```
CREATE TABLE rol
(
    id_rol integer NOT NULL DEFAULT nextval('roles_id_rol_seq'::regclass),
    nombre character varying(30) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT roles_pkey PRIMARY KEY (id_rol),
    CONSTRAINT roles_nombre_key UNIQUE (nombre)
)
CREATE TABLE usuario
(
    id_usuario integer NOT NULL DEFAULT nextval('usuarios_id_usuario_seq'::regclass),
    nombre character varying(100) COLLATE pg_catalog."default" NOT NULL,
    correo character varying(100) COLLATE pg_catalog."default" NOT NULL,
    "contraseña" character varying(64) COLLATE pg_catalog."default" NOT NULL,
    direccion text COLLATE pg_catalog."default",
    telefono character varying(20) COLLATE pg_catalog."default",
    id_rol integer NOT NULL DEFAULT 6,
    CONSTRAINT usuarios_pkey PRIMARY KEY (id_usuario),
    CONSTRAINT usuarios_correo_key UNIQUE (correo),
    CONSTRAINT fk_rol_usuario FOREIGN KEY (id_rol)
        REFERENCES public.rol (id_rol) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

CREATE TABLE categoria
(
    id_categoria integer NOT NULL DEFAULT nextval('categoria_id_categoria_seq'::regclass),
    nombre character varying(50) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT categoria_pkey PRIMARY KEY (id_categoria)
```

```
)

CREATE TABLE producto

(

    id_producto integer NOT NULL DEFAULT nextval('productos_id_producto_seq'::regclass),

    nombre character varying(50) COLLATE pg_catalog."default" NOT NULL,

    descripcion text COLLATE pg_catalog."default",

    precio numeric(10,2) NOT NULL,

    codigo character varying(10) COLLATE pg_catalog."default" NOT NULL GENERATED ALWAYS AS
(((lpad((id_producto)::text, 3, '0'::text) || upper(SUBSTRING(nombre FROM 1 FOR 3))) ||
upper("right"((nombre)::text, 1)))) STORED,

    id_categoria integer,

    CONSTRAINT productos_pkey PRIMARY KEY (id_producto),

    CONSTRAINT producto_id_categoria_fkey FOREIGN KEY (id_categoria)

        REFERENCES public.categoria (id_categoria) MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

)

CREATE TABLE Inventario

(

    "Id_producto" integer NOT NULL,

    cantidad integer,

    id_inventario integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1
MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),

    CONSTRAINT inventario_pkey PRIMARY KEY (id_inventario),

    CONSTRAINT inventario_producto_fk FOREIGN KEY ("Id_producto")

        REFERENCES public.producto (id_producto) MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE CASCADE

)
```

```
CREATE TABLE pedidos
(
    id_pedido integer NOT NULL DEFAULT nextval('pedidos_id_pedido_seq'::regclass),
    id_usuario integer NOT NULL,
    fecha timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    estado character varying(20) COLLATE pg_catalog."default",
    total numeric(10,2) NOT NULL,
    CONSTRAINT pedidos_pkey PRIMARY KEY (id_pedido),
    CONSTRAINT pedidos_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES public.usuario (id_usuario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT pedidos_estado_check CHECK (estado::text = ANY (ARRAY['Creado'::character varying, 'Pendiente de Pago'::character varying, 'Pagado'::character varying, 'En Preparación'::character varying, 'Enviado'::character varying, 'Entregado'::character varying, 'Cancelado'::character varying]::text[]))
)


CREATE TABLE ingrediente
(
    id_ingrediente integer NOT NULL DEFAULT nextval('ingrediente_id_ingrediente_seq'::regclass),
    codigo character varying(8) COLLATE pg_catalog."default" NOT NULL GENERATED ALWAYS AS (((lpad((id_ingrediente)::text, 3, '0'::text) || upper(SUBSTRING(nombre FROM 1 FOR 3))) || upper("right"((nombre)::text, 1)))) STORED,
    nombre character varying(100) COLLATE pg_catalog."default" NOT NULL,
    estado character varying(15) COLLATE pg_catalog."default" NOT NULL,
    cantidad integer NOT NULL,
    CONSTRAINT ingrediente_pkey PRIMARY KEY (id_ingrediente),
    CONSTRAINT ingrediente_estado_check CHECK (estado::text = ANY (ARRAY['Disponible'::character varying, 'Agotado'::character varying]::text[]))
)
```

```
CREATE TABLE notificaciones
(
    id_notificacion integer NOT NULL DEFAULT
nextval('notificaciones_id_notificacion_seq'::regclass),
    id_usuario integer NOT NULL,
    mensaje text COLLATE pg_catalog."default" NOT NULL,
    fecha timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT notificaciones_pkey PRIMARY KEY (id_notificacion),
    CONSTRAINT notificaciones_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES public.usuario (id_usuario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

CREATE TABLE pedido_producto
(
    id_pedido_producto integer NOT NULL DEFAULT
nextval('pedido_producto_id_pedido_producto_seq'::regclass),
    id_pedido integer NOT NULL,
    id_producto integer NOT NULL,
    cantidad integer NOT NULL,
    subtotal numeric(10,2) NOT NULL,
    CONSTRAINT pedido_producto_pkey PRIMARY KEY (id_pedido_producto),
    CONSTRAINT pedido_producto_id_pedido_fkey FOREIGN KEY (id_pedido)
        REFERENCES public.pedidos (id_pedido) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT pedido_producto_id_producto_fkey FOREIGN KEY (id_producto)
        REFERENCES public.producto (id_producto) MATCH SIMPLE
```

```
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)

CREATE TABLE pagos
(
  id_pago integer NOT NULL DEFAULT nextval('pagos_id_pago_seq'::regclass),
  id_pedido integer NOT NULL,
  metodo_pago character varying(20) COLLATE pg_catalog."default",
  estado_pago character varying(20) COLLATE pg_catalog."default",
  fecha_pago timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
  total_pagado numeric(10,2) NOT NULL,
  CONSTRAINT pagos_pkey PRIMARY KEY (id_pago),
  CONSTRAINT pagos_id_pedido_fkey FOREIGN KEY (id_pedido)
    REFERENCES public.pedidos (id_pedido) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
  CONSTRAINT pagos_metodo_pago_check CHECK (metodo_pago::text = ANY
(ARRAY['Tarjeta'::character varying, 'PayPal'::character varying, 'Efectivo'::character
varying]::text[])),
  CONSTRAINT pagos_estado_pago_check CHECK (estado_pago::text = ANY
(ARRAY['Pendiente'::character varying, 'Aprobado'::character varying, 'Rechazado'::character
varying]::text[]))
)
```