Reporte del proyecto

Sistemas de Bases de Datos

Luis Angel Arizaga Díaz

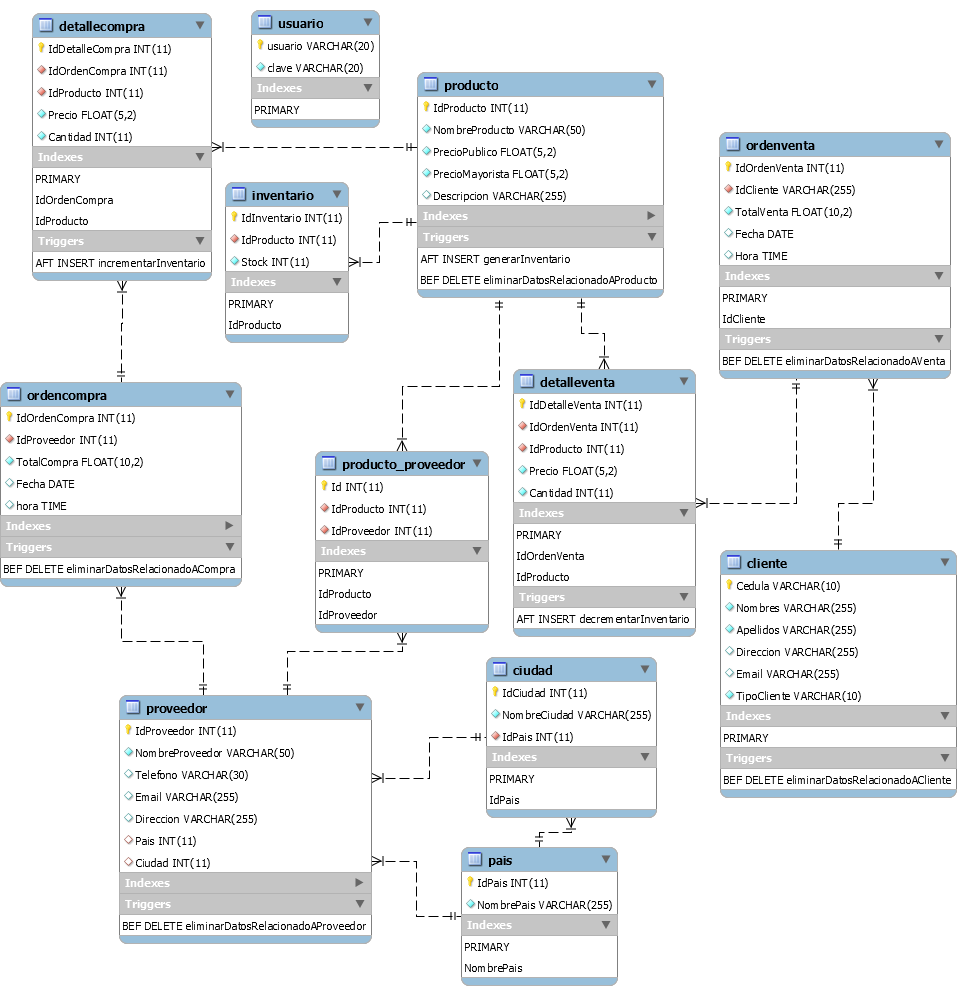
Lucio Israel Arias Sela

Leonardo Andre Castro Alban

**Herramientas utilizadas:**

* DBMS: MySQL Workbench
* Lenguaje de programación: JAVA
* Biblioteca para interfaz gráfica: JAVA SWING
* Driver de conexión Java a MySQL: mysql-connector-java-5.1.43-bin
* IDE de desarrollo de la aplicación: NetBeans
* Sistema de repositorio: GitHub

**Base de datos**



1. **Tablas:**

Está conformada por las tablas:

