



Aplicación de Gestión de Inventarios

CICLO: DESARROLLO DE APLICACIONES MULTIPLATAFORMA (DAM)

Autor: Sergio Conde Portero

Tutor: José Carlos Cepero

Periodo: Enero/Marzo 2025



ÍNDICE

TABLA DE CONTENIDOS

Resumen	3
Descripción del Proyecto y su Base de datos.....	4
Justificación del Proyecto y Estructura.....	5
Introducción	8
Objetivos y menú de la aplicación	11
Desarrollo	12
Conclusiones	17
Bibliografía	18
Anexos	19

TABLA DE IMÁGENES

Imagen 1 Diagrama de clases usando la herramienta Postgres. Fuente elaboración propia.....	4
Imagen 2 Estructura de paquetes en Netbeans. Fuente: elaboración propia.....	9
Imagen 3 Librerías utilizadas en el proyecto. Fuente: Elaboración propia.	9
Imagen 4 Uso de XAMPP en un equipo Fuente: Elaboración propia.	10
Imagen 5 Inserción de datos en phpMyAdmin en la tabla "usuarios" Fuente: Elaboración propia.	10
Imagen 6 Login con el usuario admin Fuente:Elaboración propia.....	11
Imagen 7 HOME de la aplicación junto con su menú Fuente: Elaboración propia	11
Imagen 8 Interfaz de CATEGORIAS. Fuente: Elaboración propia	12
Imagen 9 Funcionamiento del botón de guardar del registro categorías Fuente: Elaboración propia	12
Imagen 10 Funcionamiento del botón editar. Fuente: Eleboración propia.....	13
Imagen 11 Ventana emergente de eliminar. Fuente: Elaboración propia.....	13
Imagen 12 Funcionamiento del botón eliminar. Fuente: Elaboración propia	13
Imagen 13 Registro de clientes Fuente: Elaboración propia	14
Imagen 14 Registro de Proveedores. Fuente: Elaboración propia.....	14
Imagen 15 Registro de usuarios. Fuente: Elaboración propia.	15
Imagen 16 Menú desplegable del tipo de usuario. Fuente: Elaboración propia.....	15
Imagen 17 Registro de Productos. Fuente: Elaboración propia	15
Imagen 18 Pestaña del listado de categorías. Fuente: Elaboración propia	16
Imagen 19 Información del botón Acerca De. Fuente: Elaboración propia.	16

RESUMEN

Este proyecto desarrolla una aplicación de gestión de inventarios dirigida a pequeñas y medianas empresas, integrando diversas tablas relacionadas entre sí, como proveedores, productos, usuarios, categorías y clientes. Se implementan operaciones CRUD (Crear, Leer, Actualizar y Eliminar), permitiendo tres niveles de usuarios: almaceneros, administradores y vendedores. Cada perfil cuenta con privilegios específicos para realizar operaciones, adaptando la gestión de inventarios según el rol asignado, esencial para el flujo de ventas.

La aplicación mejora las gestiones de inventario mediante el uso del lenguaje de programación Java, vinculado a bases de datos MySQL para consultas y operaciones. Además, emplea Swing para el diseño de interfaces gráficas, ofreciendo una experiencia de usuario funcional y amigable.

El alcance del proyecto incluye desde la conceptualización y diseño de la aplicación hasta la implementación completa de las funcionalidades CRUD y pruebas de usabilidad, garantizando la integridad y efectividad del sistema. Esta herramienta práctica y escalable optimiza la gestión de inventarios, ofreciendo una solución eficiente y accesible para pequeñas y medianas empresas.

El proyecto es posible descargarlo a través del siguiente enlace:

<https://github.com/Sergiocondeportero/TFGDAMSergioCondePortero>

DESCRIPCIÓN DEL PROYECTO Y SU BASE DE DATOS

En la actualidad, las pequeñas y medianas empresas suelen enfrentar dificultades para gestionar sus inventarios de manera eficiente. Esta aplicación de escritorio está diseñada para resolver esos problemas, optimizando el flujo de trabajo empresarial de forma rápida y efectiva. Utiliza una base de datos relacional y permite realizar diversas operaciones a través de un ejecutable desarrollado en Java, ofreciendo una solución práctica y accesible para mejorar la administración de inventarios.

