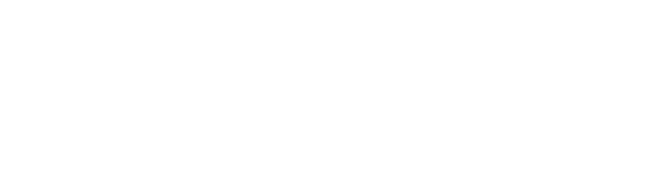


**LABORATORIO DE COMPUTACIÓN II**

Trabajo Práctico Integrador

1W3 - Grupo Nº6



113227 - Ros, Natalia

112791 - Ulloa, Mariano

112710 - Uriza, Valentina

112809 - Urquiza, Juan

112728 - Verdugo, Carlos

Fecha de entrega: 24/10/2021

**Índice**

[Diagrama del Modelo Relacional de la Base de Datos 2](#_Toc86000527)

[Reportes 3](#_Toc86000528)

# Diagrama del Modelo Relacional de la Base de Datos



# Reportes

--1) Crear una vista que permita ver los vehiculos y las autopartes.

CREATE VIEW V\_VEHICULOS

AS

SELECT p.descripcion 'Vehiculo', p.precio\_unitario 'Precio', p.stock 'Stock',

m.marca 'Marca', mo.modelo 'Modelo', s.segmento 'Segmento', tc.tipo\_combustible 'Tipo combustible',

tt.tipo\_transmision 'Tipo transmision'

FROM PRODUCTOS p, DETALLES\_PRODUCTO dp, MARCAS m, MODELOS mo, SEGMENTOS s,

TIPOS\_COMBUSTIBLE tc, TIPOS\_TRANSMISION tt

WHERE p.id\_detalle\_producto = dp.id\_detalle\_producto AND dp.id\_marca = m.id\_marca

AND dp.id\_modelo = mo.id\_modelo AND dp.id\_segmento = s.id\_segmento AND dp.id\_combustible = tc.id\_combustible

AND dp.id\_tipo\_transmision = tt.id\_tipo\_transmision

CREATE VIEW VIS\_AUTOPARTES

AS

SELECT p.descripcion 'Autopartes', p.precio\_unitario 'Precio', p.stock 'Stock',

m.marca 'Marca', mo.modelo 'Modelo'

FROM PRODUCTOS p, DETALLES\_PRODUCTO dp, MARCAS m, MODELOS mo

WHERE p.id\_detalle\_producto = dp.id\_detalle\_producto AND dp.id\_marca = m.id\_marca

AND dp.id\_modelo = mo.id\_modelo

--2) Crear un SP que, al cargar una factura en caso de no contar con stock, generar la oren de pedido. En caso de que no se pueda efectuar ninguna de las operaciones informar a modo de error.

CREATE PROC SP\_Generar\_Pedido

@id\_producto int,

@cantidad int,

@id\_cliente int,

@id\_tipo\_factura int,

@id\_tipo\_responsable int,

@descuento int,

@precio money,

@id\_forma\_pago int

AS

BEGIN

declare @stock int

declare @nro\_factura int

select @stock=stock from PRODUCTOS

SET @nro\_factura = (select ISNULL(MAX(nro\_factura), 0)+1 FROM FACTURAS)

begin try

if (@stock >= @cantidad)

begin

insert into FACTURAS (id\_tipo\_factura, id\_cliente, id\_tipo\_responsable, id\_forma\_pago, fecha)

VALUES (@id\_tipo\_factura, @id\_cliente, @id\_tipo\_responsable, @id\_forma\_pago, GETDATE())

insert INTO DETALLES\_FACTURA (nro\_factura, id\_producto, precio, cantidad, descuento)

VALUES (@nro\_factura, @id\_producto, @precio, @cantidad, @descuento)

end

else

insert into ORDENES\_PEDIDO (id\_producto, id\_cliente, cantidad, fecha\_pedido, fecha\_entrega)

VALUES (@id\_producto, @id\_cliente, @cantidad, GETDATE(), DATEADD(WEEK,3,GETDATE()))

end try

begin catch

select @@ERROR

select ERROR\_MESSAGE() 'Mensaje', ERROR\_SEVERITY() 'Gravedad del error', ERROR\_NUMBER() 'N° del error'

end catch

END

--3) Crear un SP que muetre los totales de facturación y promedio de facturación periódicos (mensual, trimestral, semestral o anual)

CREATE PROC SP\_TOTALES\_FACTURACION

@periodo varchar(10)

AS

BEGIN

if @periodo = 'mensual'

begin

SELECT sum(precio\*cantidad) 'Total Facturado',

SUM(precio\*cantidad) / COUNT(DISTINCT df.nro\_factura) 'Promedio facturación'

from facturas f, detalles\_factura df

where f.nro\_factura = df.nro\_factura

AND MONTH(f.fecha) = MONTH(GETDATE())

end

if @periodo = 'trimestral'

begin

select sum(precio\*cantidad) 'Total Facturado',

SUM(precio\*cantidad) / COUNT(DISTINCT df.nro\_factura) 'Promedio facturación'

from facturas f, detalles\_factura df

where f.nro\_factura=df.nro\_factura

AND MONTH(f.fecha) BETWEEN MONTH(GETDATE()) AND DATEADD(MONTH, -3, GETDATE())

end

if @periodo = 'semestral'

begin

select sum(precio\*cantidad) 'Total Facturado',

SUM(precio\*cantidad) / COUNT(DISTINCT df.nro\_factura) 'Promedio facturación'

from facturas f, detalles\_factura df

where f.nro\_factura=df.nro\_factura

AND MONTH(f.fecha) BETWEEN MONTH(GETDATE()) AND DATEADD(MONTH, -6, GETDATE())

end

if @periodo = 'anual'

begin

select sum(precio\*cantidad) 'Total Facturado',

SUM(precio\*cantidad) / COUNT(DISTINCT df.nro\_factura) 'Promedio facturación'

from facturas f, detalles\_factura df

where f.nro\_factura=df.nro\_factura

AND YEAR(f.fecha) BETWEEN YEAR(GETDATE()) AND DATEADD(YEAR, -1, GETDATE())

end

END

--4) Crear un SP que muestre las ordenes de pedido que vencen la semana proxima. Detallar codigo de producto, tipo, descripcion, cantidad, y los datos de contacto del cliente.

CREATE PROCEDURE SP\_CONSULTA\_ORDENES\_PEDIDO

AS

BEGIN

SELECT op.id\_orden\_pedido 'Orden de Pedido',

p.id\_producto 'Codigo de Producto',

p.id\_tipo\_producto 'Tipo',

p.descripcion 'Descripcion',

op.cantidad 'Cantidad',

c.nombre 'Cliente',

c.telefono 'Telefono',

c.email 'E-mail'

FROM ORDENES\_PEDIDO op

JOIN PRODUCTOS p ON op.id\_producto = p.id\_producto

JOIN CLIENTES c ON op.id\_cliente = c.id\_cliente

WHERE fecha\_entrega BETWEEN GETDATE() AND DATEADD(WEEK, 1, GETDATE())

END

--5) Crear un trigger que muestre el stock disponible de un artículo luego de efectuar la venta del mismo

create trigger TR\_VENTA\_STOCK

on detalles\_factura for insert

as

declare @cod\_prod int, @cantidad int, @stock int

select @cod\_prod = p.id\_producto, @cantidad = cantidad , @stock = stock

from inserted i

join productos p on p.id\_producto = i.id\_producto

begin

update PRODUCTOS set stock = @stock-@cantidad where id\_producto = @cod\_prod

end