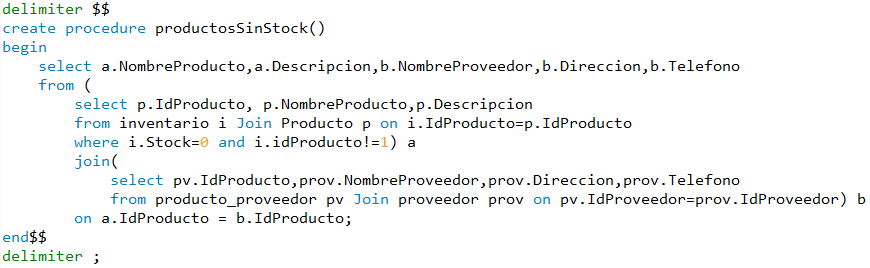
|  |  |  |  |
| --- | --- | --- | --- |
| Ciudad | Cliente | DetalleCompra | DetalleVenta |
| Inventario | OrdenCompra | OrdenVenta | País |
| Producto | Producto\_Proveedor | Proveedor | Usuario |

* Ciudad: Información de las ciudades pertenecientes a ciertos países.
* Cliente: Personas naturales que a los que se les venderá productos.
* DetalleCompra: Detalle de lo que se ha comprado.
* DetalleVenta: Detalle de lo que se ha vendido.
* Inventario: Información del stock de un producto.
* OrdenCompra: Registro de una compra realizada.
* OrdenVenta: Registro de una venta realizada.
* País: Países a los que pertenecerán los proveedores.
* Producto: Artículos que estarán disponibles para la venta.
* Producto\_Proveedor: Determina los proveedores de los productos.
* Proveedor: Compañías que proveerán de productos a nuestra empresa.
* Usuario: Usuarios con contraseñas disponibles para iniciar sesión en la aplicación.

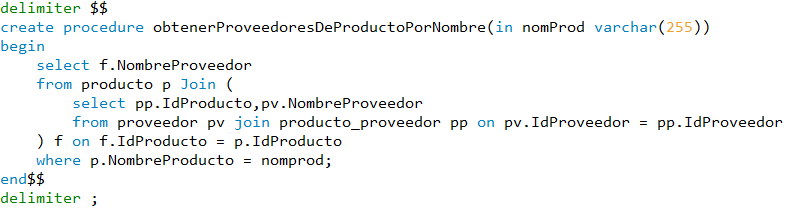
1. **Procedimientos almacenados**

Se crearon un total de 56 procedimientos almacenados (Stored Procedures) para la obtención de los datos desde la base. Estos facilitan la creación, realización y presentación de consultas en la aplicación. Por ejemplo:

* Consultar productos sin stock:



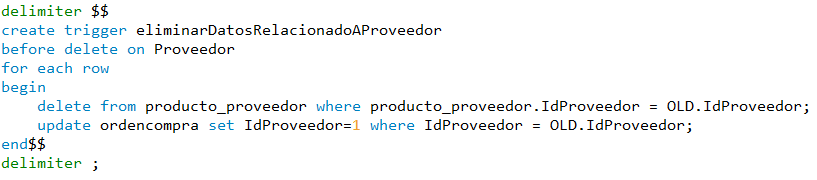
* Obtener los proveedores de cierto producto por el nombre del producto:



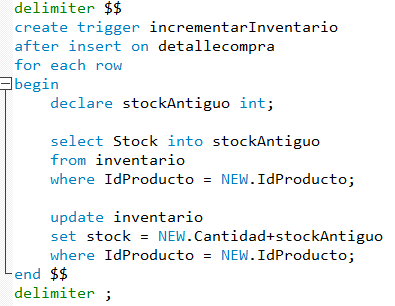
1. **Triggers**

Para cumplir con el objetivo funcional de la aplicación, y tomando en cuenta las diferentes operaciones disponibles en la aplicación (compras, ventas, incremento-decremento de stock, eliminación de datos) se decidió implementar triggers para facilitar estas actividades y que se ejecuten inmediatamente cuando se las necesite. Para esto, también, se añadió campos en las tablas, con valores desconocidos, para que tomen el lugar de los datos que son borrados. Por ejemplo:

* Eliminación de datos relacionados a un proveedor que va a ser borrado.



* Incremento de inventario al momento que se hace una compra del mismo



1. **Índices**

Para la facilitar y optimizar las búsquedas en la base de datos (Producto, Proveedor, Cliente), se crearon índices en los campos en donde se van a enfocar las búsquedas.