Antes de analizar el funcionamiento de la aplicación, es fundamental comprender cómo está organizada su estructura interna en la base de datos. Basada en un modelo Entidad-Relación de la base de datos Inventario, la aplicación incluye las siguientes entidades:

Categoría: idCategoría (clave primaria), categoría.

Cliente: idCliente (clave primaria), nombre, apellido, documento, dirección, teléfono, correo.

Producto: idProducto (clave primaria) nombre, stock, idCategoría (clave foránea de la tabla Categoría), precio.

Proveedor: idProveedor (clave primaria), nombre, apellido, documento, razónSocial, dirección, teléfono, correo.

Usuario: idUsuario (clave primaria), nombre, apellido, documento, dirección, teléfono, correo, tipoUsuario, usuario, password.

En cuanto a las relaciones entre las entidades, se observa que un producto pertenece a una categoría, mientras que una categoría puede estar asociada a varios productos.

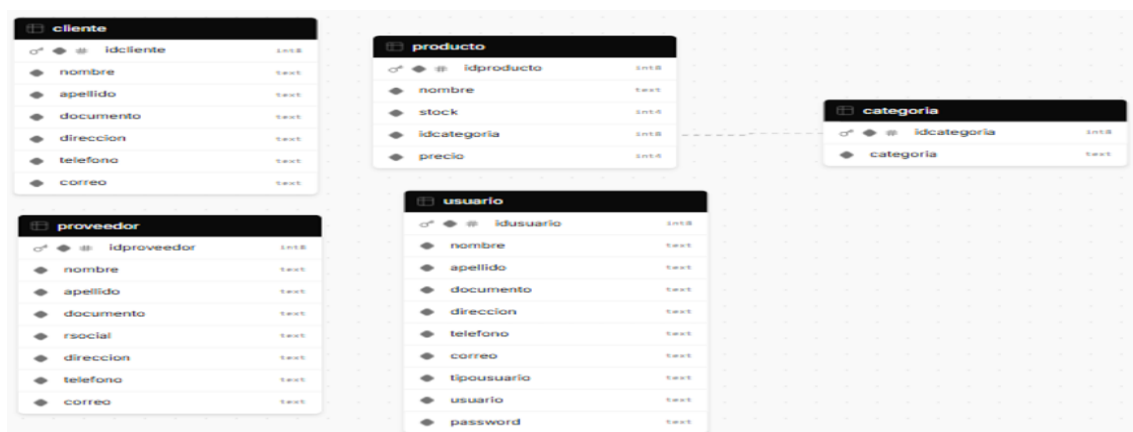


Imagen 1 Diagrama de clases usando la herramienta Postgres. Fuente elaboración propia.

Para realizar esta aplicación el nombre de la base de datos es Inventario y contiene la creación de las siguientes tablas en MySQL, siendo estas las siguientes:

```
CREATE TABLE `categorías` (  
  `idCategoria` int(11) NOT NULL AUTO_INCREMENT,  
  `categoria` varchar(30) NOT NULL,  
  PRIMARY KEY (`idCategoria`)  
)
```

```
clientesCREATE TABLE `clientes` (  
  `idCliente` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(30) NOT NULL,  
  `apellido` varchar(30) NOT NULL,  
  `documento` varchar(12) NOT NULL,  
  `direccion` varchar(100) NOT NULL,  
  `telefono` varchar(11) NOT NULL,  
  `correo` varchar(50) NOT NULL,  
  PRIMARY KEY (`idCliente`)  
)
```

```
productosCREATE TABLE `productos` (  
  `idProducto` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(100) NOT NULL,  
  `stock` int(30) NOT NULL,  
  `idCategoria` int(12) NOT NULL,  
  `precio` int(10) NOT NULL,
```

```

PRIMARY KEY (`idProducto`),

KEY `fk_categorias` (`idCategoria`),

CONSTRAINT `fk_categorias` FOREIGN KEY (`idCategoria`) REFERENCES

`categorias` (`idCategoria`)

)

```

```

proveedorCREATE TABLE `proveedor` (

`idproveedor` int(11) NOT NULL AUTO_INCREMENT,

`nombre` varchar(30) NOT NULL,

`apellido` varchar(30) NOT NULL,

`documento` varchar(12) NOT NULL,

`Rsocial` varchar(100) NOT NULL,

`direccion` varchar(100) NOT NULL,

`telefono` varchar(11) NOT NULL,

`correo` varchar(50) NOT NULL,

PRIMARY KEY (`idproveedor`)

)

