**PROYECTO | QUANTIFYQ**

**MARCOS DANIEL ROJAS CUERVO  
EDWIN RICARDO ORTEGA CUERVO**

**POLITECNICO INTERNACIONAL**

**PROGRAMACIÓN I**

**HAROLD HERNAN TORRES NEUTA**

**2025**

**Tabla de contenido**

**Resumen Proyecto 1**

**Palabras Claves 2**

**Abstract** **3**

**Introducción Proyecto4**

**Modelo Vista Controlador5**

**Base de datos** **6**

**Desarrollo 7**

**Definición de Conceptos8**

1.Resumen del proyecto.

Este proyecto de QuantifyQ tiene el objetivo de desarrollar una plataforma sencilla, utilizando el lenguaje de programación Java, enfocado en la gestión productos con ingreso de usuarios existentes o que se puedan crear. La aplicación implementará la arquitectura MVC para garantizar una separación clara entre la lógica de negocio, la interfaz gráfica y el manejo de eventos, siguiendo los principios de la POO como abstracción, encapsulamiento, herencia y polimorfismo.

La interfaz de la aplicación estará construida con componentes gráficos como JFrame, JPanel, JLabel, JButton y JTextField, permitiendo una experiencia de usuario intuitiva. En la pantalla de inicio se solicitará un usuario y contraseña para acceder al sistema, si no se encuentra en él sistema tendrá la opción de crear un usuario desde esta parte. Una vez autenticado, el usuario será dirigido al menú principal, desde donde podrá:

* Visualizar, crear, actualizar y eliminar datos de productos del inventario.
* Consultar créditos del sistema, como parte del reconocimiento al desarrollador.

El desarrollo de esta plataforma responde a las historias de usuario definidas previamente y busca cumplir con criterios de aceptación concretos, así como con entregables funcionales y documentados.

2.Palabras Claves.

**Java**: Lenguaje de programación principal del proyecto.  
**NetBeans**: Entorno de desarrollo usado para construir la aplicación.  
**MVC**: Arquitectura recomendada para organizar el proyecto.  
**CRUD**: Las funcionalidades clave del sistema (Crear, Leer, Actualizar, Eliminar).  
**Inventario**: Objetivo principal del proyecto (gestión de productos y existencias).

3.Abstract

This project aims to develop a simple desktop portal using the Java programming language, allowing the management of users and inventory products through a graphical interface. The application is designed under the MVC architecture and applies core OOP principles such as abstraction, encapsulation, inheritance, and polymorphism, promoting good software development practices.

The system allows the creation of users who can access the portal via login credentials and interact with the application through a user-friendly interface built with components such as JFrame, JPanel, JButton, JLabel, and JTextField. Once logged in, users can perform full CRUD operations (Create, Read, Update, Delete) on product records and navigate through features like user management and credit display

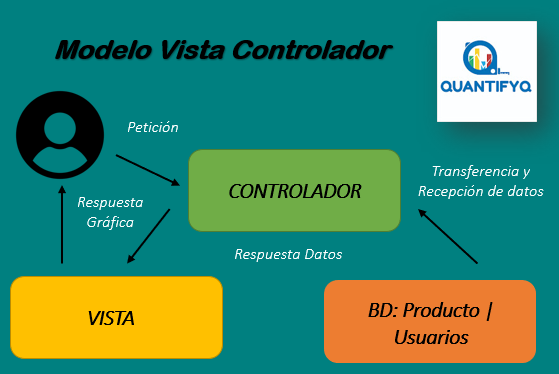
4.Introducción Proyecto

QuantifyQ es una microempresa colombiana en desarrollo desde el año 2025 con el objetivo de ofrecer soluciones tecnológicas y de análisis enfocadas en la optimización de procesos. Desde su creación hacemos la apuesta por la innovación y el desarrollo de herramientas digitales accesibles para pequeñas y medianas empresas que buscan mejorar la trazabilidad y eficiencia en sus operaciones.

El proyecto actual, desarrollado como parte directa a el manejo de inventarios, está diseñado y pensado en una aplicación de escritorio funcional, que permita gestionar productos con ingreso de portal de usuarios, teniendo conexión directa con bases de datos, construyendo un sistema de inventario ágil, intuitivo y adaptable a las necesidades de sus aliados comerciales.