* Índice único en la cedula del cliente



* Índice en el nombre del producto



**Aplicación**

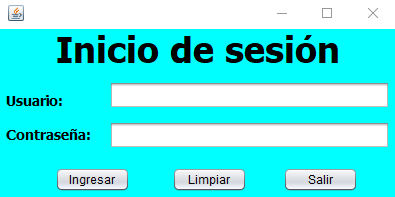
1. **Clases**

Para completar nuestro objetivo se crearon un total de 29 clases. Once de dichas clases son el esqueleto de nuestra aplicación ya que son quienes van a representar los datos en la base de datos como objetos en Java. Una de ellas es el **Main** el cual va a ejecutarse primero y va dar inicio a la aplicación. Quince de las restantes se encargan de la interfaz. Y la última, y la más importante, se encarga de conectarse directamente con la base de datos que le indiquemos con el url y ejecuta los procedimientos almacenados en la base para la creación, lectura, actualización y eliminación de datos (☻).

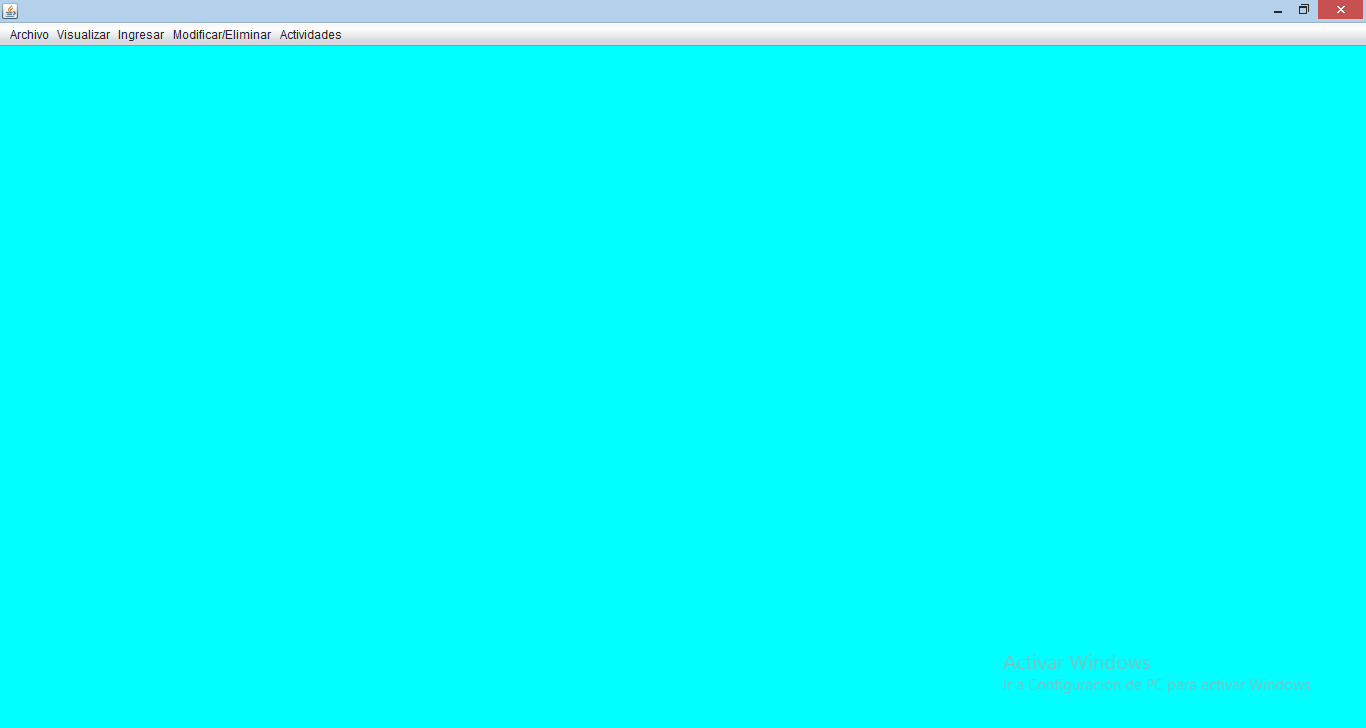
|  |  |  |
| --- | --- | --- |
| **Paquete**: | Parchapp | Parchapp.interfaz |
| **Clases:** | * Ciudad.java * Cliente.java * Connector.java (☻) * DetalleCompra.java * DetalleVenta.java * Inventario.java * OrdenCompra.java * OrdenVenta.java * Pais.java * ParchApp.java (Main) * Producto.java * Proveedor.java * Usuario.java | * Balance.java * Compra.java * Consultasvarias.java * IngresarProveedor.java * IngresoCliente.java * IngresoProducto.java * MenuPrincipal.java * ModEliCliente.java * ModEliProducto.java * ModEliProveedor.java * Sesion.java * Venta.java * VerCliente.java * VerProveedor.java * VisualizarProducto.java |

