

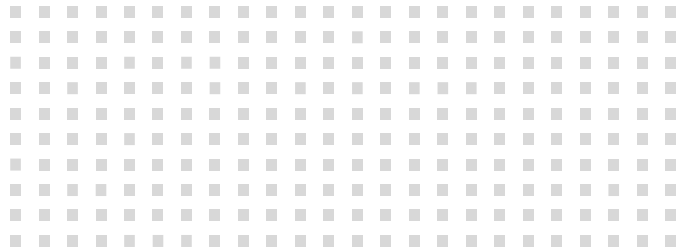


Gestor de inventarios

Ingeniería en Software

Desarrollado por:

Armando Ambrosio
Nocedal
Omar Arias Dominguez



Docente:

Maritza Flores
Dominguez



Contenido

| | |
|--|-----|
| Introducción..... | 3 |
| Planteamiento del problema..... | 4 |
| Objetivo general del proyecto..... | 5 |
| Objetivos específicos del proyecto | 6 |
| Hipótesis | 7 |
| Alcances..... | 8 |
| Justificación..... | 10 |
| Limitaciones | 12 |
| Lista de requerimientos | 13 |
| Tabla de requerimientos..... | 17 |
| Marco Teórico | 27 |
| Modelo en Cascada de desarrollo del Software | 32 |
| Etapas del modelo en Cascada..... | 33 |
| Implementación del Modelo en Cascada | 34 |
| Diagrama de clases UML | 44 |
| Prototipo de la interfaz | 61 |
| Desarrollo | 65 |
| Manual de usuario..... | 78 |
| Conclusiones..... | 111 |
| Mejoras futuras..... | 112 |
| Índice de Imágenes | 113 |
| Referencias | 115 |

Introducción

En un entorno empresarial, la gestión eficiente del inventario es esencial para garantizar operaciones rápidas y satisfacer las demandas de los clientes. Para brindar una solución efectiva, este proyecto se enfoca en desarrollar un administrador de inventario en Java. Diseñado para ayudar a las pequeñas empresas a administrar sus productos de manera eficiente, el sistema permite a los usuarios agregar nuevos productos, verificar los niveles de inventario y eliminar productos que ya no necesitan.

Los sistemas de gestión de inventario son muy importantes en los entornos empresariales por diversas razones. En primer lugar, las empresas pueden gestionar efectivamente el inventario y evitar pérdidas financieras causadas por la obsolescencia de los productos, el desabastecimiento o el exceso de existencias. Además, la visibilidad en tiempo real de la disponibilidad del producto facilita la planificación de adquisiciones y la gestión de la cadena de suministro.

Además, el registro de productos no solo ayuda a identificar y buscar productos específicos, sino que también es fundamental para realizar acciones como actualizar stock, eliminar productos obsoletos y tomar decisiones estratégicas relacionadas con la adquisición de nuevos productos.

Los sistemas de gestión de inventario también son cruciales para mejorar la satisfacción del cliente. Obteniendo información precisa sobre la disponibilidad del producto.

Planteamiento del problema

En el entorno empresarial actual, la gestión eficiente del inventario es un aspecto fundamental del éxito empresarial. Sin embargo, muchas organizaciones enfrentan importantes desafíos en este sentido debido a la falta de herramientas modernas y efectivas para gestionar adecuadamente el inventario. El proyecto de desarrollo "Gestor de Inventarios" surge en respuesta a los siguientes problemas identificados:

- Falta de automatización en la gestión de inventarios:

Muchas empresas todavía dependen de procesos manuales y hojas de cálculo para realizar un seguimiento de sus productos e inventario. Esto da como resultado una mayor probabilidad de error humano, retrasos en las actualizaciones del inventario y una falta de visibilidad en tiempo real del inventario disponible.

- Dificultades para identificar productos obsoletos:

La falta de herramientas automatizadas para identificar y gestionar productos obsoletos puede provocar la acumulación de inventario no vendible. Esto requiere una valiosa inversión de capital y espacio de almacenamiento.

- La eficiencia de actualización de la información del producto es baja:

Sin una solución implementada la actualización de los detalles del producto (como cambios en el precio, la descripción o los atributos) suele ser un proceso lento y propenso a errores.

- Riesgo de Pérdida de Clientes:

La gestión ineficiente de inventarios puede resultar en situaciones de escasez de productos, lo que a su vez puede llevar a la pérdida de clientes insatisfechos debido a la incapacidad de satisfacer sus demandas de manera oportuna.

Objetivo general del proyecto

El objetivo general de este proyecto es desarrollar un sistema de gestión de inventarios en Java que permita a los usuarios administrar eficientemente los productos, controlar los niveles de existencias, actualizar la información y gestionar de manera efectiva la eliminación de productos obsoletos. El sistema proporcionará una solución automatizada y fácil de usar para mejorar la precisión, la agilidad y la eficiencia en la administración de inventarios.

Objetivos específicos del proyecto

1. **Diseñar la Estructura del Sistema:** Definir la arquitectura, las clases y las relaciones necesarias para la representación de productos, inventarios y funcionalidades asociadas.
2. **Implementar una Interfaz de Usuario:** Desarrollar una interfaz de usuario intuitiva y amigable que permita a los usuarios acceder a las funciones del sistema de manera efectiva y que no tengan problemas con su uso.
3. **Implementar Funcionalidad de Agregar Productos:** Crear la función de agregar nuevos productos al inventario, permitiendo a los usuarios ingresar la información relevante del producto.
4. **Realizar Actualizaciones de Información:** Diseñar y programar métodos que permitan a los usuarios actualizar la cantidad, así como el retiro de productos en stock.
5. **Crear Función de Eliminación de Productos:** Implementar la capacidad de eliminar productos que ya no estén disponibles o que vayan a ser retirados.
6. **Implementar Búsqueda:** Desarrollar funciones de búsqueda que permitan a los usuarios encontrar productos por su ID.

Hipótesis

Se postula que, mediante la implementación de un Gestor de Inventarios en Java, será posible mejorar significativamente la eficiencia en la administración de inventarios. Al automatizar procesos que antes eran manuales como el seguimiento de existencias y la actualización de información de manera escrita y en continuas hojas de cálculo.

Alcances

1. Registro de Productos:

- Los usuarios podrán registrar nuevos productos en el sistema, ingresando información como nombre, descripción, precio y cantidad inicial en stock.

2. Visualización de Inventarios:

- Los usuarios podrán acceder y visualizar el inventario existente en el sistema.

3. Actualización de Información de Productos:

- Los usuarios tendrán la capacidad de actualizar la información de productos existentes, lo que incluye modificar su precio, descripción y cantidad en stock.

4. Eliminación de Productos:

- Se proporcionará a los usuarios la opción de eliminar productos que ya no estén disponibles o que se hayan vuelto obsoletos.

5. Búsqueda de Productos:

- Los usuarios podrán buscar productos por su ID y obtener detalles sobre su ID, nombre, precio, descripción y existencias.

6. Interfaz de Usuario Intuitiva:

- El sistema contará con una interfaz de usuario intuitiva y fácil de usar, diseñada para usuarios de diversos niveles de experiencia técnica.

7. Seguridad de Acceso:

- Se implementará un sistema de autenticación para garantizar que solo usuarios autorizados puedan acceder al sistema.

8. Rendimiento Eficiente:

- El sistema se diseñará y optimizará para manejar eficientemente un gran número de productos, evitando ralentizaciones significativas en las operaciones clave.

9. Escalabilidad:

- La arquitectura del sistema se diseñará teniendo en cuenta la posibilidad de escalabilidad, de modo que pueda adaptarse a un aumento en la cantidad de productos o usuarios sin requerir cambios drásticos en la infraestructura.

10. Mantenibilidad:

- El código fuente se documentará adecuadamente y se seguirán buenas prácticas de programación para facilitar futuras actualizaciones y mejoras en el sistema.

11. Integración Futura:

- Se dejará espacio para la posible integración con otros sistemas o aplicaciones empresariales existentes en el futuro.

12. Soporte Técnico:

- Se proporcionará documentación de usuario y manuales para ayudar a los usuarios a comprender y utilizar el sistema de manera efectiva.
- Se establecerá un mecanismo de soporte técnico para atender consultas y problemas relacionados con el sistema.

13. Pruebas y Validación:

- Se llevarán a cabo pruebas exhaustivas, incluyendo pruebas de unidad y pruebas de integración para garantizar la calidad y el funcionamiento adecuado del sistema.

Justificación

En el contexto de pequeñas empresas con recursos limitados la gestión eficiente de inventarios se convierte en un desafío clave que afecta directamente su viabilidad y competitividad en el mercado. Un gestor de inventario en Java diseñado específicamente para estas empresas tiene sentido por las siguientes razones:

- **Reducción de costos**

La mayoría de las pequeñas empresas no tienen los recursos financieros para comprar costosas licencias de software especializado. Desarrollar un gestor de inventario utilizando Java proporciona una solución económica ya que se basa en tecnología de código abierto y no requiere inversión en software especializado esto permite a las pequeñas empresas aprovechar al máximo los presupuestos limitados.

- **Facilidad de uso**

El software especializado suele requerir una amplia formación y una larga curva de aprendizaje, lo que puede suponer una barrera para las pequeñas empresas. Un gestor de inventario de Java se puede diseñar con una interfaz de usuario intuitiva y fácil de usar, lo que minimiza la necesidad de una capacitación exhaustiva y promueve la adopción por parte de los empleados.

- **Escalabilidad**

A medida que las pequeñas empresas crecen, sus necesidades de gestión de inventario pueden cambiar. A medida que su negocio crece, el gestor de inventario en Java se puede ampliar y ajustar para adaptarse a mayores cantidades de productos u otros requisitos.

- **Personalización**

El software especializado a menudo incluye características innecesarias para las pequeñas empresas, lo que puede dificultar su adopción y uso. Un gestor de inventarios en Java se puede adaptar específicamente a las necesidades de una pequeña empresa, lo que significa que solo se incluirán las funcionalidades necesarias, evitando la complejidad innecesaria.

- **Mejora de la eficiencia**

Un gestor de inventarios en Java automatiza tareas manuales como el seguimiento de productos, la actualización del inventario y la información asociada a los productos esto mejora significativamente la eficiencia operativa y reduce la probabilidad de errores, aumentando así la productividad y reduciendo los costos operativos.

Limitaciones

- **Escalabilidad:** Dependiendo de la arquitectura y el diseño del sistema, podría haber limitaciones en cuanto a la capacidad de manejar grandes volúmenes de productos y usuarios simultáneos de manera eficiente.
- **Rendimiento:** A medida que la cantidad de productos y usuarios vaya creciendo, el rendimiento del sistema puede verse afectado. Las funciones de búsqueda, actualización y eliminación podrían volverse más lentas si no se optimizan adecuadamente.
- **Seguridad:** La seguridad de los datos es una preocupación importante dentro del sistema. Si no se implementan adecuadamente medidas de autenticación y autorización, podría haber riesgo de acceso no autorizado a la información del inventario y perder información del inventario.
- **Capacidades de Personalización:** Dependiendo de la complejidad del sistema y del tiempo disponible, podría haber limitaciones en la capacidad de personalizar la interfaz de usuario y las funciones según las necesidades de cada negocio.
- **Usabilidad:** Si la interfaz de usuario no se diseña de manera intuitiva, los usuarios podrían tener dificultades para aprender y utilizar el sistema de manera eficiente por lo que el sistema no podría ser aprovechado.
- **Actualización y Mantenimiento:** Si no se planifica un proceso de evolución y mantenimiento adecuado el sistema puede volverse obsoleto con el tiempo y enfrentar problemas de compatibilidad con nuevas tecnologías.

Lista de requerimientos

1. **Elección de tecnología:** Para el desarrollo del sistema se va a emplear el lenguaje de programación Java y NetBeans como entorno de desarrollo integrado. Además, se va a utilizar GitHub para llevar un control de versiones y volver a una versión anterior del software en caso de que una implementación del código produzca fallos en los demás componentes del sistema.
2. **Registro de nuevos productos:** Los usuarios podrán ingresar los detalles de un nuevo producto, incluyendo nombre, id, descripción, precio y cantidad inicial a través de una interfaz con 5 campos para cada atributo del producto y un botón de guardar para almacenar la información introducida.

La forma en que se va a almacenar la información es a través de una estructura de datos en Java la cual es un array.

3. **Menú de opciones:** El usuario puede seleccionar una de las siguientes opciones por medio de una interfaz en donde cada opción va a tener un botón:

1. Agregar producto al inventario:
2. Eliminar producto del inventario:
3. Buscar producto en el inventario.
4. Listar productos en el inventario.
5. Editar producto en el inventario.
6. Crear nuevo usuario
7. Cerrar sesión.

Una vez que se haya seleccionado la opción se va a generar un panel con los campos necesarios para cada uno a excepción del botón de cerrar sesión el cual vuelve al inicio de sesión del programa para restringir el acceso a cualquiera.

4. **Visualización de Inventario:** Los usuarios podrán acceder y visualizar el inventario existente en el sistema. Se devolverá una lista de productos en forma de tabla en donde se enlistan por columnas sus datos como ID, nombre, descripción, existencias y precio.
5. **Búsqueda de Productos:** Los usuarios podrán buscar productos por su ID. Una vez que el producto se haya encontrado se va a mostrar en pantalla toda la información asociada a ese producto su nombre, descripción, existencias y precio.
6. **Creación de nuevos usuarios:** El usuario principal del sistema va a tener la opción de agregar nuevos usuarios al sistema y definir los roles que tendrá cada uno, el sistema manejará tres roles:
 - Administrador: Este rol va a tener acceso a todas las operaciones del sistema y en particular a la operación de crear nuevos usuarios.
 - Gestor de inventario: Este rol va a tener acceso a las operaciones de agregar producto, buscar producto, listar productos la operación de eliminar productos y editar producto la opción de crear nuevos usuarios están restringidas a este rol.
 - Usuario regular: Este rol va a tener acceso a las operaciones de buscar producto y listar productos el resto de las operaciones van a estar restringidas.
7. **Actualización de información de productos:** Se permitirá a los usuarios modificar la información de un producto, como cambiar su precio, nivel de existencias, descripción y nombre.
8. **Eliminación de productos:** Los usuarios podrán eliminar productos que ya no están disponibles o que se vayan a retirar del inventario ingresando el ID del producto esta operación va a remover toda la información asociada al producto como el ID, precio, nombre, descripción y existencias una vez que se haya hecho la operación no hay manera de recuperar la información en caso de un error.

9. **Rendimiento:** El sistema debe ser capaz de manejar un gran número de productos de manera eficiente. Además, es necesario establecer tiempos máximos de respuesta para consultas y actualizaciones de inventario. Es decir, la capacidad del sistema para responder rápidamente a las solicitudes de los usuarios. Para lograr esto es importante llevar a cabo un buen diseño y la optimización del software, así como la elección de hardware adecuado.
10. **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de entender para el usuario final. La interfaz debe estar diseñada de tal manera que los usuarios puedan navegar, buscar, actualizar y gestionar productos e inventario de manera sencilla y sin requerir una formación técnica extensa. Por lo que se deben tomar en cuenta ciertos aspectos antes de realizar el prototipo de la interfaz como formularios intuitivos cuando los usuarios necesiten ingresar información, una vista de lista y detalles de tal manera que la información de los productos es clara y fácil de identificar, acciones claras utilizando botones y enlaces con etiquetas claras que expresen las acciones que se pueden realizar.
11. **Escalabilidad:** El sistema debe ser diseñado de manera que pueda escalarse fácilmente en caso de un aumento en la cantidad de productos o usuarios. Esto implica que puedan incorporarse nuevos componentes y la implementación de bases de datos, así como mecanismos de recuperación de datos.
12. **Mantenibilidad:** El código debe estar bien estructurado y documentado para facilitar futuras actualizaciones y mejoras.
13. **Seguridad de Acceso:** Se implementará un sistema de autenticación para garantizar que solo usuarios autorizados puedan acceder al sistema, se van a conceder los nombres de usuario y contraseña para ingresar al sistema, para autenticar a los usuarios se va a mostrar una ventana solicitando el nombre del usuario y la contraseña una vez ingresados los datos se va a realizar una verificación para revisar que coincidan los datos y posteriormente dar acceso a las operaciones del inventario.

14. **Pruebas y Validación:** Se llevarán a cabo pruebas exhaustivas, incluyendo pruebas de unidad y pruebas de integración para garantizar la calidad y el funcionamiento adecuado del sistema.
15. **Integración con sistemas externos:** Posibilidad de integrar el sistema con otros sistemas. El sistema debe estar diseñado adecuadamente para integrarlo a otro sistema como lo puede ser un sistema de ventas con la integración a este sistema, es posible mantener actualizados los niveles de inventario en tiempo real, lo que ayuda a evitar la sobreventa o la falta de productos. También puede permitir la generación automática de facturas o la actualización de precios.
16. **Disponibilidad:** Garantizar que el sistema esté disponible para los usuarios la mayor parte del tiempo. Los clientes deben ser capaces de manejar errores locales de manera adecuada es por ello que a través del manual de usuario el cliente conocerá las acciones a realizar en caso de un error del sistema y evitar perder información valiosa.

Tabla de requerimientos

| Funcionales | No Funcionales |
|---|---|
| <p>Elección de tecnología y paradigma de programación: Para el desarrollo del sistema se va a emplear el lenguaje de programación Java para desarrollar toda la aplicación y NetBeans como entorno de desarrollo integrado. Además, se va a utilizar GitHub para llevar un control de versiones y volver a una versión anterior del software en caso de que una implementación del código produzca fallos en los demás componentes del sistema.</p> <p>En cuanto al paradigma de programación se va a emplear el orientado a objetos por la capacidad de usar objetos para modelar entidades del mundo real y las clases representan plantillas para definir las características (atributos) y comportamientos que realizan los objetos.</p> | <p>Rendimiento: El sistema debe ser capaz de manejar un gran número de productos de manera eficiente. Además, es necesario establecer tiempos máximos de respuesta para consultas y actualizaciones de inventario. Es decir, la capacidad del sistema para responder rápidamente a las solicitudes de los usuarios. Para lograr esto es importante llevar a cabo un buen diseño y la optimización del software, así como la elección de hardware adecuado.</p> |
| <p>Registro de nuevos productos:</p> <p>Se emplea como estructura de datos la clase ArrayList la cual es una lista dinámica que permite almacenar y gestionar un conjunto de datos de manera flexible el espacio de la lista se gestiona de manera dinámica esto es útil ya que no se está implementando una base de datos, el arrayList puede manejar una cantidad considerable de datos y realizar operaciones como añadir y eliminar.</p> | |

Se crea una clase Productos que representa las características de un producto esta tiene los siguientes atributos

1. **idProducto** (int): Identificador único del producto.
2. **nombre** (String): Nombre del producto.
3. **descripcion** (String): Descripción del producto.
4. **existencia** (int): Cantidad en existencia del producto.
5. **precio** (double): Precio del producto.

Cada uno de estos atributos es declarado como privado siguiendo el principio de encapsulación para proteger los datos y acceder a métodos no autorizados desde otras clases.

La clase emplea un constructor para inicializar los valores de un producto. Además, se construyen métodos getters y setters para obtener y modificar los datos de cada uno de los atributos de un producto estos métodos son declarados como public para poder acceder a ellos fuera de la clase.

Se crea una clase Inventario, esta clase maneja las principales operaciones del inventario, en esta se declara un ArrayList que va a almacenar objetos de la clase Productos, en este clase hay un método agregarProducto que devuelve un valor boolean además el método toma como parámetro un objeto **Producto** llamado **producto** que es el producto que se desea agregar al inventario primero se obtiene el ID

| | |
|--|--|
| <p>del producto que se va a agregar mediante la llamada al método getIdProducto del objeto producto. Se utiliza un bucle for-each para recorrer la lista de productos existentes en el inventario se compara la ID del producto que se va a agregar con el ID del producto existente en el inventario se retorna un valor false en caso de que el producto ya se encuentre en el inventario en caso contrario el producto se agrega al inventario y se retorna un valor verdadero.</p> | |
| <p>Menú de opciones:</p> <p>Como parte de la interfaz de la aplicación se declara una clase MainMenuWindow que va a representar la ventana principal de la aplicación con un menú que va a contener las siguientes opciones:</p> <ul style="list-style-type: none"> • Agregar producto al inventario. • Eliminar producto del inventario. • Buscar producto en el inventario. • Listar productos en el inventario. • Editar producto en el inventario. • Crear nuevo usuario • Cerrar sesión. <p>La clase contiene:</p> <ul style="list-style-type: none"> • Una ventana JFrame como ventana principal. • Una barra de herramientas JToolBar que contiene botones para realizar diversas acciones. | <p>Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de entender para el usuario final. La interfaz debe estar diseñada de tal manera que los usuarios puedan navegar, buscar, actualizar y gestionar productos e inventario de manera sencilla y sin requerir una formación técnica extensa. Por lo que se deben tomar en cuenta ciertos aspectos antes de realizar el prototipo de la interfaz como formularios intuitivos cuando los usuarios necesiten ingresar información, una vista de lista y detalles de tal manera que la información de los productos es clara y fácil</p> |

| | |
|--|--|
| <ul style="list-style-type: none"> • Contiene dos paneles mainPanel y usuarioPanel • mainPanel: Utiliza un CardLayout para cambiar dinámicamente entre diferentes ventanas de contenido. • usuarioPanel: Es un panel ubicado en el lado izquierdo que contiene información del usuario como una imagen y botones para crear un usuario y cerrar sesión. | <p>de identificar, acciones claras utilizando botones y enlaces con etiquetas claras que expresen las acciones que se pueden realizar.</p> |
| <p>Visualización de Inventario: En la clase SistemaGestionInventarios a traves del método iniciarSesion que retorna un valor boolean se recorre un arrayList de objetos de la clase Usuarios en esta clase se declaran atributos como:</p> <ul style="list-style-type: none"> • private int id: identificador único del usuario • private String nombreUsuario: Nombre del usuario • private String contraseña: Contraseña del usuario • private String rol: Rol del usuario <p>Estos atributos sirven para manejar niveles de acceso en la aplicación y garantizar un nivel de seguridad en el acceso de la aplicación.</p> <p>Cuando se llama al método iniciarSesion se recorre el arrayList de objetos de tipo Usuario y se verifica que las credenciales registradas como nombre y contraseña sean iguales a las que el usuario ingreso sí coinciden las credenciales se retorna un valor verdadero y se</p> | |

| | |
|--|--|
| <p>establece un usuarioLogeado para mantener una sesión temporal en caso contrario se retorna un valor de falso.</p> <p>En la clase SistemaGestionInventarios. Si la sesión es válida, se va a invocar a un método listarProductosEnInventario, que devolverá una lista de productos en forma de tabla en donde se enlistan por columnas sus datos como ID, nombre, descripción, existencias y precio.</p> | |
| <p>Editar Productos:</p> <p>En la clase SistemaGestionInventarios se llama al método iniciarSesion en el cual se recorre un arrayList de objetos de tipo Usuario y se verifica que las credenciales registradas como nombre y contraseña sean iguales a las que el usuario ingreso sí coinciden las credenciales se retorna un valor verdadero y se establece un usuarioLogeado para mantener una sesión temporal en caso contrario se retorna un valor de falso.</p> <p>Se invoca un método editarProductoEnInventario que toma como como parámetros un objeto de tipo usuario, un objeto de tipo producto, dos datos de tipo String que indican el campo a editar y el nuevo valor en este método se realiza una validación invocando un método getter para obtener el valore asociado al rol de usuario y verificar el nivel de acceso que tiene el usuario para realizar la operación si no tiene acceso se</p> | <p>Mantenibilidad: El código debe estar bien estructurado y documentado para facilitar futuras actualizaciones y mejoras.</p> |

| | |
|---|--|
| <p>devuelve una ventana de dialogo JOptionPane para mostrar un mensaje indicando que no se tiene permiso.</p> <p>A traves del objeto inventario de la clase Inventario que permite llevar a cabo operaciones del inventario se invoca al método editarProducto con argumentos un objeto de tipo producto, un String de campo a editar y un String de nuevo valor. En el método de la clase inventario se valida el campo a editar y se realiza la conversión al tipo de dato que lo requiere y se invoca un método setter de la clase Producto para actualizar el nuevo valor.</p> | |
| <p>Búsqueda de Productos: En la clase SistemaGestionInventarios se llama al método iniciarSesion en el cual se recorre un arrayList de objetos de tipo Usuario y se verifica que las credenciales registradas como nombre y contraseña sean iguales a las que el usuario ingreso sí coinciden las credenciales se retorna un valor verdadero y se establece un usuarioLogeado para mantener una sesión temporal en caso contrario se retorna un valor de falso.</p> <p>Se invoca un método buscarProductoEnInventario que retorna un objeto de tipo producto de la clase Producto y toma como parámetro un dato de tipo int que identifica el id del producto a buscar verifica si el usuario esta logeado y si se cumple la</p> | <p>Pruebas y validación: Se llevarán a cabo pruebas exhaustivas, incluyendo pruebas de unidad y pruebas de integración para garantizar la calidad y el funcionamiento adecuado del sistema.</p> |

