

Manual técnico del software de la ferretería llavenuda



Elaborado por:

Sara Mabel López Chalin 5390-22-6672

José Adolfo Aldana Chigna 5390-22-10734

Mario Sebastián González Arenales - 5390-22-8808

Erwin Javier Andrade López 5390-20-3306

Contenido

Manual técnico del software de la ferretería llavenuda	1
.....	1
Introducción	1
Descripción	1
Funcionalidades del sistema:	2
Registro y mantenimiento de usuarios y artículos.	2
Tecnologías utilizadas:	2
Descripción detallada del código	2
Bibliotecas utilizadas.	3
Funciones	5
Verificación de conexión a la base de datos	5
Visualización de usuarios	6
Inserción de usuarios	6
Estructura del código	6
Versionamiento del software	9
Dependencias del software	10
Recomendaciones y consejos	11

Introducción

En este manual , hemos discutido sobre un sistema de gestión de ferretería desarrollado en C++ utilizando MySQL como base de datos. Este sistema permite a los usuarios registrar y mantener el inventario de productos, gestionar usuarios, clientes y personal, y generar informes y estadísticas sobre las ventas y el inventario.

El sistema cuenta con una interfaz de consola simple que muestra un menú principal con opciones para gestionar el inventario, los clientes y el personal. Cada opción tiene submenús con diversas funciones, como ver, agregar y eliminar registros.

Para ejecutar el sistema, se requiere un sistema operativo compatible con C++ y MySQL, como Windows, un servidor de MySQL instalado y la biblioteca de MySQL Connector/C++ configurada en el sistema. Además, se necesita un compilador de C++ compatible con C++11.

En el sistema de gestión de ferretería proporciona una solución eficiente para administrar el inventario de productos y realizar tareas relacionadas con usuarios, clientes y personal. Utilizando C++ y MySQL, se ha desarrollado un código que interactúa con la base de datos y permite realizar operaciones CRUD. La interfaz de consola simplifica la interacción con el sistema, ofreciendo opciones de gestión y generación de informes.

Descripcion

El sistema de ferretería es una aplicación de software diseñada para ayudar a los usuarios a registrar y mantener el inventario de productos de una tienda de ferretería, así como para generar informes y estadísticas sobre las ventas y el inventario. Este manual técnico proporciona información detallada sobre el diseño, la implementación y el funcionamiento del sistema.

Funcionalidades del sistema:

Registro y mantenimiento de usuarios y artículos.

Sistema de gestión de usuarios: permite agregar nuevos usuarios, ver la lista de usuarios existentes y eliminar usuarios. Sistema de gestión de artículos: permite agregar nuevos artículos, ver la lista de artículos existentes y eliminar artículos. Administración del inventario de productos, el registro de clientes y el registro de personal. Interfaz de consola simple con un menú principal que muestra tres opciones: inventario, clientes y personal. Cada opción tiene su propio menú con varias opciones de gestión, como ver, agregar y eliminar registros.

Tecnologías utilizadas:

El sistema está escrito en C++ y utiliza la librería MySQL Conector/C++ para conectarse a una base de datos MySQL.

Descripción detallada del código:

El código se conecta a una base de datos MySQL para realizar operaciones CRUD en una tabla llamada "Personal". El programa permite registrar y mantener información de clientes, usuarios, productos y ventas en una base de datos MySQL. El usuario debe ingresar su nombre de usuario y contraseña en una pantalla de inicio de sesión en la consola para acceder al menú principal del sistema. El programa interactúa con la tabla "usuarios" para verificar si los datos de inicio de sesión son correctos. El menú principal del sistema muestra tres opciones: inventario, clientes y personal.

Cada opción tiene varias opciones de gestión, como ver, agregar y eliminar registros.

Requisitos del sistema o programa

Para ejecutar el código del sistema de ferretería en C++ y MySQL, se requiere lo siguiente:

- Un sistema operativo compatible con C++ y MySQL, como Windows 7 o superior.
- Un servidor de MySQL instalado en la computadora.
- La biblioteca de MySQL Connector/C++ instalada y configurada en el sistema.
- Un compilador de C++ compatible con C++11, como Visual Studio Code o CodeBlocks.
- La computadora debe tener al menos un procesador Intel Core i3, 4GB de memoria RAM y 10GB de espacio libre en el disco duro.

