

ESCUELA POLITÉCNICA NACIONAL

PROYECTO FINAL

BASE DE DATOS Y PROGRAMACIÓN AVANZADA

REALIZADO POR:

JOEL MENDOZA

PABLO TRUJILLO

**OBJETIVO: El Proyecto tiene como fin realizar una base de datos de un negocio de comidas, también realizar una interfaz gráfica en Netbeans.**

**RESUMEN: La finalidad de este proyecto es ayudar al cliente como al administrador, al momento de atender y ser atendido ahorrando tiempo, seguridad y una correcta administración de los datos.**

1. **INTRODUCCIÓN**

El implemento de una interfaz grafica y el uso adecuado de la base de datos pueden ayudar a que se revolucione la manera de atender al cliente, esto lleva a que se de una sensación de una atención mas satisfactoria.

Para esto se ha hecho uso del programa SQL server para crear la base de datos, así teniendo un mejor uso de los datos y almadeándolas de una manera correcta en tablas y tuplas. También se ha hecho uso del programa de Netbeans para implementar la interfaz gráfica, teniendo como resultado un programa que facilite la interacción del cliente al momento de comprar y del administrador controlando el negocio.

***Justification***

*El restaurant va optimizar los recursos al usar el programa que estará vinculado a una base de datos.*

1. ***Objetivos Generales***

*Crear una base de datos que esté vinculada con un programa realizado en Netbeans (Java) para que permita manipular y gestionar procedimientos comúnes al momento de realizar una compra en este restaurante.*

1. **MARCO TEORICO**

Una base de datos correctamente diseñada le proporciona acceso a información actualizada y precisa. Dado que un diseño correcto es esencial para lograr los objetivos en trabajar con una base de datos, dedique tiempo necesaria para obtener información sobre los principios de un buen diseño tenga sentido. Al final, que es mucho más probable que acabe con una base de datos que satisfaga sus necesidades y fácilmente puede acomodar el cambio.

En este artículo se proporciona instrucciones para planear una base de datos de escritorio. Aprenderá cómo decidir qué información que necesita, cómo dividir la información en las tablas y columnas adecuadas y las tablas se relacionan entre sí. Debe leer este artículo antes de crear la primera base de datos de escritorio.

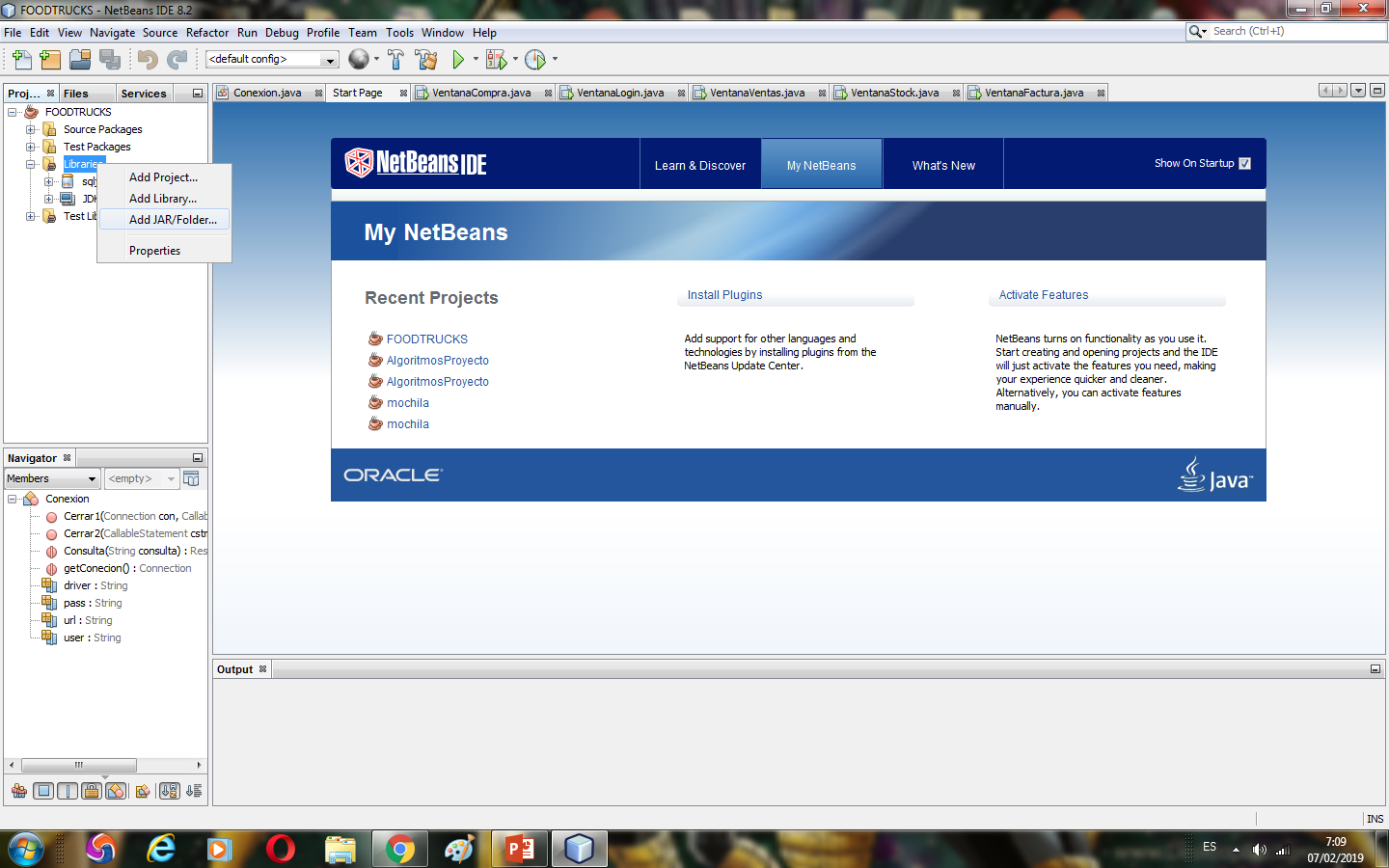
1. **DESARROLLO**

Antes de iniciar a desarrollar el Proyecto vamos a establecer la conexión entre nuestro compilador de Java Netbeans y nuestro gestor de Base de datos de SQLServer.

