



## **Ingeniería de datos 1**

### **Trabajo Práctico Obligatorio - 2° cuatrimestre 2023**

**WeClover**

#### **Integrantes:**

1138357	Lavezzari, Ignacio
1149229	Morinigo , Ignacio
1147619	Alan Orlando
1151151	Redaelli, Nicolas

#### **Docentes:**

Franco Salazar

Gustavo Escadell

## INDICE

Contexto del Proyecto .....	3
Propósito del Proyecto .....	3
Alcance del Proyecto .....	3
Narrativa .....	3
DER.....	5
Modelo Relacional .....	6
Arquitectura Física de la Base de Datos .....	7
Procedimiento de carga.....	9
Modificación .....	15
Eliminación.....	16
Consultas.....	19
Conclusión.....	26

## Introducción

### Contexto del Proyecto

WeClover se enfrenta al desafío de optimizar sus operaciones comerciales, específicamente en el ámbito de la gestión de ventas. La empresa se especializa en la comercialización de indumentaria y comprende la importancia de implementar un sistema robusto que permita organizar y monitorear cada aspecto de sus transacciones y relaciones con los clientes.

### Propósito del Proyecto

El propósito principal de este proyecto es diseñar una Base de Datos que sirva como cimiento sólido para la gestión eficiente de las ventas. La base conceptual se centra en la relación entre clientes, pedidos, pagos, empleados y productos, proporcionando una estructura lógica que facilite la recuperación de información relevante y el análisis detallado de la actividad comercial.

### Alcance del Proyecto

El alcance de este proyecto incluye la definición de entidades clave, sus atributos y las relaciones entre ellas. Se abordarán consultas específicas que la Base de Datos debe ser capaz de responder, con el fin de cumplir con los requisitos operativos y de análisis de WeClover. El diseño propuesto se enfocará en la flexibilidad y escalabilidad, permitiendo futuras adaptaciones según las necesidades del negocio.

### Narrativa

Queremos armar un modelo de una Base de Datos para la gestión de venta de una empresa de indumentaria, sabemos en un principio que un cliente puede realizar varios pedidos y que a su vez este pedido tendrá un detalle de pedido y además sabemos que un cliente puede realizar varios pagos y un pago solo puede realizarlo una persona.

Cada Cliente estará de forma única en la Base de Datos, sabemos que cada usuario tiene un código de cliente único, el nombre del mismo, apellido1 y apellido2 donde este último no debe ser obligatorio, sabemos que cada cliente también tiene un número de identidad obligatorio, su fecha de nacimiento y atributos de ubicación y contacto al cliente como país (el cliente puede realizar pedidos desde Brasil, Paraguay y Argentina), ciudad, región, código postal, teléfono y mail estos atributos son claves y suma importancia ya que nos ayudara a identificar a cada usuario, sabemos a su vez que varios clientes pueden ser atendidos por varios empleados.

Cada empleado estará de forma única en la Base de Datos, el Cliente tiene un código único de empleado, un nombre, apellido1 y apellido2 donde en este último un empleado podría no tener un segundo apellido y además sabemos que cada empleado tiene un mail de contacto obtenido por la empresa, el mismo nos ayudara a tomar contacto con el empleado.

Un pedido tiene un código único de pedido, el código de cliente al que esta vinculado el pedido, la fecha en la que se hizo el pedido, una fecha esperada de pedido, una fecha de entrega y a su vez un pedido tiene el estado de pedido (por ejemplo, espera, entregado, rechazado) y el total del mismo, estos atributos son de suma importancia y ayuda a

establecer una línea base para controlar la gestión de entrega del pedido, además, un pedido puede tener varios detalles de pedido y varios detalles de pedido pueden tener un producto.

Cada detalle de pedido tiene un código único que identifica a cada uno, además un detalle de pedido tiene un código de pedido que a idéntica a el pedido que pertenece el mismo, un detalle de pedido debe poseer un código producto y la cantidad de productos que se quieren adquirir y cambien un subtotal que nos dará el costo del mismo.

Con respecto al Pago sabemos que cada uno de estos posee un código de identificación único y el código de cliente que realizo el pago, donde un cliente puede realizar varios pagos y un pago está asociado a un cliente, también, un pago tiene asociado el código de forma de pago al que se asocia el mismo, la fecha en la que se hizo el pago y el total del mismo. Un pago puede tener una forma de pago mientras que una forma de pago puede tener asociado varios pagos.

Cada forma de pago tiene un código de identificación único y este a su vez tiene el detalle de pago que hace referencia a la forma en la que se hizo el mismo (por ejemplo, tarjeta de débito, tarjeta de crédito, transferencia o efectivo).

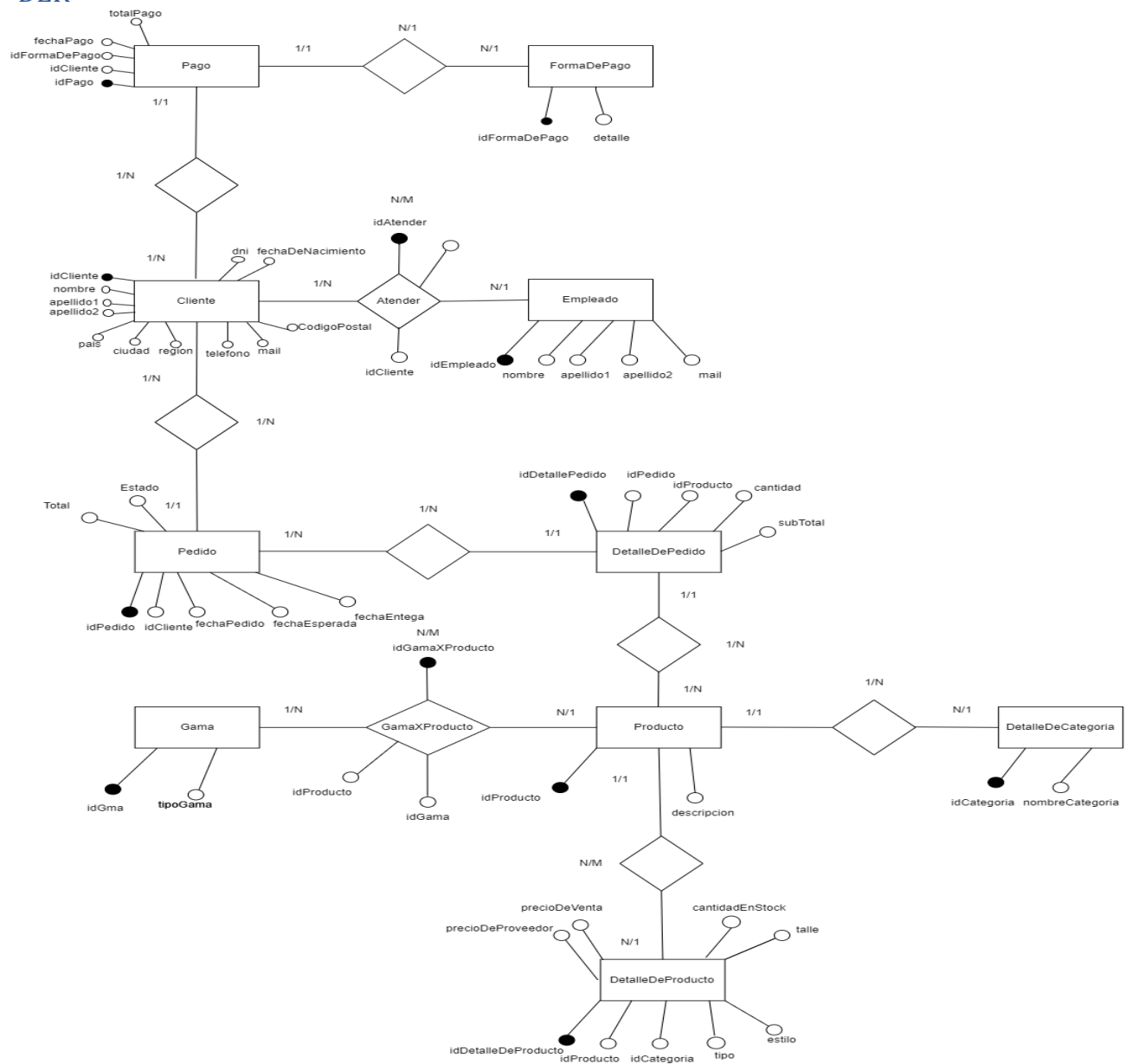
Cada Producto tiene una clave única de identificación y la descripción que nos brinda con mas detalle el tipo de producto, sabemos que varios productos pueden pertenecer a una gama de producto y que varias gamas de producto pueden tener varios productos asociados. También, un producto tiene asociado el detalle de producto, donde un producto puede tener varios detalles de producto y así mismo un detalle de producto puede tener un producto asociado.