Es importante mencionar que el código ha sido probado específicamente en Windows 10, pero puede funcionar en otros sistemas operativos compatibles. Además, se recomienda utilizar un compilador de C++ actualizado y una versión reciente del servidor de MySQL para garantizar la compatibilidad y la estabilidad del sistema.

Bibliotecas utilizadas.

- `iostream`: para entrada y salida en consola.
- `mysql.h`: para interactuar con una base de datos MySQL.
- `mysql_error.h`: para manejar errores específicos de MySQL

Estructuras de datos

El programa utiliza las siguientes estructuras para manejar los datos de la base de datos:

- `Usuarios`:
- `ID_Usuario`: int
- `Tipo_Usuario`: int
- `Perfil_Usuario`: char[45]

- Nombre_Usuario: char[45]
- Pass_Usuario: int
- Clientes:
- ID_Cliente: int
- Nombre_Clientes: char[45]
- Apellido_Clientes: char[45]
- Telefono_Clientes: int
- ID_Usuario_Clientes: int
- Articulos:
- ID_Articulos: int
- Nombre_Articulo: char[45]
- Cantidad_Articulo: int
- Precio_Articulo: int
- Personal:
- ID_Personal: int
- Nombre_Personal: char[45]
- Apellido_Personal: char[45]
- Puesto_Personal: char[45]
- Telefono_Personal: int
- Fecha_Incorporacion_year: int
- Fecha_Incorporacion_mes: char[45]
- ID_Usuario_Personal: int
- Proveedores:

- ID_Proveedores: int
- Nombre_Proveedores: char[45]
- CorreoElectronico_Proveedores: char[45]
- Telefono_Proveedores: int
- ID_articulo_proveedores: int

Funciones

- El programa utiliza varias funciones para llevar a cabo diferentes tareas.
- La función "menu_reportes()" muestra un menú con cuatro opciones: compras, ventas, kardex y salir. Dependiendo de la opción seleccionada por el usuario, se muestra un mensaje de bienvenida correspondiente. Si se selecciona la opción 4, el programa regresa al menú principal. Si se ingresa una opción inválida, se muestra un mensaje de error y se solicita nuevamente la entrada del usuario.
- La función "menu_registro_y_Mantenimiento()" muestra un menú con opciones para registrar y mantener usuarios, clientes, artículos, personal y proveedores. Utiliza la estructura "cConn" para conectarse a una base de datos MySQL y diferentes estructuras de datos para manejar los datos de cada tabla en la base de datos. Utiliza un ciclo do-while para mostrar el menú y procesar la entrada del usuario. Cada opción del menú llama a una función que realiza operaciones específicas en la base de datos, como la inserción o actualización de registros.
- La función "registrar_usuario(MYSQL *objDatos, cConn connDB)" permite al usuario registrar un nuevo usuario en la tabla "Usuarios" de la base de datos. El usuario ingresa los datos requeridos en la consola y luego se utiliza una consulta SQL para insertar los datos en la tabla.

Verificación de conexión a la base de datos

- La función "verificarConexion(MYSQL* objDatos)" verifica que la conexión a la base de datos se haya realizado correctamente. Toma como parámetro un objeto de conexión a la base de datos "objDatos" y devuelve un valor booleano que indica si la conexión se realizó correctamente o no.

Visualización de usuarios

- La función "verUsuarios(MYSQL* objDatos)" muestra en la consola todos los usuarios registrados en la tabla "usuarios" de la base de datos. Realiza una consulta a la tabla y recorre los resultados obtenidos para mostrar la información correspondiente.

Insertión de usuarios

- La función "insertarUsuario(MYSQL* objDatos)" permite insertar un nuevo usuario en la base de datos. Solicita al usuario los datos del nuevo usuario (ID, tipo de usuario, perfil, nombre y contraseña) y utiliza una consulta INSERT para agregar los datos a la tabla.