Para lo cual descargaremos un conector:



Posteriormente importaremos el conector a NetBeans:



A continuación crearemos una clase denominada “Conexión” en la cual estableceremos la conexión entre NetBeans y SQL

*public class Conexion {*

*static String user="supervisor";*

*static String pass="12345";*

*static String driver="com.microsoft.sqlserver.jdbc.SQLServerDriver";*

*static String url="jdbc:sqlserver://DESKTOP-APM83FO\\SQLEXPRESS:1433;databaseName=FOODTRUCKS";*

*public static Connection getConecion() {*

*Connection con = null;*

*try {*

*Class.forName(driver);*

*con = DriverManager.getConnection(url, user, pass);*

*} catch (ClassNotFoundException | SQLException e) {*

*e.printStackTrace();*

*}*

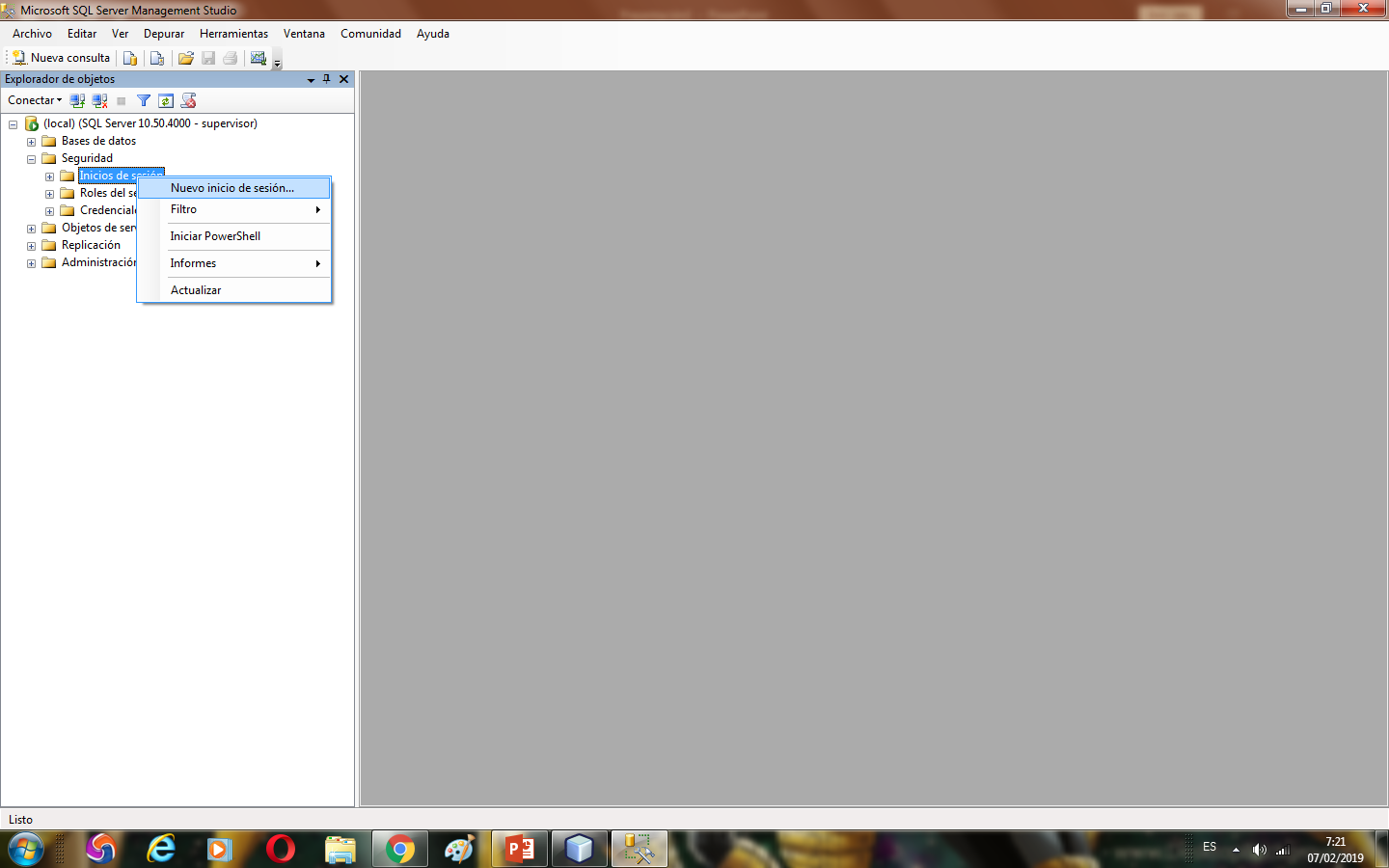
*return con;*

*}*

El string URL lo obtenemos de aquí:

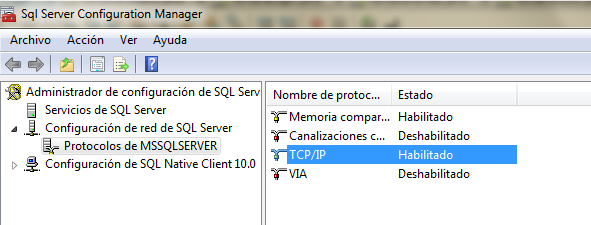


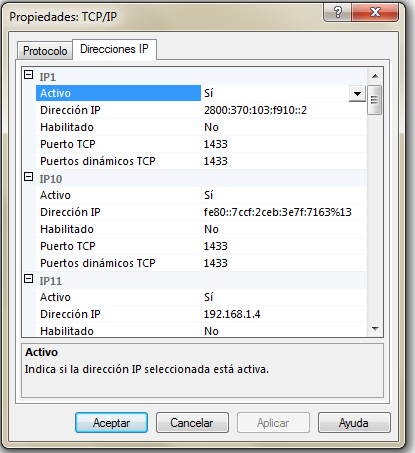
Los Strings user y supervisor son con los que nos conectamos a nuestra Base de Datos, previo a esto debemos crear un usuario con su respectiva contraseña, en este caso creamos supervisor con contraseña 12345