1. **Interfaz Gráfica**

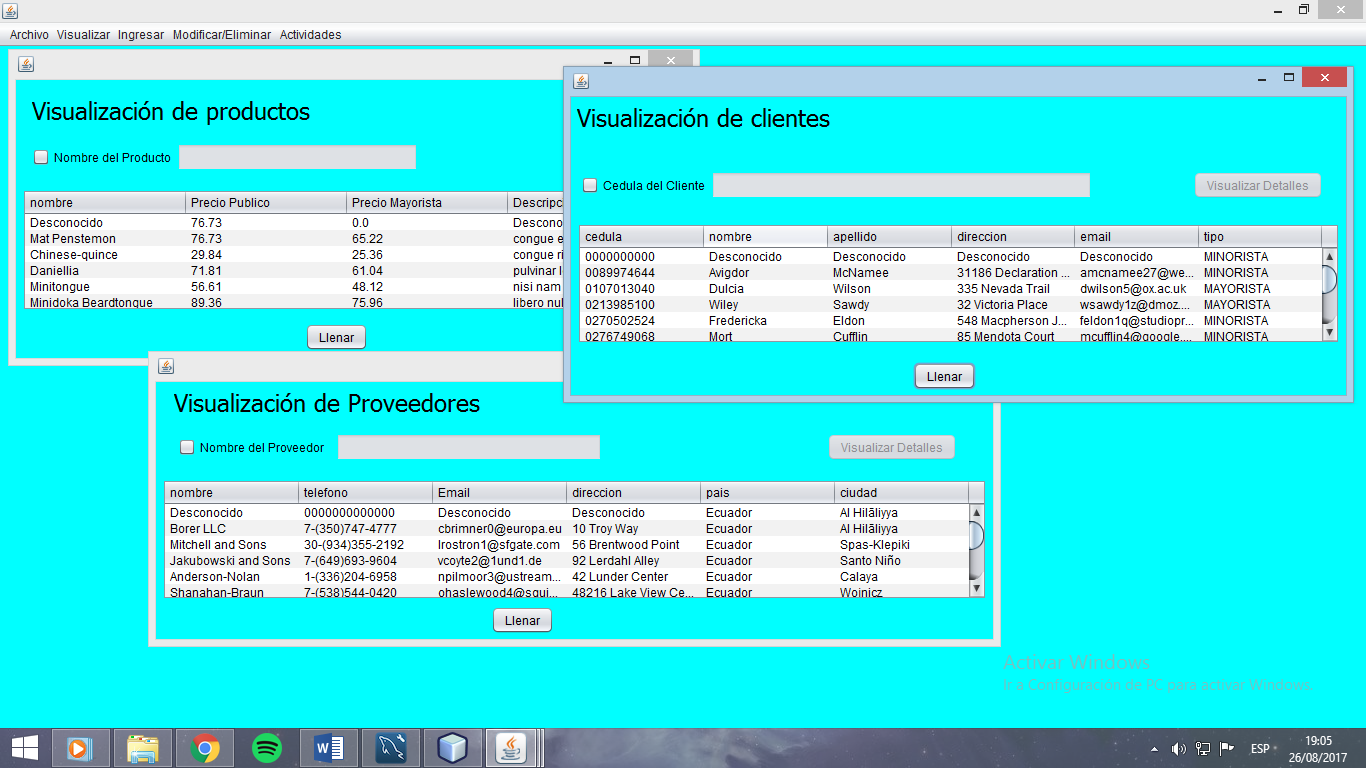
Al ejecutar la aplicación, la primera ventana en aparecer será la de inicio de sesión. Aquí el personal que esté usando la aplicación deberá ingresar su usuario registrado en la base de datos junto con su contraseña.

