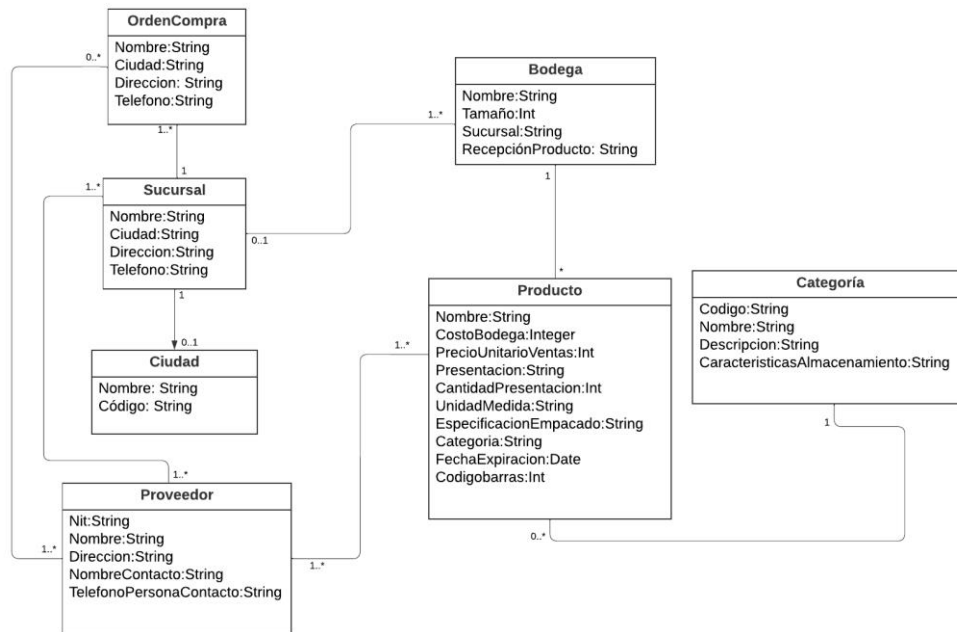
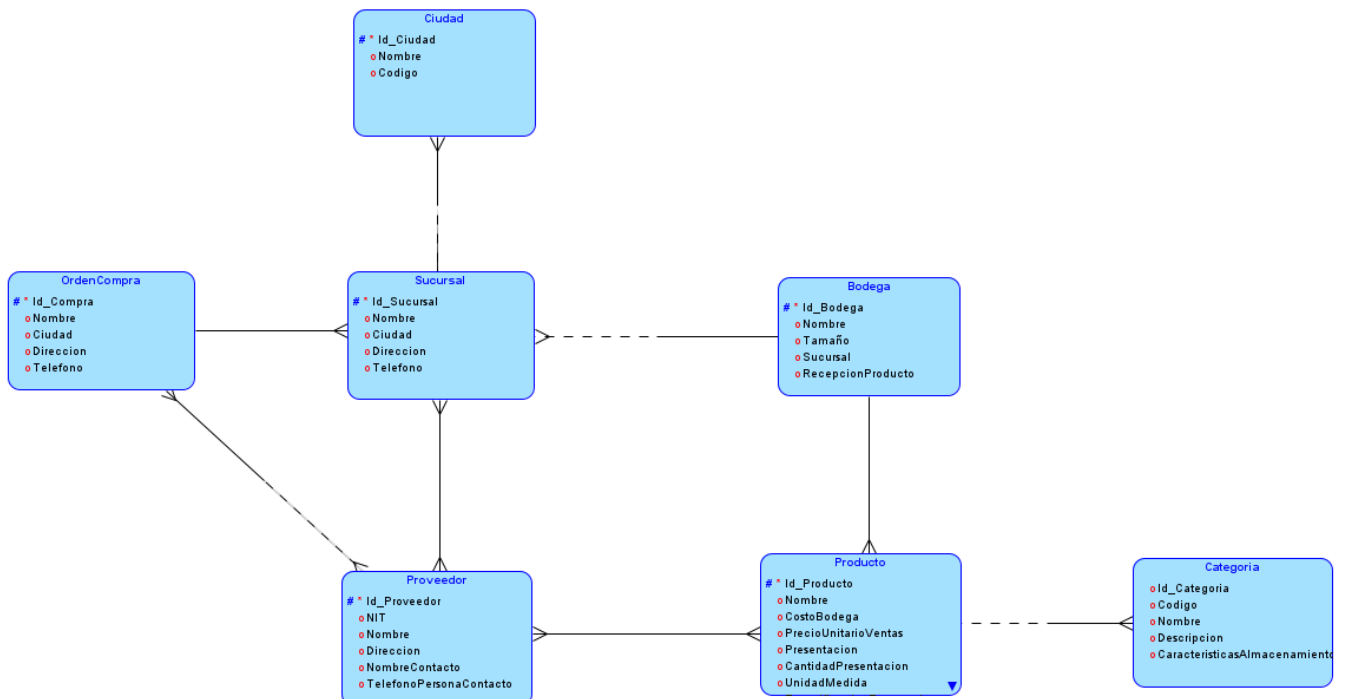


