

Trabajo Práctico Final

El trabajo práctico final consiste en el desarrollo de una aplicación implementada en Java. La misma será desarrollada con Eclipse como IDE, y con los lineamientos dados en las prácticas.

Las pautas generales del trabajo:

- ✓ Comisiones de 2 personas.
- ✓ Para este trabajo se destinarán las últimas 4 clases del curso, debiéndose entregar antes de finalizar el mismo.
- ✓ Se deberá entregar la aplicación funcionando y con toda la funcionalidad solicitada. Para la entrega se deberá construir un archivo en formato JAR con todas las clases y recursos usados en la aplicación con la finalidad de poder ejecutarlo desde el SO. Para el armado del archivo JAR, se cuenta con un ANEXO, que el profesor explicará.

Descripción de la funcionalidad de la aplicación:

Usted es contratado para realizar un sistema de **Facturación**, para lo cual deberá llevar el control de los **productos** comprados por los **clientes**.

Funcionalidad:

La **facturación** de productos a un cliente consiste en chequear la disponibilidad en el stock de los productos, decrementar la cantidad vendida y calcular el monto total de la factura considerando el IVA y los descuentos que se aplican de acuerdo al volumen de productos que se compran.

La aplicación deberá contar con interfaz de usuario gráfica siguiendo el patrón MVC. Se adjuntan imágenes a modo ilustrativo, que ejemplifican las ventanas que se espera que cuente la aplicación. Puede usar otros componentes de GUI como botones con imágenes, combo-box, etc. La aplicación deberá proveer de una ventana principal, en la cual a través de un *toolbar* o una barra de menús se podrá acceder a la funcionalidad del sistema.

El sistema deberá proporcionar los mecanismos necesarios para:

- dar de **alta** a nuevos clientes, **baja** y **modificación** de los ya existentes.
- dar de alta **alta** a nuevos productos y **modificación** de los ya existentes. Tenga en cuenta que el precio de un producto es sin IVA.

El sistema también contará con un módulo de consultas para obtener información diversa de **Clientes**, **Productos** y **Facturaciones** realizadas.

Detalles de implementación:

El proceso de facturación consiste en la generación de una **factura** con los productos que compra el cliente. Para realizar la compra, el **cliente** seleccionará **productos** de una lista que contiene todos los productos existentes (implementada con un JList, JTable, etc.) y los irá agregando a la lista de productos a comprar (implementada con una JTable). Además debe ingresar la cantidad a comprar de cada producto (JTextField). El sistema chequeará que esta cantidad no supere al stock disponible para ese producto, caso contrario le sugerirá al usuario la cantidad máxima disponible del mismo. El sistema debe permitir eliminar productos de la lista de productos a facturar (por ej. si un cliente selecciona un producto y luego se arrepiente, debe poder eliminarlo).

Porcentaje de descuentos a aplicar:

Si de un mismo **producto** lleva entre 5 y 10 unidades aplicar un descuento del 5% (sobre el producto)

Si de un mismo **producto** lleva más de 10 unidades aplicar un descuento del 10%.(sobre el producto)

Una vez que el **Cliente** haya confirmado la compra, el sistema deberá emitir la factura y aplicarle al precio total obtenido el IVA (21%) lo cual determinará el valor final de la **factura**. Con todos estos valores, se

Espacio de Actualización - Programación Orientada a Objetos

deberá registrar en la Base de Datos la **factura** correspondiente al cliente y así mismo actualizar los stocks de los productos comprados por el usuario.

Las clases **Cliente**, **Producto** y **Factura** están descriptas en los diagramas de clase UML que figuran debajo. Cada una de estas clases define variables de clase que permiten obtener códigos o números únicos en forma incremental para identificar a cada cliente, producto y factura respectivamente. Defina y use variables de clase de acuerdo a lo visto en la práctica 15.

Base de Datos:

Ud. tendrá una Base de Datos Access que se le entregará en el momento de realizar este trabajo final. Deberá definir un Origen de Datos tal cual lo aprendido en el curso. La BD estará pre-cargada con un stock inicial de productos y un conjunto de clientes.