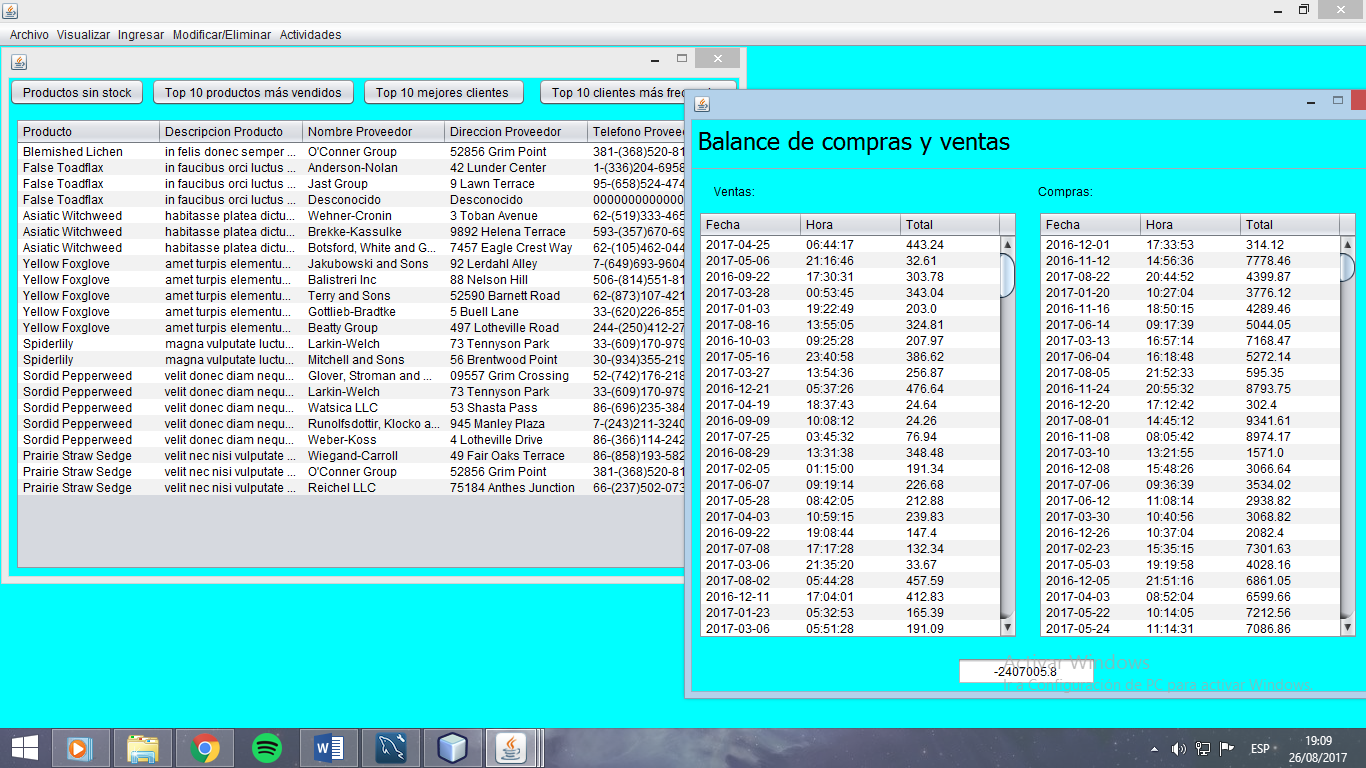


Al iniciar sesión correctamente aparecerá la pantalla del menú principal. Esta ventana contiene una barra de menú para acceder a las diferentes funciones de la aplicación.