## ANÁLISIS Y MODELO CONCEPTUAL

### A) Modelo conceptual en UML



### B) Modelo conceptual en E/R



## DISEÑO DE LA BASE DE DATOS

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

ENTREGA 1 - DISEÑO  
SISTEMAS TRANSACCIONALES

A) Modelo de datos relacional en Excel

Sucursal					
id_sucursal	nombre	ciudad	dirección	teléfono	id_bodega
PK	NN	NN	NN	NN	FK, NN
S1	Sucursal Norte	Bogotá	Calle 123 #45-67	3112233445	B1
S2	Sucursal Sur	Bucaramanga	Carrera 7 #89-10	3203344556	B2
S3	Sucursal Oriente	Bogotá	Av. Caracas #56-78	3155566777	B4
S4	Sucursal Occidente	Bogotá	Calle 80 #12-34	3104455667	B3
S5	Sucursal Centro	Bucaramanga	Carrera 15 #34-56	3005566777	B5

Bodega		
id_bodega	nombre	tamaño_m2
PK	NN	NN
B1	Bodega Principal	500
B2	Bodega Secundaria	300
B3	Bodega Norte	200
B4	Bodega Sur	400
B5	Bodega Centro	250

Producto											
id_producto	nombre	costo_bodega	precio_unitario	presentación	cantidad_presentación	unidad_medida	especificación_empacado	categoría	fecha_expiración	código_barras	id_bodega
PK	NN	NN	NN	NN	NN	NN	NN	FK, NN	CK	ND, NN	FK
P1	Papas fritas Les frites	4200	9000	Paquetón de 5 paquetes	1000	gr	150 cm³, 15 gr	C1	12/12/2027	10101010	B2
P2	Shampoo Mi pelo rubio	5000	10000	Botella de 500 ml	500	ml	200 cm³, 20 gr	C2	1/01/2026	11111111	B1
P3	Jugo de naranja	3000	7000	Caja de 12 unidades	2400	ml	180 cm³, 25 gr	C3	10/10/2025	12121212	B5
P4	Leche deslactosada	4000	8000	Caja de 6 unidades	6000	ml	220 cm³, 30 gr	C1	15/08/2027	13131313	B4
P5	Aceite de oliva	6000	12000	Botella de 1 litro	1000	ml	250 cm³, 35 gr	C4	20/07/2026	14141414	B3

Categoría			
id_categoria	nombre	descripción	características_almacenamiento
PK	NN	NN	NN
C1	Perecedero	Productos con fecha límite	Refrigeración necesaria
C2	No perecedero	Productos sin caducidad	Almacenamiento en seco
C3	Aseo	Productos de limpieza	Evitar contacto con alimentos
C4	Congelado	Productos congelados	Almacenamiento a -18°C
C5	Bebidas	Bebidas alcohólicas y no alcohólicas	Almacenamiento en lugar fresco y seco