Arquitectura de la Aplicación:

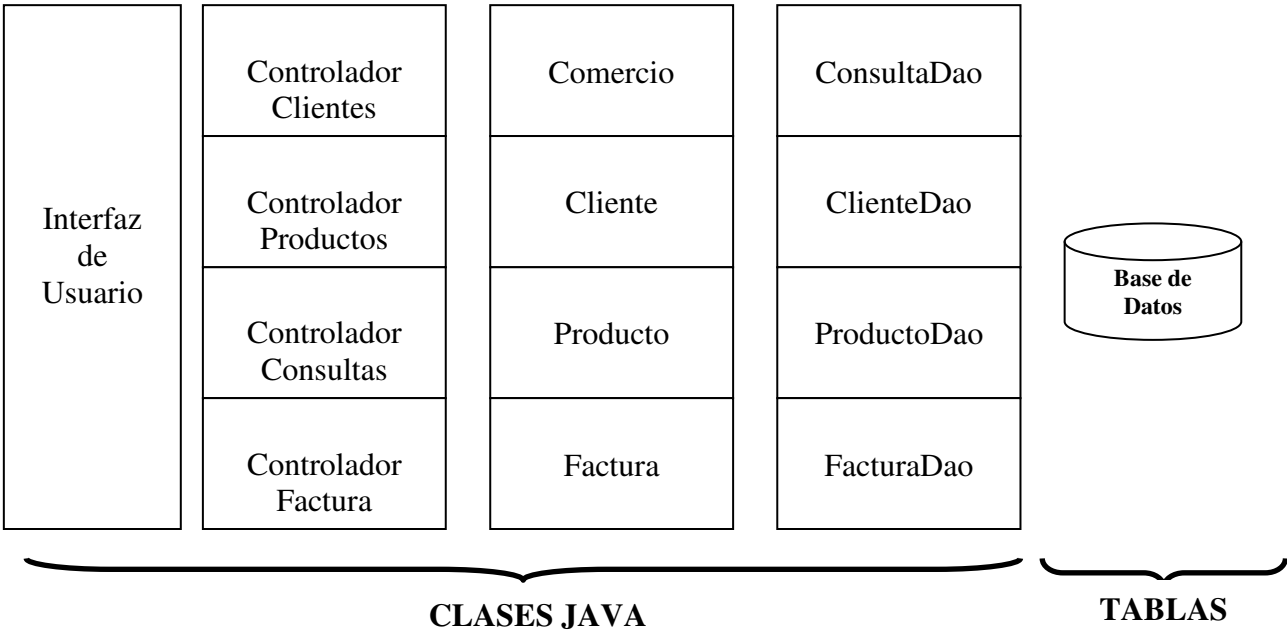
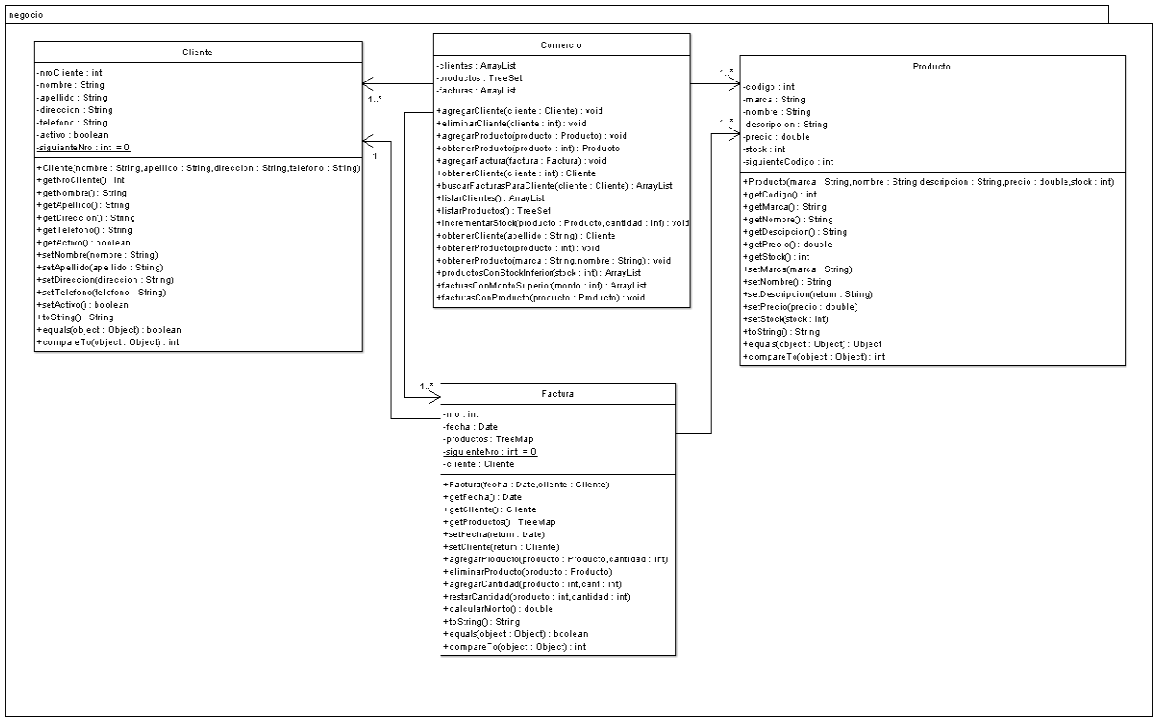


