

**Antonio Orozco Varela**

Base de Datos  
1 DAW Mañana  
IES Alixar

# Tienda de Móviles

20 de Marzo del 2025



<b>1. Título y Descripción</b>	<b>3</b>
<b>2. Diagrama MER</b>	<b>3</b>
<b>3. Modelo Relacional a Tablas</b>	<b>3</b>
<b>4. Carga Masiva</b>	<b>3</b>
<b>5. Consultas Multitable</b>	<b>3</b>
<b>6. Vistas</b>	<b>3</b>
<b>7. Funciones y Procedimientos</b>	<b>3</b>
<b>8. Triggers</b>	<b>4</b>
<b>9. Valoración Personal</b>	<b>4</b>

## 1. Título y Descripción

### **Tienda de Móviles**

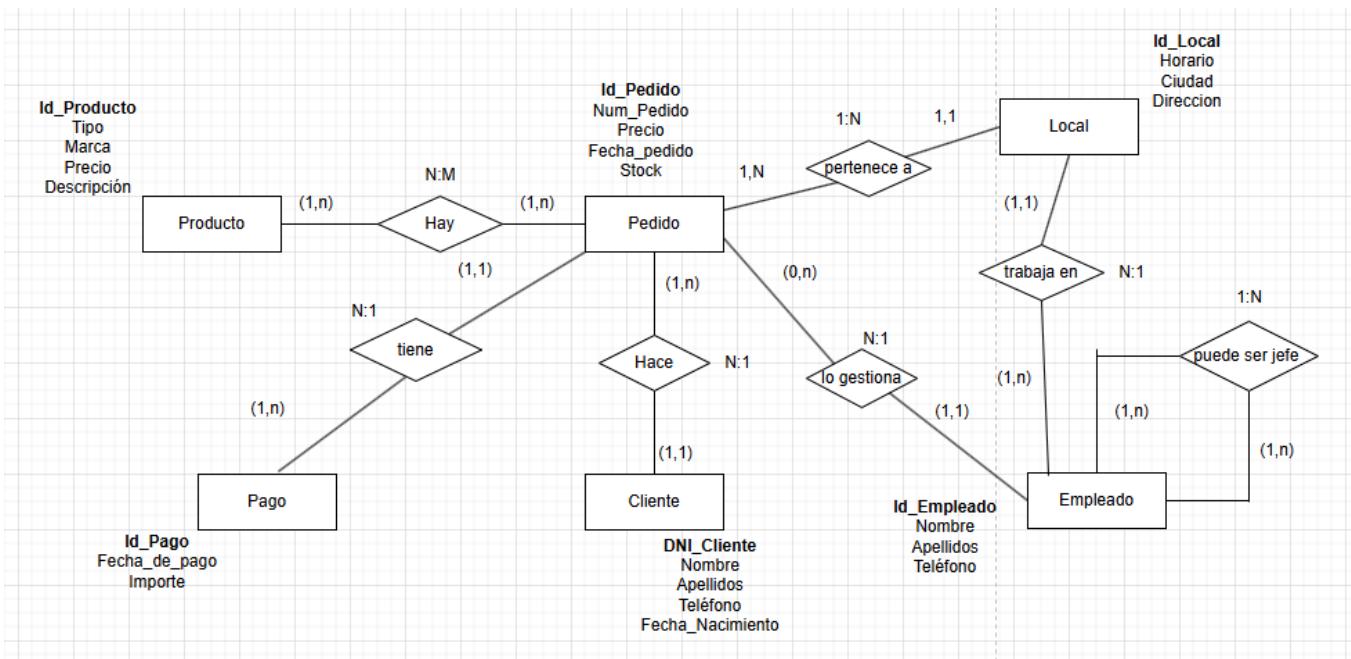
Este proyecto trata sobre la base de datos de una tienda de móviles que vende un poco de todo, desde móviles hasta fundas, cargadores y otros accesorios de todas las marcas y precios. En esta base de datos tenemos varias entidades como pueden ser los productos, los pedidos, los clientes, los empleados, los locales que hay, etc...



