

**PROYECTO FINAL JAVA
SISTEMA DE VENTAS UNAL MARKET**

Presentado por:

Julio Mario Torres Fandiño - jutorresf@unal.edu.co
Kevin Leandro Rojas Marín - krojasm@unal.edu.co

Profesor:

Juan Bernardo Gomez Mendoza
ibgomez@unal.edu.co

Manizales, Caldas



**Universidad Nacional de Colombia
Facultad de Administración
Departamento de Informática y Computación
2022**



CONTENIDO

1. Descripción	3
2. Etapas de elaboración y arquitectura de la solución:	4
2.1 Etapa 1 - Creación Base de datos	5
2.2 Etapa 2 - Creación Formulario Login	8
2.3 Etapa 3 - Elaboración interfaz ventana principal	9
2.4 Etapa 4 - Elaboración apartado 'MANTENIMIENTO' - Módulo Cliente	10
2.5 Etapa 5, 6 - Elaboración apartado 'MANTENIMIENTO' - Módulo Producto y Empleado	13
2.6 Etapa 7 - Elaboración interfaz gráfica apartado 'VENTAS' - Módulo 'Generar Venta'	16
2.7 Etapa 8 - Elaboración métodos apartado 'VENTAS' - Módulo 'Generar Venta'	17
3. Glosario	22
4. Referencias	24



1. Descripción

Sistema de Ventas JK es un software utilizado como herramienta de gestión empleada para registrar las cantidades de mercancías existentes en un negocio, así como para automatizar todo lo relacionado con los procesos de ventas, mantenimiento de clientes, productos y empleados. Este sistema facilita el registro de ventas, clientes e inventario de un local de ventas independientemente de los productos que se tengan en stock, conectado a un sistema gestor de bases de datos que emplea el modelo relacional, en un lenguaje de consulta estructurado (MySQL), conectado a un servidor local por XAMPP.

Sistema de Ventas JK surgió debido al alto costo de programas similares para la gestión de datos, este sistema de información es asequible y portable, ya que se puede ejecutar en cualquier sistema operativo descargando el ejecutable del programa. El software cuenta con un sistema básico de seguridad por medio de validación de credenciales (nombre de usuario y contraseña), por lo que está orientado y hecho para que lo manipulen los colaboradores pertenecientes al negocio. Para que la experiencia que va a tener el usuario sobre el programa, se cuenta con una interfaz gráfica intuitiva que favorece la curva de aprendizaje y hace que la interacción que tiene el usuario sobre el software sea haga de manera más natural, gracias a las validaciones que se hacen en caso que se registren datos erróneos.

Antes de describir las etapas de elaboración y la arquitectura de la solución para este software, mi compañero y yo nos reunimos a hacer un diagrama de flujo detallando el funcionamiento general del sistema, para posteriormente implementar el código fuente que le da vida al proyecto.

Diagrama de flujo explicando la conceptualización general del problema a abordar concebida inicialmente.

<https://drive.google.com/file/d/1Lyao9yefZPUJcW6j6Jz9CrsLxPHTOVYb/view?ts=6386c6ce>

NOTA: Se recomienda visualizar el diagrama de flujo en la aplicación web draw.io para mejor resolución de la imagen.



2. Etapas de elaboración y arquitectura de la solución:

En la elaboración del software que respalda el proyecto, se empleó el paradigma de programación orientado a objetos, desarrollado en el lenguaje de programación JAVA implementado en el **IDE** Netbeans el cual cuenta con la herramienta GUI para facilitar el diseño de la interfaz gráfica. Ya en la interacción usuario-máquina, *Sistema de Ventas JK* cuenta con módulos dispuestos en un menú situado en la parte superior de la ventana principal, en donde el usuario puede elegir entre diferentes opciones, una de ellas es el módulo de '**MANTENIMIENTO**' en donde se registran, actualizan, eliminan y agregan los productos, clientes o empleados. Otra de las funcionalidades principales se refiere al apartado de '**VENTAS**' en la opción '**Generar Venta**', allí el empleado tiene que llenar un formulario con los campos asociados al cliente y al producto, en donde se van agregando a la tabla de facturación para posteriormente generar la venta con el subtotal por producto y el total vendido. En estos módulos se despliegan diferentes opciones en donde cada una redirecciona a su ventana correspondiente, allí se disponen todos los objetos, formularios en cajas de texto, paneles y botones necesarios para que la experiencia entre el programa y el usuario sea intuitiva, fácil y natural.

A continuación se describirán los métodos y la manera en cómo se abordaron cada uno de los apartados y módulos del proyecto, divididos por etapas.