Con respecto al detalle de producto, un detalle de producto tiene una clave de identificación única, el código de identificación único asociado al producto y también está asociado a la categoría de producto (por ejemplo masculino, femenino, unisex) donde varios detalles de producto pueden estar asociados a una categoría, por otro lado detalle de producto cuenta con tipo que da una visión del modelo de producto al que se refiere, estilo que describe el color o diseño del producto, el talle del producto que hace referencia al tamaño del mismo, la cantidad en stock y por otro lado el precio de venta que hace referencia al precio al que se ofrece a los clientes y el precio del proveedor que el mismo es el costo de adquisición por parte de la empresa. Donde el precio de venta debe ser mayor al precio de proveedor.

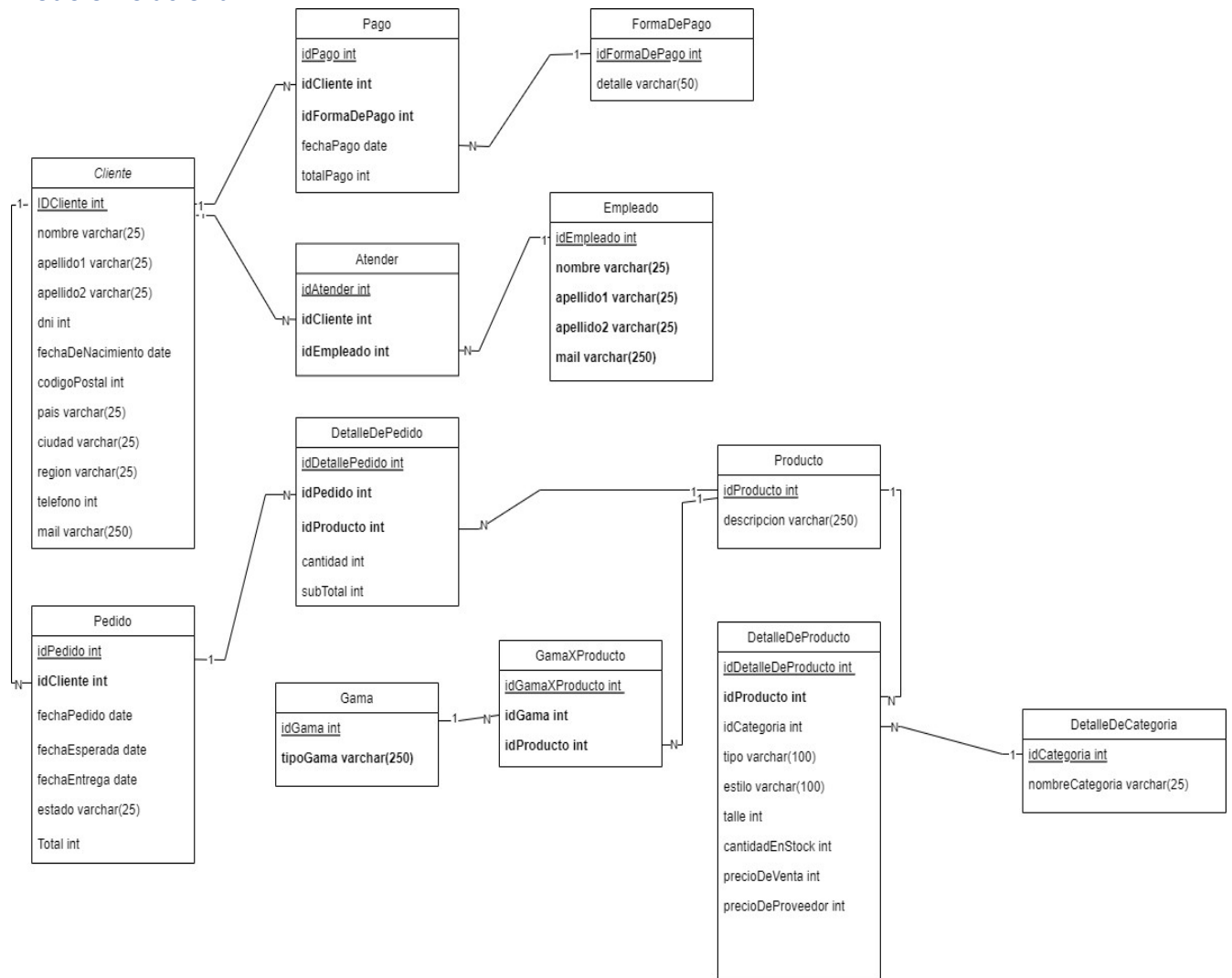
El detalle de categoría tiene una clave de identificación única del mismo junto con el nombre de la misma que hace referencia al genero al que pertenece el producto (por ejemplo, masculino, femenino, unisex).

Se sabe que la categoría posee una clave de identificación única y el tipo de gama que hace referencia al nombre de la misma. También, Un código de gama puede estar asociado a varios productos.

DER



## Modelo Relacional



## Arquitectura Fisica de la Base de Datos

```
create table Cliente(
    idCliente int primary key identity(1,1),
    nombre varchar(25) not null,
    apellido1 varchar(25) not null,
    apellido2 varchar(25),
    dni int not null,
    fechaDeNacimiento date not null,
    codigoPostal int not null,
    pais varchar(25) not null,
    ciudad varchar(25) not null,
    region varchar(25) not null,
    telefono int not null,
    mail varchar(250) not null
);

create table Empleado(
    idEmpleado int primary key identity(1,1),
    nombre varchar(25) not null,
    apellido1 varchar(25) not null,
    apellido2 varchar(25),
    mail varchar(250) not null
);

create table Atender(
    idAtender int primary key identity(1,1),
    idCliente int not null,
    idEmpleado int not null,
    constraint FK_cliente foreign key (idCliente) references Cliente(idCliente),
    constraint FK_empleado foreign key (idEmpleado) references Empleado(idEmpleado)
);

create table FormaDePago(
    idFormaDePago int primary key identity(1,1),
    detalle varchar(50) not null
    constraint CK_detalle check (detalle in ('Tarjeta de Credito','Tarjeta Debito','Tranferencia','Al Contado'))
);

create table Pago(
    idPago int primary key identity(1,1),
    idCliente int not null,
    idFormaDePago int not null,
    fechaPago date not null,
    totalPago int not null
    constraint CK_TotalPago check (totalPago >= 1),
    constraint FK_formaDePago foreign key (idFormaDePago) references FormaDePago(idFormaDePago),
    constraint FK_idClienteP foreign key (idCliente) references Cliente(idCliente)
);

create table Gama(
    idGama int primary key identity(1,1),
    tipoGama varchar(250) not null
);

create table Producto(
    idProducto int primary key identity(1,1),
    descripcion varchar(250) not null
);

create table GamaXProducto(
    idGamaXProducto int primary key identity(1,1),
    idGama int not null,
    idProducto int not null
    constraint FK_Gama foreign key (idGama) references Gama(idGama),
    constraint FK_ProductoG foreign key (idProducto) references Producto(idProducto)
);
```

```
create table DetalleDeCategoria(  
    idCategoria int primary key identity(1,1),  
    nombreCategoria varchar(25) not null  
);  
  
create table DetalleDeProducto(  
    idDetalleDeProducto int primary key identity(1,1),  
    idProducto int not null,  
    idCategoria int not null,  
    tipo varchar(100) not null,  
    estilo varchar(100) not null,  
    talla varchar(25) not null,  
    cantidadEnStock int not null,  
    precioDeVenta int not null,  
    precioDeProveedor int not null,  
    constraint CK_Talle check (talle in ('XS','S','M','L','XL','XXL')),  
    constraint CK_CantidadEnStock check (cantidadEnStock > -1),  
    constraint CK_precioDeVenta check (precioDeVenta >= 1),  
    constraint CK_PDeProveedor check (precioDeProveedor >= 1),  
    constraint FK_Producto foreign key (idProducto) references Producto(idProducto) on delete cascade,  
    constraint FK_CategoriaDet foreign key (idCategoria) references DetalleDeCategoria (idCategoria)  
);  
  
create table Pedido(  
    idPedido int primary key identity(1,1),  
    idCliente int not null,  
    fechaPedido date not null,  
    fechaEspera date not null,  
    fechaEntrega date not null,  
    estado varchar(25),  
    Total int  
    constraint CK_total check (total >= 1),  
    constraint CK_estado check (estado in ('Espera','Entregado','Rechazado')),  
    constraint FK_idcliente foreign key (idCliente) references Cliente(idCliente)  
);  
  
create table DetalleDePedido(  
    idDetallePedido int primary key identity(1,1),  
    idPedido int not null,  
    idProducto int not null,  
    cantidad int not null,  
    subTotal int not null,  
    constraint CK_cantidad check (cantidad >= 1),  
    constraint CK_subTotal check (subTotal >= 1),  
    constraint FK_idProducto foreign key (idProducto) references Producto(idProducto),  
    constraint FK_Pedido foreign key (idPedido) references Pedido(idPedido)  
);
```