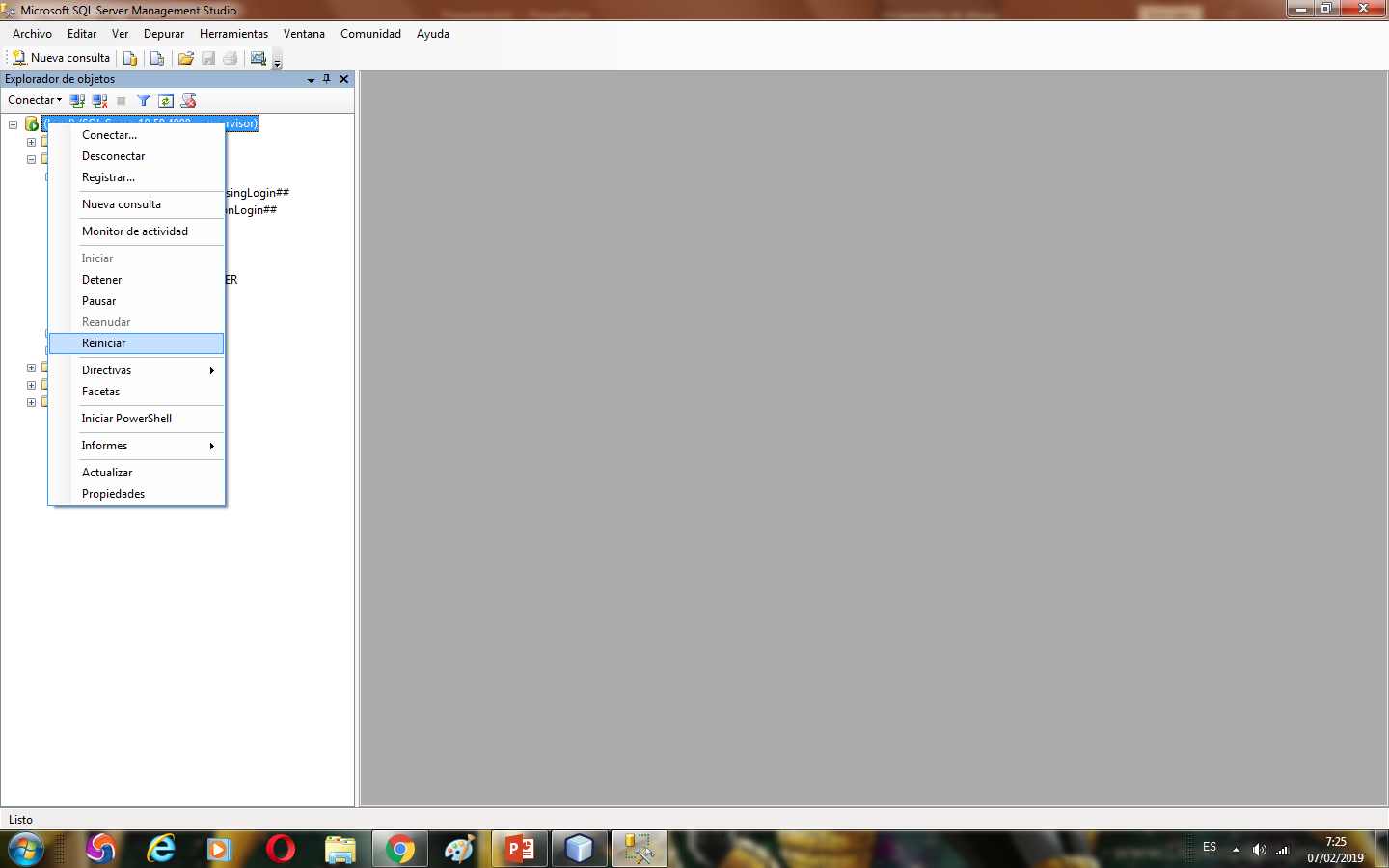
Una vez creado el inicio de sesión, en la pestaña Roles activamos la casilla “sysadmin” y en el apartado estado colocamos que esté disponible.

Para evitar errores, vamos a asegurarnos de que se esté usando el puerto 1433 para lo cual iremos al Administrador de configuración de SQL, dentro de TCP, primero lo activaremos y luego colocaremos los puertos 1433.

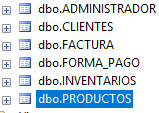




Una vez hecho esto, para que se apliquen los cambios reiniciamos el servidor en SQL de la siguiente manera:



Una vez establecida satisfactoriamente la conexión vamos a iniciar con la creación de la estructura de la base de datos para lo cual hemos tomado las siguientes entidades:



Las siguientes entidades contienen los siguientes atributos:

/\*CREACION DE TABLAS\*/

/\*CLIENTE\*/

USE FOODTRUCKS

CREATE TABLE CLIENTES

( NomCliente varchar(25),

ApeCliente varchar(25),

IDCliente numeric(10) primary key,

DirCli varchar(50),

FechNacCli date,

TelfCliente numeric(10),

CorreoCli varchar (40)

)

/\*PRODUCTOS\*/

USE FOODTRUCKS

CREATE TABLE PRODUCTOS

( CodProd numeric(10) primary key,

NomProd varchar(59),

PrecioProd decimal(3,2),

StockProd numeric(5),

CategoriaProd numeric(1)

)

