(DAS) Documento de Arquitectura del Sistema

**Versión:** 1.0  
**Proyecto:** **SuperLocaliza**  
**Fecha:** 07-10-2025  
**Documento mantenido por:** **Nicki Bravo (jefe de Proyecto)**  
**Aprobadores:** **Juan Pérez (Patrocinador principal)**, **María Gómez (Gerente Comercial)**

Identificación de Documento

|  |  |
| --- | --- |
| **Identificación** | DAS-SuperLocaliza-1.0 |
| **Proyecto** | SuperLocaliza |
| **Versión** | 1.0 |

|  |  |
| --- | --- |
| **Documento mantenido por** | Nicki Bravo — Jefe de Proyecto |
| **Fecha de ultima revisión** | 07/10/2025 |
| **Fecha de próxima revisión** | 30/11/2025 |

|  |  |
| --- | --- |
| **Documento aprobado por** | Nicki Bravo (jefe de Proyecto) - Juan Pérez (Patrocinador principal) |
| **Fecha de última aprobación** | 08/10/2025 |

Historia de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 07/10/2025 | |  | | --- | | 1.0 |  |  | | --- | |  | | Versión inicial del DAS (baseline) | Nicki Bravo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla de Contenidos

[1 Introducción 3](#_Toc84537077)

[1.1 Contexto del Problema 3](#_Toc84537078)

[1.2 Propósito 3](#_Toc84537079)

[1.3 Ámbito 3](#_Toc84537080)

[1.4 Definiciones, acrónimos y abreviaciones 3](#_Toc84537081)

[1.5 Referencias 3](#_Toc84537082)

[1.6 Resumen ejecutivo 3](#_Toc84537083)

[1.7 Representación 4](#_Toc84537084)

[2 Metas y Restricciones de la Arquitectura 5](#_Toc84537085)

[2.1 Metas de la arquitectura 5](#_Toc84537086)

[2.2 Restricciones de la Arquitectura 5](#_Toc84537087)

[2.3 Otros antecedentes y consideraciones 5](#_Toc84537088)

[3 Vista de Escenarios 6](#_Toc84537089)

[3.1 Modelo de Casos de Uso 6](#_Toc84537090)

[3.2 Casos de Usos Extendidos 6](#_Toc84537091)

[3.3 Especificación de los Escenarios de Calidad Relevantes 7](#_Toc84537092)

[4 Vista de Procesos 8](#_Toc84537093)

[5 Vista Lógica 9](#_Toc84537094)

[5.1 Parte Estructural ( Diagrama de Clases y Diagrama Relacional) 9](#_Toc84537095)

[5.1.1 Descripción de Clases 9](#_Toc84537096)

[5.1.2 Descripción de Tablas 10](#_Toc84537097)

[5.2 Parte Dinámica (Diagrama de Secuencias) 11](#_Toc84537098)

[6 Vista de Desarrollo o Despliegue 12](#_Toc84537099)

[7 Vista Fisica 13](#_Toc84537100)

[8 Decisiones de Diseño y Selección de Alternativas 14](#_Toc84537101)

[9 Análisis de Reutilización 15](#_Toc84537102)

Introducción

Contexto del Problema

Los supermercados reportan **dificultad para que los clientes localicen productos**, **inventario poco eficiente** y **escasa comunicación con clientes**, lo que afecta la satisfacción, la productividad del personal y las ventas. El proyecto **SuperLocaliza** aborda esto con una app para **buscar y ubicar productos** y una plataforma web para **gestionar inventario y promociones** sobre una **BD SQL centralizada**; la necesidad y validación aparecen en el Acta y minutas.

Propósito

Definir la **arquitectura del sistema** que permitirá implementar SuperLocaliza de forma segura, escalable y mantenible: vistas 4+1, decisiones tecnológicas y restricciones; servirá como **guía para diseño, desarrollo y validación** y como referencia común para los interesados del proyecto.

Ámbito

El sistema comprende:

* **Aplicación móvil** (Flutter) para clientes: **búsqueda y localización** en tienda y recepción de **promociones/notificaciones**.
* **Plataforma web** (Django/Python) para administración de **productos, clientes, proveedores, usuarios y promociones** (CRUD).
* **API/servicios** conectados a **BD SQL central** para información **en tiempo real** de inventario, promociones y ubicación.

Fuera de alcance en esta versión: **pagos en línea** e **integraciones externas avanzadas** (se consideran futuras extensiones según ERS).