| | |
|---|---|
| <p>condición se devuelve una llamada al metodo buscarProducto con argumento de tipo entero el id del producto ingresado a traves del objeto inventario.</p> <p>En el metodo buscarProducto de la clase Inventario que devuelve un objeto de tipo producto y tiene como parámetro un dato de tipo int del id del producto, en el método se recorre el array de objetos de tipo producto y se utiliza un método getter para obtener el valor del id de cada uno de los productos y compararlo con el id que recibe el método de esta forma se realiza la verificación si el producto se encuentra en el inventario.</p> | |
| <p>Eliminación de productos:</p> <p>Búsqueda de Productos: En la clase SistemaGestionInventarios se llama al método iniciarSesion en el cual se recorre un arrayList de objetos de tipo Usuario y se verifica que las credenciales registradas como nombre y contraseña sean iguales a las que el usuario ingreso si coinciden las credenciales se retorna un valor verdadero y se establece un usuarioLogeado para mantener una sesión temporal en caso contrario se retorna un valor de falso.</p> <p>Se invoca un método eliminarProductoEnInventario que es un metodo vacio y toma como parámetro un objeto de tipo usuario y un objeto de tipo producto</p> | <p>Integración con sistemas externos: Posibilidad de integrar el sistema con otros sistemas.</p> |

| | |
|---|---|
| <p>verifica si el usuario esta logueado y se invoca un método getter para obtener el rol de usuario para validar el nivel de acceso si la condición se cumple la condición se invoca una llamada al metodo eliminarProducto que toma como argumento un objeto de tipo producto.</p> <p>En el metodo eliminarProducto de la clase Inventario recibe el objeto de tipo producto y se utiliza el metodo remove para eliminarlo del arrayList de objetos.</p> | |
| <p>Creación de usuarios:</p> <p>En la clase SistemaGestionInventarios se llama al método iniciarSesion en el cual se recorre eun arrayList de objetos de tipo Usuario y se verifica que las credenciales registradas como nombre y contraseña sean iguales a las que el usuario ingreso sí coinciden las credenciales se retorna un valor verdadero y se establece un usuarioLogeado para mantener una sesión temporal en caso contrario se retorna un valor de falso.</p> <p>Se capturan los datos ingresados por el usuario a traves de una interfaz y se verifica que el usuario este logeado y se invoca un metodo getter para obtener el rol del usuario y compararlo con el valor de administrado que es el rol que tiene acceso para crear nuevos usuarios. Si se cumple la condición se crea un nuevo objeto de tipo usuario a traves de la clase Usuarios mandando los datos correspondientes</p> | <p>Disponibilidad: Garantizar que el sistema esté disponible para los usuarios la mayor parte del tiempo. Los clientes deben ser capaces de manejar errores locales de manera adecuada es por ello que a través del manual de usuario el cliente conocerá las acciones a realizar en caso de un error del sistema y evitar perder información valiosa.</p> |

| | |
|--|--|
| <p>al constructor de la clase. Y a través del objeto sistema que controla las operaciones se invoca el método agregarUsuario mandando como argumento el objeto recién creado.</p> <p>El método agregarUsuario de la clase sistemaGestionInventarios toma como parámetro un objeto de la clase usuario y antes de agregar el usuario al arrayList de usuarios se realiza una validación en la que se invoca un método getter para obtener el id del usuario y después de obtener el valor se hace una llamada al método existeUsuarioConId que retorna un valor boolean este toma como parámetro un id y se recorre el arrayList por medio de un bucle for-each, a medida que se recorre cada uno de los objetos se obtiene los valores de sus id y se compara con la actual para comprobar que no sean idénticas si no existen id idénticas se devuelve true y si el valor que se devolvió en el método es verdadero el nuevo usuario se agrega al arrayList</p> | |
| | <p>Escalabilidad: El sistema debe ser diseñado de manera que pueda escalarse fácilmente en caso de un aumento en la cantidad de productos o usuarios.</p> |
| | <p>Seguridad de Acceso: Se implementará un sistema de</p> |

| | |
|--|--|
| | autenticación para garantizar que solo usuarios autorizados puedan acceder al sistema. El programa va a solicitar al usuario que inicie sesión proporcionando su nombre de usuario y contraseña. Antes de iniciar sesión en el programa se cargan dos usuarios (usuario1 y usuario2) para poder iniciar sesión en el programa. |
|--|--|

Marco Teórico

La gestión de inventario es una función importante en cualquier entorno empresarial porque afecta directamente la eficiencia operativa, la satisfacción del cliente y la rentabilidad de una empresa. En este contexto, se plantea el proyecto de desarrollo de “Gestor de Inventarios en Java” con el objetivo de resolver desafíos comunes relacionados con la gestión de productos e inventarios en entornos empresariales.

La gestión de inventario implica monitorear, controlar y optimizar los productos y recursos disponibles en una organización. En otras palabras, se trata de garantizar que la empresa tenga siempre el producto adecuado en el lugar adecuado y en el momento adecuado. Esto es fundamental para evitar la escasez de productos que afecte las ventas y la satisfacción del cliente y para minimizar la inversión de capital en el exceso de inventario.

El proyecto Gestor de Inventarios en Java tiene como objetivo resolver desafíos comunes en la gestión de inventario, como la falta de automatización y las ineficiencias en el seguimiento de las fluctuaciones de productos e inventario. La mayoría de los sistemas tradicionales carecen de la funcionalidad necesaria para manejar eficazmente la creciente diversidad de productos y la complejidad de la gestión de inventario.

El sistema se centrará en proporcionar soluciones modernas y eficaces que permitan a las empresas gestionar sus productos de forma eficaz. Los usuarios podrán agregar nuevos productos, consultar niveles de stock, actualizar información relevante y eliminar productos obsoletos, todo a través de una interfaz intuitiva y fácil de usar.

Otros sistemas existentes

- **SAP Inventory Management:** SAP ofrece una suite de soluciones empresariales que incluyen gestión de inventarios. Su módulo de gestión de inventarios proporciona herramientas para el seguimiento de existencias, planificación de inventario y optimización de la cadena de suministro.

- **Oracle Inventory Management:** Oracle proporciona un sistema integral de gestión de inventarios que se integra con su suite de aplicaciones empresariales. Ofrece capacidades de seguimiento de existencias en tiempo real, control de costos y gestión de pedidos de reposición.
- **WMS (Warehouse Management System):** Los sistemas de gestión de almacenes son aplicaciones especializadas que se centran en la gestión de inventarios en almacenes y centros de distribución. Ejemplos populares incluyen Manhattan Associates, HighJump, y Blue Yonder (anteriormente JDA Software).
- **QuickBooks Inventory:** QuickBooks es un software de contabilidad ampliamente utilizado que incluye módulos de gestión de inventarios para pequeñas y medianas empresas. Ofrece características básicas de seguimiento de existencias y generación de informes.
- **Zoho Inventory:** Zoho Inventory es una solución basada en la nube que permite a las empresas gestionar su inventario, ventas y compras en un solo sistema. Ofrece seguimiento de productos, gestión de pedidos y análisis de inventarios.
- **Odoo Inventory:** Odoo es una plataforma de código abierto que incluye una amplia gama de módulos empresariales. Su módulo de inventario proporciona funcionalidades como el seguimiento de existencias, la gestión de ubicaciones y la optimización de pedidos.
- **TradeGecko:** TradeGecko es una solución de gestión de inventarios en la nube diseñada para empresas de comercio electrónico y mayoristas. Ofrece seguimiento de existencias, administración de pedidos y sincronización con múltiples canales de venta.
- **Fishbowl Inventory:** Fishbowl es un sistema de gestión de inventarios diseñado para integrarse con QuickBooks. Ofrece funciones avanzadas de seguimiento de existencias, gestión de órdenes de compra y ventas, y automatización de procesos.

- **DEAR Inventory:** DEAR Inventory es un sistema de gestión de inventarios en la nube que se integra con múltiples plataformas de comercio electrónico. Ofrece seguimiento de productos, gestión de pedidos y análisis de inventarios.
- **Cin7:** Cin7 es una solución de gestión de inventarios que se adapta a empresas minoristas, mayoristas y de fabricación. Ofrece funciones de seguimiento de existencias, gestión de pedidos y automatización de procesos.

Comparación con SAP Inventory Management

SAP Inventory Management es parte de la suite de productos SAP ERP (Enterprise Resource Planning) una de las soluciones de software empresarial más utilizadas en todo el mundo. Esto ha contribuido a su amplia adopción en empresas grandes y medianas en diversos sectores.

Este sistema se va a tomar en cuenta por su amplia adopción en diversas empresas de manera que se va a hacer una comparación de sus características con las funcionalidades del sistema a desarrollar.

1. Registro de Productos:

- **Sistema:** Permite a los usuarios registrar nuevos productos en el sistema, incluyendo información como nombre, descripción, precio y cantidad inicial en stock.
- **SAP Inventory Management:** Permite la creación de registros de productos con detalles como nombre, descripción, precio y unidades iniciales en stock. Además, SAP permite la importación masiva de datos de productos desde fuentes externas.

2. Visualización de Inventarios:

- **Sistema:** Permite a los usuarios acceder y visualizar el inventario existente en el sistema.

- SAP Inventory Management: Ofrece una vista completa de los inventarios, mostrando el estado actual de las existencias, ubicaciones y movimientos de productos. Además, permite generar informes detallados sobre el inventario.

4. Eliminación de Productos:

- Sistema: Proporciona a los usuarios la opción de eliminar productos obsoleto.
- SAP Inventory Management: Permite la eliminación de productos, pero generalmente con procedimientos de control de acceso y registros de auditoría para rastrear quién y cuándo se eliminaron los productos.

5. Búsqueda de Productos:

- Sistema: Permite a los usuarios buscar productos por su ID.
- SAP Inventory Management: Ofrece capacidades de búsqueda avanzada y filtrado que permiten a los usuarios buscar productos utilizando múltiples criterios, como nombre, código, ubicación, etc.

6. Interfaz de Usuario Intuitiva:

- Sistema: Cuenta con una interfaz de usuario intuitiva diseñada para usuarios sin un conocimiento técnico.
- SAP Inventory Management: Proporciona una interfaz de usuario profesional, pero suele requerir una curva de aprendizaje mayor debido a su complejidad y funcionalidades avanzadas.

Tecnologías empleadas

- Java

Java es el lenguaje de programación central en este proyecto. Debido a su portabilidad, flexibilidad y robustez Java permite la creación de aplicaciones escalables y confiables, lo que lo convierte en una elección ideal para un sistema de gestión de inventarios que pueda escalar con futuras mejoras.

- Biblioteca Swing

Swing es una biblioteca gráfica de Java que se utiliza para desarrollar interfaces de usuario. Proporciona componentes gráficos como botones, etiquetas, tablas y paneles que son esenciales para crear una interfaz de usuario intuitiva y atractiva.

- IDE

Se va a hacer uso del IDE NetBeans para desarrollar el proyecto. Esta herramienta proporciona un entorno de desarrollo eficiente y facilita la escritura de código Java y la creación de interfaces de usuario con Swing.

Modelo en Cascada de desarrollo del Software

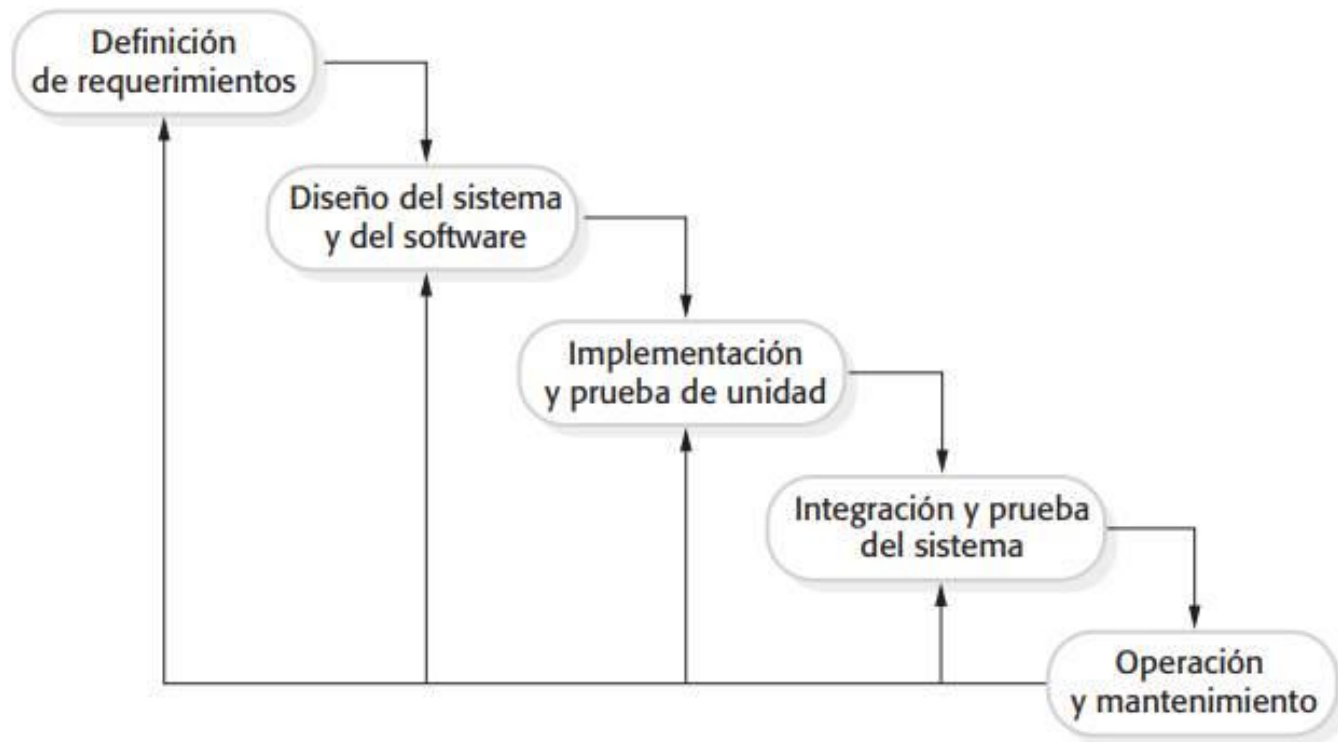


Figura 1. Modelo de Desarrollo de Software

Nota. Ingeniería de Software. (p. 30), por Ian Sommerville, 2011, Pearson.

Etapas del modelo en Cascada

1. Definición de requerimientos

Definición: En esta etapa los servicios, restricciones y metas, se definen a partir de las consultas con los usuarios, o bien se dividen aquí en los funcionales y los No funcionales. Entonces se definen en detalle y sirven como una especificación del sistema.

2. Diseño del sistema y del software

Definición: En esta fase se dividen los requerimientos en el hardware y el software. Posteriormente se establece una arquitectura completa del sistema, el diseño del software idéntica y establece las abstracciones fundamentales para el correcto desarrollo del software.

3. Implementación

Definición: Durante esta etapa el diseño del software se identifica problemas con los requerimientos; durante el diseño del código se encuentran problemas y así sucesivamente. El proceso de software no es modelo lineal simple, si no que implica una serie de iteraciones de las actividades de desarrollo.

4. Integración y pruebas de sistema

Definición: Los programas o las unidades individuales se integran y se prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente.

5. Operación y mantenimiento

Definición: En esta última etapa el software se pone en funcionamiento, se descubren errores y omisiones en los requerimientos originales del software. Los errores de programación y de diseño emergen. Se identifica la necesidad de una nueva funcionalidad.

Implementación del Modelo en Cascada

1. Definición de requerimientos

Requisitos funcionales

1. **Registro de nuevos productos:** Los usuarios podrán ingresar los detalles de un nuevo producto, incluyendo nombre, id, descripción, precio y cantidad inicial.
2. **Visualización de Inventario:** Los usuarios podrán acceder y visualizar el inventario existente en el sistema.
3. **Búsqueda de Productos:** Los usuarios podrán buscar productos por su Id.
4. **Actualización de información de productos:** Se permitirá a los usuarios modificar la información de un producto, como cambiar su precio, nivel de existencias, descripción y nombre.
5. **Eliminación de productos:** Los usuarios podrán eliminar productos que ya no están disponibles o que se vayan a retirar del inventario.
6. **Creación de nuevos usuarios:** El usuario principal del sistema va a tener la opción de agregar nuevos usuarios al sistema y definir los roles que tendrá cada uno, el sistema manejará tres roles:
 - **Administrador:** Este rol va a tener acceso a todas las operaciones del sistema y en particular a la operación de crear nuevos usuarios.
 - **Gestor de inventario:** Este rol va a tener acceso a las operaciones de agregar producto, buscar producto, listar productos la operación de eliminar productos, editar producto y crear nuevos usuarios están restringidas a este rol.
 - **Usuario regular:** Este rol va a tener acceso a las operaciones de buscar producto y listar productos el resto de las operaciones van a estar restringidas

7. Menú de opciones: El usuario puede seleccionar una de las siguientes opciones por medio de una interfaz en donde cada opción va a tener un botón:

- Agregar producto al inventario:
- Eliminar producto del inventario:
- Buscar producto en el inventario.
- Listar productos en el inventario.
- Editar producto en el inventario.
- Crear nuevo usuario
- Cerrar sesión.

Una vez que se haya seleccionado la opción se va a generar un panel con los campos necesarios para cada uno a excepción del botón de cerrar sesión el cual vuelve al inicio de sesión del programa para restringir el acceso a cualquiera.

Requisitos no funcionales

1. **Rendimiento:** El sistema debe ser capaz de manejar un gran número de productos y usuarios de manera eficiente.
2. **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de entender incluso para usuarios que no tienen conocimientos técnicos.
3. **Escalabilidad:** El sistema debe ser diseñado de manera que pueda escalarse fácilmente en caso de un aumento en la cantidad de productos o usuarios.
4. **Mantenibilidad:** El código debe estar bien estructurado y documentado para facilitar futuras actualizaciones y mejoras.
5. **Seguridad de Acceso:** Se implementará un sistema de autenticación para garantizar que solo usuarios autorizados puedan acceder al sistema.

6. **Pruebas y Validación:** Se llevarán a cabo pruebas exhaustivas, incluyendo pruebas de unidad y pruebas de integración para garantizar la calidad y el funcionamiento adecuado del sistema.
7. **Tiempo de respuesta:** Establecer tiempos máximos de respuesta para consultas y actualizaciones de inventario. Es decir, la capacidad del sistema para responder rápidamente a las solicitudes de los usuarios.
8. **Integración con sistemas externos:** Posibilidad de integrar el sistema con otros sistemas.
9. **Disponibilidad:** Garantizar que el sistema esté disponible para los usuarios la mayor parte del tiempo.

El sistema debe llevar un correcto manejo de excepciones y errores para garantizar que el programa no truene de manera inesperada. El programa debe ser capaz de manejar estos problemas de una manera controlada y proporcionar una respuesta adecuada al usuario o al sistema.

2. Diseño del sistema y del software

1. **Arquitectura:** Para el sistema en cuestión se va a emplear la arquitectura cliente pesado pues el enfoque en el diseño del sistema de software la mayor parte de la lógica y procesamiento se realiza en el lado del cliente, es decir, en la máquina del usuario. La mayoría de las operaciones realizadas con la gestión de inventarios, como el ingreso de productos, consulta, actualización y eliminación de stock se implementarán en la aplicación que se ejecuta en la computadora del usuario esto significa que la aplicación cliente es responsable de procesar datos y llevar cabo las operaciones necesarias.

Interfaz de usuario en el cliente: La interfaz de usuario (UI) se encuentra en la aplicación cliente esto significa que los usuarios interactúan directamente con la aplicación instalada en sus computadoras para realizar todas las operaciones relacionadas con el inventario.

Requisitos de hardware y software en el cliente: Dado que la lógica de la aplicación reside en el cliente, es importante que los usuarios tengan el hardware y el software necesarios para ejecutar la aplicación correctamente.

2. Diseño del software

Para el desarrollo del sistema se va a emplear el paradigma orientado a objetos que enfoque que se basa en la representación de entidades del mundo real como objetos los cuales tienen atributos y comportamientos asociados. Al aplicar este paradigma al desarrollo del sistema de gestión de inventarios los productos y usuarios se pueden representar como objetos, cada uno con atributos que los identifican además el código suele se suele dividir en módulos con funciones específicas, lo que facilita la creación y división de diferentes componentes del sistema.

2.1 Identificación de componentes

- **Componentes:**
- Una clase que maneja los atributos de los productos como existencias, descripción, id, nombre y precios y permite obtener los valores asociados a un determinado producto y una clase que maneja atributos básicos de un usuario como el id, nombre del usuario y contraseña.
- Una clase que maneja las operaciones con el inventario como buscar productos, registrar productos, eliminar productos, listar los productos del inventario y editar información con respecto al nombre, descripción, precio y existencias
- Una clase en la cual se hace la autenticación del usuario con la cual se va a dar acceso a las operaciones del sistema.
- Una interfaz de inicio de sesión antes de acceder a las funcionalidades del sistema y una interfaz que permite acceder a las operaciones para el manejo del inventario.

2.2 Diseño de clases y métodos

Clase Inventario:

- Descripción: La clase mantiene una lista de objetos de la clase **Producto** y permite realizar operaciones como agregar, eliminar, buscar, listar y editar productos en el inventario.
- Variables de instancia:
- **private ArrayList<Producto> productos:** Una lista que almacena objetos de la clase **Producto**.
- Métodos:
- Constructor **Inventario()**: Inicializa la lista de productos como una lista vacía al crear un objeto de la clase **Inventario**.
- **getProductos()**: Devuelve una lista de productos en el inventario.
- **setProductos(ArrayList<Producto> productos)**: Establece la lista de productos en el inventario.
- **agregarProducto(Producto producto)**: Agrega un nuevo producto al inventario, verificando si ya existe un producto con la misma ID.
- **eliminarProducto(Producto producto)**: Elimina un producto específico del inventario.
- **editarProducto(Producto producto, String campo, String nuevoValor)**: Edita un campo específico (nombre, descripción, existencia, precio) de un producto en el inventario.
- **buscarProducto(int idProducto)**: Busca un producto por su ID en el inventario y devuelve el objeto **Producto** correspondiente.
- **listarProductos()**: Devuelve una lista de todos los productos en el inventario.

Clase **Producto**:

- Descripción: Esta clase representa un producto con atributos como ID, nombre, descripción, existencia y precio.
- Variables de instancia:
- **private int idProducto:** Identificador único del producto.
- **private String nombre:** Nombre del producto.

- private String descripcion: Descripción del producto.
- private int existencia: Cantidad en existencia del producto.
- private double precio: Precio del producto.
- Métodos:
- Constructor Producto(int idProducto, String nombre, String descripcion, int existencia, double precio): Inicializa un objeto Producto con valores proporcionados.

Métodos Getter y Setter para cada variable de instancia:

- getIdProducto(), setIdProducto(int idProducto): Devuelve identificador único del producto.
- getNombre(): Devuelve el nombre del producto
- setNombre(String nombre): Establece el nombre del producto.
- getDescripcion(): Devuelve la descripción del producto.
- setDescripcion(String descripcion): Establece la descripción del producto.
- getExistencia(): Devuelve la cantidad de existencias del producto.
- setExistencia(int existencia): Establece la cantidad de existencias del producto
- getPrecio(): Devuelve el precio del producto.
- setPrecio(double precio): Establece el precio del producto.
- actualizarExistencia(int cantidad): Actualiza la cantidad en existencia del producto sumando la cantidad proporcionada.
- toString(): Devuelve una representación en formato de cadena del producto, mostrando ID, nombre, descripción, existencia y precio.

Clase **SistemaGestionInventarios**:

- Descripción: La clase coordina las acciones relacionadas con usuarios y productos en un sistema de inventario. Utiliza objetos de las clases **Usuario** e **Inventario** para realizar operaciones como agregar usuarios, iniciar sesión y gestionar productos en el inventario.
- Variables de instancia:

- `private JDialog dialog`: Es una ventana de diálogo que se utiliza para mostrar mensajes al usuario.
- `private ArrayList<Usuario> usuarios`: Representa una lista que almacena objetos de la clase `Usuario` representando a los usuarios registrados en el sistema.
- `private Inventario inventario`: Instancia de la clase `Inventario` que se utiliza para gestionar productos en el sistema.
- `public Usuario usuarioLogueado`: Representa a usuario que ha iniciado sesión en el sistema.
- Métodos:
 - Constructor `SistemaGestionInventarios()`: Inicializa la ventana de diálogo, la lista de usuarios, la instancia de `Inventario`, y establece el usuario logueado como nulo al crear un objeto de la clase.
 - `agregarUsuario(Usuario usuario)`: Agrega un nuevo usuario al sistema verificando si ya existe un usuario con el mismo ID.
 - `existeUsuarioConId(int idUsuario)`: Método privado que verifica si ya existe un usuario con el mismo ID en la lista de usuarios.
 - `iniciarSesion(String nombreUsuario, String contraseña)`: Inicia sesión de un usuario en el sistema verificando el nombre de usuario y la contraseña.
 - `cerrarSesion()`: Cierra la sesión del usuario actual.
 - `agregarProductoAlInventario(Usuario usuario, Producto producto)`: Agrega un producto al inventario si el usuario logueado y tiene los permisos necesarios.
 - `buscarProductoEnInventario(int idProducto)`: Busca un producto en el inventario por su ID siempre que haya un usuario logueado.
 - `eliminarProducto(Usuario usuario, Producto producto)`: Elimina un producto del inventario, si el usuario logueado y tiene los permisos necesarios.
 - `listarProductosEnInventario()`: Lista los productos en el inventario, si hay un usuario logueado.

