía prompts (texto/voz).

Datos principales. Prompt; Intención/Parámetros parseados; Tipo de salida (Pantalla/PDF/Excel); Fecha; Usuario.

Criterios. Intérprete basado en reglas/regex; construcción dinámica ORM/SQL; ejemplos mínimos del enunciado resueltos en ≤ 5 s; export PDF/Excel.

1.5.7 Analítica y Predicción de Ventas

Objetivo. Mostrar ventas históricas y predicciones (RF Regressor).

Datos principales. Serie histórica (fecha, categoría, total); Features temporales; Modelo serializado; Predicciones; Métricas (MAPE).

Criterios. Entrenamiento con datos históricos/sintéticos; serialización (joblib); dashboard con series y forecast; MAPE objetivo $\leq 20\%$ (dataset demo); fecha/versión del modelo visibles.

1.5.8 Visualización Web (React PWA)

Objetivo. UI responsive con gráficas y experiencia PWA.

Datos principales. Vistas: Productos, Clientes, Ventas, Reportes, Dashboard.

Criterios. Recharts/Chart.js; caché básico de vistas de reportes; Lighthouse ≥ 85 (Performance/Best Practices/Accessibility).

1.5.9 Aplicación Móvil (Flutter)

Objetivo. Acceso rápido a reportes y **notificaciones push**.

Datos principales. Lista de reportes; Favoritos; Suscripciones a alertas; Token de dispositivo.

Criterios. Ver reportes resumidos; push por eventos (p. ej., quiebre de stock, nuevo reporte); acceder a un reporte en ≤ 3 toques.

1.5.10 Notificaciones y Alertas

Objetivo. Informar eventos relevantes en tiempo real.

Datos principales. ID Notificación; Tipo (Aviso/Recordatorio/Alerta); Canal (Push/Email); Destinatario; Contenido; Estado (Enviado/Leído); Fecha/Hora.

Criterios. Envío de notificaciones por stock bajo, ventas superiores a meta, nuevo reporte listo.

1.5.11 Seguridad, Auditoría y Cumplimiento

Objetivo. Proteger datos y registrar eventos.

Datos principales. Bitácora: Usuario; Acción; Endpoint; Timestamp; IP; Resultado.

Criterios. Validaciones de entrada; CORS; CSRF (cuando aplique); roles/permisos; registros de cambios críticos (productos, precios, roles).

1.5.12 Datos y Base de Datos (PostgreSQL)

Objetivo. Esquema normalizado e índices de rendimiento.

Datos principales. Tablas de productos/clientes/ventas/pagos/comprobantes/reportes/bitácora.

Criterios. Scripts de datos sintéticos; consultas clave ≤ 300 ms con 50k ventas en entorno demo.

1.5.13 Despliegue en la Nube y Operación

Objetivo. Demo pública y operativa en Azure/AWS/GCP.

Datos principales. Dockerfiles/Compose; Variables de entorno; URL de demo; Logs básicos.

Criterios. Contenerización; despliegue exitoso; Swagger público de lectura; logs accesibles; backup básico de BD.

Marco Teórico

PUDS: Proceso iterativo-incremental, dirigido por casos de uso y arquitectura; promueve trazabilidad desde requisitos hasta pruebas.

UML 2.5+: Lenguaje estándar de modelado (Casos de Uso, Clases, Secuencia, Componentes, Despliegue) para comunicar estructura y comportamiento.

DRF: Framework para APIs REST en Python; serializadores, viewsets, permisos y browsable API.

React PWA: UI reactiva, instalable, con capacidades offline limitadas mediante Service Worker.

Flutter: SDK multiplataforma para interfaces nativas; paquetes: speech_to_text, firebase_messaging.

PostgreSQL: SGBD relacional; soporte para índices, vistas y JSON.

RandomForestRegressor: Ensamble de árboles para regresión; robusto a outliers y relaciones no lineales; se serializa con joblib.

2 PROCESO DE DESARROLLO

a) Seguir los pasos descritos por el PUDS para el desarrollo de la Aplicación

Fase 1 — Inicio

Objetivo: acordar alcance y riesgos.

Actividades: identificar actores y catálogo CU001–CU022, priorizar en 3 ciclos.

Entregables: visión 1 pág, matriz Estado/Prioridad/Riesgo, plan de iteraciones.

Fase 2 — Elaboración

Objetivo: estabilizar requisitos clave y arquitectura.

Actividades: modelo de dominio y paquetes; definir API (DRF); prototipos UI; docker-compose.yml base.

Entregables: diagramas (paquetes, dominio), OpenAPI/Swagger inicial, plan de pruebas.

Fase 3 — Construcción

Objetivo: implementar por incrementos y probar.

Actividades: desarrollo por CU, tests unit/integración/E2E, demo al cierre de cada ciclo.

Entregables por ciclo: código funcional, Swagger actualizado, colección Postman, evidencias.

Fase 4 — Transición

Objetivo: preparar demo académica.

Actividades: pruebas de sistema/aceptación, carga de datos de demo, guía de despliegue.

Entregables: build con Docker Compose (api, web, db, ai-service, push), manual rápido, reporte de pruebas.

Iteraciones (3 ciclos)

Ciclo 1: CU001–CU006 (acceso y administración básica).

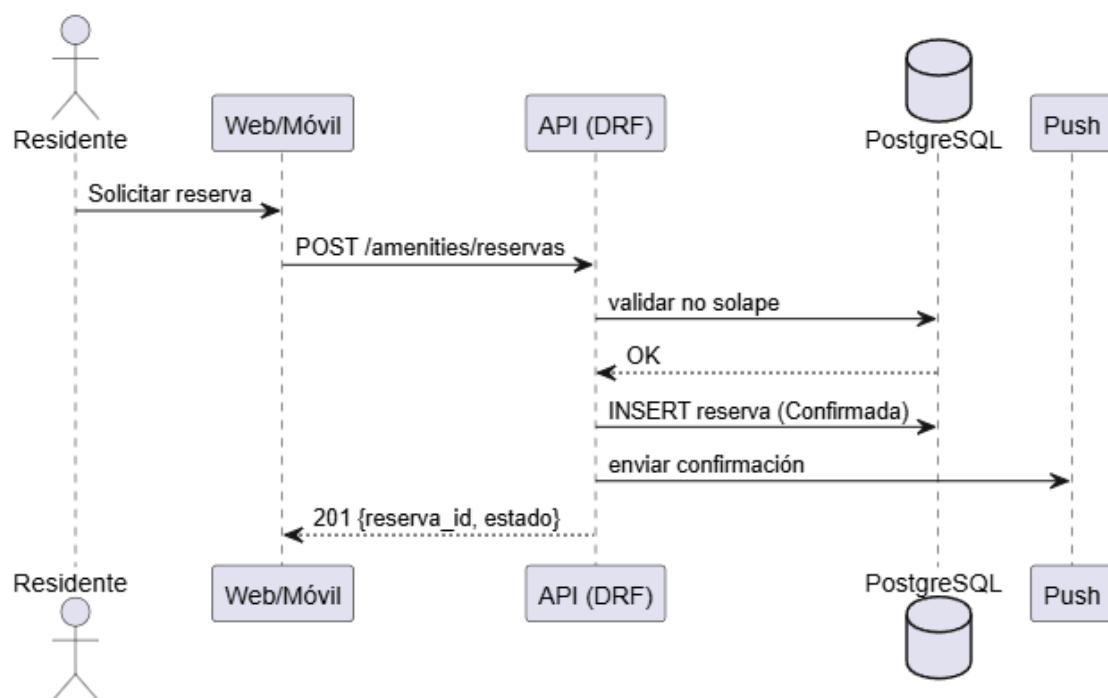
Ciclo 2: CU007–CU012 y CU016 (finanzas, reservas, notificaciones, esqueleto ai-service).

Ciclo 3: CU013–CU015 y CU017–CU022 (multas, comunicación, seguridad, reportes, inventario, seguridad+IA).

Prácticas transversales (resumen)

Git (feature→PR), DoD por CU (código+tests+Swagger), trazabilidad CU→commit→prueba, CI simple (pytest, npm test, newman).

b) Generar los modelos pertinentes utilizando UML



2.1 Flujo de Trabajo: Captura de Requisitos

2.1.1 Identificar Actores y Casos de Uso

2.1.1.1 *Actores*

Administrador (Web): administra catálogo, ofertas, garantías, reclamos, reportes y dashboard.

Cliente (Móvil): visualiza ofertas, descarga reportes propios, realiza búsquedas por voz de productos/ordenes, recibe push.

2.1.1.2 *Casos de uso*

- CU01 Iniciar sesión
- CU02 Cerrar sesión
- CU03 Gestionar roles y permisos
- CU04 Registrar producto
- CU05 Editar producto electrodoméstico
- CU06 Registrar cliente
- CU07 Editar cliente
- CU08 Listar productos
- CU09 Listar clientes
- CU10 Bitácora
- CU11 Registrar venta simple
- CU12 Generar comprobante PDF
- CU13 Ver dashboard con predicción
- CU14 Registrar garantía
- CU15 Registrar reclamo de garantía
- CU16 Consultar estado de garantía/reclamo
- CU17 Gestionar ofertas
- CU18 Notificaciones push
- CU19 Búsqueda por voz (producto/cliente)
- CU20 Reportes por voz (prompts)

2.1.2 Priorizar Casos de Uso

NRO	CASO DE USO	PRIORIDAD	ACTORES	CICLO
CU01	Iniciar sesión	Alta	Administrador, Cliente	Ciclo #1
CU02	Cerrar sesión	Alta	Administrador, Cliente	Ciclo #1
CU03	Gestionar roles y permisos	Alta	Administrador	Ciclo #1
CU04	Registrar producto	Alta	Administrador	Ciclo #1
CU05	Editar producto electrodoméstico	Alta	Administrador	Ciclo #1
CU06	Registrar cliente	Alta	Administrador	Ciclo #1
CU07	Editar cliente	Alta	Administrador	Ciclo #1
CU08	Listar productos	Alta	Administrador	Ciclo #2
CU09	Listar clientes	Alta	Administrador	Ciclo #2
CU10	Bitácora	Alta	Administrador	Ciclo #2
CU11	Registrar venta simple	Alta	Administrador	Ciclo #2
CU12	Generar comprobante PDF	Media	Administrador	Ciclo #2
CU13	Ver dashboard con predicción	Alta	Administrador	Ciclo #2
CU14	Registrar garantía	Media	Administrador	Ciclo #3

CU15	Registrar reclamo de garantía	Media	Cliente	Ciclo #3
CU16	Consultar estado de garantía/reclamo	Media	Cliente	Ciclo #3
CU17	Gestionar ofertas	Alta	Administrador	Ciclo #3
CU18	Notificaciones push	Media	Sistema→Cliente	Ciclo #3
CU19	Búsqueda por voz (producto/cliente)	Media	Cliente	Ciclo #3
CU20	Reportes por voz (prompts)	Media	Cliente	Ciclo #3

Ciclo #1

NRO	CASO DE USO	PRIORIDAD	ACTORES	CICLO
CU01	Iniciar sesión	Alta	Administrador, Cliente	Ciclo #1
CU02	Cerrar sesión	Alta	Administrador, Cliente	Ciclo #1
CU03	Gestionar roles y permisos	Alta	Administrador	Ciclo #1
CU04	Registrar producto	Alta	Administrador	Ciclo #1
CU05	Editar producto electrodoméstico	Alta	Administrador	Ciclo #1

CU06	Registrar cliente	Alta	Administrador	Ciclo #1
CU07	Editar cliente	Alta	Administrador	Ciclo #1

Ciclo #2

NRO	CASO DE USO	PRIORIDAD	ACTORES	CICLO
CU08	Listar productos	Alta	Administrador	Ciclo #2
CU09	Listar clientes	Alta	Administrador	Ciclo #2
CU10	Bitácora	Alta	Administrador	Ciclo #2
CU11	Registrar venta simple	Alta	Administrador	Ciclo #2
CU12	Generar comprobante PDF	Media	Administrador	Ciclo #2
CU13	Ver dashboard con predicción	Alta	Administrador	Ciclo #2

Ciclo #3

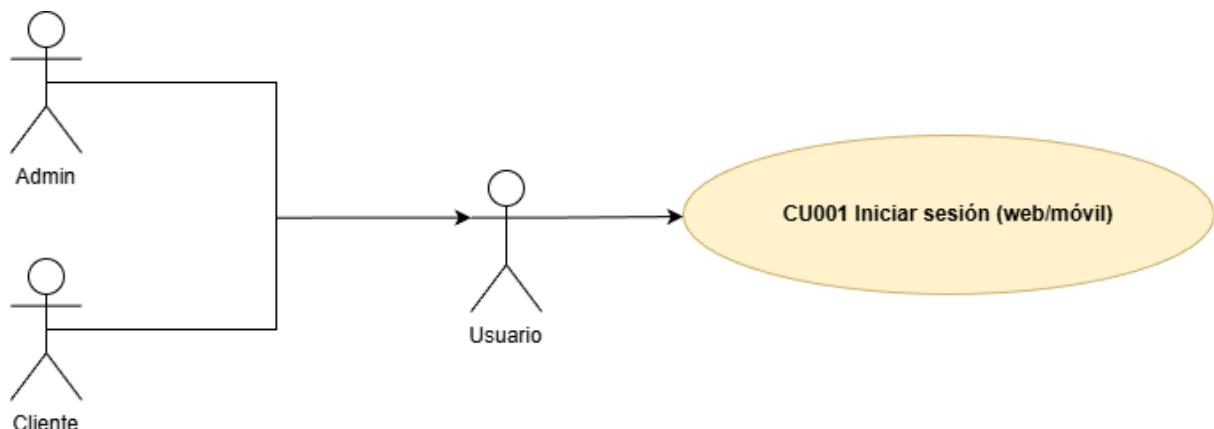
NRO	CASO DE USO	PRIORIDAD	ACTORES	CICLO
CU14	Registrar garantía	Media	Administrador	Ciclo #3
CU15	Registrar reclamo de garantía	Media	Cliente	Ciclo #3
CU16	Consultar estado de garantía/reclamo	Media	Cliente	Ciclo #3

CU17	Gestionar ofertas	Alta	Administrador	Ciclo #3
CU18	Notificaciones push	Media	Sistema→Cliente	Ciclo #3
CU19	Búsqueda por voz (producto/cliente)	Media	Cliente	Ciclo #3

2.1.3 Detalles de un Caso de Uso

Ciclo #1

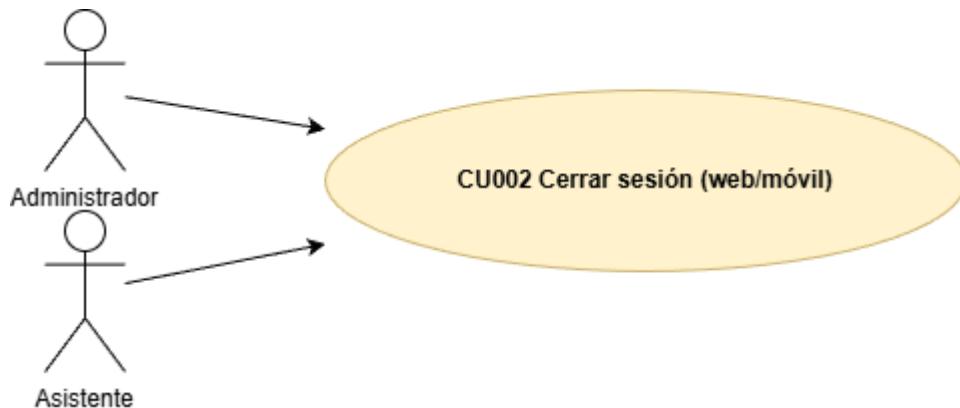
CU01 Iniciar sesión



Caso de Uso	CU1. Iniciar Sesión
Propósito	Permitir que el usuario se autentique para acceder a funcionalidades según su rol.
Resumen	El usuario ingresa credenciales; el sistema valida y entrega sesión JWT; registra el acceso en bitácora.
Actores	Administrador, cliente
Actor Iniciador	Cualquier actor registrado

Flujo de Proceso	<ol style="list-style-type: none"> 1. Usuario introduce ID y contraseña. 2. El sistema busca al usuario. 3. Si no existe, muestra error. 4. Compara contraseñas. 5. Si no coinciden, muestra error. 6. Si coinciden, inicia sesión correctamente.
Pre-Condición	El usuario debe estar registrado en el sistema.
Post-Condición	El usuario queda autenticado y accede al sistema.
Excepción	Credenciales inválidas.

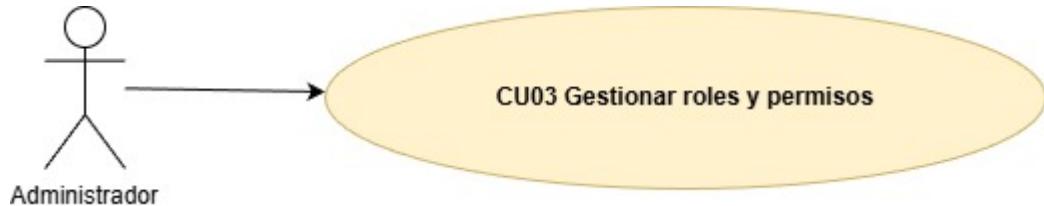
CU02 Cerrar sesión



Caso de Uso	CU2. Cerrar Sesión
Propósito	Finalizar la sesión del usuario de forma segura.
Resumen	El usuario cierra su sesión actual para prevenir accesos no autorizados.
Actores	Administrador, cliente
Actor Iniciador	Cualquier actor autenticado
Flujo de Proceso	<ol style="list-style-type: none"> 1. Usuario selecciona "Cerrar sesión". 2. Confirma (opcional). 3. El sistema finaliza la sesión. 4. Redirige a la pantalla de login.
Pre-Condición	El usuario debe haber iniciado sesión.
Post-Condición	La sesión finaliza y el usuario vuelve a la pantalla de inicio.

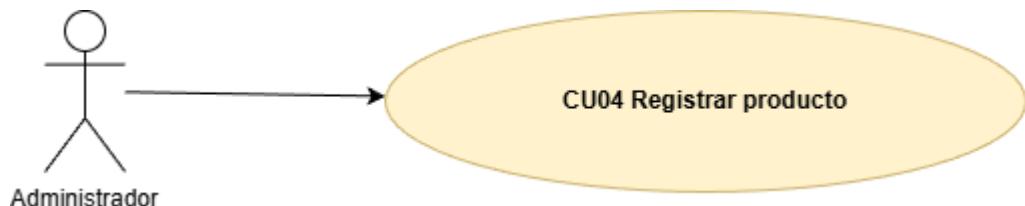
Excepción	Error de conexión al cerrar sesión.
-----------	-------------------------------------

CU03 Gestión de roles y permisos



Caso de Uso	CU3. Gestión de Roles y Permisos
Propósito	Configurar roles del sistema y asignar sus permisos.
Resumen	El administrador administra los roles existentes, sus nombres y permisos asignados.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	<ol style="list-style-type: none"> 1. Ingresar al módulo de roles. 2. Seleccionar crear, modificar, eliminar o asignar permisos. 3. Validar y guardar cambios.
Pre-Condición	El administrador debe haber iniciado sesión.
Post-Condición	Se actualiza la configuración de roles y permisos.
Excepción	Permisos inválidos. El rol está asignado a un usuario activo.

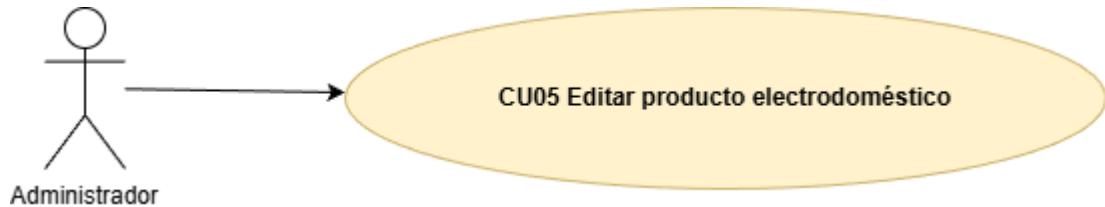
CU04 Registrar producto



Caso de Uso	CU04 Registrar producto
Propósito	Incorporar nuevos productos al catálogo.

Resumen	Alta de producto con categoría, marca, modelo, SKU, precio y stock inicial.
Actores	Administrador.
Actor Iniciador	Administrador.
Flujo de Proceso	(1) Completar formulario; (2) Validar campos obligatorios y unicidad de SKU; (3) Guardar; (4) Bitácora; (5) Actualizar listado/stock.
Pre-Condición	Catálogo de categorías disponible.
Post-Condición	Producto creado y visible en listados.
Excepción	SKU duplicado; datos inválidos; permisos insuficientes.

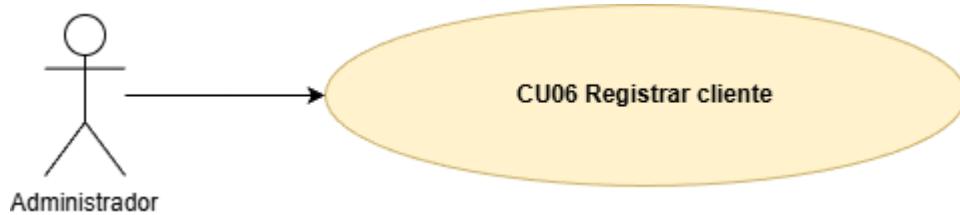
CU05 Editar producto electrodoméstico



Caso de Uso	CU05 Editar producto electrodoméstico
Propósito	Actualizar atributos de productos existentes.
Resumen	Modificar datos (precio, stock, marca/modelo, garantía por defecto, etc.) con control de historial/auditoría.
Actores	Administrador.
Actor Iniciador	Administrador.
Flujo de Proceso	(1) Seleccionar producto; (2) Editar campos; (3) Validar reglas (ej. precio ≥ 0); (4) Guardar; (5) Bitácora;

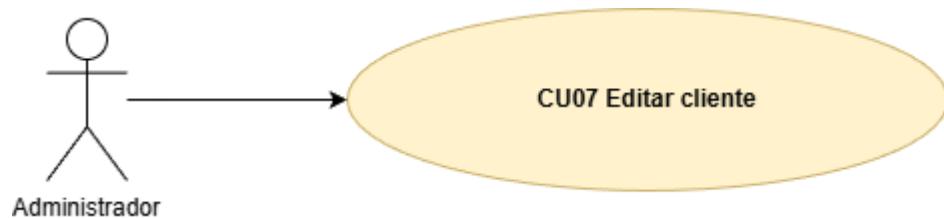
	(6) Refrescar vistas.
Pre-Condición	Producto existente.
Post-Condición	Producto actualizado y auditado.
Excepción	Conflicto de edición; datos inválidos; permisos insuficientes.

CU06 Registrar cliente



Caso de Uso	CU06 Registrar cliente
Propósito	Dar de alta un cliente con datos de contacto.
Resumen	Alta con validaciones (nombre, email/telefono, ciudad, etc.) y estado activo.
Actores	Administrador.
Actor Iniciador	Administrador.
Flujo de Proceso	(1) Completar formulario; (2) Validar obligatorios y email/CI únicos; (3) Guardar; (4) Bitácora; (5) Mostrar en listados.
Pre-Condición	Políticas de datos definidas.
Post-Condición	Cliente registrado y disponible para ventas.
Excepción	Datos duplicados; formato inválido; permisos insuficientes.

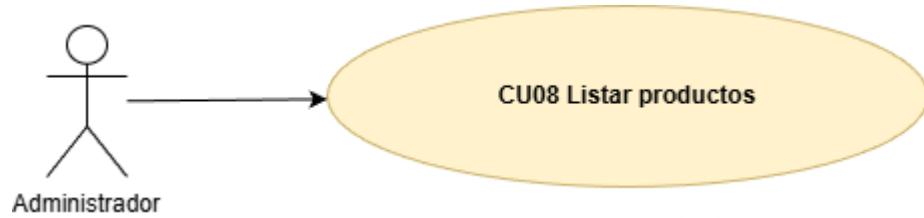
CU07 Editar cliente



Caso de Uso	CU07 Editar cliente
Propósito	Actualizar datos de un cliente existente.
Resumen	Modificar información de contacto/estado con auditoría.
Actores	Administrador.
Actor Iniciador	Administrador.
Flujo de Proceso	(1) Buscar/seleccionar cliente; (2) Editar campos; (3) Validar cambios (unicidad email/CI); (4) Guardar; (5) Bitácora.
Pre-Condición	Cliente existente.
Post-Condición	Cliente actualizado.
Excepción	Datos duplicados; permisos insuficientes; cliente inexistente.

Ciclo #2

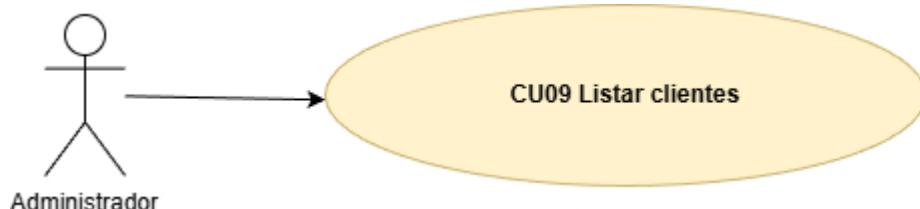
CU08 Listar productos



Caso de Uso	CU08 Listar productos
-------------	-----------------------

Propósito	Consultar el catálogo de productos con filtros y paginación.
Resumen	Muestra tabla de productos con búsqueda, ordenación, filtros por categoría/stock y paginación.
Actores	Administrador.
Actor Iniciador	Administrador
Flujo de Proceso	(1) Abrir módulo de productos; (2) Aplicar filtros/búsqueda; (3) Ver resultados paginados; (4) Seleccionar un producto para ver/editar (opcional).
Pre-Condición	Productos registrados.
Post-Condición	Lista visualizada según criterios.
Excepción	Sin resultados; error de conexión

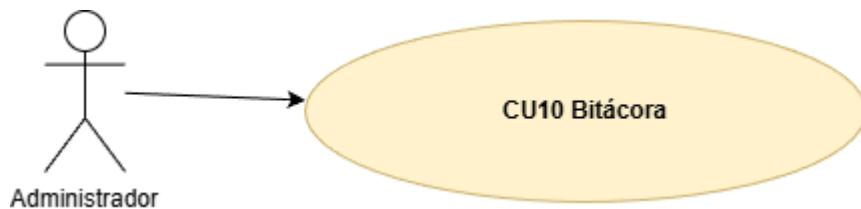
CU09 Listar clientes



Caso de Uso	CU09 Listar clientes
Propósito	Consultar el padrón de clientes con herramientas de filtrado.
Resumen	Tabla de clientes con búsqueda por nombre/CI/email, filtros por estado y paginación..
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	(1) Abrir módulo de productos; (2) Aplicar filtros/búsqueda; (3) Ver resultados paginados; (4) Seleccionar un producto para ver/editar

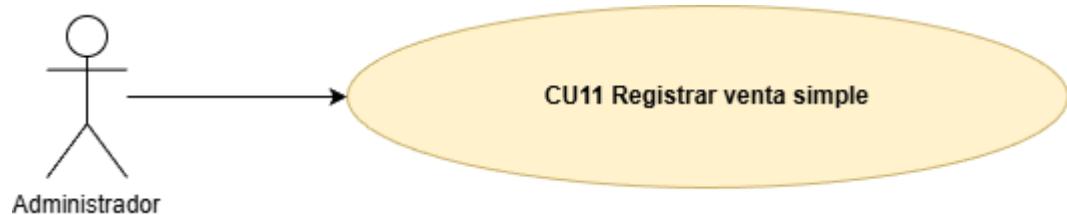
	(opcional).
Pre-Condición	Productos registrados.
Post-Condición	Lista visualizada según criterios.
Excepción	Sin resultados; error de conexión

CU10 Bitácora



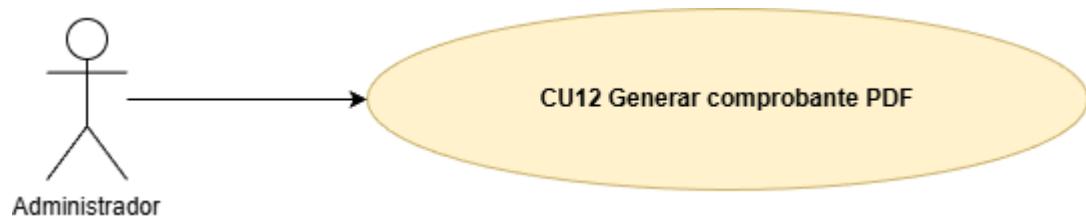
Caso de Uso	CU10 Bitácora
Propósito	Auditar acciones del sistema para trazabilidad.
Resumen	Permite filtrar y consultar eventos (login, CRUD, permisos) y exportar resultados.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	<ul style="list-style-type: none"> (1) Abrir módulo de bitácora; (2) Definir filtros (usuario, fecha, acción); (3) Consultar; (4) Revisar detalle; (5) Exportar (opcional).
Pre-Condición	Registro de auditoría habilitado.
Post-Condición	Evidencia consultada/exportada.
Excepción	Sin datos; permisos insuficientes.

CU11 Registrar venta simple



Caso de Uso	CU11 Registrar venta simple
Propósito	Registrar una venta rápida con productos del catálogo.
Resumen	Selección de cliente y productos; cálculo automático de totales; persistencia y auditoría.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	(1) Seleccionar cliente; (2) Buscar/añadir productos y cantidades; (3) Revisar precios y totales; (4) Guardar venta; (5) Actualizar stock; (6) Registrar en bitácora.
Pre-Condición	Cliente y productos existentes; stock disponible.
Post-Condición	Venta registrada con ítems y totales; stock descontado.
Excepción	Stock insuficiente; cliente inexistente; error de cálculo/pago.

CU12 Generar comprobante PDF



Caso de Uso	CU12 Generar comprobante PDF
-------------	------------------------------

Propósito	Emitir el comprobante (PDF) para una venta registrada.
Resumen	Genera archivo PDF con detalle de la venta, datos del cliente y totales; guarda URL/archivo.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	(1) Seleccionar venta; (2) Generar PDF; (3) Guardar/adjuntar al registro; (4) Permitir descarga/impresión; (5) Bitácora.
Pre-Condición	Venta registrada y completa.
Post-Condición	Comprobante disponible (PDF) asociado a la venta.
Excepción	Venta inexistente/incompleta; error de renderizado/almacenamiento.

CU13 Ver dashboard con predicción

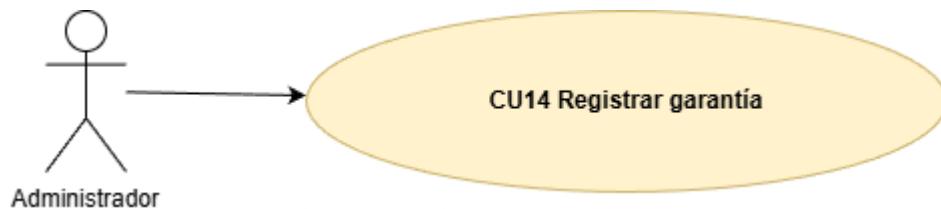


Caso de Uso	CU13 Ver dashboard con predicción
Propósito	Visualizar métricas de ventas y predicciones para soporte a decisiones.
Resumen	Muestra KPIs, tendencias y proyección (modelo ML) con filtros de fecha/categoría.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	(1) Abrir dashboard; (2) Definir rango/categorías; (3) Ver KPIs y gráficos; (4) Consultar predicción/escenarios;

	(5) Exportar (opcional).
Pre-Condición	Datos históricos disponibles; modelo entrenado/publicado.
Post-Condición	Información analítica consumida.
Excepción	Falta de datos; modelo no disponible; error de cálculo.

Ciclo #3

CU14 Registrar garantía



Caso de Uso	CU14 Registrar garantía
Propósito	Asociar una garantía a un ítem de venta con fechas y estado inicial.
Resumen	Se elige venta/ítem, se autocalcula vigencia según meses por defecto del producto y se registra la garantía.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	<ul style="list-style-type: none"> (1) Buscar venta → ítem; (2) Completar meses/fechas si aplica; (3) Validar duplicados; (4) Guardar; (5) Bitácora.
Pre-Condición	Venta e ítem existentes; producto con meses de garantía (opcional).
Post-Condición	Garantía creada (estado ACTIVE) vinculada al ítem de venta.