Estructura del código

- El código se divide en varias clases y funciones.
- La clase "ConexionDB" contiene funciones para establecer una conexión con la base de datos y realizar consultas.
- La función "m_conectar_BaseDeDatos()" se utiliza para establecer una conexión con la base de datos. Toma como argumento un objeto MYSQL* y una consulta SQL.
- La función "m_ejecutar_consulta()" se utiliza para ejecutar una consulta SQL en la base de datos. Toma como argumento un objeto MYSQL* y una consulta SQL. Devuelve un objeto MYSQL_RES* si la consulta se ejecuta correctamente, o nullptr si se produce un error.
- La función "m_mostrar_error()" se utiliza para mostrar un mensaje de error si ocurre algún problema al ejecutar una consulta SQL. Toma como argumento un objeto MYSQL* y un mensaje de error.
- La clase "Usuarios" representa a un usuario y contiene atributos como ID_Usuario, Tipo_Usuario, Perfil_Usuario, Nombre_Usuario y Pass_Usuario.
- La función "main()" es la función principal del programa. Inicia la conexión con la base de datos y muestra el menú principal al usuario.

El manual técnico del sistema de ferretería se divide en varias secciones, cada una enfocada en un tema específico. A continuación, se presenta la estructura del manual:

1. Componentes 1.1. Clase ConexionDB

- Descripción: Esta clase se encarga de gestionar la conexión a la base de datos MySQL. Proporciona métodos para conectar, desconectar y ejecutar consultas SQL en la base de datos.

1.2. Estructuras de datos

- Usuarios: Representa a un usuario de la aplicación, con información como ID, nombre de usuario y contraseña.

- Clientes: Representa a un cliente registrado en la base de datos, con información como ID, nombre, apellido, teléfono y ID de usuario.

- Productos: Representa a un producto disponible en la tienda, con información como ID, nombre, descripción, precio y existencias.

- Ventas: Representa a una venta realizada en la tienda, con información como ID, fecha, total y ID de cliente.

1.3. Funciones de manejo de la base de datos

- conectarBaseDeDatos: Conecta el programa a la base de datos MySQL utilizando la biblioteca MySQL Connector/C++.

- ejecutarConsulta: Ejecuta una consulta SQL en la base de datos utilizando la conexión establecida.

- verUsuarios: Muestra los usuarios registrados en la base de datos.

- insertarUsuario: Inserta un nuevo usuario en la base de datos.

- verClientes: Muestra los clientes registrados en la base de datos.

- insertarCliente: Inserta un nuevo cliente en la base de datos.

- verProductos: Muestra los productos disponibles en la tienda.

- insertarProducto: Inserta un nuevo producto en la base de datos.

- realizarVenta: Registra una nueva venta en la base de datos.

2. Interfaz de usuario

- Descripción: El sistema presenta una interfaz de usuario en la línea de comandos que permite al usuario interactuar con la base de datos. La interfaz muestra un menú con opciones para ver registros existentes, insertar nuevos registros y salir del programa. El usuario selecciona una opción ingresando un número correspondiente en la línea de comandos.

3. Componentes del Código 3.1. Estructura Proveedores

- Descripción: Define los campos necesarios para representar la información de un proveedor en la base de datos, como ID, nombre, correo electrónico, teléfono y ID de artículo suministrado.

3.2. Función insertarProveedor

- Descripción: Permite al usuario insertar un nuevo proveedor en la base de datos. Solicita al usuario que ingrese los valores para cada campo de la estructura Proveedores y luego construye y ejecuta una consulta SQL para insertar los datos en la tabla de proveedores.

3.3. Función mostrarProveedores

- Descripción: Recupera todos los registros de proveedores de la base de datos y los muestra en la consola. Utiliza una consulta SQL para seleccionar todos los datos de la tabla de proveedores y luego recorre el conjunto de resultados para mostrar cada proveedor en la consola.

3.4. Función main

- Descripción: Punto de entrada del programa. Inicia una conexión con la base de datos y muestra un menú en la consola para que el usuario elija entre ver proveedores, insertar proveedores o salir del programa. Según la opción seleccionada por el usuario, se invoca la función correspondiente para realizar la acción deseada.

4. Diseño de la base de datos

- Descripción: Se describe la estructura de la base de datos utilizada por el sistema de ferretería, incluyendo las tablas usuarios, productos y ventas, con sus respectivos campos y propósitos.

