

# PROYECTO TIENDA ONLINE DE ROPA DEPORTIVA

*Santiago Gomez*

*Camilo José Delgado*

*Juan Carlos Quintero*

*Jhonatan Morales*

*Carol Varela*



# CONTENIDO

01

---

Contexto

02

---

Modelo Entidad -  
Relación

03

---

Modelo relacional

04

---

Creación DB y  
Tablas

05

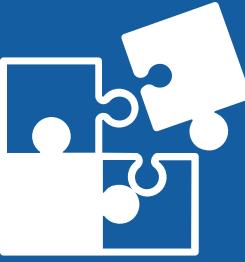
---

Agregando registros

06

---

Consultas SQL

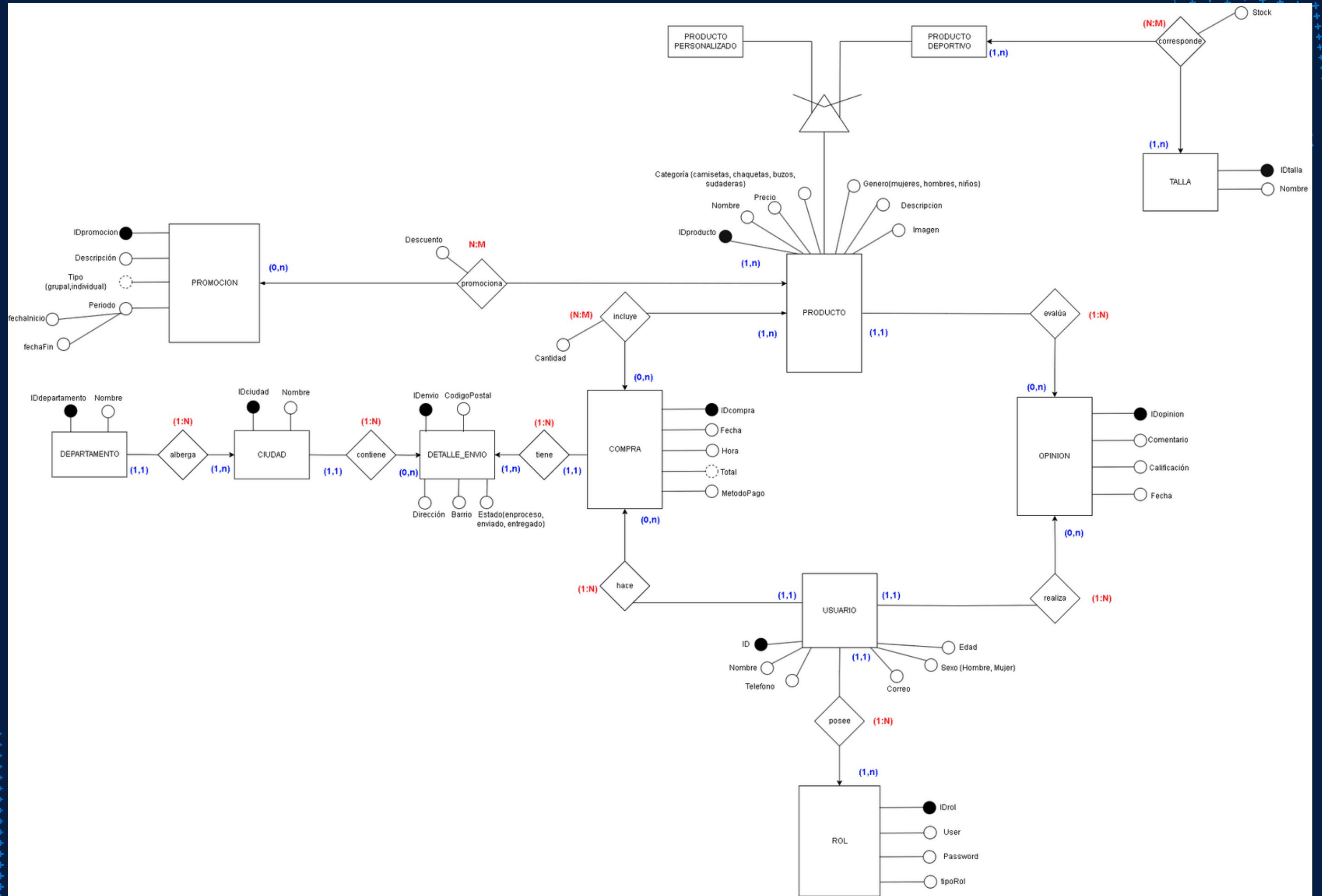


# CONTEXTO

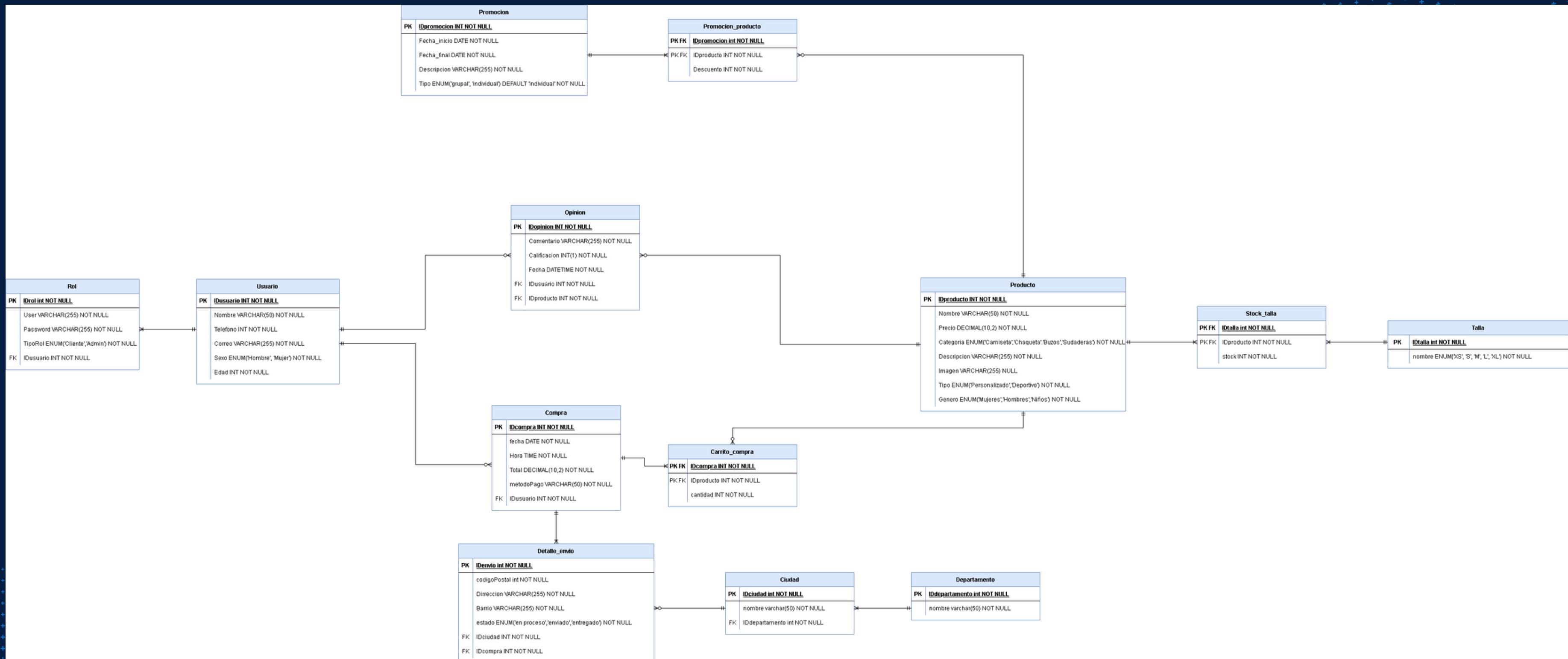
El proyecto se centra en el diseño e implementación de un sistema de almacenamiento para respaldar una tienda online especializada en la venta y personalización de ropa deportiva. La tienda busca ofrecer una experiencia fluida y escalable tanto para administradores como para clientes. Para lograrlo, se requiere un sistema de almacenamiento que permita gestionar eficientemente la amplia gama de productos disponibles. Esto incluye la organización de productos por categoría y la gestión de diseños para personalización. Además, se destaca la importancia de la flexibilidad en la gestión de promociones temporales para promover ventas y atraer clientes.