Definiciones, acrónimos y abreviaciones

|  |  |
| --- | --- |
| **ACRONIMO** | **DESCRIPCION** |
| App | Aplicación móvil (Flutter) para Android/iOS. |
| CRUD | Create, Read, Update, Delete (Altas, Bajas, Consultas y Modificaciones). |
| BD | Base de Datos; motor SQL relacional. |
| SQL | Lenguaje de consulta para bases de datos relacionales. |
| API | Interfaz para comunicación entre app/web y el servidor. |
| UI/UX | Interfaz y experiencia de usuario. |
| MVP | Producto Mínimo Viable. |
| QA | Aseguramiento de Calidad. |
| Frontend | Interfaz con la que interactúa el usuario (app/web). |
| Backend | Lógica del servidor y reglas de negocio. |

Referencias

* Acta de Constitución
* ERS (Especificación de Requisitos de Software)
* Documento Caso de Uso Extendido
* Documento Proceso de Negocio (TO-BE)
* Documento Mockups
* Minuta de reunión Portafolio-semanal
* Documento de avance de sprint
* Diccionario de Datos
* Carta Gantt
* Matriz EDT
* Matriz Control de Cambio

Resumen ejecutivo

SuperLocaliza es una solución compuesta por **app móvil** para clientes y **plataforma web** para administración que permite **buscar y ubicar productos en tienda**, gestionar **inventario y promociones** y mejorar la **experiencia de compra**. La arquitectura propuesta separa **frontend**, **API** y **BD SQL**, priorizando **seguridad (RBAC y cifrado)**, **rendimiento** en búsquedas y **alta disponibilidad**. El alcance del MVP incluye: autenticación, mantenedores (productos, categorías, sucursales, ubicaciones), publicación de ofertas, mapa/ubicación en tienda, escaneo de código y notificaciones. Se trabajará con entregas incrementales y pruebas continuas. Beneficios esperados: reducción del tiempo de búsqueda de productos, mejor trazabilidad de stock y mayor conversión de promociones. El proyecto contempla un **plazo acotado**, con hitos y responsables definidos, y deja abiertas extensiones futuras (analytics avanzados e integraciones externas).

Representación

La arquitectura del sistema SuperLocaliza está representada siguiendo el enfoque de del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Escenarios**: Muestra los **casos de uso más relevantes** desde la perspectiva de los actores y los límites de sistema. Se incluyen dos contextos:
  + **Web Administrador (usuarios y roles):** registrar, editar y eliminar usuarios; gestionar roles y permisos; visualizar y exportar reportes.
  + **App Web Usuario (funciones principales):** registrarse, iniciar sesión, explorar y buscar productos, ver detalle y **ubicación en tienda**, gestionar perfil.  
    (Se añade además un diagrama combinado que muestra el alcance completo admin + usuario).
* **Vista de Procesos**: Describe el flujo de actividades entre Usuario/Administrador y Sistema, con validaciones y decisiones. Se modelan, entre otros:
  + **Registrarse** e **Iniciar sesión**.
  + **Explorar y buscar productos** y **ver detalle con ubicación**.
  + **Gestionar usuarios y roles**, **publicar promoción** y **exportar reportes**.  
    Todos los diagramas usan carriles (Usuario/Administrador y Sistema) y muestran resultado **OK/Error** y persistencia en BD.
* **Vista Lógica**: Presenta la estructura interna de la solución en dos niveles:
  + **Diagrama de Clases:** usuarios, administrador, roles, permisos, productos, categorías, pasillo/estantería/sucursal, promociones y reportes; con asociaciones y multiplicidades.
  + **Diagrama Relacional (ER):** tablas de usuarios, roles, permisos, mapeos, productos y catálogo, promociones, con **PK/FK** y relaciones 1:N y N:M.
  + **Diagrama de Secuencia:** **login** y navegación de **catálogo** (interacción Usuario → App Web → API → BD).
* **Vista de Desarrollo o Despliegue**: Detalla los componentes de software y cómo se empaquetan/despliegan:
  + **Auth en ambas plataformas (login y register):** Front Usuario y Admin → API → Servicio de Autenticación → BD de usuarios, emisión de token y correo de confirmación.
  + **Despliegue Overview:** clientes, front-ends, **Backend API**, jobs programados, servicio de auth, base de datos, almacenamiento estático y servicios de correo/notificaciones/observabilidad.
  + **Despliegue Detallado:** CDN/WAF y balanceador, **réplicas de API en contenedores**, scheduler de jobs, **Neon Postgres**, storage de objetos, cache opcional, integraciones SMTP/Slack y canal de CI/CD.
* **Vista Fisica**: Representa los nodos y zonas donde corre el sistema y sus conexiones:
  + **Clientes** (App Web Usuario y Web Admin).
  + **Servidor** con API Gateway/Balanceador, microservicio de autenticación y módulo de reportes.
  + **Base de Datos** (**Neon Postgres** en alta disponibilidad) y **almacenamiento estático**.
  + **Sistemas externos** (correo SMTP, notificaciones Slack, pasarela de pagos).  
    Se muestran las direcciones de flujo y dependencias entre cajas para reflejar la comunicación real.