## Procedimiento de carga

### Carga de Cliente

```

create procedure cargarCliente
@nombre varchar(25),
@apellido1 varchar(25),
@apellido2 varchar(25),
@dni int,
@fechaDeNacimiento date,
@codigoPostal int,
@pais varchar(25),
@ciudad varchar(25),
@region varchar(25),
@telefono int,
@email varchar(25)
as
insert into Cliente values (@nombre,@apellido1,@apellido2,@dni,@fechaDeNacimiento,@codigoPostal,@pais,@ciudad,@region,@telefono,@email)

exec cargarCliente 'Alan','Orlando',null,45789878,'2003/02/14',1253,'Argentina','La Matanza','Buenos Aires',1134546487,'AOrlando@hotmail.com'
exec cargarCliente 'Jazmin','Martin','Salem',45778889,'2003/07/17',3457,'Argentina','Quilmes','Buenos Aires',1198348754,'Jazmin512@gmail.com'
exec cargarCliente 'Oscar','villalba',null,24984576,'1974/04/21',1253,'Argentina','La Matanza','Buenos Aires',1122460967,'Oscar_villalba@yahoo.com'
exec cargarCliente 'Micaela','Fuentes','Padilla',45986734,'2004/10/02',6753,'Argentina','Belgrano','Buenos Aires',1189369867,'MFuentes@gmail.com'
exec cargarCliente 'Florenia','Borda','Servizuela',44789832,'2002/12/22',8976,'Argentina','Claypole','Buenos Aires',1134879809,'Flor_servi@BuenosAires.edu.ar'
exec cargarCliente 'Roman','Stella',null,92456178,'1967/12/19',1881,'Argentina','San Francisco Solano','Buenos Aires',1122983409,'RomanStel@gmail.com'
exec cargarCliente 'Ana','García','López',23456789,'1992/07/10',2500,'Argentina','Rosario','Santa Fe',1122334455,'ana.garcia@hotmail.com'
exec cargarCliente 'Pedro','Santos','Silva',87654321,'1987/04/25',3200,'Argentina','San Nicolas','Rosario',1144332211,'pedro.santos@hotmail.com'
exec cargarCliente 'Maria','López','Fernández',34567890,'1990/01/15',4100,'Uruguay','Colonia','Departamento de colonia',1577889900,'maria.lopez@yahoo.com'
exec cargarCliente 'Juan','Martínez','Gómez',45678901,'1988/09/05',2800,'Argentina','Cordoba','Cordoba Capital',1188990011,'juan.martinez@gamil.com'
exec cargarCliente 'Luis','Oliveira','Silva',12345678,'1995/02/20',3300,'Paraguay','Asuncion','Manora',1500112233,'luis.oliveira@hotmail.com'
exec cargarCliente 'Sofia','Pérez','Rodríguez',56789012,'1986/06/12',4200,'Uruguay','Montevideo','Departamento de Montevideo',1122334455,'sofia.perez@gmail.com'
exec cargarCliente 'Miguel','Fernández',null,67890123,'1991/03/08',2600,'Argentina','Lomas De Zamora','Buenos Aires',1144332211,'miguel.fernandez@hotmail.com'
exec cargarCliente 'Isabel','Silva','Martins',34567890,'1989/10/30',3700,'Argentina','Bahia Blanca','Buenos Aires',1577889900,'isabel.silva@hotmail.com'
exec cargarCliente 'Eduardo','Rodríguez',null,90123456,'1994/05/22',4300,'Argentina','Las Grutas','Rio Negro',1588990011,'eduardo.rodriquez@hotmail.com'
exec cargarCliente 'Carla','Gómez','Fernández',23456789,'1985/12/18',2990,'Argentina','Rosario','Santa Fe',1500112233,'carla.gomez@gmail.com'
exec cargarCliente 'Antonio','Santos',null,56789012,'1993/08/14',3400,'Argentina','Corrientes','Corrientes Capital',1122334455,'antonio.santos@gmail.com'
exec cargarCliente 'Rosa','Pérez','García',12345678,'1986/04/02',4511,'Uruguay','Montevideo','Departamento de Montevideo',1144332211,'rosa.perez@gmail.com'
exec cargarCliente 'Fernando','Martínez','López',23456789,'1990/11/20',3098,'Argentina','Buenos Aires','Palermo',1177889909,'fernando.martinez@gmail.com'
exec cargarCliente 'Clara','Oliveira','Silva',57654321,'1987/07/15',3645,'Argentina','Salta','San Lorenzo',1188990011,'clara.oliveira@gmail.com'

```

### Cargar Empleado

```

create procedure cargarEmpleado
@nombre varchar(25),
@apellido1 varchar(25),
@apellido2 varchar(25),
@email varchar(250)
as
insert into Empleado values (@nombre,@apellido1,@apellido2,@email)

```

```

exec cargarEmpleado 'Juan','Pérez','Gómez','juan.perez@weColver.com.ar'
exec cargarEmpleado 'María','González','López','maria.gonzalez@weColver.com.ar'
exec cargarEmpleado 'Carlos','Martínez','Fernández','carlos.martinez@weColver.com.ar'
exec cargarEmpleado 'Laura','López','Rodríguez','laura.lopez@weColver.com.ar'
exec cargarEmpleado 'Pedro','Sánchez','Hernández','pedro.sanchez@weColver.com.ar'
exec cargarEmpleado 'Ana','Ramírez','García','ana.ramirez@weColver.com.ar'
exec cargarEmpleado 'Luis','Díaz','Pérez','luis.diaz@weColver.com.ar'
exec cargarEmpleado 'Elena','Fernández',NULL,'elena.fernandez@weColver.com.ar'
exec cargarEmpleado 'Miguel','Moreno','Jiménez','miguel.moreno@weColver.com.ar'
exec cargarEmpleado 'Carmen','Ruiz','Sánchez','carmen.ruiz@weColver.com.ar'
exec cargarEmpleado 'Javier','Gómez',NULL,'javier.gomez@weColver.com.ar'
exec cargarEmpleado 'Rosa','Martín','Serrano','rosa.martin@weColver.com.ar'
exec cargarEmpleado 'Alejandro','Torres',NULL,'alejandro.torres@weColver.com.ar'
exec cargarEmpleado 'Isabel','Ortega','Núñez','isabel.ortega@weColver.com.ar'
exec cargarEmpleado 'Sergio','Hernández','Vega','sergio.hernandez@weColver.com.ar'
exec cargarEmpleado 'Lucía','Morales','Silva','lucia.morales@weColver.com.ar'
exec cargarEmpleado 'Pablo','Vargas','Ramos','pablo.vargas@weColver.com.ar'
exec cargarEmpleado 'Eva','Jiménez',NULL,'eva.jimenez@weColver.com.ar'
exec cargarEmpleado 'Antonio','López','Cabrera','antonio.lopez@weColver.com.ar'
exec cargarEmpleado 'Beatriz','García',NULL,'beatriz.garcia@weColver.com.ar'