5.MVC

* **Modelo**: Contiene las clases que representan los datos del sistema, como Producto y Usuario. También gestiona la conexión y comunicación con la base de datos (guardar, consultar, actualizar y eliminar registros).
* **Vista**: Incluye todos los elementos gráficos que interactúan con el usuario, como las pantallas de inicio de sesión, menú, el apartado de productos. Estos elementos están construidos con componentes JFrame, JPanel, JTable, JTextField, entre otros.
* **Controlador**: Es el encargado de recibir las acciones del usuario (por ejemplo, hacer clic en “Guardar producto” o “Iniciar sesión”), procesarlas y coordinar la respuesta adecuada entre el modelo y la vista. También valida datos y gestiona el flujo de navegación dentro del sistema.



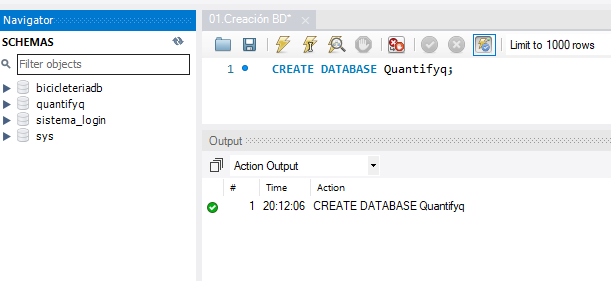
6.Base de datos

Se realiza la creación de la base de datos de QuantifyQ para el almacenamiento de las tablas de usuarios y productos que serán usadas en el desarrollo de la aplicación en Java.

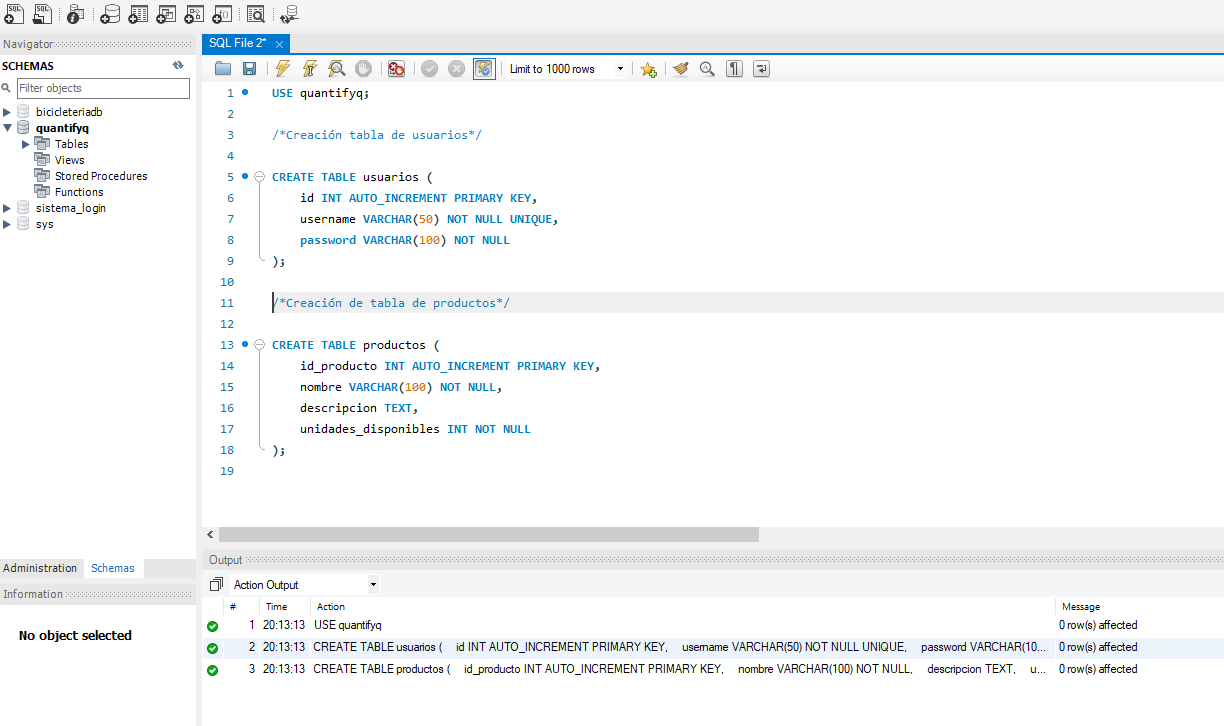
Desarrollo clave de la base de datos:

* Uso de llaves primarias con autoincremental.
* Campos que no reciben nulos.
* Formatos de campos.