2.1 Etapa 1 - Creación Base de datos

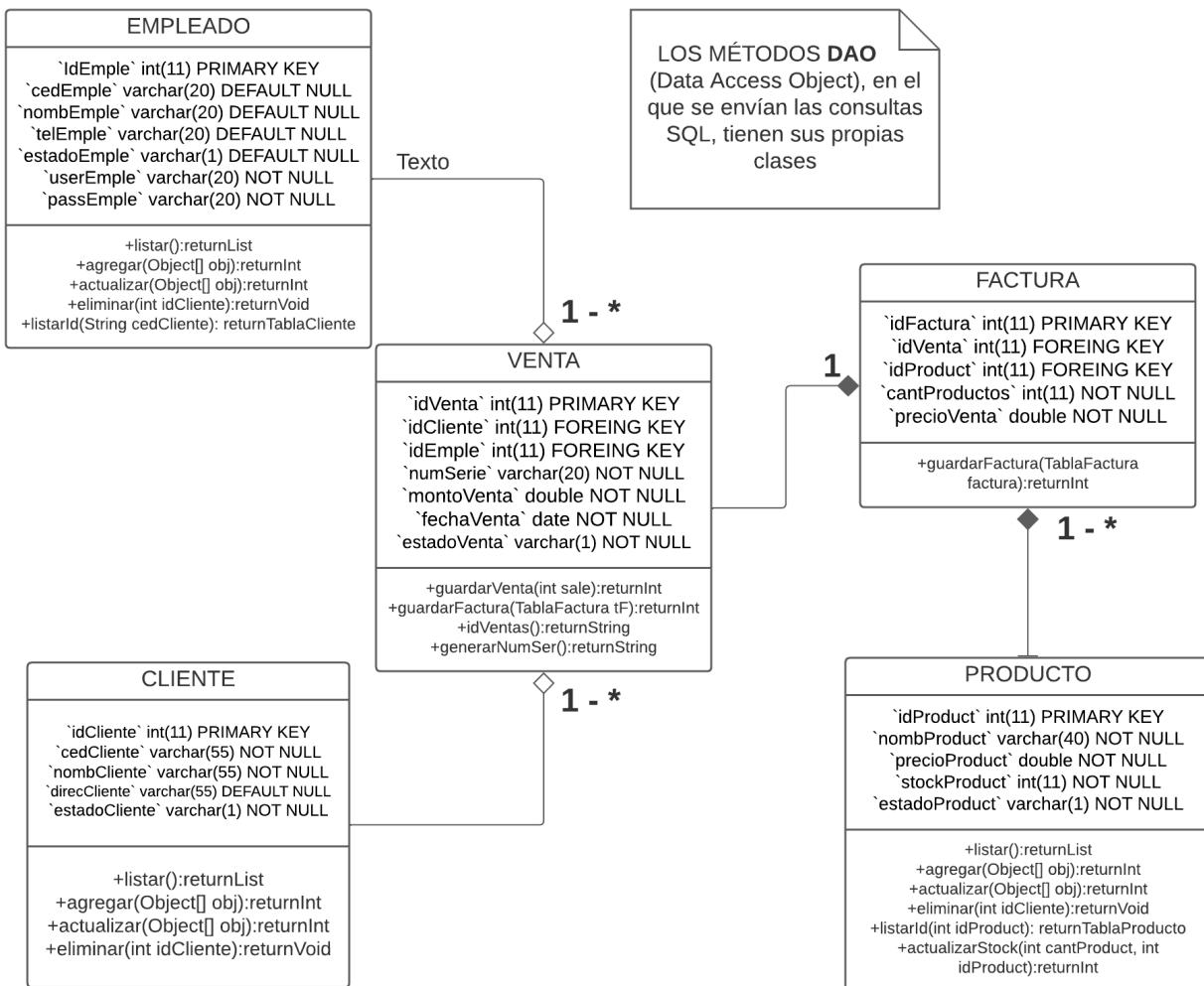
Responsable: Julio Mario T. - Kevin Leandro R.

Fecha: 11 de noviembre de 2022

Duración: 5 horas

La manera en que abordamos el problema inicialmente nos llevó a investigar y documentar todo lo relacionado a la elaboración de un sistema de ventas para la gestión de inventario, clientes y productos que permitieran ser almacenados en una base de datos con un modelo relacional el cuál facilita la posterior implementación de clases en el lenguaje de programación estipulado.

Para ello hicimos el siguiente diagrama entidad - relación:





Una vez culminado el diagrama entidad-relación, implementamos las consultas necesarias para materializar el diagrama en el SGBD.

phpMyAdmin

Servidor: localhost Base de datos: sistemaVentasJK

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Más

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
CLIENTE	Examinar Estructura Buscar Insertar Vaciar Eliminar	15	InnoDB	utf8mb4_general_ci	16.0 KB	-
EMPLEADO	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_general_ci	16.0 KB	-
FACTURA	Examinar Estructura Buscar Insertar Vaciar Eliminar	12	InnoDB	utf8mb4_general_ci	48.0 KB	-
PRODUCTO	Examinar Estructura Buscar Insertar Vaciar Eliminar	12	InnoDB	utf8mb4_general_ci	16.0 KB	-
VENTA	Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_general_ci	48.0 KB	-
5 tablas Número de filas		53	InnoDB	utf8mb4_general_ci	144.0 KB	0 B

Seleccionar todo Para los elementos que están marcados:

Imprimir Diccionario de datos

Crear nueva tabla Consola

Arquitectura en cada una de las tablas de la base de datos 'sistemaVentasJK'

phpMyAdmin

Servidor: localhost Base de datos: sistemaVentasJK Tabla: CLIENTE

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Más

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idCliente	int(11)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar
2	cedCliente	varchar(55)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
3	nombrCliente	varchar(55)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
4	direcCliente	varchar(55)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar
5	estadoCliente	varchar(1)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar

Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice Espacial Texto completo Agregar a columnas centrales Eliminar de las columnas centrales

TABLA CLIENTE

phpMyAdmin

Servidor: localhost Base de datos: sistemaVentasJK Tabla: EMPLEADO

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Más

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	IdEmple	int(11)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar
2	cedEmple	varchar(20)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar
3	nombrEmple	varchar(20)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar
4	telEmple	varchar(20)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar
5	estadoEmple	varchar(1)	utf8mb4_general_ci		Sí	NULL			Cambiar Eliminar
6	userEmple	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
7	passEmple	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar

TABLA EMPLEADO



SISTEMA DE VENTAS JK

Examinar | Estructura | SQL | Buscar | Insertar | Exportar | Importar | Privilegios | Operaciones | Más

Estructura de tabla | Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idProduct	int(11)		UNSIGNED	No	Ninguna			Cambiar Eliminar Más
2	nombrProduct	varchar(40)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
3	precioProduct	double			No	Ninguna			Cambiar Eliminar Más
4	stockProduct	int(11)			No	Ninguna			Cambiar Eliminar Más
5	estadoProduct	varchar(1)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice

TABLA PRODUCTO

phpMyAdmin

Examinar | Estructura | SQL | Buscar | Insertar | Exportar | Importar | Privilegios | Operaciones | Más