- editarProducto(Usuario usuario, Producto producto, String campo, String nuevoValor): Edita un producto del inventario, si el usuario logueado tiene los permisos necesarios.
- agregarUsuario(Usuario usuario): Realiza una validación llamando al método existeUsuarioConId y después agrega un nuevo usuario a la lista de usuarios.
- existeUsuarioConId (int idUsuario): Recorre la lista de usuarios y verifica si el id ya está registrado retorna un valor true si se encontró un id igual y retorna false en caso contrario.
- setUsuarios(ArrayList<Usuario> usuarios): Establece la lista de usuarios en el sistema.
- getUsuarios(): Retorna la lista de usuarios en el sistema.
- getInventario(): Retorna la instancia de Inventario asociada al sistema.

Clase **Usuario**:

- Descripción: La clase representa un usuario en un sistema, con identificación única, nombre de usuario, contraseña y un rol asociado.
- Variables de instancia:
 - private int idUsuario: Identificador único del usuario.
 - private String nombreUsuario: Nombre de usuario.
 - private String contraseña: Contraseña del usuario.
 - private String rol: Rol asociado al usuario.
- Métodos:
 - Constructor Usuario(int idUsuario, String nombreUsuario, String contraseña, String rol): Inicializa un objeto Usuario con los valores proporcionados para el identificador, nombre de usuario, contraseña y rol.
 - getIdUsuario(): Retorna el identificador único del usuario.
 - setIdUsuario(int idUsuario): Establece el identificador único del usuario.
 - getNombreUsuario(): Retorna el nombre de usuario.

- **setNombreUsuario(String nombreUsuario):** Establece el nombre de usuario.
- **getContraseña():** Retorna la contraseña del usuario.
- **setContraseña(String contraseña):** Establece la contraseña del usuario.
- **getRol():** Retorna el rol del usuario.
- **setRol(String rol):** Establece el rol del usuario.

Clase GuardarArchivo

Descripción: La clase proporciona métodos para guardar información de usuarios y productos en archivos CSV

Métodos:

- **guardarUsuarios(String archivoUsuarios, ArrayList<Usuario> usuarios):** Este método recibe el nombre del archivo y una lista de objetos Usuario
- **guardarProductos(String archivoProductos, ArrayList<Producto> productos):** Este método recibe el nombre del archivo y una lista de objetos Producto.

Clase CargarArchivo

Descripción: La clase proporciona métodos para cargar información de usuarios y productos desde archivos CSV.

Métodos

- **cargarUsuarios(String archivoUsuarios):** Este método lee datos desde un archivo CSV que contiene información de usuarios.
- **cargarProductos(String archivoProductos):** Este método lee datos desde un archivo CSV que contiene información de producto.

Clase **Main** (Clase principal con el método **main**):

- Descripción: La clase contiene el método principal que inicia la aplicación

Método Principal **main(String [] args):**

- Este metodo es el punto de entrada para la aplicación
- Crea una instancia de SistemaGestionInventarios llamada sistema.

- Carga usuarios desde el archivo usuarios.csv utilizando `CargarArchivo.cargarUsuarios()` si la carga es exitosa se actualiza la lista de usuarios en el sistema.
- Carga productos desde el archivo productos.csv utilizando `CargarArchivo.cargarProductos()` si la carga es exitosa se actualiza la lista de productos en el inventario del sistema.
- Inicia la ventana de inicio de sesión `LoginWindow` pasando una instancia del sistema como parámetro.

Diagrama de clases UML

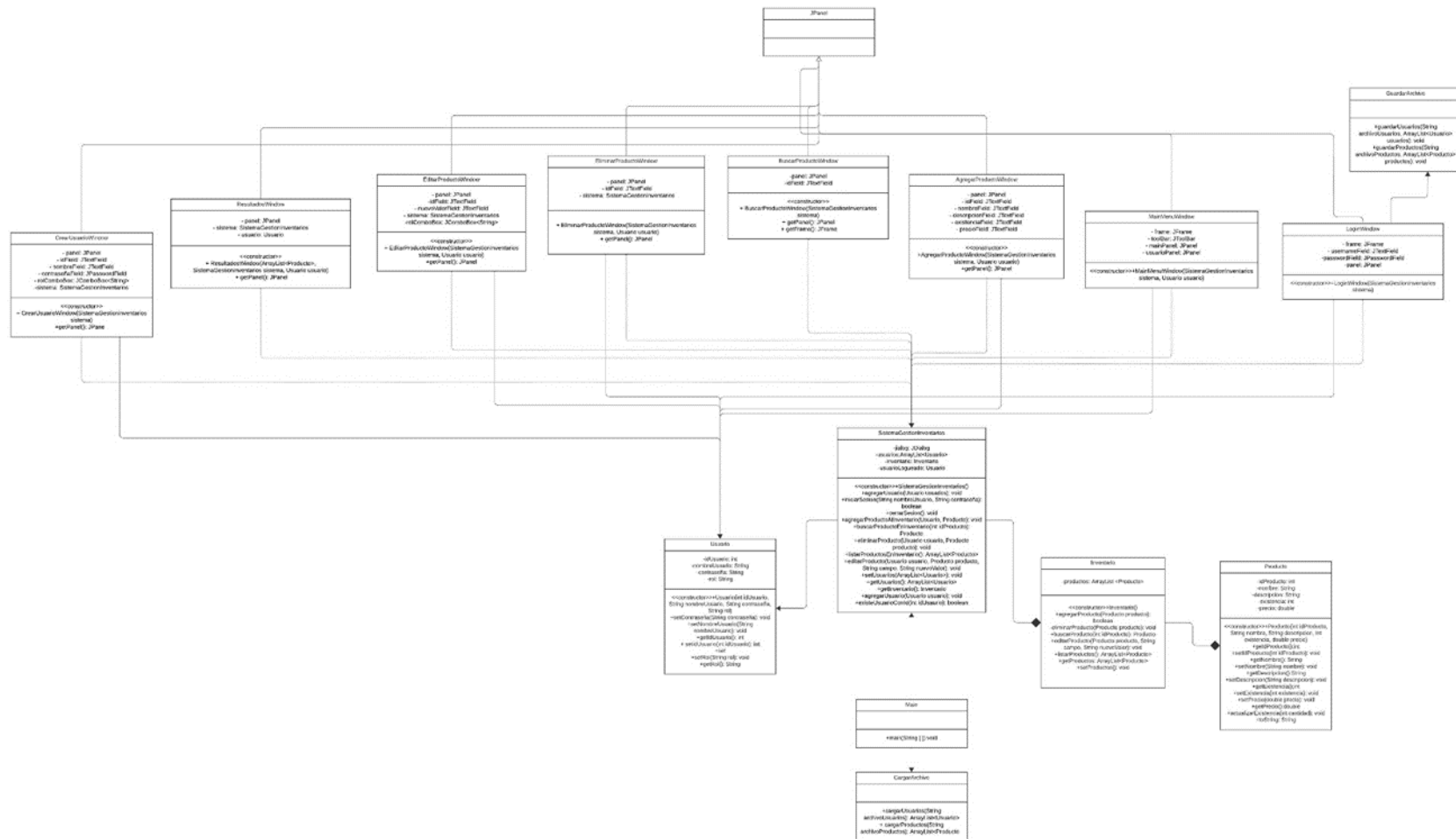


Figura 2. Diagrama UML

3. Implementación

Elección de tecnología: Para el desarrollo del sistema se va a emplear el lenguaje de programación Java y NetBeans como entorno de desarrollo integrado. Además, se va a utilizar GitHub para llevar un control de versiones y volver a una versión anterior del software en caso de que una implementación del código que produzca fallos en los demás componentes del sistema.

Código fuente: En base al diseño de clase sus métodos y variables de instancia se codifica cada una de estas.

Clase Usuario

```
6 public class Usuario {
7     private int idUsuario;        // Identificador único del usuario.
8     private String nombreUsuario; // Nombre de usuario.
9     private String contraseña;    // Contraseña del usuario.
10    private String rol;            // Rol de usuario
11
12    /**
13     * Constructor para crear un objeto de la clase Usuario.
14     *
15     * @param idUsuario    Identificador único del usuario.
16     * @param nombreUsuario Nombre de usuario.
17     * @param contraseña    Contraseña del usuario.
18     * @param rol           Rol del usuario.
19     */
20
21    public Usuario(int idUsuario, String nombreUsuario, String contraseña, String rol) {
22        this.idUsuario = idUsuario;
23        this.nombreUsuario = nombreUsuario;
24        this.contraseña = contraseña;
25        this.rol = rol;
26    }
27
28    /**
29     * Obtiene el identificador único del usuario.
30     *
31     * @return El identificador único del usuario.
32     */
33    public int getIdUsuario() {
34        return idUsuario;
35    }
36
```

Figura 3. Implementacion

Clase Producto

```
8 public class Producto {
9     private int idProducto;        // Identificador único del producto.
10    private String nombre;          // Nombre del producto.
11    private String descripcion;      // Descripción del producto.
12    private int existencia;          // Cantidad en existencia del producto.
13    private double precio;           // Precio del producto.
14
15    /**
16     * Constructor para crear un objeto de la clase Producto.
17     *
18     * @param idProducto    Identificador único del producto.
19     * @param nombre         Nombre del producto.
20     * @param descripcion    Descripción del producto.
21     * @param existencia     Cantidad en existencia del producto.
22     * @param precio         Precio del producto.
23     */
24
25    public Producto(int idProducto, String nombre, String descripcion, int existencia, double precio) {
26        this.idProducto = idProducto;
27        this.nombre = nombre;
28        this.descripcion = descripcion;
29        this.existencia = existencia;
30        this.precio = precio;
31    }
32
33    /**
34     * Obtiene el identificador único del producto.
35     *
36     * @return El identificador único del producto.
37     */
38    public int getIdProducto() {
39        return idProducto;
40    }
41
```

Figura 4. Implementacion

Clase Inventario

```
16 public Inventario() {
17     this.productos = new ArrayList<>();
18 }
19
20 public ArrayList<Producto> getProductos() {
21     return productos;
22 }
23
24 /**
25  * Establece la lista de productos en el inventario.
26  *
27  * @param productos La nueva lista de productos.
28  */
29 public void setProductos(ArrayList<Producto> productos) {
30     this.productos = productos;
31 }
32
33 /**
34  * Agrega un producto al inventario.
35  *
36  * @param producto El objeto Producto que se desea agregar al inventario.
37  */
38 public boolean agregarProducto(Producto producto) {
39     int idProducto = producto.getIdProducto();
40
41     // Verificar si ya existe un producto con la misma ID
42     for (Producto p : productos) {
43         if (p.getIdProducto() == idProducto) {
44             return false; // Producto con la misma ID ya existe
45         }
46     }
47 }
```

Figura 5. Implementacion

Clase SistemaGestionInventarios

```
22 public SistemaGestionInventarios() {
23     dialog = new JDialog();
24     this.usuarios = new ArrayList<>();
25     this.inventario = new Inventario();
26     this.usuarioLogueado = null;
27 }
28
29 /**
30  * Agrega un usuario al sistema.
31  *
32  * @param usuario El objeto Usuario que se desea agregar.
33  */
34 public void agregarUsuario(Usuario usuario) {
35     if (!existeUsuarioConId(idUsuario: usuario.getIdUsuario())) {
36         usuarios.add(e: usuario);
37         JOptionPane.showMessageDialog(parentComponent: dialog, message: "Usuario creado con éxito.");
38     } else {
39         JOptionPane.showMessageDialog(parentComponent: dialog, message: "Ya existe un usuario con el mismo ID. No se puede agregar.");
40     }
41 }
42
43 private boolean existeUsuarioConId(int idUsuario) {
44     for (Usuario usuario : usuarios) {
45         if (usuario.getIdUsuario() == idUsuario) {
46             return true; // Ya existe un usuario con el mismo ID
47         }
48     }
49     return false; // No hay usuario con el mismo ID
50 }
51 }
```

Figura 6. Implementacion

Clase GuardarArchivo

```
17 public class GuardarArchivo {
18     public static void guardarUsuarios(String archivoUsuarios, ArrayList<Usuario> usuarios) {
19         try (BufferedWriter writer = new BufferedWriter(new FileWriter(archivoUsuarios))) {
20             for (Usuario usuario : usuarios) {
21                 writer.write(usuario.getIdUsuario() + "," + usuario.getNombreUsuario() + "," + usuario.getContraseña() + "," + usuario.getRol());
22                 writer.newLine();
23             }
24         } catch (IOException e) {
25             e.printStackTrace();
26         }
27     }
28
29     public static void guardarProductos(String archivoProductos, ArrayList<Producto> productos) {
30         try (BufferedWriter writer = new BufferedWriter(new FileWriter(archivoProductos))) {
31             for (Producto producto : productos) {
32                 writer.write(
33                     producto.getIdProducto() + "," +
34                     producto.getNombre() + "," +
35                     producto.getDescripcion() + "," +
36                     producto.getExistencia() + "," +
37                     producto.getPrecio()
38                 );
39                 writer.newLine();
40             }
41         } catch (IOException e) {
42             e.printStackTrace();
43         }
44     }
45 }
```

Figura 7. Implementacion

Clase CargarArchivo

```
17 public class CargarArchivo {
18     public static ArrayList<Usuario> cargarUsuarios(String archivoUsuarios) {
19         ArrayList<Usuario> usuarios = new ArrayList<>();
20         try (BufferedReader reader = new BufferedReader(new FileReader(archivoUsuarios))) {
21             String line;
22             while ((line = reader.readLine()) != null) {
23                 String[] usuarioData = line.split(regex: ",");
24                 if (usuarioData.length == 4) {
25                     int idUsuario = Integer.parseInt(usuarioData[0]);
26                     String nombreUsuario = usuarioData[1];
27                     String contraseña = usuarioData[2];
28                     String rol = usuarioData[3];
29                     Usuario usuario = new Usuario(idUsuario, nombreUsuario, contraseña, rol);
30                     usuarios.add(usuario);
31                 }
32             }
33         } catch (IOException e) {
34             e.printStackTrace();
35         }
36         return usuarios;
37     }
38
39     public static ArrayList<Producto> cargarProductos(String archivoProductos) {
40         ArrayList<Producto> productos = new ArrayList<>();
41         try (BufferedReader reader = new BufferedReader(new FileReader(archivoProductos))) {
42             String line;
43             while ((line = reader.readLine()) != null) {
44                 String[] productoData = line.split(regex: ",");
45                 if (productoData.length == 5) {
46                     int idProducto = Integer.parseInt(productoData[0]);
47                     String nombre = productoData[1];

```

Figura 8. Implementacion

4. Integración y pruebas de sistema

En esta fase se lleva a cabo el diseño de casos de prueba para evaluar la integración y correcto funcionamiento de los componentes del sistema.

1. **Agregar Producto al Inventario:**

Descripción: Este caso de prueba evalúa la funcionalidad de agregar un nuevo producto al inventario y verifica que se haya agregado correctamente.

Pasos:

1. Inicie el sistema de gestión de inventarios.
2. Inicie sesión con un usuario válido (por ejemplo, "usuario1" y "contraseña1").
3. Agregue un nuevo producto al inventario con los siguientes detalles:
 - ID del producto: 9824
 - Nombre del producto: "jabón"
 - Descripción del producto: "Descripción del producto Jabón"
 - Existencia del producto: 10
 - Precio del producto: 12.5

Resultados Esperados:

- El sistema debe aceptar las credenciales de inicio de sesión y dar acceso a las funciones del sistema.
- Después de agregar el producto, el sistema debe mostrar un mensaje de éxito.
- El producto "Jabón" debe estar presente en el inventario.
- Al listar los productos en el inventario, se debe mostrar "Jabón" con los detalles proporcionados.

Condiciones de Éxito:

- El producto se agrega con éxito al inventario.
- Los detalles del producto agregado coinciden con los proporcionados en los pasos.

Condiciones de Fallo:

- Las credenciales de inicio de sesión son incorrectas.
- No se muestra un mensaje de que la operación se realizó con éxito.
- El producto no aparece en el inventario.
- Al listar los productos en el inventario, no se encuentra "Producto1" con los detalles proporcionados.

2. Eliminar Producto del Inventario:

Descripción: Este caso de prueba evalúa la funcionalidad de eliminar un producto del inventario y verifica que el producto ya no esté en el inventario después de ser eliminado.

Pasos:

1. Iniciar el sistema de gestión de inventarios.
 2. Iniciar sesión con un usuario válido (por ejemplo, "usuario1" y "contraseña1").
 3. Agregue un nuevo producto al inventario con sus respectivos datos:
- ID del producto: 1
 - Nombre del producto: "Producto1"
 - Descripción del producto: "Descripción del Producto 1"
 - Existencia del producto: 10
 - Precio del producto: 50.0
 - Elimine el producto del inventario utilizando su ID (ID del producto: 1).

Resultados Esperados:

- El sistema debe aceptar las credenciales de inicio de sesión y mostrar un mensaje que indique que la sesión fue iniciada con éxito.

- Después de agregar el producto, el sistema debe mostrar un mensaje que indique que el producto fue agregado.
- El producto "Producto1" debe estar presente en el inventario después de agregarlo.
- Después de eliminar el producto, el sistema debe mostrar un mensaje que indique que el producto fue eliminado.
- El producto "Producto1" no debe estar en el inventario.
- Si el usuario no tiene el permiso para eliminar producto del inventario aparecerá un mensaje que indique que no tiene el permiso para realizar esta acción.
- **Condiciones de Éxito:**
 - El producto se agrega con éxito al inventario.
 - El producto se elimina con éxito del inventario.
 - Después de la eliminación, el producto no aparece en el inventario.
 - Los mensajes de éxito se muestran adecuadamente.

Condiciones de Fallo:

- Las credenciales de inicio de sesión son incorrectas.
- No se muestra un mensaje de producto agregado.
- No se muestra un mensaje de producto eliminado.
- El producto permanece en el inventario después de la eliminación.

3. Editar Producto en el Inventario:

Descripción: Este caso de prueba evalúa la funcionalidad de editar un producto en el inventario después de agregarlo y verifica que los cambios realizados en varios de sus atributos (nombre, descripción, existencia y precio) se reflejen correctamente en el producto.

Pasos:

1. Iniciar el sistema de gestión de inventarios.
2. Iniciar sesión con un usuario válido (por ejemplo, "usuario1" y "contraseña1").
3. Agregue un nuevo producto al inventario con sus datos correspondientes:
4. ID del producto: 1
5. Nombre del producto: "Producto1"
6. Descripción del producto: "Descripción del Producto 1"
7. Existencia del producto: 10
8. Precio del producto: 50.0
9. Editar el producto recién agregado, escribiendo el ID para poder realizar una modificación y después intentar cambiar uno de sus atributos:
10. Cambie el nombre a "Nuevo valor"
11. Cambie la descripción a "Nuevo valor"
12. Cambie la existencia a 15
13. Cambie el precio a 60.0

Resultados Esperados:

- El sistema debe aceptar las credenciales de inicio de sesión y mostrar un mensaje que indique el inicio de sesión con éxito.
- Después de agregar el producto, el sistema debe mostrar un mensaje que indique que el producto fue agregado al inventario con éxito.
- El producto "Producto1" con los detalles originales debe estar presente en el inventario.
- Después de la edición del producto, el sistema debe mostrar un mensaje que indique que el producto fue editado con éxito.
- El producto editado debe tener los nuevos valores en los atributos especificados (nombre, descripción, existencia y precio).

Condiciones de Éxito:

- Los productos se agregan con éxito al inventario.
- La edición del producto se realiza con éxito.
- Los mensajes de éxito se muestran adecuadamente.
- El producto editado tiene los nuevos valores en los atributos especificados.

Condiciones de Fallo:

- Las credenciales de inicio de sesión son incorrectas.
- No se muestra el mensaje que indica que el producto fue agregado al inventario con éxito.
- La edición del producto no se realiza correctamente.
- Los mensajes de éxito no se muestran correctamente.
- Los cambios en los atributos del producto no se reflejan correctamente.

Buscar Producto en el Inventario:

Descripción: Este caso de prueba evalúa la funcionalidad de buscar un producto en el inventario y mostrar la información asociada a ese producto.

Pasos:

1. Iniciar el sistema de gestión de inventarios.
2. Iniciar sesión con un usuario válido (por ejemplo, "usuario1" y "contraseña1").
3. Agregar varios productos al inventario con detalles diferentes, incluyendo un producto con un ID específico que se buscará más adelante.
4. Buscar el producto en el inventario por su ID, el cual se conoce previamente.

Resultados Esperados:

- El sistema debe aceptar las credenciales de inicio de sesión y mostrar un mensaje que indica el inicio de Sesión se realizó con éxito.
- Varios productos deben estar presentes en el inventario con sus respectivos detalles.
- El producto buscado por su ID debe ser encontrado y sus detalles deben coincidir con los detalles originales.

Condiciones de Éxito:

- Los productos se agregan con éxito al inventario.
- El producto buscado por su ID es encontrado en el inventario.

Condiciones de Fallo:

- Las credenciales de inicio de sesión son incorrectas.
- El producto buscado no se encuentra en el inventario.

Listar Productos en el Inventario:

Descripción: Este caso de prueba evalúa la funcionalidad de listar todos los productos en el inventario y verifica que se muestren todos los productos correctamente.

Pasos:

1. Inicie el sistema de gestión de inventarios.
2. Inicie sesión con un usuario válido (por ejemplo, "usuario1" y "contraseña1").
3. Agregue varios productos al inventario con detalles diferentes.
4. Liste todos los productos en el inventario.

Resultados Esperados:

- El sistema debe aceptar las credenciales de inicio de sesión y mostrar un mensaje que indique que la sesión fue iniciada con éxito.

- Varios productos deben estar presentes en el inventario con sus respectivos detalles.
- Al listar los productos en el inventario, todos los productos agregados previamente deben mostrarse correctamente, incluyendo sus detalles.

Condiciones de Éxito:

- Los productos se agregan con éxito al inventario.
- Al listar los productos, se muestran todos los productos en el inventario.
- Los detalles de los productos coinciden con los detalles originales.

Condiciones de Fallo:

- Las credenciales de inicio de sesión son incorrectas.
- No se muestran todos los productos en el inventario al listarlos.
- Los detalles de los productos no coinciden con los detalles originales.

Crear Nuevo Usuario:

Descripción: Este caso de prueba evalúa la funcionalidad de crear nuevo usuario en el sistema de gestión de inventarios y verificar que el nuevo usuario pueda iniciar sesión y asegurar que tiene acceso a las operaciones permitidas de acuerdo con su rol de usuario.

Pasos:

1. Iniciar el sistema de gestión de inventarios.
2. Iniciar sesión con el usuario de administrador con las siguientes credenciales.
3. Nombre de usuario: usuario1.
4. Contraseña: contraseña1
5. Crear un nuevo usuario con las siguientes credenciales.
6. ID de usuario: 1234.
7. Nombre de usuario: usuario3.
8. Contraseña: contraseña3.

9. Rol: Especificar un rol: Administrador, gestor de inventario o usuario regular.
10. Crear el nuevo usuario.
11. Cerrar la sesión actual.
12. Iniciar sesión con las credenciales del nuevo usuario.

Resultados esperados:

- El sistema debe aceptar las credenciales del nuevo usuario y mostrar un mensaje que indique que el usuario fue creado con éxito.
- Al volver iniciar sesión el sistema debe aceptar las credenciales del nuevo usuario y mostrar el mensaje de inicio de sesión exitoso.
- Al cerrar la sesión actual y después iniciar sesión con las credenciales del nuevo usuario se debe mostrar un mensaje de inicio de sesión exitoso.
- El nuevo usuario debe tener acceso a aquellas operaciones que han sido definidas de acuerdo con su rol:
 - **Administrador:**
 - Agregar productos
 - Buscar productos
 - Editar producto
 - Eliminar Producto
 - Listar Productos
 - Crear nuevo usuario
 - **Gestor de Inventario:**
 - Agregar productos
 - Editar productos
 - Buscar productos
 - Listar Productos
 - **Usuario regular:**
 - Buscar productos
 - Listar Productos

- Cuando se intenta acceder a una función que el usuario no tiene permitido por su rol de usuario el sistema manda un mensaje indicando que no tiene permisos para realizar la acción.

Condiciones de Éxito:

- El nuevo usuario se crea con éxito.
- El nuevo usuario puede acceder a las operaciones definidas de acuerdo al rol de usuario.
- Al iniciar sesión con el nuevo usuario el inicio es exitoso y se puede acceder a las operaciones del sistema.