Proveedor					
id_proveedor	NIT	nombre	dirección	nombre_contacto	teléfono_contacto
PK	NN, ND	NN	NN	NN	NN
P1	900123456	Proveedora Andes S.A	Calle 1 #23-45, Bogotá	Juan Pérez	3101234567
P2	800654321	Alimentos del Norte	Carrera 2 #34-56, Bogotá	María García	3129876543
P3	900987654	Suministros y Cía.	Av. 3 #45-67, Bogotá	Luis Martínez	3137654321
P4	800123987	Distribuciones Sur	Calle 4 #56-78, Bogotá	Ana Rodríguez	3158765432
P5	900876543	Importadora Occidente	Carrera 5 #67-89, Bogotá	Carlos Gómez	3165432109

Orden de Compra				
id_orden_compra	fecha_emisión	estado	id_sucursal	id_proveedor
PK	NN	NN	FK, NN	FK, NN
O1	10/01/2024	vigente	S1	P2
O2	15/02/2024	entregada	S2	P3
O3	20/03/2024	anulada	S3	P4
O4	25/04/2024	vigente	S4	P5
O5	30/05/2024	entregada	S5	P1

202125304 -David Ortiz

202215891 - José Salgado

202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

Recepción de Productos		
id_recepcion	fecha_recepcion	id_bodega
PK	NN	FK, NN
R1	15/01/2024	B1
R2	20/02/2024	B2
R3	25/03/2024	B3
R4	30/04/2024	B4
R5	5/05/2024	B5

Ciudad		
id_ciudad	Nombre	Codigo
PK	NN	FK, NN
CU1	Bogotá	88001
CU2	Tunja	88002
CU3	Bucaramanga	88003
CU4	Pereira	88004
CU5	Cartagena	88005

Recepción de Productos - Producto (Recepción de productos puede incluir varios productos)	
id_recepcion	id_producto
FK	FK
R1	P1
R1	P2
R2	P3
R3	P4
R4	P5

### B) Normalización

#### 1. Sucursal

##### 1FN (Primera Forma Normal)

- Todos los atributos contienen valores atómicos (indivisibles). Cada fila es única gracias a la clave primaria id\_sucursal.

Cumple 1FN.

##### 2FN (Segunda Forma Normal)

- La clave primaria es id\_sucursal, y todos los demás atributos (nombre, ciudad, dirección, teléfono, id\_bodega) dependen completamente de esta clave.

Cumple 2FN.

3FN (Tercera Forma Normal)

- No hay dependencias transitivas, ya que todos los atributos dependen directamente de la clave primaria id\_sucursal. El id\_bodega es una clave foránea que no crea dependencias transitivas.  
Cumple 3FN.

## 2. Bodega

1FN

- Todos los atributos tienen valores indivisibles, y cada fila es única por la clave primaria id\_bodega.  
Cumple 1FN.  
2FN
- La clave primaria es id\_bodega, y los atributos nombre y tamaño\_m2 dependen completamente de esta clave.  
Cumple 2FN.  
3FN
- No existen dependencias transitivas, ya que todos los atributos dependen solo de id\_bodega.  
Cumple 3FN.

## 3. Producto

1FN

- Todos los campos contienen valores atómicos, como nombre, costo\_bodega, precio\_unitario, etc. La clave primaria es id\_producto, lo que asegura la unicidad de cada fila.  
Cumple 1FN.  
2FN
- La clave primaria es id\_producto, y todos los atributos dependen completamente de esta clave. Las FKs id\_bodega e id\_categoria solo indican relaciones y no afectan la dependencia de los atributos de la clave principal.  
Cumple 2FN.  
3FN
- No existen dependencias transitivas entre los atributos. Aunque id\_bodega y id\_categoria son claves foráneas, estas no crean dependencias no deseadas entre los atributos.  
Cumple 3FN.

## 4. Categoría

1FN

- Todos los atributos (nombre, descripción, características\_almacenamiento) son atómicos y cada fila es única gracias a la clave primaria id\_categoria.  
Cumple 1FN.  
2FN
- Todos los atributos dependen completamente de id\_categoria, que es la clave primaria.  
Cumple 2FN.  
3FN

- No hay dependencias transitivas. Cada atributo depende solo de id\_categoria.  
Cumple 3FN.

#### 5. Proveedor

1FN

- Todos los atributos (NIT, nombre, dirección, nombre\_contacto, teléfono\_contacto) son atómicos, y la clave primaria id\_proveedor asegura la unicidad de las filas.

Cumple 1FN.