Estructura de tabla | Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idVenta	int(11)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar
2	idCliente	int(11)		UNSIGNED	No	Ninguna			Cambiar Eliminar
3	idEmpleo	int(11)		UNSIGNED	No	Ninguna			Cambiar Eliminar
4	numSerie	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
5	montoVenta	double			No	Ninguna			Cambiar Eliminar
6	fechaVenta	varchar(55)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
7	estadoVenta	varchar(1)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar

Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice

TABLA VENTA

phpMyAdmin

Examinar | Estructura | SQL | Buscar | Insertar | Exportar | Importar | Privilegios | Operaciones | Más

Estructura de tabla | Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idFactura	int(11)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	idVenta	int(11)		UNSIGNED	No	Ninguna			Cambiar Eliminar Más
3	idProduct	int(11)		UNSIGNED	No	Ninguna			Cambiar Eliminar Más
4	cantProductos	int(11)			No	Ninguna			Cambiar Eliminar Más
5	precioVenta	double			No	Ninguna			Cambiar Eliminar Más

Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice

Espacial Texto completo Agregar a columnas centrales Eliminar de las columnas centrales

TABLA FACTURA



2.2 Etapa 2 - Creación Formulario Login

Responsable: Julio Mario Torres

Fecha: 13 de noviembre de 2022

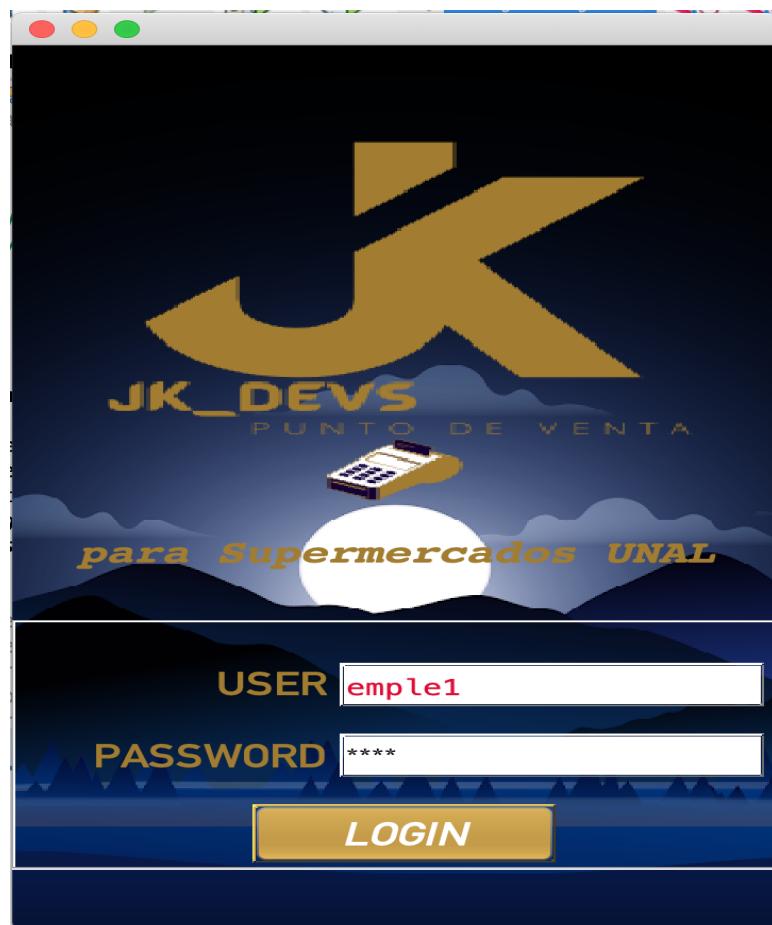
Duración: 6 horas

Para ello creamos el proyecto en el IDE Netbeans, el cual lo nombramos sistemaVentasJK. En la estructura del proyecto le agregamos la librería mysql.connector para conectar la base de datos anteriormente creada al proyecto hecho en java. Para la creación de esta etapa se crearon las clases tanto de tipo JFrame como de tipo DAO para la extracción de datos por las consultas enviadas a la base de datos, como de tipo entidad en donde se instancian las columnas de la tabla '*'EMPLEADO'*'. Para la creación del "*LoginForm*" importamos imágenes en formato .png y .jpg al paquete '*Imagenes*' para asignarlos a cada jLabel que contiene la imagen.

Después de ello se hizo la clase '*Conexion.java*' en el paquete '*Modelo*', la cual tiene como objetivo aplicar el método '*conectarBD()*', en donde se establece la conexión por medio de la función '*.getConnection()*'.

Fecha: 14 de noviembre de 2022

Al siguiente día con la ayuda del GUI de Netbeans se realizó la siguiente interfaz del formulario Login:



Este formulario consta de dos campos, uno para ingresar el nombre del usuario y el otro para la contraseña, todo esto asociado a los empleados que ya se encuentren registrados dentro de la base de datos.



Posterior a esto, se implementó un método en la configuración del evento cuando el usuario le da click al *jButton 'Login'*, el cual se encarga de validar los campos de texto asociados al nombre de usuario y contraseña registrados en la tabla ‘EMPLEADO’.

Si la validación sale exitosa, se procede a ejecutar el método que muestra la ventana principal del programa.