```

```

usuariosCREATE TABLE `usuarios` (

`idUsuario` int(11) NOT NULL AUTO_INCREMENT,

`nombre` varchar(30) NOT NULL,

`apellido` varchar(30) NOT NULL,

`documento` varchar(12) NOT NULL,

`direccion` varchar(100) NOT NULL,

`telefono` varchar(11) NOT NULL,

```

```
`correo` varchar(50) NOT NULL,  
  
`tipoUsuario` varchar(20) NOT NULL,  
  
`usuario` varchar(15) NOT NULL,  
  
`password` varchar(255) NOT NULL,  
  
PRIMARY KEY (`idUsuario`)  
  
)
```

Después de analizar las tablas creadas en MySQL, se procederá a evaluar su funcionamiento en la aplicación mediante diversas operaciones CRUD (Crear, Leer, Actualizar y Eliminar). Estas operaciones demostrarán cómo las tablas interactúan con el sistema a través de sentencias SQL ejecutadas desde las clases de la aplicación.

Crear: Inserción de nuevos registros en tablas como productos, clientes o proveedores mediante sentencias INSERT INTO.

Leer: Consulta y visualización de datos almacenados utilizando SELECT, con capacidad de filtrar y buscar información específica.

Actualizar: Modificación de registros existentes mediante UPDATE, como cambios en stock o datos de contacto.

Eliminar: Borrado de registros innecesarios utilizando DELETE, con medidas de seguridad para evitar errores.

Estas operaciones se implementan mediante clases específicas, asegurando una integración eficiente entre la lógica de negocio y la base de datos. En conjunto, el proceso validará el correcto funcionamiento del sistema, garantizando una solución confiable y eficaz para la gestión de inventarios.

JUSTIFICACIÓN DEL PROYECTO Y ESTRUCTURA DEL PROYECTO

Este proyecto es fundamental para las pequeñas empresas, ya que aborda problemas comunes en la gestión de inventarios que impactan directamente en su operativa y rentabilidad. Muchas empresas enfrentan dificultades para mantener un control adecuado de sus inventarios, lo que puede generar desabastecimientos o exceso de productos. La aplicación proporciona un control preciso, eliminando estos problemas y garantizando una oferta oportuna de productos sin recurrir a excesos ni faltantes.

La automatización de los procesos de gestión de inventarios reduce costos operativos al eliminar tareas manuales y errores humanos. Además, la actualización en tiempo real de los datos permite a los gerentes tomar decisiones informadas sobre compras y reabastecimientos, adaptándose rápidamente a las fluctuaciones del mercado y mejorando la competitividad.

Este sistema también contribuye a la satisfacción del cliente al reducir retrasos en los pedidos y asegurar la disponibilidad de productos. Además, facilita la digitalización de procesos, modernizando las operaciones y optimizando recursos, lo que abre la puerta a futuras mejoras e innovación.

Antes de adentrarnos en el desarrollo de la aplicación, es importante entender cómo se ha diseñado y estructurado el proyecto. Para ello, se ha empleado el Modelo Vista-Controlador (MVC), una arquitectura que ayuda a separar las responsabilidades de la interfaz de usuario, la lógica de negocio y los datos. Esto mejora la mantenibilidad y escalabilidad del sistema.

El proyecto está organizado en varios paquetes que facilitan su gestión y funcionalidad:

Paquete InventarioTFG: Este paquete contiene el archivo principal (main) que se encarga de arrancar la aplicación, iniciando el flujo de trabajo en la interfaz.

Paquete Vista: Aquí se encuentran todas las clases encargadas de la visualización de los datos y la interfaz gráfica. Estas clases interactúan con las entidades de la base de datos previamente definidas, mostrando la información de manera estructurada y accesible para el usuario.

Paquete Modelo: Este paquete alberga las clases que definen las variables y atributos que se utilizarán en la aplicación, como los objetos correspondientes a productos, clientes, proveedores, etc. Incluye los métodos setters, getters y constructores, que permiten gestionar la información. Además, contiene la clase Conexión, responsable de establecer la comunicación con la base de datos MySQL.