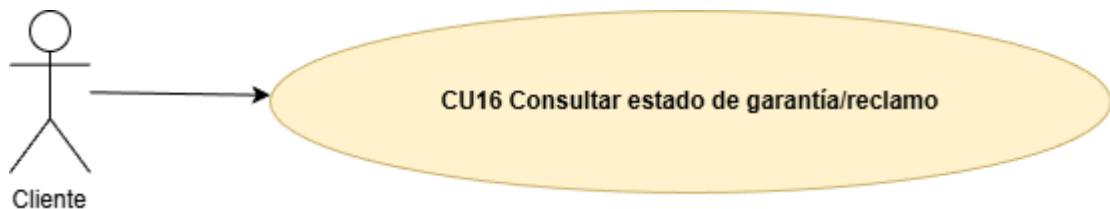
Excepción	Ítem sin venta; garantía ya registrada; datos inválidos.
-----------	--

CU15 Registrar reclamo de garantía



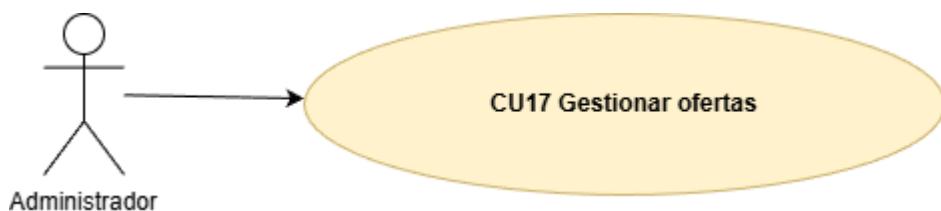
Caso de Uso	CU15 Registrar reclamo de garantía
Propósito	Permitir al cliente abrir un reclamo sobre una garantía activa.
Resumen	El cliente selecciona su garantía, describe el problema y crea un reclamo (OPEN).
Actores	Cliente
Actor Iniciador	Cliente
Flujo de Proceso	(1) Autenticarse; (2) Ver garantías; (3) Seleccionar garantía activa; (4) Describir problema; (5) Crear reclamo; (6) Confirmación/push.
Pre-Condición	Garantía válida y activa asociada al cliente.
Post-Condición	Reclamo registrado (OPEN) y visible para seguimiento.
Excepción	Garantía expirada/inexistente; reclamo duplicado; datos insuficientes.

CU16 Consultar estado de garantía/reclamo



Caso de Uso	CU16 Consultar estado de garantía/reclamo
Propósito	Mostrar al cliente el estado y el historial de su garantía/reclamo.
Resumen	El cliente ve listado y detalle (fechas, estado, resoluciones, observaciones).
Actores	Cliente
Actor Iniciador	Cliente
Flujo de Proceso	(1) Autenticarse; (2) Abrir “Mis garantías/reclamos”; (3) Filtrar/seleccionar; (4) Ver detalle/estado; (5) Descargar evidencia (opcional).
Pre-Condición	Registros asociados al cliente.
Post-Condición	Información consultada.
Excepción	Sin registros; falta de asociación; error de conexión.

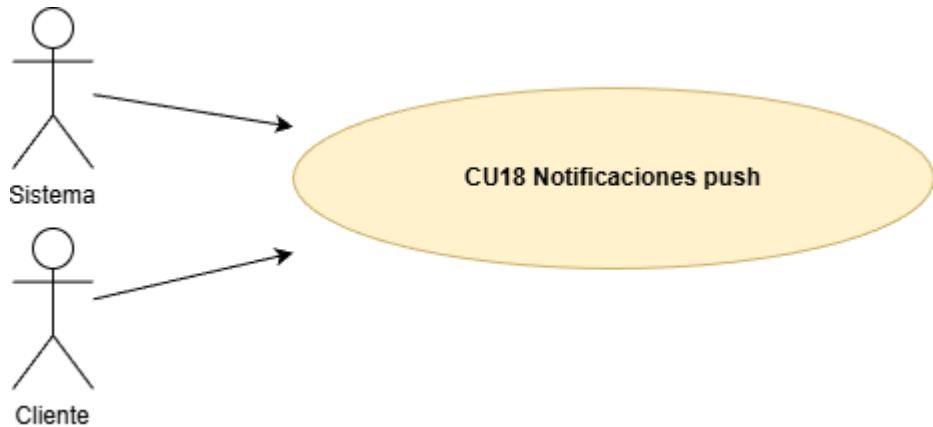
CU17 Gestionar ofertas



Caso de Uso	CU17 Gestionar ofertas
Propósito	Crear/editar/activar ofertas y asociarlas a productos.

Resumen	Se definen vigencias, títulos y productos incluidos; se activa/desactiva la oferta.
Actores	Administrador
Actor Iniciador	Administrador
Flujo de Proceso	(1) Nueva oferta; (2) Fechas, título, descripción; (3) Asociar productos; (4) Activar; (5) Bitácora.
Pre-Condición	Productos existentes.
Post-Condición	Oferta activa y visible para cliente.
Excepción	Fechas inválidas; solapamiento no permitido; sin productos asociados.

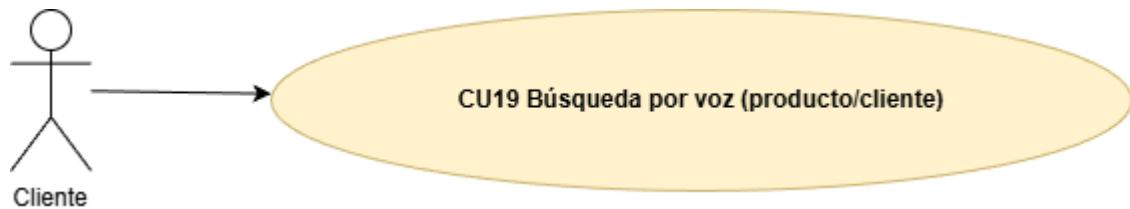
CU18 Notificaciones push



Caso de Uso	CU18 Notificaciones push
Propósito	Enviar avisos transaccionales/operativos al cliente.
Resumen	El sistema genera una push (p. ej., nuevo reporte, oferta activa, reclamo actualizado) y la envía a dispositivos suscritos.
Actores	Sistema, Cliente
Actor Iniciador	Sistema

Flujo de Proceso	(1) Generar evento; (2) Construir mensaje (título, cuerpo, data); (3) Enviar a FCM/APNs; (4) Registrar estado (QUEUED/SENT/FAILED); (5) Cliente abre/lee.
Pre-Condición	Token de dispositivo registrado; permisos de notificación aceptados.
Post-Condición	Notificación entregada/registrada.
Excepción	Token inválido; fallo de proveedor push; dispositivo sin permisos.

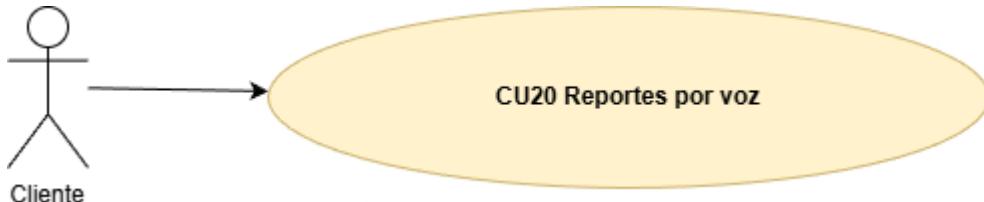
CU19 Búsqueda por voz (producto/cliente)



Caso de Uso	CU19 Búsqueda por voz (producto/cliente)
Propósito	Permitir consultas rápidas mediante entrada de voz.
Resumen	Se captura audio, se transcribe, se interpreta intención (producto/cliente) y se muestran resultados.
Actores	Cliente
Actor Iniciador	Cliente
Flujo de Proceso	(1) Presionar "mic"; (2) Capturar audio; (3) Transcribir; (4) Parsear intención y parámetros; (5) Consultar API; (6) Mostrar resultados.
Pre-Condición	Permiso de micrófono; conectividad.
Post-Condición	Resultados mostrados según intención.

Excepción	Fallo de transcripción; intención ambigua; sin resultados.
-----------	--

CU20 Reportes por voz (prompts)

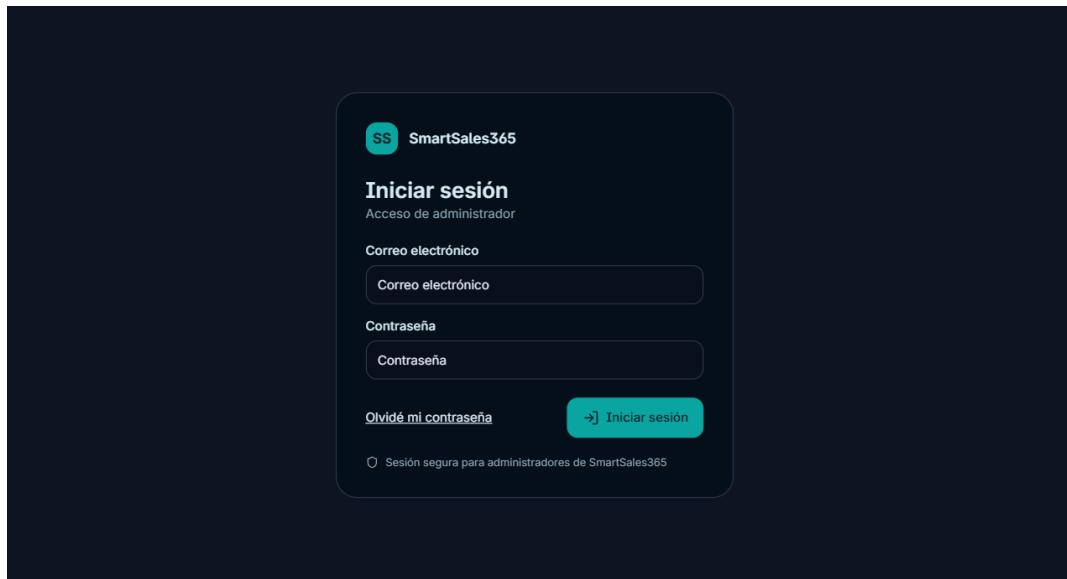


Caso de Uso	CU20 Reportes por voz
Propósito	Generar reportes a demanda a partir de un prompt dictado.
Resumen	Se convierte voz→texto, se interpreta el prompt (rango, filtros, formato) y se genera reporte
Actores	Cliente
Actor Iniciador	Cliente
Flujo de Proceso	(1) Presionar “mic”; (2) Dictar prompt; (3) Transcribir; (4) Parsear intención/parametros; (5) Ejecutar consulta/plantilla; (6) Mostrar o descargar; (7) Bitácora.
Pre-Condición	Catálogo de reportes/interpretador de prompts disponible.
Post-Condición	Reporte entregado (SCREEN/PDF/EXCEL) y registrado.
Excepción	Prompt inválido; permisos insuficientes; tiempo de ejecución excedido.

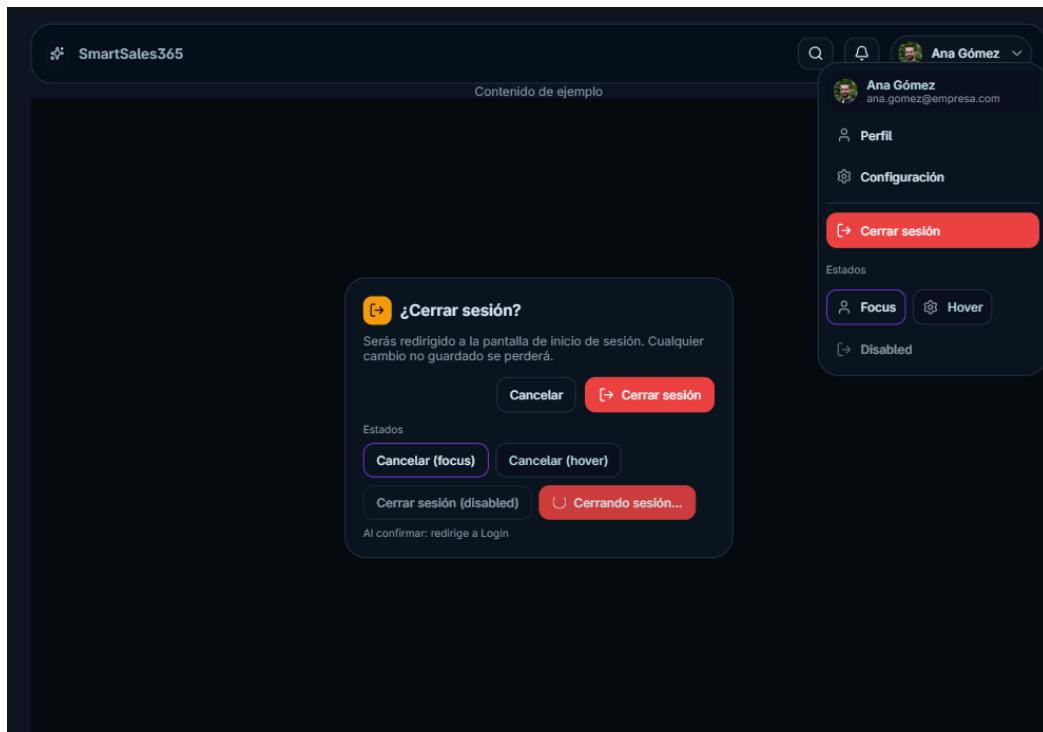
2.1.4 Prototipo de Interfaz de Usuario

Ciclo #1

CU01 Iniciar sesión



CU02 Cerrar sesión



CU03 Gestionar roles y permisos

The screenshot shows the 'IAM/Roles>List' page. On the left sidebar, there are three items: 'IAM/Roles>List', 'IAM/Roles>Detail', and 'IAM/Users/AssignRoles'. The main area displays a table with three rows:

Code	Name	# Permisos	Acciones
admin	Administrador	24	<button>Editar</button> <button>Eliminar</button>
sales_mgr	Sales Manager	12	<button>Editar</button> <button>Eliminar</button>
viewer	Solo lectura	6	<button>Editar</button> <button>Eliminar</button>

Below the table, it says 'Mostrando 1-3 de 3'. At the bottom right, there are navigation buttons: '1', '2', and '3'. A search bar at the top right says 'Buscar rol'.

IAM/Roles/Detail

This section shows the details for the 'sales_mgr' role. It has fields for 'Code' (sales_mgr) and 'Name' (Sales Manager). There are 'Cancelar' and 'Guardar' buttons. To the right, a 'Permisos' section lists several permissions with checkboxes:

- iam.roles.read
- iam.roles.write
- iam.users.read
- iam.users.assign
- sales.opportunities.read
- sales.opportunities.write

A note at the bottom says 'Selecciona múltiples permisos'.

CU04 Registrar producto

The screenshot shows the 'Catalog/ProductForm>New' page. On the left sidebar, there are three items: 'Catalog>List', 'Catalog/ProductForm>New', and 'Catalog/Categories'. The main area has fields for product details:

Categoría Selecciona categoría (category.name)	Marca Apple
Modelo iPhone 16	SKU IP16-256-BLK
Precio 1299.00	Stock 25
Origen de fábrica China	Garantía por defecto (meses) 24

A red error message 'SKU ya existe. Debe ser único.' is displayed next to the SKU field. A note below the stock field says 'Este valor se usa si el vendedor no especifica garantía. En meses.' There is a 'Best seller' toggle switch labeled 'is_best_seller'.

At the bottom right are 'Cancelar' and 'Guardar' buttons.

CU05 Editar producto electrodoméstico

SmartSales365

Catalog > ProductForm > Edit
Catalog/ProductForm/Edit

Historico de cambios

- > Precio actualizado por m.suarez — 2025-10-02 14:21 1299.00 → 1249.00
- > Stock ajustado por system — 2025-09-28 09:03 25 → 18
- > Creación de producto por admin — 2025-09-15 11:40 SKU IP16-256-BLK

Categoría	Marca
Smartphones	Apple
Modelo	SKU
iPhone 16	IP16-256-BLK
Precio	Stock
1299.00	25
Debe ser único	Debe ser mayor o igual a 0
Origen de fábrica	Garantía por defecto (meses)
China	24
Best seller	En meses
<input checked="" type="checkbox"/> is_best_seller	

Guardar **Eliminar**

CU06 Registrar cliente

SmartSales365

Clients > ClientForm > New
Clients/ClientForm/New

Aplicar máscara

Nombre completo	Email	Teléfono (máscara opcional)
		<input type="button" value="Aplicar máscara"/>
Validación: debe tener formato usuario@dominio		
Sugerencia de máscara: +CC AA NNNN-NNNN		
Ciudad		
	<input type="button" value="Cancelar"/> Guardar	

CU07 Editar cliente

SmartSales365

Clients > ClientForm > Edit
Clients/ClientForm/Edit

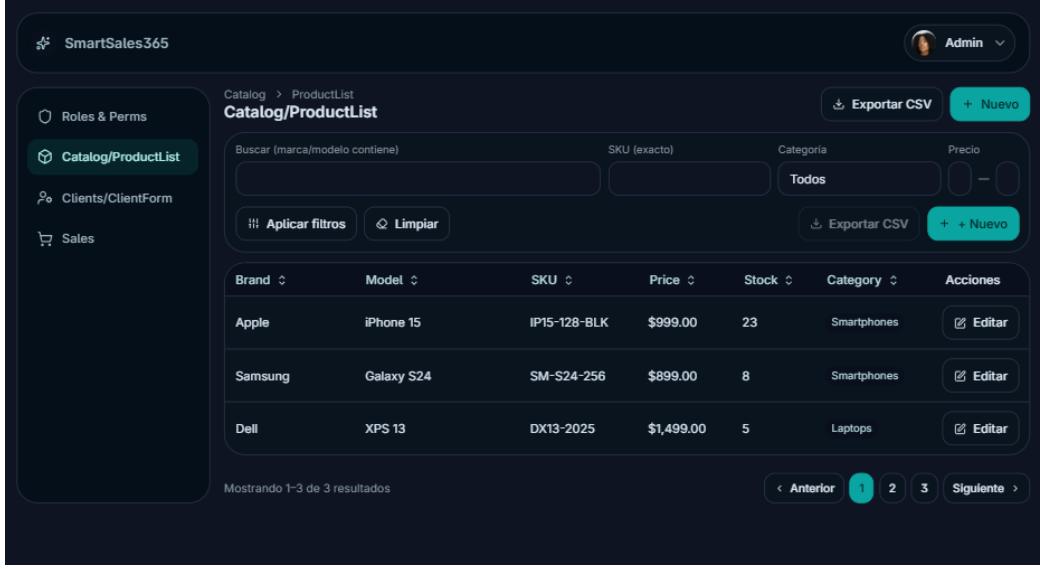
Datos **Historial de compras** **Garantías**

Nombre completo	Email	Teléfono (máscara opcional)
Ana Rodriguez	ana@mail.com	+54 9 11 1234-5678
Validación: debe tener formato usuario@dominio		
Sugerencia de máscara: +CC AA NNNN-NNNN		
Ciudad	Buenos Aires	
<input type="button" value="Cancelar"/> Guardar		

Fecha	Total	Estado
2025-06-05	\$420,00	Completada
2025-05-12	\$199,90	Pendiente
2025-03-21	\$1.249,00	Devuelta

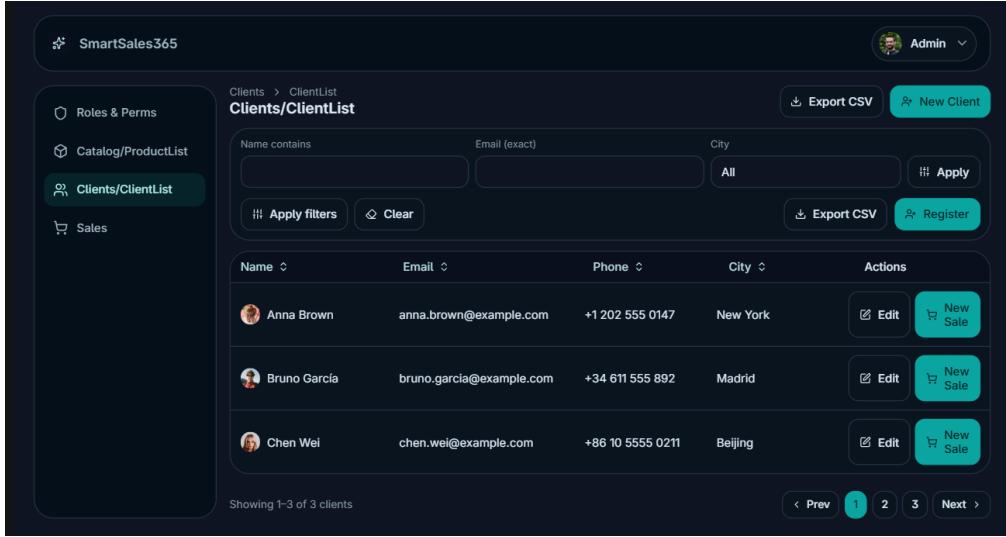
Ciclo #2

CU08 Listar productos



The screenshot shows the SmartSales365 application interface for listing products. The top navigation bar includes the logo, user profile (Admin), and a search bar. On the left, there's a sidebar with 'Roles & Perms', 'Catalog/ProductList' (which is highlighted in green), 'Clients/ClientForm', and 'Sales'. The main content area is titled 'Catalog > ProductList Catalog/ProductList'. It features a search bar with fields for 'Buscar (marca/modelo contiene)', 'SKU (exacto)', 'Categoria', and 'Precio'. Below the search bar are buttons for 'Aplicar filtros' (Apply filters) and 'Limpiar' (Clear). A table lists three products: Apple iPhone 15, Samsung Galaxy S24, and Dell XPS 13. The table columns are Brand, Model, SKU, Price, Stock, Category, and Actions. Each product row has an 'Editar' (Edit) button. At the bottom, it says 'Mostrando 1-3 de 3 resultados' and includes navigation buttons for 'Anterior' (Previous), page numbers 1, 2, 3, and 'Siguiente' (Next).

CU09 Listar clientes



The screenshot shows the SmartSales365 application interface for listing clients. The top navigation bar includes the logo, user profile (Admin), and a search bar. On the left, there's a sidebar with 'Roles & Perms', 'Catalog/ProductList', 'Clients/ClientList' (which is highlighted in green), and 'Sales'. The main content area is titled 'Clients > ClientList Clients/ClientList'. It features a search bar with fields for 'Name contains', 'Email (exact)', and 'City'. Below the search bar are buttons for 'Apply filters' and 'Clear'. A table lists three clients: Anna Brown, Bruno Garcia, and Chen Wei. The table columns are Name, Email, Phone, City, and Actions. Each client row has an 'Edit' button and a 'New Sale' button. At the bottom, it says 'Showing 1-3 of 3 clients' and includes navigation buttons for 'Prev' (Previous), page numbers 1, 2, 3, and 'Next'.

CU10 Bitácora

The screenshot shows the SmartSales365 Audit/Logs page. The left sidebar has a dark theme with white text and icons. The main area is titled "Audit/Logs" and shows a table of audit logs. The table columns are: ts (timestamp), user, action, entity, entity_id, and detalle (detail). The logs listed are:

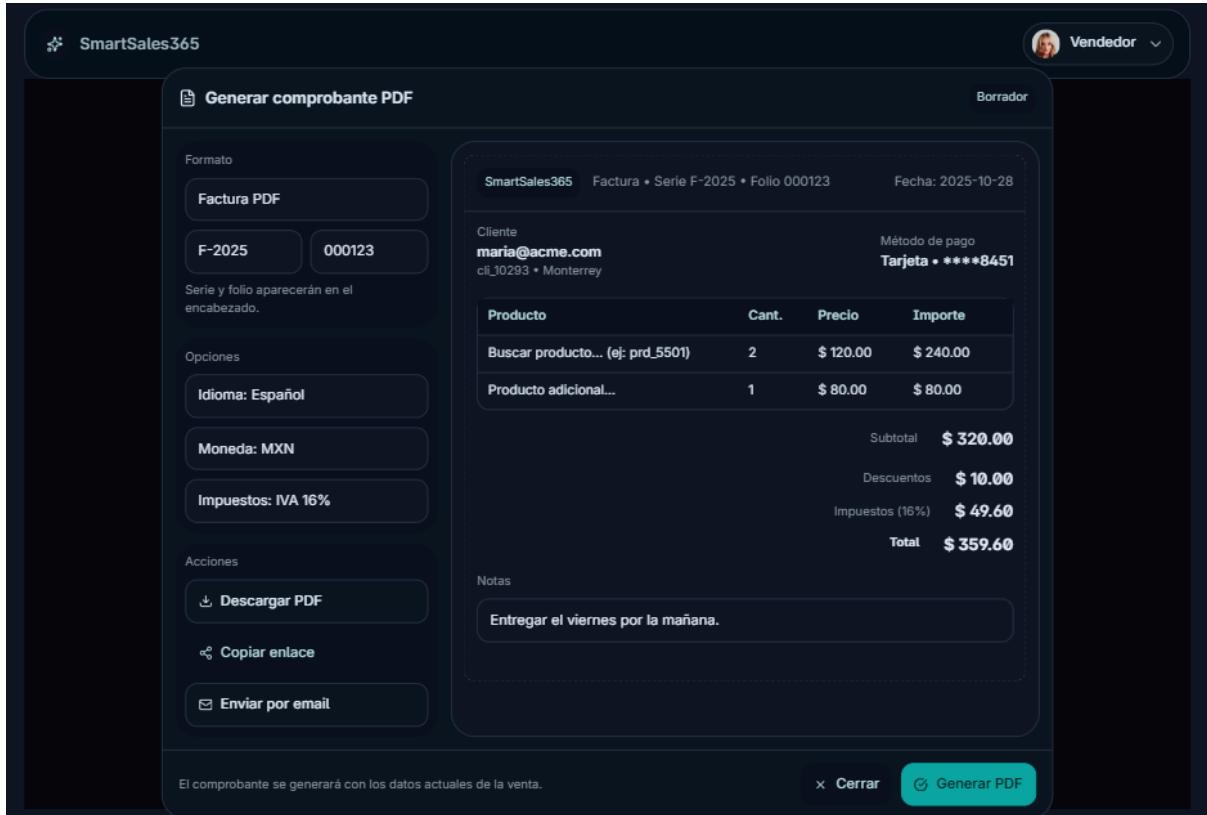
ts	user	action	entity	entity_id	detalle
2025-10-28 09:21:14 UTC	maria@acme.com	LOGIN	auth_session	sess_9f23a	ver detalle
2025-10-27 18:02:03 UTC	juan@acme.com	CREATE_CLIENT	client	cli_10293	ver detalle
2025-10-27 17:11:44 UTC	ana@acme.com	GENERATE_RECEIPT	receipt	rcp_77881	ver detalle
2025-10-27 16:40:10 UTC	ana@acme.com	UPDATE_PRODUCT	product	prd_5501	ver detalle

At the bottom, it says "Mostrando 1-4 de 124 eventos" and has navigation buttons for Prev, Next, and page numbers 1, 2, 3.

CU11 Registrar venta simple

The screenshot shows the SmartSales365 "Registrar venta simple" (Simple Sale Registration) page. The left sidebar has a dark theme with white text and icons. The main area is titled "Registrar venta simple". It has sections for "Cliente" (Client) and "Productos" (Products). The "Cliente" section includes fields for "Buscar cliente (email o nombre)" (Search client (email or name)) with value "maria@acme.com", "Email" with value "maria@acme.com", "Cliente ID" with value "cli_10293", and "Ciudad" (City) with value "Monterrey". The "Productos" section includes two rows of product details: one row for "Buscar producto... (ej: prd_5501)" with value "2" and "precio" \$120.00, and another for "Producto adicional..." with value "1" and "precio" \$80.00. Below these are summary totals: Subtotal \$320.00, Descuentos \$10.00, Impuestos (16%) \$49.60, and Total \$359.60. The "Pago" (Payment) section includes fields for "Método de pago" (Payment method) with value "Tarjeta", "Referencia" (Reference) with value "8451", "Fecha" (Date) with value "2025-10-28", "Tipo de comprobante" (Type of document) with value "Factura", "# Serie" (Series #) with value "F-2025", and "Folio" (Folio) with value "000123".

CU12 Generar comprobante PDF



The screenshot shows the SmartSales365 application interface for generating a PDF invoice. The top navigation bar includes the brand logo, user profile, and a 'Borrador' (Draft) button. The main window has a title 'Generar comprobante PDF' and a status 'Formato' (Format). On the left, there are several configuration sections: 'Formato' (Factura PDF, F-2025, 000123), 'Opciones' (Idioma: Español, Moneda: MXN, Impuestos: IVA 16%), and 'Acciones' (Descargar PDF, Copiar enlace, Enviar por email). A note at the bottom states: 'El comprobante se generará con los datos actuales de la venta.' On the right, the generated PDF document is displayed. It includes header information: 'SmartSales365 Factura • Serie F-2025 • Folio 000123' and 'Fecha: 2025-10-28'. It lists a client ('maria@acme.com, cli_10293 • Monterrey') and a payment method ('Tarjeta • ****8451'). The invoice details two items: 'Buscar producto... (ej: prd_5501)' (2 units, \$120.00 each) and 'Producto adicional...' (1 unit, \$80.00). The total breakdown is: Subtotal \$320.00, Descuentos \$10.00, Impuestos (16%) \$49.60, and Total \$359.60. A note at the bottom of the PDF says: 'Entregar el viernes por la mañana.'

SmartSales365

Vendedor

Generar comprobante PDF

Borrador

Formato

Factura PDF

F-2025 000123

Serie y folio aparecerán en el encabezado.

Opciones

Idioma: Español

Moneda: MXN

Impuestos: IVA 16%

Acciones

↓ Descargar PDF

✉ Copiar enlace

✉ Enviar por email

El comprobante se generará con los datos actuales de la venta.

SmartSales365 Factura • Serie F-2025 • Folio 000123 Fecha: 2025-10-28

Cliente maria@acme.com, cli_10293 • Monterrey Método de pago Tarjeta • ****8451

Producto	Cant.	Precio	Importe
Buscar producto... (ej: prd_5501)	2	\$ 120.00	\$ 240.00
Producto adicional...	1	\$ 80.00	\$ 80.00

Subtotal **\$ 320.00**
Descuentos **\$ 10.00**
Impuestos (16%) **\$ 49.60**
Total \$ 359.60

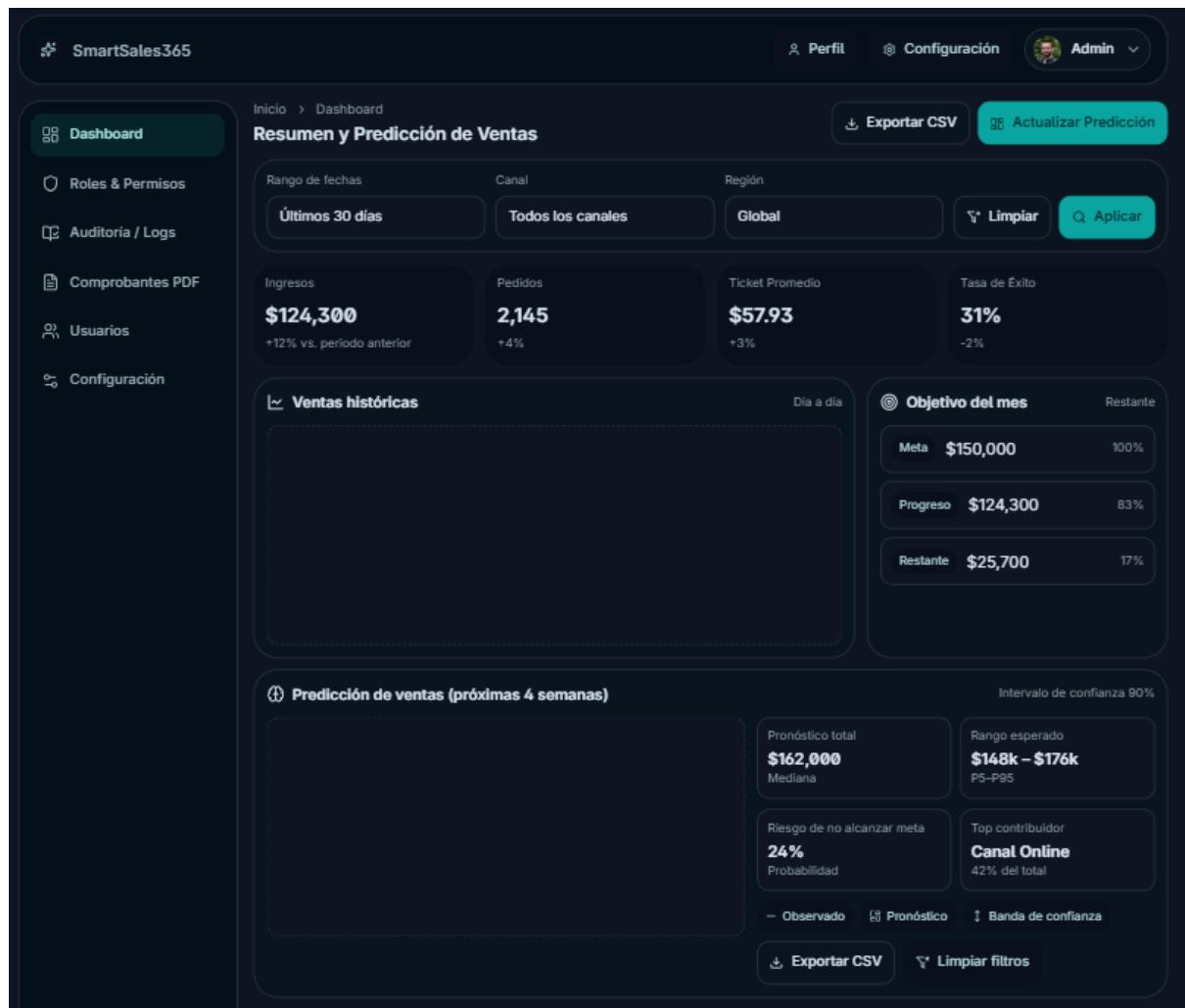
Notas

Entregar el viernes por la mañana.

x Cerrar

Generar PDF

CU13 Ver dashboard con predicción



Ciclo #3

CU14 Registrar garantía

SmartSales365

Profile Settings Admin

Inicio > Garantías > Registrar

Registrar garantía

Detalles de la garantía

Campos obligatorios

Número de serie	Producto
SN-	Seleccionar producto
Fecha de compra	Fecha de activación
AAAA-MM-DD	AAAA-MM-DD
Cliente (email)	Tienda / Canal
cliente@dominio.com	Online

Autocomplete con app_user.email

Descripción del problema (opcional)

Sin incidencias al registrar

Pais

Seleccionar

Región

Seleccionar

Resumen

Previsualización

Estado: **Borrador**

Vencimiento: **2026-01-15**

Cliente: **cliente@dominio.com**

Producto: —

Adjuntos

0 archivos

Aún no hay archivos adjuntos.