2.3 Etapa 3 - Elaboración interfaz ventana principal

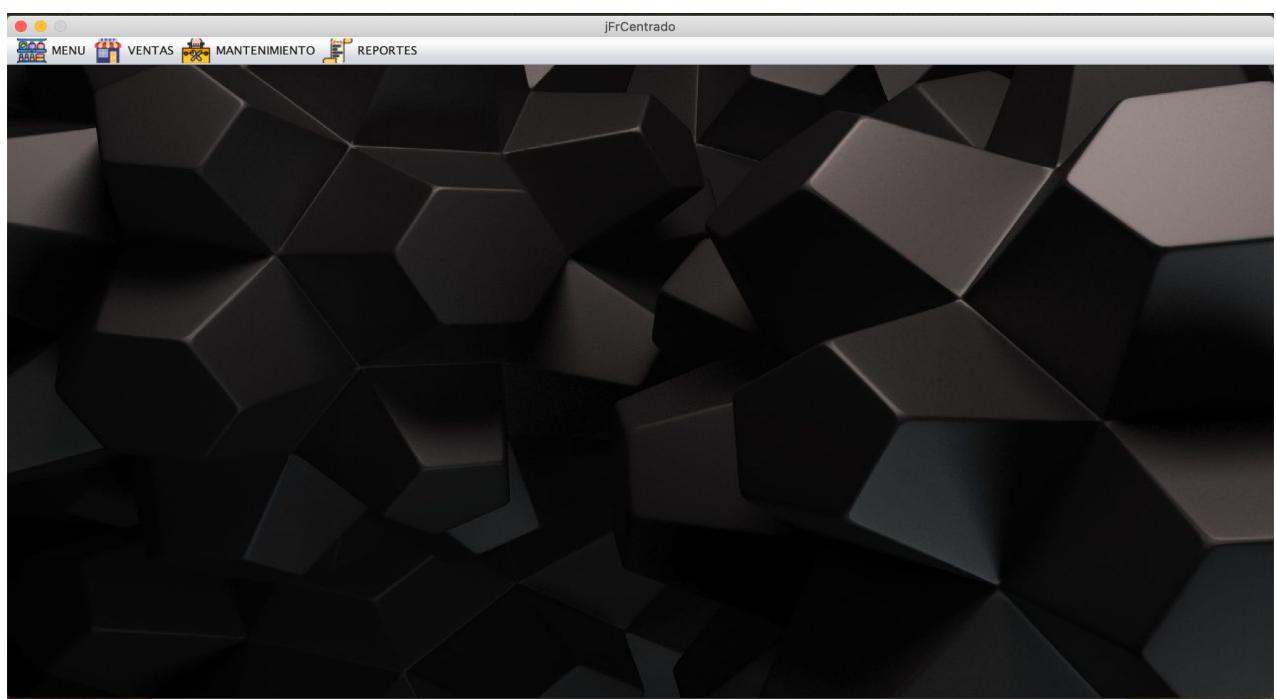
Responsable: Julio Mario Torres - Kevin Leandro R.

Fecha: 15 de noviembre de 2022

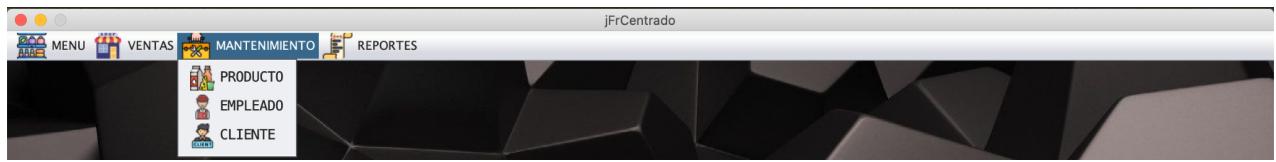
Duración: 6 horas

Una vez el usuario logra pasar la validación del ‘Formulario Login’, se debe mostrar en pantalla la ventana principal asociada a la clase de tipo JFrame en el paquete ‘Vistas’ con nombre ‘VentanaPrincipal.java’.

Para ello con la ayuda de la herramienta GUI de IDE Netbeans, se realizó la interfaz gráfica para la ventana principal del sistema de ventas:



Como se puede observar, en la parte superior de la ventana principal se muestran diferentes apartados, que cada uno contiene sus módulos correspondientes. Al momento en que el usuario pasa el puntero por encima de los apartados se despliegan los módulos asociados a cada apartado, de la siguiente manera: *ejemplo con el apartado ‘MANTENIMIENTO’*



2.4 Etapa 4 - Elaboración apartado ‘MANTENIMIENTO’ - Módulo Cliente

Responsable: Kevin Leandro R.

Fecha: 16 de noviembre de 2022

Duración: 4 horas

Cada módulo tiene asociado una `JInternalFrame` de la librería `java.swing`. Allí se crean los objetos de las clases que están en el paquete ‘*Modelo*’ ya sean de tipo DAO clases relacionadas a cada tabla creada en la base de datos.

Con esto en mente, para el módulo de cliente, en el paquete ‘*Modelo*’ se crearon las clases ‘*TablaCliente.java*’ y ‘*ClienteDao.java*’, en donde se implementan los métodos correspondientes para capturar lo ingresado en las cajas de texto y hacer las validaciones correspondientes.

Con la ayuda del GUI de Netbeans, se creó la siguiente interfaz gráfica que le da vida al módulo Cliente: (*Clase creada en el paquete ‘Vistas’ con nombre ‘ClienteForm.java’*)



MODULO CLIENTES



ID	Cédula	Nombre	Dirección	Estado
1	5892452	Bernardo Gomez	Crra 33B-No 97-28	1
2	897568	Laura Rendon nn	Barrio Manzanares	1
3	7549868	Pepito Gomez	Barrio Palermo	1
4	6882567	Juan Gonzalez Mendieta	Barrio Varsovia	1
5	678236	Juana rrsf	Barrio Apostolico	1
6	773692	Juanito Maraña	Fabela 13 de julio	1
12	34848	Densel Washinton	EEUU	1
13	859392	Kelly Dahiana	Cable	1
15	88382032	Maicol Mendoza	Cra 66 No 89-23	1
16	523299	Gloria Amparo	Conjunto Residencial lo...	1
17	1231231	Ian Ix	La enea Cra 77 no 1231	1
18	123	Empleado de Prueba	UNAL	1
19	75074591	Alejandro Castaño	Crra 9na #10-79 Barri...	1
20	1193206011	Ian Ix Alarcon	Cra 36A #95-09	1
21	1002608925	Leandro Garcia	Barrio Malabar	1

Esta interfaz consta de varios objetos heredados de la clase `java.swing`. Entre ellos cajas de texto, botones y una tabla que está asociada a los registros de la tabla '`CLIENTE`' en la base de datos que respalda el sistema de ventas.