Diagrama UML de Clases:



Interfaz de Usuario de la aplicación:

Esta interfaz de usuario es a modo ilustrativo, Ud. podrá usar las componentes Swing que desee.

Clientes

Pantallas de Alta, Baja y Modificación. Listado de Clientes.

Nuevo Cliente

Nombre

Apellido

Dirección

Teléfono

Aceptar

Cancelar

Baja cliente

Cliente

1

Federico

Orlando

50 y 115

411-1111

Aceptar

Cancelar

Modificación de Cliente

Nombre

Apellido

Dirección

Teléfono

Seleccione el Cliente a Modificar

Nro Cliente	Nombre	Apellido	Dirección	Teléfono
1	Federico	Orlando	50 y 115	411-1111

Listado de Clientes

Nro Cliente	Nombre	Apellido	Dirección	Teléfono
1	Federico	Orlando	50 y 115	411-1111

Productos

Pantallas de Alta, Modificaciones e Incrementar stock. Listado de Productos.

Nuevo Producto

Nombre

Marca

Precio

Stock

Descripción

Modificación de Producto

Nombre

Marca

Precio

Stock

Descripción

Seleccione el Producto a Modificar

Nro Producto	Nombre	Marca	Descripción	Precio	Stock
1	M2N	Asus	Motherboard...	400	4

Facturas

Factura

Cliente

Nro Cliente	Nombre	Apellido	Dirección	Teléfono
1	Federico	Orlando	50 y 115	411-1111

Producto

Nro Producto	Nombre	Marca	Descripción	Precio	Stock
1	M2N	Asus	Motherboard A...	400	4