2FN

- Cada atributo depende completamente de la clave primaria id\_proveedor.

Cumple 2FN.

3FN

- No hay dependencias transitivas en esta relación.

Cumple 3FN.

#### 6. Orden de Compra

1FN

- Todos los atributos (fecha\_emisión, estado, id\_sucursal, id\_proveedor) son indivisibles, y la clave primaria id\_orden\_compra asegura que cada fila es única.

Cumple 1FN.

2FN

- Cada atributo depende completamente de la clave primaria id\_orden\_compra. Las claves foráneas id\_sucursal e id\_proveedor solo establecen relaciones, no afectan la dependencia de los atributos.

Cumple 2FN.

3FN

- No hay dependencias transitivas, ya que todos los atributos dependen directamente de la clave primaria.

Cumple 3FN.

#### 7. Recepción de Productos

1FN

- Todos los atributos son atómicos, y la clave primaria id\_recepcion asegura que cada fila es única.

Cumple 1FN.

2FN

- Todos los atributos dependen completamente de la clave primaria id\_recepcion. La FK id\_bodega solo establece una relación y no afecta la dependencia.

Cumple 2FN.

3FN

- No hay dependencias transitivas en esta relación.

Cumple 3FN.

#### 8. Ciudad

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

1FN

- Todos los atributos son indivisibles, y la clave primaria id\_ciudad garantiza la unicidad de las filas.  
Cumple 1FN.

2FN

- Todos los atributos dependen completamente de id\_ciudad.  
Cumple 2FN.

3FN

- No hay dependencias transitivas en esta relación.  
Cumple 3FN.

### 9. Recepción de Productos - Producto

1FN

- Esta tabla tiene una clave compuesta por id\_recepcion y id\_producto. Ambos campos son atómicos y no hay valores repetidos o no atómicos.

Cumple 1FN.

2FN

- Los atributos dependen completamente de la clave compuesta (id\_recepcion e id\_producto).  
Cumple 2FN.

3FN

- No hay dependencias transitivas, ya que todos los atributos dependen de la clave compuesta.  
Cumple 3FN.

## JUSTIFICACION UML A RELACIONAL

### 1. Entidad: Sucursal

- **Atributos en UML:** Nombre, Ciudad, Dirección, Teléfono.
- **Relación en UML:** Está asociada con las entidades **Ciudad** y **Bodega**.

### Modelo Relacional:

En el modelo relacional, **Sucursal** se transforma en una tabla con los siguientes atributos:

- id\_sucursal (clave primaria)
- nombre
- ciudad (clave foránea que referencia la entidad **Ciudad**)
- direccion
- telefono

### Justificación:

- En UML, **Sucursal** tiene una relación 1 con **Ciudad**, lo que se traduce en una clave foránea (id\_ciudad) en el modelo relacional.
- También tiene una relación N:1 con **Bodega**, que se traduce en la clave foránea id\_bodega para vincular sucursales con las bodegas.

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

### 2. Entidad: Ciudad

- **Atributos en UML:** Nombre, Código.
- **Relación en UML:** Está relacionada con **Sucursal**.

#### Modelo Relacional:

La entidad **Ciudad** se transforma en una tabla:

- id\_ciudad (clave primaria)
- nombre
- codigo

#### Justificación:

- La clave primaria id\_ciudad es necesaria para identificar cada ciudad de manera única.
- **Sucursal** tiene una relación N:1 con **Ciudad**, por lo que **Sucursal** tendrá una clave foránea id\_ciudad para indicar la relación.

### 3. Entidad: Bodega

- **Atributos en UML:** Nombre, Tamaño, Sucursal, RecepciónProducto.
- **Relación en UML:** Relacionada con **Sucursal** y **Producto**.

#### Modelo Relacional:

**Bodega** se transforma en la tabla:

- id\_bodega (clave primaria)
- nombre
- tamaño
- id\_sucursal (clave foránea de la relación con **Sucursal**)
- recepcion\_producto

#### Justificación:

- La relación N:1 entre **Bodega** y **Sucursal** se traduce en una clave foránea id\_sucursal en la tabla **Bodega**.
- **Bodega** también tiene una relación con **Producto** a través de **Recepción de Productos**, que se maneja en la tabla relacional adecuada.