Metas y Restricciones de la Arquitectura

A continuación, se revisan las metas y restricciones de la arquitectura.

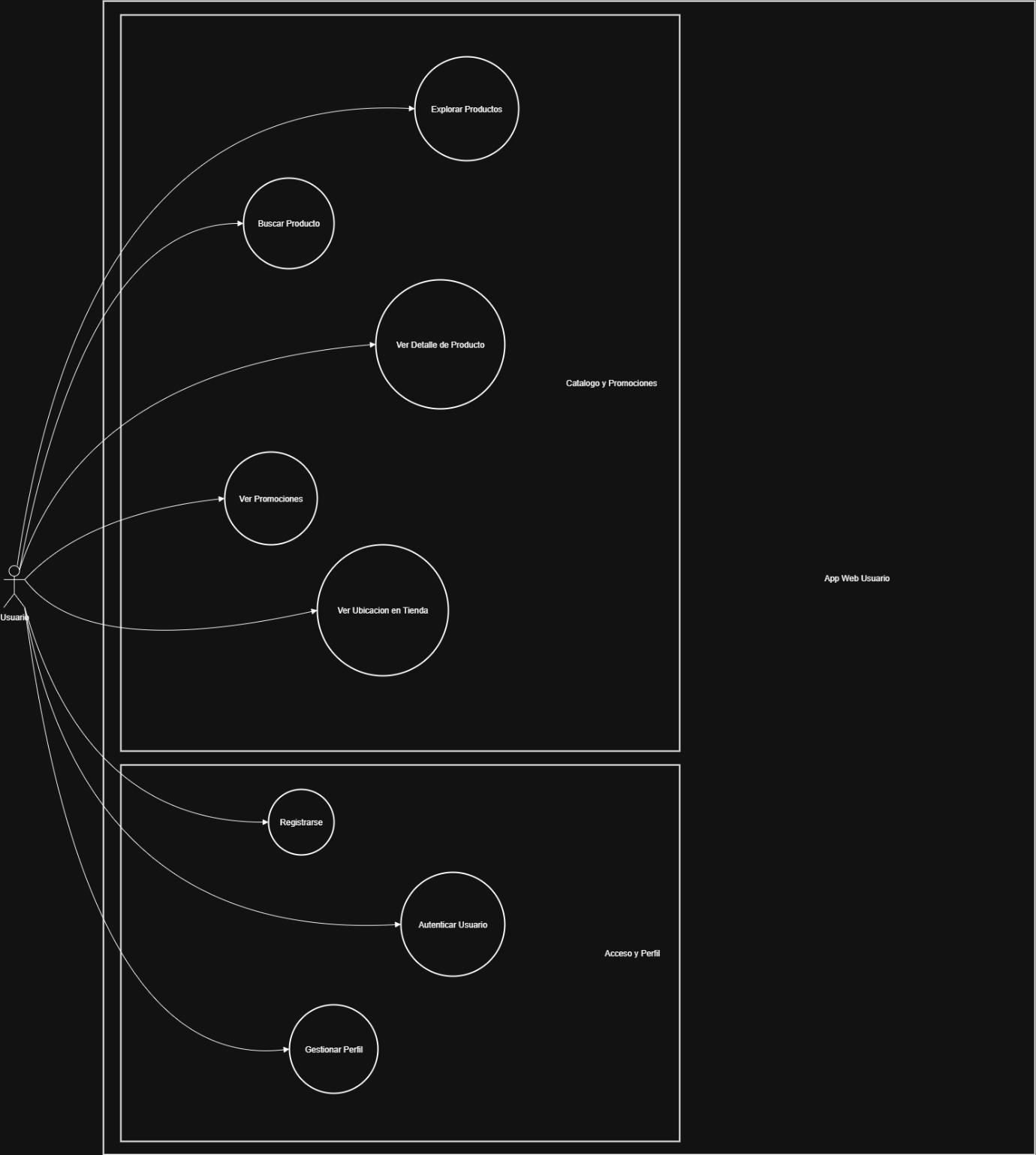
Metas de la arquitectura

De acuerdo con los requerimientos funcionales y no funcionales, se definen los siguientes **atributos de calidad** y metas medibles:

* **Desempeño y capacidad**
  + p95 tiempo de respuesta: **≤ 500 ms** para catálogos/búsquedas y **≤ 800 ms** para autenticación.
  + Concurrencia objetivo: **200** usuarios simultáneos; **10.000** usuarios activos mensuales.
* **Disponibilidad**
  + Servicio global **≥ 99,5%** mensual; base de datos gestionada con **alta disponibilidad**.
* **Escalabilidad**
  + **Horizontal** para API/Frontend con balanceador; **autoelevación** de plan BD si crece la carga.
* **Seguridad**
  + Capa TLS (**HTTPS** en todo el tráfico).
  + Passwords con **bcrypt/argon2**; **JWT** con expiración y refresh; bloqueo tras **5 intentos fallidos**.
  + **MFA** para Web Admin; control de **roles y permisos**.
  + Buenas prácticas **OWASP** (validaciones de entrada, CSRF, XSS, etc.).
* **Modificabilidad/Reuso**
  + Arquitectura **modular** (Frontend, API, Auth); contratos **REST** versionados.
  + Separación de **dominio** (usuarios/roles, catálogo, promociones, reportes).
* **Observabilidad y auditoría**
  + **Logs centralizados**, métricas y alertas; retención mínima **30 días**.
  + Bitácora de cambios en usuarios/roles y operaciones administrativas.
* **Portabilidad**
  + Despliegue en **contenedores**; CI/CD automatizado.
* **Experiencia de usuario**
  + **Primer render < 2 s** en conexión típica; accesibilidad nivel **WCAG AA**.
* **Respaldo y recuperación**
  + **Backups diarios** automáticos; RPO **≤ 24 h** y RTO **≤ 2 h** para BD.

Restricciones de la Arquitectura

Levantadas con los stakeholders; condicionan decisiones de diseño e implementación:



* **Tiempo de construcción**
  + **MVP** en **X semanas** (iteraciones quincenales); alcance priorizado por **valor**.
* **Presupuesto**
  + Preferencia por **servicios gestionados** y planes **free/low-cost** (hosting frontend, API, BD Postgres, correo).
* **Infraestructura**
  + **Base de datos Postgres gestionada** (alta disponibilidad); no se contempla infraestructura on-premise.
  + Despliegue en **nube pública** con **balanceador** y **CDN** para frontends.
* **Tecnología**
  + Frontend **web** (SPA) para Usuario y **backoffice web** para Administrador.
  + **API REST** como contrato único; **servicio de autenticación** centralizado.
  + Integraciones: **SMTP** (correo), **Slack/Webhook** (notificaciones); pasarela de pagos **opcional**.
* **Cumplimiento y datos**
  + Datos personales mínimos; **cifrado en tránsito** (TLS) y **en reposo** en BD/almacenamiento.
  + Políticas de **privacidad** y **retención** acordes al proyecto.
* **Compatibilidad**
  + Navegadores modernos (Chromium/Firefox/Edge/Safari actual); dispositivos móviles y escritorio.
* **Operación**
  + Ventanas de mantenimiento planificadas fuera de horario peak.
  + Sin soporte 24/7 en MVP; alertas en horario laboral (escalable según demanda).
* **Otros componentes de software**
  + Se privilegia **software libre**/licencias permisivas; no se contempla adquirir suites propietarias adicionales salvo aprobación.

Otros antecedentes y consideraciones

La empresa desarrolladora cuenta con un framework y prácticas que permiten satisfacer los requerimientos arquitectónicos del proyecto:

* **Frontend (React / Next.js).** Componentización, SSR/SSG para buen rendimiento, routing declarativo y accesibilidad; facilita mantenibilidad de App Web Usuario y Web Admin.
* **Backend (Django + Django REST Framework).** Arquitectura modular por apps, ORM para Postgres, API REST versionada, pruebas unitarias y reutilización.
* **Autenticación y seguridad.** JWT con expiración, hashing Argon2/Bcrypt, middleware de seguridad (XSS/CSRF), CORS controlado y roles/permisos para Web Admin; todo sobre TLS.
* **Persistencia y datos.** **Neon Postgres** gestionado (alta disponibilidad), migraciones con Django Migrations y backups automáticos con retención definida.
* **Observabilidad.** Logs centralizados, métricas y alertas; auditoría de operaciones administrativas (alta y cambios de usuarios/roles).
* **Integraciones.** SMTP para correo transaccional (confirmación/recuperación), Slack/Webhook para notificaciones operativas; pasarela de pagos opcional.
* **Entrega continua y despliegue.** Contenedores Docker, CI/CD con GitHub Actions, despliegue en nube pública con balanceador y CDN para frontends.
* **Calidad.** Pruebas unitarias e integración, linting y análisis de vulnerabilidades en el pipeline.

Vista de Escenarios

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

Modelo de Casos de Uso

Agregar el modelo de caso uso general del sistema

Ilustración 1: Diagrama de Caso Uso General del Sistema:

**Usuario (APP):**

Imagen que contiene Forma

El contenido generado por IA puede ser incorrecto.

**Administrador (Pagina Web):**

Diagrama, Dibujo de ingeniería

El contenido generado por IA puede ser incorrecto.

Casos de Usos Extendidos

Los casos de uso considerados son los más relevantes para el desarrollo de la arquitectura. Se adjunta el documento o planilla caso uso.

**Link para plantilla de caso de uso:**

[**Documento Caso Uso Extendido.docx - Documentos de Google**](https://docs.google.com/document/d/1A_J7wpNbJyIRfsdmwreft9qH1gaVQeb1/edit)

A continuación, se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso Extendido”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Nombre** | **Actores** | **Prioridad** |
| CU-001 | Autentificar Usuario | Todos los usuarios | Media |
| CU-002 | Registrarse | Usuario | Alta |
| CU-003 | Localizar Producto en Supermercado | Usuario | Alta |
| CU-006 | Explorar Catálogo | Usuario | Alta |
| CU-007 | Buscar Producto | Usuario | Alta |
| CU-008 | Ver Detalle y Ubicación en Tienda | Usuario | Alta |
| CU-012 | Gestionar Usuarios | Administrador | Alta |
| CU-013 | Gestionar Roles y Permisos | Administrador | Alta |
| CU-014 | Administrar Catálogo de Productos | Administrador | Alta |
| CU-016 | Exportar Reportes | Administrador | Media |

Especificación de los Escenarios de Calidad Relevantes

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

|  |  |  |
| --- | --- | --- |
| Identificador: ID-01 | | |
| Escenario(s): | | CU-001 Autentificar Usuario |
| Atributos de Calidad relevantes: | | Seguridad, Desempeño, Usabilidad |
| Componentes del Escenario | Estímulos: | El usuario ingresa email y contraseña y presiona Ingresar. |
| Fuente del estimulo | Usuario legítimo (considerar intentos maliciosos). |
| Ambiente: | Producción; API y BD disponibles; tráfico normal/alto. |
| Artefacto: | Módulo de autenticación (API Auth con JWT y roles). |
| Respuesta: | Validación de credenciales, emisión de token y redirección según rol. |
| Medida de Respuesta | p95 ≤ 800 ms; bloqueo tras 5 intentos fallidos; evento de login auditado < 1 s. |

|  |  |  |
| --- | --- | --- |
| Identificador: ID-02 | | |
| Escenario(s): | | CU-002 Registrarse |
| Atributos de Calidad relevantes: | | Seguridad, Usabilidad, Desempeño |
| Componentes del Escenario | Estímulos: | Envío de formulario de registro. |
| Fuente del estimulo | Usuario final. |
| Ambiente: | Producción; SMTP habilitado; tráfico normal. |
| Artefacto: | API de registro, verificación de correo, políticas de contraseña. |
| Respuesta: | Alta de cuenta, envío de correo de confirmación, estado “pendiente” hasta confirmar. |
| Medida de Respuesta | p95 ≤ 800 ms en creación; 100% correos duplicados rechazados; correo emitido < 30 s. |