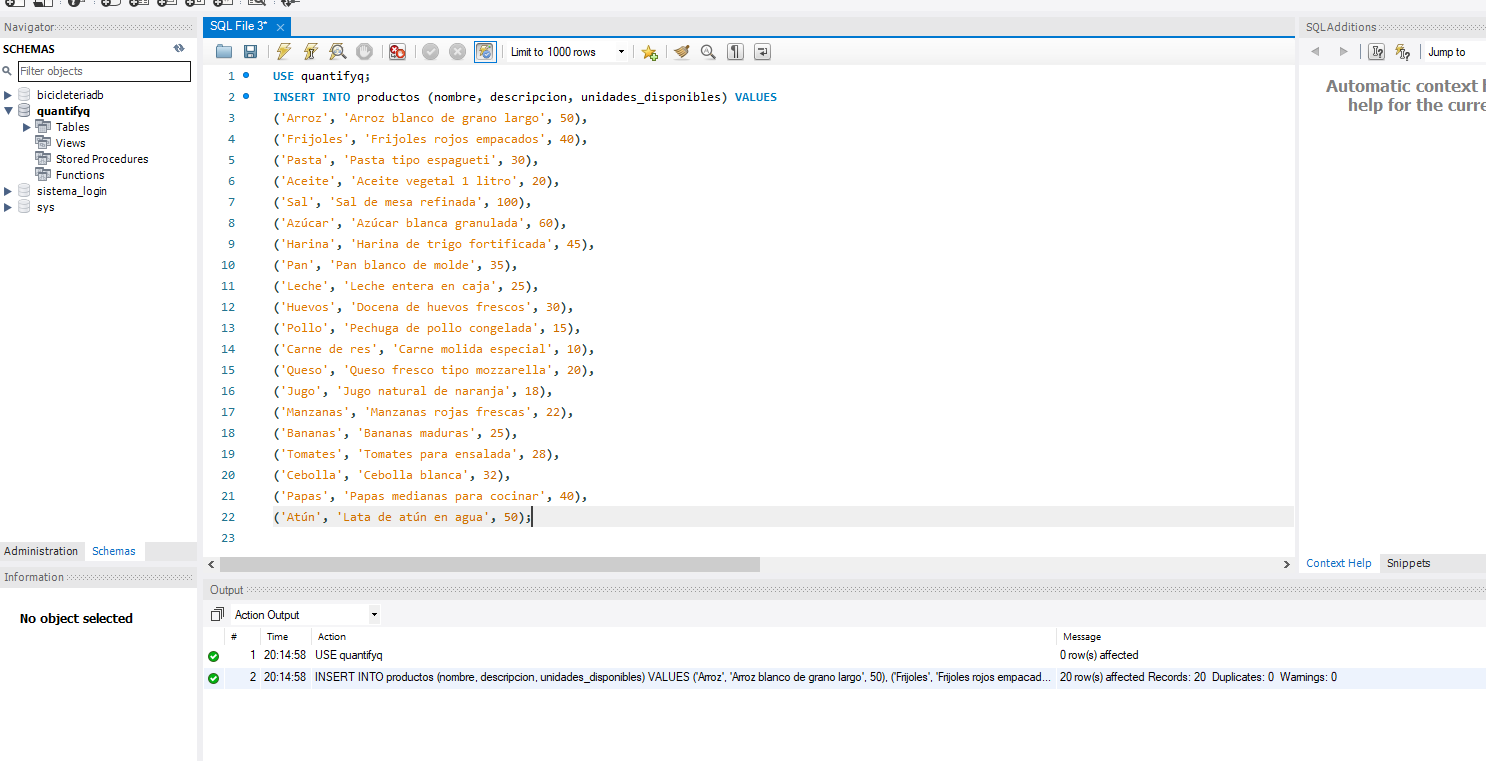
**Creación de la base de datos:**

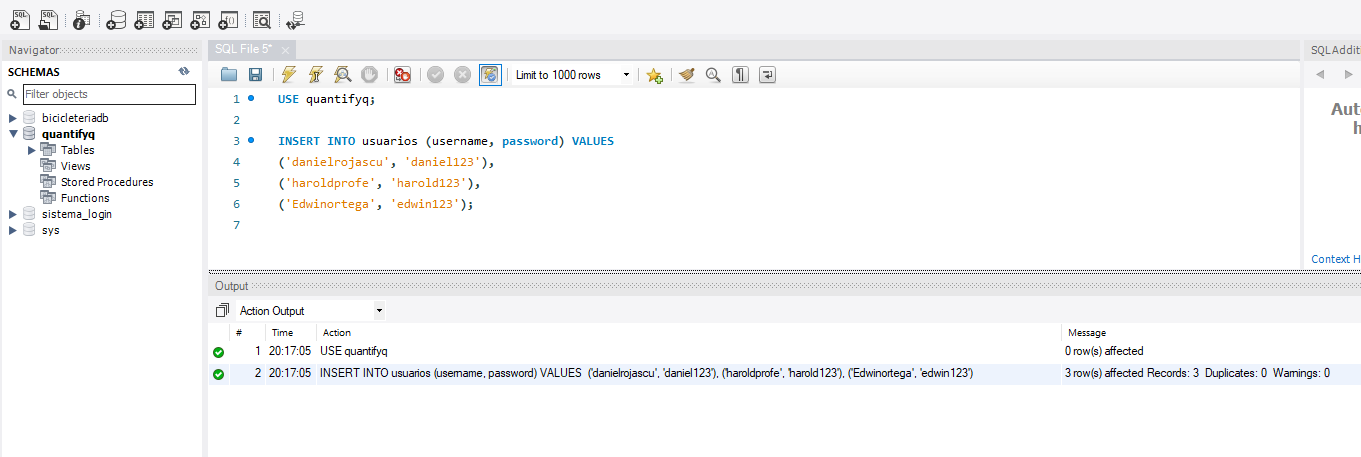


**Creación de las tablas:**

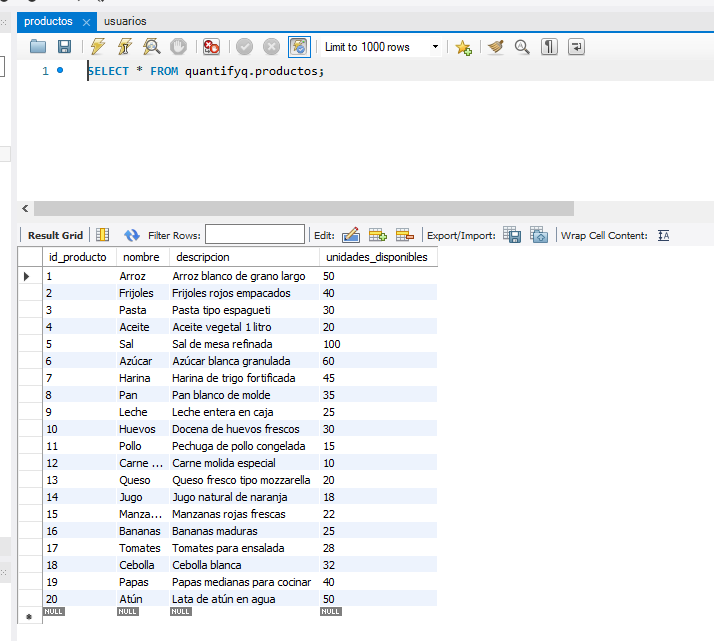


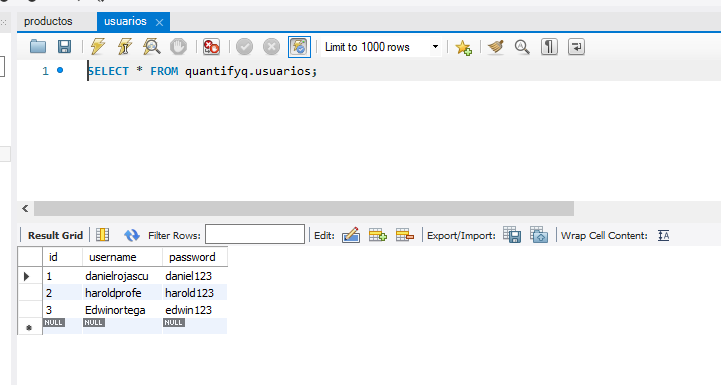
**Ingesta de datos en la tabla de productos y usuarios:**





**Visualización de las tablas en MySQL:**





7.Desarrollo

Proceso….

* Diagrama de Flujo:
* Seudocódigo:
* Diccionario de datos:

8.Definición de conceptos

Proceso….

PPO: (Programación orienta a objetos)

* Abstracción:
* Encapsulamiento:
* Polimorfismo:
* Herencia:
* JFrame:
* JPanel:
* JLabel:
* JButton:
* JTextField: