

PUNTOS IMPORTANTES:

- **Un usuario solo tiene un rol:** Si un usuario quiere trabajar en varias empresas tendrá que crearse un usuario distinto para cada empresa
- **Un usuario que tiene el rol de vendedor en una empresa A no puede ser vendedor de otra empresa B:** Esto para evitar la fuga de datos, en caso de que una cuenta fuera robada lo ideal es que solo los datos de la empresa en la que trabaja sean expuestos. En cambio si un mismo vendedor trabaja en dos o más empresas si su cuenta es robada la exposición de información aumenta a dos empresas afectadas
- **Un usuario con rol de cliente puede ser cliente de varias empresas a la vez:** Cuando un usuario se loguea se le mostrará las empresas a las que tiene acceso a ver sus productos (Las empresas pueden aceptar o no a sus clientes). Si el usuario sólo está registrado en una empresa entonces se le mostrará directamente esta empresa
- **Usuario con el rol admin:** Es el que registra a las empresas con o sin suscripción
- **Una empresa registra a sus usuarios con el rol de vendedores:** Puede tener muchos o solo uno
- **Un cliente puede registrarse y solicitar el acceso a empresas**
- **Relacion "cuenta_con":** Dentro de esta relación se guardará la información de los productos que se compre
- **Relación "renueva":** En esta relación se guardará los productos que se comprarán de para renovar stock

Tablas

- Rol: tabla donde se guardan los tipos de roles
- Usuario: Generalización que hereda su PK a Cliente, Admin, Usuario_Empresa
- Cliente: Todos los clientes con datos como nit, telefono
- Admin: Yo datos como correo, password, nombre
- Usuario_Empresa: Puede ser dueño o vendedor de acuerdo al rol, administra su empresa
- Empresa: Empresa como tal
- Venta: Guarda las ventas
- Compra_Stock: guarda las renovaciones del stock
- Plan: planes vip, mensual, anual, free, etc
- Suscripcion: todas las suscripciones existentes
- Producto: todos los productos
- Proveedor: los proveedores a los que se les compra para renovar el stock
- Archivo: archivos como imagenes de los productos
- Categoria: categorias de los productos

Relaciones entre tablas:

- Generalización Usuario -> Cliente, Admin, Usuario_empresa (se heredará solo la PK de usuario a las subclases)
- Usuario -> Rol (un usuario solo puede tener un rol y un rol puede pertenecer a varios usuarios)
- Cliente -> Empresa (un cliente pertenece a una empresa y una empresa puede tener muchos clientes N:M) al ser una relacion N:M se creará una tabla que almacene los que clientes son de que empresa (es necesario porque un cliente podría no haber comprado nada pero la empresa lo registró como su cliente)

- Admin -> Empresa (un administrador registra varias empresas y una empresa es registrada por un admin 1:N)
- Usuario_Empresa -> Empresa (un usuario_empresa administra una empresa, una empresa puede ser administrada por varios usuario_empresa N:1)
- Cliente -> Producto (un cliente reserva varios productos y un producto puede ser reservado por varios clientes N:M) al ser una relacion N:M se creará una tabla que almacene los detalles de la reserva
- Cliente -> Venta (un cliente hace varias compras y una compra es solo hecha por un cliente 1:N)
- Venta -> Producto (una venta esta compuesta por varios productos y un producto puede estar en varias ventas M:N) al ser una relacion N:M se creará una tabla que almacene los detalles de la venta
- Usuario_Empresa -> Compra_Stock (un vendedor renueva stock varias veces y una compra puede se hecha por un vendedor 1:N)
- Compra_Stock -> Producto (una compra puede tener varios productos y un producto puede estar en varias compras N:M) al ser una relacion N:M se creará una tabla que almacene los detalles de la renovacion de stock
- Empresa -> Suscripcion (una empresa puede estar registrada en un plan y una suscripcion solo tiene una empresa 1:1)
- Plan -> Suscripcion (un plan puede estar en varias suscripciones y una suscripcion solo puede tener un plan 1:N)
- Empresa -> Producto (una empresa tiene varios productos y un producto solo pertenece a una empresa 1:N)
- Empresa -> categoria (una empresa tiene varias categorias y una categoria solo pertenece a una empresa 1:N)
- Producto -> Proveedor (un producto puede ser poveido por un proveedor y un proveedor provee varios productos N:1)
- Producto -> Archivo (un producto puede tener varios archivos y un archivo pertenece solo a un producto 1:N)
- Producto -> Categoria (un producto pertenece a una categoria y una categoria tiene varios productos)