Condiciones de Fallo:

- El nuevo usuario con las credenciales especificadas no se puede crear.
- Se puede crear el mismo usuario con el mismo id, pero con diferentes roles.
- Al iniciar sesión con el nuevo usuario se rechaza el acceso al sistema y se muestra el mensaje de inicio de sesión fallido.
- Cuando se concede el acceso al sistema no se verifica adecuadamente el rol de usuario y se pueden acceder a funciones que no tiene permitidas los roles especificados.
 - El rol de Gestor de Inventarios puede acceder a las funciones de:
 - Eliminar producto
 - Crear nuevo usuario
 - El rol de Usuario regular puede acceder a las funciones de:
 - Agregar producto
 - Editar producto
 - Eliminar producto
 - Crear nuevo usuario
- Cuando se concede el acceso al sistema y no se verifica adecuadamente el rol de usuario no se pueden acceder a funciones que tiene permitidas los roles especificados.
 - **Administrador:**
 - Agregar productos

- Buscar productos
- Editar producto
- Eliminar Producto
- Listar Productos
- Crear nuevo usuario
- **Gestor de Inventario:**
 - Agregar productos
 - Editar producto
 - Eliminar producto
 - Buscar productos
 - Listar Productos
- **Usuario regular:**
 - Buscar productos
 - Listar Productos

Iniciar Sesión

Descripción: Este caso de prueba evalúa la funcionalidad de iniciar sesión en el sistema de gestión de inventarios tanto con credenciales válidas como con credenciales incorrectas y verifica que el acceso se maneje correctamente en ambos casos.

Pasos:

1. Iniciar el sistema de gestión de inventarios.
2. Intentar iniciar sesión con un usuario válido utilizando las siguientes credenciales:
3. Nombre de usuario: "usuario1"
4. Contraseña: "contraseña1"
5. Intentar iniciar sesión con un usuario o contraseña incorrectos utilizando credenciales incorrectas.

Resultados Esperados:

- Si las credenciales son correctas, el sistema debe mostrar un mensaje que indica que la sesión fue iniciada con éxito y permitir el acceso al sistema.
- Si las credenciales son incorrectas, el sistema debe mostrar un mensaje que indique que el usuario o contraseña son incorrectos y rechazar el acceso al sistema.

Condiciones de Éxito:

- Las credenciales válidas permiten iniciar sesión con éxito.
- Se muestra el mensaje que la sesión fue iniciada con éxito.
- Se puede acceder al sistema después del inicio de sesión.
- Las credenciales incorrectas no permiten el inicio de sesión.
- Se muestra el mensaje de inicio de sesión fallido.
- El acceso al sistema se rechaza después de intentar el inicio de sesión con credenciales incorrectas.

Cerrar Sesión:

Descripción: Este caso de prueba evalúa la funcionalidad de cerrar sesión en el sistema de gestión de inventarios y verificar que el usuario ya no esté logueado en el sistema después de cerrar la sesión.

Pasos:

5. Iniciar el sistema de gestión de inventarios.
6. Iniciar sesión con un usuario válido utilizando las siguientes credenciales:
7. Nombre de usuario: "usuario1"
8. Contraseña: "contraseña1"
9. Cierre la sesión del usuario logueado.

Resultados Esperados:

- Después de iniciar sesión con éxito, el sistema debe mostrar un mensaje que indica que la sesión fue iniciada con éxito.
- Después de cerrar la sesión, el sistema debe mostrar un mensaje "inicio de sesión exitoso" o un mensaje similar.

- Después de cerrar la sesión, el usuario ya no debe estar logueado en el sistema y no debe poder acceder a las funciones que requieren una sesión iniciada.

Condiciones de Éxito:

- El usuario puede iniciar sesión con éxito.
- El usuario puede cerrar la sesión con éxito.
- Después de cerrar la sesión, el usuario ya no está logueado en el sistema y no puede acceder a funciones que requieren una sesión iniciada.

Condiciones de Fallo:

- No se puede iniciar sesión con las credenciales válidas.
- No se puede cerrar la sesión con éxito.
- Después de cerrar la sesión, el usuario todavía está logueado en el sistema y puede acceder a funciones que requieren una sesión iniciada.

5. Operación y mantenimiento

Una vez que se hayan llevado a cabo las suficientes pruebas el sistema se entrega al usuario final lo que implica la instalación y configuración del software en el equipo del cliente, dado que el usuario final va a estar haciendo uso del software probándolo en su día a día va a ser necesario establecer mecanismos de comunicación y soporte para poder atender consultas, problemas relacionados con el software, preguntas del usuario y problemas de rendimiento, lo que a su vez da a lugar la corrección de errores que no se detectaron durante las etapas anteriores y la resolución de problemas que surgen durante la operación.

Va a ser común que se tengan que realizar actualizaciones y mejoras en el software para mejorar su rendimiento, interfaz y funcionalidad por lo que va a ser fundamental escuchar las necesidades del cliente adoptando un proceso de gestión de cambios para evaluar los cambios en el software de esta manera se mantiene un control en los cambios a realizar y evitar la introducción de nuevos errores que desencadenen una falla catastrófica en el sistema además cualquier ajuste en el software debe ser documentado para tener una comprensión más profunda del software.

Prototipo de la interfaz

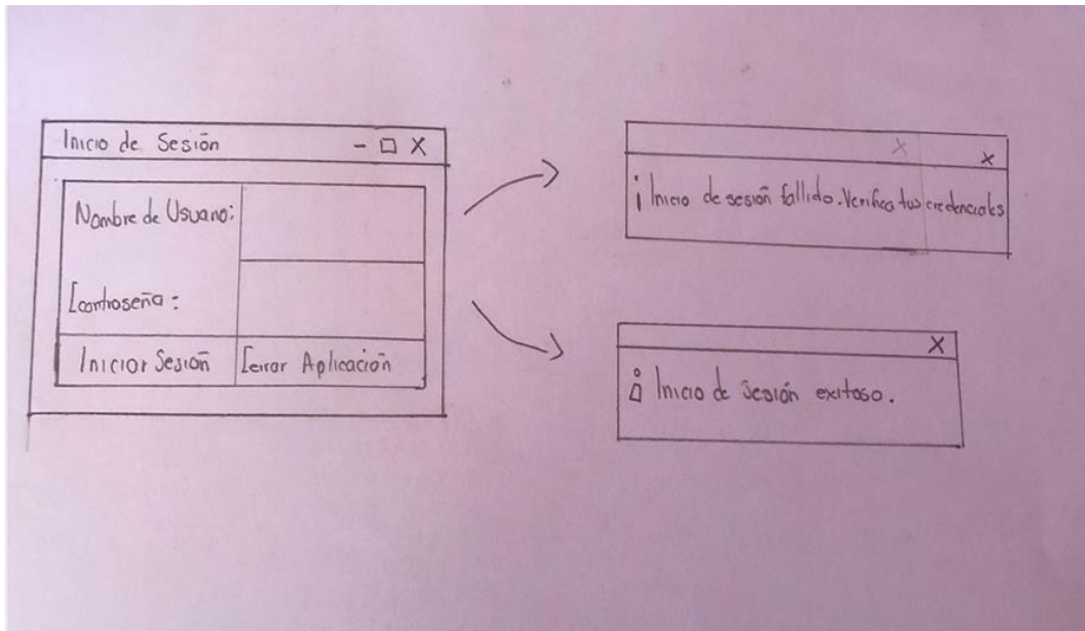


Figura 9. Prototipo Interfaz

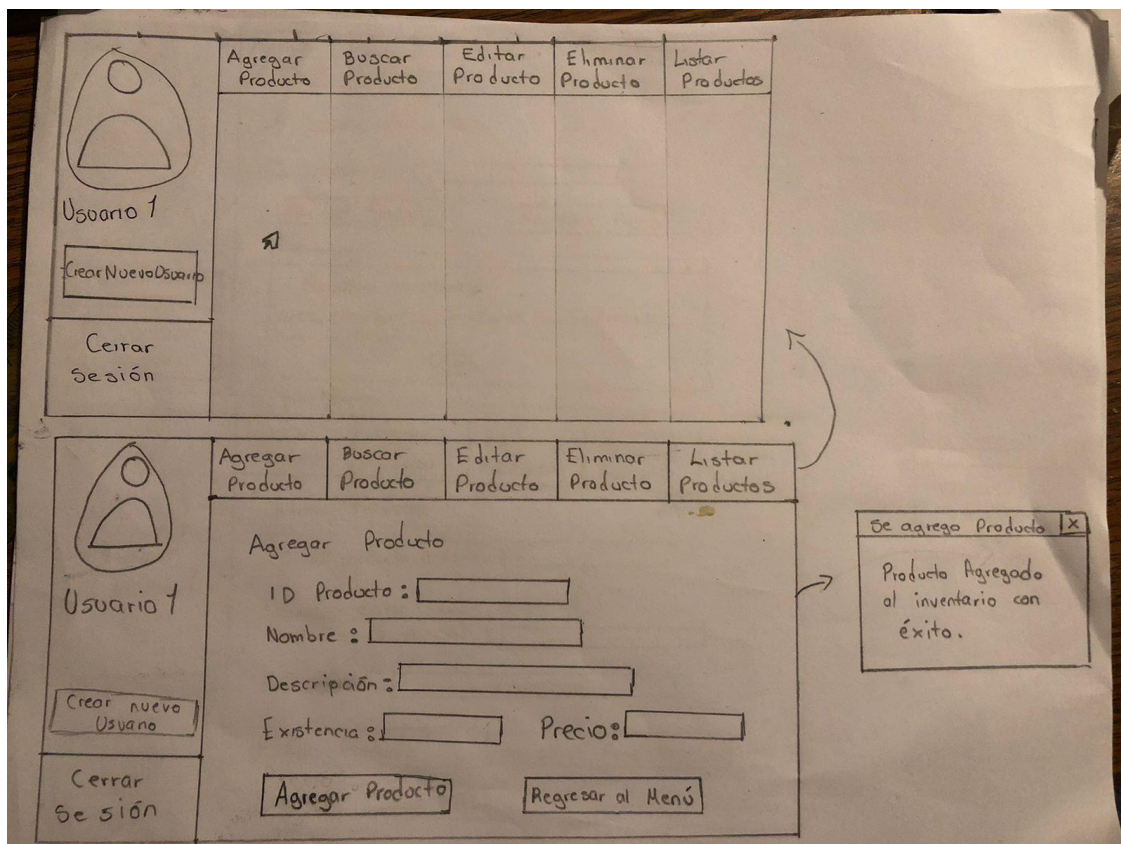


Figura 10. Prototipo Interfaz

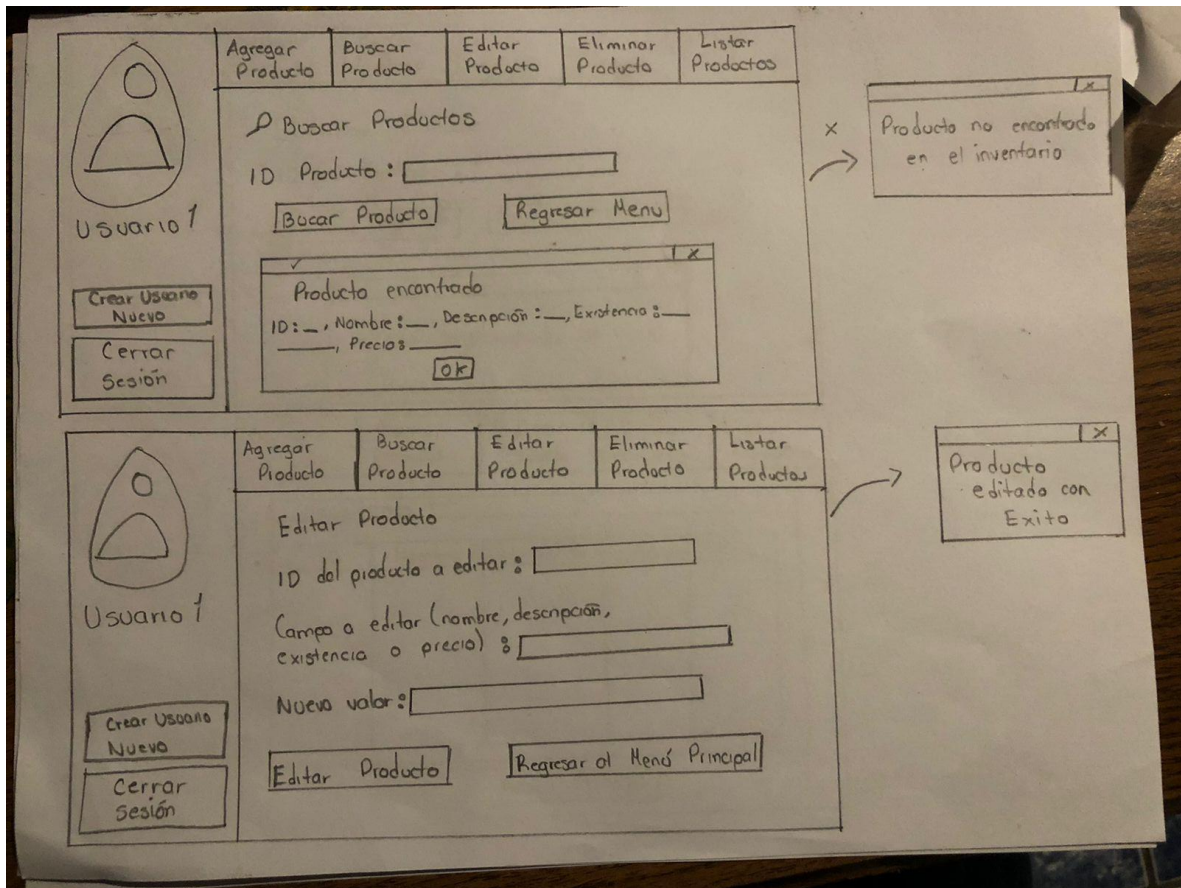


Figura 11. Prototipo Interfaz

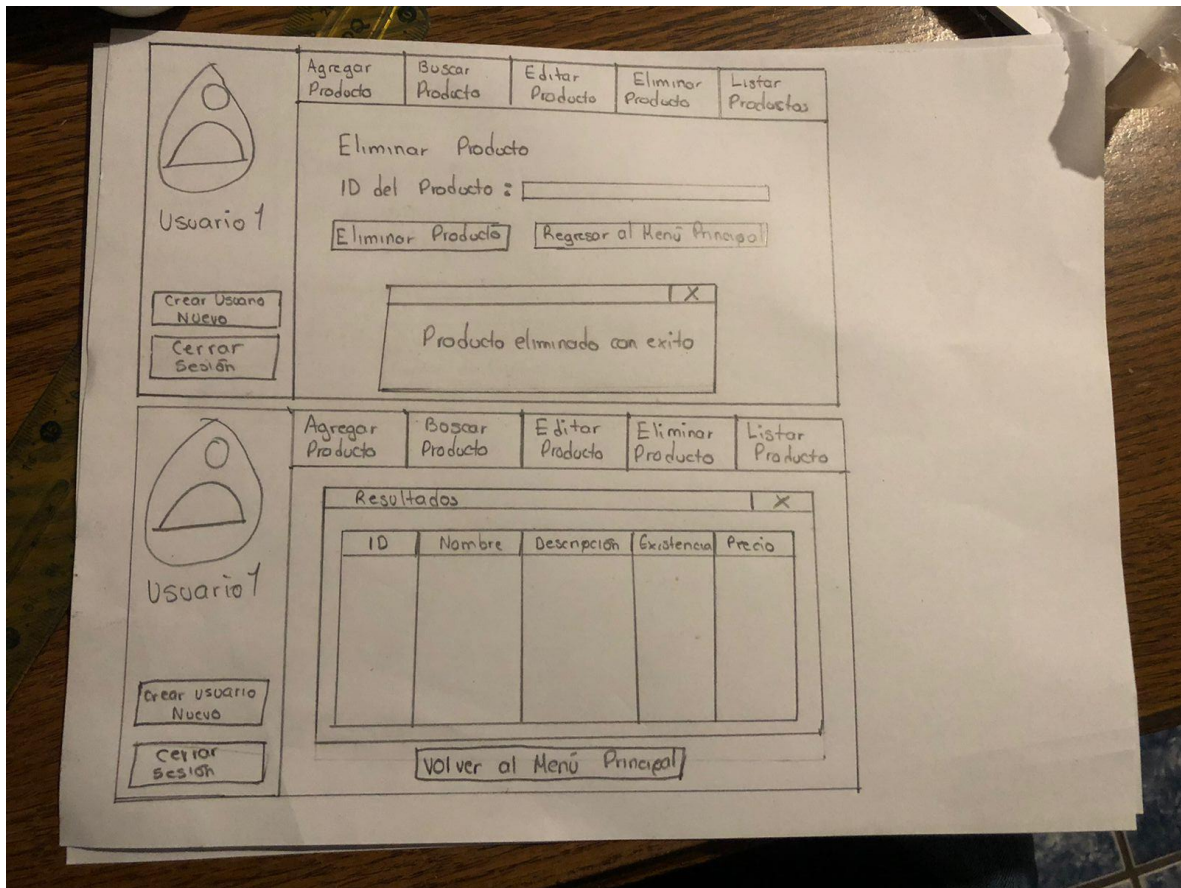


Figura 12. Prototipo Interfaz


| | Agregar Producto | Buscar Producto | Editar Producto | Eliminar Producto | Listar Producto | | | | | | |
|--|--|--------------------|--------------------|----------------------|--------------------|---------------|---|----------------------|--|-----------------|--|
|  Usuario 1 | Crear nuevo usuario ID de Usuario: <input type="text"/> Nombre de Usuario: <input type="text"/> Contraseña: <input type="password"/> Rol : <table border="1"> <tr> <td>Administrador</td> <td>▼</td> </tr> <tr> <td>Gestor de Inventario</td> <td></td> </tr> <tr> <td>Usuario Regular</td> <td></td> </tr> </table> | | | | | Administrador | ▼ | Gestor de Inventario | | Usuario Regular | |
| Administrador | ▼ | | | | | | | | | | |
| Gestor de Inventario | | | | | | | | | | | |
| Usuario Regular | | | | | | | | | | | |
| Crear Nuevo Usuario Cerrar Sesión | Crear Usuario Regresar al Menú Principal | | | | | | | | | | |

Figura 13. Prototipo Interfaz

Desarrollo

Para llevar a cabo el desarrollo de la interfaz el código es dividido en diferentes clases para poder realizar un manejo adecuado de las funciones, captura de datos, validaciones y componentes de la interfaz.

Ventana de Inicio de Sesión



Figura 14. Desarrollo: Ventana - Inicio Sesión e Implementación

Descripción:

Es la primera pantalla que se desplegará cuando se ejecute la aplicación, se crean dos campos de texto para realizar la captura de datos el nombre de usuario y la contraseña.

Cuando se hace clic en el botón “Iniciar Sesión” se verifica el nombre de usuario y la contraseña ingresados si la autenticación es exitosa, se muestra un mensaje de inicio de sesión exitoso y se cierra la ventana de inicio de sesión

pasando con otra ventana para continuar con la aplicación principal. Si la autenticación falla se muestra un mensaje de error.

Cuando se hace clic en el botón “Cerrar Aplicación” se finaliza la ejecución del programa cerrando la pantalla de inicio de sesión.

Pantalla principal de la aplicación

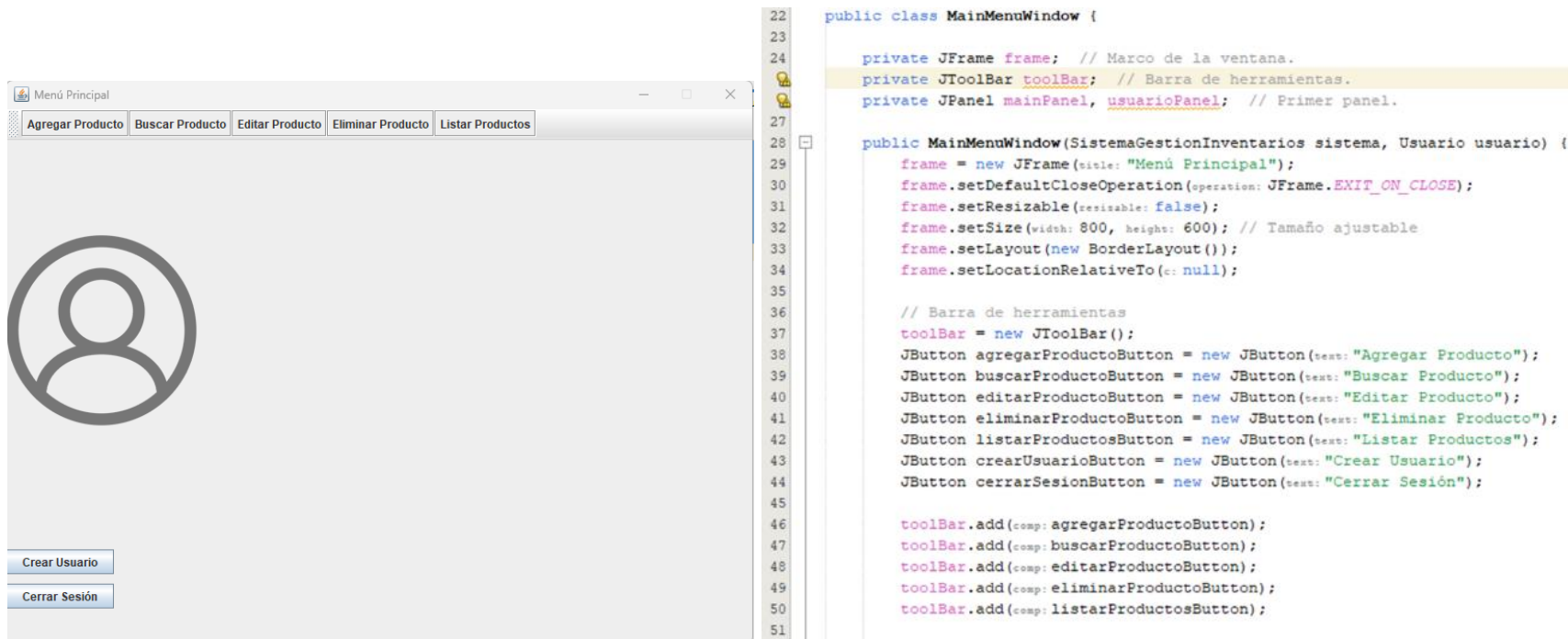


Figura 15. Pantalla principal e Implementacion

Descripción:

Debajo de la imagen que representa a un usuario anonimo se encuentran dos botones el botón “Crear Usuario” la cual va a realizar un cambio en la pantalla generando un nuevo panel para añadir nuevos usuarios y el botón “Cerrar sesión” que cierra la sesión actual y vuelve a la ventana de inicio de sesión.

En la barra se encuentren los botones para realizar las principales acciones en la aplicación, como agregar, buscar, editar y eliminar productos, listar productos, crear usuarios y cerrar sesión. Cada botón tiene una acción que se ejecutará cuando

se haga clic en él. Dependiendo del botón presionado se realizará una acción específica la cual es cambiar la pantalla para agregar, buscar, editar o eliminar un producto, listar productos en el inventario. Cada botón desplegará un panel diferente para las operaciones del inventario.

Panel de Agregar Producto en la ventana principal

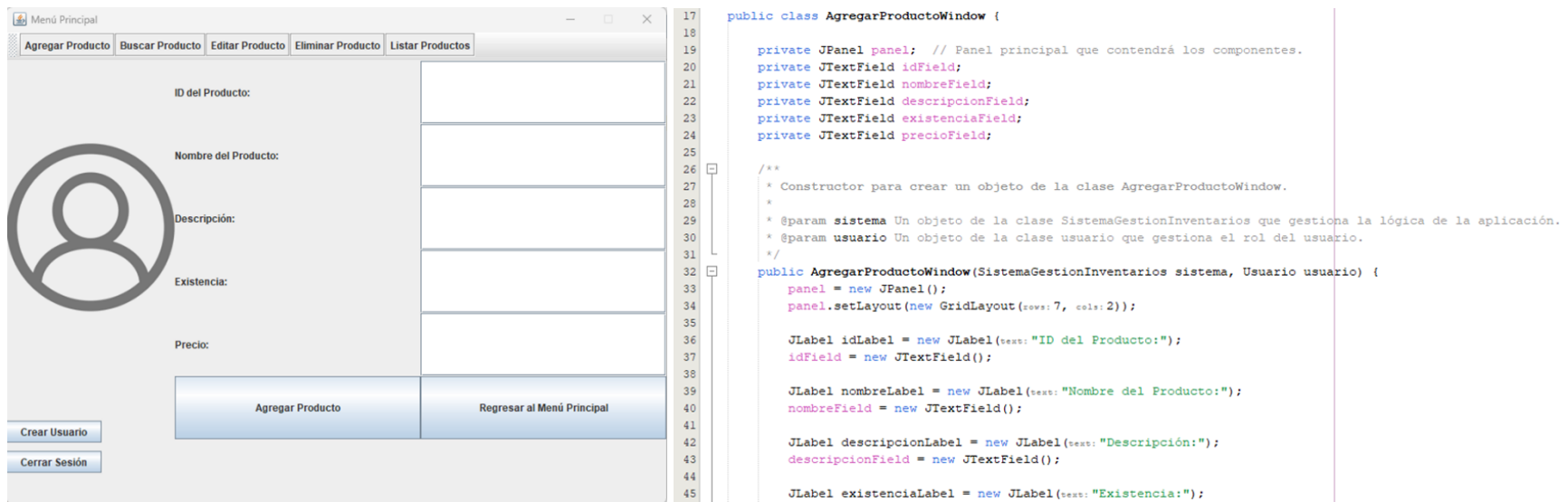


Figura 16. Panel Agregar Producto e Implementación

Descripción:

Cuando se hace clic en el botón “Agregar Producto” en la barra de botones se crea un nuevo un panel con campos de texto para capturar los datos necesarios, para después agregar un nuevo producto al inventario de la aplicación.

