**Documento Técnico Detallado - Stash (Versión Supabase/PostgreSQL)**

### 1. Introducción

Stash es una plataforma web integral para la gestión de inventario, ventas, clientes y análisis financiero, orientada a negocios pequeños y medianos. El sistema cuenta con control de acceso por roles, donde los usuarios pueden registrarse como **propietarios** (crean su negocio) o **empleados** (se unen a uno existente).

Esta versión modernizada se apoya completamente en **Supabase** para: - **Autenticación** (con roles personalizados) - **Base de datos PostgreSQL** (relacional, escalable y cloud-native) - **Almacenamiento** (para logos e imágenes internas del negocio)

El backend está construido en **Java Spring Boot** y consume Supabase a través de su API y cliente PostgreSQL nativo.

### 2. Requerimientos Funcionales

#### 2.1 Autenticación y Usuarios

* Registro de usuario mediante Supabase Auth:
  + Nombre completo
  + Correo (email)
  + Contraseña segura (validada por Supabase)
  + Selección de rol (propietario o empleado)
* Verificación por correo automática (activada desde Supabase)
* Post-login:
  + Selección: “Soy propietario” o “Soy empleado”
  + Propietario: crea negocio
  + Empleado: ingresa código de negocio

#### 2.2 Negocios (para Propietarios)

* Creación de negocio:
  + Nombre
  + Tipo (categoría)
  + Dirección y ciudad
  + Logo cargado a Supabase Storage
  + Código de invitación (generado aleatoriamente)
* Personalización:
  + Horario de atención
  + Impuestos y moneda
  + Configuración de locales

#### 2.3 Inventario

* Categorías jerárquicas hasta 3 niveles
* Productos:
  + SKU automático
  + Proveedor principal
  + Stock mínimo y actual
  + Variantes (color/talla)
  + Histórico de precios
* Proveedores:
  + Nombre y contacto
  + Sistema de valoración

#### 2.4 Ventas y Punto de Venta

* POS:
  + Búsqueda por código, nombre, categoría
  + Division de pagos y calculadora integrada
  + Asociación con cliente
* Deudas:
  + Registro de pagos pendientes
  + Generación de letras y abonos

#### 2.5 Reportes y Análisis

* Dashboard del negocio:
  + Ventas por semana/mes
  + Inventario bajo
  + Clientes con deudas
  + Rendimiento por empleado
* Exportaciones:
  + PDF, CSV, Excel

### 3. Arquitectura y Base de Datos

#### 3.1 Autenticación Supabase

* Supabase gestiona login, registro y verificación de correo
* Uso de “user metadata” para guardar rol: propietario o empleado
* Backend valida los tokens JWT firmados por Supabase con su clave pública (JWK)
* UID de Supabase se usa como clave foránea en las demás tablas

#### 3.2 Base de Datos PostgreSQL (estructura general)

Tablas principales necesarias:

* users
* businesses
* employees
* categories
* products
* product\_variants
* suppliers
* sales
* sale\_details
* clients
* debts
* debt\_payments
* expenses
* balance\_reports
* notifications
* uploads (para registrar archivos de Supabase Storage)

Todas las tablas usarán uuid como clave primaria, alineado con Supabase Auth.

### 4. Endpoints Relevantes del API

| Método | Ruta | Descripción |
| --- | --- | --- |
| POST | /api/auth/verify | Verifica token Supabase y crea usuario si no existe |
| POST | /api/business/create | Crear negocio (solo propietarios) |
| POST | /api/business/invite | Empleado se une con código |
| GET | /api/business/dashboard | KPIs del negocio |
| POST | /api/inventory/create | Crear producto |
| GET | /api/inventory/alerts | Ver alertas de stock bajo |
| POST | /api/sales/register | Registrar una venta completa |
| POST | /api/clients/create | Registrar cliente nuevo |
| POST | /api/debts/register | Registrar deuda |
| POST | /api/debts/pay | Pagar abono de deuda |
| GET | /api/reports/balance | Obtener balance del negocio |

### 5. Consideraciones de Implementación

#### 5.1 Seguridad

* Verificación de JWT Supabase en cada request
* Reglas de acceso por rol (middleware Spring Boot)
* Reglas en Supabase Storage para acceso por negocio

#### 5.2 Performance

* Uso de vistas materializadas para KPIs
* Indexado por business\_id, user\_id y fecha
* Limitación de resultados en queries pesadas

#### 5.3 Almacenamiento (Supabase Storage)

* Logos de negocio e imágenes de productos se almacenan en Supabase Storage
* URL firmada temporalmente para acceso seguro desde el frontend
* Subidas controladas desde frontend o backend según permisos

### 6. Flujos de Usuario Clave

#### 6.1 Propietario

1. Se registra y elige “propietario”
2. Se verifica correo
3. Entra al sistema, crea su negocio
4. Se le genera código de invitación

#### 6.2 Empleado

1. Se registra y elige “empleado”
2. Verifica correo
3. Inicia sesión, ingresa código de invitación
4. Se asocia al negocio y obtiene su rol

Este documento puede ser consumido por desarrolladores humanos o IA para construir la aplicación usando Supabase, PostgreSQL y Spring Boot como tecnologías principales.