|  |  |  |
| --- | --- | --- |
| Identificador: ID-03 | | |
| Escenario(s): | | CU-003 Localizar Producto en Supermercado / CU-008 Ver Detalle y Ubicación |
| Atributos de Calidad relevantes: | | Usabilidad, Desempeño |
| Componentes del Escenario | Estímulos: | El usuario abre el detalle de un producto desde la búsqueda. |
| Fuente del estimulo | Usuario final (móvil y escritorio). |
| Ambiente: | Producción; conectividad móvil promedio. |
| Artefacto: | API de detalle y ubicación (pasillo/estantería), vista de mapa/ruta. |
| Respuesta: | Se muestra ubicación clara y ruta sugerida dentro de la tienda. |
| Medida de Respuesta | Información visible ≤ 3 clics; tiempo a ubicación ≤ 5 s; ≥ 90% tasa de éxito en prueba de tareas |

|  |  |  |
| --- | --- | --- |
| Identificador: ID-04 | | |
| Escenario(s): | | CU-012 Gestionar Usuarios / CU-013 Gestionar Roles y Permisos |
| Atributos de Calidad relevantes: | | Seguridad, Auditoría |
| Componentes del Escenario | Estímulos: | Intento de acceder a ruta administrativa sin permisos. |
| Fuente del estimulo | Usuario autenticado sin rol adecuado o actor malicioso. |
| Ambiente: | Operación normal. |
| Artefacto: | Middleware de autorización por roles/permisos; JWT. |
| Respuesta: | Acceso denegado (403) y registro de auditoría. |
| Medida de Respuesta | 100% accesos no autorizados bloqueados; log con usuario, IP, endpoint y timestamp < 1 s. |

|  |  |  |
| --- | --- | --- |
| Identificador: ID-05 | | |
| Escenario(s): | | CU-016 Exportar Reportes |
| Atributos de Calidad relevantes: | | Operatividad, Observabilidad, Desempeño |
| Componentes del Escenario | Estímulos: | Administrador solicita exportación CSV/PDF con filtros. |
| Fuente del estimulo | Administrador o tarea programada. |
| Ambiente: | Producción; dataset hasta 100k filas. |
| Artefacto: | Servicio de reportes, cola/tareas, almacenamiento de archivos. |
| Respuesta: | Generación asíncrona; notificación y enlace de descarga; reintento ante falla. |
| Medida de Respuesta | Reporte ≤ 2 min (100k filas); alerta operativa < 1 min; reintento automático ≤ 5 min; MTTR ≤ 30 min. |

|  |  |  |
| --- | --- | --- |
| Identificador: ID-06 | | |
| Escenario(s): | | CU-014 Administrar Catálogo de Productos (impacta a CU-003 Localizar y CU-008 Ver Detalle y Ubicación) |
| Atributos de Calidad relevantes: | | Consistencia de datos, Operatividad, Modificabilidad |
| Componentes del Escenario | Estímulos: | El Administrador cambia pasillo/estantería o stock de un producto. |
| Fuente del estimulo | Administrador (Web). |
| Ambiente: | Producción, usuarios consultando desde la App. |
| Artefacto: | API de Mantenedores, BD Postgres (transacciones), caché de consultas, mecanismo de invalidación/cola de eventos. |
| Respuesta: | La actualización se confirma atómicamente; se invalida la caché y se propaga el cambio; la App muestra la nueva ubicación sin lecturas desactualizadas. |
| Medida de Respuesta | Confirmación de transacción < 1 s; invalidación de caché < 2 s; propagación a la App ≤ 5 s; 0% lecturas sucias/fantasmas; tasa de éxito de propagación ≥ 99,9%. |

Vista de Procesos

Ilustración 2: Diagramas de Actividades

**Registro:**

Imagen que contiene Forma

El contenido generado por IA puede ser incorrecto.

**Inicio Sesión:**

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto. Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**Explorar y Buscar:**

Imagen que contiene Forma

El contenido generado por IA puede ser incorrecto.

**Exportar Reportes:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Gestionar Usuarios y Roles:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Publicar Promociones:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Ver Detalle y Ubicación en Tienda:**

Diagrama

El contenido generado por IA puede ser incorrecto.

Vista Lógica

A continuación, se presenta una vista lógica de la aplicación expresado en tres diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (clases) y a la base de datos (modelo relacional).otra vista que representa la parte dinámica (secuencias).