### 4. Entidad: Producto

- **Atributos en UML:** Nombre, CostoBodega, PrecioUnitario, Presentación, CantidadPresentación, UnidadMedida, EspecificaciónEmpacado, Categoría, FechaExpiración, Códigobarras.
- **Relación en UML:** Relacionada con **Bodega** y **Categoría**.

#### Modelo Relacional:

**Producto** se convierte en:

- id\_producto (clave primaria)
- nombre
- costo\_bodega

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

- precio\_unitario
- presentación
- cantidad\_presentación
- unidad\_medida
- especificacion\_empacado
- fecha\_expiracion
- codigo\_barras
- id\_categoria (clave foránea que referencia a **Categoría**)
- id\_bodega (clave foránea de la relación con **Bodega**)

### Justificación:

- En UML, **Producto** tiene una relación N:1 con **Bodega** y **Categoría**, lo que se refleja en las claves foráneas id\_bodega e id\_categoria en el modelo relacional.
- Se añaden todas las características propias del producto, tal como se definen en el UML.

### 5. Entidad: Categoría

- **Atributos en UML:** Código, Nombre, Descripción, CaracterísticasAlmacenamiento.
- **Relación en UML:** Relacionada con **Producto**.

### Modelo Relacional:

**Categoría** se transforma en la tabla:

- id\_categoria (clave primaria)
- codigo
- nombre
- descripcion
- caracteristicas\_almacenamiento

### Justificación:

- La relación N:1 entre **Producto** y **Categoría** se representa mediante la clave foránea id\_categoria en la tabla **Producto**.

### 6. Entidad: Proveedor

- **Atributos en UML:** NIT, Nombre, Dirección, NombreContacto, TeléfonoPersonaContacto.
- **Relación en UML:** Relacionada con **OrdenCompra**.

### Modelo Relacional:

**Proveedor** se convierte en la tabla:

- id\_proveedor (clave primaria)
- nit
- nombre
- direccion
- nombre\_contacto
- telefono\_contacto



**Justificación:**

- En UML, **Proveedor** tiene una relación N:1 con **Orden de Compra**, lo que se traducirá en una clave foránea id\_proveedor en la tabla **OrdenCompra**.

**7. Entidad: Orden de Compra**

- **Atributos en UML:** Nombre, Ciudad, Dirección, Teléfono.
- **Relación en UML:** Relacionada con **Sucursal** y **Proveedor**.

**Modelo Relacional:**

**Orden de Compra** se convierte en:

- id\_orden\_compra (clave primaria)
- nombre
- direccion
- telefono
- id\_sucursal (clave foránea que hace referencia a la tabla **Sucursal**)
- id\_proveedor (clave foránea que referencia a la tabla **Proveedor**)

**Justificación:**

- Las relaciones N:1 que tiene **OrdenCompra** con **Sucursal** y **Proveedor** en UML se reflejan en las claves foráneas id\_sucursal e id\_proveedor en el modelo relacional.

**ESCENARIOS DE PRUEBA**

**1. Restricción de Integridad de Unicidad (Primary Key, Unique)**

Escenario 1: Prueba de restricción de unicidad en el campo ID Cliente

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email)  
VALUES (1, 'Carlos', 'Pérez', 'carlos.perez@example.com');
- **Explicación:** Se inserta una tupla con ID\_Cliente igual a 1. Si se intenta insertar otra tupla con el mismo ID\_Cliente, la base de datos debe rechazarla para mantener la unicidad de la clave primaria.
- **Resultado esperado:** La primera inserción es exitosa. Un intento de inserción posterior con el mismo ID\_Cliente resultará en una violación de la restricción de unicidad.

Escenario 2: Prueba de restricción de unicidad en el campo Email

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email)  
VALUES (2, 'Ana', 'López', 'ana.lopez@example.com');
- **Explicación:** Se inserta una tupla con un Email único. Si se intenta insertar otra tupla con el mismo Email, la base de datos debe rechazarla para mantener la unicidad del correo electrónico.

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

- **Resultado esperado:** La primera inserción es exitosa. Un intento de inserción posterior con el mismo Email resultará en una violación de la restricción de unicidad.

