

# DOCUMENTACIÓN TÉCNICA SIGAH

## Sistema de Gestión de Ayudas Humanitarias

**Versión:** 1.0.0  
**Fecha:** Diciembre 2024

### 1. DESCRIPCIÓN GENERAL

**SIGAH** es un sistema integral para la gestión de inventarios, solicitudes, entregas y devoluciones de ayudas humanitarias. Permite el control completo del flujo desde la recepción de productos hasta la entrega a beneficiarios.

#### Características Principales

- **Control de Inventario** con sistema FEFO (First Expiry, First Out)
- **Gestión de Kits** con composición flexible de productos
- **Flujo de Solicitudes y Entregas** con segregación de funciones
- **Sistema RBAC** (Role-Based Access Control) con permisos granulares
- **Notificaciones** por WhatsApp, email y sistema interno
- **Reportes** con exportación a Excel y PDF
- **Auditoría completa** de todas las operaciones

### 2. STACK TECNOLÓGICO

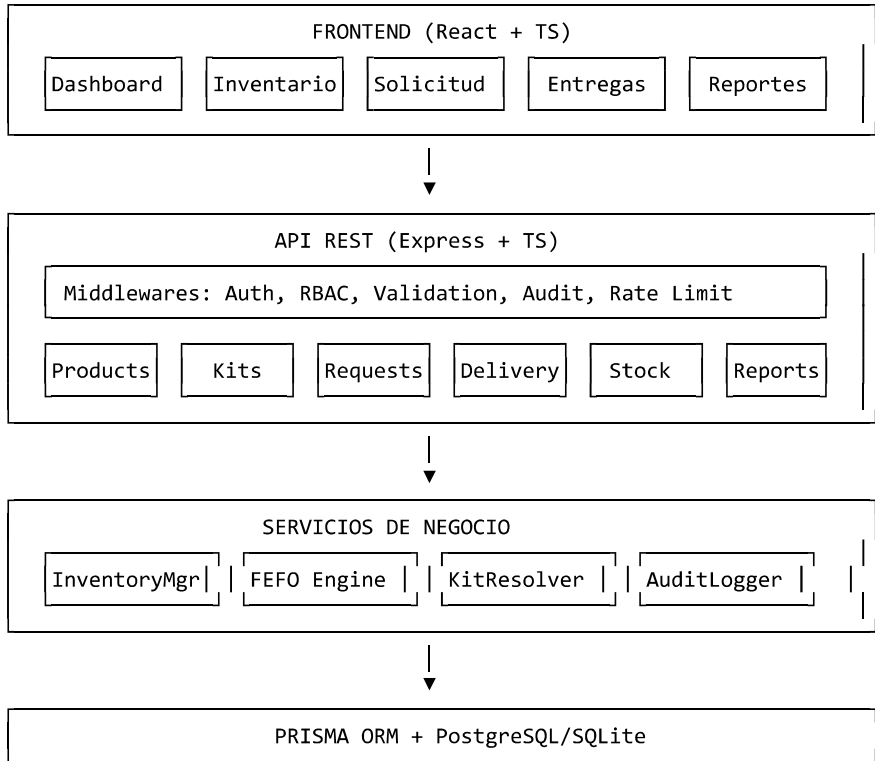
#### Backend

| Tecnología | Versión | Propósito | |-----|-----|-----| | Node.js | >= 18.0.0 | Runtime | | Express | 4.18.x | Framework web | | TypeScript | 5.3.x | Lenguaje tipado | | Prisma ORM | 5.7.x | ORM y migraciones | | SQLite/PostgreSQL | - | Base de datos | | Socket.io | 4.8.x | Comunicación real-time | | JWT | 9.x | Autenticación | | Winston | 3.x | Logging | | PDFKit | 0.14.x | Generación de PDFs | | XLSX | 0.18.x | Exportación a Excel |

#### Frontend

| Tecnología | Propósito | |-----|-----| | React | Framework UI | | TypeScript | Lenguaje tipado | | TailwindCSS | Estilos | | React Router | Navegación | | Recharts | Gráficos | | Lucide React | Iconografía |

### 3. ARQUITECTURA DEL SISTEMA



## 4. ESTRUCTURA DEL PROYECTO

```

sigah/
├── backend/
│   ├── src/
│   │   ├── config/           # Configuración de la app
│   │   ├── middleware/       # Auth, RBAC, validación, auditoría
│   │   ├── routes/           # Endpoints de la API (17 archivos)
│   │   ├── services/         # Lógica de negocio (12 servicios)
│   │   ├── utils/            # Utilidades
│   │   ├── index.ts          # Punto de entrada
│   │   └── seed.ts           # Datos de prueba
│   ├── prisma/
│   │   └── schema.prisma     # Modelo de datos
│   └── tests/                # Tests unitarios
├── frontend/
│   ├── src/
│   │   ├── components/       # Componentes reutilizables
│   │   ├── pages/            # Vistas (22 páginas)
│   │   ├── contexts/         # Context providers
│   │   ├── services/         # API clients
│   │   ├── hooks/            # Custom hooks
│   │   ├── i18n/             # Internacionalización
│   │   ├── types/            # Tipos TypeScript
│   │   └── App.tsx           # Rutas principales
├── README.md
└── ARCHITECTURE.md

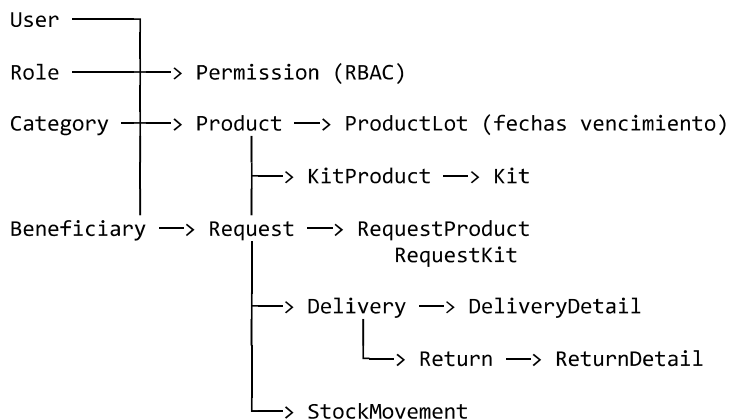
```

## 5. MODELO DE DATOS

### Entidades Principales

| Entidad | Descripción | |-----|-----| | **User** | Usuarios del sistema con roles y permisos | | **Role** | Roles con permisos asignables | | **Permission** | Permisos por módulo y acción | | **Category** | Categorías de productos | | **Product** | Productos del inventario | | **ProductLot** | Lotes con fecha de vencimiento | | **Kit** | Kits de ayuda compuestos | | **Beneficiary** | Beneficiarios de ayudas | | **Request** | Solicitudes de ayuda | | **Delivery** | Entregas realizadas | | **Return** | Devoluciones | | **StockMovement** | Movimientos de inventario | | **Notification** | Notificaciones del sistema | | **AuditLog** | Registro de auditoría |

### Diagrama Relacional Simplificado



## 6. MÓDULOS DEL SISTEMA

### 6.1 Dashboard

Ruta: /

- KPIs en tiempo real: Total productos, solicitudes pendientes, entregas del mes, alertas
- Gráficos: Movimientos por mes (barras), distribución por categoría (torta)
- Tablas: Productos próximos a vencer, últimas solicitudes
- Accesos rápidos según rol del usuario

### 6.2 Gestión de Inventario

**Rutas:** /inventory, /inventory-admin, /inventory/:id

- Lista de productos con filtros y búsqueda
- CRUD completo de productos
- Gestión de lotes con fechas de vencimiento (productos perecederos)
- Movimientos de stock: Entradas, salidas, ajustes
- Alertas: Stock bajo, productos próximos a vencer
- Sistema FEFO: Selección automática de lotes más próximos a vencer

## 6.3 Gestión de Kits

**Rutas:** /kits, /kits/:id

- Lista de kits configurados
- Creación de kits con composición de productos
- Verificación de disponibilidad en tiempo real
- Cálculo automático de kits disponibles según stock

## 6.4 Beneficiarios

**Ruta:** /beneficiaries

- Registro de beneficiarios con datos personales
- Tipos de documento: CC, TI, CE, Pasaporte
- Tipos de población: Desplazados, refugiados, vulnerables, etc.
- Historial de solicitudes por beneficiario

## 6.5 Solicitudes

**Rutas:** /requests, /requests/new, /requests/:id

**Estados del flujo:**

1. REGISTERED - Registrada
2. IN\_REVIEW - En revisión
3. APPROVED - Aprobada
4. REJECTED - Rechazada
5. DELIVERED - Entregada
6. PARTIALLY\_DELIVERED - Parcialmente entregada
7. CANCELLED - Cancelada

## 6.6 Entregas

**Rutas:** /deliveries, /deliveries/new/:requestId

**Flujo de entregas con segregación de funciones:**

1. Crear Solicitud (OPERADOR)
2. Autorizar (AUTHORIZER)
3. Recibir en Bodega (WAREHOUSE)
4. Preparar Productos (WAREHOUSE)
5. Marcar Lista (WAREHOUSE)
6. Entregar al Beneficiario (DISPATCHER)

**Importante:** Cada paso debe ser realizado por persona diferente.

## 6.7 Devoluciones

**Ruta:** /returns

- Registro de devoluciones sobre entregas
- Razones: Dañado, entrega incorrecta, no reclamado, vencido, duplicado
- Condición del producto: Bueno, dañado, vencido, parcial

- Reintegro automático al inventario (si aplica)

## 6.8 Reportes

Ruta: /reports

- Reporte de solicitudes por período
- Reporte de entregas con detalles
- Reporte de movimientos de inventario
- Exportación a PDF y Excel
- Filtros por fecha, estado, categoría

## 6.9 Gestión de Usuarios

Ruta: /users

- CRUD de usuarios
- Asignación de roles
- Activar/desactivar usuarios
- Delegación temporal de funciones
- Configuración de notificaciones (WhatsApp, Telegram, Email)

## 6.10 Gestión de Roles

Ruta: /roles

- CRUD de roles personalizados
- Permisos granulares por módulo y acción
- Roles del sistema protegidos (no eliminables)

## 6.11 Notificaciones

Rutas: /notifications, /send-notifications

- Centro de notificaciones en tiempo real
- Canales: Sistema interno, WhatsApp, Email, Telegram
- Niveles de criticidad: Informativo, Bajo, Normal, Medio, Alto, Crítico

## 6.12 Auditoría

Ruta: /inventory-audit

- Log completo de todas las operaciones
- Entidad, acción, usuario, timestamp
- Valores antes/después del cambio

# 7. SISTEMA DE ROLES Y PERMISOS (RBAC)

## Roles Predefinidos

Rol	Descripción	Permisos Principales
<b>Administrador</b>	Acceso total	Todo el sistema
<b>Autorizador</b>	Autoriza entregas	Ver inventario, autorizar entregas, aprobar solicitudes
<b>Bodega</b>	Gestiona inventario	CRUD inventario, recibir/preparar entregas
<b>Despachador</b>	Realiza entregas	Ver entregas, entregar a beneficiarios
<b>Operador</b>	Crea solicitudes	Beneficiarios, solicitudes, ver inventario
<b>Consulta</b>	Solo lectura	Ver dashboard, reportes

## Módulos y Acciones Disponibles

Módulo	Acciones Disponibles
dashboard	view
inventory	view, create, edit, delete, export, adjust
kits	view, create, edit, delete
beneficiaries	view, create, edit, delete, export
requests	view, create, edit, delete, approve, reject
deliveries	view, create, authorize, receive, prepare, deliver, cancel
returns	view, create, process
reports	view, export
users	view, create, edit, delete, activate
roles	view, create, edit, delete, assign

# 8. API ENDPOINTS PRINCIPALES

## Autenticación

Método	Endpoint	Descripción	Acción
POST	/api/auth/login	Iniciar sesión	GET   /api/auth/me
POST	/api/auth/change-password	Cambiar contraseña	POST   /api/auth/logout   Cerrar sesión

## Productos

Método	Endpoint	Descripción	Acción
GET	/api/products	Listar productos	POST   /api/products
POST	/api/products	Crear producto	GET   /api/products/:id   Obtener producto    PUT   /api/products/:id   Actualizar producto    DELETE   /api/products/:id   Desactivar producto    GET   /api/products/:id/lots   Lotes del producto    POST   /api/products/:id/lots   Agregar lote

## Inventario

Método	Endpoint	Descripción	Acción
GET	/api/inventory/stock	Stock actual	GET   /api/inventory/movements   Historial movimientos    POST   /api/inventory/adjustment   Ajuste manual    GET   /api/inventory/expiring   Productos por vencer    GET   /api/inventory/summary   Resumen inventario

## Kits

Método	Endpoint	Descripción	Acción
GET	/api/kits	Listar kits	POST   /api/kits   Crear kit    GET   /api/kits/:id   Obtener kit    PUT   /api/kits/:id   Actualizar kit    GET   /api/kits/:id/availability   Disponibilidad

## Beneficiarios

Método	Endpoint	Descripción	Acción
GET	/api/beneficiaries	Listar beneficiarios	POST   /api/beneficiaries   Crear beneficiario    GET   /api/beneficiaries/:id   Obtener beneficiario    PUT   /api/beneficiaries/:id   Actualizar

## Solicitudes

Método	Endpoint	Descripción	Acción
GET	/api/requests	Listar solicitudes	POST   /api/requests   Crear solicitud    GET   /api/requests/:id   Obtener solicitud    PUT   /api/requests/:id   Actualizar    PATCH   /api/requests/:id/status   Cambiar estado

## Entregas

Método	Endpoint	Descripción	Acción
GET	/api/deliveries	Listar entregas	POST   /api/deliveries   Crear entrega    PATCH   /api/deliveries/:id/authorize   Autorizar    PATCH   /api/deliveries/:id/receive   Recibir en bodega    PATCH   /api/deliveries/:id/prepare   Preparar    PATCH   /api/deliveries/:id/ready   Marcar lista    PATCH   /api/deliveries/:id/deliver   Entregar

## Dashboard y Reportes

Método	Endpoint	Descripción	Acción
GET	/api/dashboard/summary	KPIs	GET   /api/dashboard/charts   Datos gráficos    GET   /api/reports/requests   Reporte solicitudes    GET   /api/reports/deliveries   Reporte entregas    GET   /api/reports/export/:type/:format   Exportar

# 9. INSTALACIÓN Y CONFIGURACIÓN

## Requisitos Previos

- Node.js >= 18.0.0
- PostgreSQL >= 14.0 (o SQLite para desarrollo)
- npm >= 8.0

## Instalación Backend

```
cd backend
```

```
# Configuración
cp .env.example .env
# Editar DATABASE_URL y JWT_SECRET en .env
```

```
# Instalación
npm install
npm run db:generate
npm run db:push
npm run db:seed # Datos de prueba
```

```
# Ejecutar en desarrollo
npm run dev
```

**Variables de entorno requeridas:**

```
DATABASE_URL="postgresql://user:pass@localhost:5432/sigah"
JWT_SECRET="clave-secreta-muy-segura"
PORT=3001
```

## Instalación Frontend

```
cd frontend
npm install
npm run dev
```

El frontend estará disponible en <http://localhost:3000> El backend estará disponible en <http://localhost:3001>

## 10. USUARIOS DE PRUEBA

Después de ejecutar `npm run db:seed`:

Rol	Email	Contraseña	Administrador	admin@sigah.com	admin123	Autorizador	autorizador@sigah.com	admin123	Bodega	bodega@sigah.com	admin123	Despachador	despachador@sigah.com	admin123	Operador	operador@sigah.com	admin123	Consulta	consulta@sigah.com	admin123

## 11. DECISIONES DE DISEÑO

- FEFO (First Expired, First Out):** Para productos perecederos, el sistema selecciona automáticamente lotes con fecha de vencimiento más próxima.
- Fechas de vencimiento por lote:** Cada lote tiene su propia fecha de vencimiento, no el producto.
- Segregación de funciones:** Cada paso del flujo de entrega debe ser realizado por persona diferente.
- Soft Delete:** Productos, kits y usuarios se desactivan en lugar de eliminarse para mantener integridad referencial.
- Stock nunca negativo:** Se valida antes de cada operación de salida.
- Auditoría completa:** Todas las operaciones críticas se registran con valores antes/después.

## 12. PANTALLAS DEL SISTEMA

### Capturas de Pantalla Requeridas

#	Pantalla	Ruta	Descripción
1	Login	/login	Pantalla de inicio de sesión
2	Dashboard	/	Panel principal con KPIs y gráficos
3	Inventario	/inventory	Lista de productos con filtros
4	Detalle Producto	/inventory/:id	Información completa y lotes
5	Gestión Inventario	/inventory-admin	CRUD de productos y ajustes
6	Kits	/kits	Lista de kits configurados
7	Detalle Kit	/kits/:id	Composición y disponibilidad
8	Beneficiarios	/beneficiaries	Lista y gestión de beneficiarios
9	Solicitudes	/requests	Lista de solicitudes
10	Nueva Solicitud	/requests/new	Formulario de solicitud
11	Entregas	/deliveries	Panel de entregas con flujo
12	Devoluciones	/returns	Gestión de devoluciones
13	Reportes	/reports	Generación y exportación
14	Usuarios	/users	Gestión de usuarios
15	Roles	/roles	Gestión de roles y permisos
16	Notificaciones	/notifications	Configuración notificaciones

**Nota:** Agregar capturas de pantalla ejecutando la aplicación y tomando screenshots con Win + Shift + S

## 13. CONTACTO Y SOPORTE

Para soporte técnico o consultas sobre el sistema SIGAH, contactar al equipo de desarrollo.

Documento generado automáticamente - SIGAH v1.0.0