Fecha: 17 de noviembre de 2022

Responsable: Kevin Leandro R.

Duración: 3 horas

Para el mantenimiento de la tabla '`CLIENTE`' se implementó una interface en el paquete '`Modelo`' con nombre '`CRUD.java`' en donde se crearon los métodos de listar, agregar, actualizar y eliminar. Esta interface se implementa en las clases en donde se quiere realizar el mantenimiento crud del módulo correspondiente.

Botón 'AGREGAR': Cuando el usuario da click sobre este botón con nombre

'`btnAgregar`' se invocan los métodos de '`agregar()`', '`limpiarTabla()`', '`listar()`', '`nuevo()`'.

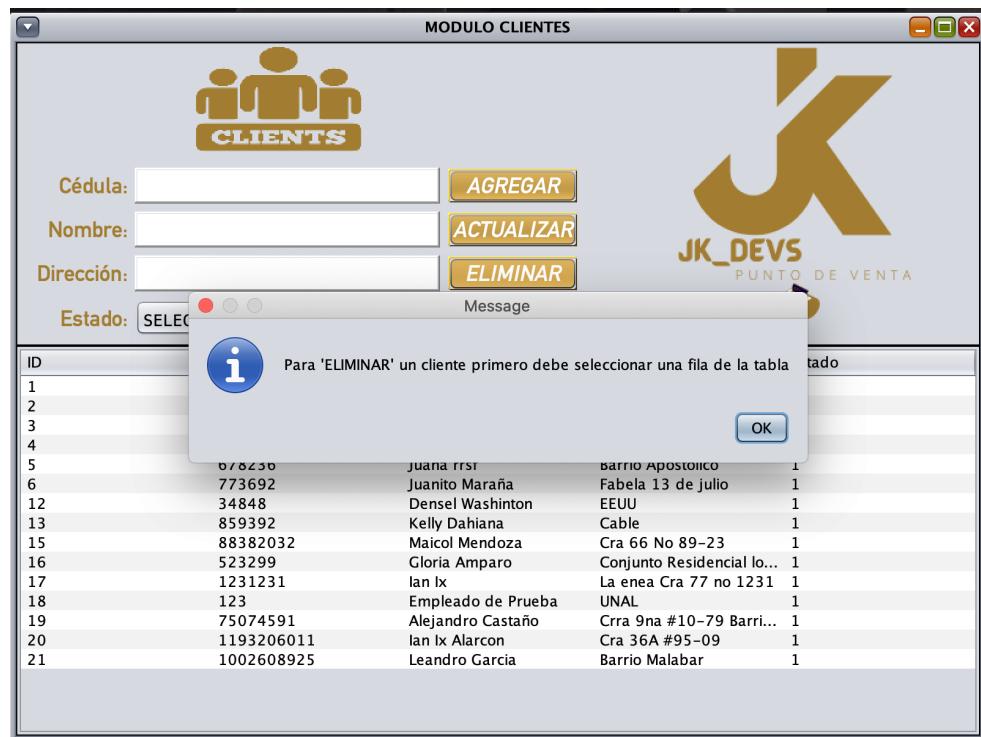
'`agregar()`': Para ello se crea un arreglo de tipo '`Object[]`' en donde a cada posición se le asigna el cuadro de texto correspondiente y se crea un objeto de tipo DAO y se llama la función '`agregar(Object[] obj)`' que recibe como parámetro el arreglo de objetos, al cuál se le aplica la consulta '`INSERT INTO nombre_tabla`' según el '`idCliente`'.



Botón ‘ACTUALIZAR’: Cuando el usuario da click sobre este botón con nombre ‘btnActualizar’ primero se valida que haya escogido con el mouse una fila de la tabla que aparece en la interfaz gráfica, en caso contrario se despliega una ventana emergente solicitando que primero escoja un registro en la tabla ‘jTabla’. Una vez hecha esta validación, se procede a llamar a los métodos ‘actualizar()’ y según el ‘idCliente’ ingresado en la caja de texto correspondiente se hace el mismo procedimiento que en el botón agregar.

Botón ‘ELIMINAR’: Se realiza el mismo procedimiento que en el botón actualizar, ya que para eliminar un registro también se debe haber escogido una fila en la ‘jTabla’

Botón ‘NUEVO’: Aquí no se realiza ninguna validación, ya que este método consiste en limpiar todos las cajas de texto para que el usuario pueda volver a llenarlas para registrar un nuevo cliente.



ejemplo al darle click al botón ‘ELIMINAR’ sin haber seleccionado una fila en la ‘jTabla’



2.5 Etapa 5, 6 - Elaboración apartado ‘MANTENIMIENTO’ - Módulo Producto y Empleado

Responsable: Julio Mario Torres

Fecha: 18, 19, 20 de noviembre de 2022

Duración: 7 horas

Para la elaboración de estos otros dos módulos, se realizó el mismo proceso que se hizo para el módulo ‘CLIENTES’, con el mismo tipo de validaciones.

Lo único que cambia es que para el módulo ‘PRODUCTO’ se puede agregar la columna referente al ‘*idProduct*’ el cual es la llave primaria de la entidad/tabla PRODUCTO registrada en la base de datos. Para ello se hizo una validación para que el usuario no pueda ingresar un producto con una ‘*idProduct*’ ya registrada.