```

### Cargar Atender

```
create procedure cargarAtender
@idCliente int,
@idEmpleado int
as
insert into Atender values (@idCliente,@idEmpleado)
```

```
exec cargarAtender 1, 1
exec cargarAtender 3, 2
exec cargarAtender 5, 3
exec cargarAtender 7, 4
exec cargarAtender 9, 5
exec cargarAtender 11, 6
exec cargarAtender 1, 7
exec cargarAtender 13, 8
exec cargarAtender 15, 9
exec cargarAtender 3, 10
exec cargarAtender 2, 11
exec cargarAtender 5, 12
exec cargarAtender 7, 13
exec cargarAtender 1, 14
exec cargarAtender 11, 15
exec cargarAtender 13, 16
exec cargarAtender 15, 17
exec cargarAtender 17, 18
exec cargarAtender 19, 19
exec cargarAtender 1, 20
```

#### Cargar Producto

```
create procedure cargarProducto
@descripcion varchar(250)
as
insert into Producto values (@descripcion)
```

```
EXEC cargarProducto 'Camiseta'
EXEC cargarProducto 'Pantalones'
EXEC cargarProducto 'Chaqueta'
EXEC cargarProducto 'Vestido'
EXEC cargarProducto 'Camisa'
EXEC cargarProducto 'Bufanda'
EXEC cargarProducto 'Gorro'
EXEC cargarProducto 'Zapatillas'
EXEC cargarProducto 'Falda'
EXEC cargarProducto 'Abrigo'
EXEC cargarProducto 'Suéter'
EXEC cargarProducto 'Sandalias'
EXEC cargarProducto 'Ojotas'
EXEC cargarProducto 'Blusa'
EXEC cargarProducto 'Calzas'
EXEC cargarProducto 'Calcetines'
EXEC cargarProducto 'Traje De Baño'
EXEC cargarProducto 'Sombrero'
EXEC cargarProducto 'Chaleco'
EXEC cargarProducto 'Guantes'
EXEC cargarProducto 'Pijama'
```

#### Cargar DetalleDeProducto

```
create procedure cargarDetalleDeProducto
@idProducto int,
@idCategoria int,
@tipo varchar(100),
@estilo varchar(100),
@talle varchar(25),
@cantidadEnStock int,
@precioDeVenta int,
@precioDeProveedor int
as
insert into DetalleDeProducto values (@idProducto,@idCategoria,@tipo,@estilo,@talle,@cantidadEnStock,@precioDeVenta,@precioDeProveedor)
```

```
EXEC cargarDetalleDeProducto 1, 1, 'Jean', 'Azul', 'M', 50, 25000, 18000
EXEC cargarDetalleDeProducto 2, 2, 'Cuero', 'Negro', 'L', 30, 35000, 28000
EXEC cargarDetalleDeProducto 3, 3, 'Sintético', 'Rojo', 'S', 40, 18000, 12000
EXEC cargarDetalleDeProducto 4, 1, 'Tela', 'Verde', 'XL', 25, 30000, 22000
EXEC cargarDetalleDeProducto 5, 2, 'Denim', 'Blanco', 'S', 60, 28000, 20000
EXEC cargarDetalleDeProducto 6, 3, 'Algodón', 'Amarillo', 'M', 45, 22000, 15000
EXEC cargarDetalleDeProducto 7, 1, 'Lino', 'Gris', 'L', 35, 32000, 24000
EXEC cargarDetalleDeProducto 8, 2, 'Poliéster', 'Negro', 'XL', 28, 40000, 30000
EXEC cargarDetalleDeProducto 9, 3, 'Seda', 'Rosa', 'S', 20, 28000, 20000
EXEC cargarDetalleDeProducto 10, 1, 'Tejido', 'Azul Marino', 'M', 50, 25000, 18000
EXEC cargarDetalleDeProducto 11, 2, 'Denim', 'Vino', 'L', 30, 35000, 28000
EXEC cargarDetalleDeProducto 12, 3, 'Algodón', 'Blanco', 'S', 40, 18000, 12000
EXEC cargarDetalleDeProducto 13, 1, 'Lino', 'Negro', 'XL', 25, 30000, 22000
EXEC cargarDetalleDeProducto 14, 2, 'Sintético', 'Gris', 'S', 60, 28000, 20000
EXEC cargarDetalleDeProducto 15, 3, 'Tejido', 'Verde', 'M', 45, 22000, 15000
EXEC cargarDetalleDeProducto 16, 1, 'Denim', 'Marrón', 'L', 35, 32000, 24000
EXEC cargarDetalleDeProducto 17, 2, 'Poliéster', 'Beige', 'XL', 28, 40000, 30000
EXEC cargarDetalleDeProducto 18, 3, 'Algodón', 'Negro', 'S', 20, 28000, 20000
EXEC cargarDetalleDeProducto 19, 1, 'Sintético', 'Azul', 'M', 50, 25000, 18000
EXEC cargarDetalleDeProducto 20, 2, 'Denim', 'Blanco', 'L', 30, 35000, 28000
```

#### Cargar DetalleDeCategoria

```
create procedure cargarDetalleDeCategoria
@nombreCategoria varchar(25)
as
insert into DetalleDeCategoria values (@nombreCategoria)

EXEC cargarDetalleDeCategoria 'Hombre'
EXEC cargarDetalleDeCategoria 'Mujer'
EXEC cargarDetalleDeCategoria 'Unisex'
```

#### Cargar Gama

```
create procedure cargarGama
@tipoGama varchar(250)
as
insert into Gama values (@tipoGama)
```

```
EXEC cargarGama 'Casual';
EXEC cargarGama 'Formal';
EXEC cargarGama 'Deportivo';
EXEC cargarGama 'Elegante';
EXEC cargarGama 'Urbano';
EXEC cargarGama 'Bohemia';
EXEC cargarGama 'Retro';
EXEC cargarGama 'Moderna';
EXEC cargarGama 'Vintage';
EXEC cargarGama 'Minimalista';
EXEC cargarGama 'Colorida';
EXEC cargarGama 'Estacional';
EXEC cargarGama 'Ecuestre';
EXEC cargarGama 'Playera';
EXEC cargarGama 'Nocturna';
EXEC cargarGama 'Aventurera';
EXEC cargarGama 'Confortable';
EXEC cargarGama 'Chic';
EXEC cargarGama 'Sostenible';
EXEC cargarGama 'Edicion Limitada';
```

