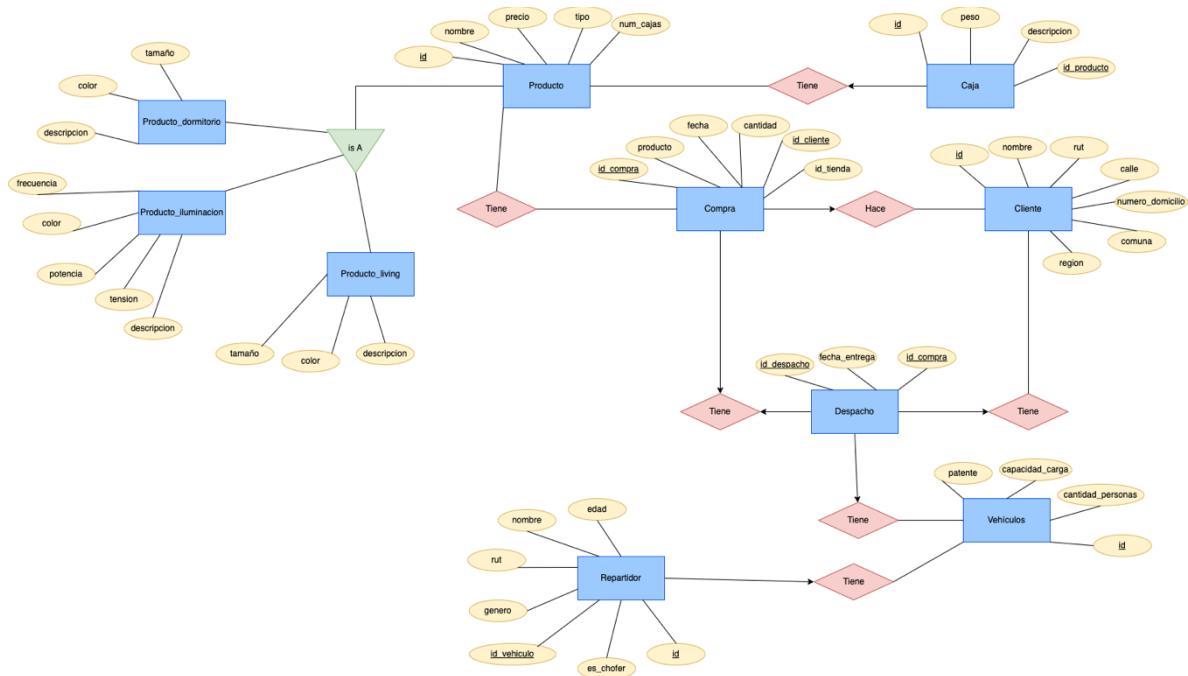


# Entrega 2 IIC2413

Diego Mir Bezanilla  
Matías Lecaros Torres

## Diagrama E/R:



## Esquema relacional:

Pondremos los datatypes entre corchetes [] y las llaves primarias o foráneas entre paréntesis curvos (), esto lo haremos para las tablas que nos corresponden como grupo de número par:

### Tabla de productos:

- id (PK) [INT]
- nombre [VARCHAR]
- precio [INT]
- numero\_cajas [INT]

- tipo [VARCHAR]

#### **Tabla de productos\_dormitorio:**

- id\_producto (PK, FK → Tabla de productos) [INT]
- tamaño [VARCHAR]
- color [VARCHAR]
- descripcion [VARCHAR]

#### **Tabla de productos\_living:**

- id\_producto (PK, FK → Tabla de productos) [INT]
- dimensiones [VARCHAR]
- material [VARCHAR]
- carga [DECIMAL]

#### **Tabla de productos\_iluminacion:**

- id\_producto (PK, FK → Tabla de productos) [INT]
- descripcion [VARCHAR]
- color [VARCHAR]
- frecuencia [DECIMAL]
- potencia [DECIMAL]
- tensión [DECIMAL]

#### **Tabla de cajas:**

- id (PK) [INT]
- id\_producto (FK → Tabla de productos) [INT]
- peso [DECIMAL]
- descripción [VARCHAR]

#### **Tabla de clientes:**

- id (PK) [INT]
- nombre [VARCHAR]
- rut [VARCHAR]
- calle [VARCHAR]
- numero\_domicilio [INT]
- comuna [VARCHAR]
- región [VARCHAR]

#### **Tabla de compras:**

- id\_compra (PK) [INT]
- fecha [DATE]
- id\_producto (FK → Tabla de productos) [INT]
- id\_cliente (FK → Tabla de clientes) [INT]
- id\_tienda (FK → Tabla de tiendas) [INT]
- cantidad [INT]

#### **Tabla de despachos:**

- id\_despacho (PK) [INT]
- id\_compra (FK → Tabla de compras) [INT]
- fecha\_entrega [DATE]
- id\_repartidor (FK → Tabla de repartidores)<sup>1</sup>

#### **Tabla de vehiculos:**

- id (PK) [INT]
- patente [VARCHAR]
- capacidad\_carga [INT]
- cantidad\_personas [INT]

#### **Tabla de repartidores:**

- id (PK) [INT]

- nombre [VARCHAR]
- rut [VARCHAR]
- edad [INT]
- genero [VARCHAR]
- id\_vehiculo (FK → Tabla de vehiculos) [INT]
- es\_chofer [BOOLEAN]<sup>2</sup>

Es necesario mencionar que, de la forma en que venía la tabla productos tenía dependencias parciales, ya que existen atributos específicos para cada uno de los productos de tipo living, iluminación o dormitorio. Esto nos indicó que el modelo debía ser ajustado para estar en BCNF, para esto creamos una tabla principal de productos con los atributos que los productos tienen en común y luego creamos subtablas para cada tipo de producto con los atributos específicos, quedando como fue presentada anteriormente.