Paquete Controlador: En este paquete se encuentran las clases encargadas de gestionar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar). Estas clases son el puente entre la interfaz de usuario y la base de datos, controlando el flujo de datos y ejecutando las acciones requeridas.

Paquete Notificación: Un paquete adicional que gestiona las notificaciones dentro de la aplicación. Este se utiliza para informar al usuario si una operación (como la creación, actualización o eliminación de un producto) se ha realizado con éxito o ha generado algún error, mostrando ventanas de notificación para mejorar la interacción con el usuario.

Paquete Librerías: Contiene los archivos JAR necesarios para que la aplicación funcione correctamente, como bibliotecas para la conexión a la base de datos o para la gestión de la interfaz gráfica.

Esta estructura modular permite que el proyecto sea fácilmente escalable y mantenible, asegurando que se pueda ampliar con nuevas funcionalidades o adaptarse a cambios futuros sin comprometer la integridad del sistema.

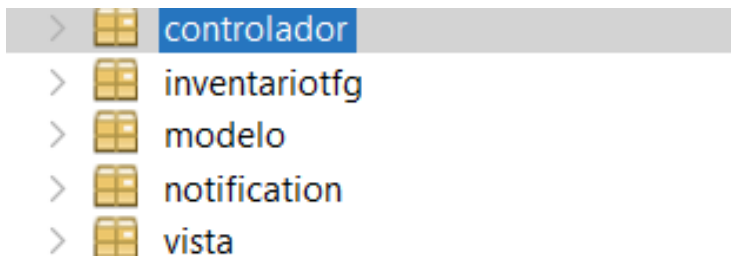


Imagen 2 Estructura de paquetes en Netbeans. Fuente: elaboración propia.



Imagen 3 Librerías utilizadas en el proyecto. Fuente: Elaboración propia.

INTRODUCCIÓN

La gestión de inventarios es un aspecto crítico en el funcionamiento de cualquier empresa, como puede ser el de las pequeñas y medianas empresas (PYMES), ya que impacta directamente en su rentabilidad y eficiencia operativa. En un entorno empresarial cada vez más competitivo y dinámico, es fundamental contar con sistemas que permitan un control adecuado y una toma de decisiones informadas respecto al stock de productos. Diversos estudios han demostrado que una gestión eficiente del inventario no solo mejora el flujo de caja de la empresa, sino que también aumenta la satisfacción del cliente, lo que se traduce en un incremento de las ventas y en una mayor lealtad hacia la marca.

Esta aplicación tiene como objetivo optimizar la gestión del inventario. Para lograrlo, se utiliza XAMP como herramienta para visualizar y gestionar de manera más eficiente las tablas y los datos previamente creados en MySQL. Además, se procede a realizar la inserción de datos, generándose el usuario admin con el rol de administrador.

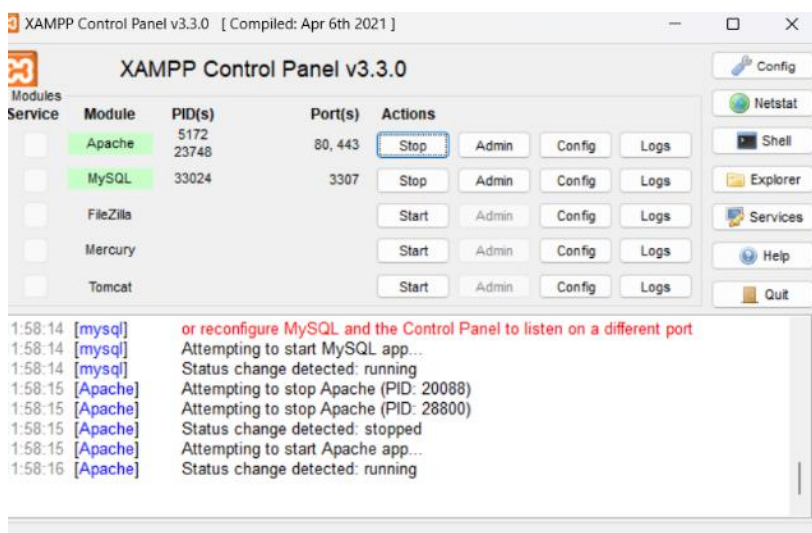


Imagen 4 Uso de XAMPP en un equipo Fuente: Elaboración propia.

```
1 INSERT INTO `usuarios` (`idUsuario`, `nombre`, `apellido`, `documento`, `direccion`, `telefono`, `correo`,  
  `tipoUsuario`, `usuario`, `password`) VALUES (NULL, 'admin', 'admin', 'admin', 'administrador', '123456',  
  'admin@admin.com', 'administrador', 'admin', 'admin');
```

Imagen 5 Inserción de datos en phpMyAdmin en la tabla "usuarios" Fuente: Elaboración propia.