### 2. Restricción de Integridad Referencial (Foreign Key)

Escenario 3: Prueba de integridad referencial en el campo ID\_Cliente en la tabla Pedido

- **Inserción:**  
INSERT INTO Pedido (ID\_Pedido, Fecha, ID\_Cliente)  
VALUES (101, '2024-09-01', 1);
- **Explicación:** Se inserta una tupla en la tabla Pedido con un ID\_Cliente que existe en la tabla Cliente. La integridad referencial asegura que el ID\_Cliente debe existir en la tabla Cliente.
- **Resultado esperado:** La inserción es exitosa ya que el ID\_Cliente 1 existe en la tabla Cliente.

Escenario 4: Prueba de violación de integridad referencial

- **Inserción:**  
INSERT INTO Pedido (ID\_Pedido, Fecha, ID\_Cliente)  
VALUES (102, '2024-09-02', 999);
- **Explicación:** Se intenta insertar una tupla en la tabla Pedido con un ID\_Cliente que no existe en la tabla Cliente. Esto debe resultar en una violación de la integridad referencial.
- **Resultado esperado:** La inserción falla porque el ID\_Cliente 999 no existe en la tabla Cliente.

### 3. Restricción de Integridad de Dominio

Escenario 5: Prueba de integridad de dominio en el campo Email

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email)  
VALUES (3, 'Luis', 'Martínez', 'luis.martinez@example.com');
- **Explicación:** Se inserta un correo electrónico en el formato correcto, cumpliendo con el dominio esperado.
- **Resultado esperado:** La inserción es exitosa.

Escenario 6: Prueba de violación de integridad de dominio

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email)  
VALUES (4, 'Sara', 'Gómez', 'sara.gomez#example.com');
- **Explicación:** Se intenta insertar un correo electrónico con un formato incorrecto. Esto debe resultar en una violación de la integridad de dominio.

- **Resultado esperado:** La inserción falla debido a que el formato del correo electrónico no cumple con las reglas de integridad de dominio.

#### **4. Restricción de Integridad de Entidad (NOT NULL)**

##### Escenario 7: Prueba de integridad de entidad en el campo Nombre

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email)  
VALUES (5, NULL, 'Rojas', 'roberto.rojas@example.com');
- **Explicación:** Se intenta insertar una tupla donde el campo Nombre es NULL. Si el campo Nombre está definido como NOT NULL, esta inserción debe fallar.
- **Resultado esperado:** La inserción falla debido a que el campo Nombre no permite valores nulos.

##### Escenario 8: Prueba de integridad de entidad en el campo ID\_Cliente

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email)  
VALUES (NULL, 'María', 'Castro', 'maria.castro@example.com');
- **Explicación:** Se intenta insertar una tupla donde el campo ID\_Cliente es NULL. Dado que ID\_Cliente es una clave primaria, debe cumplir con la restricción NOT NULL.
- **Resultado esperado:** La inserción falla debido a que ID\_Cliente no permite valores nulos.

#### **6. Prueba de Integridad de Unicidad (Unique) en la tabla Producto**

##### Escenario 9: Prueba de restricción de unicidad en el campo Codigo\_Producto

- **Inserción:**  
INSERT INTO Producto (Codigo\_Producto, Nombre\_Producto, Precio)  
VALUES ('PROD001', 'Camiseta', 25000);
- **Explicación:** Se inserta un producto con un Codigo\_Producto único. La base de datos debe asegurar que este código no se repita en ninguna otra fila.
- **Resultado esperado:** La primera inserción es exitosa. Si se intenta insertar un nuevo producto con el mismo Codigo\_Producto, debe fallar.

##### Escenario 10: Prueba de violación de unicidad en el campo Codigo\_Producto

- **Inserción:**  
INSERT INTO Producto (Codigo\_Producto, Nombre\_Producto, Precio)  
VALUES ('PROD001', 'Gorra', 15000);

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

- **Explicación:** Se intenta insertar un nuevo producto con el mismo Codigo\_Producto ya existente. Esto debe resultar en una violación de la restricción de unicidad.
- **Resultado esperado:** La inserción falla debido a la duplicación del Codigo\_Producto.