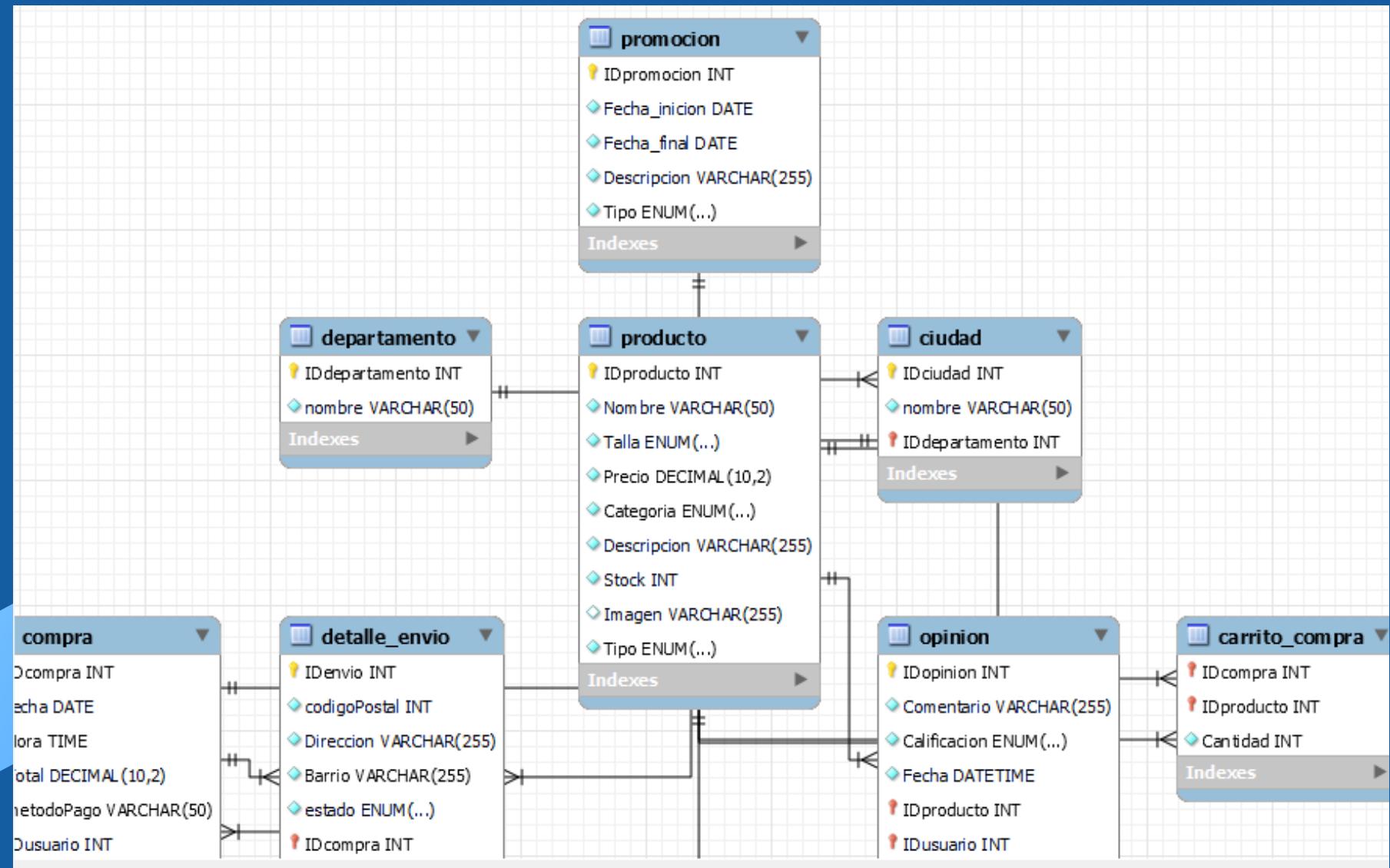
# Modelo Entidad – Relación



# Modelo Relacional



# CREACIÓN BD



# CREACIÓN BD

```
-- MySQL Script generated by MySQL Workbench
-- Sun Mar  3 23:37:03 2024
-- Model: New Model  Version: 1.0
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- Schema PROYECTO
```

```
-- Schema PROYECTO
```

```
CREATE SCHEMA IF NOT EXISTS `PROYECTO` DEFAULT CHARACTER SET
utf8 ;
USE `PROYECTO` ;
```