OBJETIVOS Y MENÚ DE LA APLICACIÓN

El objetivo de este proyecto es implementar una gestión eficiente del inventario. Para ello, se han desarrollado múltiples operaciones CRUD (crear, leer, actualizar y eliminar) en las tablas creadas, otorgando permisos específicos a distintos tipos de usuarios: administrador, almacenero y vendedor. Esto permite que cada usuario gestione el inventario de manera adecuada según su rol. El acceso a la aplicación comienza con el inicio de sesión en la ventana de Login, donde el usuario ingresa sus credenciales, por ejemplo, el nombre de usuario y la contraseña del administrador (admin).



Imagen 6 Login con el usuario admin Fuente:Elaboración propia

Una vez dentro de la aplicación se puede observar que el diseño del usuario consta de un menú a la izquierda con varios botones, Home, Categorías, Clientes, Usuarios, Proveedores y Productos, por defecto la aplicación te llevará a Home donde te mostrará una breve descripción de las distintas funciones del Menú.

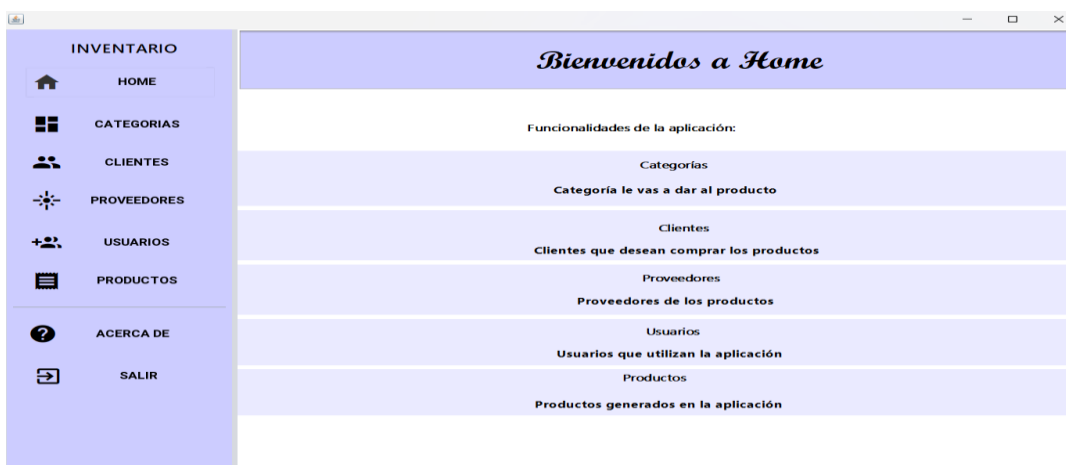


Imagen 7 HOME de la aplicación junto con su menú Fuente: Elaboración propia

DESARROLLO

La aplicación se desarrolló siguiendo el patrón de diseño Modelo-Vista-Controlador (MVC) y fue programada en Java utilizando la biblioteca Swing para implementar interfaces gráficas. Dentro de la aplicación, al seleccionar la opción Categorías, se pueden gestionar las categorías de productos mediante operaciones CRUD (crear, leer, actualizar y eliminar).

Para crear una categoría, basta con ingresar el nombre, mientras que el ID se genera automáticamente gracias a la configuración de autoincremento en la base de datos. Además, al seleccionar una categoría, es posible eliminarla utilizando la opción Eliminar. También se puede buscar una categoría ingresando su ID en el campo correspondiente. Sí se intenta eliminar una categoría, aparecerá una ventana de confirmación solicitando al usuario confirmar la acción antes de proceder con la eliminación. Esto asegura un manejo cuidadoso de los datos.