Atributos:

- **Rol:** id(PK), rol, descripción, estado
- **Usuario:** id(PK), rol_id (FK de Rol), email, password_hash, estado(activado, bloqueado), fecha_creación, fecha_modificacion, ultimo_login
- **Cliente:** usuario_id(PK + FK de Usuario), nit, telefono, direccion, fecha_registro
- **Admin:** usuario_id(PK + FK de Usuario), nombre, telefono
- **Usuario_Empresa:** usuario_id(PK + FK de Usuario), empresa_id (FK de empresa), estado, fecha_asignación
- **Empresa:** id(PK), admin_id (FK de Admin), nombre, razon_social, nit, direccion, telefono, email, estado
- relacion **Tiene:** cliente_id, empresa_id (PK), fecha_registro, estado
- **Plan:** id(PK), nombre, precio, duracion_dias, limite_productos, limite_usuarios, estado
- **Suscripcion:** id(PK), empresa_id(FK de Empresa unico), plan_id(FK de Plan), fecha_inicio, fecha_fin, estado, fecha_creacion
- **Categoria:** id(PK), empresa_id(FK de Empresa), nombre, descripcion, estado

- **Proveedor:** id(PK), empresa_id(FK de empresa), nombre, telefono, email, direccion, estado
- **Producto:** id(PK), empresa_id(FK de Empresa), categoria_id(FK de Categoria), proveedor_id(FK de Proveedor), nombre, descripcion, precio, stock_actual, stock_minimo, estado, fecha_creacion
- **Archivo:** id(PK), producto_id(FK de Producto), binario, orden, fecha_creacion
- **Venta:** id(PK), empresa_id(FK de Cliente), fecha, total, estado
- relacion **cuenta_con:** venta_id(FK de Venta), producto_id(FK de Producto), cantidad, precio_unitario, subtotal
- relacion **reserva:** cliente_id(FK de Cliente), producto_id(FK de Producto), cantidad, fecha_reserva, estado
- **Compra_Stock:** id(PK), empresa_id(FK de Empresa), usuario_empresa_id(FK de Usuario_empresa), proveedor_id(FK de Proveedor), fecha, total, estado
- **relacion renueva:** id(PK), compra_stock_id(FK de Compra_Stock), producto_id(FK de Producto), cantidad, costo_unitario, subtotal

Modelo Relacional:

- **Rol:** id_rol(PK), rol, descripción, estado
- **Usuario:** id_usuario(PK), email, password_hash, estado(activado, bloqueado), fecha_creación, fecha_modificación, ultimo_login, rol_id (FK de Rol).
- **Cliente:** id_usuario(PK), nit, telefono, direccion, fecha_registro, nombre
- **Admin:** id_usuario(PK), nombre, telefono
- **Usuario_Empresa:** id_usuario(PK), estado, fecha_asignación, id_empresa(FK)
- **Empresa:** id_empresa(PK), nombre, razon_social, nit, direccion, telefono, email, estado, admin_id (FK de Usuario_Empresa)
- **relacion Tiene:** id_usuario(PK), id_empresa(PK), fecha_registro, estado
- **Plan:** id_plan(PK), nombre, precio, duracion_dias, limite_productos, limite_usuarios, estado, descripcion
- **Suscripcion:** id_Suscripcion(PK), fecha_inicio, fecha_fin, estado, fecha_creacion, plan_id(FK de Plan), id_empresa (FK)
- **Producto:** id_producto(PK), nombre, descripcion, precio, stock_actual, stock_minimo, estado, fecha_creacion, id_categoria(FK de Categoria), id_proveedor(FK de Proveedor), id_empresa (FK)
- **Proveedor:** id_proveedor(PK), nombre, teléfono, email, dirección, estado, id_empresa (FK)
- **Archivo:** id_archivo(PK), binario, orden, fecha_creacion, id_producto(FK de Producto).
- **Categoria:** id_categoria(PK), nombre, descripcion, estado, id_empresa(FK de Empresa)
- **Venta:** id_venta(PK), fecha, total, estado, id_cliente(FK de Cliente)
- **relacion cuenta_con:** id_venta(FK de Venta), id_producto(FK de Producto), cantidad, precio_unitario, subtotal

- **relacion reserva:** id_cliente(FK de Cliente), id_producto(FK de Producto), cantidad, fecha_reserva, estado, fecha_expiracion
- **Compra_Stock:** id_stock(PK), fecha, total, estado, id_userEmpresa(FK de Usuario Empresa)
- **relacion renueva:** id_producto(PK), id_stock(FK de Compra_Stock), cantidad, costo_unitario, subtotal

Sugerencias:

intentos fallidos en usuario para proteger

en cliente poner nombre

empresa agregar el plan activo (si habra consultas con esto)

plan poner una descripcion

cambiar atributos repetido de cuenta con igual a renueva

REGISTRO, AUTENTICACION Y LOGUEO

BACKEND

1 Models:

- Rol, Usuario, Empresa, Usuario_Empresa Plan y Suscripcion (core/[models.py](#)) (isaac)
- Admin, Vendedor, Cliente (cuentas/[models.py](#)) (araceli)

2 Serializadores (cuentas/[serializers.py](#)): (araceli)

- Para registro de usuarios diferentes por rol
- Para login
- Para perfiles de usuario
- Para cambio de contraseña

3 Views (cuentas/[views.py](#)): (isaac)

- RegisterView (registro para diferentes roles)
- LoginView (autenticacion JWT)
- LogoutView
- UserProfileView (Obtener perfil de usuario logueado)
- ChangePasswordView
- EmpresaAccessView (el cliente pueda ver sus empresas)

4 URLs: (cada quien)

- Rutas para todas las vistas de autenticacion (cuentas/urls.py)
- Rutas especificas por rol (cuentas/urls.py)
- Incluir las rutas de cuentas.urls (backend/urls.py)

5 Configuracion Django (backend/[settings.py](#)): (araceli)

- Configurar AUTH_USER_MODEL (si usas modelo custom)
- Configurar JWT (instalar [djanogetframework-simplejwt](#))
- Configurar CORS para el frontend
- Configurar base de datos multi-tenant

- Middleware para determinar empresa actual

6 Middleware para tenant (core/[middleware.py](#)): (isaac)

- Middleware para identificar la empresa del usuario actual
- Lógica para filtrar queries por id_empresa
- Manejo de subdominios o headers para identificar tenant

7 Managers y QuerySets (core/[managers.py](#)): (araceli)

- Custom Manager para filtrar por empresa automáticamente
- Querysets que incluyan id_empresa en todos los filtros

8 Permisos (cuentas/[permissions.py](#)): (isaac)

- IsAdmin (solo rol admin)
- IsVendedor (solo rol vendedor + verifica empresa)
- IsCliente (solo rol cliente)
- BelongsToEmpresa (verifica que usuario pertenezca a la empresa)
- IsEmpresaActive (verifica suscripción activa)

9 Signals (cuentas/[signals.py](#)):(cada quien)

- Crear perfil específico al crear usuario
- Asignar permisos iniciales
- Crear relaciones usuario-empresa automáticamente