### Cargar Pedido

```
create procedure cargarPedido
@idCliente int,
@fechaPedido date,
@fechaEsperada date,
@fechaentrega date,
@estado varchar(25),
@total int
as
insert into Pedido values (@idCliente,@fechaPedido,@fechaEsperada,@fechaentrega,@estado,@total)
```

```
exec cargarPedido 1, '2020-10-10', '2020-10-24', '2020-10-24', 'Entregado', 20000
exec cargarPedido 1, '2022-05-15', '2022-05-20', '2022-05-18', 'Entregado', 15000
exec cargarPedido 2, '2021-07-02', '2021-07-10', '2021-07-08', 'Rechazado', 18000
exec cargarPedido 5, '2023-01-20', '2023-01-25', '2023-01-23', 'Espera', 22000
exec cargarPedido 10, '2022-11-08', '2022-11-15', '2022-11-12', 'Entregado', 13000
exec cargarPedido 15, '2021-09-30', '2021-10-05', '2021-10-03', 'Espera', 25000
exec cargarPedido 8, '2023-03-12', '2023-03-18', '2023-03-15', 'Rechazado', 17000
exec cargarPedido 12, '2022-08-05', '2022-08-10', '2022-08-08', 'Espera', 19000
exec cargarPedido 3, '2021-12-18', '2021-12-25', '2021-12-22', 'Entregado', 20000
exec cargarPedido 6, '2023-02-25', '2023-03-02', '2023-02-28', 'Espera', 16000
exec cargarPedido 18, '2022-04-03', '2022-04-10', '2022-04-08', 'Rechazado', 21000
exec cargarPedido 4, '2021-10-14', '2021-10-20', '2021-10-18', 'Entregado', 23000
exec cargarPedido 11, '2023-05-08', '2023-05-15', '2023-05-12', 'Espera', 14000
exec cargarPedido 7, '2022-06-22', '2022-06-28', '2022-06-25', 'Rechazado', 26000
exec cargarPedido 9, '2021-11-11', '2021-11-18', '2021-11-15', 'Espera', 18000
exec cargarPedido 14, '2023-04-01', '2023-04-08', '2023-04-05', 'Entregado', 17000
exec cargarPedido 13, '2022-09-19', '2022-09-26', '2022-09-23', 'Espera', 24000
exec cargarPedido 20, '2021-08-08', '2021-08-15', '2021-08-12', 'Rechazado', 20000
exec cargarPedido 17, '2023-06-05', '2023-06-12', '2023-06-10', 'Espera', 22000
exec cargarPedido 16, '2022-03-27', '2022-04-03', '2022-03-30', 'Entregado', 16000
exec cargarPedido 19, '2021-12-10', '2021-12-18', '2021-12-15', 'Rechazado', 19000
```

### Cargar GamaXProducto

```
create procedure cargarGamaXProducto
@idGama int,
@idProducto int
as
insert into GamaXProducto values (@idGama,@idProducto)
```

```
exec cargarGamaXProducto 1, 1; -- Casual - Camiseta --
exec cargarGamaXProducto 5, 2; -- Urbano - Pantalones --
exec cargarGamaXProducto 3, 3; -- Deportivo - Chaqueta --
exec cargarGamaXProducto 4, 4; -- Elegante - Vestido --
exec cargarGamaXProducto 5, 5; -- Urbano - Camisa --
exec cargarGamaXProducto 6, 6; -- Bohemia - Bufanda --
exec cargarGamaXProducto 7, 7; -- Retro - Gorro --
exec cargarGamaXProducto 8, 8; -- Moderna - Zapatillas --
exec cargarGamaXProducto 9, 9; -- Vintage - Falda --
exec cargarGamaXProducto 10, 10; -- Minimalista - Abrigo --
exec cargarGamaXProducto 11, 11; -- Colorida - Suéter --
exec cargarGamaXProducto 12, 12; -- Estacional - Sandalias --
exec cargarGamaXProducto 13, 13; -- Ecuestre - Ojotas --
exec cargarGamaXProducto 14, 14; -- Playera - Blusa --
exec cargarGamaXProducto 15, 15; -- Nocturna - Calzas --
exec cargarGamaXProducto 16, 16; -- Aventurera - Calcetines --
exec cargarGamaXProducto 17, 17; -- Confortable - Traje De Baño --
exec cargarGamaXProducto 18, 18; -- Chic - Sombrero --
exec cargarGamaXProducto 19, 19; -- Sostenible - Chaleco --
exec cargarGamaXProducto 20, 20; -- Edicion Limitada - Guantes --
```

#### Cargar DetalleDePedido

```
create procedure cargarDetalleDePedido
@idPedido int,
@idProducto int,
@cantidad int,
@subTotal int
as
insert into DetalleDePedido values (@idPedido,@idProducto,@cantidad,@subTotal)

EXEC cargarDetalleDePedido 1, 1, 5, 20000
EXEC cargarDetalleDePedido 2, 2, 3, 15000
EXEC cargarDetalleDePedido 3, 3, 7, 18000
EXEC cargarDetalleDePedido 4, 4, 2, 22000
EXEC cargarDetalleDePedido 5, 5, 4, 13000
EXEC cargarDetalleDePedido 6, 6, 6, 25000
EXEC cargarDetalleDePedido 7, 7, 8, 17000
EXEC cargarDetalleDePedido 8, 8, 1, 19000
EXEC cargarDetalleDePedido 9, 9, 3, 20000
EXEC cargarDetalleDePedido 10, 10, 5, 16000
EXEC cargarDetalleDePedido 11, 11, 2, 21000
EXEC cargarDetalleDePedido 12, 12, 4, 23000
EXEC cargarDetalleDePedido 13, 13, 6, 14000
EXEC cargarDetalleDePedido 14, 14, 8, 26000
EXEC cargarDetalleDePedido 15, 15, 1, 18000
EXEC cargarDetalleDePedido 16, 16, 3, 17000
EXEC cargarDetalleDePedido 17, 17, 5, 24000
EXEC cargarDetalleDePedido 18, 18, 2, 20000
EXEC cargarDetalleDePedido 19, 19, 4, 22000
EXEC cargarDetalleDePedido 20, 20, 6, 16000
```

#### Cargar Pago

```
create procedure cargarPago
@idCliente int,
@idFormaDePago int,
@fechaPago date,
@totalPago int
as
insert into Pago values (@idCliente,@idFormaDePago,@fechaPago,@totalPago)
```



```
exec cargarPago 1,1,'2020-10-10',20000
exec cargarPago 1,1,'2022-05-15',15000
exec cargarPago 2,3,'2021-07-02',18000
exec cargarPago 5,2,'2023-01-20',22000
exec cargarPago 10,4,'2022-11-08',13000
exec cargarPago 15,1,'2021-09-30',25000
exec cargarPago 8,3,'2023-03-12',17000
exec cargarPago 12,2,'2022-08-05',19000
exec cargarPago 3,4,'2021-12-18',20000
exec cargarPago 6,1,'2023-02-25',16000
exec cargarPago 18,2,'2022-04-03',21000
exec cargarPago 4,3,'2021-10-14',23000
exec cargarPago 11,4,'2023-05-08',14000
exec cargarPago 7,1,'2022-06-22',26000
exec cargarPago 9,2,'2021-11-11',18000
exec cargarPago 14,3,'2023-04-01',17000
exec cargarPago 13,4,'2022-09-19',24000
exec cargarPago 20,1,'2021-08-08',20000
exec cargarPago 17,2,'2023-06-05',22000
exec cargarPago 16,3,'2022-03-27',16000
exec cargarPago 19,4,'2021-12-10',19000
```

Cargar FormaDePago

```
create procedure cargarFormaDePago
@detalle varchar(50)
as
insert into FormaDePago values (@detalle)
```

```
EXEC cargarFormaDePago 'Tarjeta de Credito'
EXEC cargarFormaDePago 'Tarjeta Debito'
EXEC cargarFormaDePago 'Tranferencia'
EXEC cargarFormaDePago 'Al Contado'
```

## Modificación

### Modificar Producto

```
create procedure ModificarDetalleP  
@idProducto int,  
@descripcion varchar(250)  
as  
update Producto set descripcion = @descripcion where idProducto = @idProducto
```

### Modificar DetalleDeProducto

```
create procedure modificarPrecioP  
@idDetalleDeProducto int,  
@PrecioDeVenta int,  
@precioDeProveedor int  
as  
update DetalleDeProducto set precioDeVenta = @PrecioDeVenta where idDetalleDeProducto = @idDetalleDeProducto;  
update DetalleDeProducto set precioDeProveedor = @precioDeProveedor where idDetalleDeProducto = @idDetalleDeProducto;  
  
create procedure modificarStockP  
@idDetalleDeProducto int,  
@cantidadEnStock int  
as  
update DetalleDeProducto set cantidadEnStock = @cantidadEnStock where idDetalleDeProducto = @idDetalleDeProducto;
```