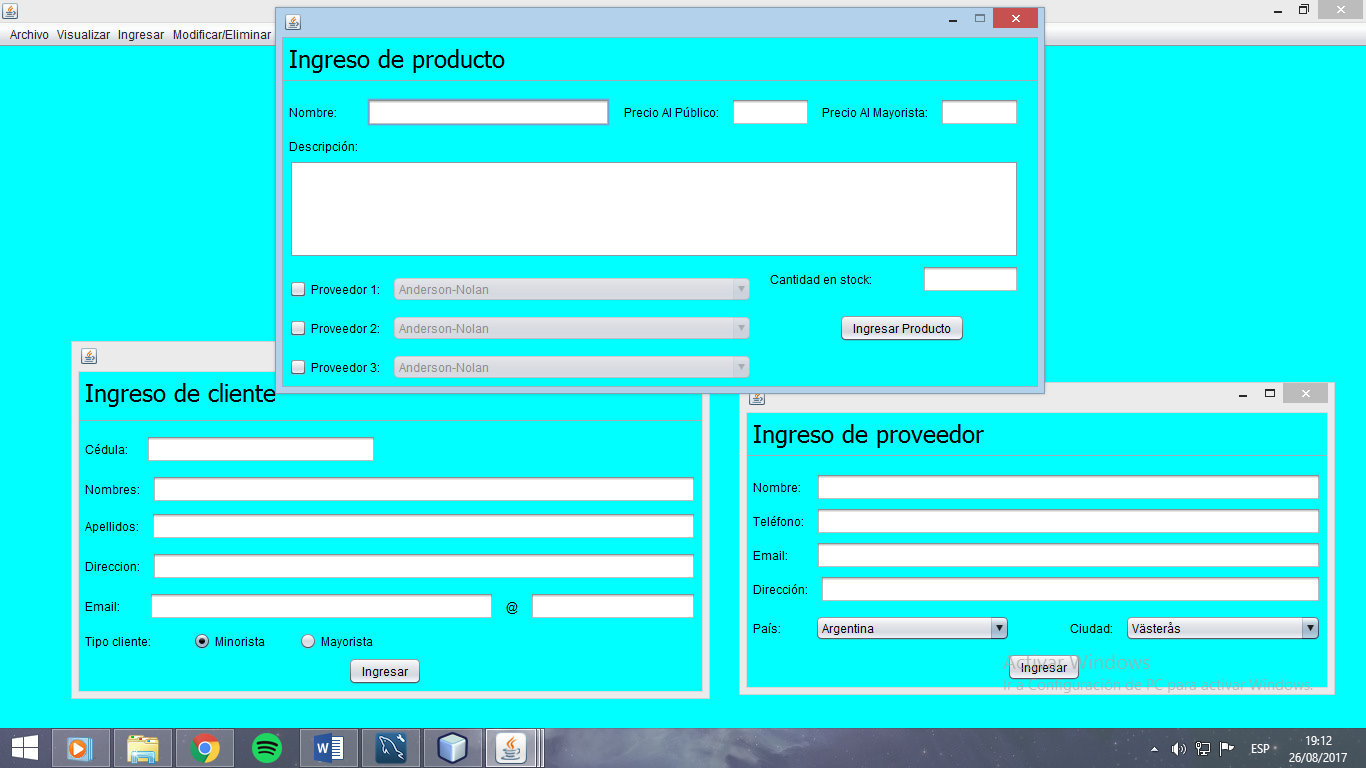


En la pestaña ‘Visualizar’ se podrá acceder a consultas de diferentes sectores de información de la base de datos, así como también podrá observar una opción para consultar un balance histórico de ingresos y salidas de capital, y además una ventana que podrá resolver las preguntas más frecuentes.