Metadata JSON

```
{"serial": "SN-", "customer": "cliente@dominio.com", "product": null, "purchase_date": "AAAA-MM-DD", "activation_date": "AAAA-MM-DD", "coverage": "Estándar", "duration_months": 12, "channel": "Online", "attachments": []}
```

No hay resultados coincidentes con los filtros.

Exportar CSV Limpiar

Guardar registro

Generar comprobante PDF Copiar enlace

Adjuntar comprobante Escanear factura

Registro guardado correctamente.

No se pudo guardar el registro. Intenta de nuevo.

CU15 Registrar reclamo de garantía

The screenshot displays the SmartSales365 application interface. The top navigation bar includes the logo, 'SmartSales365', 'Profile', 'Settings', and a user account section for 'Admin'. On the left, a sidebar menu lists: Dashboard, Roles & Permisos, Auditoria / Logs, Comprobantes PDF, Registrar garantía (highlighted in blue), and Reclamo de garantía. The main content area shows the 'Registrar reclamo de garantía' (Register warranty claim) page. The page header includes 'Inicio > Garantías > Registrar reclamo'. It features several input fields and sections: 'Datos del reclamo' (Claim details) with fields for serial number (SN-), product selection, and client information (email: cliente@dominio.com, channel: Online); 'Resumen' (Summary) showing status (Borrador), priority (Media), and client (cliente@dominio.com); 'Adjuntos' (Attachments) which is currently empty; and a 'Metadata JSON' section containing the following JSON object:

```
{"claim_id":null,"serial":"SN-","product":null,"customer":"cliente@dominio.com","claim_date":"AAAA-MM-DD","priority":"Media","claim_type":"Reparación","resolution_method":"Servicio técnico","channel":"Online","description":"","attachments":[]}
```

At the bottom, there are two status messages: a green box stating 'Reclamo enviado correctamente.' (Claim sent successfully.) and a red box stating 'No se pudo enviar el reclamo. Intenta de nuevo.' (The claim could not be sent. Try again.).

CU16 Consultar estado de garantía/reclamo

The screenshot displays the SmartSales365 application interface, specifically the 'Garantías' (Warranties) section. The top navigation bar includes links for 'Profile', 'Settings', and a user account labeled 'Admin'. The main content area is titled 'Consultar estado de garantía / reclamo' (Check warranty/reclamo status). It features a search bar with placeholder text 'Usa número de reclamo o serie' and several filter options: 'Nº Reclamo' (CLM-), 'Nº Garantía / Serie' (SN-), 'Correo del cliente' (cliente@dominio.com), 'Estado' (Todos), 'Prioridad' (Todas), and 'Rango de fechas' (AAAA-MM-DD a AAAA-MM-DD). Below the filters is a button to 'Buscar' (Search) and another to 'Escanear comprobante' (Scan document). A table lists three warranty/reclamo entries:

Reclamo	Cliente	Producto	Estado	Prioridad	Fecha
CLM- 10234	ana@acme.com	SmartWatch X	En revisión	Alta	2025-02-01
CLM- 10212	carlos@beta.io	Headphones Z	Resuelto	Media	2025-01-18
CLM- 10190	luis@contoso.com	Router Pro	Requiere info	Baja	2025-01-05

A message below the table states 'No hay resultados para los filtros aplicados.' (No results for the applied filters). A green banner indicates 'Resultados actualizados.' (Results updated). A red banner displays the error message 'Error al cargar resultados. Intenta nuevamente.' (Error loading results. Try again.). To the right, a 'Detalle del reclamo' (Reclamo details) panel shows the claim number 'CLM-10234' and status 'En revisión' (Under review). A log of events is listed:

- Repcionado**: 2025-02-01 09:10. Message: 'Se creó el reclamo y se registraron los datos del cliente.'
- En revisión**: 2025-02-02 10:22. Message: 'Equipo técnico evaluando evidencia adjunta.'
- Solicitud de información**: 2025-02-03 14:05. Message: 'Se solicitó video adicional del fallo.'

At the bottom right, there are buttons for 'Metadata JSON', 'Exportar CSV', and 'Limpiar filtros'.

```
{"claim_id": "CLM-10234", "serial": "SN-9933-AX", "product": "SmartWatch X", "customer": "ana@acme.com", "priority": "Alta", "status": "En revisión", "created_at": "2025-02-01", "last_update": "2025-02-03", "channel": "Online", "attachments": ["factura.pdf"], "sla_days": 7}
```

CU17 Gestionar ofertas

Gestionar ofertas

Filtros

Nombre de la oferta	SKU / Producto	Estado
Semana Cyber	SKU-	Todos

Tipo	Rango de fechas	Canal
Porcentaje	AAAA-MM-DD a AAAA-MM-DD	Online

Detalle de la oferta

Nombre	Estado
Semana Cyber	Activa

Tipo	Descuento
Porcentaje	20%

Inicio	Fin
2025-02-10	2025-02-18

Metadata

```
{"offer_id": "OFF-2045", "name": "Semana Cyber", "type": "percentage", "value": "20", "products": [{"sku": "SWX-001", "name": "SmartWatch X"}], "channels": ["Online", "App"], "status": "active", "start_date": "2025-02-10", "end_date": "2025-02-18", "limits": {"per_user": 1, "stock_cap": 500}, "conditions": {"min_cart": 0, "regions": ["AR", "CL"]}}
```

CU18 Notificaciones push

Notificaciones push

Filtros

Titulo	Estado	Audiencia
Rebajas de verano	Todos	Todos los usuarios

Rango de fechas	Plataforma	Etiqueta
AAAA-MM-DD a AAAA-MM-DD	Web + Mobile	ofertas

Crear campaña

Título	Subtítulo
Rebajas de verano	Hasta 40% OFF

Campaña

Campaña	Audiencia	Estado	Programación
Rebajas de verano	Usuarios activos (45K)	Enviando 2025-07-01 10:00	@ Ver
Recordatorio carro	Abandonos 24h	Pausada diario 09:00	@ Ver
Bienvenida	Nuevos registros	Finalizada Último envío 2025-10-01	@ Ver

No hay campañas para los filtros aplicados.

Campañas actualizadas correctamente.

No se pudieron cargar las campañas.

CU19 Búsqueda por voz (producto/cliente)

The screenshot displays the SmartSales365 application interface. At the top, there is a navigation bar with the brand name "SmartSales365" and user profile options for "Profile", "Settings", and "Admin".

The main content area is titled "Búsqueda por voz (producto / cliente)". It features a "Captura de voz" section where users can dictate their query. A microphone icon indicates the recording function. Below the microphone, examples of what to say are shown: "precio del iPhone 15", "cliente María Pérez", and "stock de auriculares".

The "Idioma" (Language) is set to "Español (ES)".

The "Resultado" (Results) section shows a table with columns: Tipo, Nombre, Identificador, and Estado. It lists two entries:

- Producto iPhone SKU- En 15 Pro IP15P- stock 256 GB 256. An "Ver" (View) button is next to it.
- Cliente María CLI- Activa Pérez 009871. An "Perfil" (Profile) button is next to it.

At the bottom of the results section, there are "Exportar CSV" (Export to CSV) and "Limpiar filtros" (Clear filters) buttons.

The "Metadata JSON" section contains the following code:

```
{"query": "producto: iPhone 15 Pro", "entities": [{"type": "product", "value": "iPhone 15 Pro"}], "filters": {"stock": "any", "price_min": null, "price_max": null}}
```

CU20 Reportes por voz (prompts)

SmartSales365

Profile Settings Admin

Inicio > Reportes > Voz

Reportes por voz (prompts)

Generador de reporte por voz

Dí o escribe un prompt de reporte

Ejemplos: Ventas por región último trimestre, Top 10 clientes por facturación, Ofertas convertidas este mes.

Prompt de reporte:

"muéstrame ventas por vendedor y categoría en los últimos 30 días"

Formato:

Tabla

Generar reporte Reintentar Exportar CSV

Limpiar filtros

Filtros

Rango de fechas: Últimos 30 días

Entidad: Ventas

Agrupar por: Vendedor

Exportar CSV Limpiar filtros

Transcripción

Entrada: "muéstrame ventas por vendedor y categoría en los últimos 30 días"

Metadata JSON

Prompt estructurado:

```
{"entity": "sales", "metrics": ["revenue"], "groupBy": ["seller", "category"], "dateRange": {"type": "last", "value": 30, "unit": "days"}, "format": "table"}
```

Presets

Atajos:

- Resumen semanal: Ventas totales, promedio ticket. Usar
- Clientes top: Por facturación trimestral. Usar
- Ofertas ganadas: Comparativa mensual. Usar
- Reclamos garantía: Por producto. Usar

Resultados del reporte

Vista previa

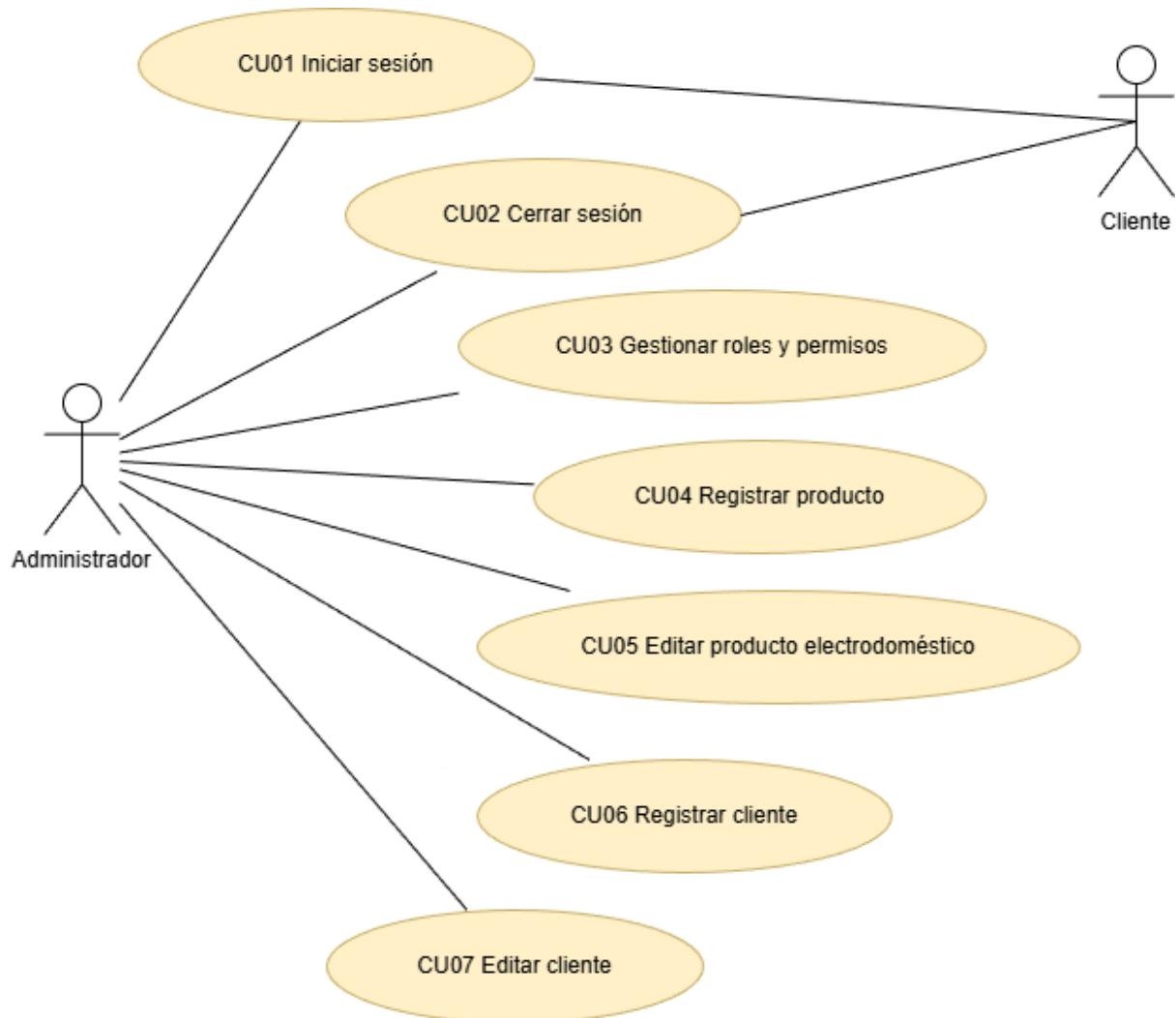
Métrica	Clave	Periodo	Estado
Ventas	Región Norte	Últimos 30 días	+12%
Ciencia	Top 10	Mes actual	10
Ofertas	Tasa de conversión	Q2	24%

Ver Detalle Exportar

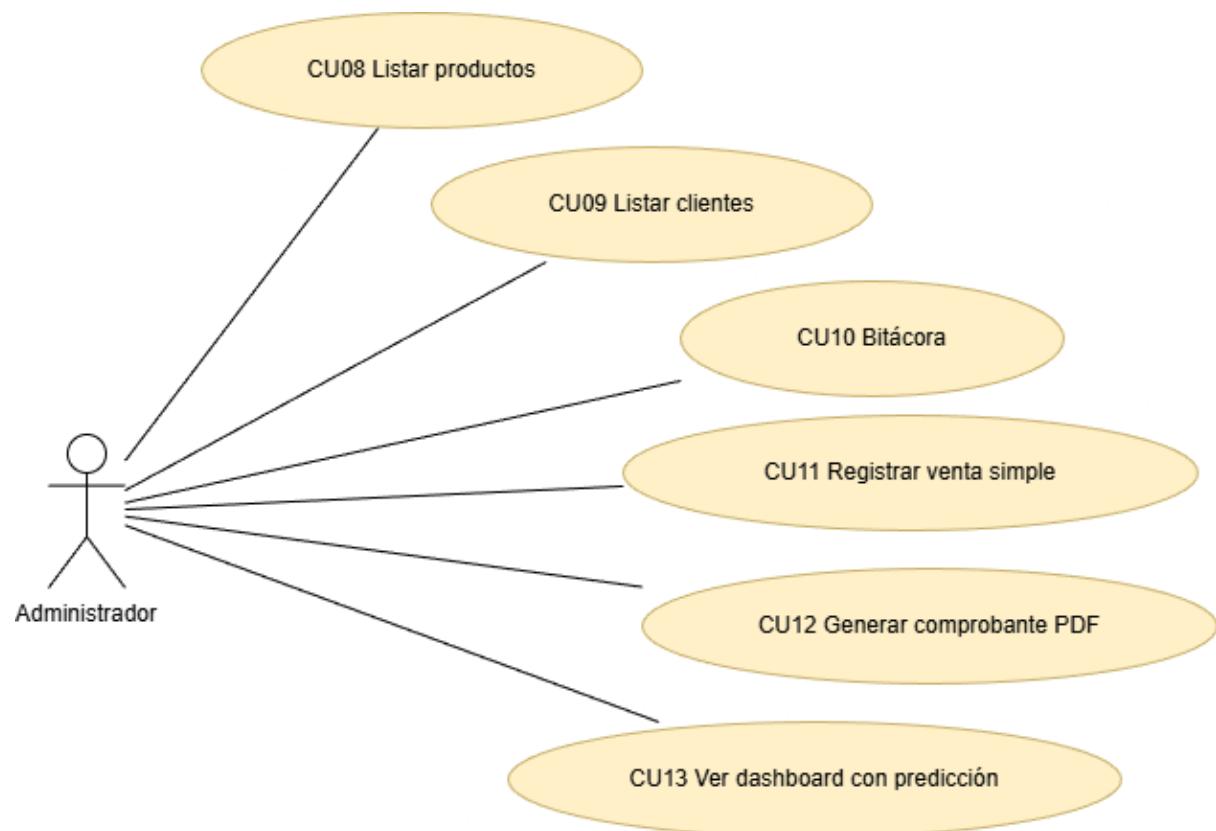
No se encontraron datos para el prompt. Ajusta el periodo o la métrica.

2.1.5 Estructurar el Modelo de Caso de Uso

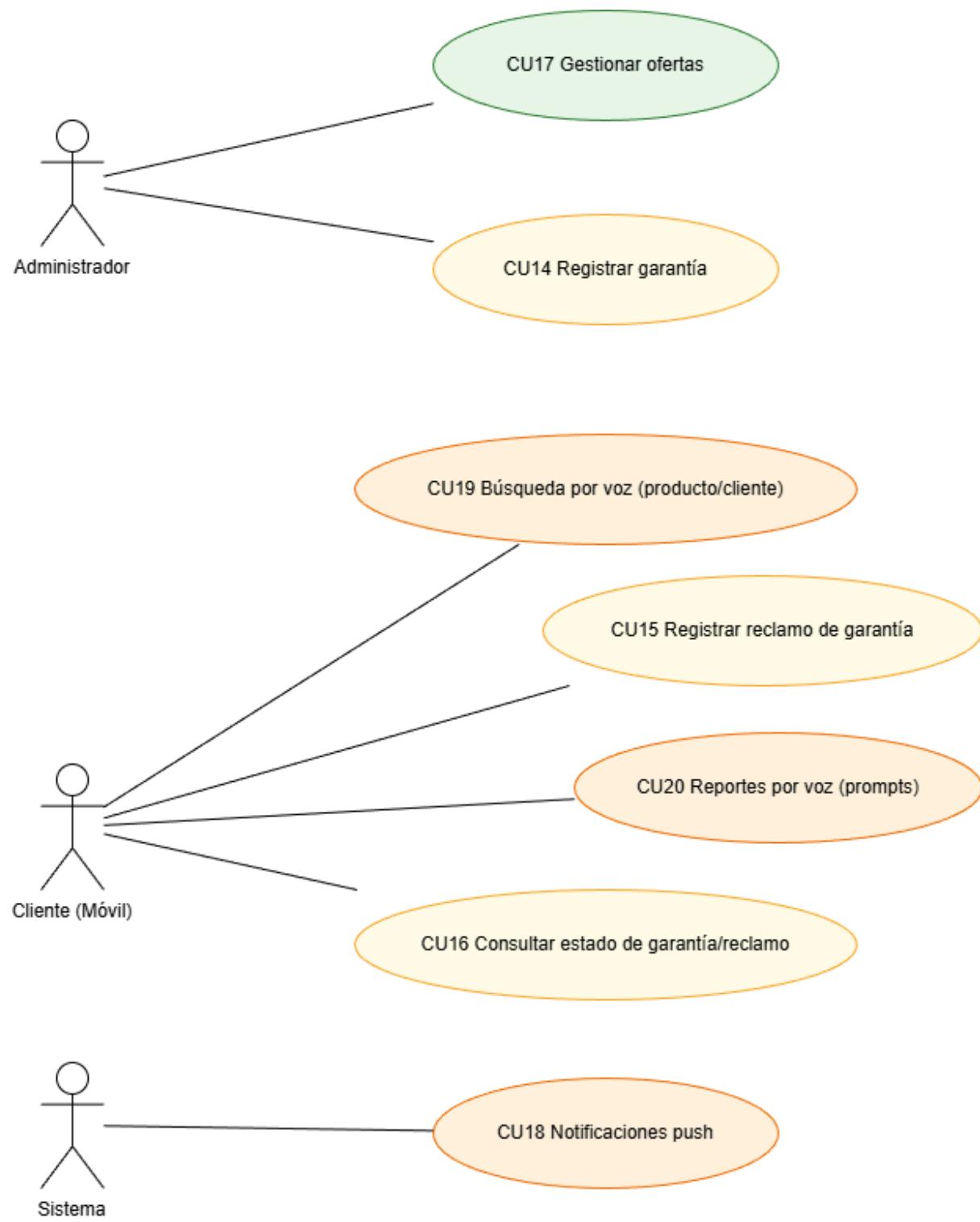
Ciclo #1



Ciclo #2



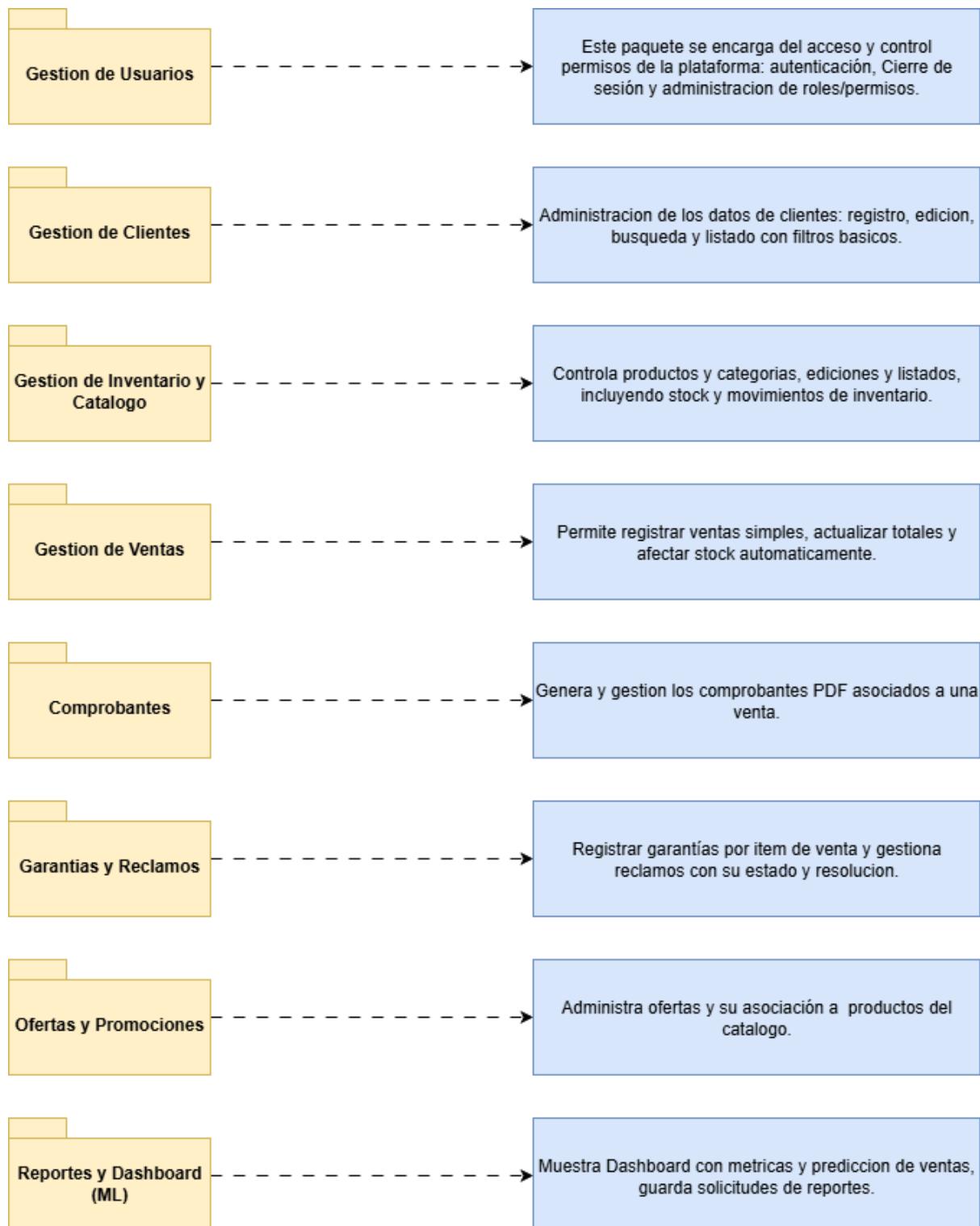
Ciclo #3



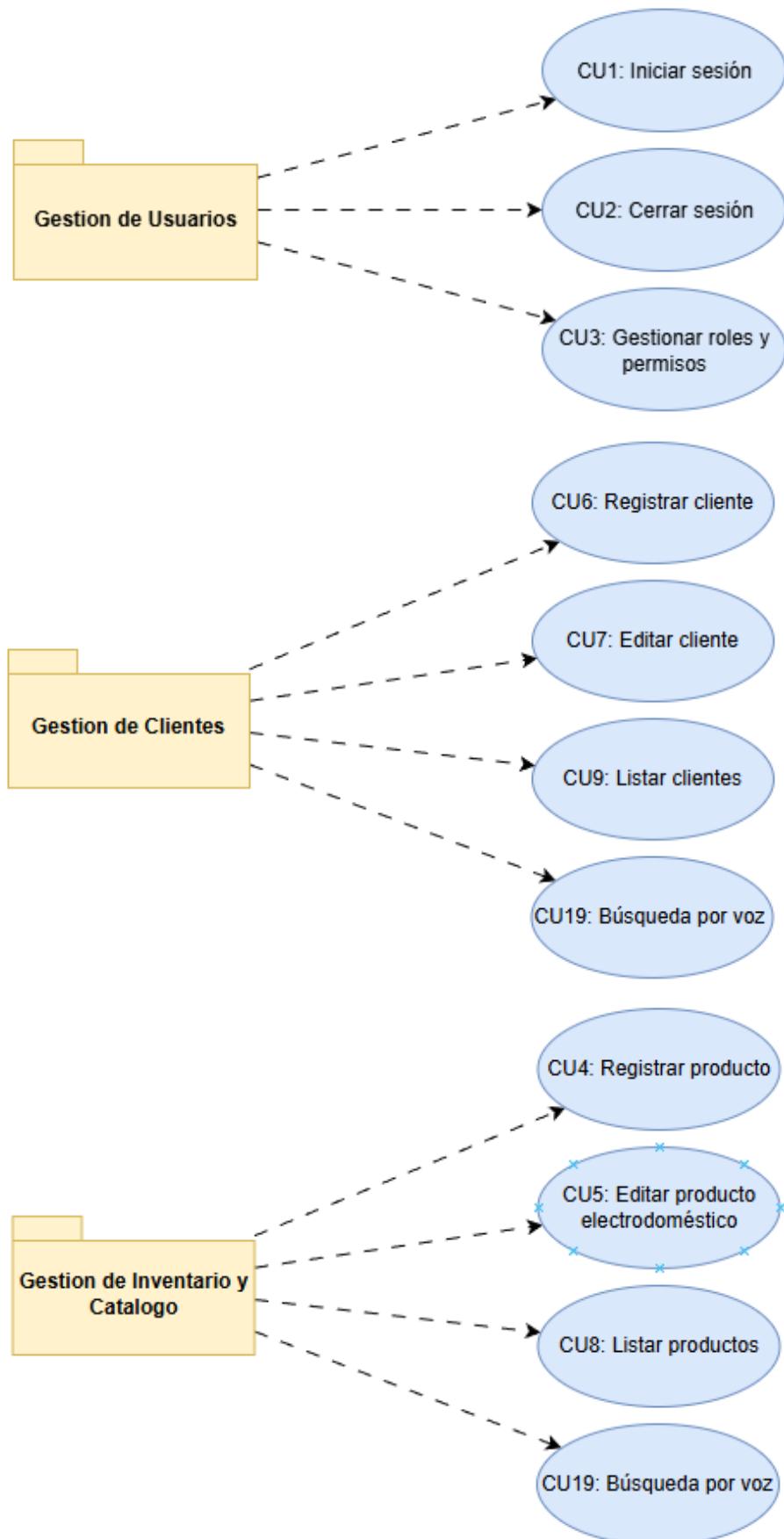
2.2 Flujo de Trabajo: Análisis

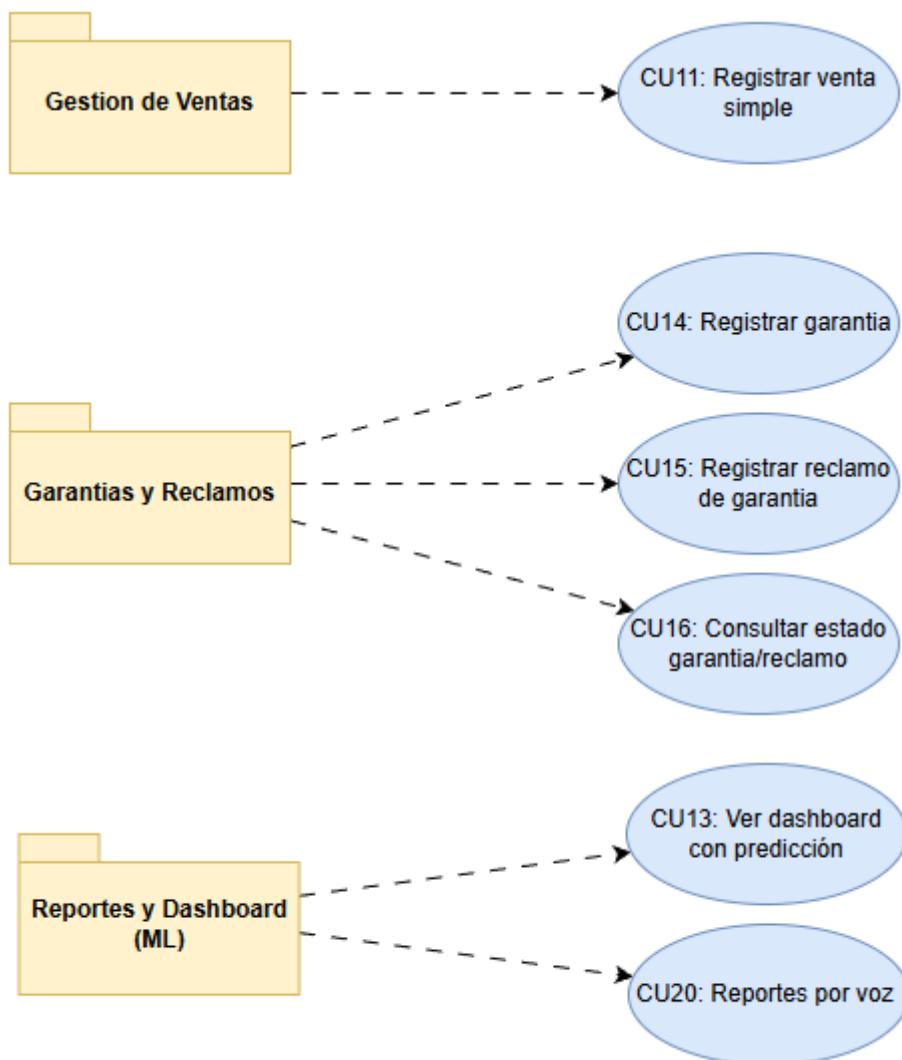
2.2.1 Análisis de Arquitectura

2.2.1.1 Identificar Paquetes

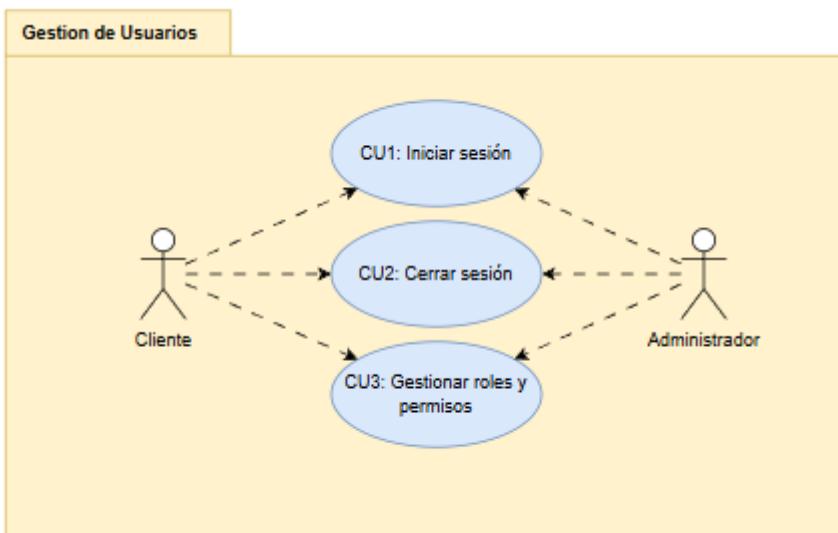


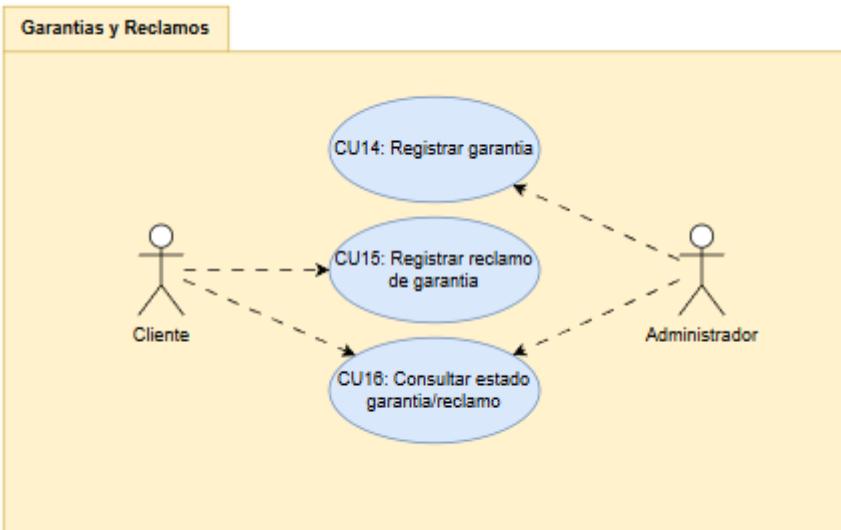
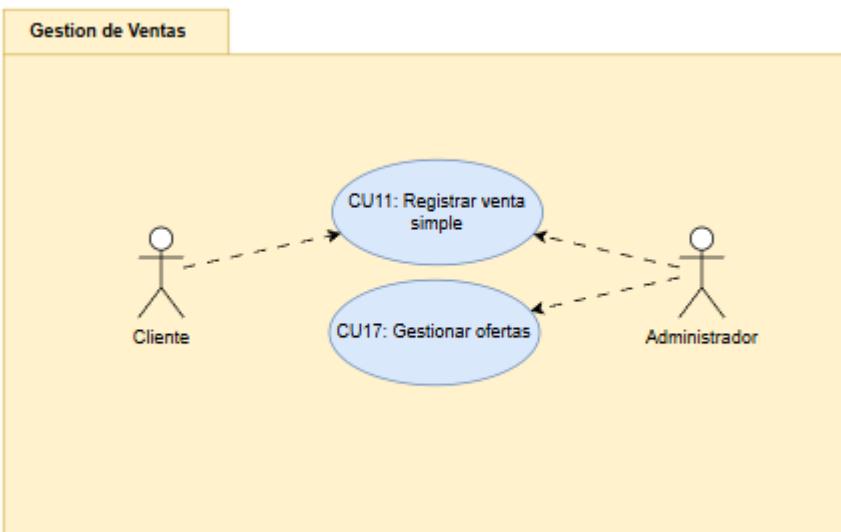
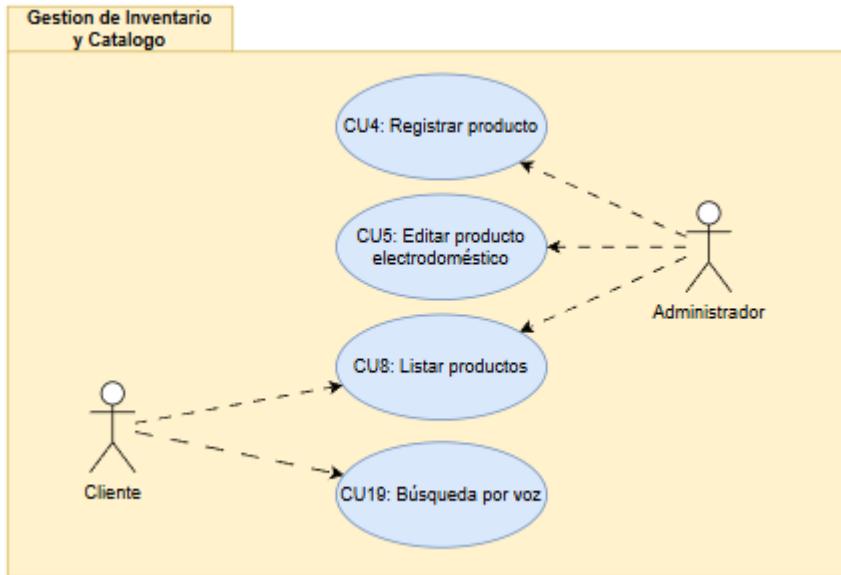
2.2.1.2 Relacionar Paquetes y Casos de Uso