Cuando el usuario hace clic en el botón de agregar producto se capturan los datos ingresados en los campos de texto (ID, nombre, descripción, existencia y precio), se validan y se crea un nuevo objeto Producto con estos datos. Si los datos no son válidos se muestra un mensaje de error en una ventana emergente. Luego de agregar el producto, se limpian los campos de texto para que el usuario pueda ingresar otro producto.

Panel de Buscar Producto en la ventana principal

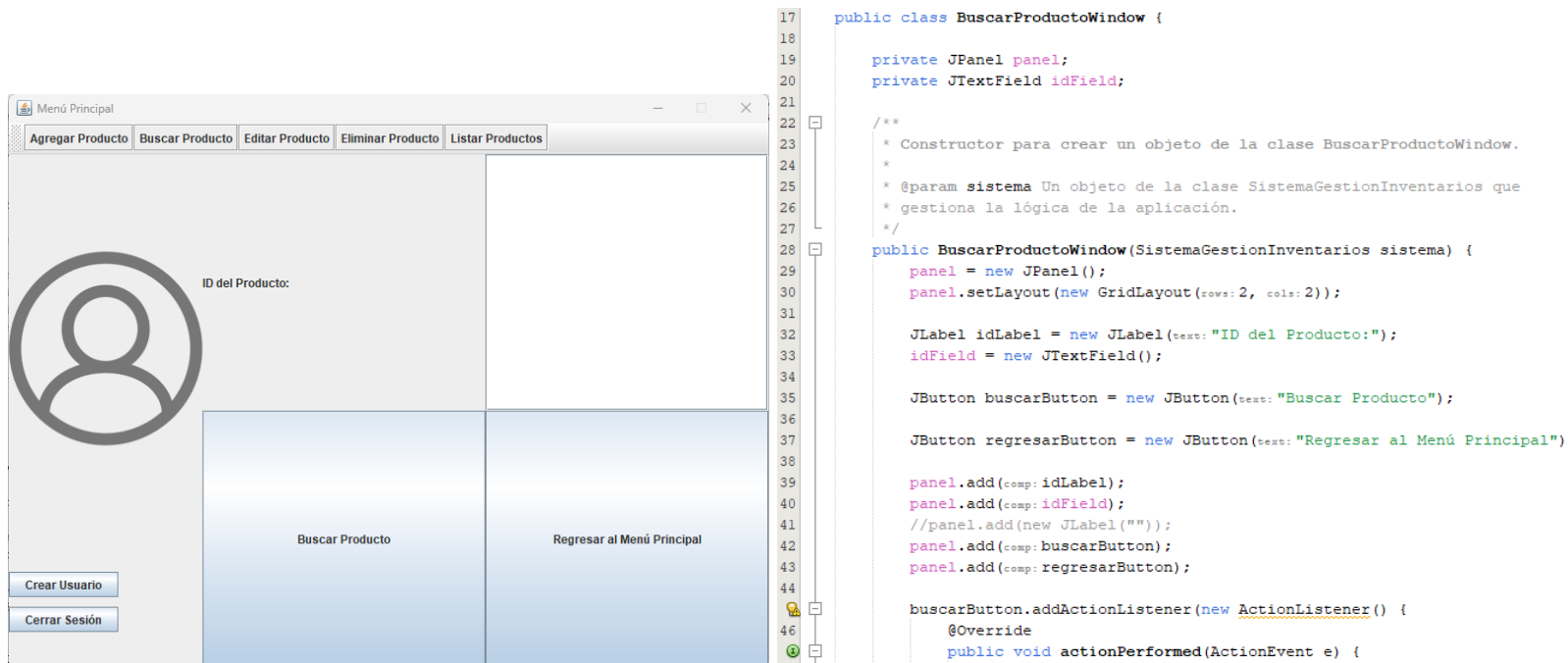


Figura 17. Pantalla Agregar Producto e Implementacion

Descripción:

Cuando se hace clic en el botón “Buscar Producto” en la barra de botones se crea un nuevo un panel con un campo de texto para capturar el dato necesario para realizar una consulta.

Cuando el usuario hace clic en el botón buscar producto, se captura el ID ingresado en el campo de texto, se valida y se utiliza para buscar un producto en el inventario.

- Si se encuentra un producto con el ID especificado, se muestra una ventana emergente que presenta los detalles del producto encontrado. Si no se encuentra el producto, se muestra un mensaje informando que el producto no está en el inventario.

Panel de Eliminar Producto en la ventana principal

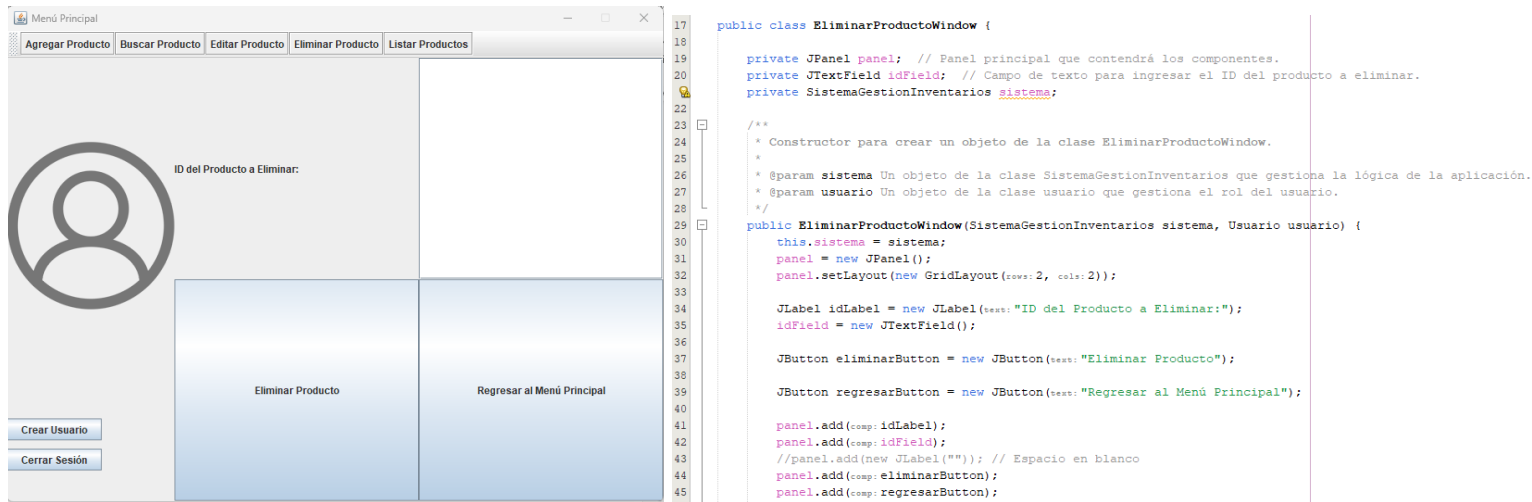


Figura 18. Panel Eliminar Producto e Implementación

Descripción:

Cuando se hace clic en el botón “Eliminar Producto” en la barra de botones se crea un nuevo un panel con un campo de texto para capturar el ID del producto para realizar la eliminación del producto correspondiente.

Cuando el usuario hace clic en el botón eliminar producto, se captura el ID ingresado en el campo de texto, se valida y se utiliza para buscar un producto en el inventario.

Si se encuentra un producto con el ID especificado, se realiza la eliminación del producto en el inventario si no se encuentra el producto se muestra un mensaje indicando que el producto no está en el inventario.

Si el producto no se encuentra o si el producto es eliminado se borra el campo de texto para que el usuario ingrese otro ID.

Panel de Editar Producto en la ventana principal

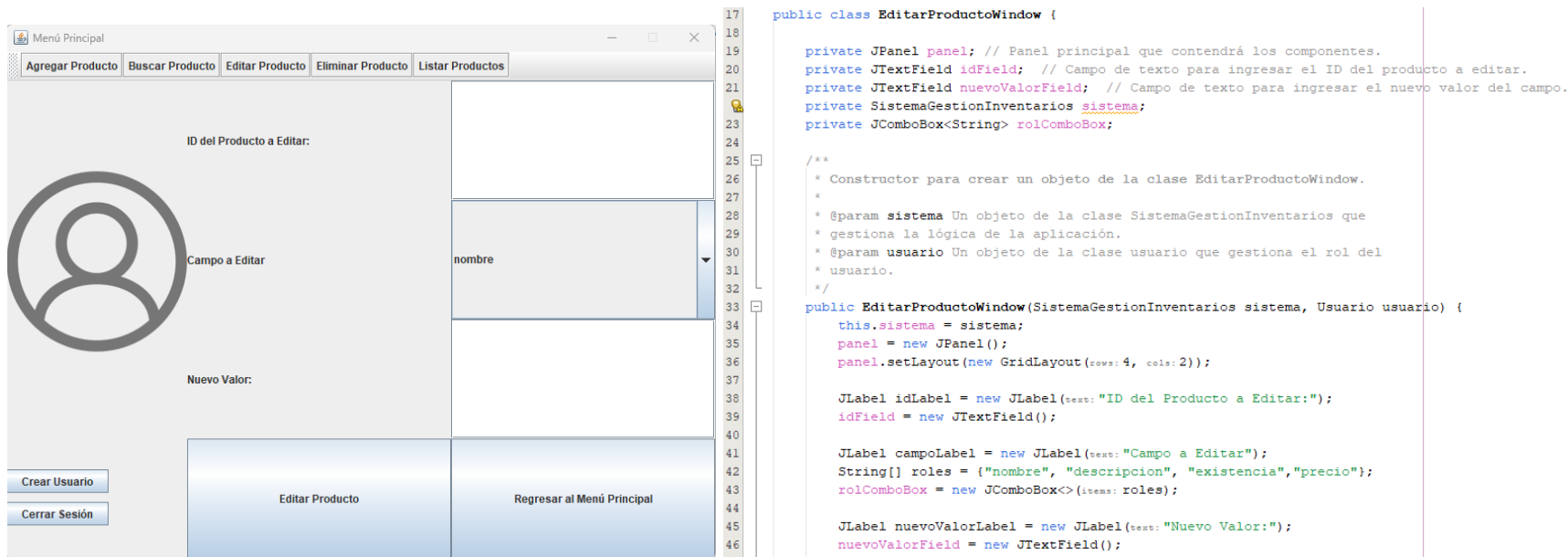


Figura 19. Panel Editar Producto e Implementación

Descripción:

Cuando se hace clic en el botón “Editar Producto” en la barra de botones se crea un nuevo un panel con dos campos de texto para capturar el ID del producto, el nuevo valor y un campo de selección para elegir un campo a editar como el nombre, precio, existencia y descripción para realizar modificación del producto correspondiente.

Cuando el usuario hace clic en el botón de editar producto se capturan los datos ingresados en los campos de texto (ID, campo a editar y nuevo valor), se validan y se busca el producto en el inventario con el ID especificado.

Si se encuentra el producto se utiliza el método `editarProducto` para editar el campo especificado con el nuevo valor para el producto. Si el producto no se encuentra se muestra un mensaje informando que el producto no está en el inventario.

Panel de Crear Usuario en la ventana principal

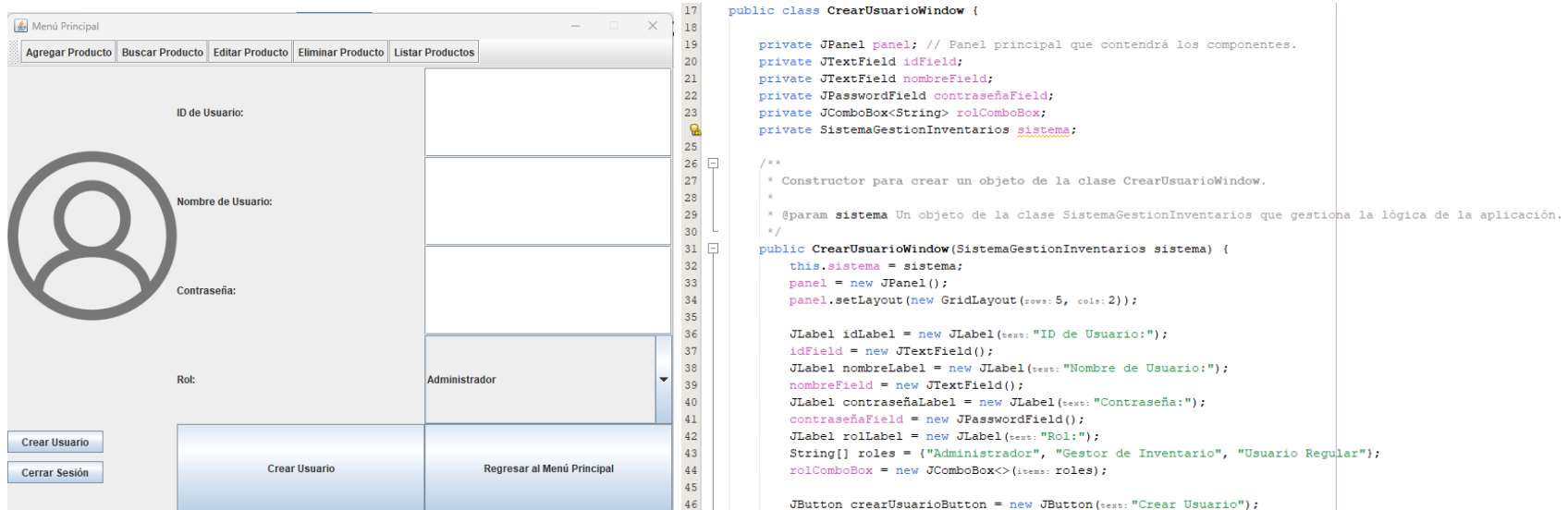


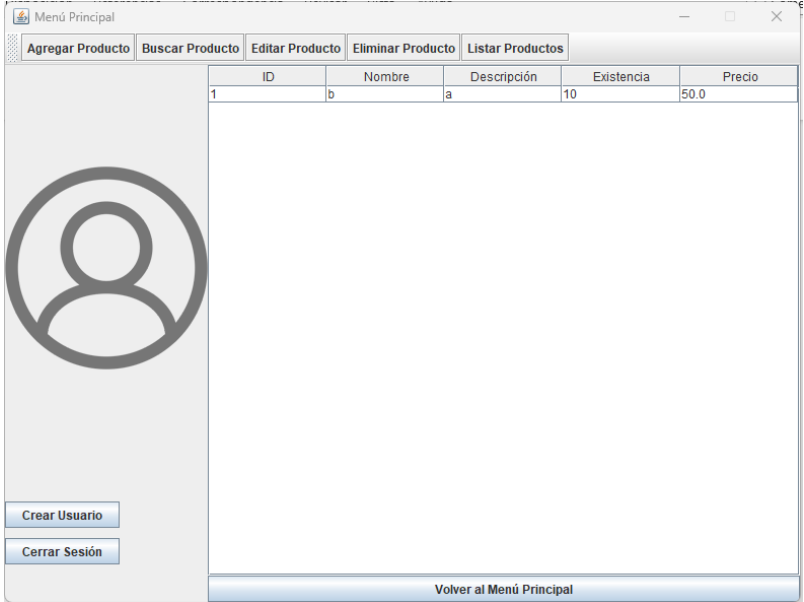
Figura 20. Panel Crear Usuario e Implementación

Descripción:

Cuando se hace clic en el botón “Crear Usuario” en la barra de botones se crea un nuevo un panel con tres campos de texto para capturar el ID del usuario, nombre de usuario, una contraseña y un campo de selección para elegir un rol “Administrador”, “Gestor de Inventario” y “Usuario Regular” para la creación de un nuevo usuario.

Cuando el usuario hace clic en el botón crear usuario se capturan los datos ingresados en los campos de texto y en el campo de selección. Luego, se verifica si el usuario logueado, si el ID ya está registrado y si tiene permisos para crear usuarios, en este caso, se valida que el rol del usuario logueado sea Administrador. Si el usuario logueado cumple con los requisitos, se crea un nuevo usuario y se agrega al sistema se muestra un mensaje de confirmación si el usuario se crea con éxito, si el usuario logueado no cumple con los requisitos, se muestra un mensaje de error.

Panel de Listar Productos en la ventana principal



```

21 public class ResultadosWindow {
22     private JPanel panel; // Panel principal.
23     private SistemaGestionInventarios sistema;
24     private Usuario usuario;
25
26     /**
27      * Constructor para crear un objeto de la clase ResultadosPanel.
28      *
29      * @param productos Una lista de objetos Producto que se mostrarán en la tabla.
30      * @param sistema Un objeto de la clase SistemaGestionInventarios que gestiona la lógica de la aplicación.
31      * @param usuario Un objeto de la clase usuario que gestiona el rol del usuario.
32      */
33     public ResultadosWindow(ArrayList<Producto> productos, SistemaGestionInventarios sistema, Usuario usuario) {
34         this.sistema = sistema;
35         this.usuario = usuario;
36         panel = new JPanel();
37         panel.setLayout(new BorderLayout());
38
39         // Crear un DefaultTableModel con las columnas deseadas
40         DefaultTableModel model = new DefaultTableModel();
41         model.addColumn(columnName: "ID");
42         model.addColumn(columnName: "Nombre");
43         model.addColumn(columnName: "Descripción");
44         model.addColumn(columnName: "Existencia");
45         model.addColumn(columnName: "Precio");
46
47         // Llenar la tabla con los productos
48         for (Producto producto : productos) {
49             model.addRow(new Object[] {
50                 producto.getIdProducto(),
51                 producto.getNombre(),

```

Figura 21. Panel Listar Productos e Implementación

Descripción:

Cuando se hace clic en el botón “Listar Productos” en la barra de botones se crea un nuevo un panel en donde se construye una tabla de resultados. Se definen las columnas de la tabla: "ID", "Nombre", "Descripción", "Existencia" y "Precio".

Luego, se llena la tabla con los productos proporcionados en la lista del inventario. Se recorre la lista y se agrega cada producto como una fila en la tabla. A medida que se agregan más productos en la tabla aparecerá una barra para poder desplazarse a medida por la tabla a medida que crece.

Manual de usuario

Los procedimientos para seguir se describen en base a las siguientes pantallas:

- Pantalla de inicio de sesión
- Pantalla principal de la aplicación

Pantalla de inicio de sesión

Se le van a proporcionar dos cuentas para poder iniciar sesión en el sistema.

- **Cuenta 1:**

Nombre de usuario: usuario1

Contraseña: contraseña1

- **Cuenta 2:**

Nombre de usuario: usuario2

Contraseña: contraseña2

Va a tener como cuenta principal la cuenta número 1, así que asegúrese de mantener en privado esta contraseña. Con esta cuenta usted va a contar con permisos exclusivos ya que con ella va a poder agregar nuevos usuarios para el manejo del inventario en cambio la cuenta 2 no tiene permisos para las acciones que se han descrito.

Para iniciar sesión debe de ingresar los datos de usuario y contraseña en los campos en blanco.

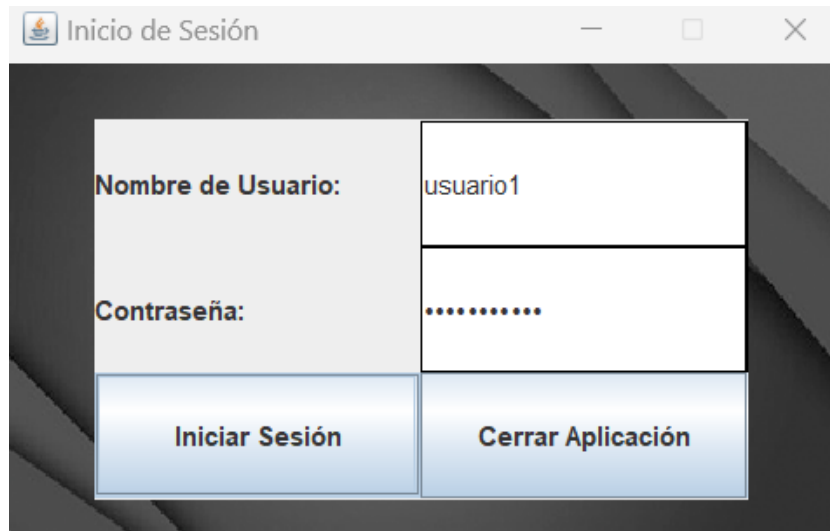


Figura 22. Guia de usuario

Después debe de hacer clic en iniciar sesión para acceder a las funciones de la aplicación

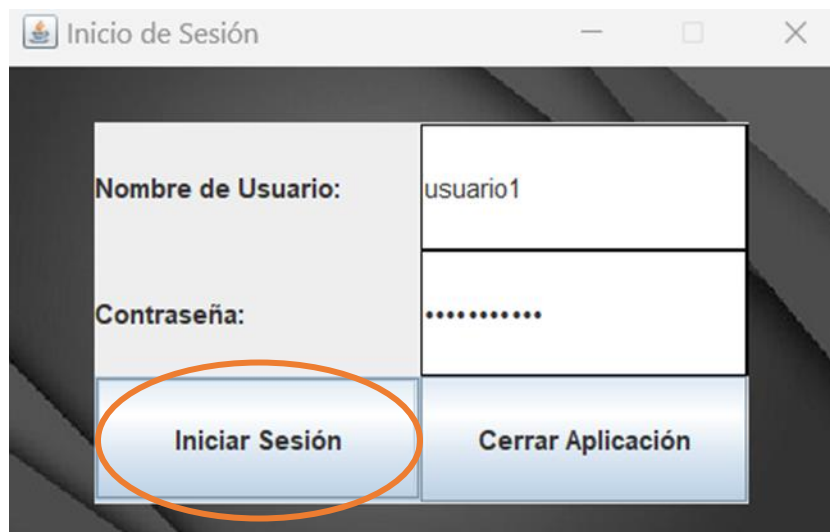


Figura 23. Guia de usuario

Si ingreso correctamente sus datos le aparecerá un mensaje de inicio de sesión exitoso.

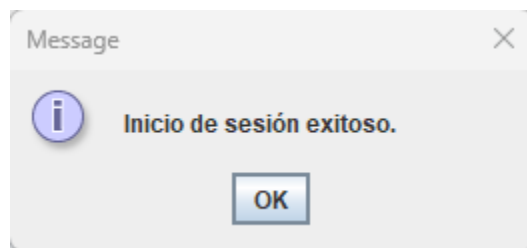




Figura 24. Guía de usuario

Haga clic en el botón  o en el botón  para continuar.

Si no ingresa bien los datos de usuario y contraseña le aparecerá un mensaje de inicio de sesión fallido, esto significa que usted ingreso incorrectamente ya sea la contraseña, el usuario o ambas.

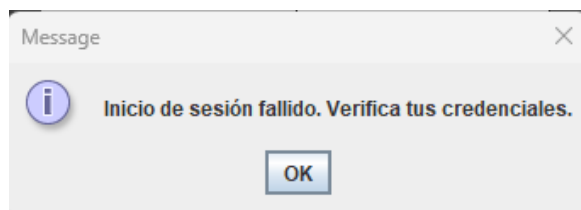




Figura 25. Guía de usuario

Para volver a ingresar sus datos haga clic en el botón  o en el botón . Y repita la acción nuevamente.

En caso de perder ambas contraseñas, tener problemas con iniciar sesión en la aplicación u otra situación en donde la aplicación tenga problemas, por favor ponerse en contacto al siguiente correo/teléfono para llevar a cabo acciones requeridas.

Pantalla principal de la aplicación

Después de iniciar sesión de manera exitosa se le mostrara la siguiente pantalla.