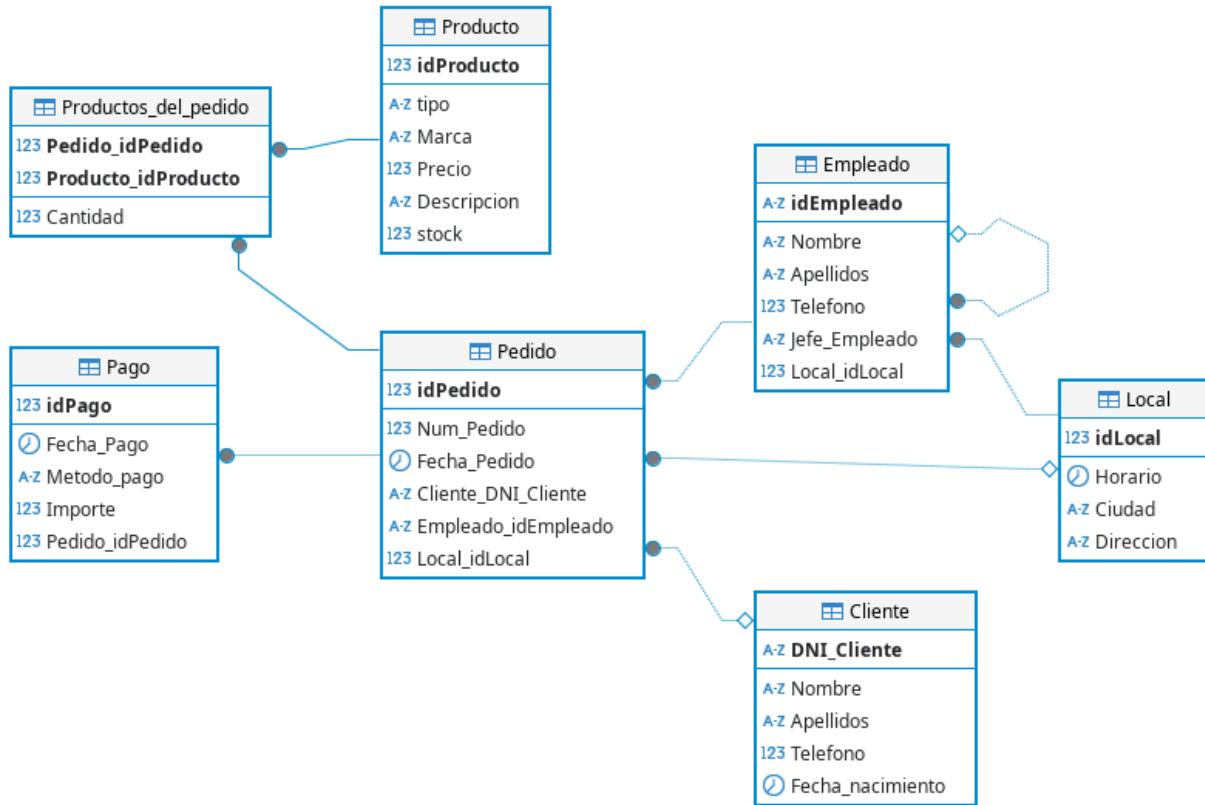
## 2. Diagrama MER

Diagrama Entidad Relación:

- Un producto puede estar en varios pedidos y en un pedido pueden haber varios productos por lo que se forma una relación de muchos a muchos
- Un pedido puede hacerse con uno o varios pagos pero el pago se puede hacer de un solo pedido por lo que es una relación de uno a muchos
- Un cliente puede hacer uno o varios pedidos pero un pedido sólo puede hacerse por un cliente por lo que es una relación de uno a muchos
- Un empleado puede gestionar cero o muchos pedidos y un pedido sólo puede ser gestionado por un empleado
- Un empleado puede ser empleado o jefe. Un jefe puede tener uno o varios empleados y un empleado puede tener uno o varios jefes
- Un empleado puede trabajar solo en un mismo local y en un local pueden trabajar varios empleados
- Un pedido puede pertenecer a un solo local y un local puede tener varios pedidos



### 3. Modelo Relacional a Tablas



## 4. Carga Masiva

Para la carga masiva de datos esta vez he usado Chat GPT ya que Mockaroo me dio muchos problemas la ultima vez y he generado los archivos con ChatGPT directamente ya que es más efectivo a la hora de llenar los campos

Pedido.csv • LibreOffice Calc

Archivo Editar Ver Insertar Formato Estilos Hoja Datos Herramientas Ventana Ayuda