5. Implementación del sistema

- Descripción: Se explica la composición del sistema de ferretería, que consta de varios archivos de código fuente. El archivo principal es "main.cpp", que contiene la interfaz de usuario y la lógica principal del sistema. También se mencionan otros archivos que contienen funciones auxiliares para interactuar con la base de datos y generar informes.

Versionamiento del software

En esta versión inicial del código, se implementa un sistema de gestión de personal para una ferretería utilizando la base de datos MySQL. El programa permite ver y registrar información sobre el personal de la ferretería.

El código se divide en funciones para facilitar el mantenimiento y la legibilidad. El menú principal ofrece las siguientes opciones:

1. Ver Personal: Permite visualizar los registros de personal almacenados en la base de datos. La información se muestra en la consola en un formato legible.

2. Insertar Personal: Permite ingresar los datos de un nuevo miembro del personal y almacenarlos en la base de datos.

3. Salir: Finaliza la ejecución del programa.

El código utiliza la biblioteca MySQL Connector/C++ para interactuar con la base de datos. Se implementa una función de conexión a la base de datos que establece la conexión y verifica su éxito.

En la opción "Ver Personal", se realiza una consulta a la base de datos para recuperar los registros de la tabla "Personal". Los datos se muestran en la consola en un formato legible.

En la opción "Insertar Personal", se solicita al usuario que ingrese los datos del nuevo miembro del personal. Estos datos se utilizan para construir una consulta SQL de inserción, que se ejecuta en la base de datos.

El programa utiliza pausas en la ejecución para permitir al usuario leer la información mostrada y proporcionar una experiencia interactiva.

Se ha realizado un esfuerzo para mejorar la estructura del código, agregar comentarios descriptivos y utilizar nombres de variables significativos para mejorar la comprensión y el mantenimiento del código.

Este es el primer lanzamiento del programa de gestión de personal para la ferretería, y se espera seguir mejorando y agregando nuevas características en versiones futuras.

Dependencias del software

El código que hemos codificado esta escrito en C++. Por lo tanto, se necesita un compilador de C++ compatible, como GCC (GNU Compiler Collection) o MinGW (Minimalist GNU for Windows), para compilar y ejecutar el código.

Biblioteca de MySQL Connector/C++: Para interactuar con la base de datos MySQL, se requiere la biblioteca de MySQL Connector/C++. Esta biblioteca proporciona las funciones y herramientas necesarias para establecer una conexión con la base de datos, enviar consultas SQL y recuperar los resultados.

MySQL Server: Para almacenar los datos del sistema de gestión de personal, se necesita un servidor de bases de datos MySQL. Debes asegurarte de tener MySQL Server instalado y configurado correctamente en tu sistema. Puedes descargarlo desde el sitio web oficial de MySQL y seguir las instrucciones de instalación específicas para tu sistema operativo.

Entorno de desarrollo integrado (IDE): Aunque no es estrictamente necesario, un IDE puede facilitar el desarrollo y la depuración del código. Algunos IDE populares para programar en C++ incluyen Visual Studio Code, Code::Blocks, Dev-C++, Eclipse CDT y Xcode (para usuarios de macOS).

Es importante asegurarse de tener todas las dependencias y herramientas necesarias instaladas correctamente en tu sistema para poder compilar y ejecutar el código de manera adecuada. Además, es recomendable tener conocimientos básicos de C++ y MySQL para comprender y modificar el código según sea necesario de los dueños de la ferreteria.

Recomendaciones y consejos

Lee el código línea por línea: Toma el tiempo para leer el código línea por línea, prestando atención a los detalles. Esto te ayudará a entender cómo se estructura y cómo fluye la lógica del programa.

Comprende la lógica y los algoritmos utilizados: Analiza la lógica y los algoritmos empleados en el código. Identifica las estructuras de control, como bucles y condicionales, y comprende cómo se utilizan para lograr el comportamiento deseado.

Familiarízate con los nombres de variables y funciones: Presta atención a los nombres de las variables y las funciones utilizadas en el código. Los nombres descriptivos y significativos pueden darte pistas sobre su propósito y funcionalidad.

Analiza las estructuras de datos utilizadas: Observa las estructuras de datos, como matrices o estructuras, que se utilizan para almacenar y manipular la información. Comprende cómo se utilizan estas estructuras y cómo interactúan entre sí.