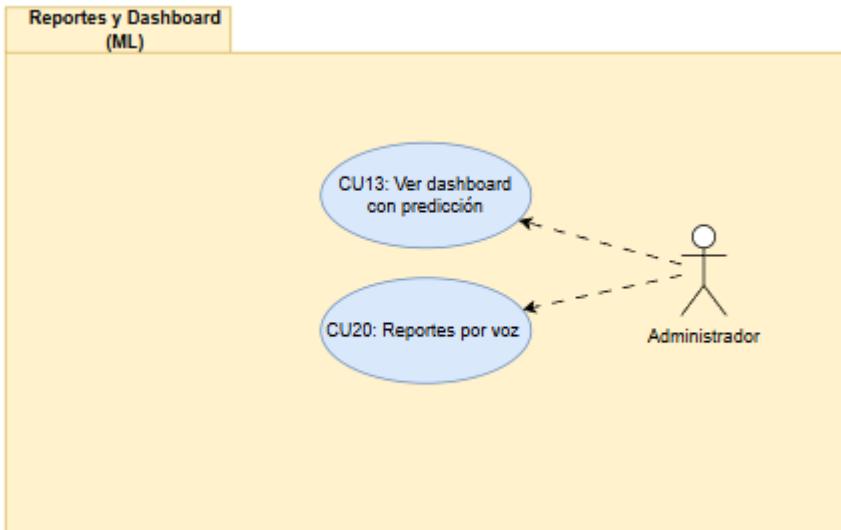




2.2.1.3 Vista de Paquetes



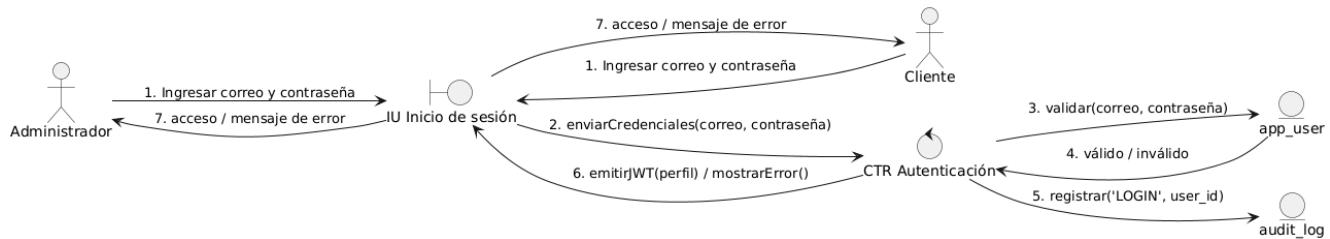




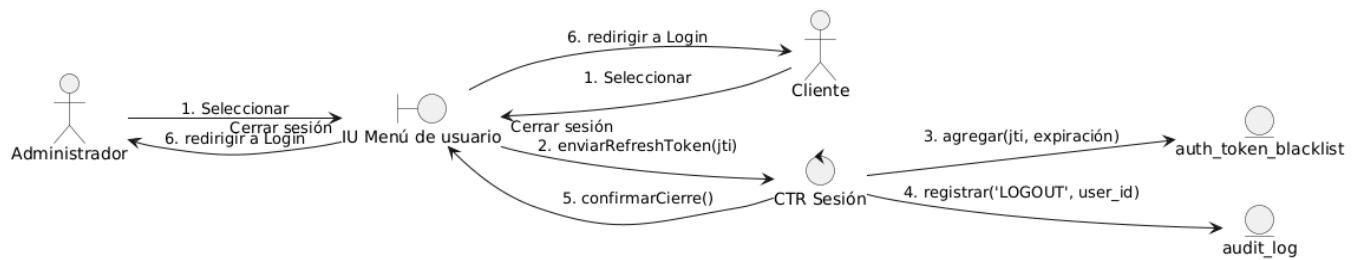
2.2.2 Análisis de Caso de Uso

Ciclo #1

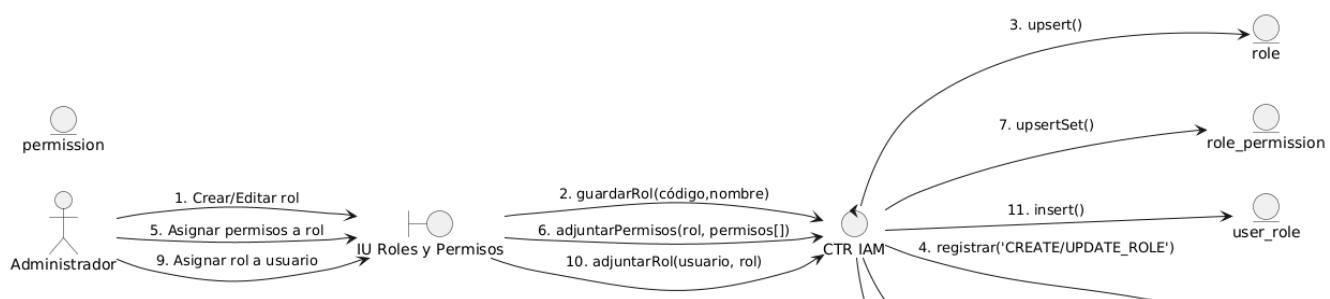
CU01 Iniciar sesión



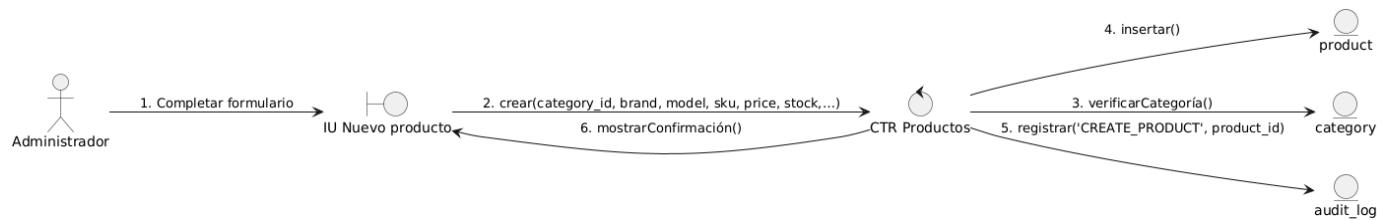
CU02 Cerrar sesión



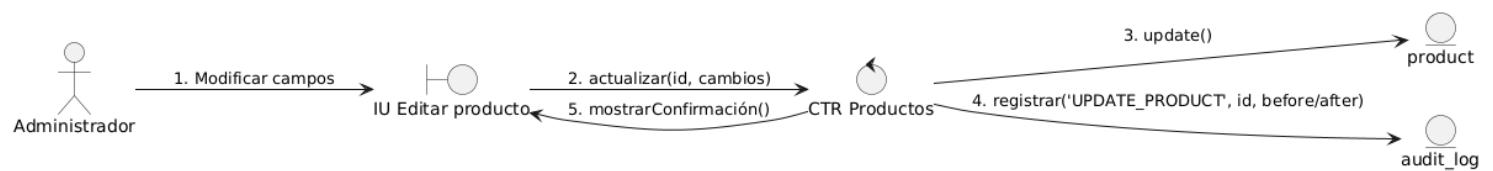
CU03 Gestionar roles y permisos



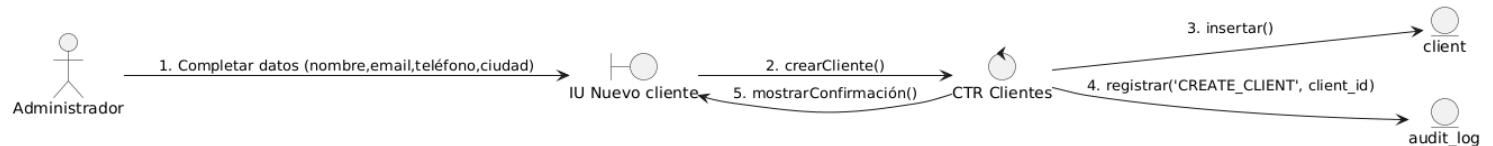
CU04 Registrar producto



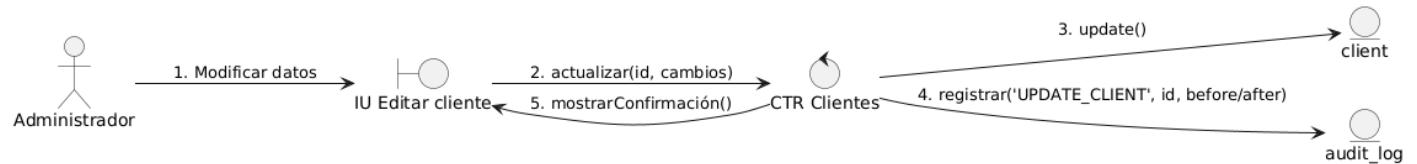
CU05 Editar producto electrodoméstico



CU06 Registrar cliente

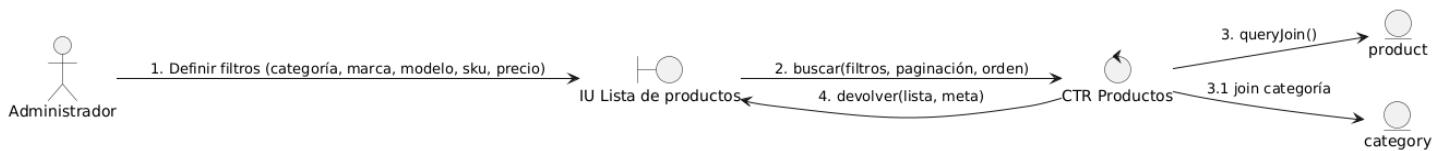


CU07 Editar cliente

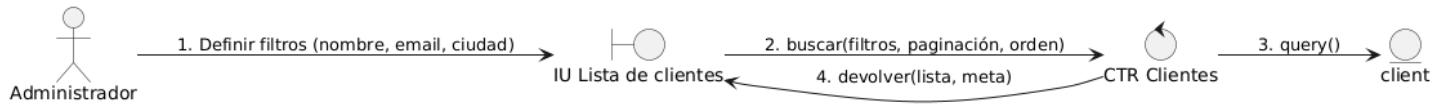


Ciclo #2

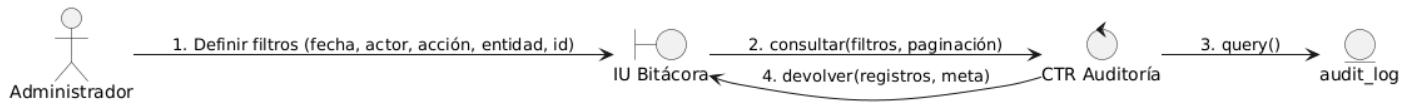
CU08 Listar productos



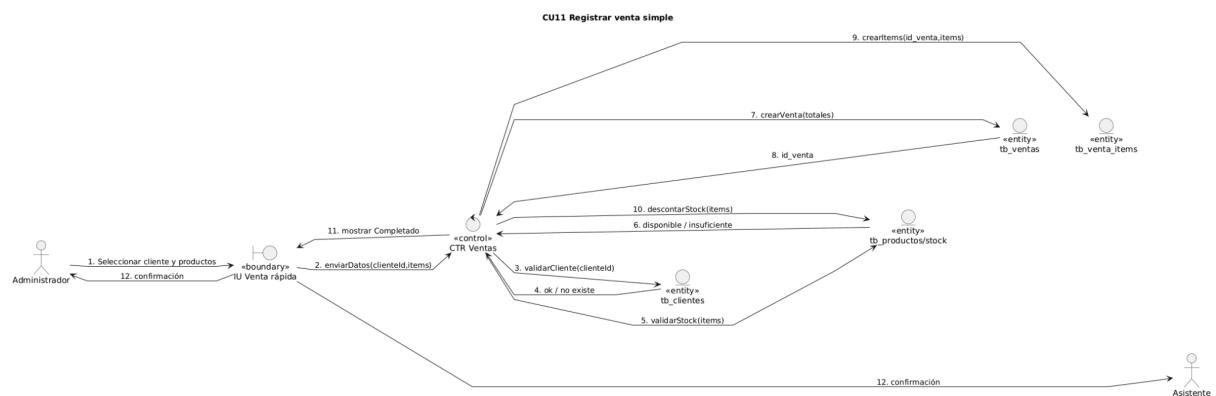
CU09 Listar clientes



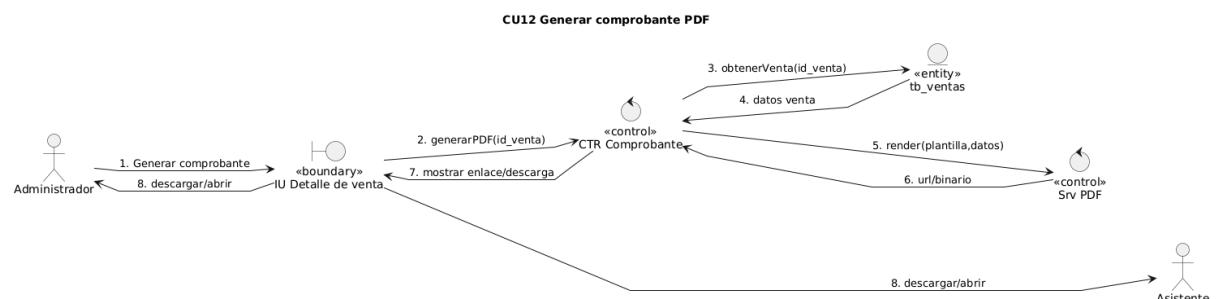
CU10 Bitácora



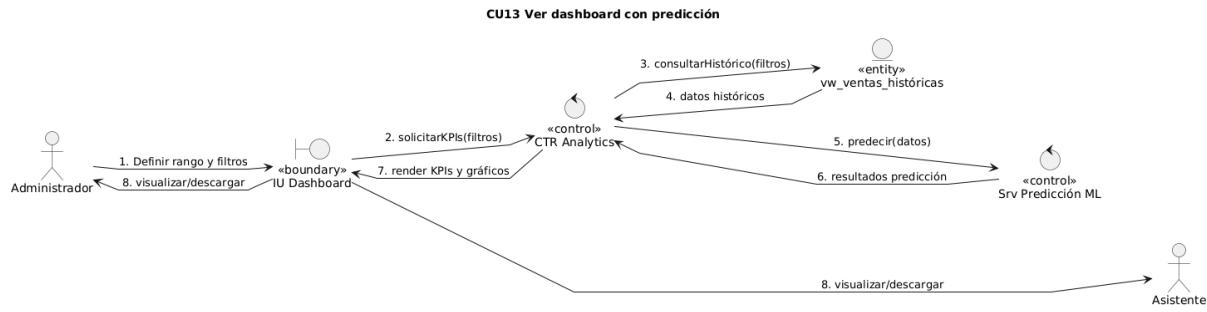
CU11 Registrar venta simple



CU12 Generar comprobante PDF

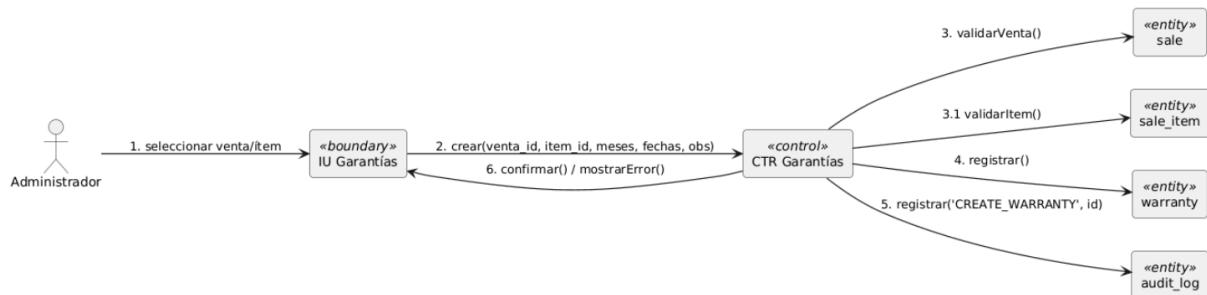


CU13 Ver dashboard con predicción



Ciclo #3

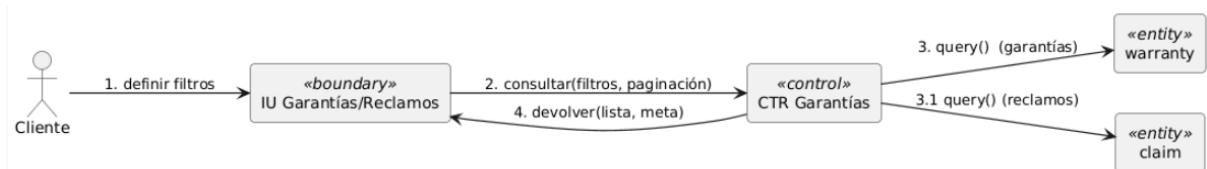
CU14 Registrar garantía



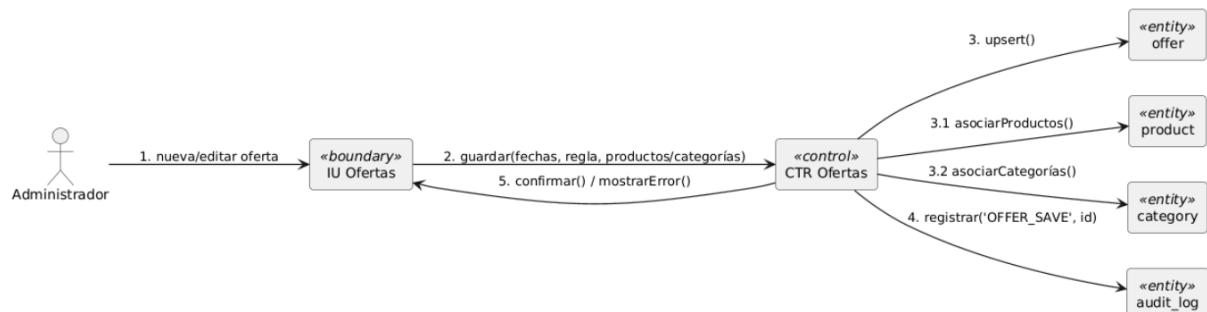
CU15 Registrar reclamo de garantía



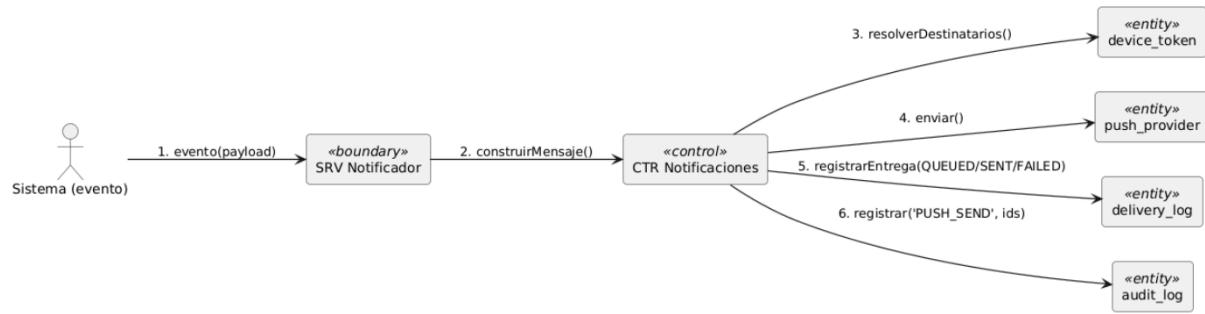
CU16 Consultar estado de garantía/reclamo



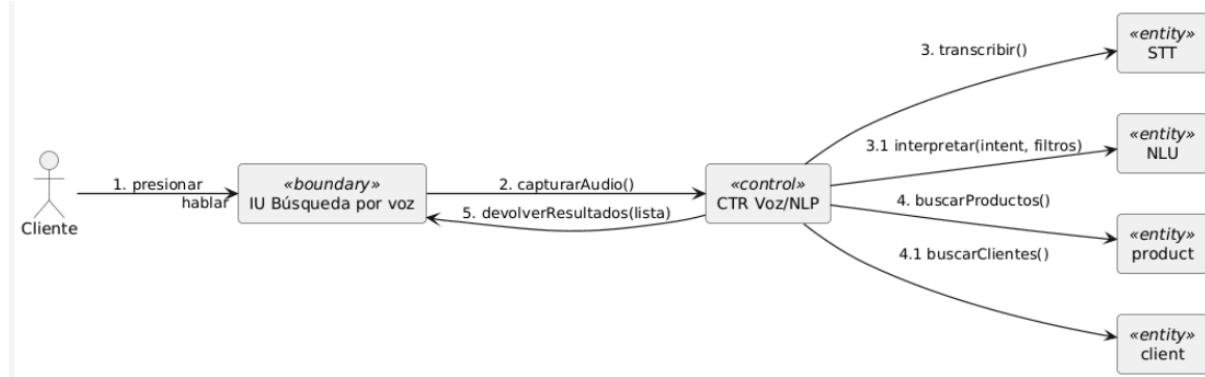
CU17 Gestionar ofertas



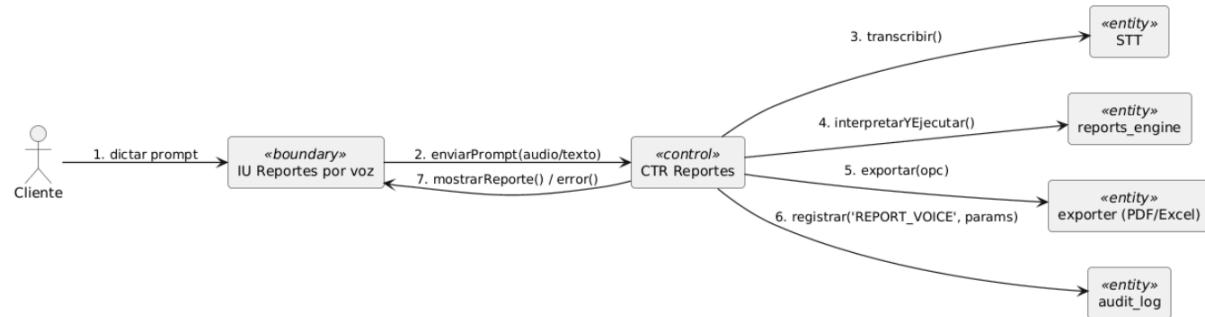
CU18 Notificaciones push



CU19 Búsqueda por voz (producto/cliente)



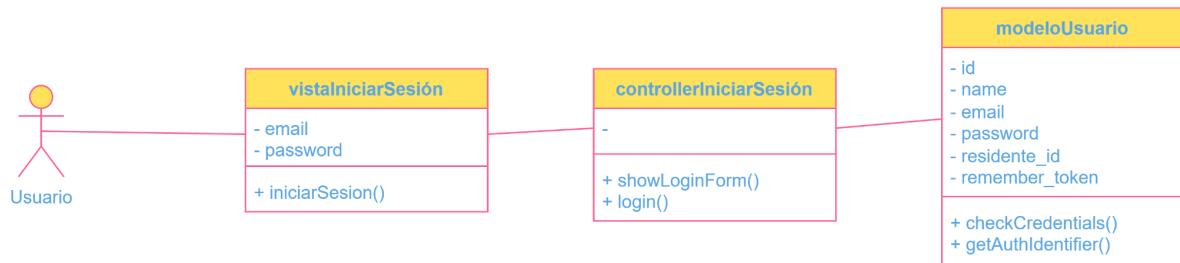
CU20 Reportes por voz (prompts)