Liberation Sans 10 pt N K S A fx Σ = idPedido

A	B	C	D	E	F	G
1	idPedido	Num_Pedido	Fecha_Pedido	Cliente_DNI_Cliente	Empleado_idEmpleado	
2	1	86116	2024-05-18 19:11:32.086506	17751149D	E0193	
3	2	94519	2024-04-12 19:11:32.086524	06397317N	E0939	
4	3	87130	2024-06-14 19:11:32.086528	52976867E	E0006	
5	4	92472	2024-07-30 19:11:32.086532	32478331J	E0247	
6	5	33549	2024-08-15 19:11:32.086536	83010402Z	E0720	
7	6	45470	2024-12-19 19:11:32.086539	15702072D	E0264	
8	7	94159	2024-10-09 19:11:32.086543	53998155I	E0350	
9	8	35108	2024-07-08 19:11:32.086548	19083829J	E0047	
10	9	68197	2025-03-25 19:11:32.086553	34969387Q	E0938	
11	10	84753	2024-12-27 19:11:32.086557	40489594K	E0184	
12	11	85785	2024-12-28 19:11:32.086561	50362073Q	E0186	
13	12	11893	2025-02-24 19:11:32.086564	06259269C	E0141	
14	13	60975	2024-12-18 19:11:32.086567	54928459J	E0392	
15	14	48714	2024-11-01 19:11:32.086570	37367180M	E0249	
16	15	23555	2024-09-24 19:11:32.086574	67768092I	E0707	
17	16	57065	2024-05-03 19:11:32.086577	06693582F	E0670	
18	17	91718	2025-02-11 19:11:32.086581	49254267U	E0853	
19	18	53824	2025-02-16 19:11:32.086584	47282499J	E0420	
20	19	86766	2024-08-04 19:11:32.086587	90082701W	E0489	
21	20	42926	2024-09-14 19:11:32.086591	82851680N	E0267	
22	21	37669	2025-01-05 19:11:32.086594	80023861V	E0494	
23	22	72551	2024-05-16 19:11:32.086597	90678209S	E0413	
24	23	96237	2024-12-24 19:11:32.086601	26866571T	E0476	
25	24	35119	2024-07-31 19:11:32.086604	30124321W	E0310	
26	25	80627	2024-04-29 19:11:32.086608	50803922B	E0730	
27	26	62513	2024-12-27 19:11:32.086612	09420718A	E0301	
28	27	86372	2025-01-02 19:11:32.086615	09690623X	E0104	
29	28	97181	2024-09-13 19:11:32.086618	91456736Q	E0045	
30	29	78814	2024-06-17 19:11:32.086621	56020654S	F0800	

## 5. Consultas Multitablea

### Consulta 1

Ciientes que han gastado más que la media general de clientes

The screenshot shows a SQL editor interface with a tree view on the left and a main workspace on the right. The workspace contains a query window with the following code:

```
select c.Nombre,sum(p2.Importe)
from Cliente c
join Pedido p
on c.DNI_Cliente =p.Cliente_DNI_Cliente
join Pago p2
on p.idPedido =p2.Pedido_idPedido
group by c.Nombre
having sum(p2.Importe) >
(select avg(total_cliente)
from(
select sum(p2.Importe)as total_cliente
from Cliente c
join Pedido p
on c.DNI_Cliente =p.Cliente_DNI_Cliente
join Pago p2
on p.idPedido =p2.Pedido_idPedido
group by c.Nombre )as sub);
```

Below the query window is a results grid titled "Cliente 1". It displays one row of data:

	A-Z Nombre	123 sum(p2.Importe)
1	Maria	258.120

The entire results grid is highlighted with a red rectangle.

## Consulta 2

### El producto más vendido por cada local

```
⑤ SELECT l.Ciudad ,p2.Descripcion,sum(pdp.cantidad) as cantidad_vendida
from Local l
join Pedido p
on l.idLocal =p.Local_idLocal
join Productos_del_pedido pdp
on p.idPedido =pdp.Pedido_idPedido
join Producto p2
on pdp.Producto_idProducto =p2.idProducto
group by idLocal,p2.idProducto
order BY l.Ciudad, cantidad_vendida DESC;
```

SQL | Ciudad.p2.Descripcion.sum(pdp.cantidad) as cantidad\_vendida Enter a SQL expression to filter results (use Ctrl+Space)

A-Z Ciudad	A-Z Descripcion	123 cantidad_vendida
Barcelona	Motorola Smartwatch modelo 183 con características avanzadas	5
Barcelona	Apple Smartwatch modelo 527 con características avanzadas	5
Barcelona	Huawei Accesorio modelo 813 con características avanzadas	5
Barcelona	Motorola Smartwatch modelo 443 con características avanzadas	5
Barcelona	OnePlus Smartwatch modelo 506 con características avanzadas	5
Barcelona	Motorola Accesorio modelo 597 con características avanzadas	5
Barcelona	Xiaomi Accesorio modelo 423 con características avanzadas	5
Barcelona	Motorola Accesorio modelo 339 con características avanzadas	5