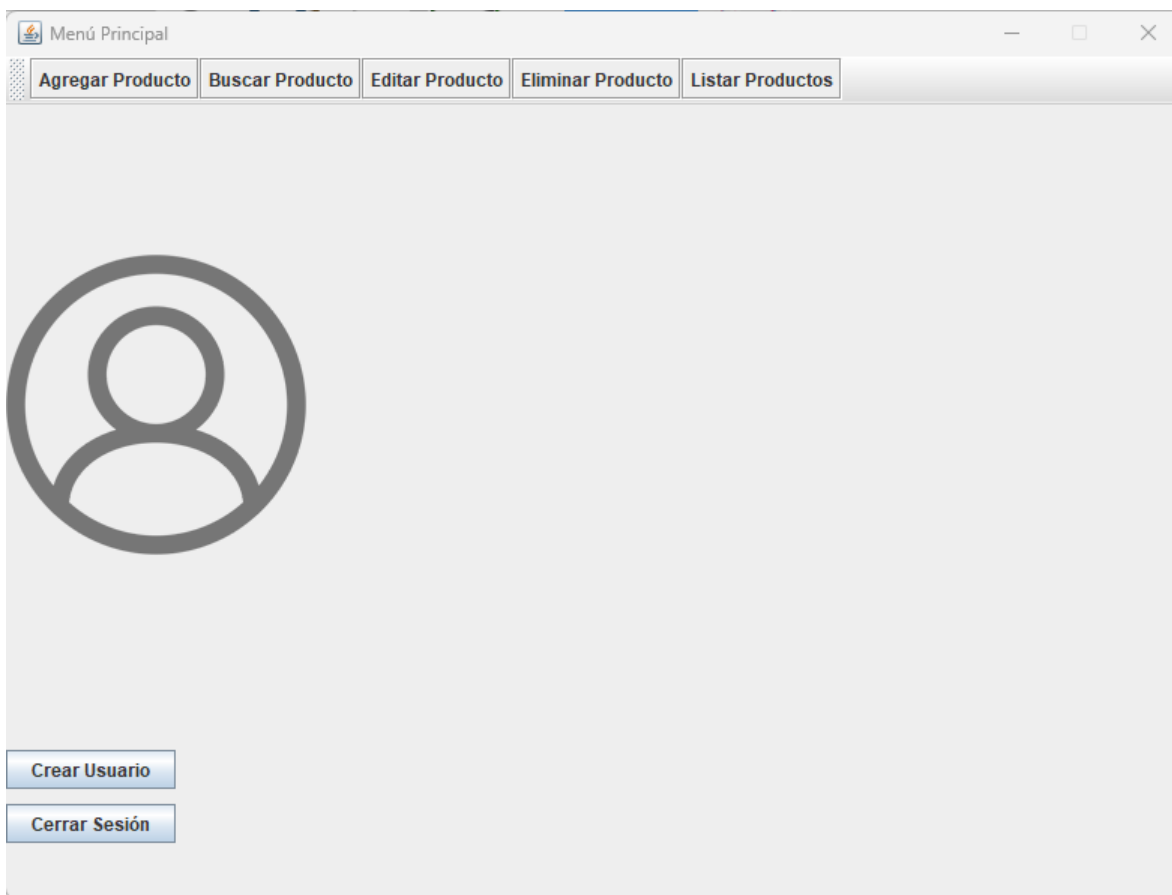
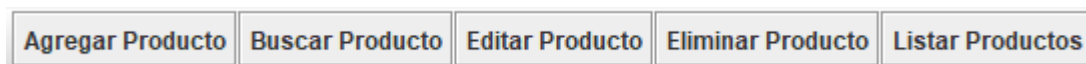


Figura 26. Guía de usuario

Hay una barra de opciones en la parte superior esta es la barra en la que debe poner atención



estas son las opciones que usted puede realizar en el sistema, para poder acceder a ellas tiene que hacer clic en una de las opciones.

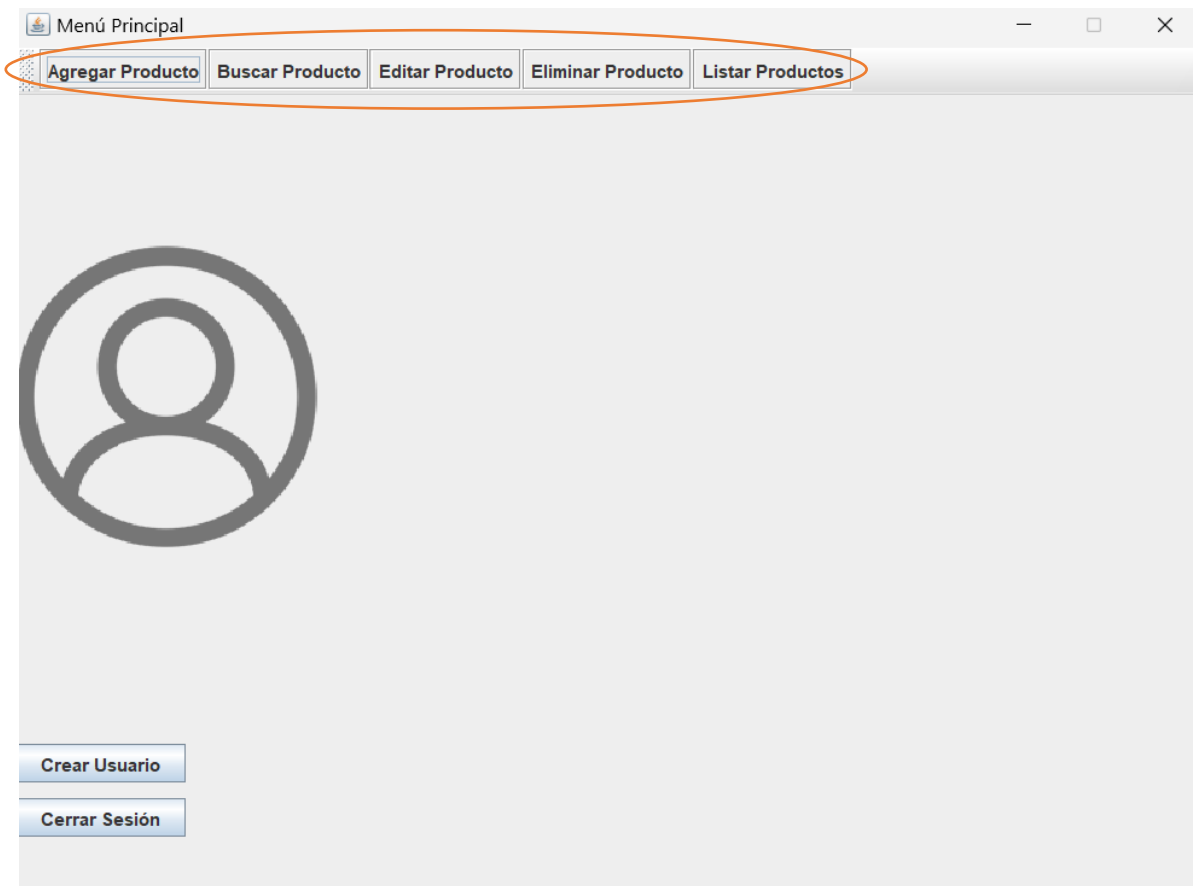


Figura 27. Guía de usuario

A continuación, se muestra cómo debe navegar y que acciones debe realizar por la aplicación.

- Agregar producto

Cuando usted haga clic en el botón de “Agregar Producto” le aparecerá la siguiente pantalla.

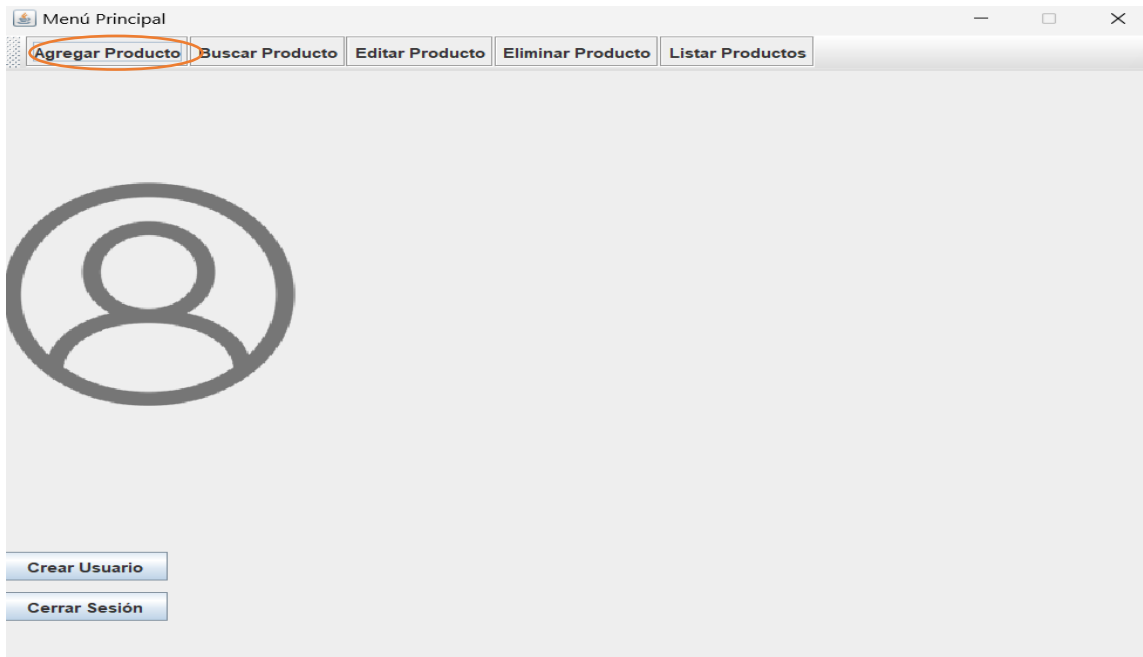


Figura 28. Guía de usuario

Una vez que hizo clic en el botón le aparecerá la siguiente pantalla:

The screenshot shows a web application window titled "Menú Principal". At the top, there is a navigation bar with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The main content area is divided into two columns. The left column contains a large circular icon with a person silhouette and five labels: "ID del Producto:", "Nombre del Producto:", "Descripción:", "Existencia:", and "Precio:". The right column contains five corresponding text input fields. At the bottom of the form, there are two large blue buttons: "Agregar Producto" and "Regresar al Menú Principal". On the far left, there are two smaller buttons: "Crear Usuario" and "Cerrar Sesión".

Figura 29. Guía de usuario

En ella debe hacer clic en los bloques en blanco y escribir los datos necesarios del producto que va a registrar.

- ID del producto: Este va a ser un identificador del producto, debe ingresar un numero entero.
- Nombre del producto: El nombre del producto.
- Descripción del producto: Información extra del producto que va a ingresar.
- Existencia: Cantidad de productos que posee, debe ingresar un numero entero.
- Precio: Precio del producto en el inventario, debe ingresar ya sea un numero entero o un numero con decimales.

The screenshot shows a web application window titled "Menú Principal". At the top, there is a horizontal menu with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The "Agregar Producto" button is highlighted. On the left side, there is a large circular icon representing a user profile. Below this icon are two buttons: "Crear Usuario" and "Cerrar Sesión". The main area of the window contains a form for adding a product. The form has five labeled input fields: "ID del Producto:", "Nombre del Producto:", "Descripción:", "Existencia:", and "Precio:". To the right of these fields is a large orange circle that encompasses the input area. At the bottom of the form, there are two buttons: "Agregar Producto" and "Regresar al Menú Principal".

Figura 30. Guía de usuario

A continuación, se le muestra un ejemplo de cómo ingresar un producto al inventario asegúrese de ingresar el ID correctamente ya que este valor no es posible modificarlo y es muy importante para realizar acciones como buscar un producto, eliminar producto y realizar modificaciones en los datos de los productos. A continuación, se muestra un ejemplo de cómo agregar un producto al inventario.

The screenshot shows a web application window titled 'Menú Principal'. At the top, there is a navigation bar with five buttons: 'Agregar Producto', 'Buscar Producto', 'Editar Producto', 'Eliminar Producto', and 'Listar Productos'. The main area contains a form for adding a product. On the left side of the form is a large, faint watermark of a person's head and shoulders. The form fields are as follows:

| Label | Value |
|----------------------|-----------------------------|
| ID del Producto: | 2271834 |
| Nombre del Producto: | Jabón liquido |
| Descripción: | Producto de limpieza (1 LT) |
| Existencia: | 50 |
| Precio: | 30 |

At the bottom of the form, there are two buttons: 'Agregar Producto' and 'Regresar al Menú Principal'. To the left of these buttons, there are two smaller buttons: 'Crear Usuario' and 'Cerrar Sesión'.

Figura 31. Guía de usuario

Una vez que haya ingresado todos los datos necesarios haga clic en el botón. Si ingreso correctamente los datos el sistema le mandara un mensaje indicándole que el producto se registró exitosamente.

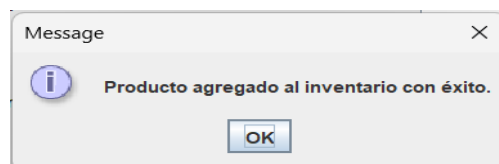


Figura 32. Guía de usuario

El sistema no le dejara ingresar el producto si no ingreso una cantidad de existencias fija, el precio del producto y el identificador del producto (ID). Sino hace esto el sistema le indicara que debe ingresar datos válidos y posteriormente eliminara todos los campos que usted haya llenado.

Figura 33. Guía de usuario

Esos tres datos son importantes para ingresar productos, campos como el nombre y descripción puede dejarlos en blanco.

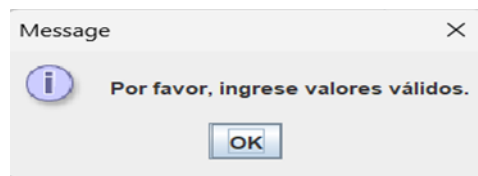


Figura 34. Guía de usuario

En caso de que haya ingresado mal un dato como el nombre, descripción, precio y existencias usted va a poder modificar estos datos para manejar su inventario.

- **Buscar producto**

Cuando usted haga clic en el botón de “Buscar Producto” le aparecerá la siguiente pantalla.

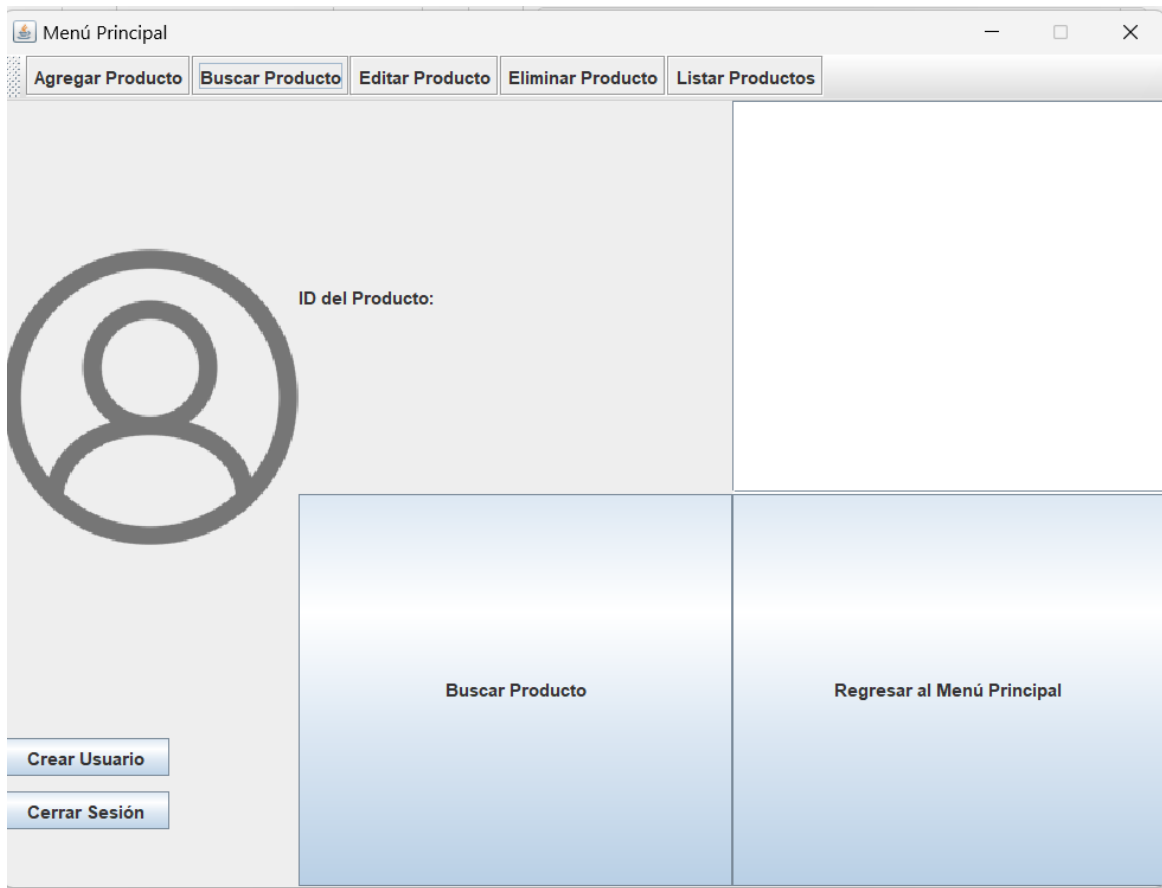


Figura 35. Guía de usuario

Usted deberá ingresar el ID del producto para obtener la información asociada como su ID, nombre, descripción, existencia y precio. Una vez que haya ingresado el id haga clic en el botón “Buscar producto” para realizar la búsqueda. A continuación, se muestra un ejemplo:

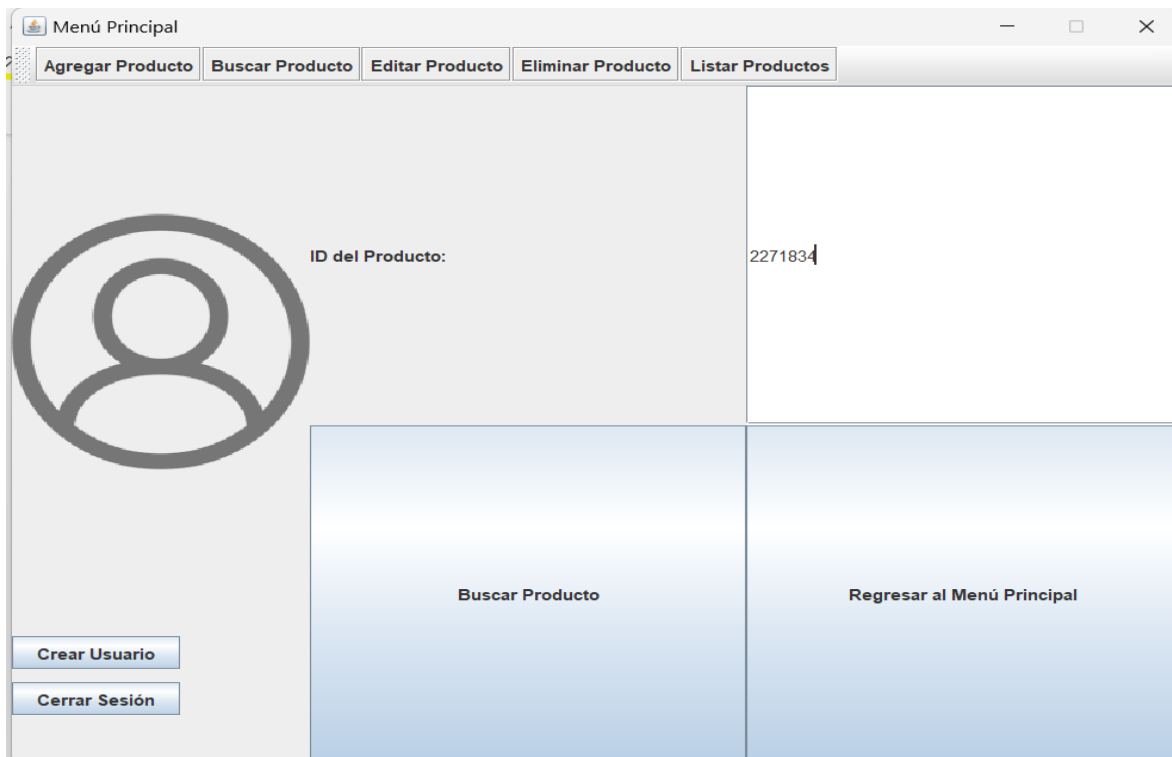


Figura 36. Guía de usuario

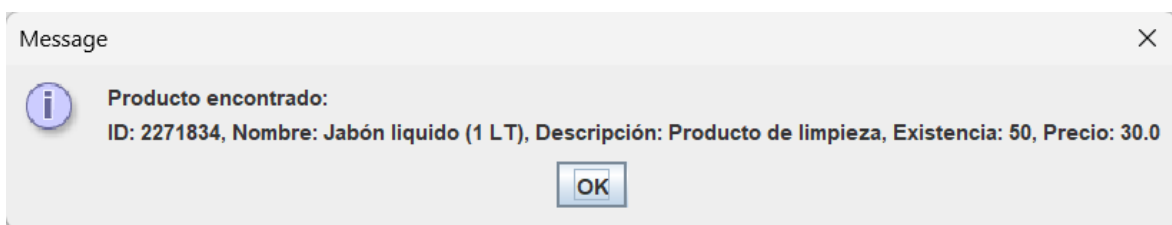


Figura 37. Guía de usuario

Si el producto no está registrado o ingreso mal el ID le aparecerá un mensaje indicándole que el producto no está registrado.

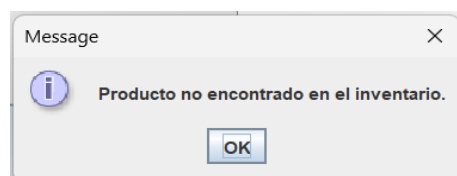


Figura 38. Guía de usuario

Si usted ingresa un dato no valido como letras le aparecerá el siguiente mensaje.

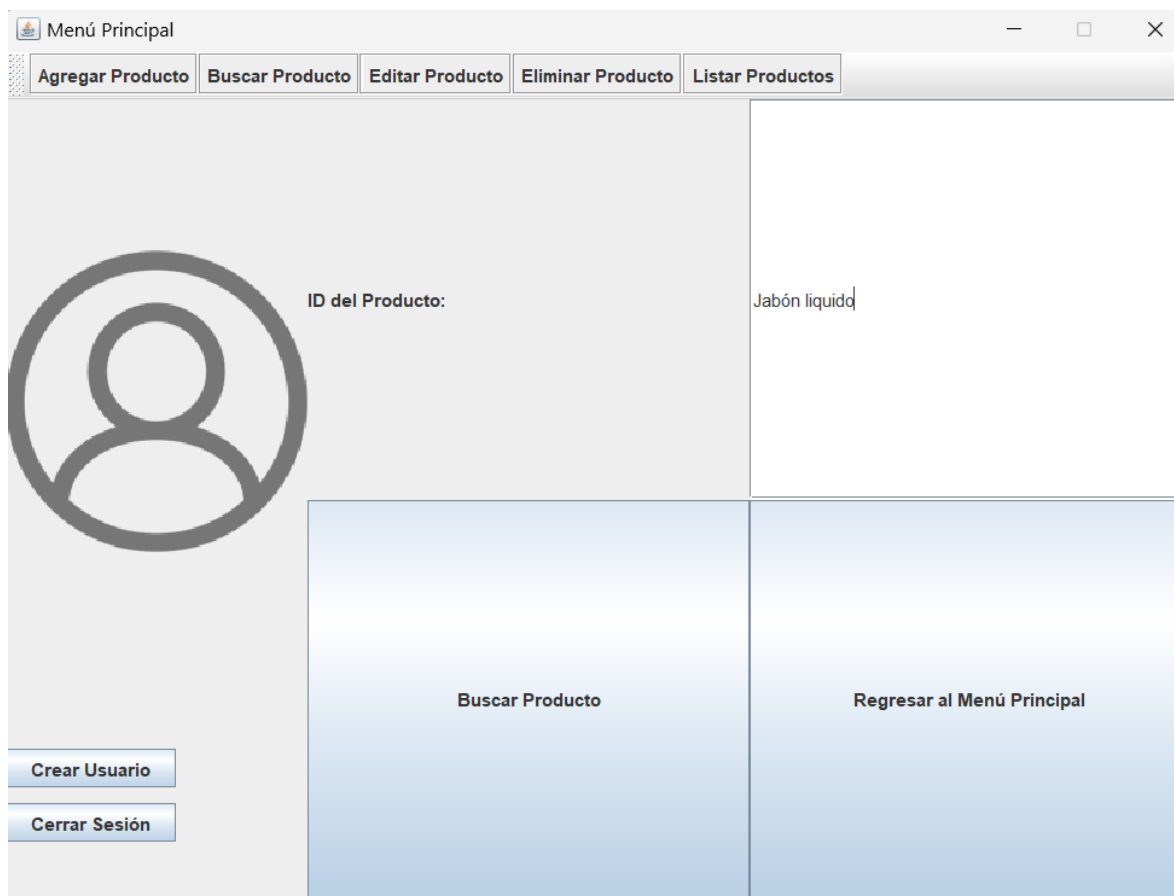


Figura 39. Guía de usuario

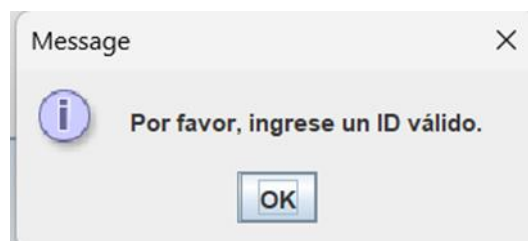


Figura 40. Guía de usuario

- Eliminar producto en el inventario

Cuando usted haga clic en el botón de “Eliminar Producto en el Inventario” le aparecerá la siguiente pantalla.