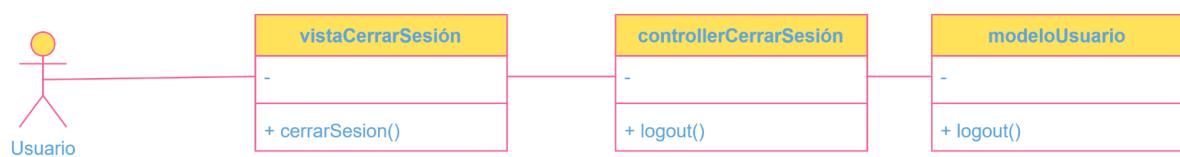
2.2.3 Análisis de una Clase

Ciclo #1

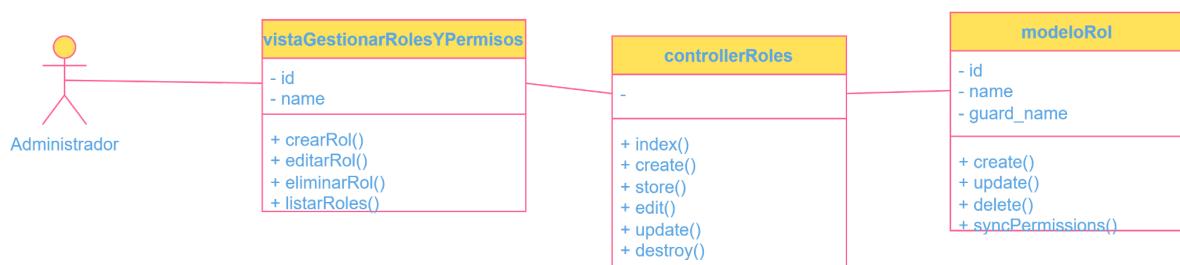
CU1 Iniciar sesión



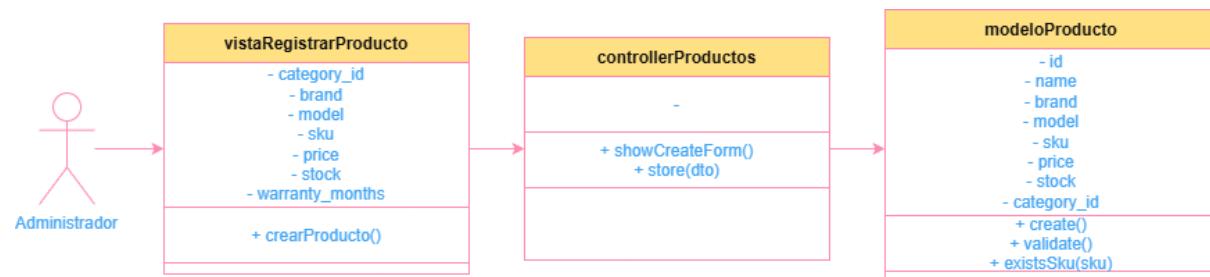
CU2 Cerrar sesión



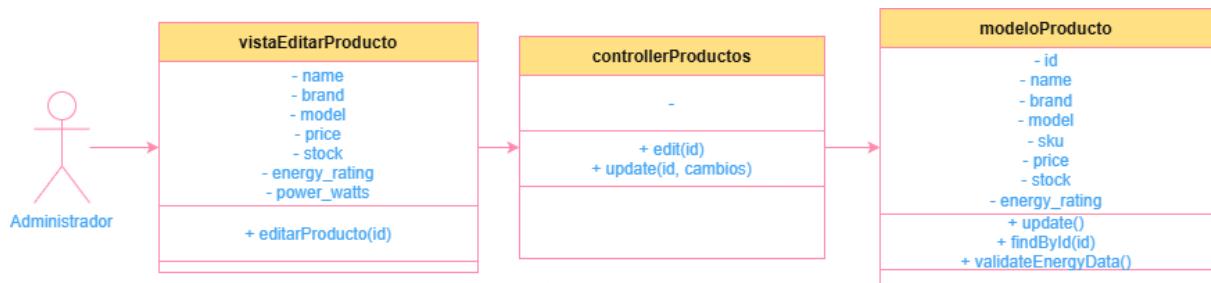
CU3 Gestionar roles y permisos



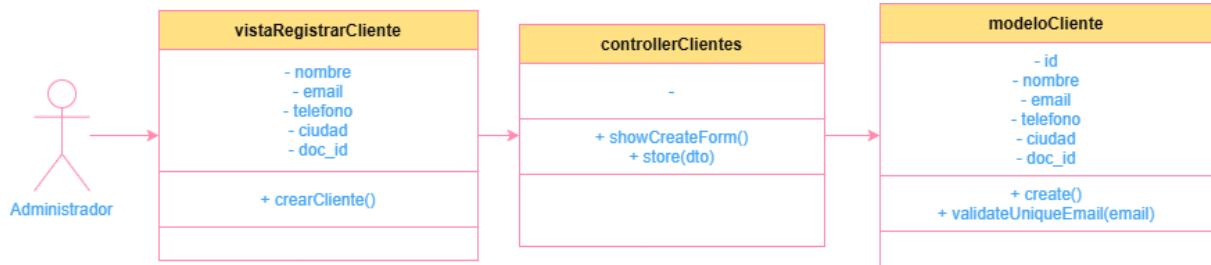
CU04 Registrar producto



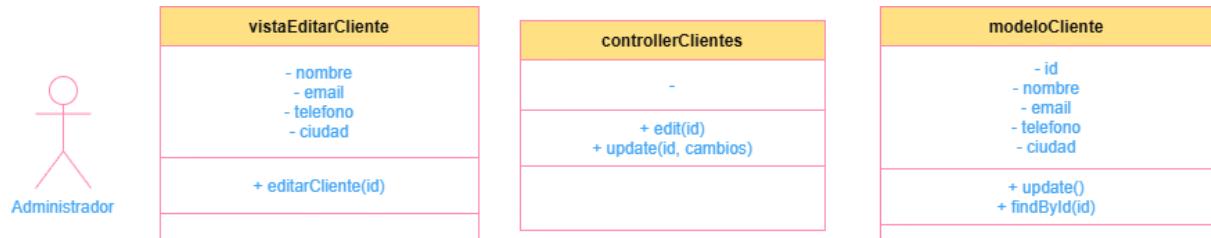
CU05 Editar producto electrodoméstico



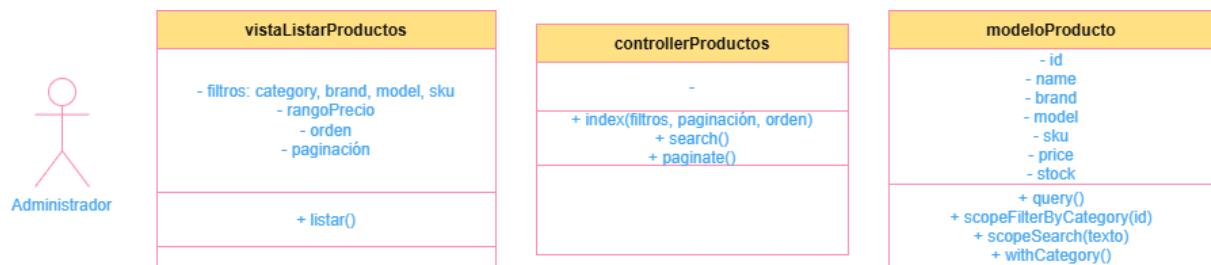
CU06 Registrar cliente



CU07 Editar cliente

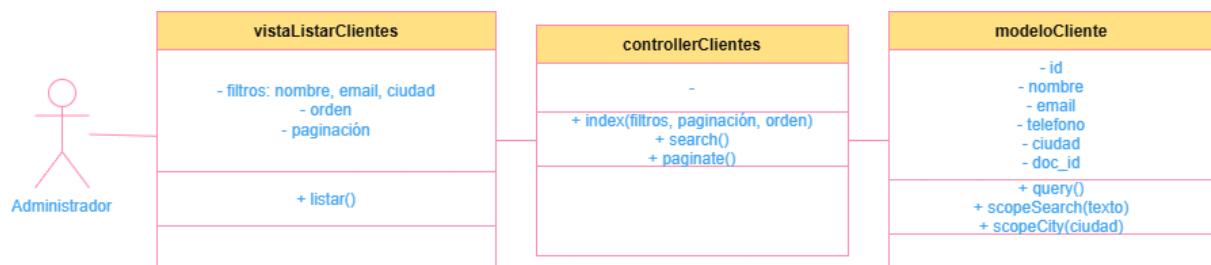


CU08 Listar productos

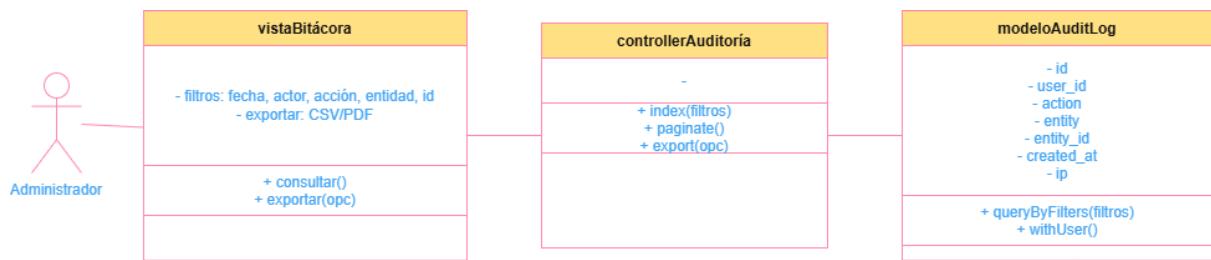


Ciclo #2

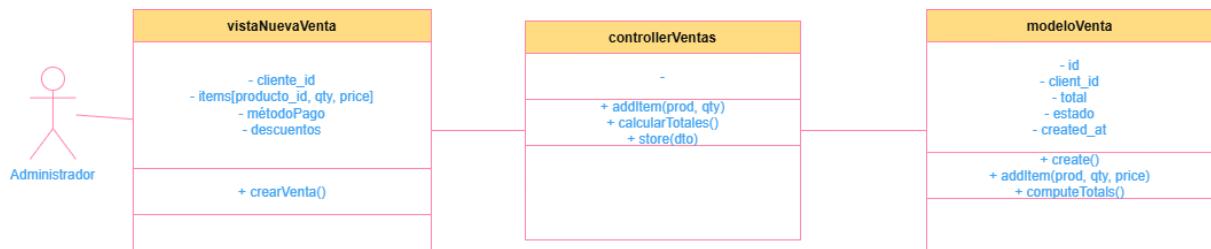
CU09 Listar clientes



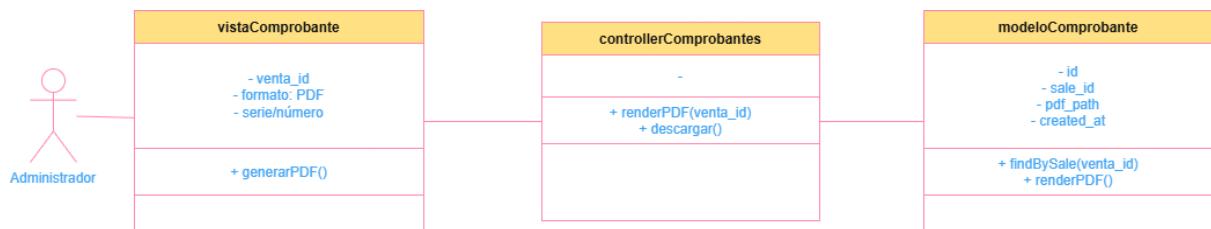
CU10 Bitácora



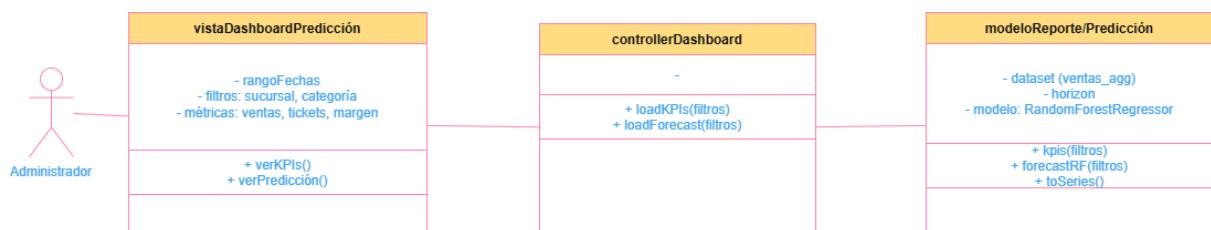
CU11 Registrar venta simple



CU12 Generar comprobante PDF

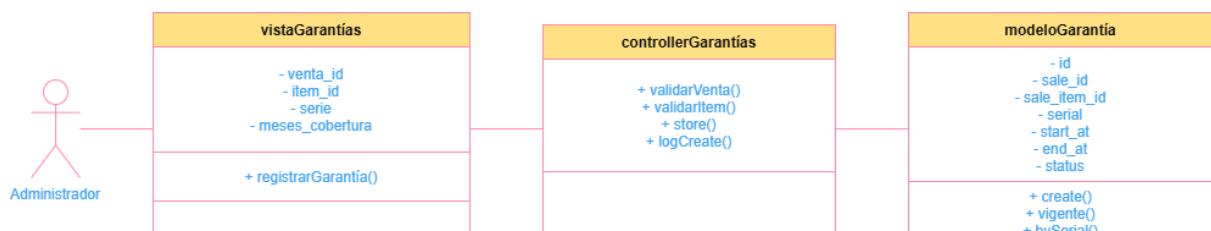


CU13 Ver dashboard con predicción

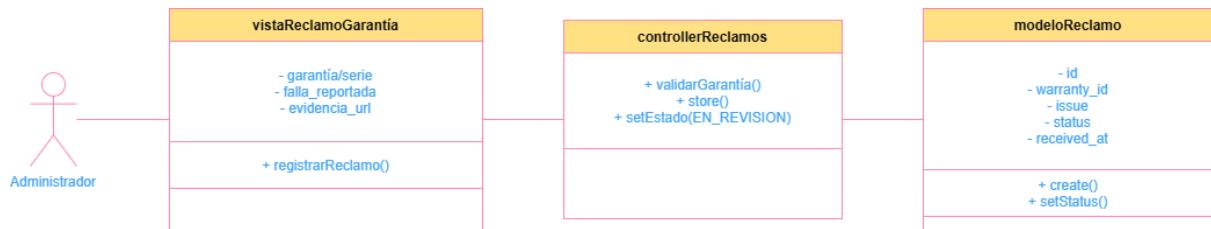


Ciclo #3

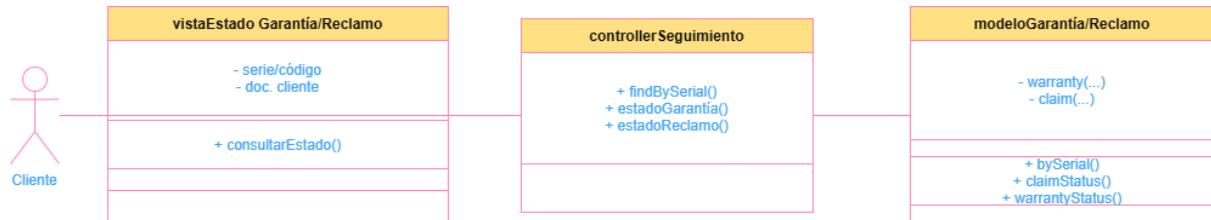
CU14 Registrar garantía



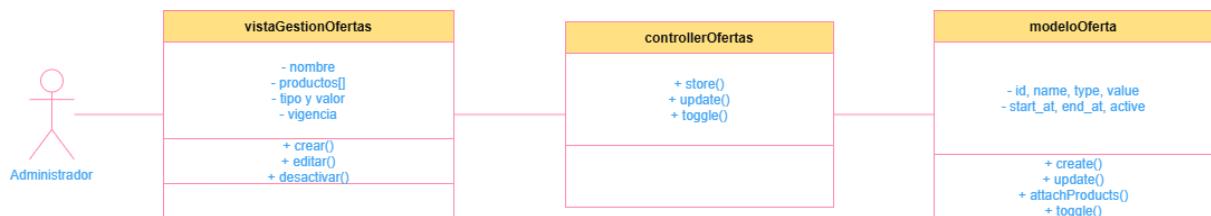
CU15 Registrar reclamo de garantía



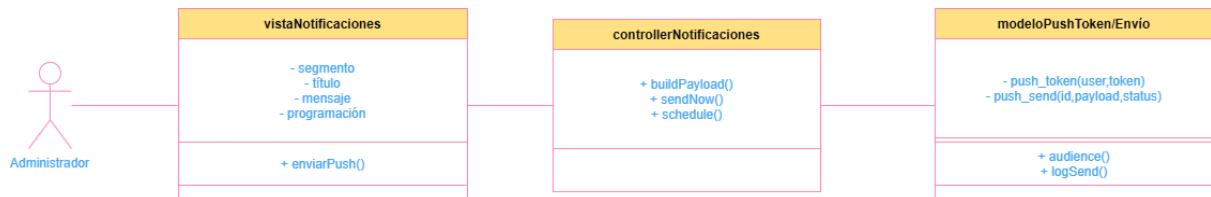
CU16 Consultar estado de garantía/reclamo



CU17 Gestionar ofertas



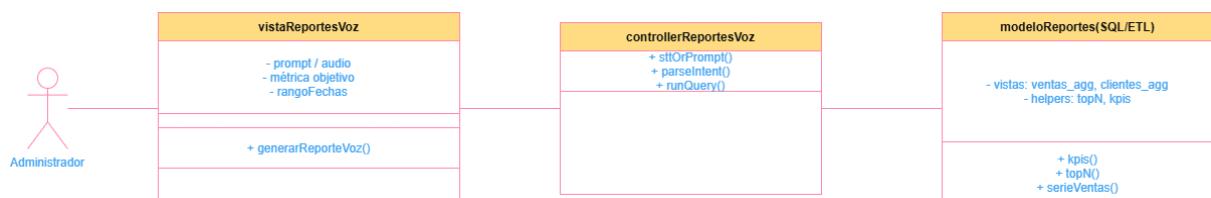
CU18 Notificaciones push



CU19 Búsqueda por voz (producto/cliente)

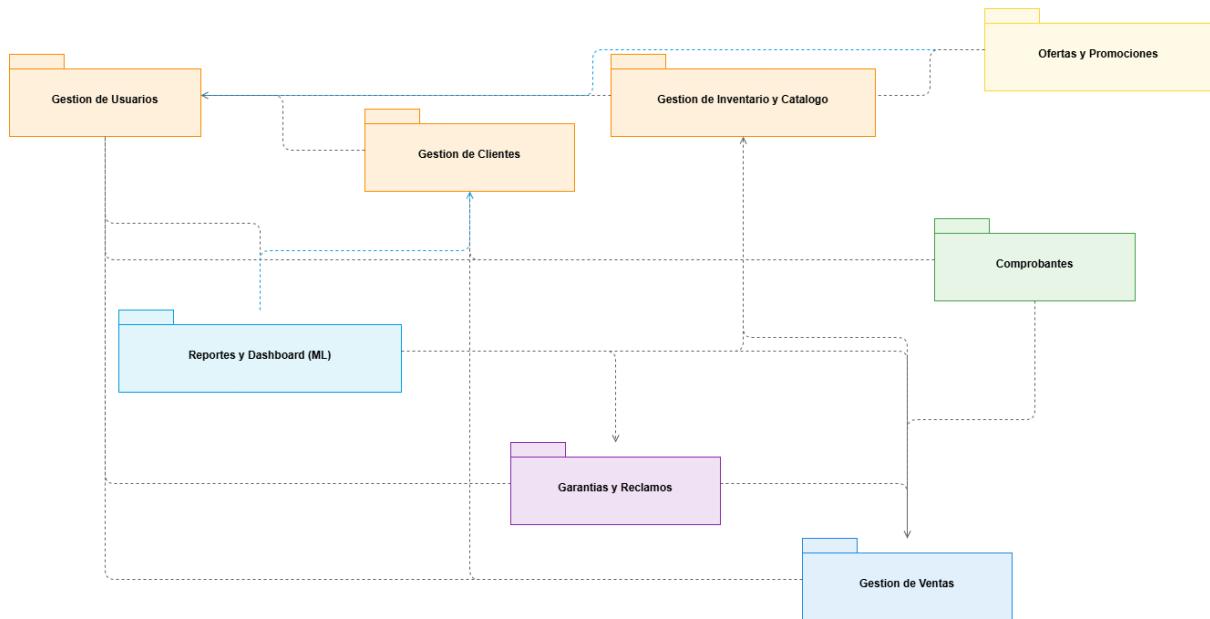


CU20 Reportes por voz (prompts)



2.2.4 Analizar paquete

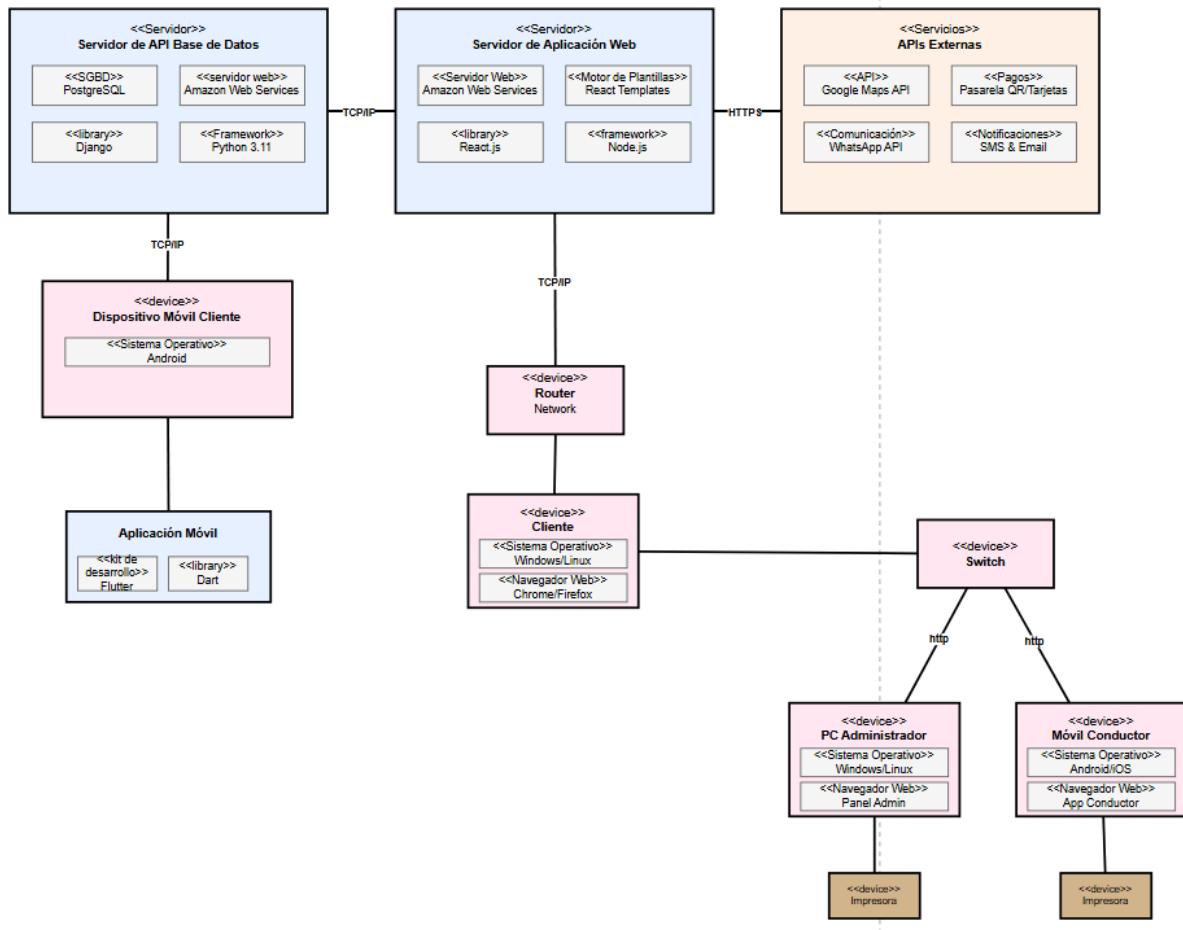
Analisis de Paquetes (UML)



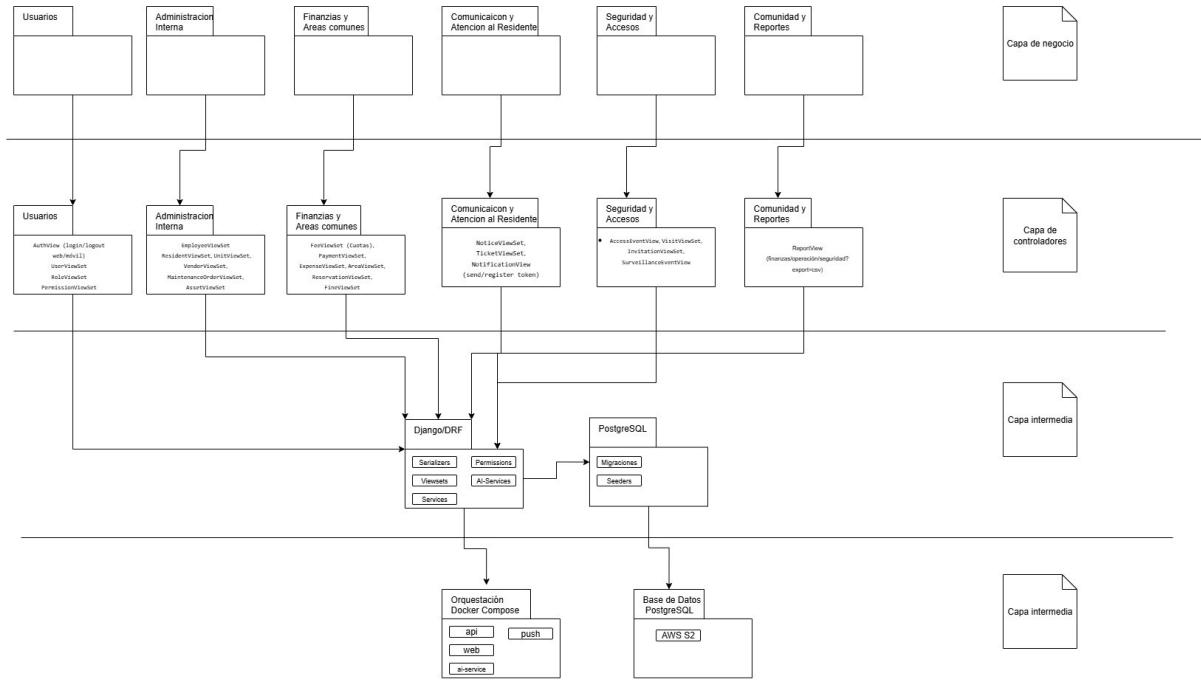
2.3 Flujo de Trabajo: Diseño

2.3.1 Diseño de Arquitectura

2.3.1.1 Diseño físico (*Diagrama de despliegue*)



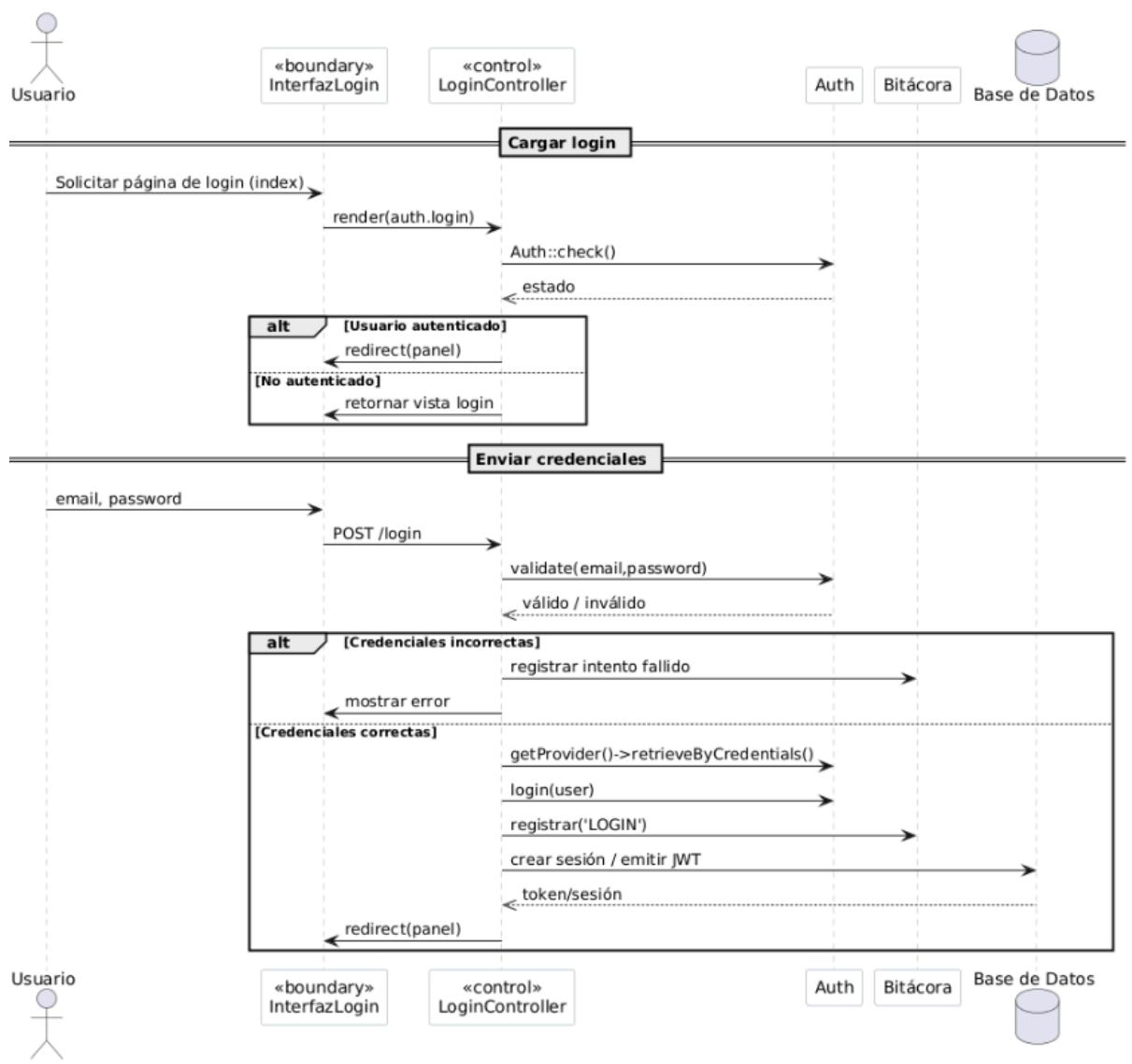
2.3.1.2 Diseño Lógico (Diagrama de Paquete)