## Consulta 3

Mostrar por cada cliente el total gastado, el número de pedidos y el promedio gastado por pedido, pero solo si ha hecho al menos 2 pedidos

```
④ select c.Nombre,sum(p2.Importe )as total_gastado,COUNT(p.idPedido ) as total_pedidos,
       avg(p2.Importe ) as promedio_por_pedido
  from Cliente c
  JOIN Pedido p
    on c.DNI_Cliente =p.Cliente_DNI_Cliente
   join Pago p2
    on p.idPedido =p2.Pedido_idPedido
   group by c.Nombre
   having count(p.idPedido) >= 2 ;
```

Consulta 1 X

select c.Nombre,sum(p2.Importe )as total\_gastado,COI | Enter a SQL expression to filter results (use Ctrl+Space)

c	A-Z Nombre	123 total_gastado	123 total_pedidos	123 promedio_por_pedido
	Ana	99.046	88	1.125,5227
	Miguel	106.972	103	1.038,5631
	Juan	94.898	93	1.020,4086
	Lucía	116.073	109	1.064,8899
	Javier	105.875	102	1.037,9902
	Pedro	107.388	97	1.107,0928
	María	258.120	106	2.435,0943
	Sofia	109.340	113	967,6106
	Carlos	104.812	99	1.058,7071
	Laura	88.135	90	979,2778

## Consulta 4

Mostrar los empleados que han gestionado pedidos con un importe total superior al importe medio de todos los empleados

```

select e.Nombre,sum(p2.Importe) as total_facturado,
       count(p.idPedido) as total_pedidos
  from Empleado e
  join Pedido p
    on e.idEmpleado = p.Empleado_idEmpleado
  join Pago p2
    on p.idPedido = p2.Pedido_idPedido
 group by e.idEmpleado
 having sum(p2.Importe) > (
      select avg(sub.total_empleado)
        from (
          select sum(p4.Importe) as total_empleado
            from Empleado e2
            join Pedido p3
              on e2.idEmpleado = p3.Empleado_idEmpleado
            join Pago p4
              on p3.idPedido = p4.Pedido_idPedido
            group by e2.idEmpleado)as sub);

```

Empleado 1 X

select e.Nombre,sum(p2.Importe) as total\_facturado, c |  Enter a SQL expression to filter results

	A-Z Nombre	123 total_facturado	123 total_pedidos
1	Pedro	3.107	4
2	Elena	5.428	4
3	Manuel	2.054	2
4	Manuel	3.924	4
5	Manuel	2.303	2
6	Carlos	3.825	3
7	Marta	3.030	2
8	Laura	4.454	3
9	Laura	2.352	2
10	Laura	2.153	2
11	Elena	7.178	5

## Consulta 5

Mostrar los productos cuya cantidad total vendida supera el promedio de cantidad vendida entre todos los productos

The screenshot shows the MySQL Workbench environment. On the left is a toolbar with various icons for database management. The main area has a query editor window containing the following SQL code:

```
④ select p.idProducto,sum(pdp.Cantidad)as total
  from Producto p
  join Productos_del_pedido pdp
    ON p.idProducto =pdp.Producto_idProducto
   group by p.idProducto
   having total > (
    SELECT avg(sub.total_cantidad)
      from (
        select sum(pdp2.Cantidad) as total_cantidad
          from Producto p2
        join Productos_del_pedido pdp2
          ON p2.idProducto =pdp2.Producto_idProducto
         group by p2.idProducto
       ) as sub);
```

Below the query editor is a results grid titled "Producto 1 X". The grid displays two columns: "idProducto" and "total". A red box highlights the first row of the grid, which contains the values 5 and 8 respectively.

Grilla	123 idProducto	123 total
1	5	8
2	13	7
3	15	8
4	16	11
5	18	6
6	22	7
7	25	5
8	27	6
9	32	9
10	35	8
11	37	14
12	39	5

## 6. Vistas

Vista 1

vista que muestre solo los clientes que han hecho al menos 3 pedidos, han gastado más de 500 € en total

```
create view vista_clientes_frecuentes as
select c.DNI_Cliente,c.Nombre,count(p.idPedido)as total_pedidos,
sum(pdp.cantidad*p.idPedido)as total_gastado
from Cliente c
join Pedido p
on c.DNI_Cliente=p.Cliente_DNI_Cliente
join Productos_del_pedido pdp
on p.idPedido=pdp.Pedido_idPedido
group by c.DNI_Cliente
having total_pedidos >= 3
AND total_gastado > 500;
```