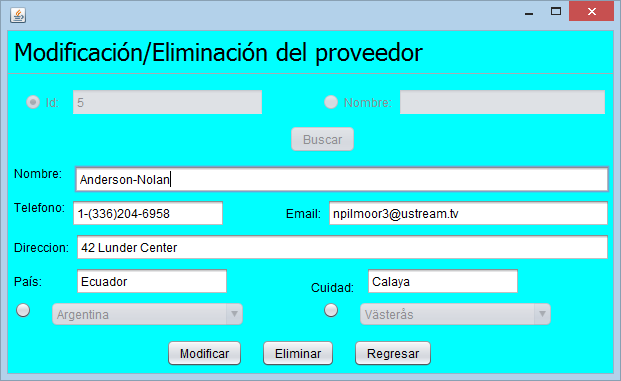




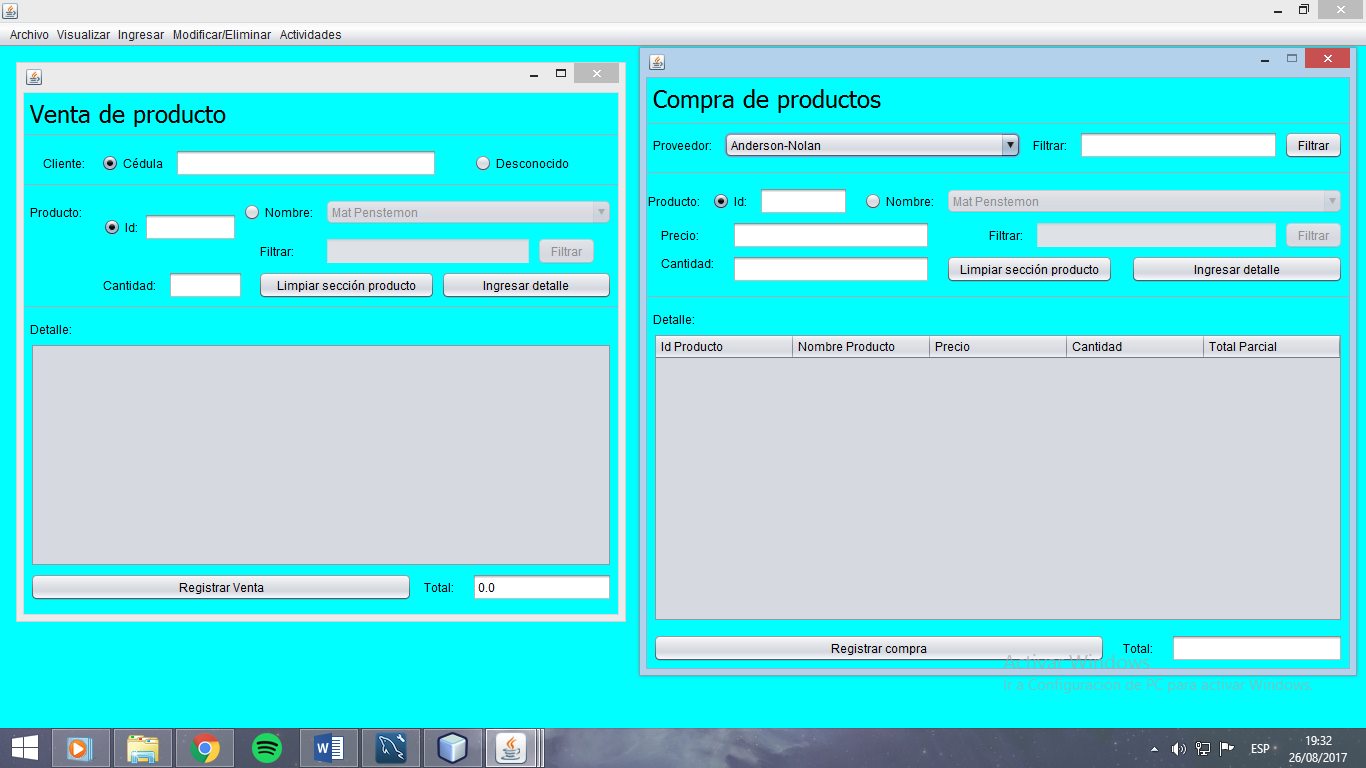
En la pestaña ‘Ingresar’ se puede añadir información, directamente, a la base de datos. Las opciones válidas para el ingreso directo son: Cliente, Producto, Proveedor. Se deben ingresar datos válidos en los diferentes espacios para escribir o el programa invalidará el ingreso de datos.



Si en algún momento se añade un valor erróneo, o se desea eliminar algún dato que se considera no se va a usar más, el sistema brinda una opción para modificar y eliminar registros en la base de datos. Esto se puede hacer sabiendo el nombre específico o el ‘id’ de dicho campo a modificar.



Por otro lado, en la pestaña ‘Actividades’ se podrá registrar las diferentes salidas y entradas de mercadería que ocurren en la empresa. Se puede notar una interfaz gráfica que facilita la interacción del usuario con la aplicación.



**Repositorio**

Se creó un repositorio en GitHub.com para poder llevar un buen trabajo en equipo y tener un respaldo de los cambios realizados en la aplicación.

Link del repositorio: <https://github.com/llullanqui/BasesProyecto>