Imagen 8 Interfaz de CATEGORIAS. Fuente: Elaboración propia



Imagen 9 Funcionamiento del botón de guardar del registro categorías Fuente: Elaboración propia



Imagen 10 Funcionamiento del botón editar. Fuente: Elaboración propia



Imagen 11 Ventana emergente de eliminar. Fuente: Elaboración propia



Imagen 12 Funcionamiento del botón eliminar. Fuente: Elaboración propia

Existen otros apartados dentro del menú de gestión de inventario como son Clientes, Proveedores, Usuarios y Productos, en los diferentes apartados se permiten realizar las operaciones CRUD de manera similar al visto en Categorías, pero en sus respectivas tablas. Cabe destacar algunas particularidades: Para la tabla clientes posee los campos id, nombre, apellido, documento, dirección, teléfono y correo. En la tabla proveedores tiene el campo id, nombre, apellido, documento, R.social , Dirección, teléfono y correo. En el caso de Usuarios, existen tres tipos de roles disponibles: almacenero, vendedor y administrador, los cuales pueden seleccionarse desde un menú desplegable del tipo de usuario. Para Productos, al estar vinculada la tabla de productos con la tabla de categorías, se incluye un botón que facilita la búsqueda del ID de la categoría. Al pulsarlo, se abre una ventana en la que se puede seleccionar la categoría deseada.

INVENTARIO

Registro de clientes

Completa la información de los clientes

ID:

Nombre:

Apellido:

Documento:

Dirección:

Teléfono:

Correo:

ID	Nombre	Apellido	Documento	Dirección	Teléfono	Correo
1	julio	perez	123	123	134	1234
2	diego	conde	1234	perez	098712376	diegoconde
3	daniel	perez	123456	1234567	1234	
4	carlos	perez	124567	123456	1234567	1234

Imagen 13 Registro de clientes Fuente: Elaboración propia

INVENTARIO

Registro De Proveedores

Completa la Información de los Proveedores.

ID:

Nombre:

Apellido:

Documento:

R Social:

Dirección:

Teléfono:

Correo:

ID	Nombre	Apellido	Documento	R Social	Dirección	Teléfono	Correo
1	Juan	Garcia	12345	2235675	Rafael Y.	94899356	1234

Imagen 14 Registro de Proveedores. Fuente: Elaboración propia

Registro De Usuarios
Complete la Información de los Usuarios.

ID:

Nombre:

Apellido:

Documento:

Dirección:

Teléfono:

Correo:

Tipo Usuario:

Usuario:

Password:

ID	Nombre	Apellido	Doc.	Dire.	Teléfono	Correo	Tipo	Usuario	Pass.
1	helo	pizar	7480	ay la	9875	helo	admi	guihel	1234
3	Julio	Pérez	1	1	1	1	Vend	11	1
4	admin	admin	admin	admi	1234	admi	admi	admin	admin
5	Ana	Ruiz	2	2	2	2	Alma	Ana1	clave

Guardar **Editar** **Eliminar** **Buscar**

Imagen 15 Registro de usuarios. Fuente: Elaboración propia.

Tipo USuario

Almacenero

Administrador

Vendedor

Almacenero

Password

Imagen 16 Menú desplegable del tipo de usuario. Fuente: Elaboración propia.

Registro De Productos
Complete la Información de los Productos.

ID:

Nombre:

Stock:

Id categoría: **Q**

Categoría:

Precio de Venta:

ID	Nombre	Stock	IdCat	Precio V
1	ginebra	0	1	14.0
2	ron	0	1	12.0
3	Cordo	0	2	12.0

Guardar **Editar** **Eliminar** **Buscar**

Imagen 17 Registro de Productos. Fuente: Elaboración propia

LISTADO

ID	Nombre
1	bebidas
2	comida

Imagen 18 Pestaña del listado de categorías. Fuente: Elaboración propia

Al pulsar el botón Acerca de, se muestra información más detallada sobre la aplicación.