Cantidad

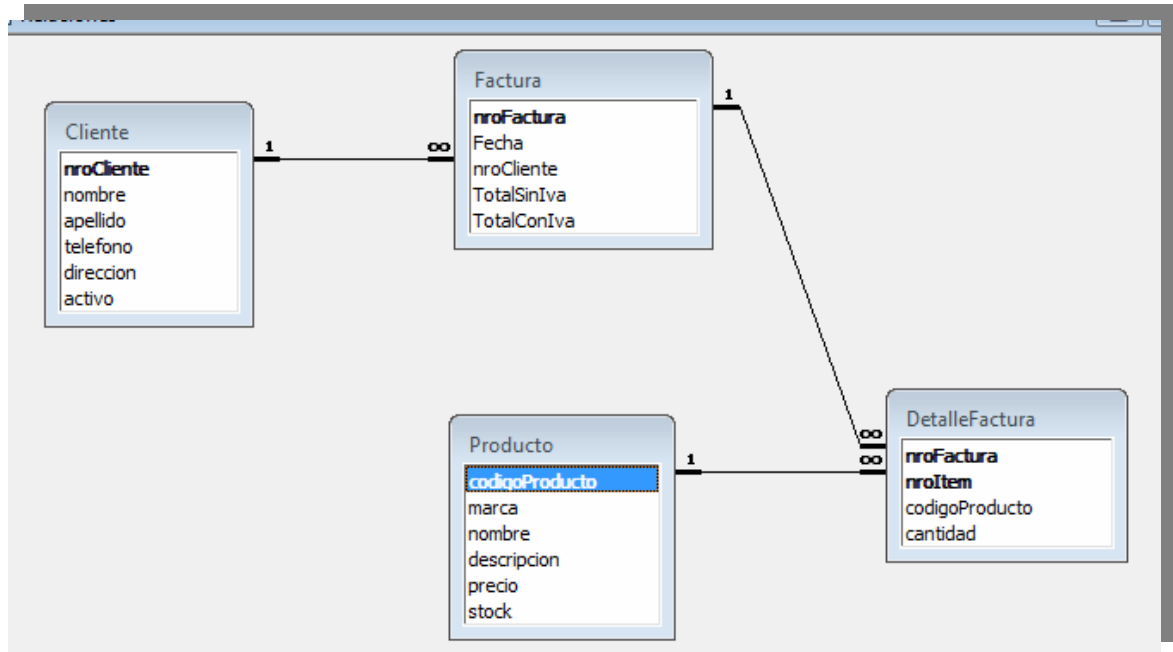
Productos comprados

Cantidad	Marca	Nombre	Precio
1	Asus	M2N	400

Consultas

Obtener un Cliente buscándolo por número de cliente o por Apellido.
 Obtener un Producto buscándolo por código de producto o la combinación de marca y nombre.
 Obtener todos los Productos que tengan un stock inferior a un valor dado.
 Todas las Facturas perteneciente a un cliente
 Todas las Facturas que superen un monto dado.
 Todas las Facturas que contengan un producto particular.

Modelo de Datos:



Instrucciones (Ejemplos) SQL para realizar toda la funcionalidad solicitada.

Inserción de un Cliente

```

    Insert into Cliente (nroCliente, nombre, apellido, telefono, direccion, activo)
    values (89,"Martin","Gomez","174287878","18 N78","S")
    
```

Eliminación de un Cliente (Baja Lógica)

```

    Update Cliente
    set activo = "No"
    where nroCliente = 12
    
```

Modificación de los datos de un Cliente

```

    Update Cliente
    set campoAModificar = valorModificado
    where campoCondicion = valorCondicion
    
```

Listados de los Clientes

*Select * from Cliente*

Inserción de un Producto

*insert into Producto(codigoProducto, marca, nombre, descripcion, precio, stock)
values (3,"Admiral","TV P21","Televisor 21 Pulgadas",673.99, 5)*

Modificación de los datos de un Producto

*update Producto
set Marca = "Philips"
where codigoProducto = 1*

Incrementar el stock de un Producto

*update Producto
set stock = stock + 10
where codigoProducto = 1*

Listados de los Productos

*Select * from Cliente*

Inserción de un Factura:

*insert into Factura (nroFactura, fecha, nroCliente, totalSinIva, totalConIva)
values(1,'03/11/2007',1,987,1194.27)
insert into DetalleFactura(nroFactura, nroItem, codigoProducto, cantidad)
values (1,1,1,1)*

Consultas

Obtener un Cliente buscándolo por número o por apellido

*Select * from Cliente where nroCliente = valor
Select * from Cliente where apellido= valor*

Obtener un Producto buscándolo por código o la combinación de marca y nombre

*Select * from Producto where codigoProducto = valor
Select * from Producto where marca = valorMarca and nombre = valorNombre*

Obtener todos los productos que tengan un stock inferior a un valor dado

*Select * from Producto where stock < valor*

Facturas pertenecientes a un cliente

*Select *
From Factura as fac INNER JOIN detalleFactura AS det ON fac.nroFactura = det.nroFactura
Where fac.nroCliente=1*

Facturas que superen un monto dado

*Select * From Factura Where totalConIva > valor*

Facturas que contengan un producto particular

*Select *
From Factura as fac INNER JOIN detalleFactura AS det ON fac.nroFactura = det.nroFactura
Where det.codigoProducto =1*

Definición de las clases Data Access Object (DAO)

ClienteDao

```
public void agregarCliente (Cliente cliente)
public eliminarCliente (Cliente cliente)
```

ProductoDao

```
public void agregarProducto (Producto producto)
public void incrementarStock(Producto producto, int cantidad)
```

FacturaDao

```
public void agregarFactura(Factura factura)
```

ConsultaDao

```
public Cliente obtenerCliente(int idCliente)
public Cliente obtenerCliente(String apellido)
public Collection obtenerTodosLosClientes()
public Collection obtenerTodosLosProductos()
public Factura buscarFacturaParaCliente(Cliente cliente)
public Collection productosConStockInferiorA(int cantidad)
public Collection facturasConMontoSuperior(double monto)
public Collection facturasConProducto(Producto producto)
```