A-Z DNI_Cliente	A-Z Nombre	123 total_pedidos	123 total_gastado
00205789R	Miguel	4	6.086
00406951U	Lucía	7	6.817
01757263H	Lucía	5	11.956
02734780Q	Carlos	3	3.864
05083852K	Sofía	4	8.127
05260094X	Javier	4	7.404
05496452V	Juan	5	6.588
06259269C	Ana	3	1.138
06693582F	Laura	7	5.985
07218142V	Pedro	5	3.986
07393740Y	Juan	3	2.670
08406819C	Juan	3	6.075
08443752G	Maria	3	5.819

## Vista 2

**Productos vendidos con su total de ingresos generados y el total de pedidos distintos en los que ha estado**

```
create view vista_productos_ingresos as
SELECT p.idProducto,sum(pdp.Cantidad)as total_productos_vendidos,
sum(p.idProducto*pdp.Cantidad)as total_facturado,
count(DISTINCT pdp.Pedido_idPedido)as productos_en_pedidos_dif
from Producto p
join Productos_del_pedido pdp
on p.idProducto =pdp.Producto_idProducto
group by p.idProducto;
```

Product 1 X

SELECT p.idProducto,sum(pdp.Cantidad)as total\_produ |  Enter a SQL expression to filter results (use Ctrl+Space)

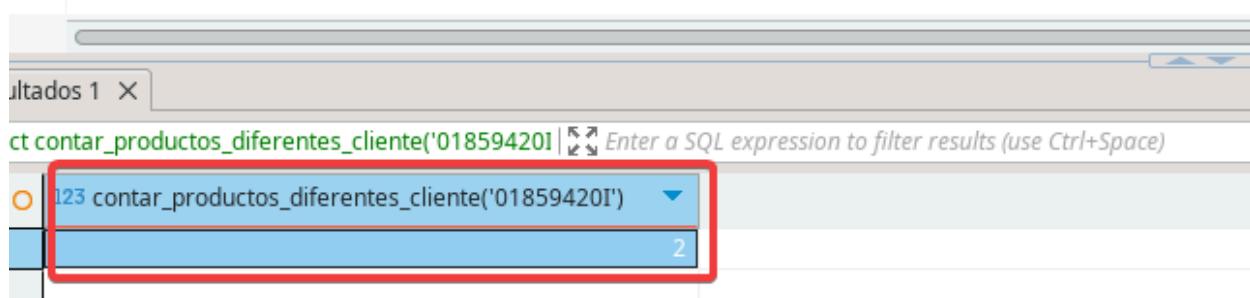
123 idProducto	123 total_productos_vendidos	123 total_facturado	123 productos_en_pedidos_dif
1	1	1	1
4	4	16	1
5	8	40	1
6	4	24	1
8	3	24	1
9	1	9	1
10	2	20	1
11	3	33	1
13	7	91	2
14	2	28	1
15	8	120	2
16	11	176	3
18	6	108	2
..	..	..	..

## 7. Funciones y Procedimientos

### Función 1

función que reciba el DNI de un cliente y devuelva cuántos productos diferentes ha comprado en total

```
DELIMITER $$  
CREATE FUNCTION contar_productos_diferentes_cliente(dni_Cliente VARCHAR(45))  
RETURNS INT  
DETERMINISTIC  
BEGIN  
    DECLARE salida INT DEFAULT 0;  
    select count(DISTINCT p2.idProducto ) into salida  
    from Cliente c  
    join Pedido p  
    on c.DNI_Cliente =p.Cliente_DNI_Cliente  
    join Productos_del_pedido pdp  
    on p.idPedido =pdp.Pedido_idPedido  
    join Producto p2  
    on pdp.Producto_idProducto =p2.idProducto  
    where c.DNI_Cliente =dni_Cliente;  
  
    return salida;  
  
END $$  
DELIMITER ;  
  
select contar_productos_diferentes_cliente('01859420I');
```



The screenshot shows the MySQL Workbench interface with the following details:

- The title bar says "Ultados 1 X".
- The SQL tab contains the function definition and a call to it: "select contar\_productos\_diferentes\_cliente('01859420I');".
- The results tab displays the output of the function call: "123 contar\_productos\_diferentes\_cliente('01859420I')".
- A red box highlights the result value "2" in the results table.