# CREACIÓN TABLAS

```
-- Schema PROYECTO
-----
CREATE SCHEMA IF NOT EXISTS `PROYECTO` DEFAULT CHARACTER SET utf8 ;
USE `PROYECTO` ;
```

```
-- Table `PROYECTO`.`Usuario`
-----
CREATE TABLE IF NOT EXISTS `PROYECTO`.`Usuario` (
  `IDusuario` INT NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(50) NOT NULL,
  `Telefono` INT NOT NULL,
  `Correo` VARCHAR(255) NOT NULL,
  `Sexo` ENUM('Hombre', 'Mujer') NULL,
  `Edad` INT NULL,
  PRIMARY KEY (`IDusuario`)
)
ENGINE = InnoDB;
```

```
4
5  -- Table `PROYECTO`.`Rol`
6
7 • CREATE TABLE IF NOT EXISTS `PROYECTO`.`Rol` (
8   `IDrol` INT NOT NULL AUTO_INCREMENT,
9   `User` VARCHAR(255) NOT NULL,
0   `Password` VARCHAR(255) NOT NULL,
1   `TipoRol` ENUM('Cliente', 'Admin') NOT NULL,
2   `IDusuario` INT NOT NULL,
3   PRIMARY KEY (`IDrol`, `IDusuario`),
4   INDEX `fk_Rol_Usuario1_idx` (`IDusuario` ASC),
5   CONSTRAINT `fk_Rol_Usuario1`
6     FOREIGN KEY (`IDusuario`)
7     REFERENCES `PROYECTO`.`Usuario` (`IDusuario`)
8     ON DELETE NO ACTION
9     ON UPDATE NO ACTION)
0   ENGINE = InnoDB;
```

```
-- Table `PROYECTO`.`Producto`
-----
• CREATE TABLE IF NOT EXISTS `PROYECTO`.`Producto` (
  `IDproducto` INT NOT NULL AUTO_INCREMENT,
  `Nombre` VARCHAR(50) NOT NULL,
  `Talla` ENUM('XS', 'S', 'M', 'L', 'XL') NOT NULL,
  `Precio` DECIMAL(10,2) NOT NULL,
  `Categoria` ENUM('Camiseta', 'Chaqueta', 'Buzos', 'Sudaderas') NOT NULL,
  `Descripcion` VARCHAR(255) NOT NULL,
  `Stock` INT NOT NULL,
  `Imagen` VARCHAR(255) NULL,
  `Tipo` ENUM('Personalizado', 'Deportivo') NOT NULL,
  `Genero` ENUM('Mujeres', 'Hombres', 'Niños') NOT NULL,
  PRIMARY KEY (`IDproducto`)
)
ENGINE = InnoDB;
```

# AGREGANDO REGISTROS

```
-- Registros para la tabla Usuario
INSERT INTO PROYECTO.Usuario (Nombre, Telefono, Correo, Sexo, Edad) VALUES
('Juan Perez', 123456789, 'juan@email.com', 'Hombre', 25),
('Maria Rodriguez', 987654321, 'maria@email.com', 'Mujer', 30),
('Luisa Martinez', 555123789, 'luisa@email.com', 'Mujer', 28),
('Pedro Gomez', 789456123, 'pedro@email.com', 'Hombre', 35),
('Ana Ramirez', 321654987, 'ana@email.com', 'Mujer', 22),
('Carlos Herrera', 456789321, 'carlos@email.com', 'Hombre', 40),
('Laura Torres', 987123654, 'laura@email.com', 'Mujer', 31),
('Javier Lopez', 321789654, 'javier@email.com', 'Hombre', 27),
('Isabel Garcia', 654321987, 'isabel@email.com', 'Mujer', 33),
('Gabriel Rodriguez', 123987456, 'gabriel@email.com', 'Hombre', 29);
```

```
-- Registros para la tabla Rol
INSERT INTO PROYECTO.Rol (User, Password, TipoRol, IDusuario) VALUES
('juan_usuario', 'password123', 'Cliente', 1),
('admin_usuario', 'adminpass', 'Admin', 1),
('luisa_usuario', 'luisapass', 'Cliente', 3),
('pedro_admin', 'admin123', 'Admin', 4),
('ana_usuario', 'anapass', 'Cliente', 5),
('carlos_admin', 'admin456', 'Admin', 6),
('laura_usuario', 'laurapass', 'Cliente', 7),
('javier_admin', 'admin789', 'Admin', 8),
('isabel_usuario', 'isabelpass', 'Cliente', 9),
('gabriel_admin', 'admin987', 'Admin', 10),
('laura_admin', 'lauraadmin', 'Admin', 7);
```

```
-- Registros para la tabla Promocion
INSERT INTO PROYECTO.Promocion (Fecha_inicio, Fecha_final, Descripcion, Tipo) VALUES
('2024-03-01', '2024-03-15', 'Descuento de primavera', 'grupal'),
('2024-04-01', '2024-04-10', 'Oferta individual', 'individual'),
('2024-04-15', '2024-04-30', 'Descuento de primavera', 'grupal'),
('2024-05-01', '2024-05-10', 'Oferta individual', 'individual'),
('2024-06-01', '2024-06-15', 'Verano especial', 'grupal');
```