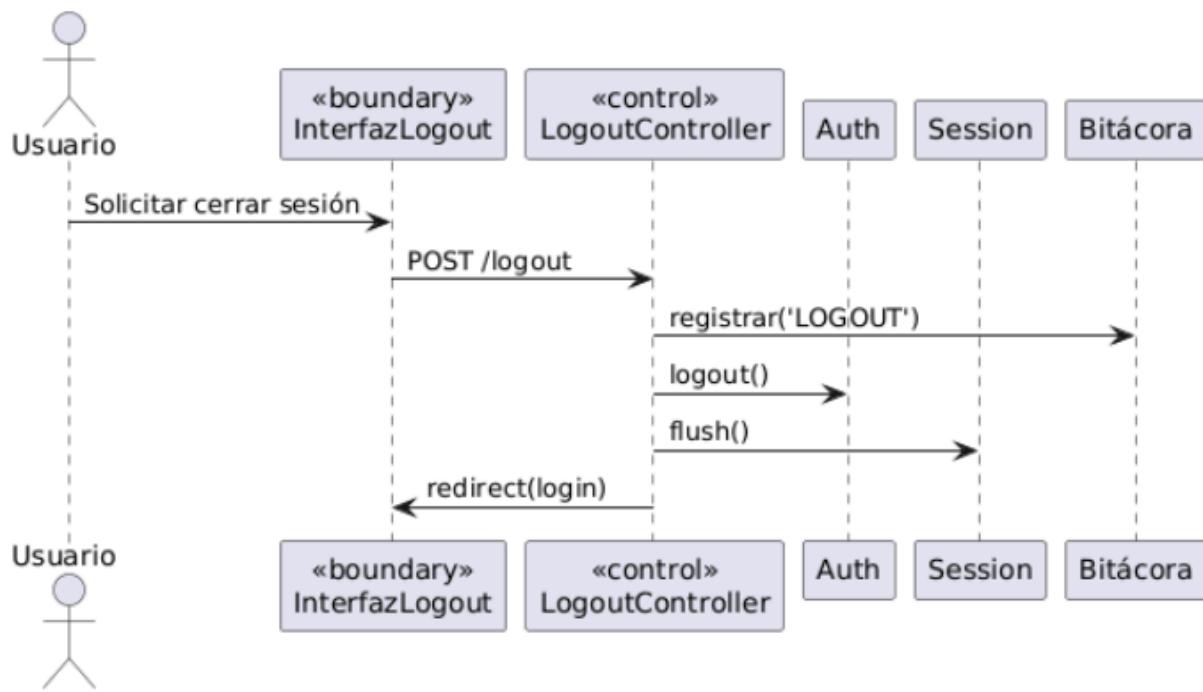
2.3.2 Diagrama de Secuencia

Ciclo #1

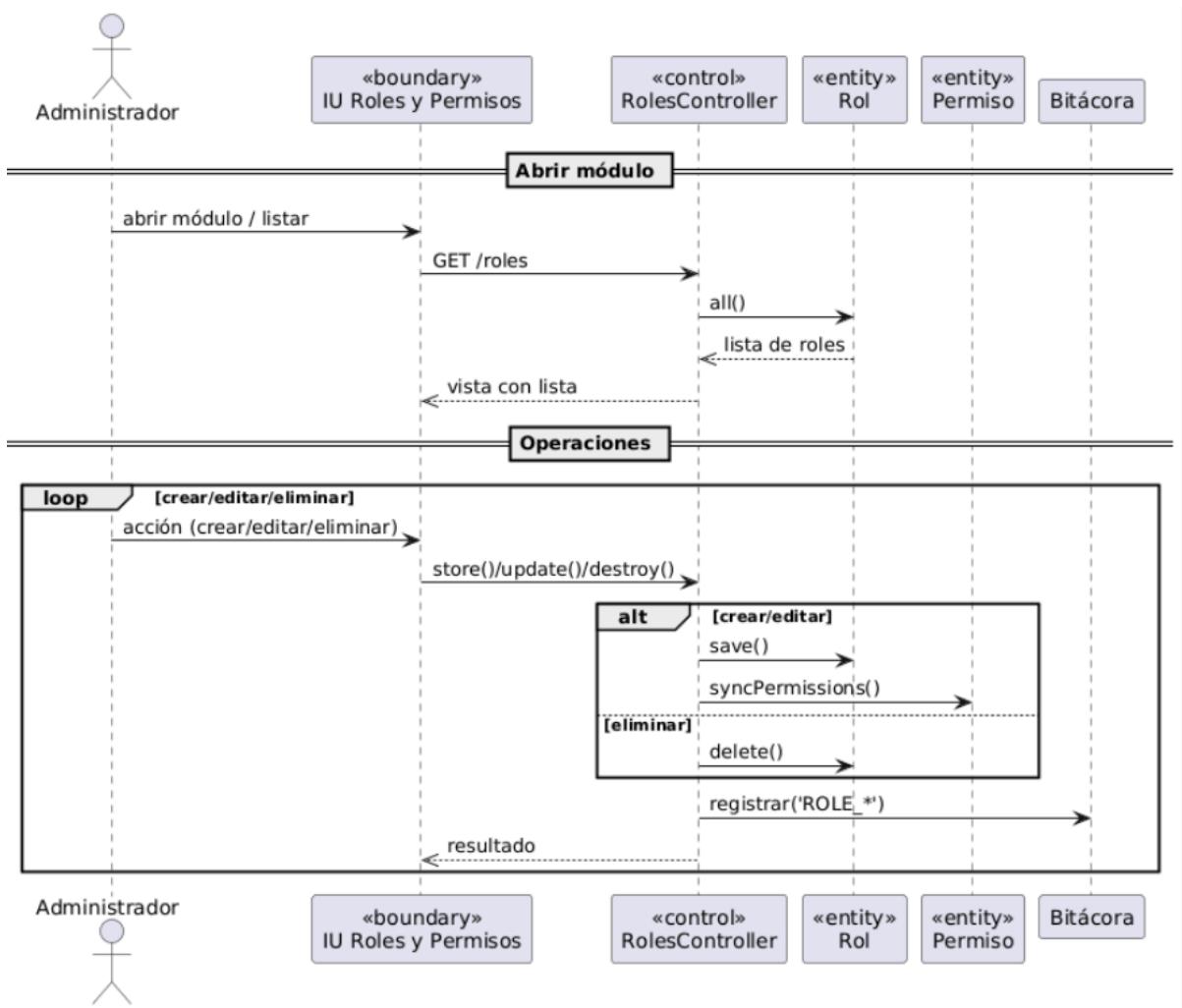
CU01 Iniciar sesión



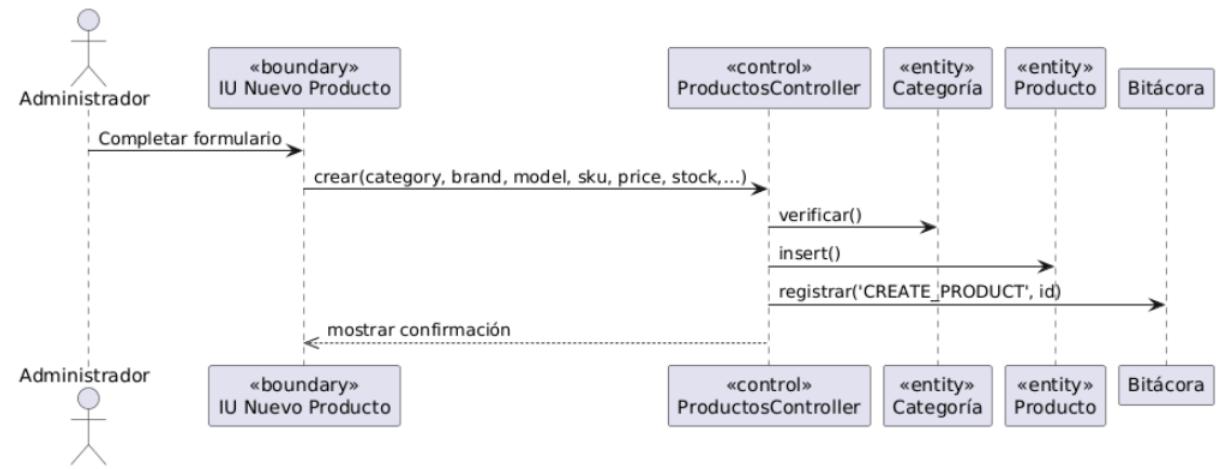
CU02 Cerrar sesión



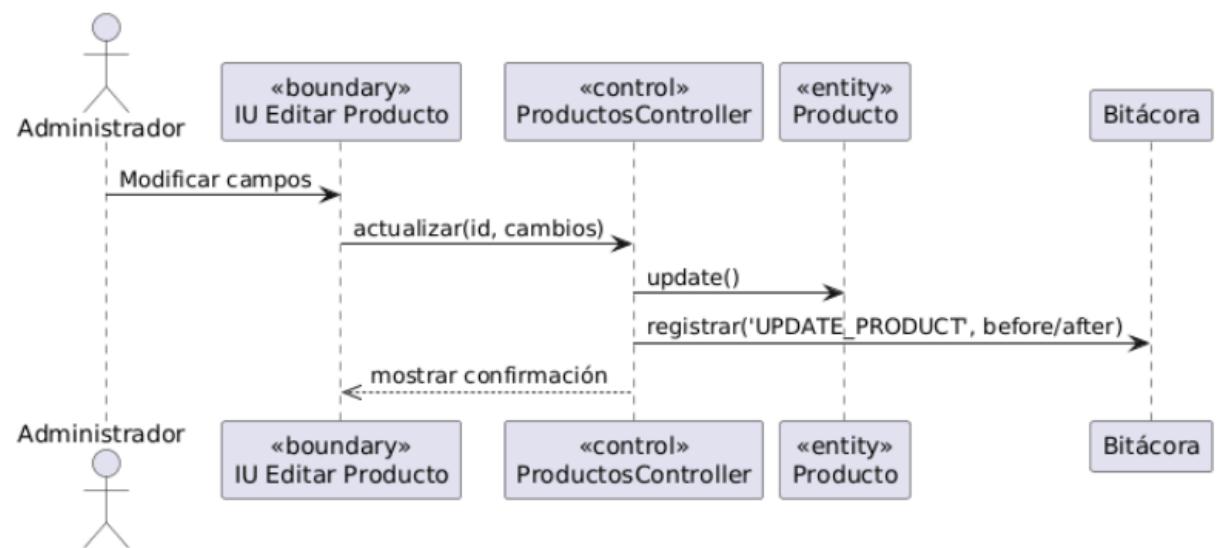
CU03 Gestionar roles y permisos



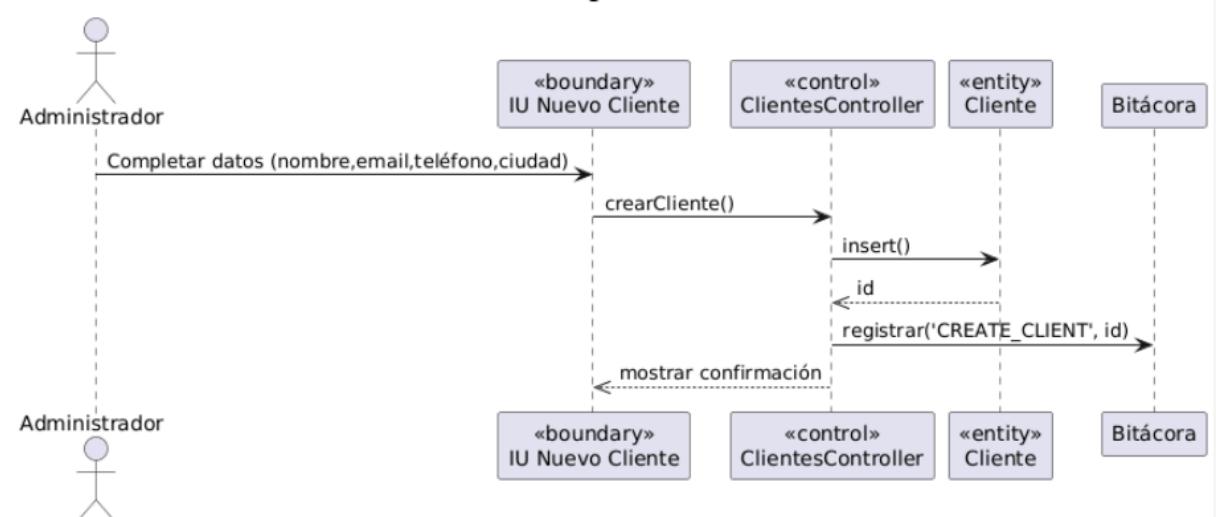
CU04 Registrar producto



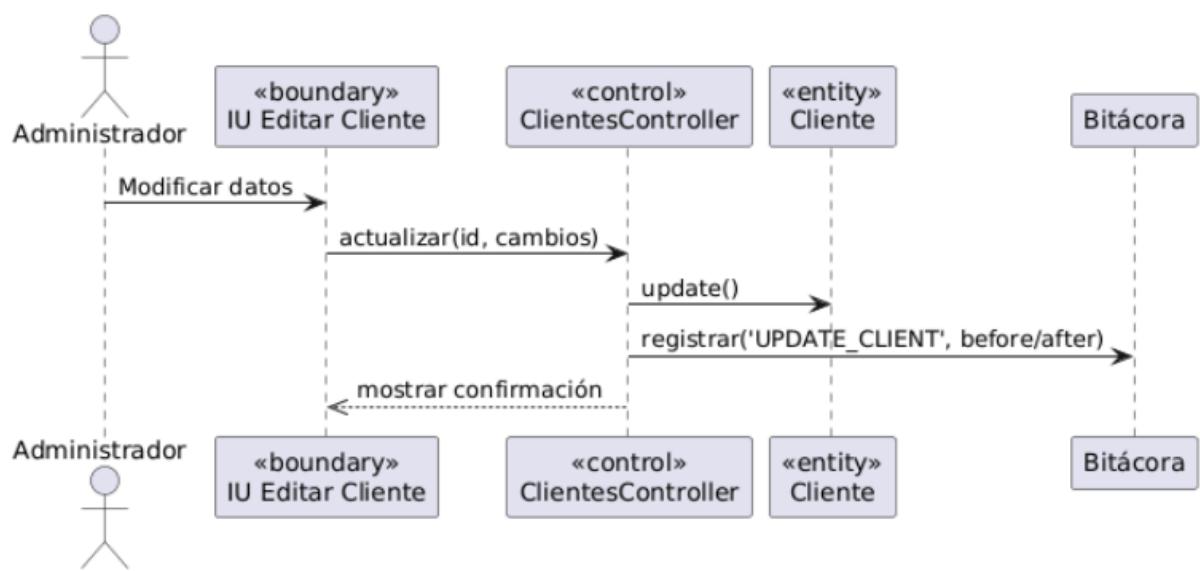
CU05 Editar producto electrodoméstico



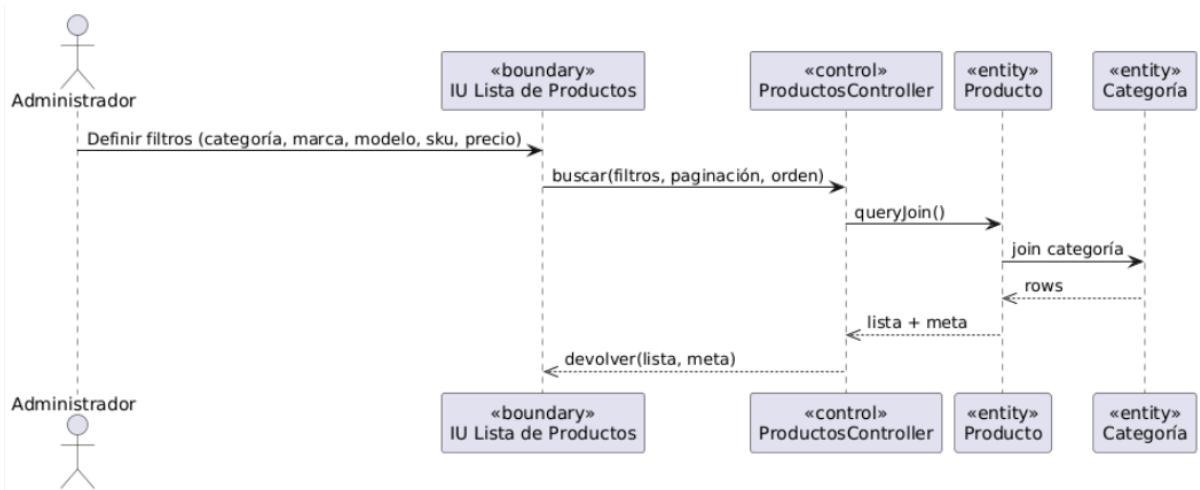
CU06 Registrar cliente



CU07 Editar cliente

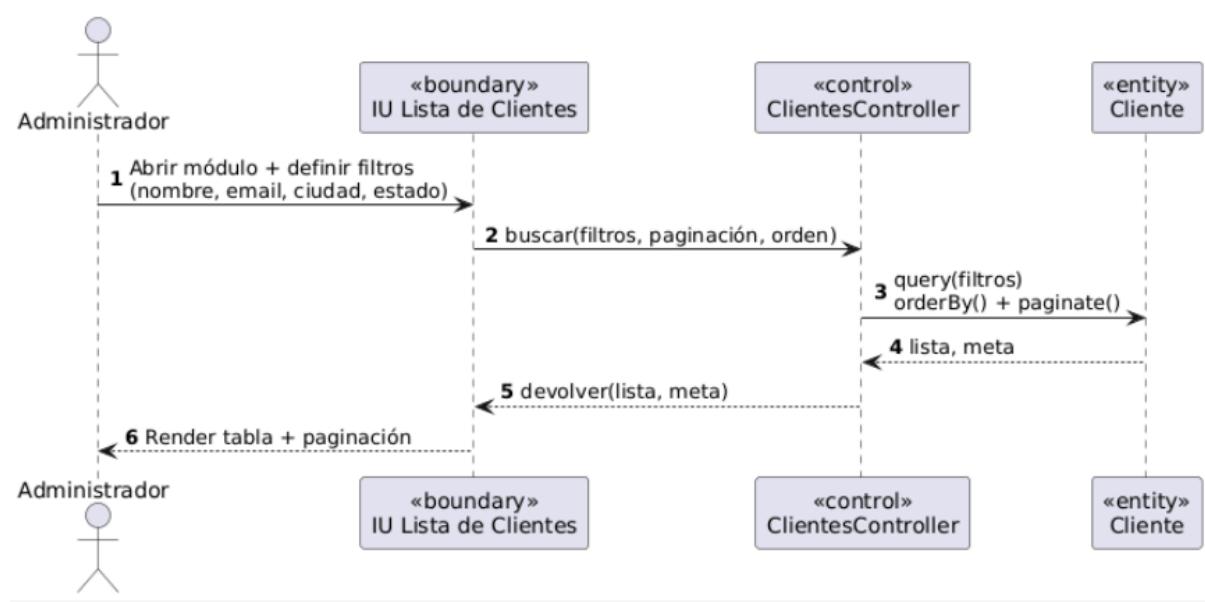


CU08 Listar productos

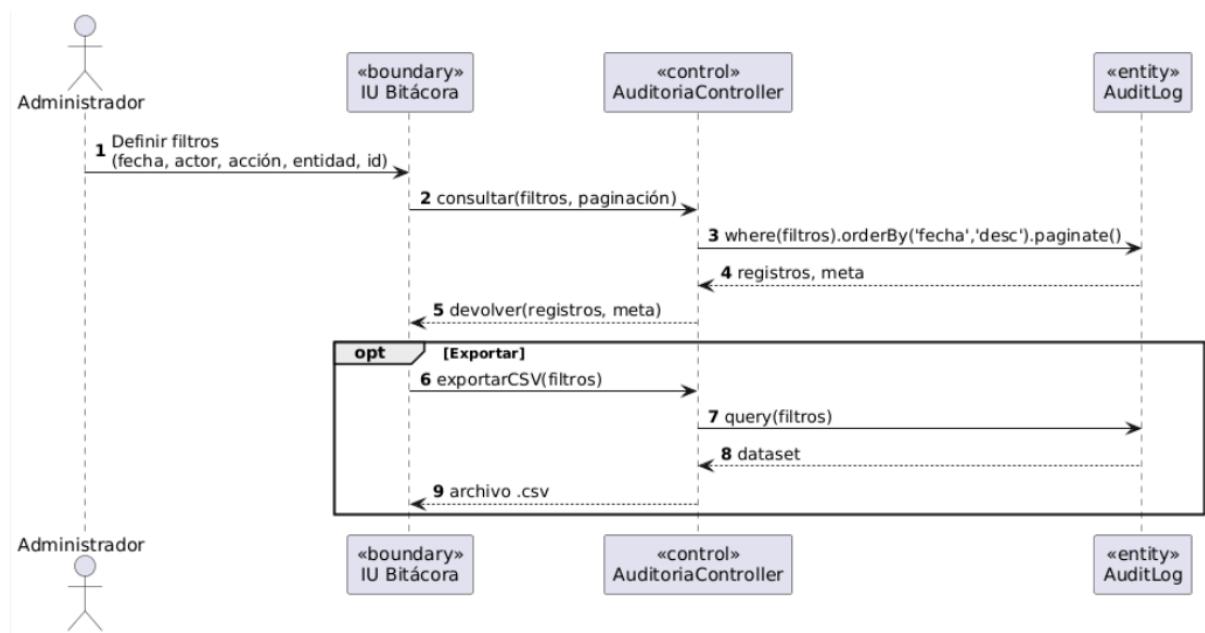


Ciclo #2

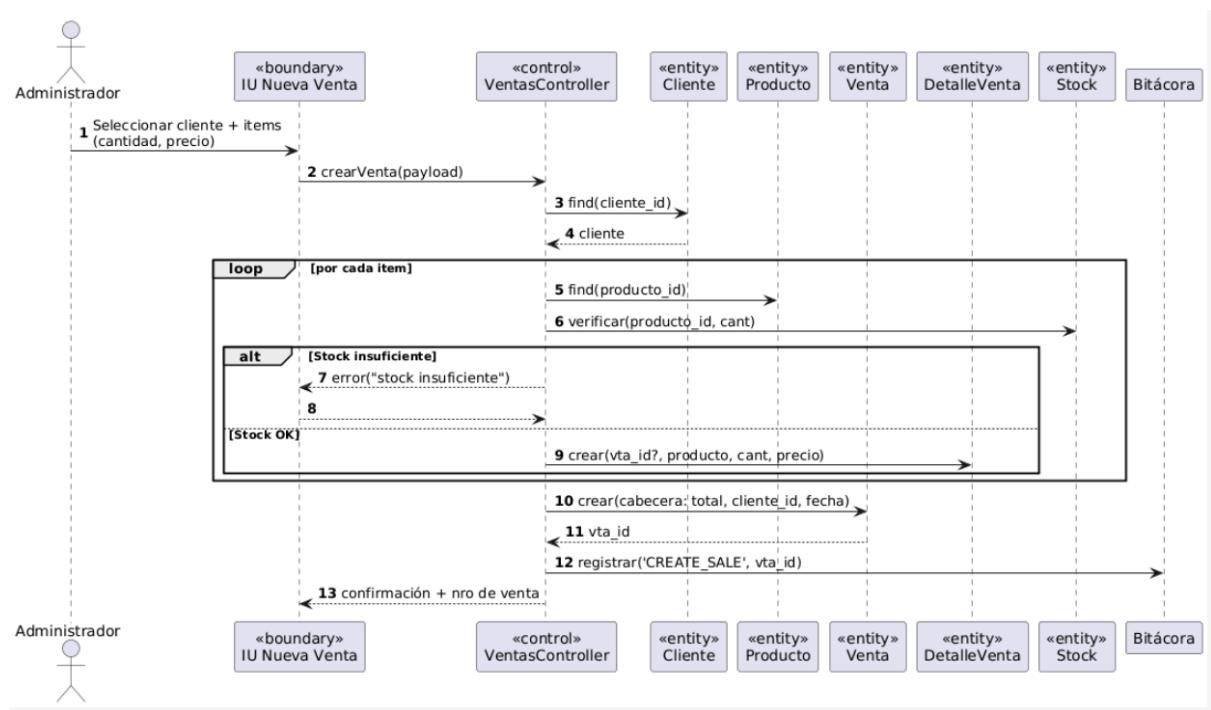
CU09 Listar clientes



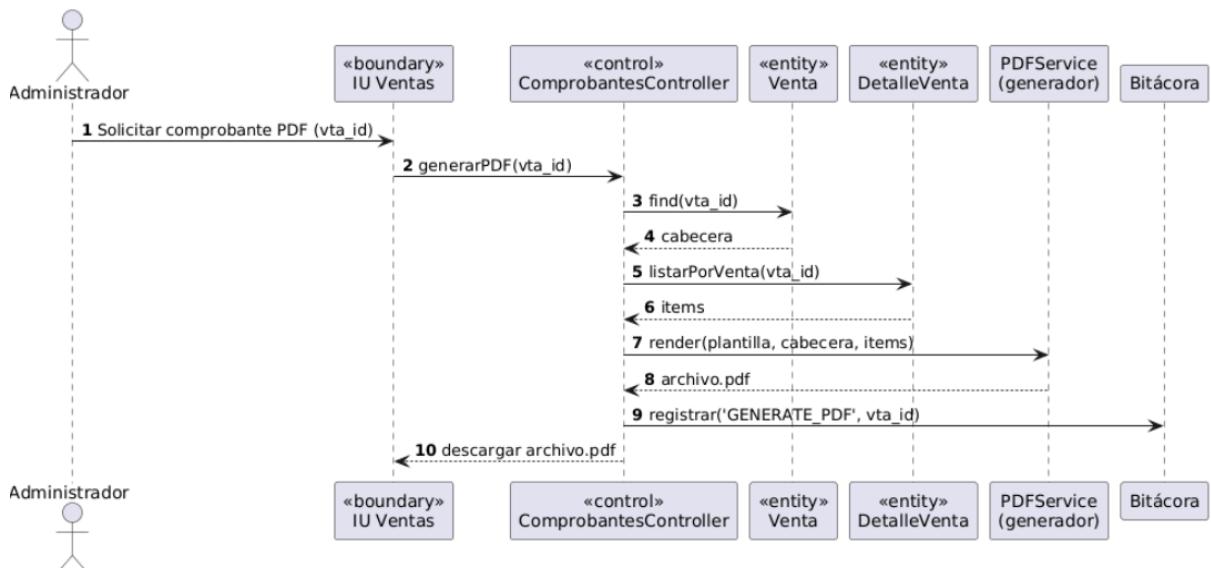
CU10 Bitácora

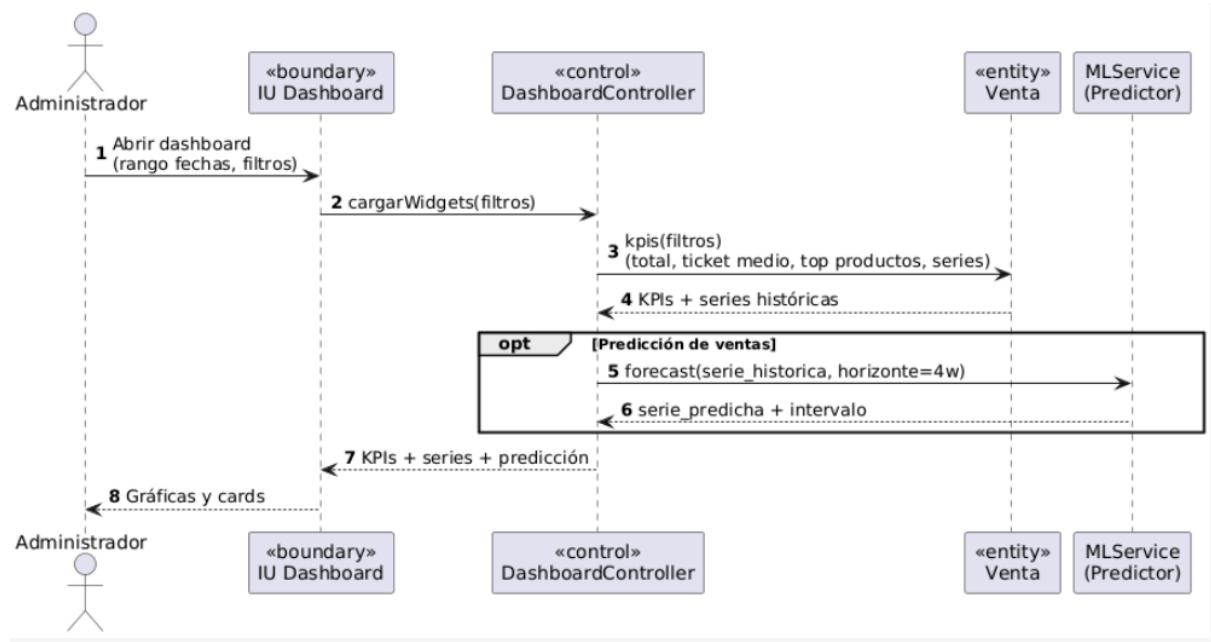


CU11 Registrar venta simple



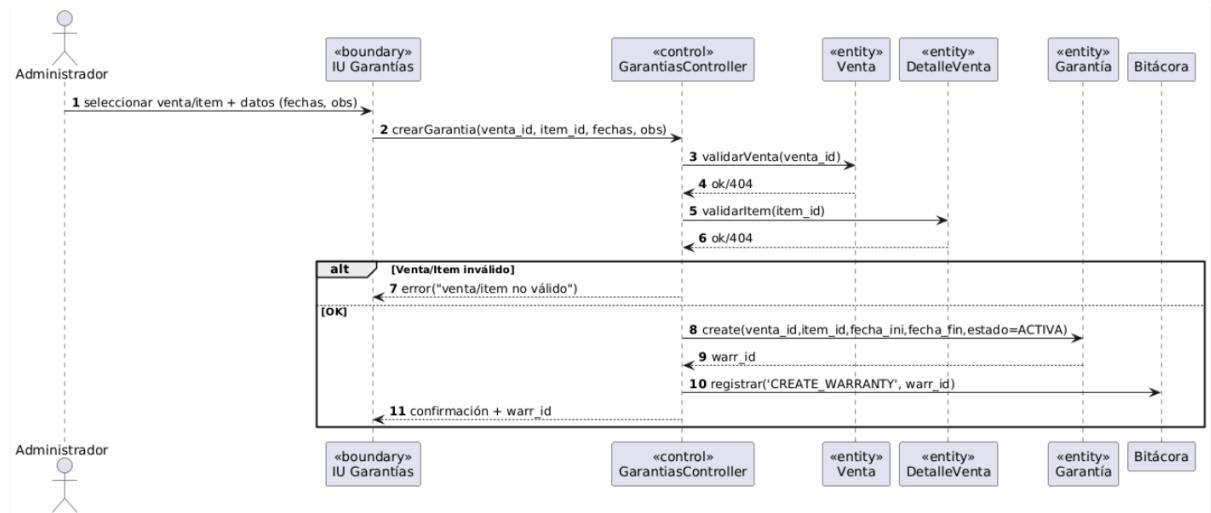
CU12 Generar comprobante PDF



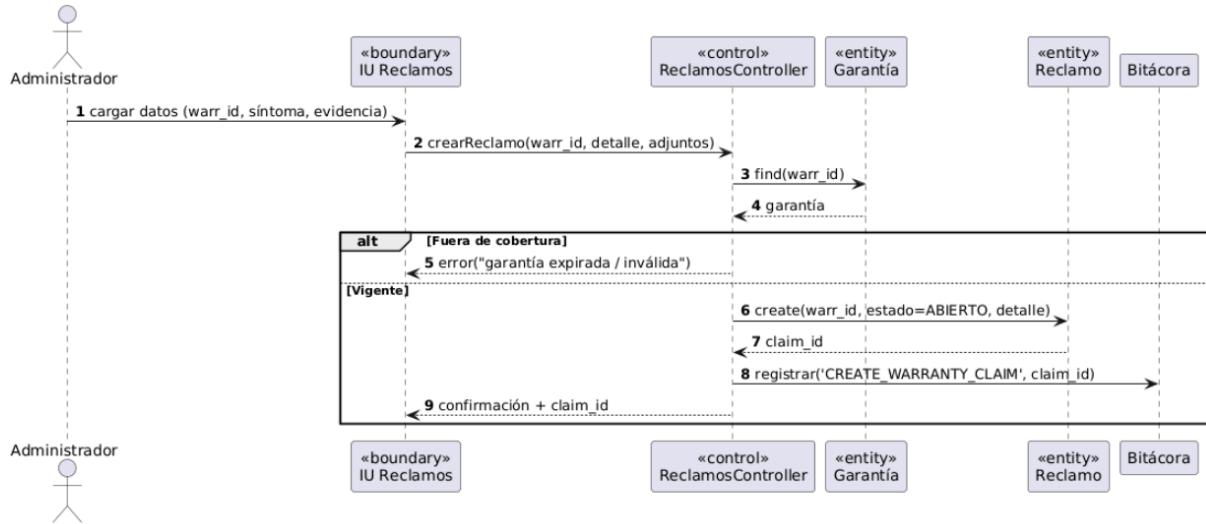


Ciclo #3

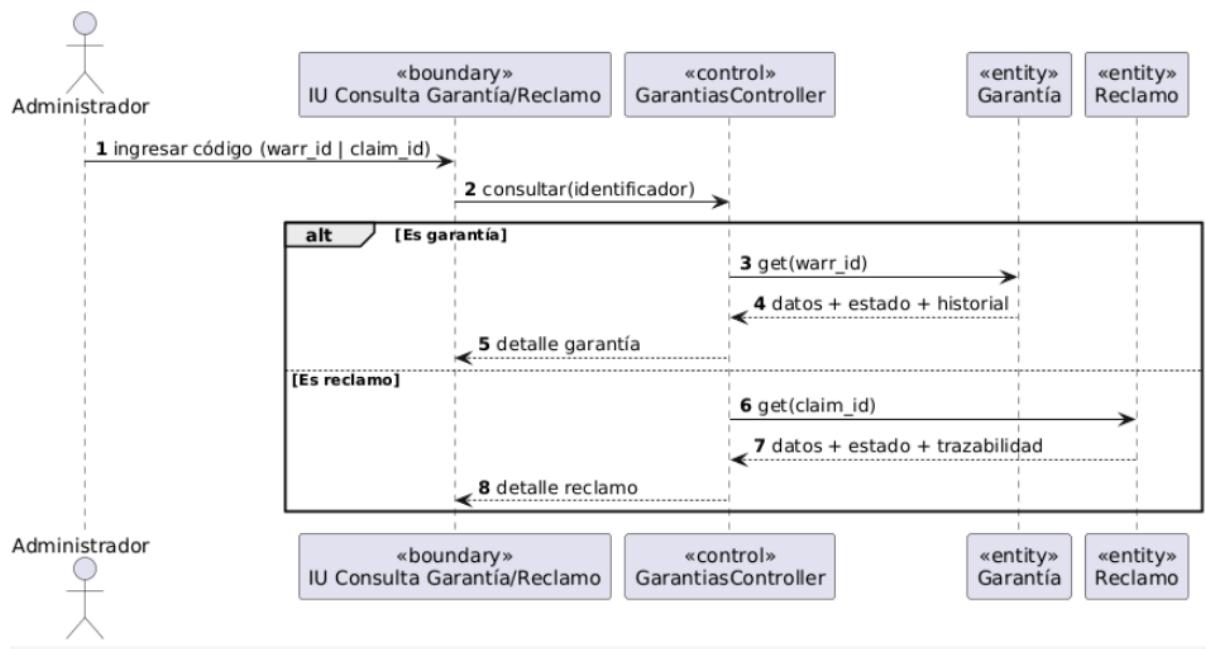
CU14 Registrar garantía



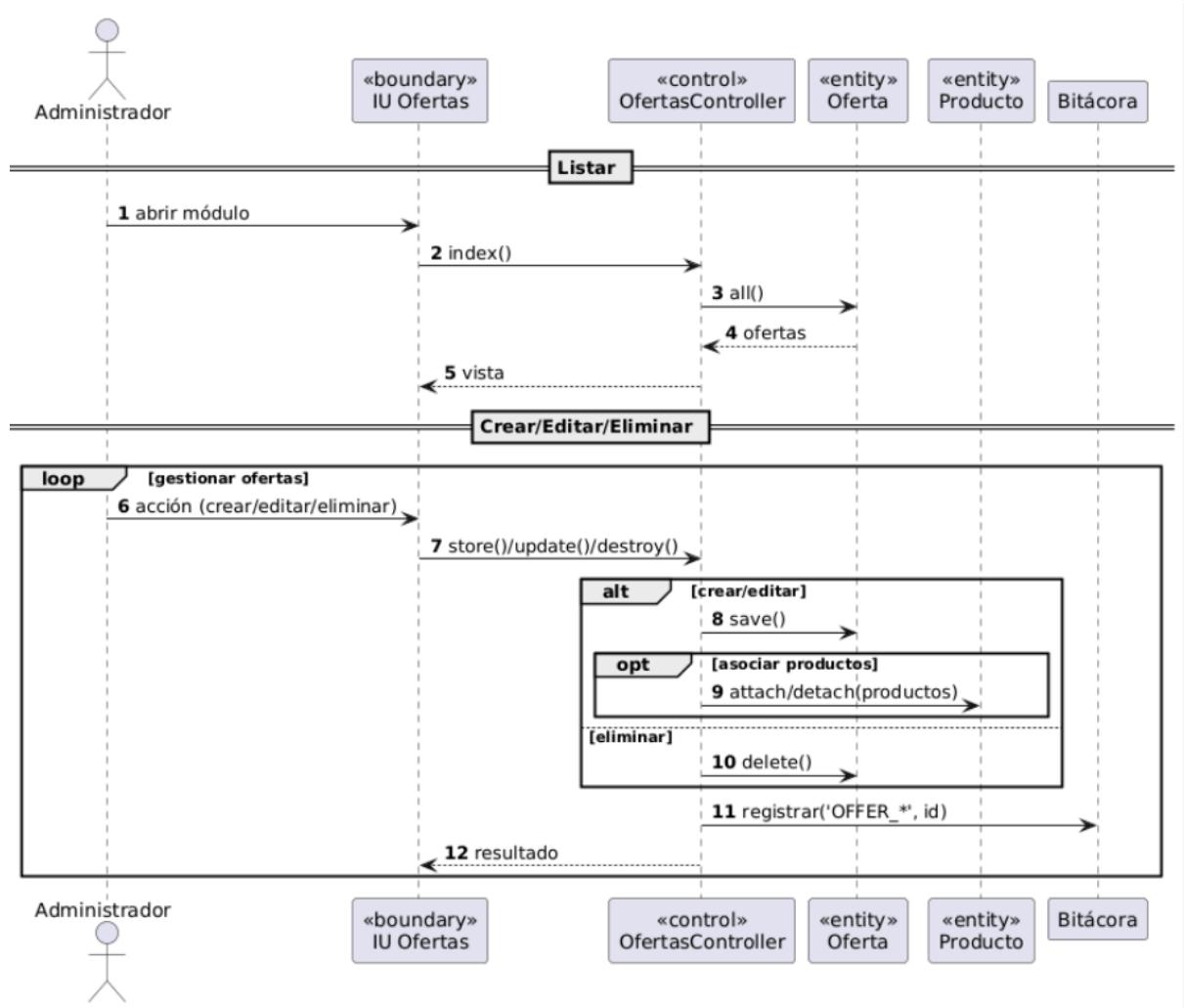
CU15 Registrar reclamo de garantía



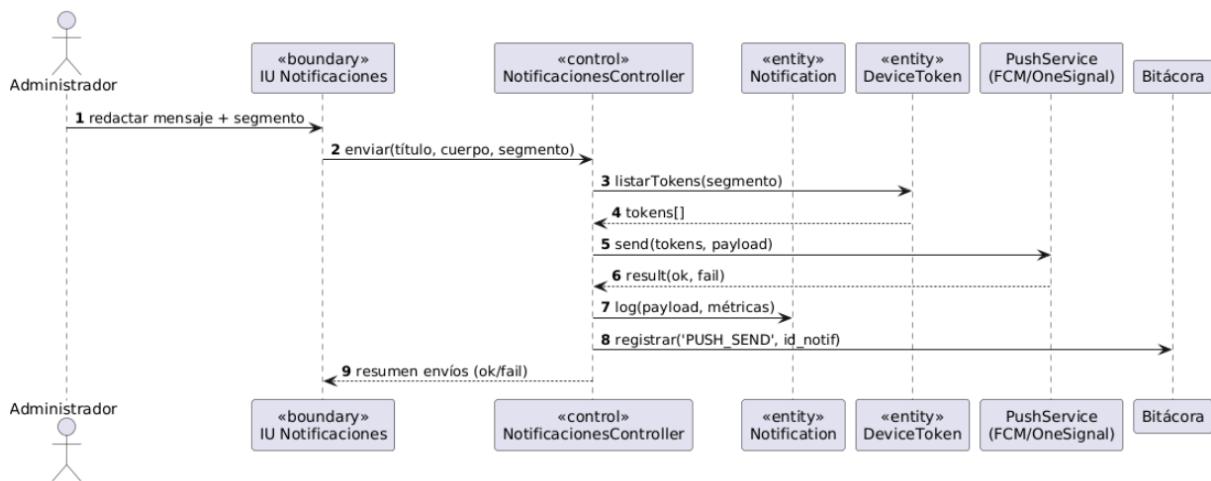
CU16 Consultar estado de garantía/reclamo