Parte Estructural ( Diagrama de Clases y Diagrama Relacional)

Ilustración 3: Diagrama de Clases

Diagrama

El contenido generado por IA puede ser incorrecto.

### Descripción de Clases

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| CL-001 | Usuario | Cliente que usa la app. Atributos: id, nombre, apellido, email, password, teléfono, dirección, fechaRegistro. Operaciones: **registrarse(), autenticar(email, password), editarPerfil().** |
| CL-002 | Administrador | Usuario del backoffice con privilegios de gestión. Operaciones: **asignarRol(usuario, rol)**, **crearUsuario()**, **editarUsuario()**, **eliminarUsuario()**, **exportarReportes(filtro)**. |
| CL-003 | Producto | Artículo disponible para consulta. Atributos: id, nombre, precio, stock, categoriaId, pasilloId, estanteriaId, sucursalId. Operaciones: **actualizarStock(cant)**, **cambiarPrecio(valor)**. |
| CL-004 | Categoria | Clasificación de productos (p. ej., abarrotes, lácteos). Atributos: id, nombre. Relación **1–N** con **Producto**. |
| CL-005 | Pasillo | Pasillo físico de la tienda para ubicar productos. Atributos: id, nombre. Relación **1–N** con **Producto**. |
| CL-006 | Estanteria | Estantería/góndola dentro de un pasillo. Atributos: id, nombre. Relación **1–N** con **Producto**. |
| CL-007 | Sucursal | Tienda física. Atributos: id, nombre, dirección. Relación **1–N** con **Producto**. |
| CL-008 | Promocion | Oferta aplicada a un producto en un rango de fechas. Atributos: id, nombre, descripción, descuento, fechaInicio, fechaFin, productoId. Operación: **vigente(hoy)**. |
| CL-009 | Reporte | Salida administrable (CSV/PDF) con filtros y rangos. Atributos: id, tipo, formato, rangoInicio, rangoFin. Operaciones: **generar()**, **exportar()**. |
| CL-0010 | Rol | Rol de acceso (Admin, Usuario, etc.). Atributos: id, nombre, descripción. Relación **N–M** con **Permiso** (vía mapeo) y con **Usuario** (vía **UsuarioRol**). |
| CL-0011 | Permiso | Permiso granular asociado a acciones del sistema. Atributos: id, código, descripción. Relación **N–M** con **Rol**. |
| CL-0012 | UsuarioRol | Entidad de mapeo para asignar roles a usuarios. Atributos: id, usuarioId, rolId. Implementa relación **N–M** entre **Usuario** y **Rol**. |

Ilustración 4: Diagrama de Base Datos (Relacional)

Mapa de colores

El contenido generado por IA puede ser incorrecto.

### Descripción de Tablas

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| TB-001 | mantenedores\_cliente (CLIENTE) | Datos del cliente: nombres, apellidos, email, teléfono, dirección y fechas de registro/actividad. |
| TB-002 | mantenedores\_cliente\_groups | Puente **N–M**: asigna grupos (roles) a clientes. |
| TB-003 | mantenedores\_cliente\_user\_permissions | Puente **N–M**: permisos directos asociados a clientes. |
| TB-004 | mantenedores\_usuario | Usuarios del sistema (admin/app): credenciales y metadatos (username, email, flags is\_active/is\_staff, último login). |
| TB-005 | mantenedores\_usuario\_groups | Puente **N–M**: asigna grupos a usuarios. |
| TB-006 | mantenedores\_usuario\_user\_permissions | Puente **N–M**: permisos directos asociados a usuarios. |
| TB-007 | auth\_group | Grupos/roles del framework de autenticación. |
| TB-008 | auth\_permission | Permisos granulares vinculados a modelos/acciones. |
| TB-009 | auth\_group\_permissions | Puente **N–M**: permisos asociados a cada grupo. |
| TB-0010 | mantenedores\_sucursal | Sucursales o tiendas físicas: nombre y dirección. |
| TB-0011 | mantenedores\_categoria | Catálogo de categorías de producto (Abarrotes, Lácteos, etc.). |
| TB-0012 | mantenedores\_pasillo | Pasillos de la tienda usados para ubicar productos. |
| TB-0013 | mantenedores\_estanteria | Estanterías/góndolas dentro de un pasillo. |
| TB-0014 | mantenedores\_producto | Maestro de productos: nombre, **precio**, **stock** y FK a **categoría**, **pasillo**, **estantería** y **sucursal**. |
| TB-0015 | mantenedores\_promocion | Promociones/ofertas: nombre, descripción, **descuento**, **fecha\_inicio**, **fecha\_fin**; FK **producto\_id**. |
| TB-0016 | mantenedores\_proveedor | Proveedores: nombre, contacto, teléfono, email y dirección. |
| TB-0017 | django\_session | Sesiones de usuario: clave, datos y fecha de expiración. |
| TB-0018 | django\_migrations | Historial de migraciones aplicadas en la base de datos. |
| TB-0019 | django\_admin\_log | Bitácora de acciones administrativas (alta/edición/baja) con usuario y timestamp. |
| TB-0020 | django\_content\_type | Tipos de contenido (app/modelo) usados por el sistema de permisos. |