# CONSULTAS SQL

## Usuario con más compras y cuánto dinero ha gastado:

```
SELECT u.nombre, count(cm.IDcompra) as numCompras, sum(cm.Total) as
dineroGastado
FROM usuario as u JOIN compra as cm USING (IDusuario)
GROUP BY cm.IDusuario
HAVING count(cm.IDusuario) = (SELECT max(numeroCompras)
FROM (SELECT cm.IDusuario, count(cm.idCompra) as numeroCompras
FROM compra as cm
GROUP BY cm.IDusuario) as usuarioMayorCompras);
```

## Calificación por producto:

```
SELECT p.nombre AS producto, AVG(o.Calificacion) AS
Promedio_Calificacion
FROM producto as p JOIN opinion as o USING (IDproducto)
GROUP BY p.Nombre;
```

Con los registros actuales:

nombre	numCompras	dineroGastado
Juan Perez	1	39.98
Maria Rodriguez	1	59.99
Luisa Martinez	1	39.99

Agregando una compra más:

nombre	numCompras	dineroGastado
Juan Perez	2	59.97

producto	Promedio_Calificacion
Camiseta Azul	5
Chaqueta Deportiva	4
Buzo Gris	4
Sudadera Roja	5
Camiseta Blanca	3.5
Chaqueta Negra	5
Buzo Rosa	4
Sudadera Azul	5
Chaqueta Verde	2
Camiseta Amarilla	5

# CONSULTAS SQL

Que persona ha hecho más reseñas y cual es la calificación promedio que ha puesto.

```
SELECT usuario.Nombre AS nombre_usuario, COUNT(opinion.IDopinion)
AS cantidad_opiniones, AVG(opinion.Calificacion) AS
promedio_calificacion
FROM usuario
JOIN opinion USING (idusuario)
GROUP BY usuario.Nombre
HAVING cantidad_opiniones = (SELECT MAX(cantidad_opiniones) FROM
(SELECT usuario.Nombre, COUNT(opinion.IDopinion) AS
cantidad_opiniones FROM usuario
JOIN opinion USING (idusuario)
GROUP BY usuario.Nombre) AS maxima_cantidad);
```

	nombre_usuario	cantidad_opiniones	promedio_calificacion
▶	Maria Rodriguez	4	3.5

# CONSULTAS SQL

## Producto con más promociones y el promedio de las promociones:

```
SELECT
    pp.IDproducto,
    COUNT(pp.IDpromocion) AS TotalPromociones,
    AVG(pp.Descuento) AS PromedioDescuento
FROM
    Promocion_producto pp
GROUP BY
    pp.IDproducto
HAVING
    COUNT(pp.IDpromocion) = (
        SELECT MAX(TotalPromociones) FROM (
            SELECT
                IDproducto,
                COUNT(IDpromocion) AS TotalPromociones
            FROM
                Promocion_producto
            GROUP BY
                IDproducto
        ) AS SubConsulta
    );
```

IDproducto	TotalPromociones	PromedioDescuento
1	1	10.0000
2	1	15.0000
3	1	12.0000
4	1	18.0000
5	1	10.0000
6	1	15.0000
7	1	20.0000
8	1	25.0000
9	1	15.0000
10	1	30.0000