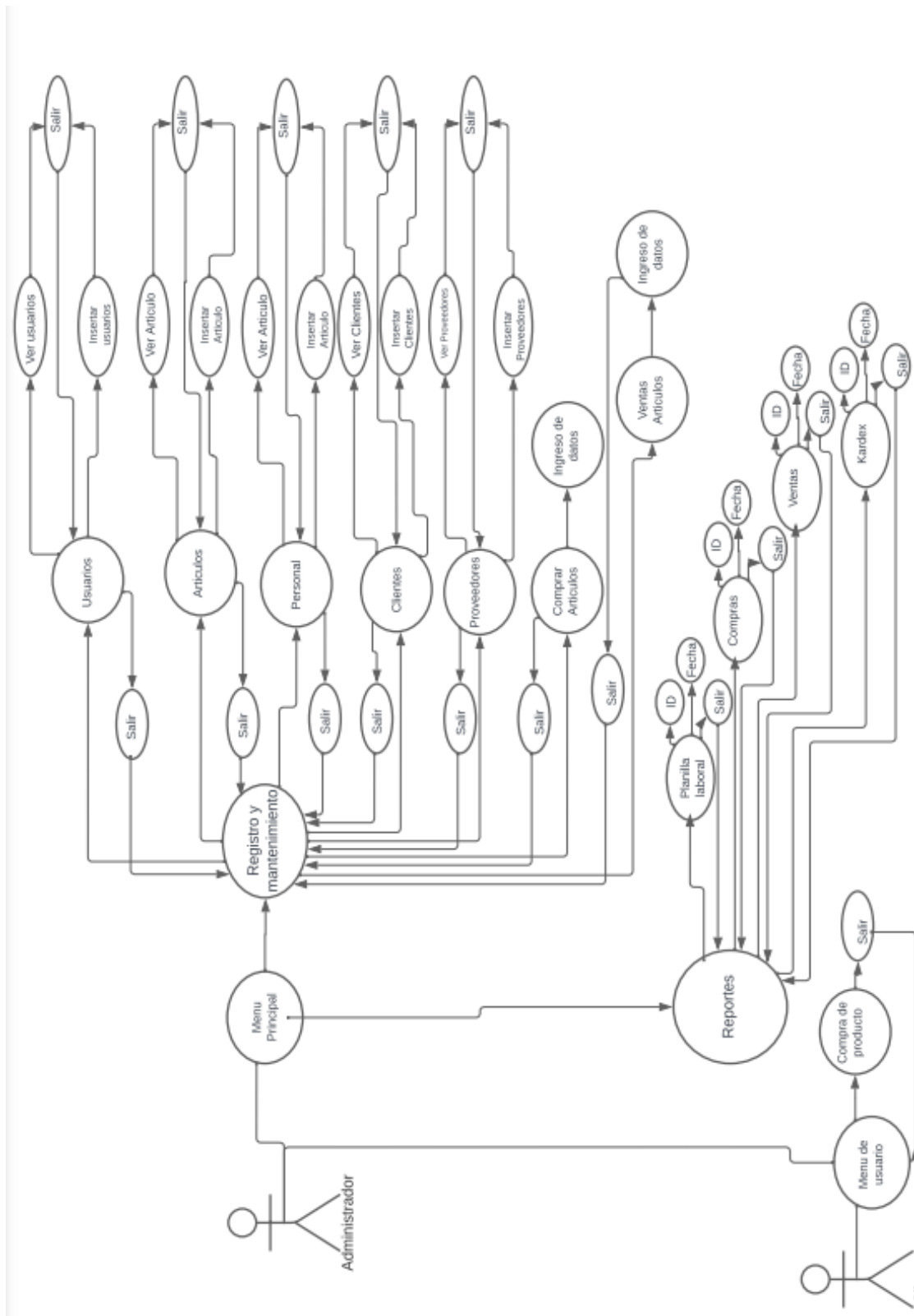
Consulta la documentación relevante: Si el código se basa en bibliotecas o frameworks específicos, consulta la documentación correspondiente. Esto te proporcionará información adicional sobre las funciones, métodos y clases utilizadas, así como ejemplos de uso.

Realiza pruebas y depuración: Experimenta con el código y realiza pruebas para observar cómo se comporta en diferentes situaciones. Utiliza técnicas de depuración para identificar posibles errores y entender cómo se están procesando los datos en cada etapa.