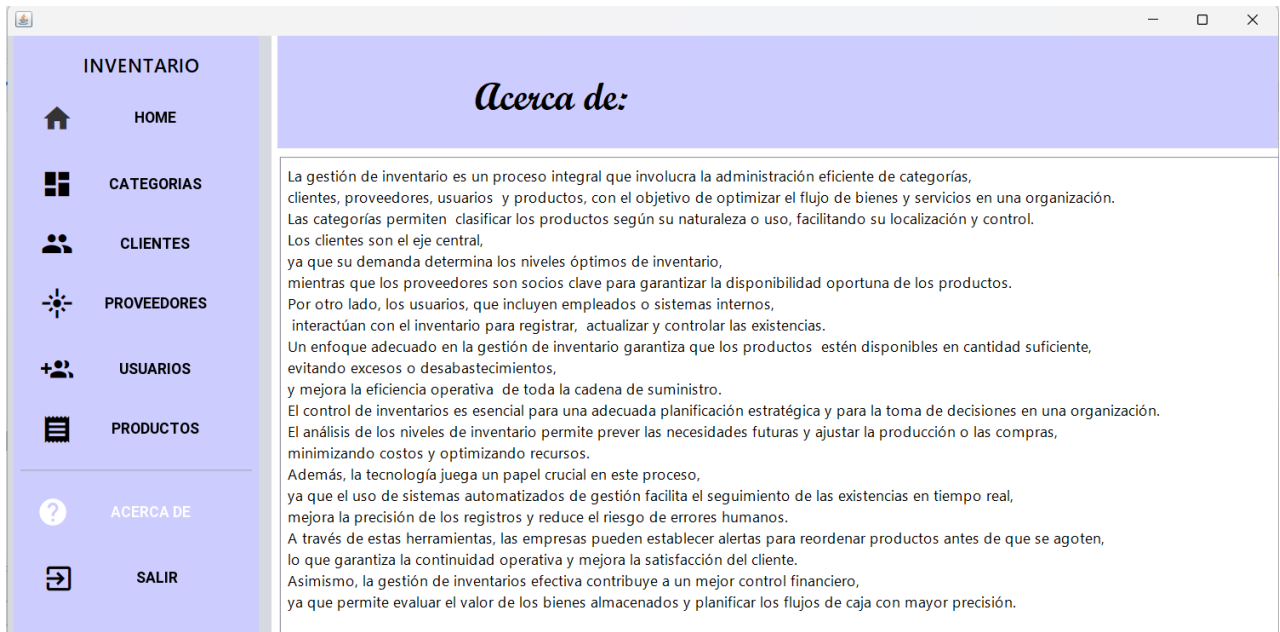


Imagen 19 Información del botón Acerca De. Fuente: Elaboración propia.

Por último, cabe destacar si se inicia sesión con un usuario Vendedor u Almacenero la aplicación tendrá unos permisos u otros de funciones a las que se puede acceder.

CONCLUSIONES

La implementación de esta aplicación de gestión de inventarios ha demostrado ser una solución eficiente y accesible para pequeñas y medianas empresas, optimizando sus procesos administrativos y operativos. A lo largo del desarrollo del proyecto, se ha logrado integrar diversas tablas interrelacionadas, permitiendo una gestión estructurada de proveedores, productos, usuarios, categorías y clientes.

Uno de los principales aportes de la aplicación es la implementación de operaciones CRUD, que permiten la creación, consulta, actualización y eliminación de registros, facilitando un control preciso del inventario. Además, la inclusión de tres niveles de usuarios (almaceneros, administradores y vendedores), cada uno con privilegios específicos, garantiza una gestión adaptada a las necesidades de cada rol dentro del flujo de ventas.

El uso del lenguaje de programación Java, en conjunto con bases de datos MySQL, ha permitido desarrollar una aplicación robusta y eficiente, asegurando integridad en la información y rapidez en las operaciones. Asimismo, la interfaz gráfica desarrollada con Swing ha proporcionado una experiencia de usuario intuitiva y funcional, favoreciendo la adopción y uso de la herramienta por parte de los distintos perfiles de usuarios.

Otro aspecto relevante es la escalabilidad del sistema, que permite futuras mejoras y adaptaciones según las necesidades cambiantes del mercado. Su diseño modular facilita la incorporación de nuevas funcionalidades y la optimización continua del rendimiento. Además, la aplicación fomenta la digitalización en empresas que aún utilizan procesos manuales, promoviendo una gestión más eficiente y acorde con las exigencias tecnológicas actuales.

En conclusión, este proyecto ha cumplido con los objetivos planteados, proporcionando una herramienta eficaz para la gestión de inventarios en pequeñas y medianas empresas. La aplicación no solo mejora la organización y control de los productos, sino que también contribuye al crecimiento y sostenibilidad de los negocios en un entorno competitivo.