### Modificar DetalleDeCategoria

```
create procedure modificarDetallaDeCategoria  
@idCategoria int,  
@nombreCategoria varchar(25)  
as  
update DetalleDeCategoria set nombreCategoria = @nombreCategoria where idCategoria = @idCategoria
```

### Modificar Gama

```
create procedure modificarGama  
@idGama int,  
@tipoDeGama varchar(250)  
as  
update Gama set tipoGama = @tipoDeGama where idGama = @idGama
```

### Modificar gamaXProducto

```
create procedure modificarGamaXProducto  
@idGamaXProducto int,  
@idGama int,  
@idProducto int  
as  
update GamaXProducto set idGama = @idGama where idGamaXProducto = @idGamaXProducto  
update GamaXProducto set idProducto = @idProducto where idGamaXProducto = @idGamaXProducto
```

### Modificar Cliente

```
create procedure modificarContactoDeCliente  
@idCliente int,  
@telefono int,  
@mail varchar(250)  
as  
update Cliente set telefono = @telefono where idCliente = @idCliente  
update Cliente set mail = @mail where idCliente = @idCliente
```

### Modificar Atender

```
create procedure modificarClienteAtendido  
@idAtender int,  
@idCliente int  
as  
update Atender set idCliente = @idCliente where idAtender = @idAtender
```

### Modificar Empleado

```
create procedure modificarContactoEmpleado  
@idEmpleado int,  
@email varchar(250)  
as  
update Empleado set mail = @email where idEmpleado = @idEmpleado
```

### Modificar Pago

```
create procedure modificarFormaDePagoDePago  
@idPago int,  
@idFormaDePago int  
as  
update Pago set idFormaDePago = @idFormaDePago where idPago = @idPago
```

### Modificar FormaDePago

```
create procedure modificarFormaDePago  
@idFormaDePago int,  
@detalle varchar(50)  
as  
update FormaDePago set detalle = @detalle where idFormaDePago = @idFormaDePago
```

### Modificar Pedido

```
create procedure modificarEstadoDePedido  
@idPedido int,  
@estado varchar(25)  
as  
update Pedido set estado = @estado where idPedido = @idPedido
```

### Modificar DetallleDePedido

```
create procedure modificarDetalleDePedido  
@idDetallePedido int,  
@idPedido int,  
@idProducto int,  
@cantidad int  
as  
update DetalleDePedido set idPedido = @idPedido where idDetallePedido = @idDetallePedido  
update DetalleDePedido set idProducto = @idProducto where idDetallePedido = @idDetallePedido  
update DetalleDePedido set cantidad = @cantidad where idDetallePedido = @idDetallePedido
```

## Eliminación

### Eliminar Producto



```
create procedure eliminarProducto  
@idProducto int  
as  
delete from Producto where idProducto = @idProducto
```

#### Eliminar DetalleDeProducto

```
create procedure eliminarDetalleP  
@idDetalleDeProducto int  
as  
delete from DetalleDeProducto where idDetalleDeProducto = @idDetalleDeProducto
```

#### Eliminar DetalleDeCategoria

```
create procedure EliminarDetalleDeCategoria  
@idCategoria int  
as  
delete from DetalleDeCategoria where idCategoria = @idCategoria
```

#### Eliminar Gama

```
create procedure eliminarGama  
@idGama int  
as  
delete from Gama where idGama = @idGama
```

#### Eliminar gamaXProducto

```
create procedure eliminarGamaXProducto  
@idGamaXProducto int  
as  
delete from GamaXProducto where idGamaXProducto = @idGamaXProducto  
  
/*Modificar y eliminar Cliente*/
```

#### Eliminar Cliente

```
create procedure eliminarCliente  
@idCliente int  
as  
delete from Cliente where idCliente = @idCliente
```

#### Eliminar Atender

```
create procedure eliminarRegistroAtencion  
@idAtender int  
as  
delete from Atender where idAtender = @idAtender
```

#### Eliminar Empleado

```
:create procedure eliminarEmpleado  
@idEmpleado int  
as  
delete from Empleado where idEmpleado = @idEmpleado
```

Eliminar Pago

```
create procedure eliminarPago  
@idPago int  
as  
delete from Pago where idPago = @idPago
```

Eliminar FormaDePago

```
:create procedure eliminarFormaDePago  
@idFormaDePago int  
as  
delete from FormaDePago where idFormaDePago = @idFormaDePago
```

Eliminar Pedido

```
:create procedure eliminarPedido  
@idPedido int  
as  
delete from Pedido where idPedido = @idPedido
```

Eliminar DetalleDePedido

```
create procedure eliminarDetalleDePedido  
@idDetallePedido int  
as  
delete from DetalleDePedido where idDetallePedido = @idDetallePedido
```

## Consultas

### SELECT

#### 1. Mostrar Stock de Producto

```
select p.idProducto,p.descripcion,dp.tipo,dp.cantidadEnStock from Producto p
inner join DetalleDeProducto dp on p.idProducto = dp.idProducto
group by p.idProducto,p.descripcion,dp.tipo,dp.cantidadEnStock
```

idProducto	descripcion	tipo	cantidadEnStock
1	Camiseta	Jean	50
2	Pantalones	Cuero	30
3	Chaqueta	Sintético	40
4	Vestido	Tela	25
5	Camisa	Denim	60
6	Bufanda	Algodón	45
7	Gorro	Lino	35
8	Zapatillas	Poliéster	28
9	Falda	Seda	20
10	Abrigo	Tejido	50
11	Suéter	Denim	30
12	Sandalias	Algodón	40
13	Ojotas	Lino	25
14	Blusa	Sintético	60
15	Calzas	Tejido	45
16	Calcetines	Denim	35
17	Traje De ...	Poliéster	28
18	Sombrero	Algodón	20
19	Chaleco	Sintético	50
20	Guantes	Denim	30

#### 2. Mostrar los clientes que recibieron tarde su producto

```
select p.fechaEspera,p.fechaEntrega,c.idCliente,c.nombre,c.apellido1,c.mail,c.telefono
from cliente c
join Pedido p on c.idCliente = p.idCliente
where DATEDIFF(DAY,p.fechaEspera,p.fechaEntrega) >=1
```

fechaEspera	fechaEntrega	idCliente	nombre	apellido1	mail	telefono
2021-11-20	2023-12-30	1	Alan	Orlando	AOrlando@hotmail.com	1134546487

#### 3. Mostrar los productos que tienen más stock que el promedio

```
SELECT p.descripcion AS Producto, d.cantidadEnStock AS Stock
FROM Producto p
JOIN DetalleDeProducto d ON p.idProducto = d.idProducto
WHERE d.cantidadEnStock > (SELECT AVG(cantidadEnStock) FROM DetalleDeProducto);
```

Producto	Stock
Camiseta	50
Chaqueta	40
Camisa	60
Bufanda	45
Abrigo	50
Sandalias	40
Blusa	60
Calzas	45
Chaleco	50

#### 4. Mostrar el Total de compras de cada cliente

```
SELECT c.idCliente,
       c.nombre AS nombreCliente,
       COUNT(p.idPedido) AS TotalCompras
FROM Cliente c
LEFT JOIN Pedido p ON c.idCliente = p.idCliente
GROUP BY c.idCliente, c.nombre;
```

idCliente	nombreCliente	TotalCompras
1	Alan	2
2	Jazmin	1
3	Oscar	1
4	Micaela	1
5	Florencia	1
6	Roman	1
7	Ana	1
8	Pedro	1
9	María	1
10	Juan	1

#### 5. Mostrar los Productos más vendidos

```
SELECT p.idProducto, p.descripcion, SUM(dp.cantidad) AS total_vendido
FROM Producto p
JOIN DetalleDePedido dp ON p.idProducto = dp.idProducto
GROUP BY p.idProducto, p.descripcion
ORDER BY total_vendido DESC;
```

idProducto	descripcion	total_vendido
7	Gorro	8
14	Blusa	8
3	Chaqueta	7
6	Bufanda	6
13	Ojotas	6
20	Guantes	6
17	Traje De Baño	5
10	Abrigo	5
1	Camiseta	5
5	Camisa	4
12	Sandalias	4
19	Chaleco	4