1. Agregamos el atributo id\_repartidor a los despachos para asignar a un repartidor a los despachos, esto dado el error cometido en las tablas que nos fueron entregadas, en donde repartidores y despachos no presentan una relación directa. Esto no fue puesto en nuestro diagrama de E/R pero está incluido en nuestras tablas.
2. Puede ser considerado como VARCHAR por el hecho de ser escrito el YES o el NO, pero como se asocia a un método un tanto binario lo dejamos como booleano en donde YES es el True y NO es False.

### **Justificación modelo:**

Para verificar si es que el modelo realizado anteriormente se encuentra en BCNF o en 3NF, necesitamos analizar las distintas dependencias funcionales. Al analizar el modelo nos damos cuenta de que las dependencias son las siguientes:

- 1) Tabla de productos:
  - id → nombre, precio, numero\_cajas, tipo
- 2) Tabla de productos\_dormitorio:
  - id\_producto → tamaño, color, descripcion

3) Tabla de productos\_living:

- id\_producto → dimensiones, material, carga

4) Tabla de productos\_iluminacion:

- id\_producto → descripcion, color, frecuencia, potencia, tension

5) Tabla cajas:

- id → id\_producto, peso, descripción

6) Tabla clientes:

- id → nombre, rut, calle, numero\_domicilio, comuna, region

7) Tabla compras:

- id\_compra → fecha, producto, id\_cliente, id\_tienda, cantidad

8) Tabla despachos:

- id\_despacho → id\_compra, fecha\_entrega, id\_repartidor

9) Tabla vehículos:

- id → patente, capacidad\_carga, cantidad\_personas

10) Tabla repartidores:

- id → nombre, rut, edad, genero, id\_vehiculo, es\_chofer

Luego de haber determinado cada una de las dependencias funcionales de nuestro esquema, podemos concluir que el modelo se encuentra en BCNF, esto porque todas las dependencias funcionales son determinantes de clave (PK) o derivan de manera directa de ellas. Como se puede ver en las tablas, cada una tiene una clave primaria definida de manera correcta, no existen dependencias funcionales transitivas ni parciales, ya que las tablas no presentan atributos dependientes de otros atributos que no son clave.

Con esto damos por sentado que el modelo está correctamente normalizado y cumple con las reglas de integridad de la teoría de bases de datos relacionales, encontrándose en la forma BCNF.

### **Consultas SQL:**

A continuación se incluyen todas las consultas en SQL que fueron implementadas en nuestra aplicación:

#### **Consulta 1**

```
SELECT c.nombre, c.calle, c.numero_domicilio, c.comuna, c.region
FROM despachos d
JOIN compras co ON d.id_compra = co.id_compra
JOIN clientes c ON co.id_cliente = c.id
WHERE d.fecha_entrega = input;
```

En donde input representa a la fecha ingresada por el usuario.

#### **Consulta 2**

```
SELECT cl.nombre, cl.calle, cl.numero_domicilio, cl.comuna,
cl.region, p.precio * co.cantidad AS valor_compra
FROM compras co
JOIN clientes cl ON co.id_cliente = cl.id
JOIN productos p ON co.id_producto = p.id
WHERE co.id_compra = input;
```

En donde input representa el ID de la compra ingresado por el usuario.

#### **Consulta 3**

```
SELECT SUM(p.num_cajas * co.cantidad) AS total_cajas
FROM compras co
JOIN productos p ON co.id_producto= p.id
WHERE co.id_compra = input;
```

En donde input representa el ID de la compra ingresado por el usuario.

#### **Consulta 4**

```
SELECT SUM(c.peso) AS peso_total
FROM despachos d
JOIN compras co ON d.id_compra = co.id_compra
JOIN cajas c ON co.id = c.id_producto
JOIN vehiculos v ON d.id_vehiculo = v.id
WHERE d.fecha_entrega = input_fecha
      AND v.patente = input_patente;
```

En donde input\_fecha representa a la fecha de entrega e input\_patente la patente ingresadas por el usuario.

#### **Consulta 5**

```
SELECT cl.region, r.genero, COUNT(r.id) AS total_repartidores,
AVG(r.edad) AS edad_promedio
FROM repartidores r
JOIN vehiculos v ON r.id_vehiculo = v.id
JOIN despachos d ON r.id = d.id_repartidor
JOIN compras co ON d.id_compra = co.id_compra
JOIN clientes cl on co.id_cliente = cl.id
WHERE cl.region = input
GROUP BY r.genero;
```

En donde input representa la región ingresada por el usuario.

#### **Consulta 6**

```
SELECT co.id_compra, COUNT(co.id_producto) AS
numero_productos, SUM(p.num_cajas) AS numero_cajas
FROM compras co
JOIN productos p ON co.id_producto = p.id
WHERE co.id_cliente = input
GROUP BY c.id_compra;
```

En donde input representa el ID del cliente ingresado por el usuario.

### **Consulta 7**

```
SELECT co.id_cliente, cl.nombre, cl.rut, cl.calle,  
cl.numero_domicilio, cl.comuna, cl.region, SUM(p.precio * co.cantidad)  
AS total_gastado  
FROM compras co  
JOIN productos p ON co.id_producto = p.id  
JOIN clientes cl ON co.id_cliente = cl.id  
WHERE cl.region = input  
GROUP BY c.id_cliente, cl.nombre, cl.rut, cl.calle,  
cl.numero_domicilio, cl.comuna, cl.region  
ORDER BY total_gastado DESC  
LIMIT 5;
```

En donde representa la región ingresada por el usuario.