Busca recursos y tutoriales adicionales: Si encuentras dificultades para entender alguna parte del código, busca recursos adicionales, como tutoriales, artículos o ejemplos relacionados. Estos recursos pueden brindarte una perspectiva adicional y explicaciones más detalladas sobre los conceptos utilizados.

Comunícate con el autor del código: Si tienes acceso al autor original del código, no dudes en comunicarte con él para obtener aclaraciones o hacer preguntas específicas sobre el código. El autor puede proporcionarte información valiosa y explicaciones adicionales sobre su implementación.

Diagrama de caso de uso



El diagrama de casos de uso describe las principales funcionalidades del programa de gestión de ventas de una ferretería. El sistema permite a los usuarios,

tanto empleados como clientes, realizar diferentes acciones relacionadas con la compra y venta de productos.

Actores:

Empleado: Representa a los empleados de la ferretería.

Cliente: Representa a los clientes que realizan compras en la ferretería.

Casos de uso:

Registrar cliente: Permite al empleado registrar los datos de un nuevo cliente en el sistema.

Realizar venta: Permite al empleado registrar una venta realizada por un cliente, seleccionando los productos y generando el total de la compra.

Gestionar inventario: Permite al empleado gestionar el inventario de productos de la ferretería, incluyendo la adición, modificación o eliminación de productos.

Consultar productos: Permite al cliente y al empleado buscar y consultar información sobre los productos disponibles en la ferretería.

Generar factura: Permite al empleado generar una factura detallada de una venta realizada para entregársela al cliente.

Realizar pago: Permite al cliente realizar el pago de una compra utilizando diferentes métodos de pago disponibles.

Administrar cuentas de empleados: Permite al administrador del sistema gestionar las cuentas de los empleados, incluyendo la creación, modificación o eliminación de cuentas.

Relaciones:

El caso de uso "Realizar venta" está asociado con los casos de uso "Consultar productos", "Generar factura" y "Realizar pago".

El caso de uso "Gestionar inventario" está asociado con los casos de uso "Consultar productos".

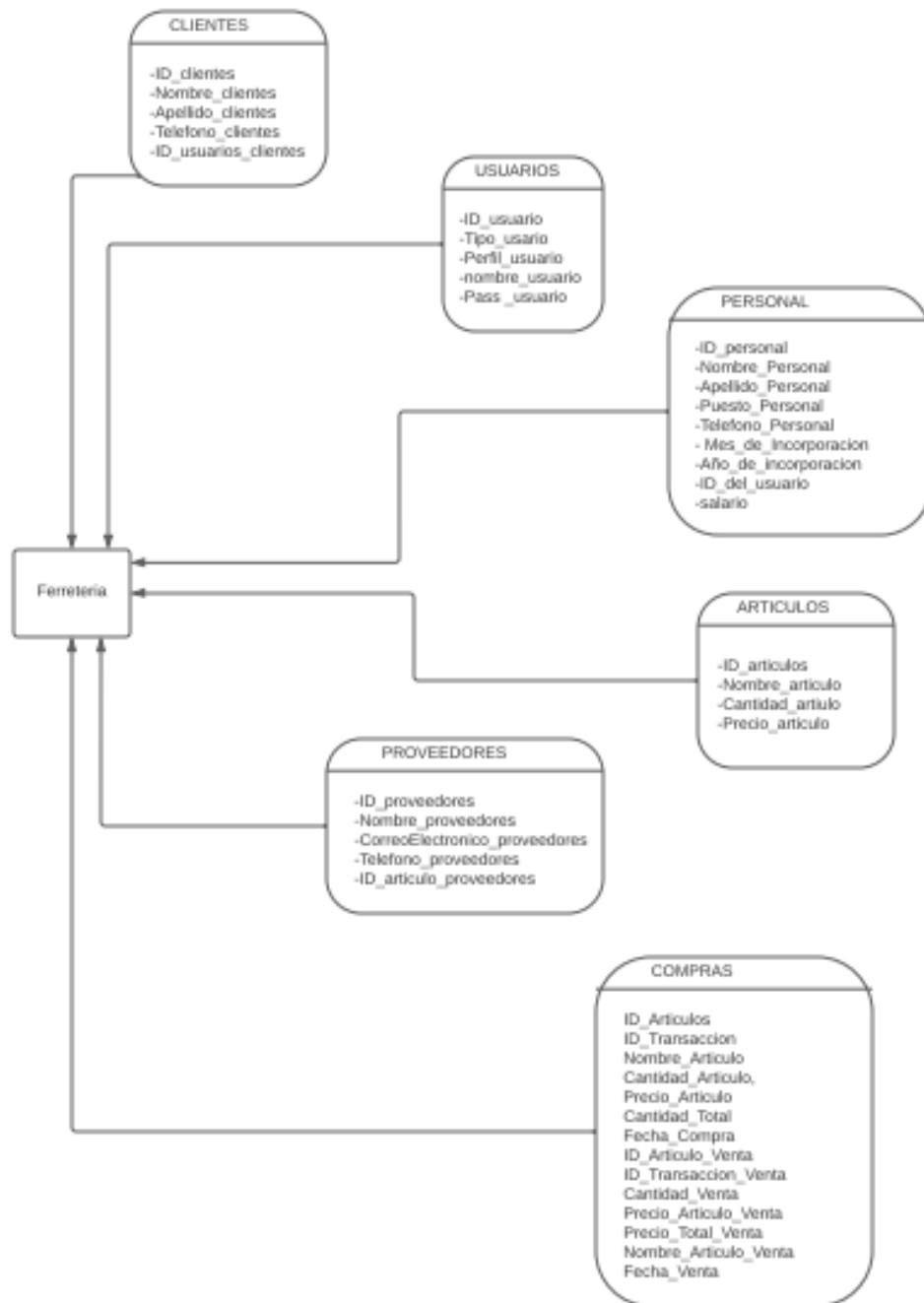
El caso de uso "Registrar cliente" está asociado con el caso de uso "Realizar venta".