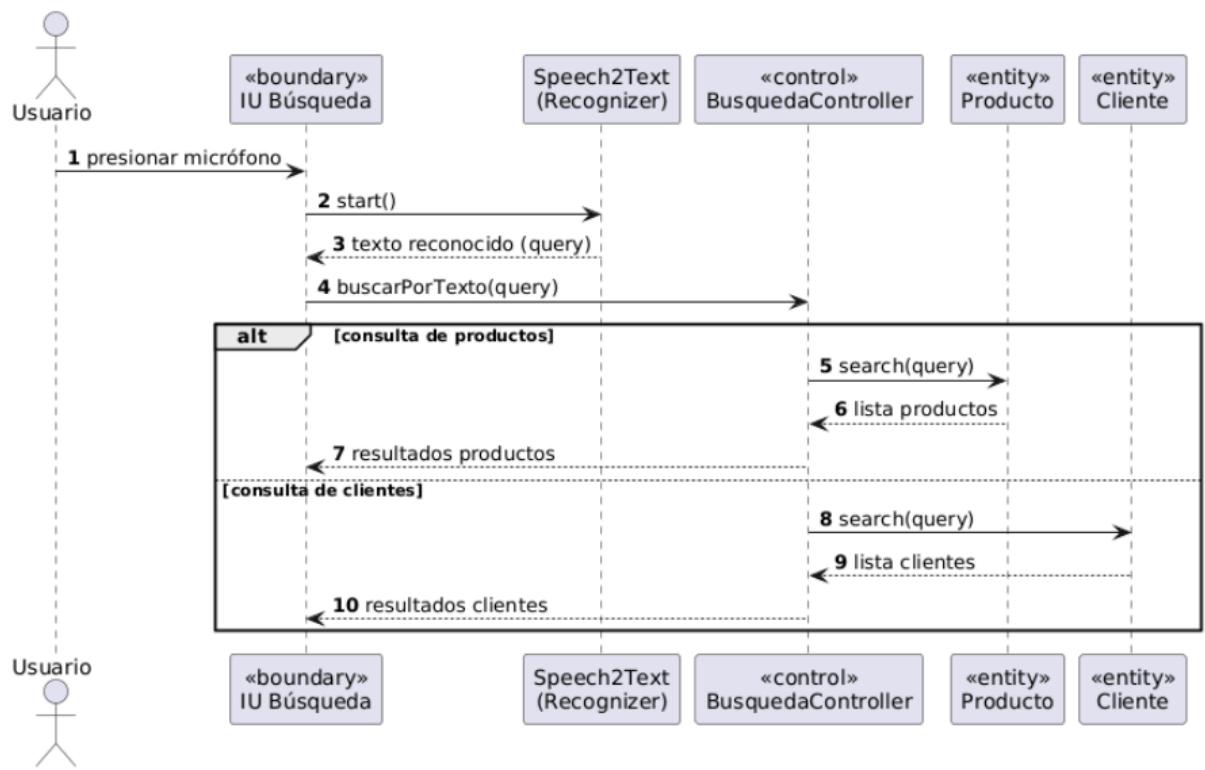
CU17 Gestionar ofertas



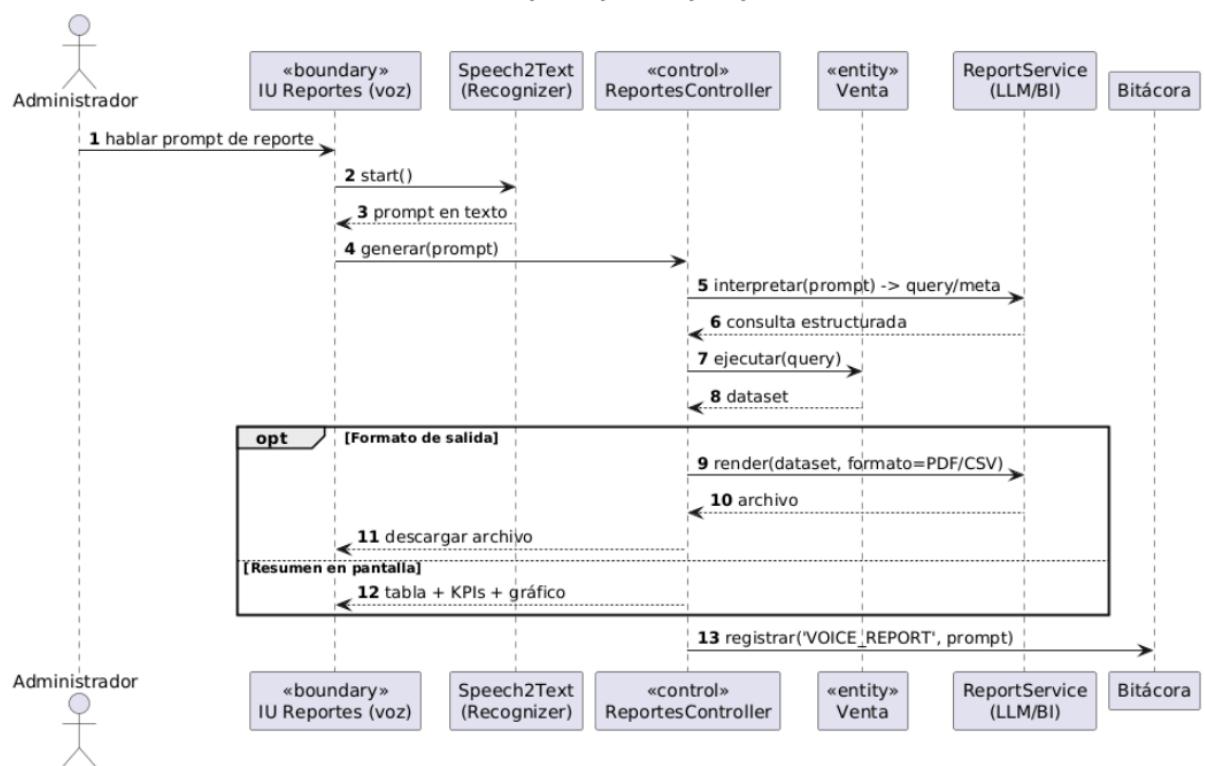
CU18 Notificaciones push



CU19 Búsqueda por voz (producto/cliente)



CU20 Reportes por voz (prompts)



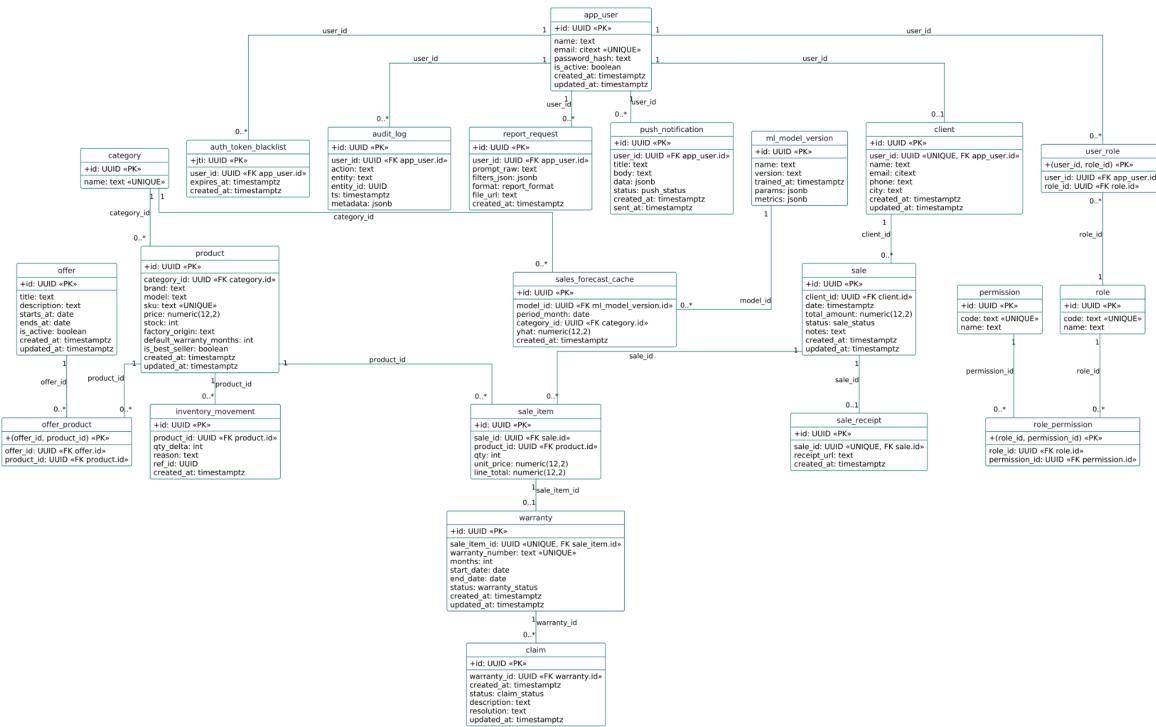
2.3.3 Diseño de Datos

2.3.3.1 Diseño de Datos Lógico

El siguiente diagrama representa el diseño lógico del sistema, basado en clases que luego se convertirán en tablas físicas de base de datos.

Cada clase contiene atributos que representan campos, y se indican las relaciones entre entidades clave del sistema.

- Diagrama de Clase



- Mapeo

PERMISO	
PK	
Id	Descripcion

ROL_PERMISO	
PK	
FK	FK
Id Permiso	Id Rol

ROL		
PK		
Id	Titulo	Descripcion

NOTIFICACION		
PK		
Id	Titulo	FechaHora

EMPLEADO				
PK				
	FK			
Codigo	<i>Ci Persona</i>	FechaIngreso	FechaSalida	Estado

REGISTRO						
PK					FK	FK
Id	FechaEntrada	FechaSalida	Estado	Motivo	<i>Id_Empleado</i>	<i>Id_visitante</i>

AUTORIZACION				
PK		FK	FK	FK
Id	Fecha	Estado	<i>Id_Empleado</i>	<i>Id_Visitante</i>

USUARIO					
PK		FK	FK	FK	
Id	Email	Contraseña	<i>Id_Rol</i>	<i>Id_Empleado</i>	<i>Id_Residente</i>

EMPRESA					
PK					
Id	Nombre	Email	Telefono	Estado	Servicio

MANTENIMIENTO							
PK							
FK					FK	FK	

PERSONA							
PK							
Ci	Nombre	Apellido	Telefono	TipoE	TipoR	TipoV	

- Normalización

Primera Forma Normal (FN1)

- Todas las tablas del modelo presentan valores atómicos.

- No existen columnas que almacenen listas de datos ni repeticiones dentro de una misma columna.
- Ejemplo: la tabla “Residente” almacena un solo nombre, un solo teléfono y un solo email por registro.

Conclusión: El modelo cumple con la Primera Forma Normal.

Segunda Forma Normal (2FN)

- Todas las tablas tienen claves primarias simples.
- No existen dependencias parciales: cada atributo depende completamente de la clave primaria.
- Ejemplo: en “Pago”, los campos monto, estado, etc., dependen íntegramente del campo id.

Conclusión: El modelo cumple con la Segunda Forma Normal.

Tercera Forma Normal (3FN)

- No existen dependencias transitivas entre atributos.
- Atributos que representan entidades independientes (como Rol, Cargo) están modelados como tablas separadas con claves foráneas bien definidas.
- Ejemplo: Usuario no contiene el nombre del rol, sino que se relaciona con Rol por su rol_id.

Conclusión: El modelo cumple con la Tercera Forma Normal.

2.3.3.2 *Diseño de Datos Físico*

- Tabla de Volumen

Usuario

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdUsuario	Entero	Identificador único del usuario.	4 bytes	No	PK
Email	Alfanumérico	Correo electrónico del usuario.	100 caracteres	No	
Nickname	Alfanumérico	Nombre del usuario.	100 caracteres	No	
Password	Alfanumérico	Contraseña del usuario	100 caracteres	No	
Estado	Alfanumérico	Estado del usuario.	20 Caracteres	No	
Telefono	Alfanumérico	Número telefónico vinculado con el usuario.	20 Caracteres	No	
IdRol	Entero	Relación al rol del usuario.	4 bytes	No	FK

Empleado

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdEmpleado	Entero	Identificador único del empleado	4 bytes	No	PK
Nombre	Alfanumérico	Nombres del empleado.	50 caracteres	No	
Apellido	Alfanumérico	Apellidos del empleado.	50 caracteres	No	
CI	Alfanumérico	Cédula de identidad del empleado	20 caracteres	Sí	
FechaIngr	Fecha	Fecha en la que el empleado ingresó	8 bytes	No	
FechaSal	Fecha	Fecha en la que el empleado salió	8 bytes	Sí	
Estado	Entero corto	Estado del empleado (1 activo, 0 inactivo)	1 byte	No	

IdCargoEmpleado	Entero	Relación al cargo del empleado	4 bytes	No	FK
IdUsuario	Entero	Relación al usuario del empleado	4 bytes	No	FK

CargoEmpleado

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdCargoEmpleado	Entero	Identificador único del cargo de empleado.	4 bytes	No	PK
Cargo	Alfanumérico	Descripción o nombre del cargo del empleado.	100 caracteres	No	
Estado	Entero corto	Estado del empleado (1 activo, 0 inactivo)	1 byte	No	

Gasto

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
id	Entero	Identificador único del gasto	4 bytes	No	PK
Descripcion	Alfanumérico	Detalle del gasto	100 caracteres	No	
Monto	Decimal	Monto económico del gasto	8 bytes	No	
FechaHora	FechaHora	Fecha y hora en que se registró el gasto	8 bytes	No	
Id_Usuario	Entero	Usuario que registró el gasto	4 bytes	No	FK
Id_Mantenimiento	Entero	Número correspondiente del mantenimiento.	4 bytes	No	FK

Mantenimiento

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave

IdMantenimiento	Entero	Identificador único del mantenimiento	4 bytes	No	PK
FechaHora	Fecha y Hora	Fecha y hora del mantenimiento	8 bytes	No	
Descripcion	Alfanumérico	Descripción del mantenimiento solicitado	100 caracteres	No	
Monto	Decimal	Costo estimado del mantenimiento	8 bytes	No	
Estado	Alfanumérico	Estado (pendiente, en proceso, finalizado)	20 caracteres	No	
IdUsuario	Entero	Usuario que solicitó el mantenimiento	4 bytes	No	FK
IdEmpresa	Entero	Empresa contratada para el mantenimiento	4 bytes	No	FK
IdArea	Entero	Área que recibe el mantenimiento	4 bytes	No	

Cuotas

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdCuota	Entero	Identificador único de la cuota	4 bytes	No	PK
FechaEmisión	Fecha	Fecha de emisión de la cuota	8 bytes	No	
Estado	Alfanumérico	Estado actual de la cuota	20 caracteres	No	
Monto	Decimal	Costo de la cuota a pagar.	8 bytes	No	
Descripcion	Alfanumérico	Descripción acerca de la cuota	255 caracteres	No	
FechaVencimiento	Fecha	Fecha de vencimiento de la cuota	8 bytes	No	
IdResidente	Entero	Relación de la cuota con el residente.	4 bytes	No	FK
IdReserva	Entero	Identificador de la cuota de reserva	4 bytes	Sí	FK

IdMantenimiento	Entero	Identificador de la cuota de mantenimiento	4 bytes	Sí	FK
-----------------	--------	--	---------	----	----

Pago

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdPago	Entero	Identificador único del pago	4 bytes	No	PK
FechaPago	Fecha	Fecha de emisión del pago	8 bytes	No	
Referencia	Alfanumérico	Referencia e información del pago.	50 caracteres	Sí	
MetodoPago	Alfanumérico	Información del método de pago.	50 caracteres	Sí	
Monto	Decimal	Costo de la cuota a pagar.	8 bytes	No	
IdCuota	Entero	Relación del pago en cuota.	4 bytes	No	FK

Empresa

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IDEmpresa	Entero	Identificador único de la empresa	4 bytes	No	PK
Nit	Alfanumérico	Número de Identificación Tributaria	30 caracteres	No	
Nombre	Alfanumérico	Nombre de la empresa	50 caracteres	No	
Telefono	Alfanumérico	Teléfono de contacto	20 caracteres	No	
Email	Alfanumérico	Correo electrónico de contacto	100 caracteres	No	
Estado	Entero corto	Estado de disponibilidad de la empresa.	1 Byte	No	

Bitácora

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdBitacora	Entero	Identificador único del registro	4 bytes	No	PK
Descripcion	Alfanumérico	Acción registrada (login, update, etc.)	50 caracteres	No	
FechaHora	FechaHora	Fecha y hora en la que ocurrió la acción	8 bytes	No	
Ip	Alfanumérico	Dirección IP del usuario	45 caracteres	No	
Id_Usuario	Entero	Usuario que realizó la acción	4 bytes	No	FK

Rol

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdRol	Entero	Identificador único del permiso	4 bytes	No	PK
Nombre	Alfanumérico	Nombre del permiso	50 caracteres	No	
Estado	Entero Corto	Estado del permiso otorgado	1 Byte	No	

Permiso

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdPermiso	Entero	Identificador único del permiso	4 bytes	No	PK
Descripcion	Alfanumérico	Descripción del permiso otorgado	100 caracteres	No	
Estado	Entero Corto	Reporte del estado del permiso.	1 Byte	No	

rol_permiso

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdRolPermis o	Entero	Identificador único de la relación.	4 Bytes	No	PK
IdRol	Entero	Identificador del rol	4 bytes	No	FK
id_Permito	Entero	Identificador del permiso	4 bytes	No	FK
Estado	Entero corto	Estado actual del permiso al rol.	1 byte	No	

Notificación

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdNotificaci on	Entero	Identificador único de la notificación	4 bytes	No	PK
Titulo	Alfanuméri co	Título o encabezado de la notificación	100 caracteres	No	
Contenido	Alfanuméri co	Cuerpo o contenido del mensaje	255 caracteres	No	
FechaHora	FechaHora	Fecha y hora de creación o envío	8 bytes	No	
Tipo	Alfanuméri co	Tipo de notificación (urgente, aviso, etc.)	50 caracteres	No	
Id_Resident e	Entero	Residente que recibe la notificación	4 bytes	No	FK

Gasto

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdGasto	Entero	Identificador único del gasto	4 bytes	No	PK
Descripcion	Alfanuméri co	Descripción del gasto realizado	255 caracteres	No	

Monto	Decimal	Costo del gasto..	8 bytes	No	
FechaHora	FechaHora	Fecha y hora del gasto realizado.	8 bytes	Sí	
IdUsuario	Entero	Relación al usuario que realizó el gasto.	4 Byte	Sí	FK
IdMantenimiento	Entero	Relación del gasto en mantenimiento.	4 bytes	No	FK

Reclamo

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdReclamo	Entero	Identificador único del reclamo	4 bytes	No	PK
Titulo	Alfanumérico	Titulo del reclamo.	100 caracteres	No	
Contenido	Alfanumérico	Contenido del reclamo realizado	255 caracteres	No	
Imagen	Alfanumérico	Identificador de la imagen	255 caracteres	No	
FechaHora	FechaHora	Fecha y hora del reclamo realizado.	8 bytes	Sí	
Estado	Alfanumérico	Estado actual del reclamo.	20 caracteres	No	
IdResidente	Entero	Residente que realizó el reclamo.	4 bytes	No	FK

Multa

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdMultas	Entero	Identificador único de la multa	4 bytes	No	PK

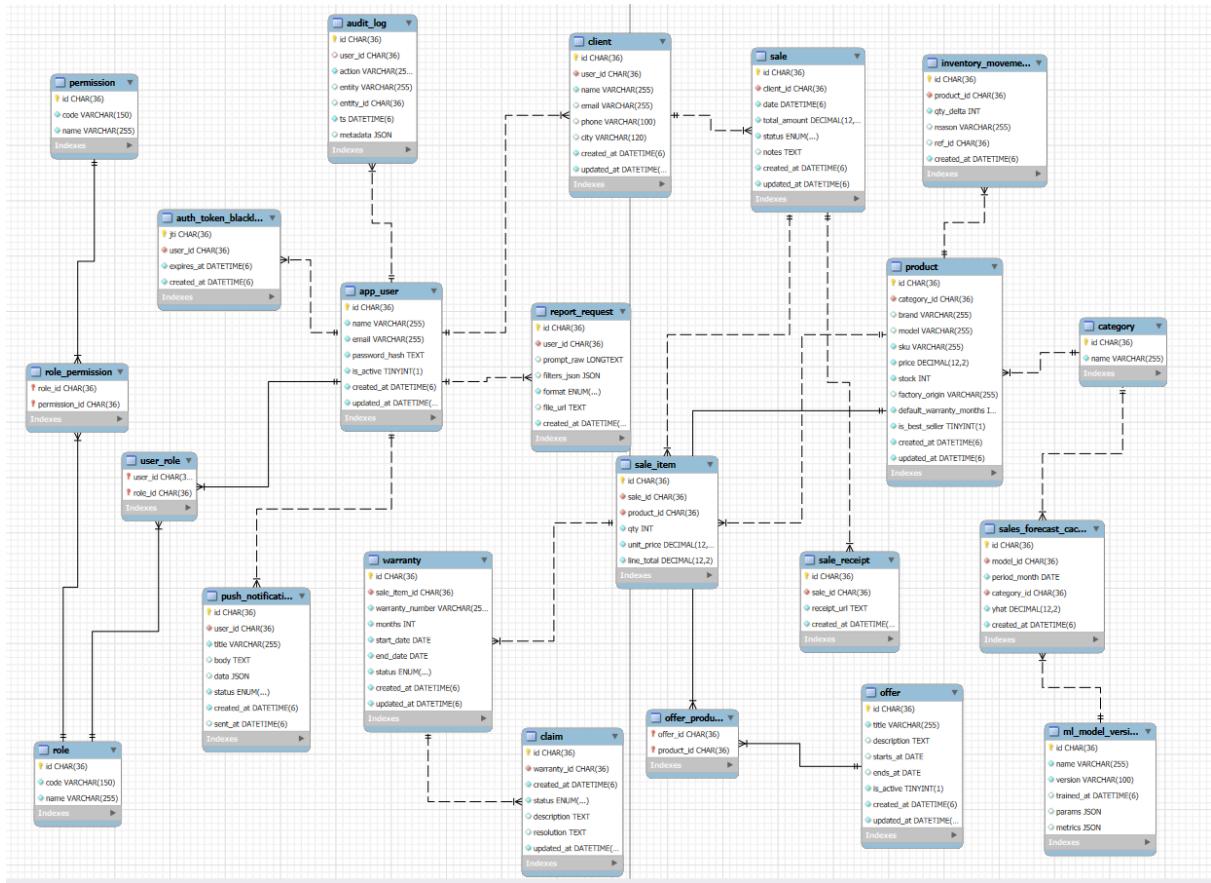
		multa			
Motivo	Alfanumérico	Motivo para la emisión de la multa.	255 caracteres	No	
Monto	Decimal	Costo de la multa	8 bytes	No	
FechaEmisión	Fecha	Fecha de la emisión de la multa.	8 bytes	Sí	
Estado	Alfanumérico	Estado actual de la multa.	20 caracteres	No	
IdResidente	Entero	Identificador del residente multado	4 bytes	No	FK
IdCuota	Entero	Identificador de la cuota relacionada con la multa	4 bytes	Sí	FK

AutorizacionVisita

Atributo	Tipo de Dato	Descripción	Tamaño	Nulo	Llave
IdAutorización	Entero	Identificador único de la autorización	4 bytes	No	PK
FechaAutorización	FechaHora	Fecha y hora en la que se autorizó la visita	8 bytes	No	
Observación	Alfanumérico	Observación particular de la autorización.	255 caracteres	Sí	
FechaEntrada	FechaHora	Fecha y hora de ingreso del visitante	8 bytes	Sí	
FechaSalida	FechaHora	Fecha y hora esperada de salida	8 bytes	Sí	
Motivo	Alfanumérico	Motivo de la visita	100 caracteres	No	
PlacaVehículo	Alfanumérico	Placa del vehículo del visitante (si aplica)	20 caracteres	Sí	
IdEmpleado	Entero	Identificador del empleado	4 bytes	No	FK
CiVisitante	Entero	Identificador del visitante	4 bytes	No	FK

IdResidente	Entero	Residente que autoriza la visita	4 bytes	No	FK
IdPropiedad	Alfanumérico	Número de casa a la que se dirige el visitante	10 caracteres	No	FK

- Diagrama Relacional



- Script

```
-- Requisitos recomendados
```

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

```
CREATE EXTENSION IF NOT EXISTS pgcrypto; -- gen_random_uuid()
```

```
CREATE EXTENSION IF NOT EXISTS citext;
```

```
-- ===== Tipos Enumerados =====
```

```
DO $$
```

```

BEGIN
IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname='sale_status') THEN
    CREATE TYPE sale_status AS ENUM ('PENDING','PAID','CANCELLED');
END IF;
IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname='warranty_status') THEN
    CREATE TYPE warranty_status AS ENUM
('ACTIVE','EXPIRED','CLAIMED','CANCELLED');
END IF;
IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname='claim_status') THEN
    CREATE TYPE claim_status AS ENUM
('OPEN','IN REVIEW','APPROVED','REJECTED','RESOLVED');
END IF;
IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname='report_format') THEN
    CREATE TYPE report_format AS ENUM ('PDF','EXCEL','SCREEN');
END IF;
IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname='push_status') THEN
    CREATE TYPE push_status AS ENUM ('QUEUED','SENT','FAILED');
END IF;
END$$;
```

-- ===== IAM / Usuarios =====

```

CREATE TABLE IF NOT EXISTS app_user(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name text NOT NULL,
    email citext NOT NULL UNIQUE,
    password_hash text NOT NULL,
    is_active boolean NOT NULL DEFAULT true,
    created_at timestamp NOT NULL DEFAULT now(),
    updated_at timestamp NOT NULL DEFAULT now()
);
```

```

CREATE TABLE IF NOT EXISTS role(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    code text NOT NULL UNIQUE,
    name text NOT NULL
);
```

```

CREATE TABLE IF NOT EXISTS permission(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    code text NOT NULL UNIQUE,
    name text NOT NULL
);

CREATE TABLE IF NOT EXISTS role_permission(
    role_id UUID NOT NULL REFERENCES role(id) ON DELETE CASCADE,
    permission_id UUID NOT NULL REFERENCES permission(id) ON DELETE CASCADE,
    PRIMARY KEY(role_id, permission_id)
);

CREATE TABLE IF NOT EXISTS user_role(
    user_id UUID NOT NULL REFERENCES app_user(id) ON DELETE CASCADE,
    role_id UUID NOT NULL REFERENCES role(id) ON DELETE CASCADE,
    PRIMARY KEY(user_id, role_id)
);

CREATE TABLE IF NOT EXISTS auth_token_blacklist(
    jti UUID PRIMARY KEY,
    user_id UUID NOT NULL REFERENCES app_user(id) ON DELETE CASCADE,
    expires_at timestamp NOT NULL,
    created_at timestamp NOT NULL DEFAULT now()
);
CREATE INDEX IF NOT EXISTS idx_blacklist_user ON auth_token_blacklist(user_id);

-- ===== Clientes =====
CREATE TABLE IF NOT EXISTS client(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID UNIQUE REFERENCES app_user(id) ON DELETE SET NULL,
    name text NOT NULL,
    email citext,
    phone text,
    city text,
    created_at timestamp NOT NULL DEFAULT now(),
    updated_at timestamp NOT NULL DEFAULT now()
);

```

```
CREATE INDEX IF NOT EXISTS idx_client_name ON client (name);
CREATE INDEX IF NOT EXISTS idx_client_email ON client (email);
```

```
-- ===== Catálogo / Inventory =====
```

```
CREATE TABLE IF NOT EXISTS category(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name text NOT NULL UNIQUE
);
```

```
CREATE TABLE IF NOT EXISTS product(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    category_id UUID NOT NULL REFERENCES category(id) ON DELETE RESTRICT,
    brand text,
    model text,
    sku text NOT NULL UNIQUE,
    price numeric(12,2) NOT NULL CHECK (price > 0),
    stock int NOT NULL CHECK (stock >= 0),
    factory_origin text,
    default_warranty_months int NOT NULL DEFAULT 0 CHECK (default_warranty_months >= 0),
    is_best_seller boolean NOT NULL DEFAULT false,
    created_at timestamp NOT NULL DEFAULT now(),
    updated_at timestamp NOT NULL DEFAULT now()
);
```

```
CREATE INDEX IF NOT EXISTS idx_product_category ON product(category_id);
CREATE INDEX IF NOT EXISTS idx_product_brand_model ON product(brand, model);
```

```
CREATE TABLE IF NOT EXISTS inventory_movement(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    product_id UUID NOT NULL REFERENCES product(id) ON DELETE CASCADE,
    qty_delta int NOT NULL,
    reason text NOT NULL,
    ref_id UUID,
    created_at timestamp NOT NULL DEFAULT now()
);
```

```
CREATE INDEX IF NOT EXISTS idx_inv_mov_product ON
inventory_movement(product_id);
```

```
-- ===== Ofertas =====
```

```
CREATE TABLE IF NOT EXISTS offer(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    title text NOT NULL,
    description text,
    starts_at date,
    ends_at date,
    is_active boolean NOT NULL DEFAULT true,
    created_at timestamptz NOT NULL DEFAULT now(),
    updated_at timestamptz NOT NULL DEFAULT now()
);
```

```
CREATE TABLE IF NOT EXISTS offer_product(
```

```
offer_id UUID NOT NULL REFERENCES offer(id) ON DELETE CASCADE,
product_id UUID NOT NULL REFERENCES product(id) ON DELETE CASCADE,
PRIMARY KEY(offer_id, product_id)
```

```
);
```

```
-- ===== Ventas / Comprobante =====
```

```
CREATE TABLE IF NOT EXISTS sale(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    client_id UUID NOT NULL REFERENCES client(id) ON DELETE RESTRICT,
    date timestamptz NOT NULL DEFAULT now(),
    total_amount numeric(12,2) NOT NULL DEFAULT 0,
    status sale_status NOT NULL DEFAULT 'PENDING',
    notes text,
    created_at timestamptz NOT NULL DEFAULT now(),
    updated_at timestamptz NOT NULL DEFAULT now()
);
```

```
CREATE INDEX IF NOT EXISTS idx_sale_client_date ON sale(client_id, date);
```

```
CREATE TABLE IF NOT EXISTS sale_item(
```

```
id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
sale_id UUID NOT NULL REFERENCES sale(id) ON DELETE CASCADE,
product_id UUID NOT NULL REFERENCES product(id) ON DELETE RESTRICT,
qty int NOT NULL CHECK (qty > 0),
```

```

unit_price numeric(12,2),
line_total numeric(12,2)
);
CREATE INDEX IF NOT EXISTS idx_sale_item_sale ON sale_item(sale_id);
CREATE INDEX IF NOT EXISTS idx_sale_item_product ON sale_item(product_id);

CREATE TABLE IF NOT EXISTS sale_receipt(
id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
sale_id UUID NOT NULL UNIQUE REFERENCES sale(id) ON DELETE CASCADE,
receipt_url text NOT NULL,
created_at timestamptz NOT NULL DEFAULT now()
);

-- ====== Garantías y Reclamos ======
CREATE TABLE IF NOT EXISTS warranty(
id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
sale_item_id UUID NOT NULL UNIQUE REFERENCES sale_item(id) ON DELETE
CASCADE,
warranty_number text NOT NULL UNIQUE,
months int NOT NULL CHECK (months >= 0),
start_date date NOT NULL,
end_date date NOT NULL,
status warranty_status NOT NULL DEFAULT 'ACTIVE',
created_at timestamptz NOT NULL DEFAULT now(),
updated_at timestamptz NOT NULL DEFAULT now()
);

CREATE TABLE IF NOT EXISTS claim(
id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
warranty_id UUID NOT NULL REFERENCES warranty(id) ON DELETE CASCADE,
created_at timestamptz NOT NULL DEFAULT now(),
status claim_status NOT NULL DEFAULT 'OPEN',
description text,
resolution text,
updated_at timestamptz NOT NULL DEFAULT now()
);
CREATE INDEX IF NOT EXISTS idx_claim_warranty ON claim(warranty_id);

```

```

-- ===== Auditoría / Reportes / Push =====
CREATE TABLE IF NOT EXISTS audit_log(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES app_user(id) ON DELETE SET NULL,
    action text NOT NULL,
    entity text,
    entity_id UUID,
    ts timestamptz NOT NULL DEFAULT now(),
    metadata jsonb NOT NULL DEFAULT '{})::jsonb
);
CREATE INDEX IF NOT EXISTS idx_audit_ts ON audit_log(ts DESC);
CREATE INDEX IF NOT EXISTS idx_audit_action ON audit_log(action);

CREATE TABLE IF NOT EXISTS report_request(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES app_user(id) ON DELETE SET NULL,
    prompt_raw text NOT NULL,
    filters_json jsonb NOT NULL DEFAULT '{})::jsonb,
    format report_format NOT NULL DEFAULT 'SCREEN',
    file_url text,
    created_at timestamptz NOT NULL DEFAULT now()
);
CREATE TABLE IF NOT EXISTS push_notification(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES app_user(id) ON DELETE SET NULL,
    title text NOT NULL,
    body text NOT NULL,
    data jsonb NOT NULL DEFAULT '{})::jsonb,
    status push_status NOT NULL DEFAULT 'QUEUED',
    created_at timestamptz NOT NULL DEFAULT now(),
    sent_at timestamptz
);
CREATE INDEX IF NOT EXISTS idx_push_status ON push_notification(status);

-- ===== ML (versionado + cache) =====