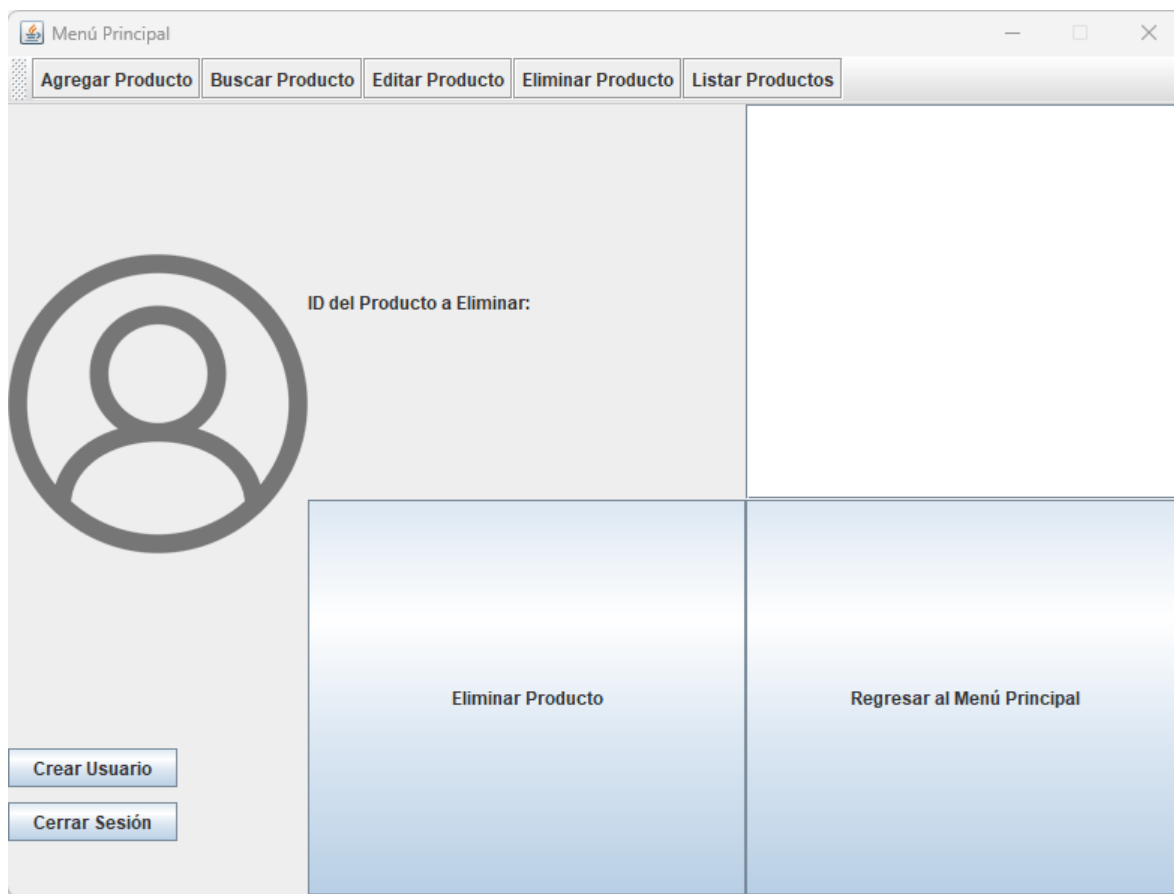


Figura 41. Guía de usuario

Usted deberá ingresar el ID del producto para eliminarlo del inventario a continuación se le muestra un ejemplo.

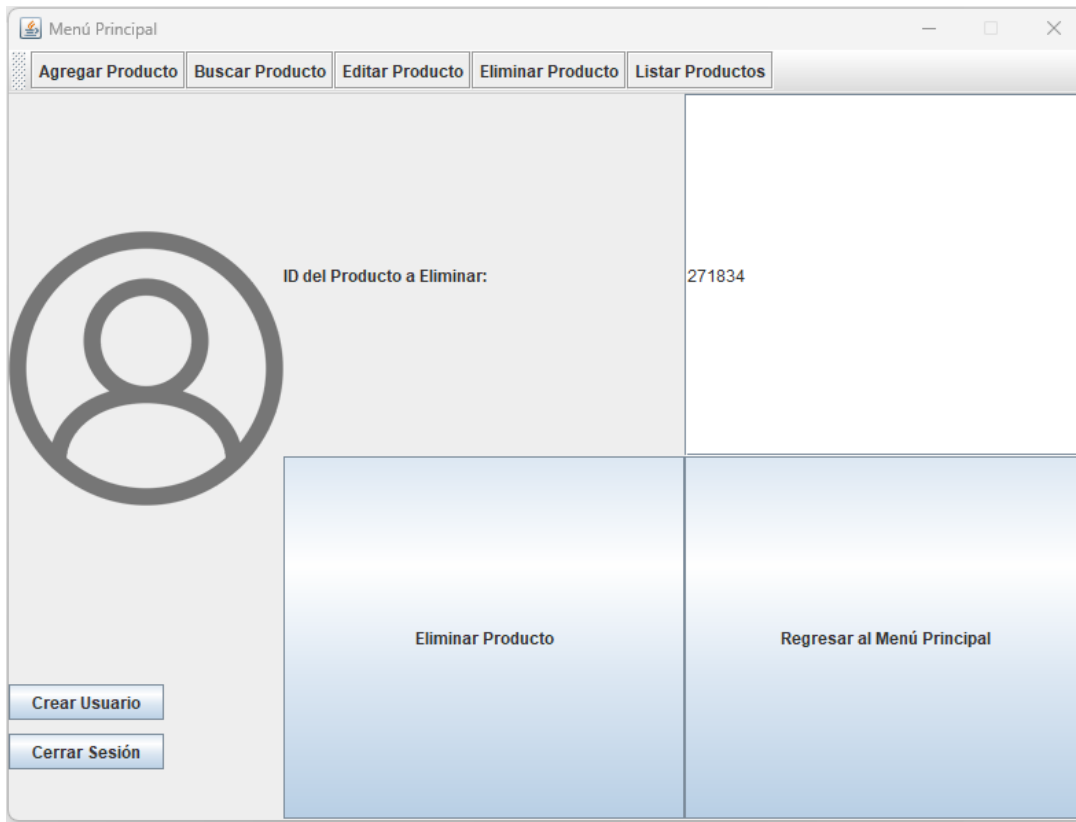


Figura 42. Guía de usuario

Una vez que tenga el ID del producto a eliminar haga clic en el botón “Eliminar Producto” ubicado debajo de los campos, esta acción removerá el producto del inventario y le aparecerá el mensaje confirmando que la acción fue realizada.

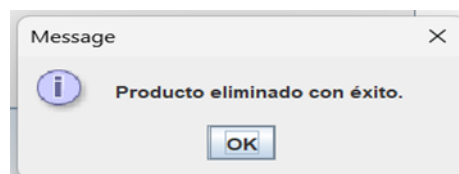


Figura 43. Guía de usuario

Tenga en cuenta que toda la información del producto se perderá y no hay mecanismos que permitan recuperar la información.

Si el producto no está registrado o ingreso mal el ID le aparecerá un mensaje indicándole que el producto no está registrado.

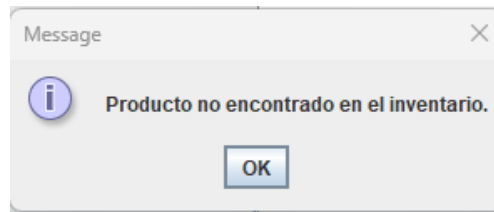


Figura 44. Guia de usuario

Si usted ingresa un dato no valido como letras le aparecerá el siguiente mensaje.

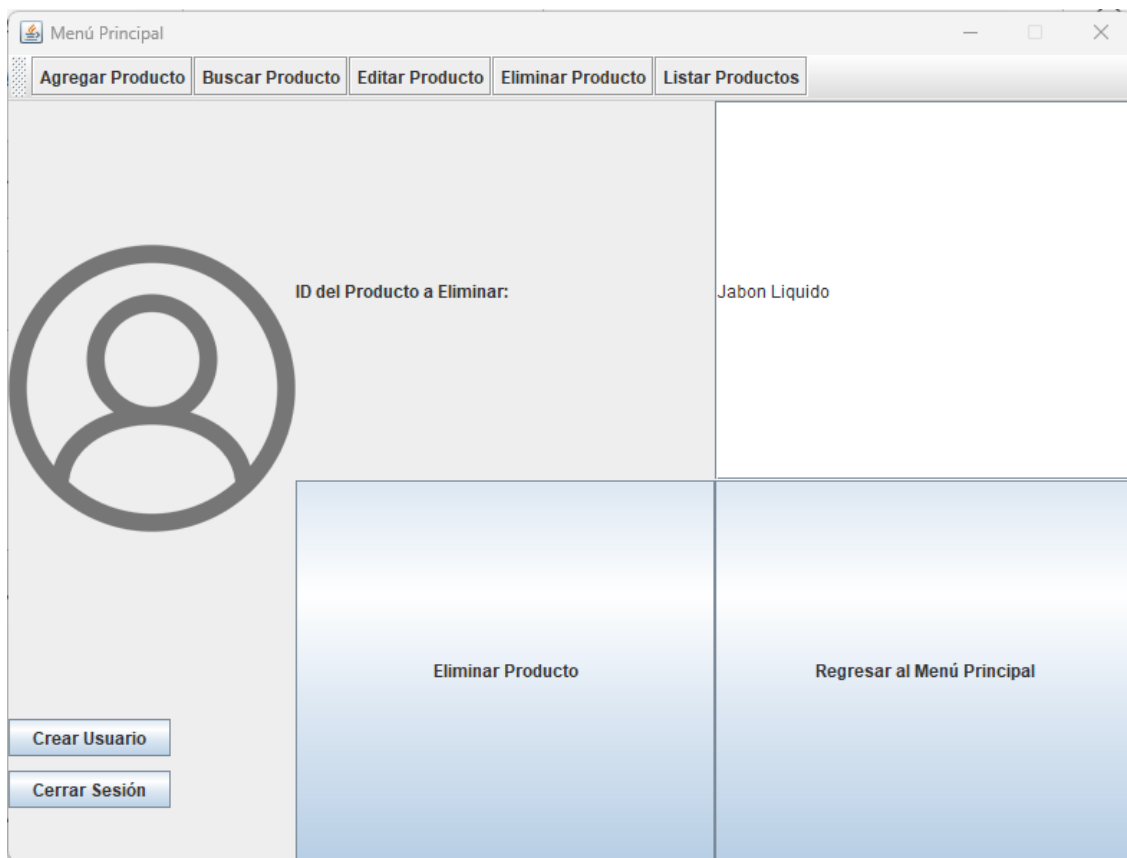


Figura 45. Guia de usuario

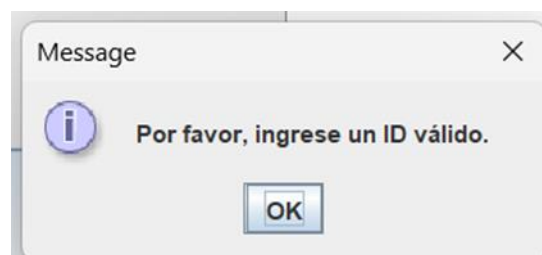


Figura 46. Guia de usuario

- **Crear Nuevo Usuario**

Usted debe conocer los roles que usted puede asignar a personas dentro de la aplicación para el manejo del inventario y que acciones pueden realizar. Los roles que se pueden asignar son los siguientes:

- **Administrador:** Este es el rol que usted va a manejar, este rol es importante por las acciones que puede realizar así que tenga cuidado en como asigna este rol a nuevas personas en el manejo de su inventario.

Con este rol se pueden llevar a cabo las siguientes acciones:

- Agregar nuevos productos
 - Buscar productos
 - Editar productos
 - Eliminar productos
 - Listar productos en el inventario
 - Crear nuevos usuarios
-
- **Gestor de inventario:** Este rol es para aquella persona que solo va a realizar operaciones con el inventario por lo que las acciones que puede realizar son las siguientes:
 - Agregar nuevos productos
 - Eliminar productos
 - Editar productos
 - Buscar productos
 - Listar productos en el inventario

Usuario regular: Este rol es para aquella persona que solo va a realizar consultas en el inventario por lo que las acciones que puede realizar son las siguientes:

- Buscar productos
- Listar productos en el inventario

Primero debe hacer clic en el botón **Crear Usuario** para que se le muestre la pantalla de crear nuevos usuarios.

The screenshot shows a web application window titled "Menú Principal". At the top, there is a horizontal menu with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The main area of the window is a form for creating a new user. On the left side of the form is a large circular icon representing a person. To the right of the icon are four input fields labeled "ID de Usuario:", "Nombre de Usuario:", "Contraseña:", and "Rol:". The "Rol:" field is a dropdown menu currently showing "Administrador". At the bottom left of the form are two buttons: "Crear Usuario" and "Cerrar Sesión". At the bottom right, there are two large blue buttons: "Crear Usuario" and "Regresar al Menú Principal".

Figura 47. Guía de usuario

Para agregar nuevos usuarios debe introducir tres datos en los campos en blanco y hacer clic en cada uno para capturar los datos. Los datos de cada campo se describen a continuación

- ID del usuario este tiene que ser un numero entero
- Nombre del usuario
- Contraseña asegúrese de ingresar una contraseña que no sea muy larga ya que puede ingresarla mal y no hay una opción para modificarla esto ocurre con los demás campos así que asegúrese de ingresar bien los datos.

Menú Principal

Agregar Producto | Buscar Producto | Editar Producto | Eliminar Producto | Listar Productos

ID de Usuario:

Nombre de Usuario:

Contraseña:

Rol:

Administrador

Crear Usuario

Cerrar Sesión

Crear Usuario

Regresar al Menú Principal

Figura 48. Guía de usuario

Para definir el rol debe de hacer clic en el siguiente icono .

Rol:

Administrador

Figura 49. Guía de usuario

El cual le desplegará las siguientes opciones:

Administrador

Administrador

Gestor de Inventario

Usuario Regular

Figura 50. Guía de usuario

Debe de hacer clic en uno de los roles que están definidos, si quiere cambiar de opción vuelva a hacer clic en el icono que antes se mostró y vuelva a seleccionar un rol. Si ya está seguro de los datos y el rol del usuario haga clic en el botón Crear Usuario.

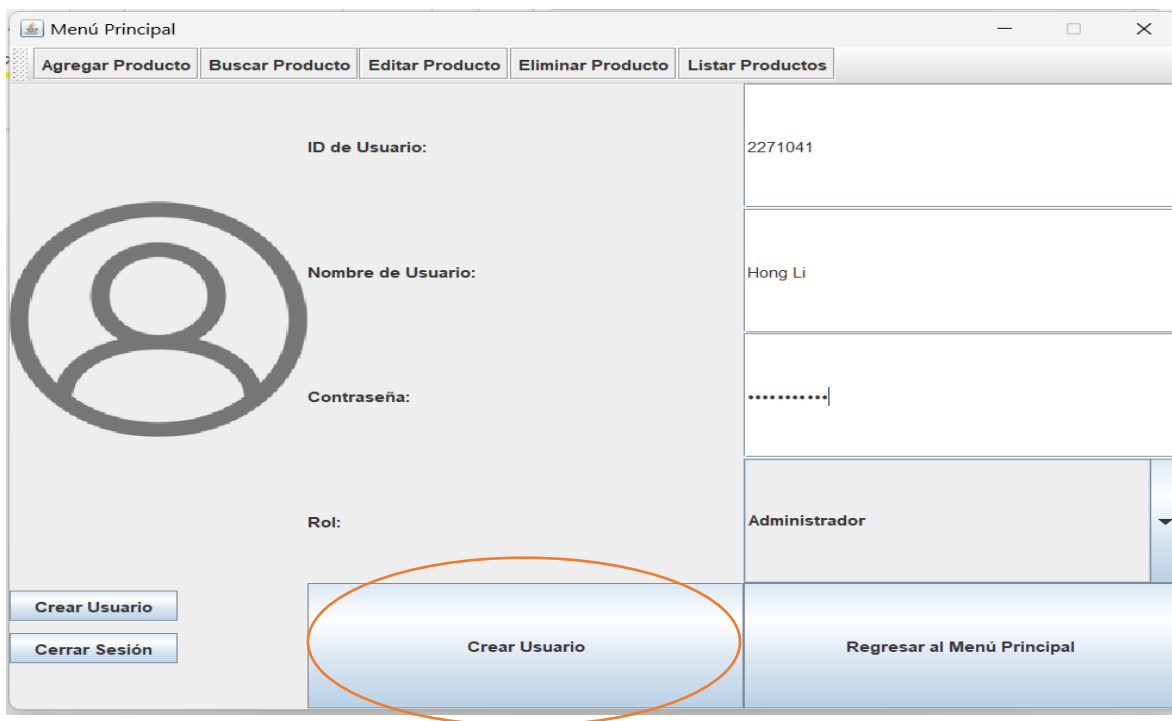


Figura 51. Guia de usuario

El sistema le arrojará un mensaje de usuario creado con éxito.

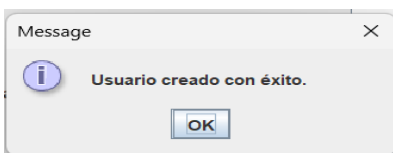


Figura 52. Guia de usuario

Si usted crear otro usuario con el mismo id el sistema no le permitirá crear al nuevo usuario y le arrojará un mensaje indicándole que el id ya está en uso. A continuación, se le muestra un ejemplo.

The screenshot shows a web application window titled "Menú Principal". At the top, there is a navigation bar with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The main area contains a user creation form. On the left, there is a large circular icon representing a user. The form fields are as follows:

- ID de Usuario:** A text input field containing the value "2271041".
- Nombre de Usuario:** A text input field containing the value "Andres".
- Contraseña:** A password input field with four dots "....".
- Rol:** A dropdown menu showing "Usuario Regular".

At the bottom left, there are two buttons: "Crear Usuario" and "Cerrar Sesión". At the bottom right, there are two large buttons: "Crear Usuario" and "Regresar al Menú Principal".

Figura 53. Guia de usuario

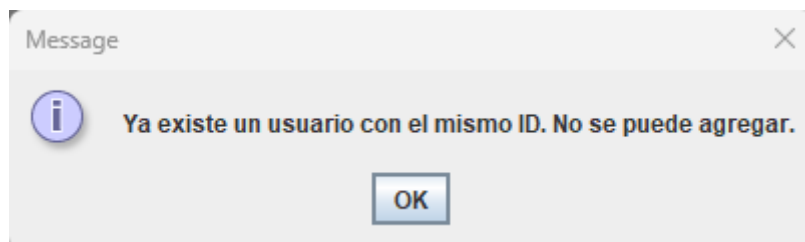


Figura 54. Guia de usuario

Si no ingresa un ID valido el sistema no le permitirá crear un nuevo usuario.

Menú Principal

Agregar Producto Buscar Producto Editar Producto Eliminar Producto Listar Productos

ID de Usuario: jsfjh3928

Nombre de Usuario: Andres

Contraseña:

Rol: Usuario Regular

Crear Usuario Cerrar Sesión

Crear Usuario Regresar al Menú Principal

Figura 55. Guia de usuario

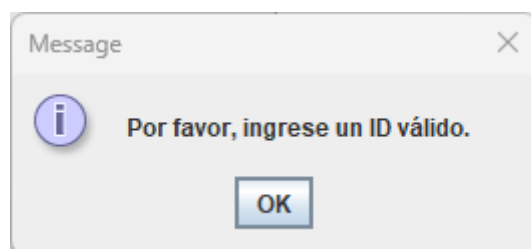


Figura 56. Guia de usuario

Si usted ingresa otros datos en el nombre de usuario, así como la contraseña el usuario va a ser creado sin problemas.

- **Listar Productos en Inventario**

Al presionar clic en este botón se le mostraran todos los productos que estén en el inventario en forma de tabla con sus datos respectivos. De esta forma puede asegurarse de que los productos realmente se han agregado al inventario.

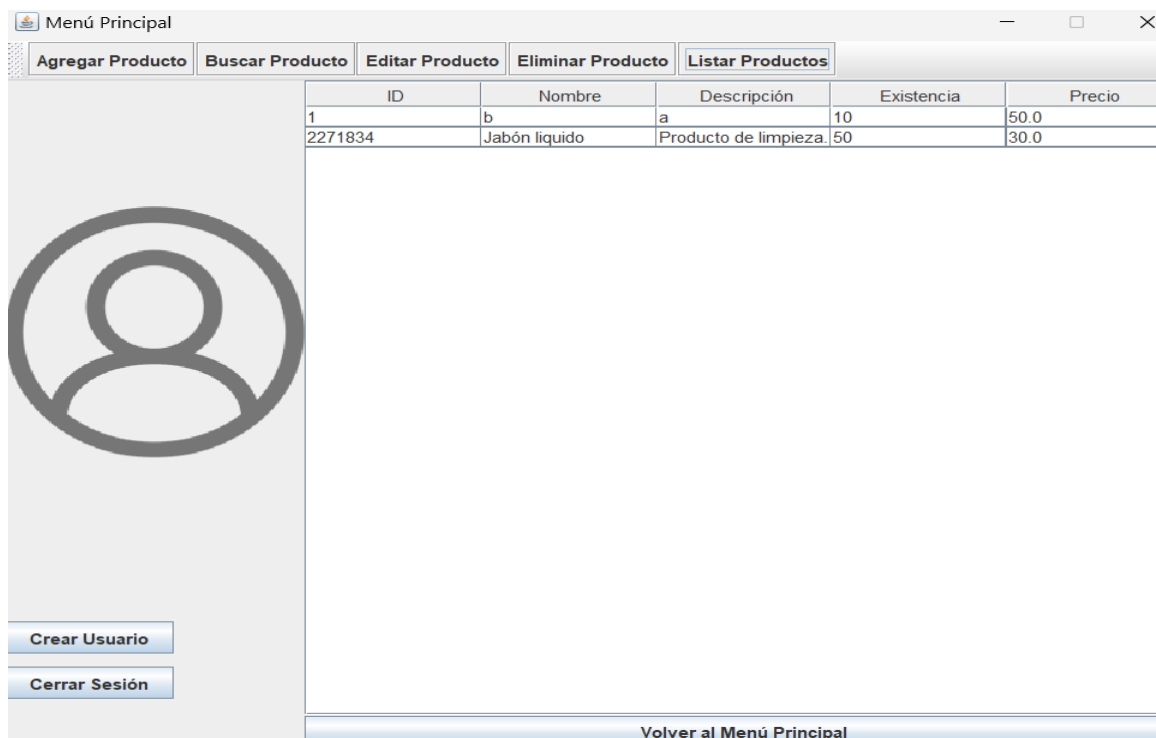


Figura 57. Guía de usuario

En cuanto más productos agregue aparecerá una barra en la parte derecha de la tabla con la cual usted podrá hacer clic y deslizar ya sea para arriba o hacia abajo para ver los productos de su inventario.

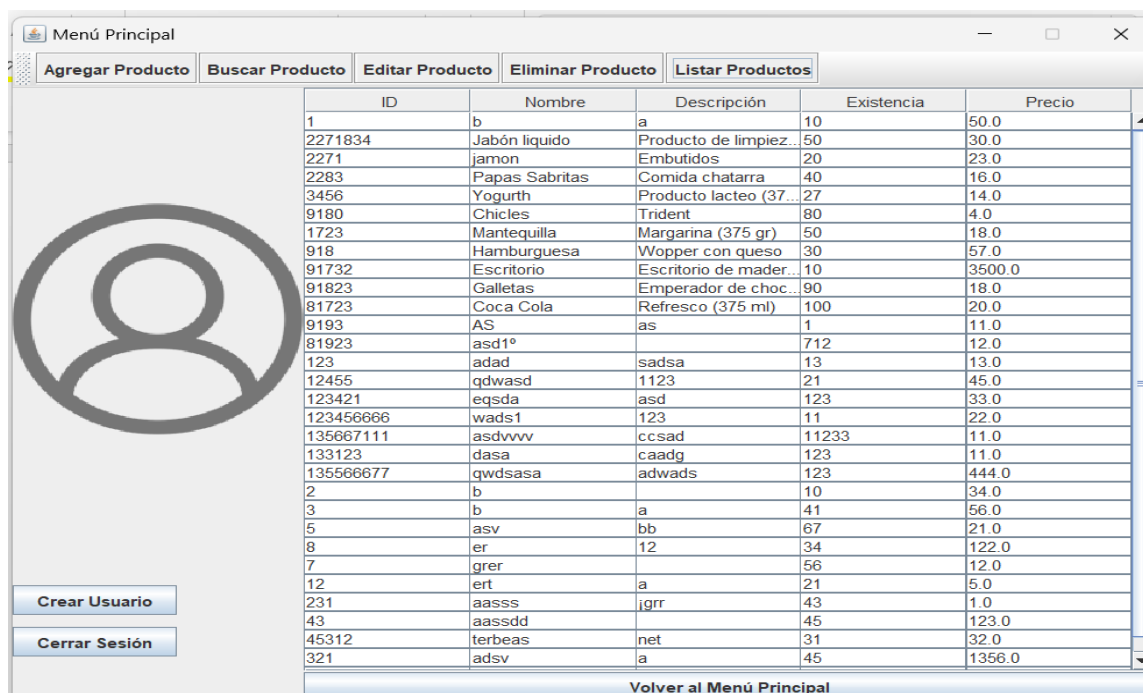


Figura 58. Guia de usuario

Al hacer clic en el botón Volver al Menú Principal lo devolverá a la pantalla del menú de opciones

- **Editar Producto**

Cuando usted haga clic en el botón de “Editar Producto” en la barra de botones le aparecerá la siguiente pantalla.