MODULO PRODUCTO


Cod. Producto:
Nombre:
Precio Venta:
Estado:
Stock:

AGREGAR




Codigo	Nombre	Precio	Stock	Estado
1	Leche COLANTA	4200.0	72	1
2	Speed Max 269ml	1300.0	197	1
3	Leche Alpina MAXILITRO	4200.0	94	1
4	Bianchi Actualizado	200.0	66	1
5	Chocorramo	2000.0	96	1
6	Azucar UNAL 500g	3200.0	97	1
7	Arroz UNAL 250g	3800.0	98	1
8	Cigarrillos Lucky (1/2)	4500.0	90	1
9	Cubeta Huevos x24	15000.0	30	1
10	Harina UNAL 250g	1800.0	40	1
11	Aceite UNAL 500ml	7800.0	43	1
3982	Festival paq(x12)	1000.0	30	1
8979	Paquete de glacitas x12	5500.0	40	1

Módulo PRODUCTO



MODULO EMPLEADO

Cédula:	<input type="text"/>	AGREGAR
Nombre:	<input type="text"/>	ACTUALIZAR
Tel:	<input type="text"/>	ELIMINAR
Estado:	SELECCIONAR	NUEVO
Username:	<input type="text"/>	
Password:	<input type="text"/>	





ID	CÉDULA	NOMBRE	TELÉFONO	ESTADO	USERNAME	PASSWORD
1	1110599607	Julio Mario	3203576262	1	emple1	1234
2	589453	Ix Mendieta	3006534194	1	emple2	999
3	77929	Manuela	3209345533	1	emple001	001
5	1110599607	Nelson Bonilla	3177405051	1	nelson01	1234
7	30333990	Leonora Jaramillo	3204726396	1	leonor123	fabiolo123
8	30333990	Leonora Jaramillo	3203564178	1	leonor123	fabiolo123

Módulo EMPLEADO



2.6 Etapa 7 - Elaboración interfaz gráfica apartado 'VENTAS' - Módulo 'Generar Venta'

Responsable: Julio Mario T.

Fecha: 21, 22 de noviembre de 2022

Duración: 5 horas distribuidas en los dos días

Con la ayuda del GUI de Netbeans, se creó la siguiente interfaz gráfica correspondiente a la clase JInternalFrame creada en el paquete 'Vistas' con nombre 'VentasForm.java':

SISTEMA DE VENTAS

PUNTO DE VENTA
Cra. 25 #62-236, Manizales, Caldas
Tel. 3203576262
No. Serie: 000000010

UNAL-MARKET

CLIENTE

PRODUCTO

STOCK

VENDE Julio Mario

NUM. SERIE	COD	PRODUCTO	CANTIDAD	PRE. UNITARIO	SUBTOTAL

CANCELAR **GENERAR VENTA** **TOTAL A PAGAR**

Módulo 'Generar Venta'



2.7 Etapa 8 - Elaboración métodos apartado 'VENTAS' - Módulo 'Generar Venta'

Responsable: Julio Mario T.

Fecha: 23 - 26 de noviembre de 2022

Duración: 12 horas repartidas en el intervalo de días trabajado.

Ya hecha la interfaz gráfica, procedimos a cambiarle el nombre a todos los objetos generados en la construcción de la interfaz. Hay ciertos aspectos a recalcar:

- Este módulo está pensado para que el usuario (que siempre va a ser un empleado) pueda consultar al cliente según su 'cedCliente', en donde se hace la validación con el método 'buscarCliente()'. Si la cédula ingresada no corresponde a ningún cliente registrado dentro de la tabla 'CLIENTE', se abre una ventana emergente en donde se le pregunta al usuario si desea registrar un nuevo cliente.

Si da click en 'YES', se le redirecciona al módulo 'CLIENTE' para que registre ese nuevo cliente según los datos correspondientes.

- Una vez completado el proceso de 'buscarCliente()' el usuario debe llenar el cuadro de texto asociado al 'codProduct', se hace la validación si ese código de producto se encuentra registrado. Si el código SI corresponde a un registro de la tabla 'PRODUCTO' se muestran los atributos asociados a ese producto en los cuadros de texto: 'jTxtProduct' - 'jTxtStock' - 'jTxtPrecio'.

- Posteriormente se le pide al usuario que escoja la cantidad de ítems que se van a facturar o que el cliente desea comprar en 'jSpinCantidad'



4. Una vez seleccionado el producto y la cantidad de ítems que el cliente desea comprar, se puede proceder a dar click en el jButton ‘AGREGAR’

NUM. SERIE	COD	PRODUCTO	CANTIDAD	PRE. UNITARIO	SUBTOTAL
1	1	Leche COLANTA	0	4200.0	0.0
1	3	Leche Alpina MAXI...	2	4200.0	8400.0
1	6	Azucar UNAL 500g	4	3200.0	12800.0
1	8	Cigarrillos Lucky (...)	2	4500.0	9000.0

‘jTablaDetalle’ cuando se agregan productos

Botón AGREGAR módulo Ventas: Al activar el evento en este botón, se ejecuta el método ‘agregarProduct()’ de la clase ‘VentasForm.java’.

En el método ‘agregarProduct()’ se agregan los atributos de cada producto en la tabla ‘jTablaDetalle’, esto se hace capturando los datos asociados a las cajas de texto correspondientes a la información del producto y guardando esos valores en un arreglo de tipo ArrayList<>. También se valida si el stock del producto solicitado es mayor a cero y por último se va agregando cada producto a la tabla ‘jTablaDetalle’

*NOTA: al agregar los productos a la tabla se calcula el subtotal que corresponde a la multiplicación de la cantidad de productos que se van a facturar por el precio de cada producto: subtotal = cantProduct * precioProduct*