/\*FACTURA\*/

USE FOODTRUCKS

CREATE TABLE FACTURA

(CodFactura numeric(6) primary key,

FechFactura date,

PrecioUnit numeric(4,2),

Cantidad numeric(6),

TotalFact numeric(6,2),

Descuento numeric(4,2),

APagar numeric(4,2)

)

USE FOODTRUCKS

CREATE TABLE FORMA\_PAGO

(IdPago numeric(6) primary key,

FormaPago numeric(1),

)

/\*ADMINISTRADOR\*/

USE FOODTRUCKS

CREATE TABLE ADMINISTRADOR

(NomAdmin varchar(25),

ApeAdmin varchar(25),

IdAdmin numeric(10) primary key,

DirAdmin varchar(50),

FechNacAdmin date,

TelfAdimin numeric(10),

CorreoAdmin varchar(50))

/\*INVENTARIO\*/

USE FOODTRUCKS

CREATE TABLE INVENTARIOS

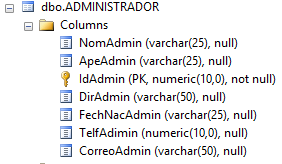
(CodInv numeric(5) primary key,

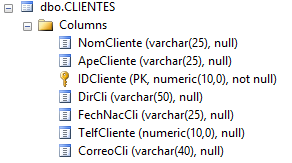
Insumo varchar(25),

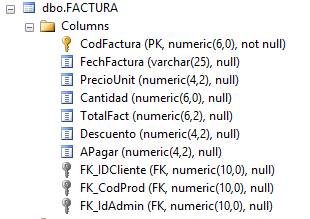
Existencias numeric(5),

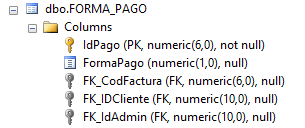
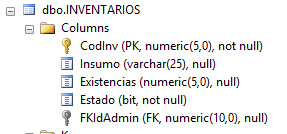
Estado bit not null default 1

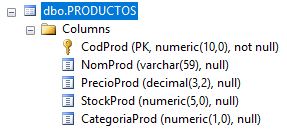
)











A cada entidad se la ha relacionado, de manera más conveniente, como es el caso las entidades: Factura, inventarios y forma de pagos.

USE FOODTRUCKS

ALTER TABLE FACTURA

ADD FK\_IDCliente numeric(10)

CONSTRAINT CLIENTES\_FACTURA FOREIGN KEY

REFERENCES CLIENTES(IDCliente)

USE FOODTRUCKS

ALTER TABLE FACTURA

ADD FK\_CodProd numeric(10)

CONSTRAINT PRODUCTOS\_FACTURA FOREIGN KEY

REFERENCES PRODUCTOS(CodProd)

USE FOODTRUCKS

ALTER TABLE FACTURA

ADD FK\_IdAdmin NUMERIC(10)

CONSTRAINT ADMINISTRADOR\_FACTURA FOREIGN KEY

REFERENCES ADMINISTRADOR(IdAdmin)

También se realizó un check para restringir los valores como son los casos

USE FOODTRUCKS

ALTER TABLE PRODUCTOS

ADD CONSTRAINT chkRowCount

CHECK (CategoriaProd >= 1 AND CategoriaProd <=3 );

GO

En esta restriccionla realizamos pues en este caso solo hay limitante tipo de productos, le vamos a direnciar poniendo 1 para los platos fuertes, 2 para las sopas y 3 para los postres.

USE FOODTRUCKS

ALTER TABLE FORMA\_PAGO

ADD CONSTRAINT chkRowFormaPago

CHECK (FormaPago >= 1 AND FormaPago <=3 );

GO

Este caso usaremos la restricción para diferenciar la forma de pago teniendo: 1 el pago en efectivo, 2 pago con tarjeta de crédito y 3 pago con tarjeta de débito.

Ingreso de datos a las tablas por medio del SQL Server. Pondremos 3 ejemplos de data entidad.

CLIENTES

USE FOODTRUCKS

INSERT INTO CLIENTES

VALUES('Nombre1','Apellido1',1720804432,

'Direccion1','1/1/1999',09987178186,

'nombre1@hotmail.es')

USE FOODTRUCKS

INSERT INTO CLIENTES

VALUES('Nombre2','Apellido2',1720801917,

'Direccion2','4/2/1998',0992514455,

'nombre2@hotmail.com')

USE FOODTRUCKS

INSERT INTO CLIENTES

VALUES('Nombre3','Apellido3',1720801909,

'Direccion3','6/3/1997',0999999999,

'nombre3@hotmail.com')

ADMINISTRADOR

USE FOODTRUCKS

INSERT INTO ADMINISTRADOR

VALUES('Admin1','A1',0120804455,'dir1',

'01/30/1996',0987654321,'admin1@hotmail.com')

USE FOODTRUCKS

INSERT INTO ADMINISTRADOR

VALUES('Admin2','A2',0120804454,'dir2',

'11/22/1995',0987654320,'admin2@hotmail.com')

USE FOODTRUCKS

INSERT INTO ADMINISTRADOR

VALUES('Admin3','A3',0120804453,'dir3',

'12/24/1997',0987654300,'admin3@hotmail.com')

PRODUCTOS

USE FOODTRUCKS

INSERT INTO PRODUCTOS

VALUES(001,'Enchiladas',3.00,40,1)

USE FOODTRUCKS

INSERT INTO PRODUCTOS

VALUES(005,'Pastel tres leches',1.50,30,3)

USE FOODTRUCKS

INSERT INTO PRODUCTOS

VALUES(008,'Sopa de tortilla', 2.5,43,2)

FACTURAS

INSERT INTO FACTURA

VALUES(1,'01/25/2019',3.00,3,9.00,0.90,8.1,

1720801905,1,120804446)

USE FOODTRUCKS

INSERT INTO FACTURA

VALUES(2,'01/25/2019',4.50,2,9.00,0.90,8.1,

1720801904,2,120804447)