```

```
CREATE TABLE IF NOT EXISTS ml_model_version(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name text NOT NULL, -- RandomForestRegressor
    version text NOT NULL, -- v1.0.0
    trained_at timestampz NOT NULL DEFAULT now(),
    params jsonb NOT NULL DEFAULT '{}':jsonb,
    metrics jsonb NOT NULL DEFAULT '{}':jsonb
);
```

```
CREATE TABLE IF NOT EXISTS sales_forecast_cache(
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    model_id UUID REFERENCES ml_model_version(id) ON DELETE SET NULL,
    period_month date NOT NULL,
    category_id UUID REFERENCES category(id) ON DELETE SET NULL,
    yhat numeric(12,2) NOT NULL,
    created_at timestampz NOT NULL DEFAULT now(),
    CONSTRAINT uq_forecast UNIQUE(model_id, period_month, category_id)
);
```

```
-- ====== Vistas para Dashboard ======
CREATE OR REPLACE VIEW vw_sales_monthly_by_category AS
SELECT date_trunc('month', s.date)::date AS month,
       c.id AS category_id,
       c.name AS category,
       SUM(si.line_total) AS total_amount,
       COUNT(DISTINCT s.id) AS sales_count
  FROM sale s
 JOIN sale_item si ON si.sale_id = s.id
 JOIN product p ON p.id = si.product_id
 JOIN category c ON c.id = p.category_id
 GROUP BY 1,2,3;
```

```
CREATE OR REPLACE VIEW vw_top_products_90d AS
SELECT p.id, p.brand, p.model, p.sku,
       SUM(si.qty) AS units,
       SUM(si.line_total) AS total_amount
  FROM sale s
```

```

JOIN sale_item si ON si.sale_id = s.id
JOIN product p ON p.id = si.product_id
WHERE s.date >= now() - interval '90 days'
GROUP BY p.id, p.brand, p.model, p.sku
ORDER BY total_amount DESC;

```

```
-- ===== Triggers de negocio =====
```

```

-- 1) Calcula line_total y autocompleta unit_price si no viene
CREATE OR REPLACE FUNCTION trg_sale_item_compute_line_total()
RETURNS TRIGGER AS $$

BEGIN
IF NEW.unit_price IS NULL THEN
    SELECT price INTO NEW.unit_price FROM product WHERE id = NEW.product_id;
END IF;
NEW.line_total := NEW.qty * NEW.unit_price;
RETURN NEW;
END$$ LANGUAGE plpgsql;

```

```

DROP TRIGGER IF EXISTS btg_sale_item_compute ON sale_item;
CREATE TRIGGER btg_sale_item_compute
BEFORE INSERT OR UPDATE ON sale_item
FOR EACH ROW EXECUTE FUNCTION trg_sale_item_compute_line_total();

```

```

-- 2) Mantiene stock (inventory_movement) y total_amount de la venta
CREATE OR REPLACE FUNCTION trg_sale_item_stock_and_totals()
RETURNS TRIGGER AS $$

DECLARE delta int;
DECLARE sid UUID;
BEGIN
IF TG_OP = 'INSERT' THEN
    delta := -NEW.qty;
    IF (SELECT stock FROM product WHERE id=NEW.product_id) + delta < 0
    THEN RAISE EXCEPTION 'Stock insuficiente'; END IF;
    UPDATE product SET stock = stock + delta, updated_at = now() WHERE id =
    NEW.product_id;
    INSERT INTO inventory_movement(product_id, qty_delta, reason, ref_id)

```

```

VALUES (NEW.product_id, delta, 'SALE', NEW.id);
sid := NEW.sale_id;

ELSIF TG_OP = 'UPDATE' THEN
    delta := -(NEW.qty - OLD.qty);
    IF (SELECT stock FROM product WHERE id=NEW.product_id) + delta < 0
    THEN RAISE EXCEPTION 'Stock insuficiente (update)'; END IF;
    UPDATE product SET stock = stock + delta, updated_at = now() WHERE id =
    NEW.product_id;
    INSERT INTO inventory_movement(product_id, qty_delta, reason, ref_id)
    VALUES (NEW.product_id, delta, 'SALE_ADJUST', NEW.id);
    sid := NEW.sale_id;

ELSIF TG_OP = 'DELETE' THEN
    delta := +OLD.qty;
    UPDATE product SET stock = stock + delta, updated_at = now() WHERE id =
    OLD.product_id;
    INSERT INTO inventory_movement(product_id, qty_delta, reason, ref_id)
    VALUES (OLD.product_id, delta, 'SALE_DELETE', OLD.id);
    sid := OLD.sale_id;
END IF;

UPDATE sale
SET total_amount = COALESCE(
    SELECT SUM(line_total) FROM sale_item WHERE sale_id = sid
    ), 0),
    updated_at = now()
WHERE id = sid;

RETURN COALESCE(NEW, OLD);

END$$ LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS atg_sale_item_stock ON sale_item;
CREATE TRIGGER atg_sale_item_stock
AFTER INSERT OR UPDATE OR DELETE ON sale_item
FOR EACH ROW EXECUTE FUNCTION trg_sale_item_stock_and_totals();

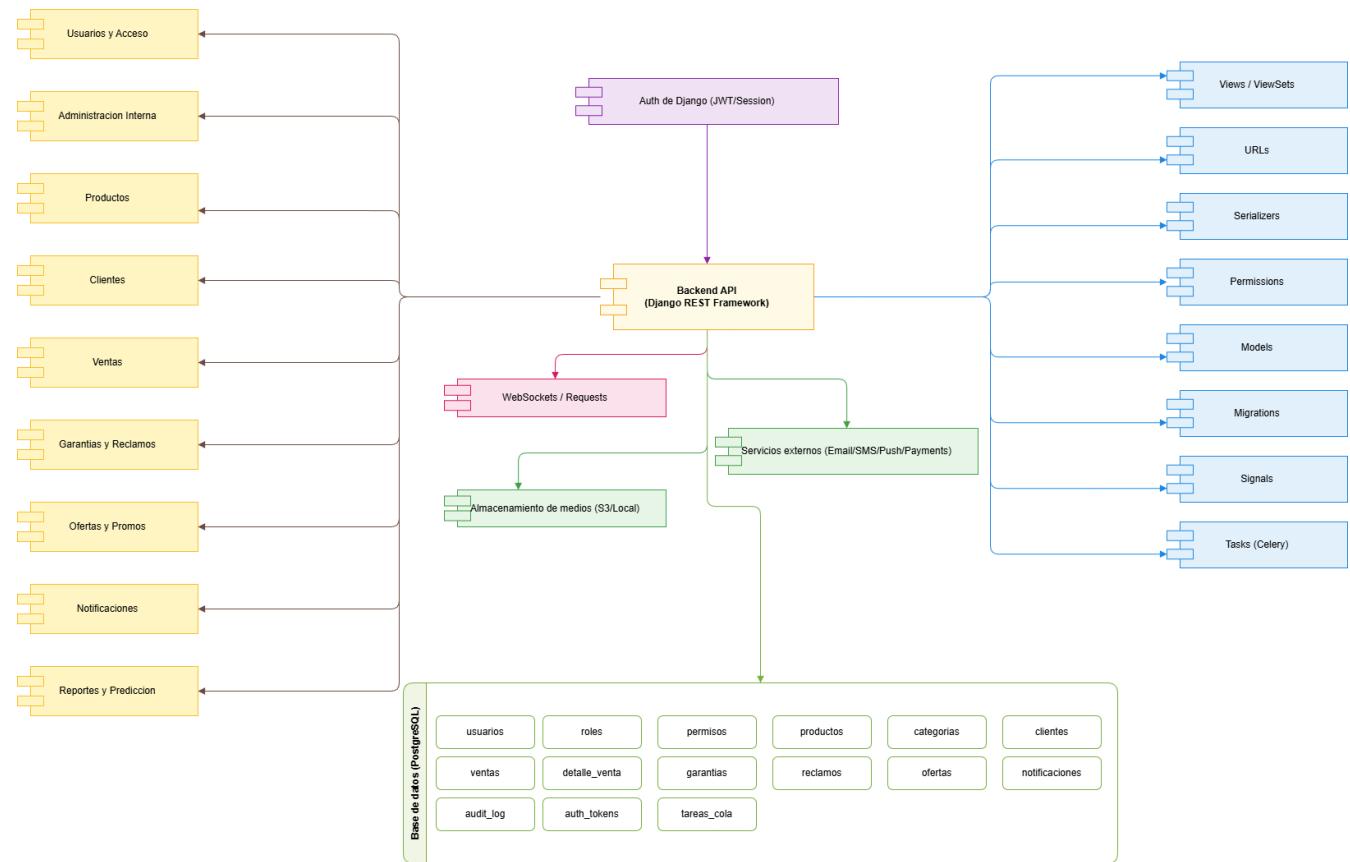
```

2.4 Flujo de Trabajo: Implementación

2.4.1 Elección de plataforma de desarrollo de Software

2.4.1.1 Implementación de la arquitectura del Sistema

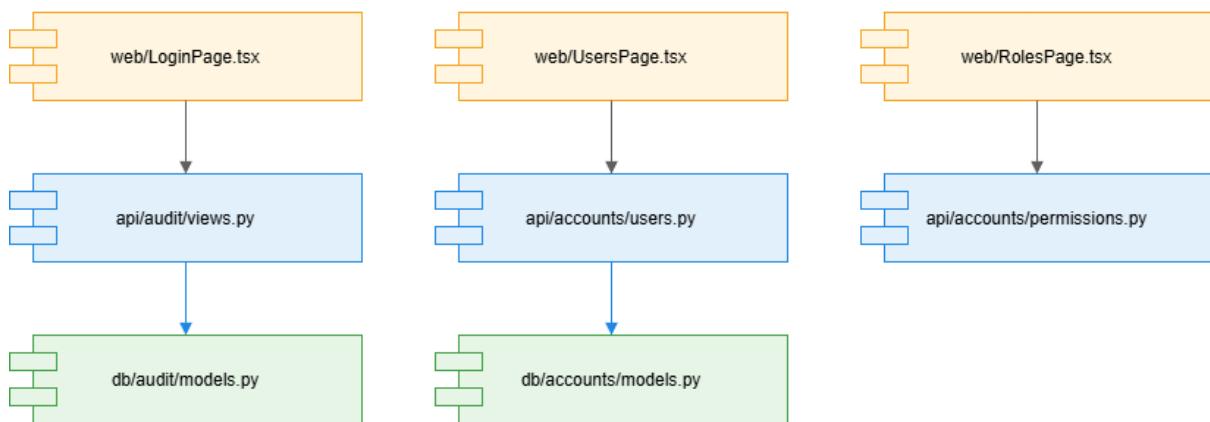
2.4.1.1.1 Diagrama de componente principal del sistema



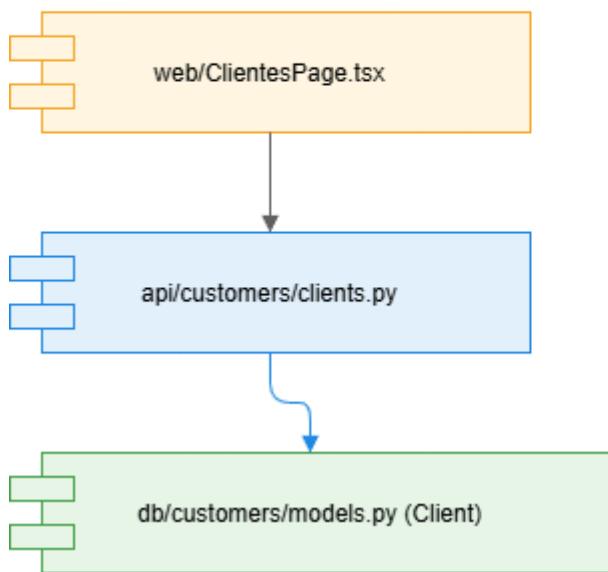
2.4.1.2 Implementación de la arquitectura del subsistema

2.4.1.2.1 Diagrama de componente de cada paquete

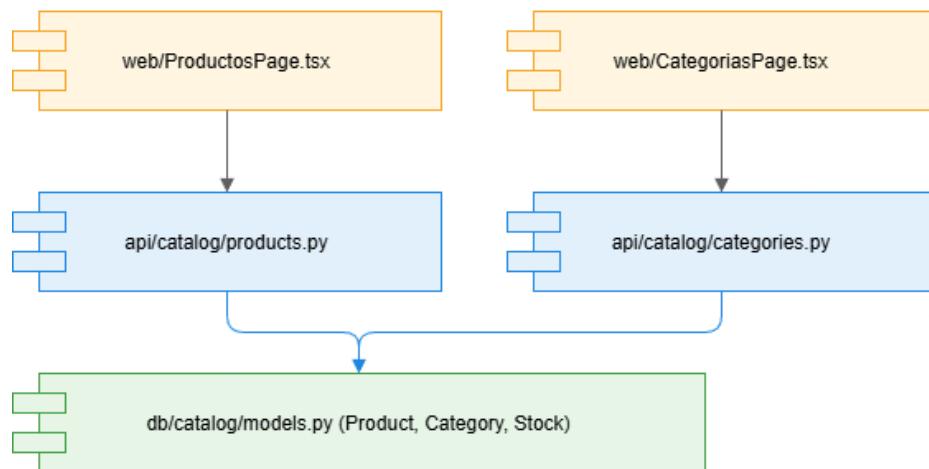
Paquete: Gestión de Usuarios



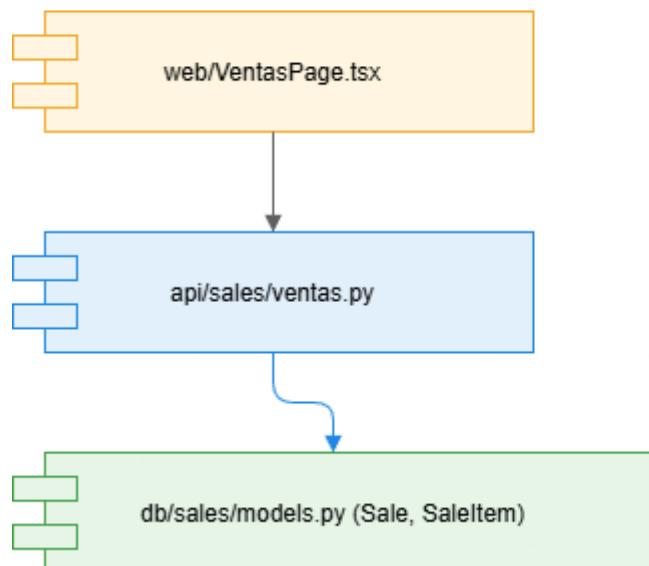
Paquete: Gestión de Clientes



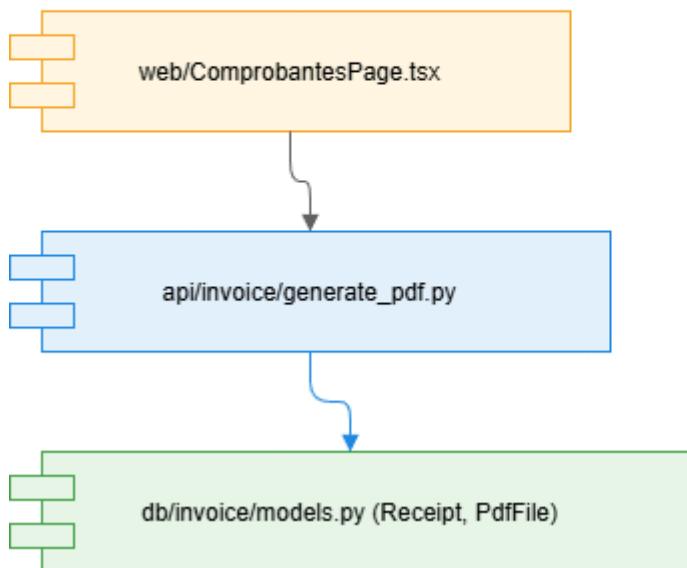
Paquete: Gestión de Inventario y Catálogo



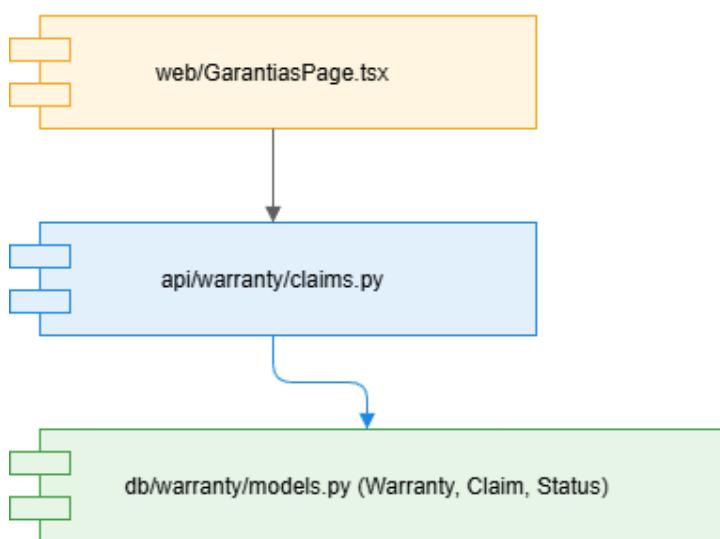
Paquete: Gestión de Ventas



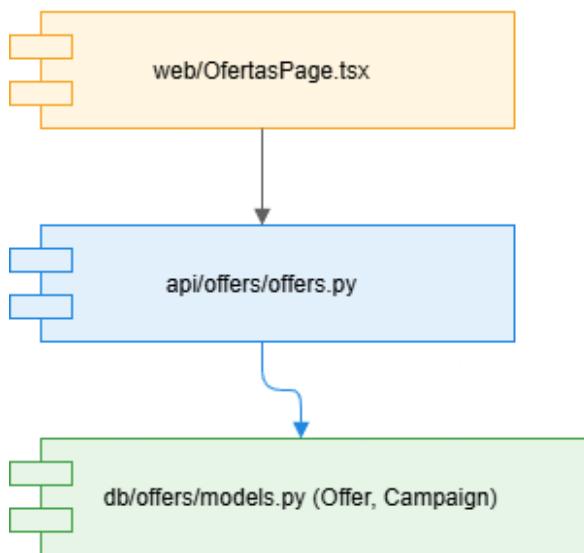
Paquete: Comprobantes



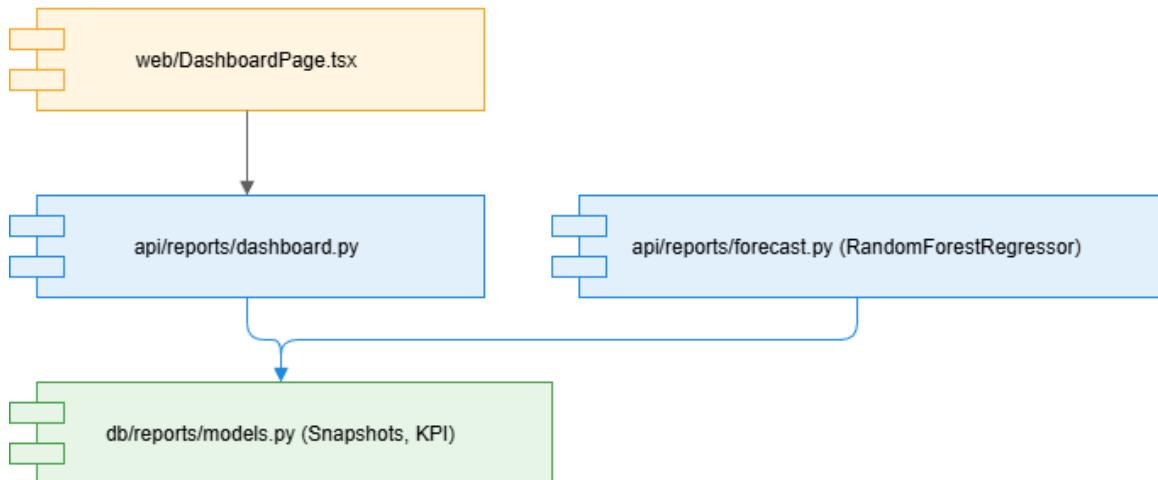
Paquete: Garantías y Reclamos



Paquete: Ofertas y Promociones



Paquete: Reportes y Dashboard (ML)



Tecnologías: Django REST + PostgreSQL + React (PWA) + Docker + JWT.

Pasos básicos

1. Clonar y preparar

- git clone ...
- Crear archivo .env con DB, JWT y DEBUG.

2. Levantar servicios (dev)

- docker compose up -d --build
- Backend en :8000, Web en :5173, Nginx en :8080.

3. Migraciones y usuario admin

- docker compose exec backend python manage.py migrate
- docker compose exec backend python manage.py createsuperuser

4. Datos mínimos (opcional)

- Categorías, productos y clientes de prueba (seed).

5. Rutas clave (CU01–CU10)

- `/api/iam/login|logout` (auth)
- `/api/iam/roles|permissions|user-roles` (IAM)
- `/api/catalog/categories|products` (productos)
- `/api/clients/clients` (clientes)
- `/api/audit/logs` (bitácora)

6. Frontend (web)

- `cd frontend-web && npm i && npm run dev`
- Páginas: Login, Productos, Clientes, Bitácora.

7. Seguridad mínima

- JWT (access/refresh), RBAC (roles/permisos), CORS en dev.

8. Listo para demo

- Entrar a `http://localhost:8080`
- Login con admin y recorrer: Productos, Clientes, Bitácora.

2.5 Flujo de Trabajo: Pruebas

Objetivo:

Verificar que las funcionalidades implementadas cumplen los requisitos definidos y que el sistema es estable antes de pasar a producción.

2.5.1 Actividades principales

1. Planificación de Pruebas

- **Responsable:** Líder de Pruebas / Analista / Docente (según tu caso).
- **Entradas:** Casos de uso, requisitos funcionales y no funcionales, arquitectura lógica.
- **Salidas:**
 - Plan de pruebas (qué se va a probar, alcance, tipos de prueba).
 - Lista preliminar de módulos a probar y prioridades.

2. Diseño de Casos de Prueba

- **Responsable:** Tester / Desarrollador.
- **Entradas:** Plan de pruebas, casos de uso, modelos (diagramas de secuencia, diagramas de clases).
- **Salidas:**
 - Casos de prueba documentados (ID, objetivo, pasos, datos de entrada, resultado esperado).
 - Matriz de trazabilidad simple (requisito ↔ casos de prueba).

3. Preparación del Entorno y Datos de Prueba

- **Responsable:** Desarrollador / DevOps (o quien sea en tu proyecto).
- **Entradas:** Configuración del sistema, script de base de datos, instaladores, contenedores, etc.
- **Salidas:**

- Entorno de pruebas configurado (BD con datos de prueba, servidor levantado, usuarios de prueba).

4. Ejecución de Pruebas

- **Responsable:** Tester / Desarrollador.
- **Entradas:** Casos de prueba, entorno preparado.
- **Salidas:**
 - Resultados de pruebas (OK / FALLA).
 - Registro de incidencias / defectos encontrados (ID, descripción, severidad, módulo).

5. Corrección de Defectos y Repruebas

- **Responsable:** Desarrollador (corregir) + Tester (volver a probar).
- **Entradas:** Registro de incidencias, código fuente.
- **Salidas:**
 - Versión corregida del módulo.
 - Incidencias cerradas o reabiertas según resultado.

6. Cierre y Reporte de Pruebas

- **Responsable:** Líder de pruebas / Equipo.
- **Entradas:** Resultados de pruebas, estado de incidencias.
- **Salidas:**
 - Informe de pruebas: casos ejecutados, porcentaje aprobados, fallos críticos, observaciones.
 - Recomendación: listo para entrega / se requiere nueva iteración.

2.5.2 Tabla resumen del flujo

#	Actividad	Rol principal	Entradas	Salidas
1	Planificación de pruebas	Líder de pruebas	Requisitos, casos de uso	Plan de pruebas
2	Diseño de casos de prueba	Tester / Desarrollor	Plan de pruebas, requisitos	Casos de prueba documentados
3	Preparar entorno y datos	Dev / Soporte	Scripts BD, configuración sistema	Entorno de pruebas listo
4	Ejecutar pruebas	Tester / Dev	Casos de prueba, entorno	Resultados de pruebas, incidencias
5	Corregir y reprobadas	Dev + Tester	Incidencias, código fuente	Errores corregidos, incidencias actualiz.
6	Cierre y reporte de pruebas	Líder de pruebas	Resultados, incidencias	Informe final de pruebas

2.5.3 Evidencias de pruebas

Captura 1: Ejemplo de caso de prueba en Excel / herramienta.

Captura 2: Resultado de ejecución de pruebas (navegador, Postman, etc.).

Captura 3: Registro de incidencias.

Objetivo: Verificar que CU01–CU10 funcionan (API + UI), con permisos y bitácora.

Qué probar

- **Auth:** login correcto/incorrecto, logout invalida token.
- **IAM:** crear rol, asignar permisos/roles, 403 si no tiene permiso.
- **Productos:** crear/editar/listar, SKU único, price>0, filtros.
- **Clientes:** crear/editar/listar, email válido, filtros.
- **Bitácora:** listar por acción/fecha, solo con permiso.

Automatización mínima

- **Backend (pytest)**
 - `pip install pytest pytest-django`
 - `pytest -q`
 - Casos: login 200/401, producto 201/409, cliente 201/400, audit 200, 403 sin permisos.
- **Web (Cypress)**
 - `npm i -D cypress`
 - Smoke: login → ver tabla de productos → ver tabla de clientes → ver bitácora.

Criterios de aceptación

- Endpoints CU01–CU10 responden OK (caminos felices + 1 error).
- Accesos indebidos devuelven **401/403**.
- Acciones clave registran **audit_log** (LOGIN/LOGOUT/CREATE/UPDATE).
- UI muestra listados con filtros y paginación básica.

Entrega mínima (evidencia)

- Captura de **Swagger/Docs** y **login exitoso**.
- JSON de una **creación de producto** (201) y **SKU duplicado** (409).
- Captura de **Bitácora** filtrada por LOGIN.

Manual de Usuario

1. Acceso a la plataforma (Web – Administrador)

1. Abre el navegador y visita: `http://localhost:8080` (o la URL que te asigne la nube).
2. Clic en **Iniciar sesión**.
3. Ingresa **correo** y **contraseña**.
4. Presiona **Entrar**. Si las credenciales son válidas, verás el **Dashboard**.

Si olvidaste tu contraseña, contacta al administrador del sistema.

Si tu cuenta está inactiva, el sistema mostrará un error de acceso.

2. Menú principal (Web)

- **Catálogo** → Productos (crear, editar, listar).
- **Clientes** → Clientes (crear, editar, listar).
- **Bitácora** → Registro de eventos del sistema.
- **Perfil / Salir** → Cerrar sesión.

3. Casos de uso (Web – Administrador)

CU01 – Iniciar sesión

- Ubicación: `/login`.
- Ingresa correo + contraseña → **Entrar**.

CU02 – Cerrar sesión

- En la esquina superior derecha, abre el menú de usuario → **Cerrar sesión**.
- Se invalida tu token de sesión y vuelves a /login.

CU03 – Gestionar roles y permisos (si tu rol lo permite)

- Menú **Usuarios / IAM**.
- **Crear rol:** “Nuevo rol” → completa *Código* y *Nombre* → **Guardar**.
- **Asignar permisos a rol:** abre el rol → pestaña **Permisos** → marca/desmarca → **Guardar**.
- **Asignar rol a usuario:** en **Usuarios** → **Editar** → **Roles** → **Agregar** → **Guardar**.

CU04 – Registrar producto

1. Menú **Catálogo** → **Productos** → **Nuevo**.
2. Completa: Categoría, Marca, Modelo, **SKU** (único), Precio (>0), Stock (≥ 0), Origen, Garantía por defecto.
3. **Guardar**.
4. Confirmación en pantalla y registro en bitácora.

CU05 – Editar producto electrodoméstico

1. Menú **Catálogo** → **Productos**.
2. Buscar el producto → **Editar**.
3. Modifica los campos necesarios (p. ej., precio, stock, best seller).

4. **Guardar.**

CU08 – Listar productos

- Menú **Catálogo → Productos.**
- Usa **filtros** (categoría, marca, SKU, precio) y **ordenamiento**.
- Soporta **paginación** (siguiente/anterior).

CU06 – Registrar cliente

1. Menú **Clientes → Clientes → Nuevo.**
2. Completa: Nombre, Email (opcional), Teléfono, Ciudad.
3. **Guardar.**

CU07 – Editar cliente

- Menú **Clientes → Clientes.**
- Selecciona un cliente → **Editar → Guardar.**

CU09 – Listar clientes

- Menú **Clientes → Clientes.**
- Filtra por **nombre, email o ciudad.**
- Exporta (si está habilitado) o usa paginación.

CU10 – Bitácora

- Menú **Bitácora**.
- Filtra por **acción** (LOGIN, LOGOUT, CREATE_PRODUCT, etc.), **fecha**, **usuario**.
- Visualiza detalles del evento (entidad, id, metadata).

4. App móvil (Cliente) – visión rápida

- **Login** básico.
- **Ofertas**: listado y detalle.
- **Reportes** propios: ver/descargar PDF/Excel (si aplica).
- **Notificaciones**: recepción de avisos (mock en dev / FCM en prod).

5. Errores comunes y soluciones

- **SKU duplicado (409)**: cambia el SKU a uno único.
- **Sin permisos (403)**: solicita al admin el rol/permisos correctos.
- **No autenticado (401)**: inicia sesión nuevamente.
- **Stock insuficiente** (al preparar CU11): ajusta el stock en Catálogo.

6. Buenas prácticas de uso

- Mantén **SKU** coherentes (formato interno).
- Actualiza **stock** después de inventarios físicos.
- Verifica **Bitácora** ante dudas de cambios.
- Usa **filtros** para agilizar listados con gran volumen.

CONCLUSIÓN

SmartSales365 implementa de forma sólida los primeros 10 casos de uso (CU01–CU10) priorizando lo esencial: autenticación, gestión de catálogo y clientes, y auditoría transversal. Se definió y creó un modelo de datos normalizado, servicios REST con Django REST Framework, PWA administrativa y app móvil de consulta, todo dockerizado y con estructura para CI/CD.

La arquitectura propuesta por capas (Presentación, Servicios, Dominio/Infra) y la traza CU ↔ paquetes aseguran extensibilidad para las siguientes funcionalidades (ventas, comprobantes, garantías/reclamos, ofertas, voz y dashboard ML). El sistema queda listo para escalar en nube y para sumar reportes dinámicos y predicción de ventas con versionado de modelos.

RECOMENDACIÓN

Seguir el roadmap: abordar CU11–CU17 (Ventas, Comprobantes, Garantías/Reclamos, Ofertas) y luego CU18–CU20 (Push/Voz/Reportes por prompts).

Endurecer seguridad: afinar RBAC por permiso fino, rotación de claves, CORS restringido y HTTPS end-to-end.

Observabilidad: agregar métricas y alertas (status API, latencias, errores 4xx/5xx).

Rendimiento: índices adicionales según uso real; considerar **particionado** por fechas en ventas si crece el volumen.

UX: atajos y búsquedas persistentes en listados; formularios con validaciones en vivo.

ML: automatizar pipelines de entrenamiento (jobs programados), verificar deriva de datos y recalibrar el modelo periódicamente.

Backups/DR: política de **backups diarios** y pruebas de restauración trimestrales.

Pruebas: mantener cobertura $\geq 70\%$ en módulos críticos y agregar **E2E** para flujos de venta y comprobante cuando se activen.

BIBLIOGRAFÍA

Django Software Foundation. (2024). *Django Documentation* (v5.x).

Tom Christie et al. (2024). *Django REST Framework Documentation*.

OpenAPI Initiative. (2024). *OpenAPI Specification 3.0/3.1*.

PostgreSQL Global Development Group. (2024). *PostgreSQL 15/16 Documentation*.

Scikit-learn Developers. (2024). *RandomForestRegressor — scikit-learn User Guide*.

Google. (2024). *Firebase Cloud Messaging Documentation*.

React Team. (2024). *React & Vite Guides*.

OMG. (2017). *UML 2.5.1 Specification*.

Pressman, R. (2019). *Ingeniería del Software: Un enfoque práctico* (8^a ed.).

Larman, C. (2004). *UML y Patrones* (3^a ed.).

Sommerville, I. (2016). *Ingeniería del Software* (10^a ed.).

Fowler, M. (2004). *Patterns of Enterprise Application Architecture*.

Bass, Clements & Kazman. (2012). *Software Architecture in Practice* (3rd ed.).

ISO/IEC 25010. (2011). *Systems and software quality models*.

Anexos

- **GitHub:**

<https://github.com/Jelsc/SmartSales365-si2.git>

- **Azure (nube):**

- <http://57.154.17.34:5173>