## Función 2

función que reciba el DNI de un cliente y devuelva el porcentaje de pedidos que ha realizado respecto al total de pedidos en el sistema

```

DELIMITER $$

CREATE FUNCTION porcentaje_pedidos_cliente(dni_Cliente VARCHAR(45))
RETURNS DECIMAL(5,2)
DETERMINISTIC
BEGIN

    DECLARE pedidos_clientes INT DEFAULT 0;
    DECLARE pedidos_totales int default 0;
    DECLARE porcentaje DECIMAL(7,2) DEFAULT 0;

    SELECT count(*) into pedidos_clientes
    FROM Cliente c
    join Pedido p
    on c.DNI_Cliente = p.Cliente_DNI_Cliente
    WHERE c.DNI_Cliente = dni_Cliente;

    select count(*) into pedidos_totales
    from Pedido p ;

    SET porcentaje = (pedidos_clientes * 100.0) / pedidos_totales;

    return porcentaje;

END $$

DELIMITER ;

```

**select porcentaje\_pedidos\_cliente('27009521H');**

Resultados 1 X

select porcentaje\_pedidos\_cliente('27009521H') | Enter a SQL expression to filter results (use Ctrl+Space)

123	porcentaje_pedidos_cliente('27009521H')	▼
		0,2

## Procedimiento 1

**procedimiento que muestre los 5 productos con más unidades vendidas en el último mes**

```

DELIMITER $$

CREATE PROCEDURE top_5_productos_vendidos_mes (IN fecha_inicio DATE,
IN fecha_fin DATE)
BEGIN
    SELECT p.idProducto, p.Descripcion, sum(pdp.Cantidad) as total_vendido,
    count(DISTINCT pdp.Pedido_idPedido) as num_pedidos
    FROM Producto p
    JOIN Productos_del_pedido pdp
    on p.idProducto = pdp.Producto_idProducto
    join Pedido pe
    on pdp.Pedido_idPedido = pe.idPedido
    WHERE pe.Fecha_Pedido BETWEEN fecha_inicio AND fecha_fin
    GROUP BY p.idProducto, p.Descripcion
    ORDER BY total_vendido DESC
    LIMIT 5;

END $$

DELIMITER ;

```

```
CALL top_5_productos_vendidos_mes('2025-01-01', '2025-12-31');
```

ducto 1 X Estadísticas 1

.L top\_5\_productos\_vendidos\_mes('2025-01-01', '20 | ↵ Enter a SQL expression to filter results (use Ctrl+Space)

123 idProducto	A-Z Descripcion	123 total_vendido	123 num_pedidos
539	OnePlus Accesorio modelo 246 con características avanzadas	10	2
82	Samsung Accesorio modelo 823 con características avanzadas	10	2
473	Huawei Tablet modelo 306 con características avanzadas	10	2
267	Sony Smartwatch modelo 952 con características avanzadas	9	2
207	Realme Smartphone modelo 265 con características avanzadas	9	2

## Procedimiento 2

**procedimiento para ver Clientes que han comprado 2 o más productos diferentes de la misma marca**

```

DELIMITER $$

CREATE PROCEDURE clientes_fieles_a_marca()
BEGIN
    SELECT c.DNI_Cliente, c.Nombre, p2.Marca,
    count(DISTINCT p2.idProducto) as productos_distintos
    FROM Cliente c
    JOIN Pedido p
    on c.DNI_Cliente = p.Cliente_DNI_Cliente
    JOIN Productos_del_pedido pdp
    on p.idPedido = pdp.Pedido_idPedido
    JOIN Producto p2
    on pdp.Producto_idProducto = p2.idProducto
    GROUP BY c.DNI_Cliente, c.Nombre, p2.Marca
    HAVING COUNT(DISTINCT p2.idProducto) >= 2
    ORDER BY c.Nombre, p2.Marca;
END $$

DELIMITER ;

CALL clientes_fieles_a_marca();

```

iente(+) 1 X Estadísticas 1

ALL clientes\_fieles\_a\_marca() |  Enter a SQL expression to filter results (use Ctrl+Space)

O	A-Z DNI_Cliente	A-Z Nombre	A-Z Marca	123 productos_distintos
0	79895707L	Ana	Huawei	2
	06259269C	Ana	Huawei	2
	32864230Z	Ana	Motorola	3
	79895707L	Ana	Realme	2
	21291802Y	Ana	Xiaomi	2
	51461667F	Ana	Xiaomi	2
	85987118R	Ana	Xiaomi	2
	27215325O	Carlos	Apple	2
	27009521H	Carlos	Apple	3
0	79838949N	Carlos	Huawei	2

### Procedimiento 3

**Procedimiento para ver el número de pedido, la fecha y el cliente al que ha atendido un empleado**

```

DELIMITER $$

create procedure pedidos_por_empleado (in p_id_empleado varchar(45)
)
begin
select p.Num_Pedido,p.Fecha_Pedido,c.Nombre as cliente
from Pedido p
join Cliente c
on p.Cliente_DNI_Cliente = c.DNI_Cliente
join Pago p2
on p.idPedido = p2.Pedido_idPedido
where p.Empleado_idEmpleado = p_id_empleado
group by p.idPedido, p.Num_Pedido, p.Fecha_Pedido, c.Nombre
order by p.Fecha_Pedido desc;
end $$

DELIMITER ;

call pedidos_por_empleado('E0009');

do(+ 1) X Estadísticas 1
pedidos_por_empleado('E0009') | Enter a SQL expression to filter results (use Ctrl+Space)



| Num_Pedido | Fecha_Pedido        | cliente |
|------------|---------------------|---------|
| 36.075     | 2024-05-13 19:11:32 | Laura   |


```

## 8. Triggers

Trigger 1

Trigger para actualizar stock al original si se cancela un pedido

```
DELIMITER $$

create trigger devolver_stock_al_eliminar
after delete on Productos_del_pedido
for each row
begin
    update Producto
    set stock = stock + OLD.Cantidad
    where idProducto = OLD.Producto_idProducto;
end $$

DELIMITER ;
```

idProducto	stock
1	246

```
delete from Productos_del_pedido
where Pedido_idPedido = 10 and Producto_idProducto = 1;
```

Query      delete from Productos\_del\_pedido  
              where Pedido\_idPedido = 10 and Producto\_idProducto = 1

Updated Rows 0

Execute time 0,008s

Start time Tue May 20 23:21:40 CEST 2025

Finish time Tue May 20 23:21:40 CEST 2025

```
DELIMITER $$

CREATE TRIGGER devolver_stock_al_eliminar
AFTER DELETE ON Productos_del_pedido
FOR EACH ROW
BEGIN
    UPDATE Producto
    SET stock = stock + OLD.Cantidad
    WHERE idProducto = OLD.Producto_idProducto;
END $$

DELIMITER ;

SELECT idProducto, stock FROM Producto WHERE idProducto = 1;

DELETE FROM Productos_del_pedido
WHERE Pedido_idPedido = 10 AND Producto_idProducto = 1;

SELECT idProducto, stock FROM Producto WHERE idProducto = 1;
```

producto 1 X

elect idProducto, stock from Producto where idProduc | Enter a SQL expression to filter results (use

	123 idProducto	123 stock
1	1	246

## Trigger 2

**Trigger para impedir que se inserte un pago con importe negativo o mayor al total del pedido**

```

DELIMITER $$

create trigger validar_pago
before insert on Pago
for each row
begin
    declare total_permitido decimal(10,2);
    declare mensaje_error varchar(255);

    select sum(pdp.Cantidad * pr.Precio)
    into total_permitido
    from Productos_del_pedido pdp
    join Producto pr on pdp.Producto_idProducto = pr.idProducto
    where pdp.Pedido_idPedido = NEW.Pedido_idPedido;

    if NEW.Importe < 0 then
        signal sqlstate '45000';
        set message_text = 'El importe no puede ser negativo.';
    end if;

    if NEW.Importe > total_permitido then
        set mensaje_error = concat('El importe no puede superar el total del pedido: ', total_permitido);
        signal sqlstate '45000';
        set message_text = mensaje_error;
    end if;
end $$

DELIMITER ;

insert into Pago (Pedido_idPedido, Fecha_Pago, Importe)
values (10, curdate(), -100);

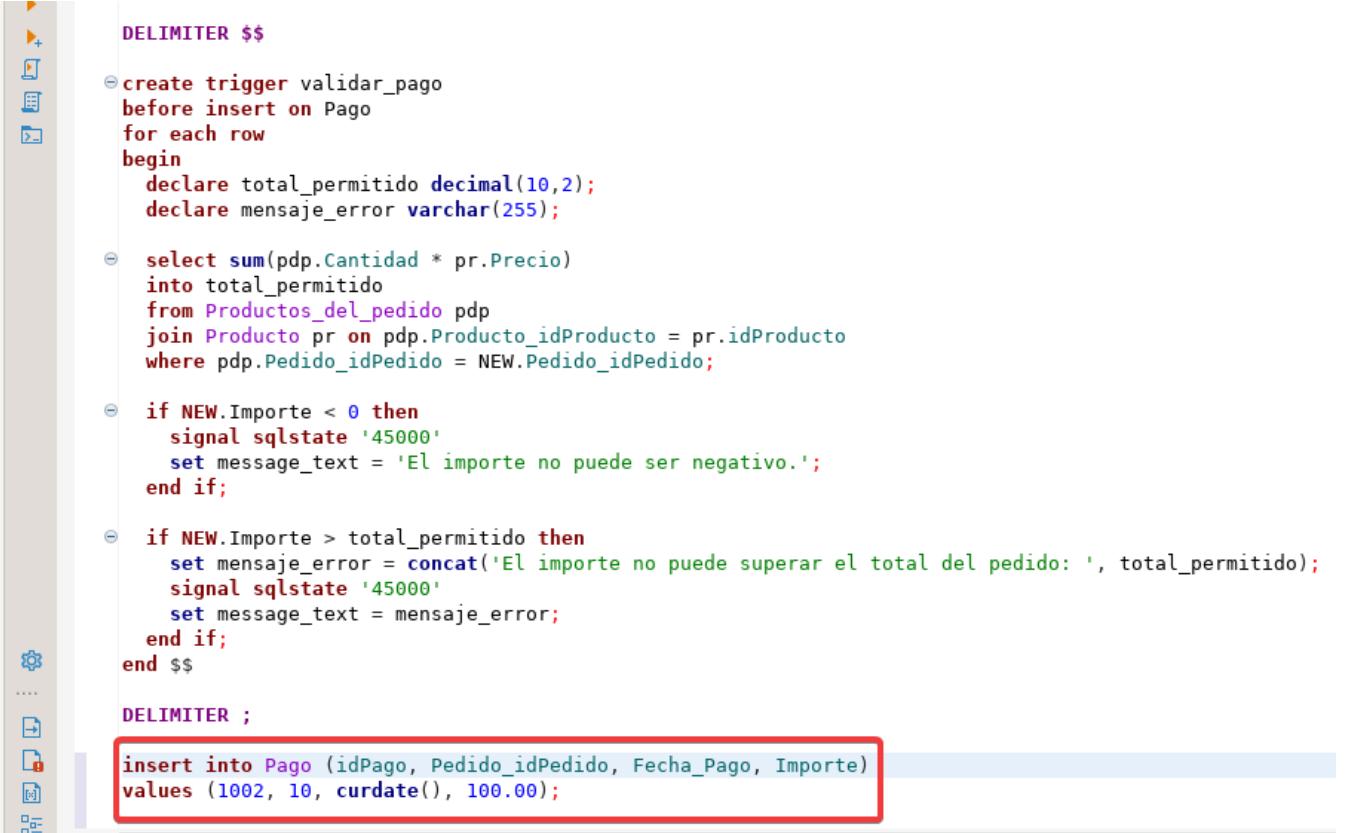
```