USE FOODTRUCKS

INSERT INTO FACTURA

VALUES(3,'01/25/2019',5.00,1,5.00,0.50,4.5,

1720801903,3,120804448)

INVENTARIOS

USE FOODTRUCKS

INSERT INTO INVENTARIOS

VALUES(1,'Tomates',100,1,120804449)

USE FOODTRUCKS

INSERT INTO INVENTARIOS

VALUES(2,'Lechugas',45,1,120804452)

USE FOODTRUCKS

INSERT INTO INVENTARIOS

VALUES (3, 'Tortillas' , 84,1, 120804446)

FORMAS\_PAGO

USE FOODTRUCKS

INSERT INTO FORMA\_PAGO

VALUES(1,1,1,1720801905,120804446)

USE FOODTRUCKS

INSERT INTO FORMA\_PAGO

VALUES(2,3,2,1720801904,120804447)

USE FOODTRUCKS

INSERT INTO FORMA\_PAGO

VALUES(3,2,3,1720801903,120804448)

Dos vistas por interés estos comando con select usamos para poder trasferir a las tablas al ala interfaz grafica:

select CodInv,Insumo,Existencias

from INVENTARIOS

select CodFactura,FechFactura,FK\_CodProd,NomProd,

PrecioUnit,Cantidad,Descuento,APagar

from FACTURA,PRODUCTOS

where FACTURA.FK\_CodProd=PRODUCTOS.CodProd

A continuación vamos a diseñar la interfaz gráfica de nuestro programa, la cual está formada por:

-VentanaLogin



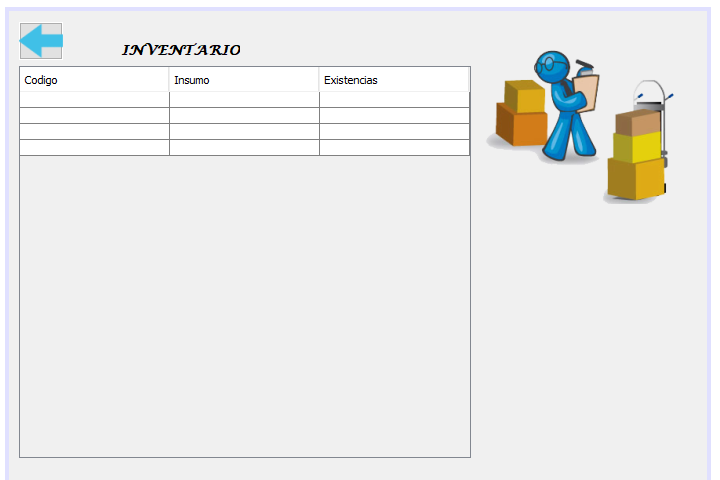
-VentanaPregunta



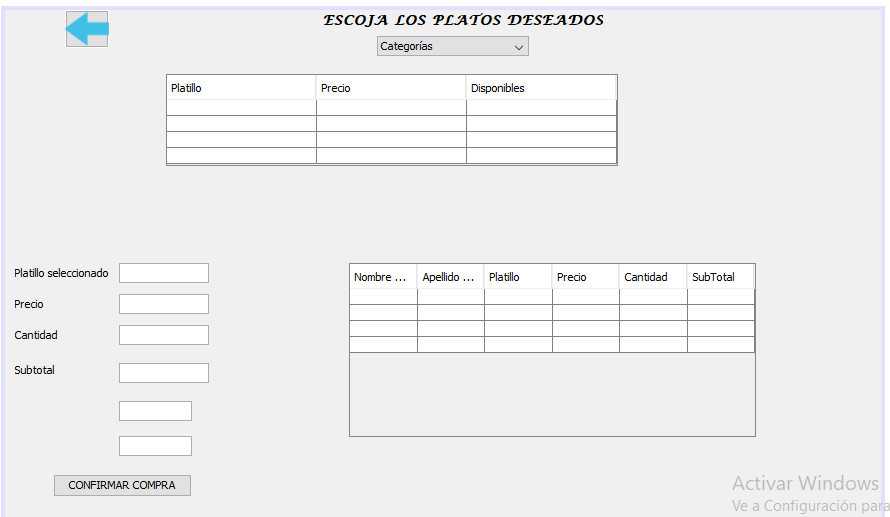
-VentanaVentas



-VentanaStock



-VentanaCompras



Bien, una vez hecha la “maqueta” de nuestro programa, vamos a desarrollar la parte lógica para lo cual vamos a definir un método en nuestra clase Conexion, este método nos permitirá enviar sentencias sql dentro de Netbeans, y de esta forma obtener los datos deseados:

*public static ResultSet Consulta(String consulta){*

*Connection con= getConecion();*

*Statement declara;*

*try{*

*declara= con.createStatement();*

*ResultSet respuesta = declara.executeQuery(consulta);*

*return respuesta;*

*}catch(SQLException e){*

*JOptionPane.showMessageDialog(null,"No se pudo establecer la conexion"+e.getMessage(),"Error de Conexion",JOptionPane.ERROR\_MESSAGE);*