Para futuras actualizaciones, se prevé el desarrollo de nuevas versiones de la aplicación que incorporen mejoras significativas. Entre las principales optimizaciones se incluye la implementación de gráficos interactivos para la visualización de datos de productos y otras tablas, lo que permitirá un análisis más intuitivo y dinámico.

Asimismo, se añadirá una funcionalidad para gestionar de manera eficiente las entradas y salidas de inventario, facilitando un control más preciso del stock. Además, se integrará la opción de generar reportes detallados para cada tabla, con la posibilidad de descargarlos en formato PDF, optimizando así la gestión y el análisis de la información.

BIBLIOGRAFÍA

- Beck, K. et al. (2001). *Manifesto for Agile Software Development*. Recuperado de agilemanifesto.org.
- Blaha, M., & Rumbaugh, J. (2005). *Modelado y Diseño Orientado a Objetos con UML*. Pearson Educación.
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
- Deitel, P., & Deitel, H. (2014). *Java: How to Program*. Prentice Hall.
- Firebase Documentation. (2021). *Firebase Cloud Messaging: Send Notifications*. Recuperado de firebase.google.com.
- García, R., & López, A. (2018). *Fundamentos de la Programación en Java*. Universidad Técnica de Madrid.
- Git Documentation. (2022). *Pro Git*. Recuperado de git-scm.com.
- Larman, C. (2004). *UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado*. Pearson Educación.
- López, T. (2017). *Introducción a la Programación Orientada a Objetos*. Ediciones del Software.
- Martínez, L. (2019). *Metodologías Ágiles para el Desarrollo de Software*. Tech Press.
- Pressman, R. (2014). *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill.
- Rodríguez, M. (2020). *Pruebas de Usabilidad: Métodos y Técnicas*. Ediciones Universitarias.
- Sánchez, J. (2020). *Gestión de Inventarios en Pequeñas Empresas: Estrategias y Herramientas*. Editorial Empresarial.
- Sommerville, I. (2011). *Software Engineering*. Addison-Wesley.

ANEXOS

Los anexos que se presentan a continuación ofrecen información adicional que complementa el contenido del proyecto de gestión de inventarios, enriqueciendo la comprensión del mismo y proporcionando un contexto más amplio. Estos anexos pueden incluir tablas, glosarios y ejemplos que aportan valor al trabajo realizado.

Anexo A: Glosario de Términos

Término	Definición
CRUD	Acrónimo que se refiere a las operaciones de Crear, Leer, Actualizar y Eliminar en la gestión de datos.
Base de Datos	Conjunto de datos estructurados que permiten el almacenamiento y recuperación eficiente de información.
Notificaciones	Mensajes enviados por la aplicación a los usuarios para informar sobre eventos importantes.
GitHub	Plataforma de desarrollo colaborativo basada en la nube que permite a los desarrolladores alojar, gestionar y controlar versiones de código fuente utilizando Git

Anexo B: Tecnologías Utilizadas

Tecnología/Herramienta	Descripción
Lenguaje de programación	Java
Base de datos	MySQL (implementada con XAMPP)
Framework de interfaz gráfica	Swing
Control de versiones	Git y GitHub
Metodologías de desarrollo	Metodologías Ágiles

Anexo C: Funcionalidades Clave de la Aplicación

Funcionalidad	Descripción
Gestión de usuarios con roles diferenciados	Administración de accesos para almaceneros, administradores y vendedores.
Operaciones CRUD en todas las tablas	Creación, lectura, actualización y eliminación de registros de productos, clientes, proveedores y categorías.
Gestión de inventario	Control preciso de las entradas y salidas de stock.
Generación de reportes	Creación de informes detallados en formato PDF sobre cada tabla del sistema.
Notificaciones y alertas	Envío de mensajes informativos sobre eventos relevantes en el inventario.

Anexo D: Mejoras Previstas para Futuras Versiones

Mejora	Descripción
Visualización de datos con gráficos interactivos	Implementación de gráficos para representar información del inventario de manera dinámica e intuitiva.
Optimización de la gestión de entradas y salidas	Mejora en el control del flujo de stock, permitiendo un registro más eficiente de movimientos de inventario.
Ampliación de la funcionalidad de reportes	Incorporación de opciones avanzadas de filtrado y exportación de informes en formato PDF.