## Procedimientos

6. Mostrar los pagos que se hicieron dado un año y un mes específico

```
create procedure consultarPago
@añO int,
@mes int
as
select *from Pago
where YEAR(fechaPago) = @añO and MONTH(fechaPago) = @mes
exec consultarPago '2023','06'
```

idPago	idCliente	idFormaDePago	fechaPago	totalPago
19	17	2	2023-06-05	22000

7. Mostrar la cantidad de pedidos que se hicieron dado un año pasado por parámetros

```
create procedure CantidadDePedidosEnElAño
@añO varchar(25)
as
select count(*) as cantidadDePedidos from Pedido
group by year(fechaPedido)
having year(fechaPedido) = @añO
exec CantidadDePedidosEnElAño '2022'
```

cantidadDePedidos
7

8. Mostrar el nombre, apellido1, DNI de los clientes que fueron atendidos por el nombre del empleado pasado por parámetros

```
create procedure aQuienAtendio
@nombre varchar(250),
@apellido1 varchar(250)
as
select e.idEmpleado, c.nombre as NombreCliente, c.apellido1 as Apellido1, c.dni as DNI
from Cliente c
inner join Atender a on c.idCliente = a.idCliente
inner join Empleado e on a.idEmpleado = e.idEmpleado
where e.nombre = @nombre and e.apellido1 = @apellido1
exec aQuienAtendio 'Juan','Pérez'
```

idEmpleado	NombreCliente	Apellido1	DNI
1	Alan	Orlando	45789878

9. Mostrar detalle de producto y cantidad de ventas tiene por mes

```
CREATE PROCEDURE ObtenerVentasPorMes
@mes_ingresar int
as
SELECT descripcion, SUM(cantidad) AS total_vendido
FROM DetalleDeProducto p
JOIN Producto produ ON p.idProducto = produ.idProducto
JOIN DetalleDePedido dp ON produ.idProducto = dp.idProducto
JOIN Pedido pedido ON dp.idPedido = pedido.idPedido
WHERE MONTH(fechaPedido) = @mes_ingresar
GROUP BY descripcion;
exec ObtenerVentasPorMes 5
```

descripcion	total_vendido
Ojotas	6
Pantalones	3

10. Obtener la venta total de todos los productos mas los ingresos totales de todas las ventas

```
CREATE PROCEDURE ObtenerTotalIngresoPorMes
@mes_ingresar int
as
SELECT MONTH(fechaPedido) AS Mes,
       SUM(dp.cantidad) AS TotalProductosVendidos,
       SUM(dp.cantidad * dp.subTotal) AS IngresoTotal
FROM DetalleDePedido dp
JOIN Pedido pedido ON dp.idPedido = pedido.idPedido
JOIN Producto produ ON dp.idProducto = produ.idProducto
WHERE MONTH(fechaPedido) = @mes_ingresar
GROUP BY MONTH(fechaPedido);

exec ObtenerTotalIngresoPorMes 3
```

Mes	TotalProductosVendidos	IngresoTotal
3	14	232000

11. Obtener el empleado que más ventas tuvo por mes

```
CREATE PROCEDURE ObtenerEmpleadoConMasVentas
@mes_ingresar int
as
SELECT TOP 1
       e.idEmpleado,
       e.nombre AS NombreEmpleado,e.apellido1 as Apellido,
       SUM(dp.cantidad) AS TotalVendido
FROM Empleado e
JOIN Atender a ON e.idEmpleado = a.idEmpleado
JOIN Cliente c ON a.idCliente = c.idCliente
JOIN Pedido pedido ON c.idCliente = pedido.idCliente
JOIN DetalleDePedido dp ON pedido.idPedido = dp.idPedido
WHERE MONTH(pedido.fechaPedido) = @mes_ingresar
GROUP BY e.idEmpleado, e.nombre,e.apellido1
ORDER BY TotalVendido DESC;

exec ObtenerEmpleadoConMasVentas 7
```

idEmpleado	NombreEmpleado	Apellido	TotalVendido
11	Javier	Gómez	7

12. Calcular el profit por producto y año

```
CREATE PROCEDURE ObtenerProfitPorAnio
@anio_ingresar int
as
SELECT
       p.idProducto,
       p.Descripcion AS Producto,
       YEAR(pedido.fechaPedido) AS Anio,
       SUM(dp.cantidad) AS TotalVendido,
       SUM(dp.PrecioDeVenta - dp.PrecioDeProveedor) AS Profit,
       SUM((dp.PrecioDeVenta - dp.PrecioDeProveedor) * dp.cantidad) AS TotalProfitPorCantidad
FROM DetalleDeProducto dp
JOIN Producto p ON dp.idProducto = p.idProducto
JOIN DetalleDePedido dped ON p.idProducto = dped.idProducto
JOIN Pedido pedido ON dped.idPedido = pedido.idPedido
WHERE YEAR(pedido.fechaPedido) = @anio_ingresar
GROUP BY p.idProducto, p.Descripcion, YEAR(pedido.fechaPedido);

exec ObtenerProfitPorAnio 2021
```

idProducto	Producto	Anio	TotalVendido	Profit	TotalProfitPorCantidad
3	Chaqueta	2021	7	6000	42000
6	Bufanda	2021	6	7000	42000
9	Falda	2021	3	8000	24000
12	Sandalias	2021	4	6000	24000
15	Calzas	2021	1	7000	7000
18	Sombrero	2021	2	8000	16000

13. Obtener el método de pago más utilizado por mes pasado por parámetro

```
CREATE PROCEDURE ObtenerMetodoPagoMasUtilizadoPorMes
@mes_ingresar int
AS
SELECT TOP 1 fechaPedido AS Mes,
MAX(detalle) AS MetodoPagoMasUtilizado,
COUNT(*) AS CantidadPedidos
FROM Pedido
JOIN Cliente c ON pedido.idCliente = c.idCliente
JOIN Pago p ON c.idCliente = p.idCliente
JOIN FormaDePago f ON p.idFormaDePago = f.idFormaDePago
WHERE MONTH(fechaPedido) = @mes_ingresar
GROUP BY fechaPedido, detalle
ORDER BY CantidadPedidos DESC;

EXEC ObtenerMetodoPagoMasUtilizadoPorMes 5
```

Mes	MetodoPagoMasUtilizado	CantidadPedidos
2022-05-15	Tarjeta de Credito	2

14. Muestra el total de pedidos en el estado ingresado

```
CREATE PROCEDURE EstadoTotalPedidos
@estado_pedido varchar(30)
as
SELECT p.estado,
dp.idPedido AS idPedido,
prod.descripcion AS descripcion
FROM Pedido p
JOIN DetalleDePedido dp ON p.idPedido = dp.idPedido
JOIN Producto prod ON dp.idProducto = prod.idProducto
WHERE p.estado = @estado_pedido
GROUP BY p.estado, dp.idPedido, prod.descripcion;

exec EstadoTotalPedidos 'espera'
```

estado	idPedido	descripcion
Espera	4	Vestido
Espera	6	Bufanda
Espera	8	Zapatillas
Espera	10	Abrigo
Espera	13	Ojotas
Espera	15	Calzas
Espera	17	Traje De Baño
Espera	19	Chaleco

15. Verificar si en cliente ha realizado Pagos y si no es así mostrar el mensaje “no hay pagos”.

```
CREATE PROCEDURE VerificarPagosCliente
@cliente_id INT
AS
SELECT idPago, totalPago
FROM Pago
WHERE idCliente = @cliente_id;

SELECT 'No hay pagos' AS mensaje
WHERE NOT EXISTS (SELECT 1 FROM Pago WHERE idCliente = @cliente_id);

exec VerificarPagosCliente 23
```

idPago | totalPago

mensaje
No hay pagos

Vistas

16. Mostrar los pagos disponibles en este momento

```
create view formaDePagoDisponible
as
select *from FormaDePago

select*from formaDePagoDisponible
```

idFormaDePago	detalle
1	Tarjeta de Credito
2	Tarjeta Debito
3	Tranferencia
4	Al Contado