Product 1 X

select idProducto, stock from Producto where idProduc | Enter a SQL expression to filter results (use Ctrl+Space)

SQL Error [1644] [45000]: El importe no puede ser negativo.

insert into Pago (Pedido\_idPedido, Fecha\_Pago, Importe)



```

DELIMITER $$

CREATE TRIGGER validar_pago
BEFORE INSERT ON Pago
FOR EACH ROW
BEGIN
    DECLARE total_permitido DECIMAL(10,2);
    DECLARE mensaje_error VARCHAR(255);

    SELECT SUM(pdp.Cantidad * pr.Precio)
    INTO total_permitido
    FROM Productos_del_pedido pdp
    JOIN Producto pr ON pdp.Producto_idProducto = pr.idProducto
    WHERE pdp.Pedido_idPedido = NEW.Pedido_idPedido;

    IF NEW.Importe < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET message_text = 'El importe no puede ser negativo.';
    END IF;

    IF NEW.Importe > total_permitido THEN
        SET mensaje_error = CONCAT('El importe no puede superar el total del pedido: ', total_permitido);
        SIGNAL SQLSTATE '45000'
        SET message_text = mensaje_error;
    END IF;
END $$

DELIMITER ;

INSERT INTO Pago (idPago, Pedido_idPedido, Fecha_Pago, Importe)
VALUES (1002, 10, curdate(), 100.00);

```

**Estadísticas 1**

Name	Value
Query	insert into Pago (idPago, Pedido_idPedido, Fecha_Pago, Importe) values (1002, 10, curdate(), 100.00)
Updated Rows	1
Execute time	0,011s
Start time	Tue May 20 23:28:39 CEST 2025
Finish time	Tue May 20 23:28:39 CEST 2025

## 9. Valoración Personal

La carga masiva me ha costado bastante de hacer ya que mockaroo me da los tipos de datos mal en algunas ocasiones y hace que haya errores por todos lados. He tenido problemas ya que al tener que cambiar varias cosas en la base de datos como relaciones he tenido que volver a hacer la carga masiva y todo de nuevo, una vez todo hecho. El hacer las consultas,funciones y procedimientos no me han costado mucho y me han ayudado un poco también para repasar para el exámen, en resumen he aprendido bastante y ha estado bastante bien.

Enlace github:

<https://github.com/Antonio-2511/Proyecto-BD>