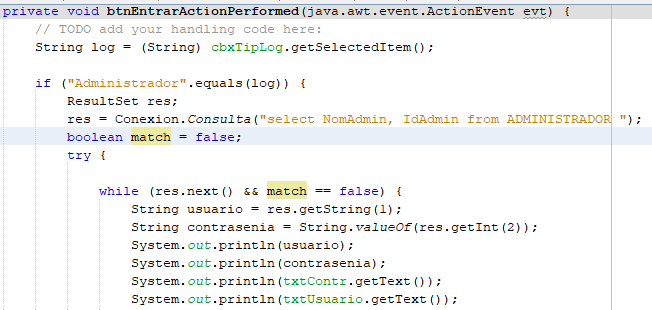
*}*

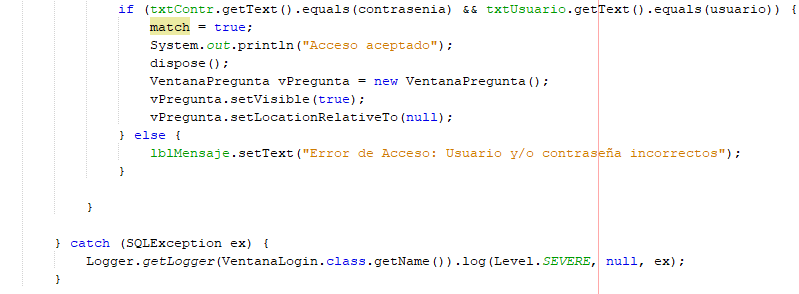
*return null;*

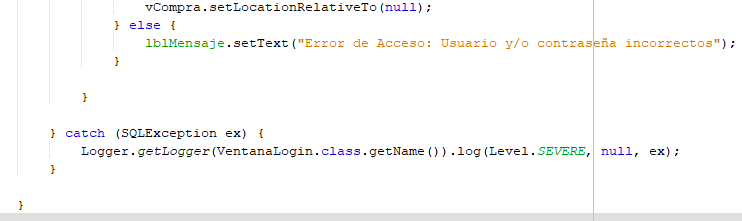
*}*

*}*

El programa inicia con la ventana de login, en donde al dar clic en “ENTRAR” compararemos los datos ingresados con lo de la tabla de Administradores o Clientes según lo escogido en el combobox, de esta manera, daremos acceso o lo denegaremos.



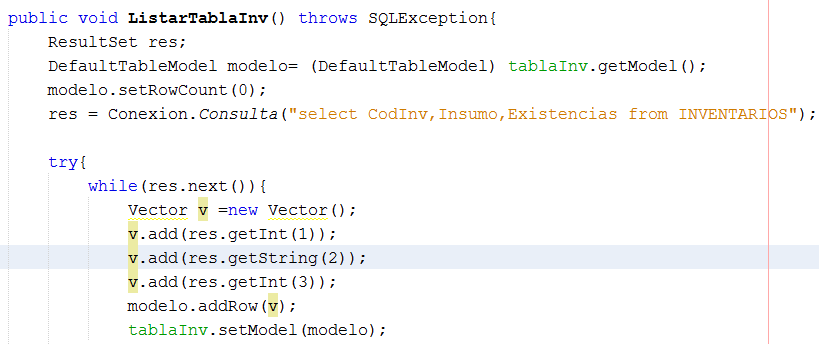


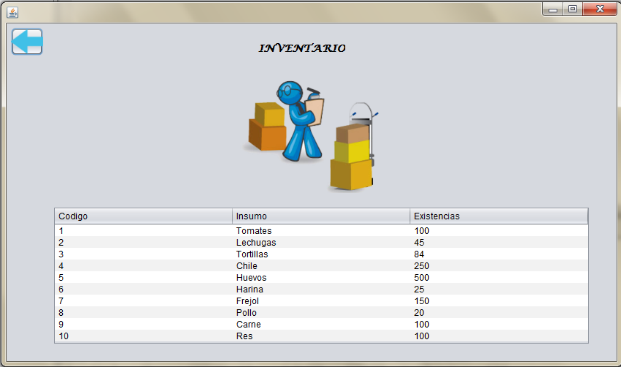


Cuando accede como administrador se tienen 2 opciones: “Ventas” y “Stock” dentro de “Ventas” se observarán las transacciones realizadas, en el “Stock” se verán los productos con los que se realizan los platillos, y en las Ventas, las transacciones realizadas.

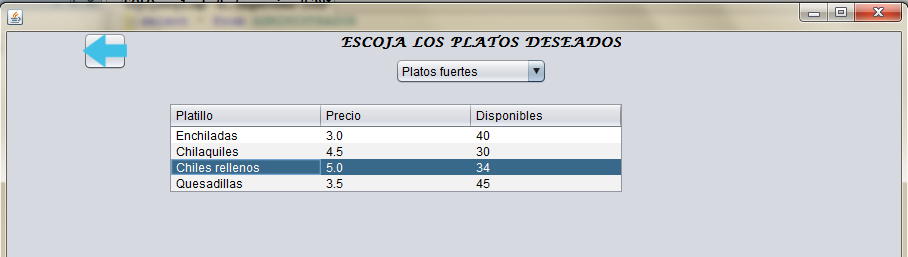
Lo principal que se hace en cada una de estas ventanas es la creación de una tabla, donde ubicaremos los datos requeridos “leyéndolos” desde nuestra Base de Datos, para esto enviamos una consulta select.

*select CodInv,Insumo,Existencias from INVENTARIOS*





Ahora, si es que el acceso se da como cliente, se nos desplegará una pantalla de compra, con nuestros platillos categorizados según “Entradas”, “Platos fuertes” y “Postres”.

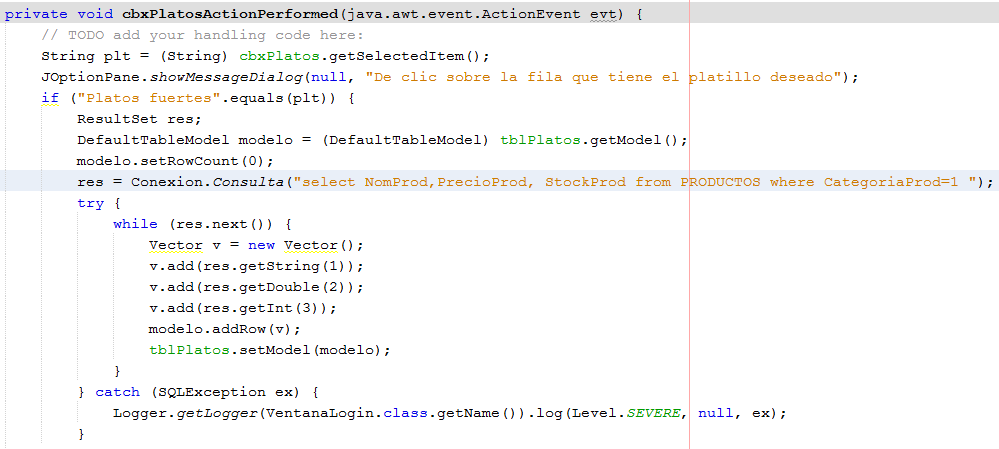


Según lo seleccionado en el ComboBox, nos muestra una tabla con información de platillos, código, Platillo, Precio, Existencias. Esto tomándolo de nuestra base de datos con un método similar al mostrado en las ventanas anteriores. La diferencia radica en que aquí enviamos una condición where a la consulta para que según eso tome los platillos de una clasificación específica. Para esto, vamos tomando los datos de la BD, los almacenamos en un vector y envíamos esto a cada fila de nuestra tabla.