## 17. mostrar los clientes atendidos junto con los empleados asociados

```
CREATE VIEW ClientesAtendidos
AS
SELECT c.nombre AS Nombre_Cliente, c.apellido1 AS Apellido_Cliente, c.telefono AS Telefono_Cliente,
c.mail AS Email_Cliente, e.nombre AS Nombre_Empleado, e.apellido1 AS Apellido_Empleado, e.mail AS Email_Empleado FROM Cliente c
INNER JOIN Atender a ON c.idCliente = a.idAtender
INNER JOIN Empleado e ON a.idEmpleado = e.idEmpleado;

SELECT *FROM ClientesAtendidos
```

Nombre_Cliente	Apellido_Cliente	Telefono_Cliente	Email_Cliente	Nombre_Empleado	Apellido_Empleado	Email_Empleado
Alan	Orlando	1134546487	AOrlando@hotmail.com	Juan	Pérez	juan.perez@weColver.com.ar
Jazmin	Martin	1198348754	JazminS12@gmail.com	Maria	González	maria.gonzalez@weColver.com.ar
Oscar	villalba	1122460967	Oscar_villalba@yahoo.com	Carlos	Martínez	carlos.martinez@weColver.com.ar
Micaela	Fuentes	1189369867	MFuentes@gmail.com	Laura	López	laura.lopez@weColver.com.ar
Florencia	Borda	1134879809	Flor_servi@BuenosAires.ed	Pedro	Sánchez	pedro.sanchez@weColver.com.ar
Roman	Stella	1122983409	RomanStel@gmail.com	Ana	Ramírez	ana.ramirez@weColver.com.ar
Ana	Garcia	1122334455	ana.garcia@hotmail.com	Luis	Díaz	luis.diaz@weColver.com.ar
Pedro	Santos	1144332211	pedro.santos@hotmail.com	Elena	Fernández	elena.fernandez@weColver.co...
Maria	López	1577889900	maria.lopez@yahoo.com	Miguel	Moreno	miguel.moreno@weColver.com.ar
Juan	Martínez	1188990011	juan.martinez@gamil.com	Carmen	Ruiz	carmen.ruiz@weColver.com.ar
Luis	Oliveira	1500112233	luis.oliveira@hotmail.com	Javier	Gómez	javier.gomez@weColver.com.ar
Sofía	Pérez	1122334455	sofia.perez@gmail.com	Rosa	Martín	rosa.martin@weColver.com.ar
Miguel	Fernández	1144332211	miguel.fernandez@hotmail.	Alejandro	Torres	alejandro.torres@weColver.com...
Isabel	Silva	1577889900	isabel.silva@hotmail.com	Isabel	Ortega	isabel.ortega@weColver.com.ar
Eduardo	Rodríguez	1588990011	eduardo.rodriguez@hotmail	Sergio	Hernández	sergio.hernandez@weColver.co...
Carla	Gómez	1500112233	carla.gomez@gmail.com	Lucía	Morales	lucia.morales@weColver.com.ar
Antonio	Santos	1122334455	antonio.santos@gmail.com	Pablo	Vargas	pablo.vargas@weColver.com.ar
Rosa	Pérez	1144332211	rosa.perez@gmail.com	Eva	Jiménez	eva.jimenez@weColver.com.ar
Fernando	Martínez	1177889909	fernando.martinez@gmail.c	Antonio	López	antonio.lopez@weColver.com.ar
Clara	Oliveira	1188990011	clara.oliveira@gmail.com	Beatriz	García	beatriz.garcia@weColver.com.ar

## 18. crear una vista que liste los productos junto con la gama del mismo

```
create view ProductosDelSistema
as
select g.idGama,g.tipoGama,p.idProducto,p.descripcion from Producto p
inner join GamaXProducto gXP on p.idProducto = gXP.idProducto
inner join Gama g on gXP.idGama = g.idGama

select *from ProductosDelSistema
```

idGama	tipoGama	idProducto	descripcion
1	Casual	1	Camiseta
5	Urbano	2	Pantalones
3	Deportivo	3	Chaqueta
4	Elegante	4	Vestido
5	Urbano	5	Camisa
6	Bohemia	6	Bufanda
7	Retro	7	Gorro
8	Moderna	8	Zapatillas
9	Vintage	9	Falda
10	Minimali...	10	Abrigo
11	Colorida	11	Suéter
12	Estacio...	12	Sandalias
13	Ecuestre	13	Ojotas
14	Playera	14	Blusa
15	Nocturna	15	Calzas
16	Aventur...	16	Calcetines
17	Confort...	17	Traje De ...
18	Chic	18	Sombrero
19	Sosteni...	19	Chaleco
20	Edicion ...	20	Guantes



19. Mostrar los pagos que se hicieron en el día junto con los clientes y su forma de pago

```
create view pagosDelDia
as
select c.nombre,c.apellido1 as primerApellido,c.telefono,c.mail,p.idPago,p.fechaPago,fp.detalle as formaDePago,p.totalPago from Pago p
inner join FormaDePago fp on p.idFormaDePago = fp.idFormaDePago
inner join Cliente c on p.idCliente = c.idCliente
where p.fechaPago = CAST(GETDATE() AS DATE)

select *from pagosDelDia
```

nombre	primerApellido	telefono	mail	idPago	fechaPago	formaDePago	totalPago
Fernando	Martínez	1177889909	fernando.martinez@gmail.c	22	2023-11-20	Al Contado	19000

20. Mostrar el mail de contacto de un empleado con el nombre y apellido pasado por parámetros

```
-- generar un procedimiento mostrando el mail de contacto
create procedure mostrarContactoEmpleado
@nombre varchar(25),
@apellido varchar(25)
as
select idEmpleado, mail from Empleado
where nombre = @nombre and apellido1 = @apellido

exec mostrarContactoEmpleado 'Juan','Pérez'
```

idEmpleado	mail
1	juan.perez@weColver.com.ar

## Conclusión

En el desarrollo de la propuesta para el sistema de gestión de venta de WeClover, se ha delineado una estructura robusta que aborda los requisitos fundamentales de la empresa en términos de administración de clientes, pedidos, pagos, empleados y productos. La implementación de una Base de Datos bien diseñada proporcionará la base necesaria para alcanzar los objetivos comerciales y mejorar la eficiencia operativa.

La arquitectura propuesta no solo cumple con los requisitos funcionales específicos, como el registro detallado de clientes, la gestión eficiente de pedidos y la asociación adecuada de pagos, sino que también aborda los requisitos no funcionales, como la seguridad de datos, la escalabilidad y la usabilidad del sistema. Se ha prestado especial atención a la integridad de los datos, asegurando la consistencia y precisión de la información almacenada.

En conclusión, la implementación exitosa de este sistema no solo optimizará los procesos internos de WeClover, sino que también sentará las bases para futuras adaptaciones y mejoras. La capacidad de consulta de información específica, como clientes atendidos, pagos por año y pedidos entregados, proporcionará a la empresa herramientas valiosas para la toma de decisiones informadas.

La colaboración del equipo en el proyecto WeClover fue muy dinámica y activa. Utilizamos WhatsApp para una comunicación ágil y Trello como herramienta principal para organizar nuestras tareas. Siguiendo la metodología Scrum. WhatsApp fue clave para una comunicación inmediata, permitiéndonos resolver problemas eficientemente y compartir ideas. Trello ofreció una visión clara del progreso y se convirtió en el espacio colaborativo para desarrollar consultas y documentación. Concluyendo que la combinación de estas herramientas y enfoques ágiles nos brindó valiosos saberes para futuras colaboraciones en equipo.

Con este proyecto, pudimos poner en práctica todo lo visto en clases y de esa manera plasmarlo en el respectivo trabajo, esta práctica nos permitió, trabajar de manera colaborativa poniendo en práctica las metodologías ágiles y ver de primera mano los beneficios de la misma, nos permitió potenciar nuestra habilidad de comunicación, liderazgo y de debate que nos servirán en un futuro adentrarnos en el ámbito laboral.