### 7. Prueba de Integridad Referencial en la tabla Detalle\_Pedido

#### Escenario 11: Prueba de integridad referencial en el campo ID\_Pedido y Codigo\_Producto

- **Inserción:**  
INSERT INTO Detalle\_Pedido (ID\_Pedido, Codigo\_Producto, Cantidad)  
VALUES (101, 'PROD001', 2);
- **Explicación:** Se inserta una tupla en la tabla Detalle\_Pedido con un ID\_Pedido y un Codigo\_Producto que existen en las tablas Pedido y Producto, respectivamente.
- **Resultado esperado:** La inserción es exitosa ya que ambas claves foráneas existen en sus respectivas tablas.

#### Escenario 12: Prueba de violación de integridad referencial en el campo Codigo\_Producto

- **Inserción:**  
INSERT INTO Detalle\_Pedido (ID\_Pedido, Codigo\_Producto, Cantidad)  
VALUES (101, 'PROD999', 3);
- **Explicación:** Se intenta insertar una tupla con un Codigo\_Producto que no existe en la tabla Producto. Esto debe resultar en una violación de la integridad referencial.
- **Resultado esperado:** La inserción falla porque el Codigo\_Producto no existe en la tabla Producto.

### 8. Prueba de Integridad de Dominio en la tabla Producto

#### Escenario 13: Prueba de integridad de dominio en el campo Precio

- **Inserción:**  
INSERT INTO Producto (Codigo\_Producto, Nombre\_Producto, Precio)  
VALUES ('PROD002', 'Pantalones', 45000);
- **Explicación:** Se inserta un precio que es un número positivo, cumpliendo con las reglas de integridad de dominio para el campo Precio.
- **Resultado esperado:** La inserción es exitosa.

#### Escenario 14: Prueba de violación de integridad de dominio en el campo Precio

- **Inserción:**  
INSERT INTO Producto (Codigo\_Producto, Nombre\_Producto, Precio)  
VALUES ('PROD003', 'Zapatos', -30000);

202125304 -David Ortiz  
202215891 - José Salgado  
202215816 - Angelica Yeraldin Rodríguez

## ENTREGA 1 - DISEÑO SISTEMAS TRANSACCIONALES

- **Explicación:** Se intenta insertar un precio negativo, lo cual debe violar la integridad de dominio.
- **Resultado esperado:** La inserción falla porque el Precio no puede ser negativo.

### 9. Prueba de Integridad de Entidad (NOT NULL) en la tabla Detalle\_Pedido

*Escenario 15: Prueba de integridad de entidad en el campo Cantidad*

- **Inserción:**  
INSERT INTO Detalle\_Pedido (ID\_Pedido, Codigo\_Producto, Cantidad)  
VALUES (101, 'PROD001', NULL);
- **Explicación:** Se intenta insertar una tupla donde el campo Cantidad es NULL. Si el campo Cantidad está definido como NOT NULL, esta inserción debe fallar.
- **Resultado esperado:** La inserción falla porque el campo Cantidad no permite valores nulos.

### 10. Prueba de Integridad Referencial en la tabla Cliente

*Escenario 16: Prueba de integridad referencial en el campo ID\_Ciudad (Claves Foráneas)*

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email, ID\_Ciudad)  
VALUES (6, 'Jorge', 'Hernández', 'jorge.hernandez@example.com', 1);
- **Explicación:** Se inserta un cliente que tiene una referencia válida a una ciudad existente en la tabla Ciudad.
- **Resultado esperado:** La inserción es exitosa ya que el ID\_Ciudad 1 existe en la tabla Ciudad.

*Escenario 17: Prueba de violación de integridad referencial en el campo ID\_Ciudad*

- **Inserción:**  
INSERT INTO Cliente (ID\_Cliente, Nombre, Apellido, Email, ID\_Ciudad)  
VALUES (7, 'Lucía', 'Ramírez', 'lucia.ramirez@example.com', 999);
- **Explicación:** Se intenta insertar un cliente con un ID\_Ciudad que no existe en la tabla Ciudad. Esto debe resultar en una violación de la integridad referencial.
- **Resultado esperado:** La inserción falla porque el ID\_Ciudad 999 no existe en la tabla Ciudad.