El caso de uso "Administrar cuentas de empleados" está asociado con el caso de uso "Gestionar inventario".

Esta síntesis del diagrama de casos de uso proporciona una visión general de las funcionalidades clave del programa de gestión de ventas para una ferretería. Recuerda que esta es solo una representación resumida y que pueden existir más casos de uso o detalles específicos según los requerimientos del sistema.

Diagrama de objetos

DIAGRAMA DE OBJETOS



El software de una ferretería es una aplicación diseñada específicamente para ayudar en la gestión y operación de una ferretería. Proporciona herramientas y funciones que facilitan diversas tareas, como la gestión de inventario, ventas, compras, facturación, control de clientes, entre otras.

Algunas características y funcionalidades comunes en el software de una ferretería incluyen: Gestión de inventario: Permite llevar un registro detallado de los productos disponibles en la ferretería, incluyendo información como descripción, precio, proveedor, cantidad en stock, entre otros. Además, facilita la gestión de las reposiciones de stock y la actualización de los niveles de inventario.

Punto de venta (POS): Proporciona una interfaz para realizar transacciones de ventas. Permite escanear o buscar productos, agregarlos al carrito de compras, calcular el total de la venta, aplicar descuentos si es necesario y generar facturas o tickets de venta. También puede integrarse con sistemas de pago, como terminales de tarjetas de crédito o débito.

Administración de clientes: Permite registrar y gestionar la información de los clientes, incluyendo sus datos de contacto, historial de compras y preferencias. Esto facilita la atención personalizada y el seguimiento de las necesidades de los clientes.

Compras y proveedores: Permite realizar el seguimiento de las compras realizadas a los proveedores, registrar las órdenes de compra, gestionar los pagos y mantener un historial de las transacciones. Además, puede ofrecer funciones de comparación de precios entre proveedores para ayudar en la toma de decisiones de compra.

Control financiero: Ayuda a mantener un registro de los ingresos y gastos de la ferretería, generando informes financieros como balances y estados de cuenta. También puede integrarse con sistemas de contabilidad para facilitar la gestión financiera.

Informes y análisis: Proporciona herramientas para generar informes y análisis sobre diferentes aspectos del negocio, como las ventas, el rendimiento de productos, el comportamiento de los clientes, entre otros. Estos informes ayudan a tomar decisiones estratégicas y mejorar la eficiencia operativa.