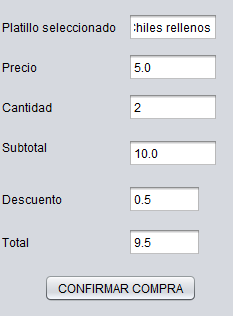
*select NomProd,PrecioProd, StockProd*

*from PRODUCTOS*

*where CategoriaProd=1*

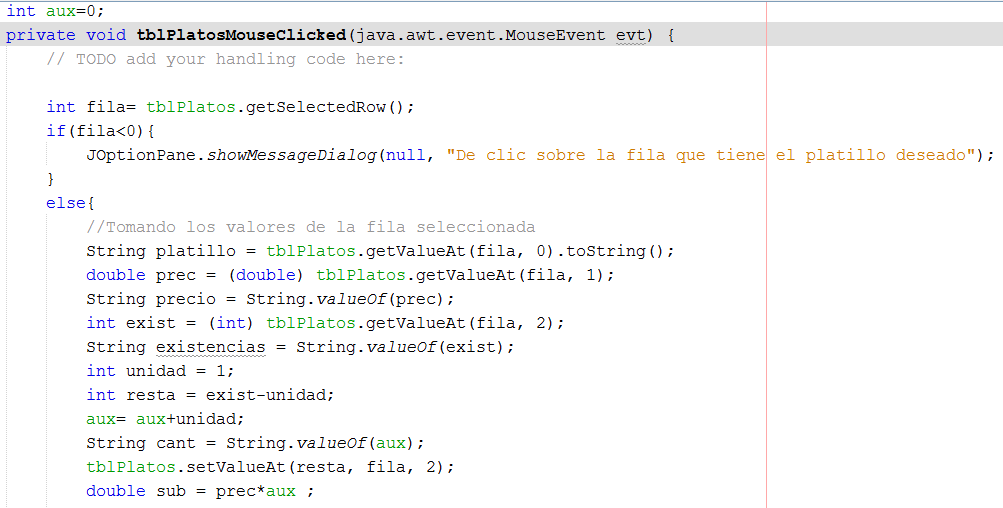


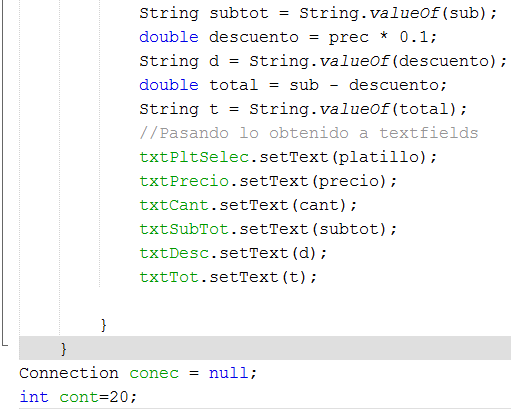
En esta misma ventana, una vez que demos clic en un registro de la tabla, ya estaremos seleccionando un producto para comprar, por lo que tendremos una “lista” de que es lo que estamos realizando.



Además de que observamos que si volvemos a dar clic en el platillo, la “Cantidad” aumentará y en la tabla las “Existencias” disminuirán.

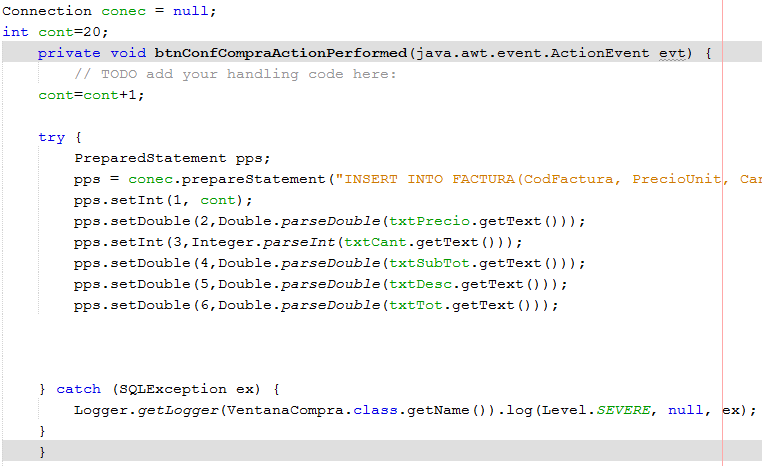
Para tomar los datos de la tabla, usamos un evento propio de los JTables, el cual lo modificamos para enviar la información a los campos de texto requeridos:





Finalmente damos clic en CONFIRMAR COMPRA que correrá este método que inserta los datos de la compra en nuestra entidad FACTURA.