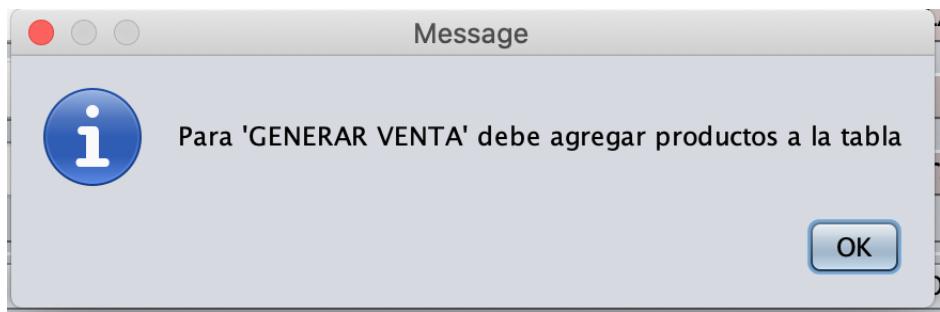
5. El total del precio de venta se calcula y se muestra en el cuadro de texto ‘jTxtTotalPagar’, esto se recorriendo toda la ‘jTablaDetalle’ con el método ‘calcularTotal()’.

```
//Se recorre cada fila de 'jTablaDetalle' y en cada iteración se calcula
//el precio total a pagar en la compra
void calcularTotal(){
    totalCompra = 0;
    for(int i = 0; i < jTablaDetalle.getRowCount(); i++){
        cantProducto = Integer.parseInt(jTablaDetalle.getValueAt(i, 3).toString());
        precioProduct = Double.parseDouble(jTablaDetalle.getValueAt(i, 4).toString());
        totalCompra = totalCompra + (cantProducto * precioProduct);
    }
    jTxtTotalPagar.setText(String.valueOf(totalCompra));
}
```

Como se puede observar, se hace uso del método ‘.setText()’ para mostrar el precio total de la venta cada vez que se agregue un producto a la ‘jTablaDetalle’

6. Cuando ya se hayan registrado todos los productos que el cliente desea comprar, se da click en ‘GENERAR VENTA’

Botón GENERAR VENTA módulo Ventas: Primero se valida si hay productos agregados a la tabla, sino se muestra una ventana emergente.



Si, si hay productos agregados a la tabla se ejecutan los siguientes métodos:
‘guardarVenta()’, ‘guardarFactura()’, ‘actualizarStock()’, ‘limpiarFormulario()’,
‘generarNumSer()’

‘guardarVenta()’: Aquí se inicializan las variables con los campos que tiene la tabla ‘VENTA’ en la base de datos. Se crea un objeto de tipo ‘TablaVenta’ y con los métodos ‘setNombAtributo()’ se le asignan los valores correspondientes al tipo de dato. Para luego crear un objeto de tipo ‘DAO’ perteneciente a la clase ‘VentaDAO’ y se llama el método de esa clase llamado ‘guardarVenta(TablaVenta venta)’ que recibe como parámetro un objeto de la clase ‘TablaVenta’ y se manda la consulta a la base de datos para registrar la venta.

	← T →		idVenta	idCliente	idEmple	numSerie	montoVenta	fechaVenta	estadoVenta
<input type="checkbox"/>		Editar		Copiar		Borrar	5	18	1 00000001
<input type="checkbox"/>		Editar		Copiar		Borrar	6	18	1 00000002
<input type="checkbox"/>		Editar		Copiar		Borrar	7	12	1 00000003
<input type="checkbox"/>		Editar		Copiar		Borrar	8	18	1 00000004
<input type="checkbox"/>		Editar		Copiar		Borrar	9	18	1 00000005
<input type="checkbox"/>		Editar		Copiar		Borrar	10	21	1 00000006
<input type="checkbox"/>		Editar		Copiar		Borrar	11	18	1 00000007
<input type="checkbox"/>		Editar		Copiar		Borrar	12	18	1 00000008
<input type="checkbox"/>		Editar		Copiar		Borrar	13	17	1 00000009



En el método 'guardarFactura()' se aplica la misma lógica utilizada para crear el método 'guardarVenta()'

	idFactura	idVenta	idProduct	cantProductos	precioVenta
<input type="checkbox"/> Editar Copiar Borrar	17	5	4	3	200
<input type="checkbox"/> Editar Copiar Borrar	18	7	1	2	4200
<input type="checkbox"/> Editar Copiar Borrar	19	7	5	4	2000
<input type="checkbox"/> Editar Copiar Borrar	20	7	3	2	4200
<input type="checkbox"/> Editar Copiar Borrar	21	8	6	2	3200
<input type="checkbox"/> Editar Copiar Borrar	22	8	7	4	3800
<input type="checkbox"/> Editar Copiar Borrar	23	9	1	6	4200
<input type="checkbox"/> Editar Copiar Borrar	24	10	1	6	4200
<input type="checkbox"/> Editar Copiar Borrar	25	11	5	4	2000
<input type="checkbox"/> Editar Copiar Borrar	26	11	6	3	3200
<input type="checkbox"/> Editar Copiar Borrar	27	11	3	6	4200
<input type="checkbox"/> Editar Copiar Borrar	28	12	1	2	4200
<input type="checkbox"/> Editar Copiar Borrar	29	13	2	3	1300
<input type="checkbox"/> Editar Copiar Borrar	30	13	7	2	3800
<input type="checkbox"/> Editar Copiar Borrar	31	13	10	4	1800
	Consola				

En la tabla 'FACTURA' se almacenan todos los productos agregados a la tabla y al momento de dar click en el botón 'GENERAR VENTA' se registran los datos en la tabla.

'actualizarStock()': En este método se utiliza un ciclo `for()` para recorrer todas las filas de la tabla 'jTablaDetalle' en donde se captura la columna asociada a la cantidad de items que se van a facturar por producto.