Parte Dinámica (Diagrama de Secuencias)

Ilustración 5: Diagramas de Secuencias

Diagrama

El contenido generado por IA puede ser incorrecto.

Vista de Desarrollo o Despliegue

En esta vista se describen las componente o modulos en las cuales se dividirá o implementará el sistema

Ilustración 6: Diagrama de componentes

**Auth Ambas Plataformas Login y Register:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Despliegue Overview:**

Diagrama

El contenido generado por IA puede ser incorrecto.

**Despliegue Detallado:**

Diagrama, Dibujo de ingeniería

El contenido generado por IA puede ser incorrecto.

Vista Fisica

En esta vista se despliegan los nodos que participan con el sistema.

Ilustración 7: Diagrama de Despliegue

Diagrama

El contenido generado por IA puede ser incorrecto.

Decisiones de Diseño y Selección de Alternativas

**Criterios:** tiempo de construcción, bajo riesgo, costo contenido, seguridad y escalabilidad básica.

* **Frontend (App principal + Web Admin)**
  + **Elegido:** React/Next.js — productividad, SSR/SSG, equipo con experiencia.
  + **Descartado:** Vue/Angular — mayor curva o menor experiencia interna.
* **Backend/API**
  + **Elegido:** Django + DRF — ORM sólido, serializadores, admin rápido.
  + **Descartado:** Node/Express o FastAPI — más “plumbing” para el MVP.
* **Auth**
  + **Elegido:** JWT (access/refresh) + roles/permisos.
  + **Descartado:** sesiones servidor / OAuth externo (más complejidad).
* **Base de datos**
  + **Elegido:** PostgreSQL gestionado (HA/backups).
  + **Descartado:** MySQL/NoSQL (sin ventajas para el dominio).
* **Esquema**
  + **Elegido:** modelo relacional normalizado (Catálogo/Promos y Usuarios/Roles/Sucursal).
  + **Descartado:** desnormalización temprana.
* **Despliegue**
  + **Elegido:** contenedores + CI/CD + hosting gestionado con balanceador.
  + **Descartado:** on-prem/IaaS puro (setup lento).
* **Archivos**
  + **Elegido:** storage de objetos para reportes/estáticos.
* **Observabilidad**
  + **Elegido:** logs estructurados, métricas y alertas simples.
* **Seguridad/Calidad**
  + **Elegido:** TLS, OWASP básico, políticas de contraseña, bloqueo por intentos, pruebas unitarias.

Análisis de Reutilización

* **Mantenedores reutilizables:** mismo módulo CRUD para **categorías, pasillos, estanterías, sucursales y productos**, variando solo el esquema/validaciones.
* **Servicio de autenticación/roles:** único servicio **JWT + roles/permisos** compartidos por **App Usuario** y **Web Admin**.
* **Validaciones y utilidades:** librería común (email/contraseña, paginación, manejo de errores, formateos).
* **Componentes UI:** **tabla, formulario, modal, filtros y paginación** reutilizados en todos los mantenedores (tema y estilos comunes).
* **Plantillas de reportes:** motor y layout **CSV/PDF** compartidos; cambian filtros y consultas.
* **Acceso a datos:** repositorios/ORM compartidos para entidades base (**Producto, Categoría, Ubicación, Promoción**).
* **Notificaciones y correo:** mismo conector **SMTP/Webhook** para confirmaciones y alertas.
* **Pipeline CI/CD:** workflows y contenedores **comunes** para frontend, API y jobs.
* **Pruebas y seguridad:** helpers de **tests**, middlewares **OWASP** y políticas de contraseña aplicados en todos los módulos.