Diagrama de clases

DIAGRAMA DE CLASES



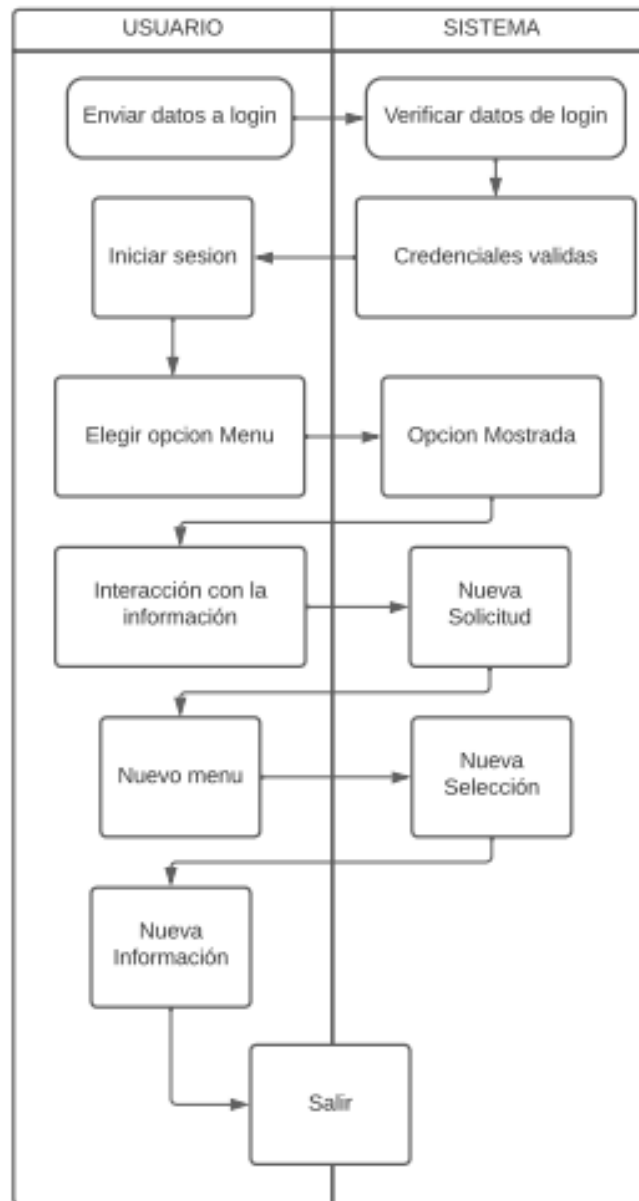
Diagrama de clases

Este diagrama de clases muestra las clases principales relacionadas con el software de una ferretería. La clase "Ferretería" representa la información general de la ferretería, como su nombre, dirección y teléfono. La clase "Producto" tiene atributos como código, descripción, precio y stock para llevar un registro de los productos disponibles. La clase "Cliente" representa la información de los clientes, como nombre, dirección y teléfono. Finalmente, la clase "Venta" contiene atributos como la fecha de la venta, el total de la venta y los productos vendidos.

Este diagrama de clases proporciona una visión general de las principales clases y sus atributos en el contexto de un software de ferretería. Recuerda que esto es solo una representación resumida y que pueden existir más clases y relaciones según los requerimientos específicos del sistema.

Diagrama de secuencia

Diagrama de secuencias



En este diagrama de secuencia, se muestra la interacción entre el Empleado y el Cliente durante el proceso de venta. Aquí están las principales etapas del proceso:

El Empleado inicia el proceso de venta y envía el subtotal de la compra al Cliente.

El Cliente calcula el descuento (si aplica) y envía la información de descuento al Empleado.

El Empleado genera la factura basada en el subtotal y el descuento proporcionados por el Cliente.

El Empleado solicita el pago al Cliente y recibe la información de pago.

Después de recibir el pago, el Empleado actualiza el stock de productos.

El Empleado finaliza la venta y se lo comunica al Cliente.

Este diagrama de secuencia muestra la secuencia de interacciones entre el Empleado y el Cliente durante el proceso de venta en una ferretería. Recuerda que este es solo un ejemplo y que las interacciones y los mensajes pueden variar según los requisitos y la lógica de negocio específica del sistema