Se crea un objeto de la clase 'TablaProducto' al cual se le asignan los valores de 'idProduct' y 'cantProduct' para luego llamar al método de la clase 'ProductoDAO' (`productDao.actualizarStock(stockActualizado, idProduct)`) que recibe como parámetros el stock actualizado correspondiente a la cantidad de items almacenados del producto menos la cantidad de el producto que se va a facturar.

```
void actualizarStock(){
    for(int i = 0; i < modelo.getRowCount(); i++){
        TablaProducto productAux = new TablaProducto();
        idProduct = Integer.parseInt(jTablaDetalle.getValueAt(i, 1).toString());
        cantProducto = Integer.parseInt(jTablaDetalle.getValueAt(i, 3).toString());
        productAux = productDao.listarId(idProduct);
        int stockActualizado = productAux.getStockProduct() - cantProducto;
        productDao.actualizarStock(stockActualizado, idProduct);
    }
}
```

método de la clase tipo JInternalFrame llamada 'VentasForm.java'



```
18
19     public int actualizarStock(int cantProduct, int idProduct){
20         String sql = "UPDATE PRODUCTO SET stockProduct=? WHERE idProduct=?";
21
22         try {
23             acceso = conex.conectarBD();
24             preStat = acceso.prepareStatement(sql);
25             preStat.setInt(1, cantProduct);
26             preStat.setInt(2, idProduct);
27             resp = preStat.executeUpdate();
28         } catch (Exception e) {
29         }
30     return resp;
31 }
```

método de la clase ProductoDAO

'limpiarFormulario()': Se limpian todas las cajas de texto del formulario 'VentasForm.java' y también se limpia la tabla 'jTablaDetalle'

'generarNumSer()': Cada vez que se genera una venta se actualiza la caja de texto asociada al número de serie de la venta.



7. La caja de texto 'jTxtVende' asociada al nombre de usuario del empleado que tiene activa la sesión es capturado de la 'TablaEmpleado'. Llamando al método 'generarNombEmple()'

```
59     void generarNombEmple(){
60         nombreEmpleSesionActual = logForm.nombEmpleSesionActual();
61         jTxtVende.setText(""+nombreEmpleSesionActual);
62     }
```

método de la clase 'VentasForm.java'

8. La caja de texto 'jTxtFecha' asociada a la fecha actual es generada por la clase 'Calendar' a través del método 'generarFecha()'

```
64     void generarFecha(){
65         Calendar calendar = new GregorianCalendar();
66         jTxtFecha.setText(""+calendar.get(Calendar.YEAR)+"/"+calendar.get(Calendar.MONTH)+"/"+calendar.get(Calendar.DAY_OF_M
67     }
68 }
```



3. Glosario

Software: Se llama así a todos los programas adicionales que se incorporan al computador, dotado ya de un sistema operativo, con el propósito de llevar a cabo un sinfín de tareas posibles: desde procesadores de texto, hojas de cálculo, navegadores de internet, aplicaciones de diseño o videojuegos, nuestro proyecto precisamente es un software de ventas y control de inventario.

Lenguaje de programación: Un lenguaje de programación es un lenguaje de computadora que los programadores utilizan para comunicarse y para desarrollar programas de software, aplicaciones, páginas webs, scripts u otros conjuntos de instrucciones para que sean ejecutadas por los ordenadores en nuestro caso trabajamos con el lenguaje Java.

Automatización: La automatización consiste en usar la tecnología para realizar tareas con muy poca intervención humana. Se puede implementar en cualquier sector en el que se lleven a cabo tareas repetitivas. Sin embargo, es más común en aquellos relacionados con la fabricación, la robótica y los automóviles, así como en el mundo de la tecnología: en el software para la toma de decisiones empresariales y los sistemas de TI, precisamente buscamos brindarle algo más de automatización a la gestión de inventario y control de ventas.

Sistema Gestor de Bases de Datos (SGBD): Un Sistema Gestor de Base de Datos (SGBD) o DataBase Management System (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible para nuestro proyecto fue indispensable la creación de una base de datos en la cual guardamos toda la información necesaria para funcionamiento correcto de nuestro proyecto.

Modelo entidad relación: El modelo entidad relación (E/R) proporciona una herramienta para representar información del mundo real a nivel conceptual. Creado en 1976 por Peter Chen, permite describir las entidades involucradas en una base de datos, así como las relaciones y restricciones de ellas, así mismo pudimos exportar entre diferentes tablas de nuestra base para realizar todas las operaciones necesarias en una venta.

Lenguaje de consulta MySQL: MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle, este mismo fue el usado en la elaboración de nuestra base de datos.



Java: Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones piedra base en la creación de nuestro proyecto.

IDE: Un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Aumenta la productividad de los desarrolladores al combinar capacidades como editar, crear, probar y empaquetar software en una aplicación fácil de usar.

GUI: El acrónimo GUI proviene de “Graphic User Interface” o Interfaz Gráfica de Usuario. Es la abreviatura que define la interfaz gráfica de cualquier aplicación, la cual fue indispensable para la creación de la interfaz gráfica del proyecto.

Herramientas de la GUI: Las GUI utilizan elementos gráficos como iconos, menús e imágenes para facilitar el manejo del usuario humano. Tanto los sistemas operativos como las aplicaciones utilizan una interfaz gráfica de usuario. De hecho, casi todos los programas para usuarios finales de hoy en día vienen con esta interfaz.

Stock: El stock se define como la mercancía que se almacena en un almacén u otro tipo de local para ser vendida o comercializada. Esta mercancía está formada por producto acabado, disponible para ser servido.



4. Referencias

- Dapp Code. (2019). Java MySQL NetBeans POS inventory System Full project with scr code #Dappcode. Lista de reproducción Youtube.
- TutusFunny. (2019). Inventory Management System step by step in Java. Lista de reproducción YouTube.
- Dapp Code. (2022). Simple pos system for small business with Bill Printing in Java | Touch screen POS.
- Guia definitiva de conversión de tipos en Java. (s/f). Stack Overflow en español. Recuperado el 30 de noviembre de 2022.
- Apache friends support forum. (s/f). Apachefriends.org. Recuperado el 30 de noviembre de 2022.
- Newest “point-of-sale” questions. (s/f). Stack Overflow. Recuperado el 30 de noviembre de 2022.