The screenshot shows a web application window titled "Menú Principal". At the top, there is a navigation bar with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The main content area is divided into several sections. On the left, there is a large circular icon representing a user. To the right of this icon, there are three labels: "ID del Producto a Editar:", "Campo a Editar", and "Nuevo Valor:". Below these labels, there are three input fields. The first input field is empty. The second input field contains the text "nombre" and has a dropdown arrow on its right side. The third input field is empty. At the bottom left, there are two buttons: "Crear Usuario" and "Cerrar Sesión". At the bottom right, there are two large buttons: "Editar Producto" and "Regresar al Menú Principal".

Figura 59. Guía de usuario

La forma de editar un producto es la siguiente:

1. Ingrese el ID del producto del cual va a realizar la modificación en el campo de texto en blanco. Si el producto no está registrado en el inventario se le notificara esto con un mensaje, en caso de que no ingrese un ID valido de igual forma se le notificara esto.

The screenshot shows a web application window titled "Menú Principal". At the top, there is a navigation bar with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The "Editar Producto" button is active. The main content area is divided into two columns. The left column contains a large circular icon of a person, the label "ID del Producto a Editar:" followed by the value "29384", the label "Campo a Editar" followed by a dropdown menu showing "nombre", and the label "Nuevo Valor:" followed by the text "Jabon Salvo". The right column contains two large blue buttons: "Editar Producto" and "Regresar al Menú Principal". At the bottom left, there are two smaller buttons: "Crear Usuario" and "Cerrar Sesión".

Figura 60. Guía de usuario

The screenshot shows a "Message" dialog box with a close button (X) in the top right corner. It contains an information icon (i) and the text "Producto no encontrado en el inventario." Below the text is an "OK" button.

Figura 61. Guía de usuario

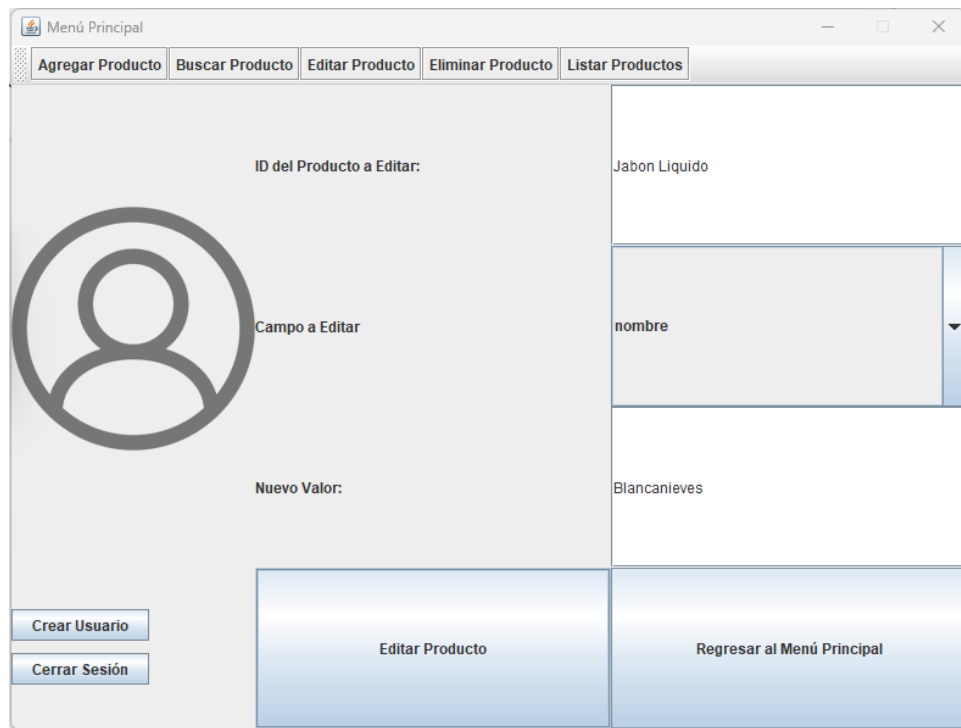


Figura 62. Guía de usuario

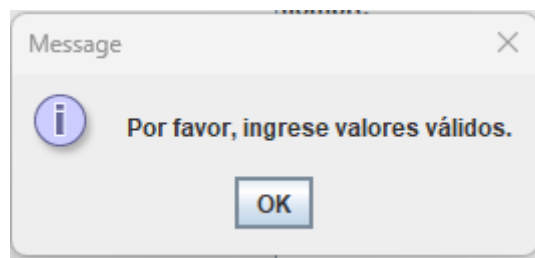



Figura 63. Guía de usuario

2. Haga clic en el boton  para desplegar las opciones que se pueden modificar de un producto:

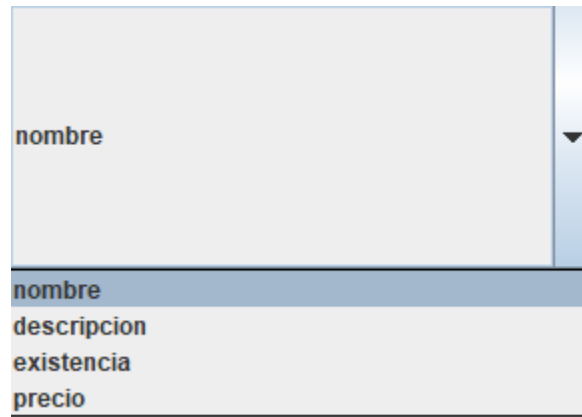


Figura 64. Guia de usuario

Seleccione la modificación que va a realizar. Si quiere cambiar de opción

vuelva a hacer clic en el botón  y seleccione otro campo a editar.

3. Ingrese el nuevo valor haciendo clic en el campo de texto en blanco. Para el ingreso de un nuevo nombre o descripción usted puede utilizar cualquier carácter que usted considere letras, comas, números y demás. Si selecciona la opción de existencias o precio asegúrese de ingresar un número válido por ejemplo si va a modificar la cantidad de existencias debe ingresar un número entero por ejemplo 3 lo mismo aplica para el precio, pero usted puede ingresar un número con decimal por ejemplo 4.5 en cualquier caso el sistema le notificará. Para el caso del producto de Jabón Líquido que se mostró en el apartado de productos el ejemplo es el siguiente:

The screenshot shows a web application window titled "Menú Principal". At the top, there is a navigation bar with five tabs: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The "Editar Producto" tab is currently selected. The main content area is divided into two columns. The left column contains a large circular icon with a stylized 'G' inside, and three labels: "ID del Producto a Editar:", "Campo a Editar", and "Nuevo Valor:". The right column contains three input fields. The first field is labeled "ID del Producto a Editar:" and contains the value "2271834". The second field is labeled "Campo a Editar" and contains the value "existencia". The third field is labeled "Nuevo Valor:" and contains the value "veinticinco". At the bottom of the window, there are four buttons: "Crear Usuario", "Cerrar Sesión", "Editar Producto", and "Regresar al Menú Principal".

Figura 65. Guía de usuario

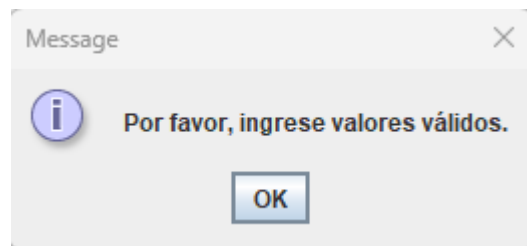


Figura 66. Guía de usuario

A continuación, se le muestra un ejemplo de cómo ingresar modificar el valor de uno de sus productos.

The screenshot shows a web application window titled "Menú Principal". At the top, there is a navigation bar with five buttons: "Agregar Producto", "Buscar Producto", "Editar Producto", "Eliminar Producto", and "Listar Productos". The "Editar Producto" button is highlighted. The main content area is divided into two columns. The left column contains a large circular icon of a person, and the right column contains three input fields. The first field is labeled "ID del Producto a Editar:" and contains the value "2271834". The second field is labeled "Campo a Editar" and contains the value "existencia". The third field is labeled "Nuevo Valor:" and contains the value "25". At the bottom of the left column, there are two buttons: "Crear Usuario" and "Cerrar Sesión". At the bottom of the right column, there are two large buttons: "Editar Producto" and "Regresar al Menú Principal".

Figura 67. Guía de usuario

Una vez que haya ingresado el ID, el valor a modificar y el nuevo valor haga clic en el botón de “Editar Producto” que está en la parte inferior de la pantalla posteriormente se le mostrara un mensaje de que el producto fue editado con éxito.

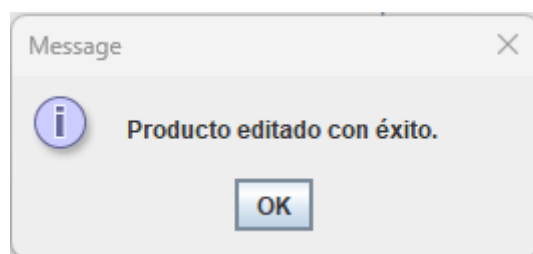


Figura 68. Guía de usuario

Si quiere ver la modificación realizada haga clic en el botón de **Listar Productos** para ver los productos del inventario y comprobar que su modificación realmente se realizó. A continuación, se le muestra si el sistema realizó la modificación.

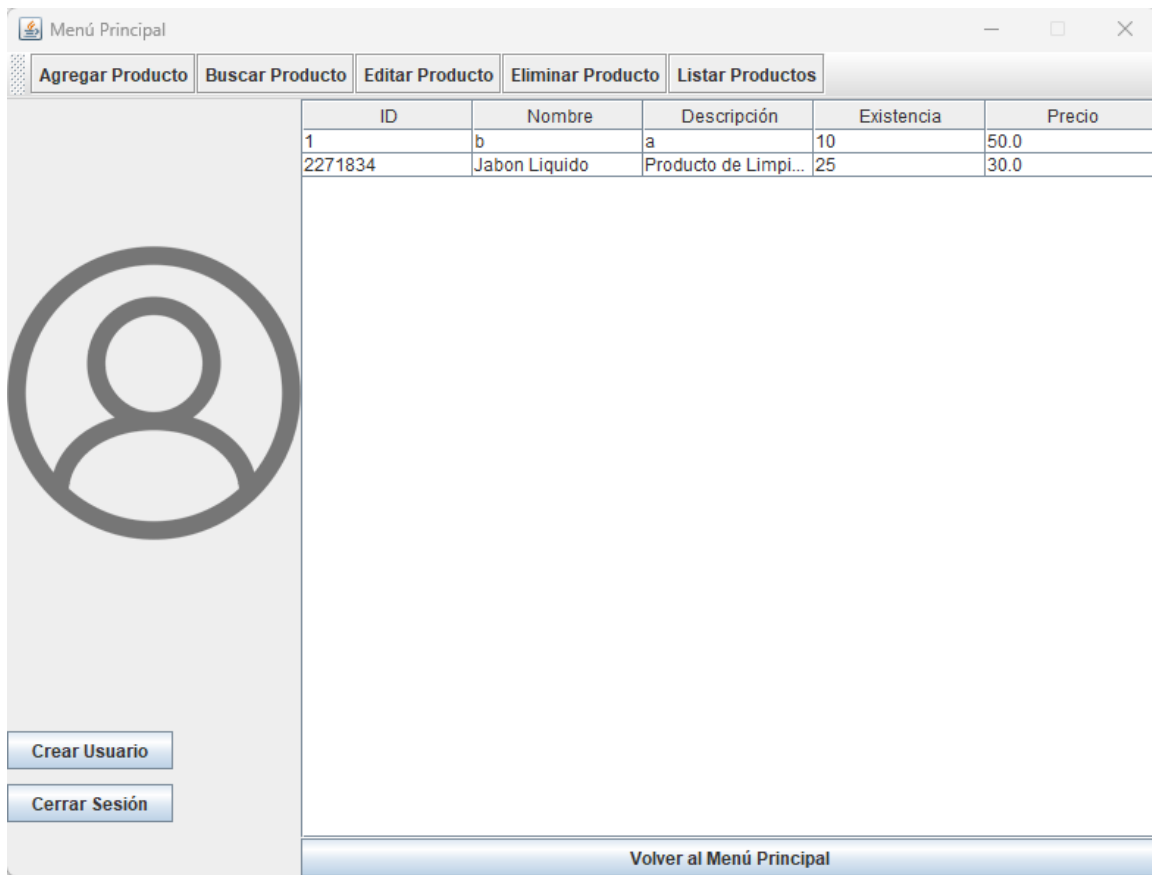


Figura 69. Guía de usuario

Usted puede modificar los detalles de su producto cuantas veces sea necesario para más modificaciones siga los pasos que anteriormente se han descrito.

- **Cerrar sesión**

Cuando haga clic en este botón el sistema inmediatamente lo regresa a la ventana de inicio de sesión

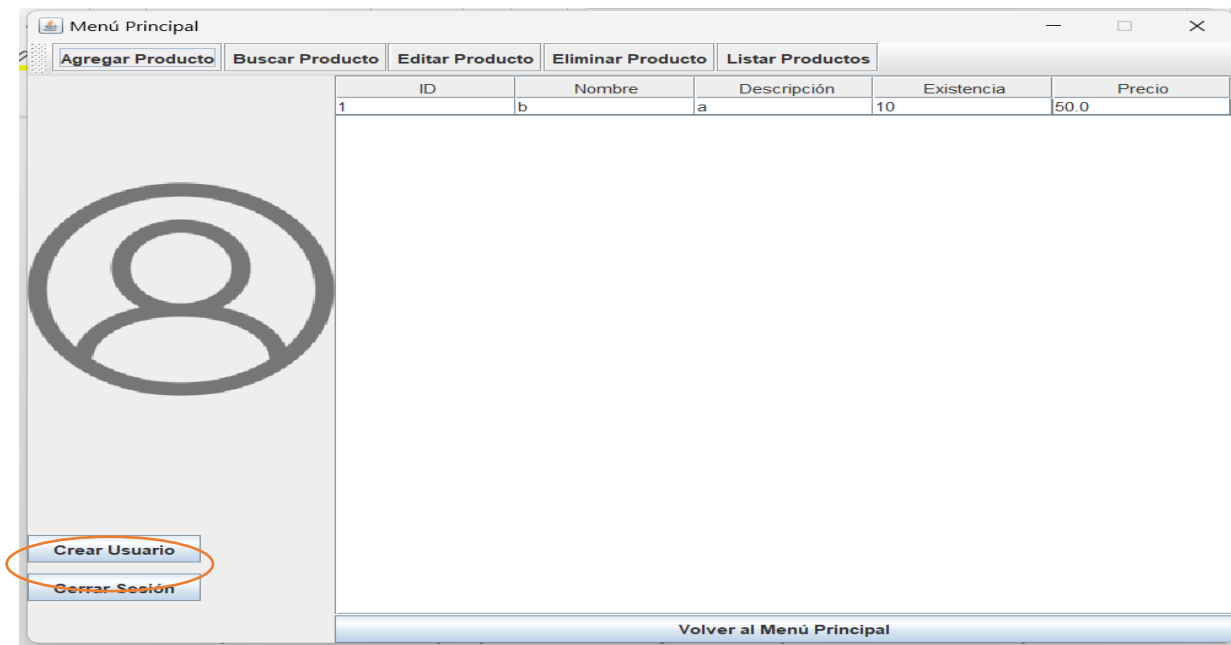


Figura 70. Guía de usuario

Para volver acceder a la aplicación debe ingresar nuevamente los datos de nombre de usuario y contraseña, así como lo hizo para ingresar al sistema en un inicio.

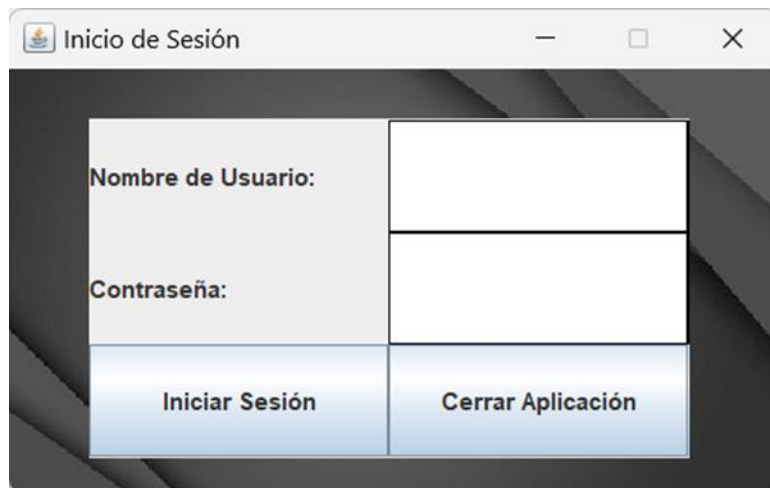
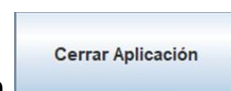



Figura 71. Guía de usuario

Para terminar con la ejecución de la aplicación haga en el botón



esta acción automáticamente guardara los productos que haya ingresado, así como

los usuarios agregados al sistema los datos van a ser almacenados en un archivo de Excel llamado “usuarios” y “productos. También puede clic en el botón  para finalizar la ejecución, pero esto no guardara los datos que ingreso así que es recomendable que haga clic en el botón “Cerrar Aplicación”.

En caso de perder su contraseña de inicio de sesión, tener problemas con iniciar sesión en la aplicación u otra situación en donde la aplicación no se ejecute adecuadamente, por favor ponerse en contacto al siguiente correo uapsoftwaresolution35@gmail.com para llevar a cabo acciones requeridas.

Conclusiones

La documentación del proyecto es importante como para el equipo de desarrolladores que trabajen con el sistema, así como para el usuario final para poder entregarla a futuros desarrolladores que mejoren el inventario y que estos puedan entender todo lo relacionado con el sistema.

Antes de empezar con un proyecto hay dos aspectos que deben ser tomados en cuenta el tiempo que se dispone y capturar los requisitos tanto funcionales como no funcionales del cliente esto para poder realizar una administración del tiempo que se dispone, planificar las actividades y asignar fechas límite. Llevar un registro de las actividades permite ver que aquellas que ya se han trabajado y dar atención a aquellas que están tomando más tiempo. La elección de la metodología en cascada fue la más adecuada para el desarrollo del proyecto ya que primero se definen los requisitos funcionales como no funcionales después se procede al diseño del sistema, la implementación, pruebas de integración finalmente la operación y mantenimiento para un proyecto escolar no es necesario llevar a cabo revisiones con un cliente y tener que prestar atención a las necesidades del cliente esto permite que el desarrollo del proyecto sea más lineal.

Realizar pruebas es una parte muy importante del proyecto y hay que determinar la cantidad de tiempo disponible hasta que llegue la fecha de entrega para poder realizar una plan con el fin de llevar a cabo las pruebas necesarias que aporten más valor. Durante el proyecto hubo ausencia de pruebas unitarias para poder probar el funcionamiento correcto de los módulos desarrollados esto pudo traer problemas durante la integración de los módulos ya que de no detectarse los errores asociados a cada módulo había una posibilidad muy alta que durante la integración de los módulos múltiples fallos en los módulos podrían haberse propagado por todo el sistema lo que conlleva un mayor tiempo en la resolución de errores y en el peor de los escenarios no poder cumplir con la fecha de entrega del software. También es necesario destacar que un buen diseño del software de la arquitectura, de las relaciones y métodos de los distintos componentes del software contribuyen a una dirección clara durante el desarrollo y la documentación del proyecto esto evita que

continuamente se estén realizando cambios en el diseño, en la documentación, en el código lo que dificulta el avance del proyecto.

Mejoras futuras

Desarrollo de una función de notificaciones para alertar al usuario sobre bajos niveles de stock y que el usuario determine un nivel necesario de existencias para mandar alertas.

Un historial de acciones que registra todas las acciones llevadas a cabo por un usuario como ingreso de productos, editar productos, eliminación de productos y demás con el propósito de llevar un seguimiento sobre los cambios realizados.

Incorporar una función que permita el manejo de múltiples inventarios esto para llevar un mejor control de productos que estén organizados en diferentes inventarios.

Agregar funcionalidades de respaldo y restauración para evitar la pérdida de datos en caso de que se haya eliminado la información por error o debido a la falla del sistema.

Permitir al usuario principal la eliminación de los usuarios registrados en el sistema y poder modificar datos como el id, el nombre de usuario y contraseña.

Implementar bases de datos para manejar una mayor cantidad de productos y usuarios, así como realizar una organización de los datos para poder implementar opciones de búsqueda más avanzadas.

Índice de Imágenes

| | |
|---|----|
| Figura 1. Modelo de Desarrollo de Software | 32 |
| Figura 2. Diagrama UML | 44 |
| Figura 3. Implementacion | 46 |
| Figura 4. Implementacion | 46 |
| Figura 5. Implementacion | 47 |
| Figura 6. Implementacion | 47 |
| Figura 7. Implementacion | 48 |
| Figura 8. Implementacion | 48 |
| Figura 9. Prototipo Interfaz | 61 |
| Figura 10. Prototipo Interfaz | 61 |
| Figura 11. Prototipo Interfaz | 62 |
| Figura 12. Prototipo Interfaz | 63 |
| Figura 13. Prototipo Interfaz | 64 |
| Figura 14. Desarrollo: Ventana - Inicio Sesión e Implementación | 66 |
| Figura 15. Pantalla principal e Implementacion | 68 |
| Figura 16. Panel Agregar Producto e Implementación | 69 |
| Figura 17. Pantalla Agregar Producto e Implementacion | 71 |
| Figura 18. Panel Eliminar Producto e Implementación | 72 |
| Figura 19. Panel Editar Producto e Implementacion | 74 |
| Figura 20. Panel Crear Usuario e Implementación | 75 |
| Figura 21. Panel Listar Productos e Implementación | 76 |
| Figura 22. Guía de usuario | 79 |
| Figura 23. Guía de usuario | 79 |
| Figura 24. Guía de usuario | 80 |
| Figura 25. Guía de usuario | 80 |
| Figura 26. Guía de usuario | 81 |
| Figura 27. Guía de usuario | 82 |
| Figura 28. Guía de usuario | 83 |
| Figura 29. Guía de usuario | 84 |
| Figura 30. Guía de usuario | 85 |
| Figura 31. Guía de usuario | 86 |
| Figura 32. Guía de usuario | 86 |
| Figura 33. Guía de usuario | 87 |
| Figura 34. Guía de usuario | 87 |
| Figura 35. Guía de usuario | 88 |
| Figura 36. Guía de usuario | 89 |
| Figura 37. Guía de usuario | 89 |
| Figura 38. Guía de usuario | 89 |
| Figura 39. Guía de usuario | 90 |
| Figura 40. Guía de usuario | 90 |
| Figura 41. Guía de usuario | 91 |
| Figura 42. Guía de usuario | 92 |

| | |
|---|-----|
| <i>Figura 43. Guia de usuario</i> | 92 |
| <i>Figura 44. Guia de usuario</i> | 93 |
| <i>Figura 45. Guia de usuario</i> | 93 |
| <i>Figura 46. Guia de usuario</i> | 93 |
| <i>Figura 47. Guia de usuario</i> | 95 |
| <i>Figura 48. Guia de usuario</i> | 96 |
| <i>Figura 49. Guia de usuario</i> | 96 |
| <i>Figura 50. Guia de usuario</i> | 96 |
| <i>Figura 51. Guia de usuario</i> | 97 |
| <i>Figura 52. Guia de usuario</i> | 97 |
| <i>Figura 53. Guia de usuario</i> | 98 |
| <i>Figura 54. Guia de usuario</i> | 98 |
| <i>Figura 55. Guia de usuario</i> | 99 |
| <i>Figura 56. Guia de usuario</i> | 99 |
| <i>Figura 57. Guia de usuario</i> | 100 |
| <i>Figura 58. Guia de usuario</i> | 101 |
| <i>Figura 59. Guia de usuario</i> | 102 |
| <i>Figura 60. Guia de usuario</i> | 103 |
| <i>Figura 61. Guia de usuario</i> | 103 |
| <i>Figura 62. Guia de usuario</i> | 104 |
| <i>Figura 63. Guia de usuario</i> | 104 |
| <i>Figura 64. Guia de usuario</i> | 105 |
| <i>Figura 65. Guia de usuario</i> | 106 |
| <i>Figura 66. Guia de usuario</i> | 106 |
| <i>Figura 67. Guia de usuario</i> | 107 |
| <i>Figura 68. Guia de usuario</i> | 107 |
| <i>Figura 69. Guia de usuario</i> | 108 |
| <i>Figura 70. Guia de usuario</i> | 109 |
| <i>Figura 71. Guia de usuario</i> | 109 |

Referencias

- **Bibliografía**

Schildt, H. (2009). Java. (7ma ed.). Mc Graw Hill

Sommerville, I. (2011). Ingeniería de Software. (9na ed.). Pearson

Pressman, R. (2010). Ingeniería del Software. (7ma ed.). Mc Graw Hill.

- **Recursos digitales**

Overview Inventory Management. (s. f.). SAP.

https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/91b21005dded4984bccccf4a69ae1300c/9d688de41e684f2c9c6b467ef8e0ba6f.html

Vaibhavi, K. (1 de octubre de 2022). SAP Inventory Management – Everything you must know. SAP.

[SAP Inventory Management – Everything you must know | SAP Blogs](#)