# CONSULTAS SQL

## Usuario con más de una Cuenta

```
SELECT usuario.nombre AS nombre  
FROM usuario  
JOIN rol  
USING (idusuario)  
GROUP BY nombre  
HAVING COUNT(idusuario) > 1;
```

nombre
Juan Perez
Laura Torres

## Hombres por encima de 30 años

```
SELECT IDusuario, Nombre, Edad  
FROM PROYECTO.Usuario  
WHERE Sexo = 'Hombre' AND Edad > 30;
```

	IDusuario	Nombre	Edad
▶	4	Pedro Gomez	35
	6	Carlos Herrera	40

## Productos y sus respectivas tallas sin stock/agotado:

```
SELECT p.nombre AS Producto, t.nombre AS Talla  
FROM producto AS p  
JOIN stock_talla AS stck USING (idProducto)  
JOIN talla AS t USING (idTalla)  
WHERE stck.stock = 0;
```

## Personas que no han comprado ningún producto

```
SELECT u.nombre AS Usuario  
FROM usuario AS u  
LEFT JOIN compra AS c ON u.IDusuario = c.IDusuario  
WHERE c.IDcompra IS NULL;
```

Usuario
Pedro Gomez
Ana Ramirez
Carlos Herrera
Laura Torres
Javier Lopez
Isabel Garcia
Gabriel Rodriguez

# CONSULTAS SQL

## Nombre del producto más vendido y su respectivo precio

```
SELECT producto.Nombre, producto.Precio,  
SUM(carrito_compra.Cantidad) AS cantidad_vendida  
FROM carrito_compra  
JOIN producto USING (idproducto)  
  
GROUP BY carrito_compra.idproducto, producto.Nombre,  
producto.Precio  
HAVING SUM(carrito_compra.Cantidad) =  
    (SELECT MAX(suma_cantidad)  
     FROM (  
         SELECT carrito_compra.idproducto,  
         SUM(carrito_compra.Cantidad) AS suma_cantidad  
         FROM carrito_compra  
         GROUP BY carrito_compra.idproducto  
     ) AS maximo_producto  
);
```

Nombre	Precio	cantidad_vendida
Camiseta Azul	19.99	2

## Nombre del producto que más ingresos ha generado

```
SELECT Producto.Nombre, SUM(Carrito_compra.Cantidad) AS  
UnidadesVendidas, SUM(Producto.Precio * Carrito_compra.Cantidad) AS  
IngresosTotales  
FROM Carrito_compra  
INNER JOIN Producto ON Carrito_compra.IDproducto =  
Producto.IDproducto  
GROUP BY Carrito_compra.IDproducto  
ORDER BY IngresosTotales DESC;
```

Nombre	UnidadesVendidas	IngresosTotales
Chaqueta Deportiva	1	59.99
Buzo Gris	1	39.99
Camiseta Azul	2	39.98

# CONSULTAS SQL

## Cantidad de productos vendido por genero

```
SELECT producto.Genero ,  
sum(carrito_compra.cantidad) as  
cantidad_ventas from producto join  
carrito_compra using  
(idproducto) group by producto.Genero having  
cantidad_ventas = (SELECT max(cantidad_ventas)  
from (SELECT producto.Genero ,  
sum(carrito_compra.cantidad) as  
cantidad_ventas from producto join  
carrito_compra using  
(idproducto) group by producto.Genero) as maximo);
```

	Genero	cantidad_ventas
▶	Hombres	4

# CONSULTAS SQL

## Ciudad en mas detalles de envio

```
SELECT Ciudad.nombre, COUNT(Detalle_envio.IDciudad) AS  
NumeroEnvios  
FROM Detalle_envio  
JOIN Ciudad ON Detalle_envio.IDciudad = Ciudad.IDciudad  
GROUP BY Ciudad.nombre having NumeroEnvios = (select  
max(NumeroEnvios) from (SELECT Ciudad.nombre,  
COUNT(Detalle_envio.IDciudad) AS NumeroEnvios  
FROM Detalle_envio  
JOIN Ciudad ON Detalle_envio.IDciudad = Ciudad.IDciudad  
GROUP BY Ciudad.nombre) as maximo_envios);
```

	nombre	NumeroEnvios
▶	Cali	1
	Bogota	1
	Medellin	1