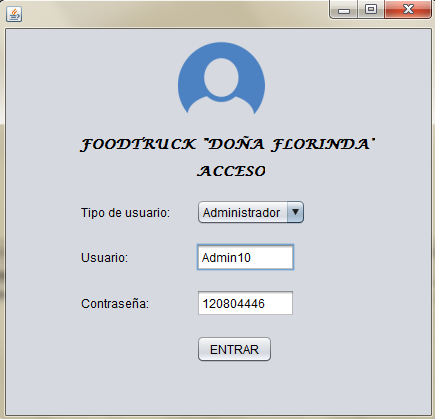
*INSERT INTO FACTURA(CodFactura, PrecioUnit, Cantidad, TotalFact, Descuento, APagar)VALUES(?,?,?,?,?,?)*

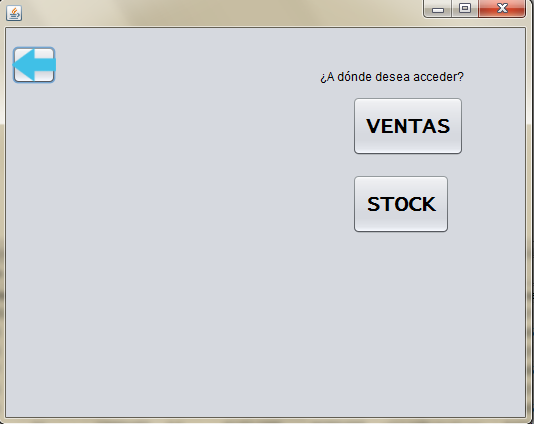


**RESULTADOS**

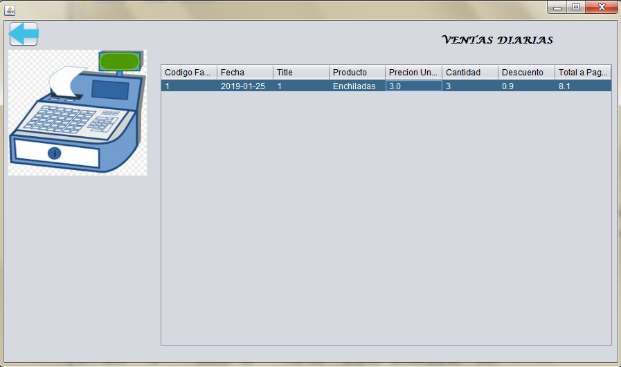
El programa corriendo iría de la siguiente manera:

Para Administrador:





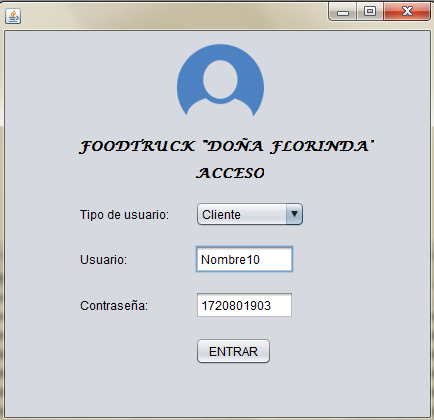
Si seleccionamos “Ventas”:

****

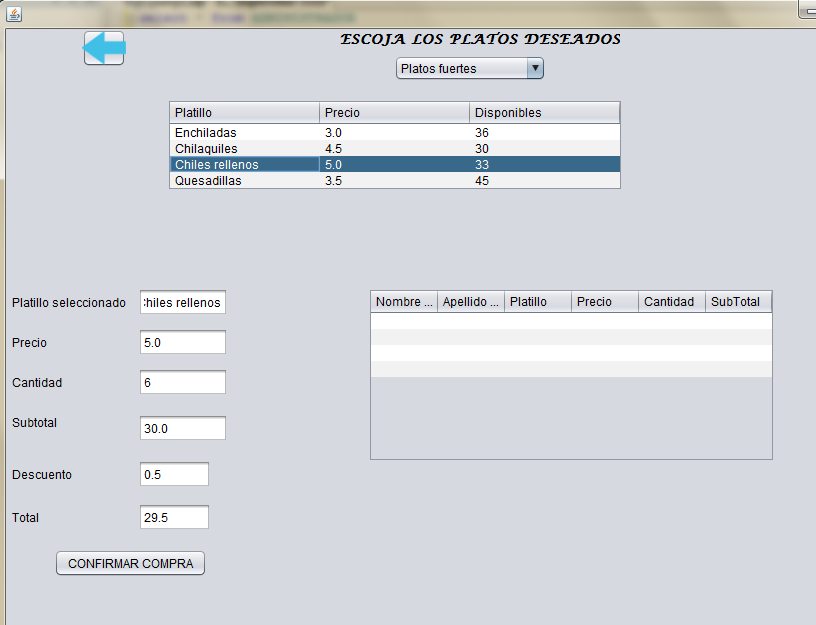
Si seleccionamos “Stock”:

****

Para cliente:







1. **CONCLUSIONES**
2. **RECOMENDACIONES**

* Ser cuidadosos en la conexión, por algún carácter mal conectado en alguno de los parámetros necesarios para que se conecte, podemos pasar horas buscando el error.
* Para diseñar un login considerar que cada tipo de acceso nos llevará a una ventana distinta.
* Crear el método consulta, sin este nos será imposible enviar sentencias SQL dentro de Netbeans, lo que implicaría trabajar con procedimientos que nos obliga a “saltar” de Netbeans a SQL y visceversa.
* Diseñar la BD, considerando que “objetos” de los que pueda tener menos de 7 registros, no deben ser entidades, deberían ser un atributo de otra entidad.
* Seguir la implementación de la Base de Datos acorde al modelo lógico
* Relacionar debidamente las entidades, eliminando columnas repetidas.

1. **BIBLIOGRAFÍA**
2. ***Páginas web***

https://www.youtube.com/watch?v=4q2sAMU5b0Y

<https://support.office.com/es-es/article/conceptos-b%C3%A1sicos-del-dise%C3%B1o-de-una-base-de-datos-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>

<https://es.slideshare.net/narkamo3/habilitar-la-autenticacin-sql-y-crear-un-nuevo-usuario-sql>

<https://www.youtube.com/watch?v=eKN-tTWVw2Q>

<https://es.stackoverflow.com/questions/1487/guia-definitiva-de-conversi%C3%B3n-de-tipos-en-java>