

## Introducción

El presente proyecto tiene como objetivo el desarrollo de un sistema de gestión integral de inventario, ventas, compras y movimientos de stock. Este sistema, implementado en **Java**, está diseñado para proporcionar una visualización clara y detallada de las operaciones diarias de una empresa, mediante gráficos interactivos y herramientas de filtrado que permiten analizar los datos por producto, categoría, proveedor o rango de fechas. La meta principal es mejorar la eficiencia en la toma de decisiones y la planificación logística a través de una interfaz de usuario intuitiva y funcional.

### Objetivos Técnicos:

- Gestión de Inventarios.
- Visualización de Graficas y Estadísticas.
- Historial.
- Control de Stock.
- Control de Ventas y Compras.

### Alcances del proyecto

- Control de inventario: Permite la gestión de productos, Categorías, Proveedores y Cantidades.
- Control de Ventas y Compras: Registros sobre las ventas y compras que se realizan en la empresa.
- Reporte de Graficas y Estadísticas: Permite ver graficas de las estadísticas de ventas y compras de productos.

## Algoritmo de toda la solución:

### Categorías

❓ Inicio

❓ Mostrar Categorías Existentes

- Subalgoritmo: MostrarCategorias
  1. Imprimir "Categorías en existencia:"
  2. Imprimir las categorías:
    - Sodas carbonatadas
    - Sodas
    - Energizantes

❓ Leer Opción del Usuario

- Imprimir las opciones disponibles:
  - 1. Agregar Categoría
  - 2. Modificar una Categoría
  -

### 3. Eliminar Categoría

- Leer opción.

#### ? Decisión

- Si opción es 1, proceder a agregar categoría.
- Si opción es 2, proceder a modificar categoría.
- Si opción es 3, proceder a eliminar categoría.
- Si no, imprimir "Ingrese una opción válida."

#### ? Agregar Categoría

- Subalgoritmo: AgregarCategoría
  1. Abrir archivo\_texto.txt en modo agregar.
  2. Escribir "Nombre|Descripción".
  3. Cerrar el archivo.

#### ? Modificar Categoría

- Subalgoritmo: ModificarCategoría
  1. Abrir archivo\_texto.txt en modo escritura.
  2. Escribir las líneas modificadas para cada categoría:
    - Sodas carbonatadas|Descripción|
    - Sodas|Descripción|
    - Energizantes|Descripción|

3. Cerrar el archivo.

#### ? Eliminar Categoría

- Subalgoritmo: EliminarCategoría

1. Abrir archivo\_texto.txt en modo lectura.
2. Crear archivo\_texto\_copia.txt en modo escritura.
3. Leer cada línea de archivo\_texto.txt.
  - Subalgoritmo: FiltrarCategoría
    1. Si la línea no contiene "Categoría" a eliminar:
      - Escribir la línea en archivo\_texto\_copia.txt.
4. Cerrar ambos archivos.
5. Sobrescribir archivo\_texto.txt con archivo\_texto\_copia.txt.

#### ? Fin

### **Características**

#### ? Inicio

#### ? Mostrar Características de Bebidas

- Subalgoritmo: MostrarCaracteristicas

1. Imprimir "Las características de cada bebida son:"
2. Imprimir cada bebida y su característica:
  -

1. Coca-Cola Original: alta en azúcar

▪

2. Coca-Cola Zero: sin azúcar

▪

3. Coca-Cola Light: poca azúcar

▪

4. Sprite: sabor a limón

▪

5. Powerade: bebida hidratante

▪

6. Monster Energy: bebida energética

 Leer Opción del Usuario

- Imprimir las opciones disponibles:

○

1. Agregar Especificación

○

2. Modificar una Especificación

○

3. Eliminar una Especificación

- Leer opcion.

#### ? Decisión

- Si opcion es 1, proceder a agregar especificación.
- Si opcion es 2, proceder a modificar especificación.
- Si opcion es 3, proceder a eliminar especificación.
- Si no, imprimir "Ingrese una opción válida."

#### ? Agregar Especificación

- Subalgoritmo: AgregarEspecificacion
  1. Abrir archivo\_texto.txt en modo agregar.
  2. Escribir "Nombre Producto|Nueva Característica".
  3. Cerrar el archivo.

#### ? Modificar Especificación

- Subalgoritmo: ModificarEspecificacion
  1. Abrir archivo\_texto.txt en modo escritura.
  2. Escribir las líneas modificadas para cada bebida:
    - Coca-Cola Original|Modificar|
    - Coca-Cola Zero|Modificar|
    - Coca-Cola Light|Modificar|
    - Sprite|Modificar|

- Powerade|Modificar|
- Monster Energy|Modificar|

3. Cerrar el archivo.

#### ? Eliminar Especificación

- Subalgoritmo: EliminarEspecificacion

1. Abrir archivo\_texto.txt en modo lectura.
2. Crear archivo\_texto\_copia.txt en modo escritura.
3. Leer cada línea de archivo\_texto.txt.
  - Subalgoritmo: FiltrarCaracteristica
    1. Si la línea no contiene "Característica" a eliminar:
      - Escribir la línea en archivo\_texto\_copia.txt.
4. Cerrar ambos archivos.
5. Sobrescribir archivo\_texto.txt con archivo\_texto\_copia.txt.

#### ? Fin

### **Especificaciones**

#### ? Inicio

#### ? Mostrar Especificaciones de Bebidas

- Subalgoritmo: MostrarEspecificaciones
  1. Imprimir "Las especificaciones de cada bebida son:"

2. Imprimir las especificaciones de cada bebida:

- 1. Coca-Cola Original: Agua carbonatada, azúcar, colorante E-150d, acidulante E-338, aromas naturales y cafeína.
- 2. Coca-Cola Zero: Agua carbonatada, edulcorantes, colorante E-150d, acidulante E-338, aromas naturales, cafeína.
- 3. Coca-Cola Light: Agua carbonatada, azúcar, extracto de hoja de stevia, colorante E-150d, acidulante E-338, aromas naturales, cafeína.
- 4. Sprite: Agua carbonatada, azúcar, ácido cítrico, aromas naturales de limón y lima, conservante.
- 5. Powerade: Agua, azúcares añadidos, ácido cítrico, citrato de potasio, cloruro de magnesio, cloruro de calcio, vitaminas B3, B6 y B12, colorantes.
- 6. Monster Energy: Agua carbonatada, sacarosa, glucosa, taurina, citrato de sodio, extracto de raíz de Panax.

? Leer Opción del Usuario

- Imprimir las opciones disponibles:



- 

1. Agregar Especificación

- 

2. Modificar una Especificación

- 

3. Eliminar una Especificación

- Leer opcion.

#### ? Decisión

- Si opcion es 1, proceder a agregar especificación.
- Si opcion es 2, proceder a modificar especificación.
- Si opcion es 3, proceder a eliminar especificación.
- Si no, imprimir "Ingrese una opción válida."

#### ? Agregar Especificación

- Subalgoritmo: AgregarEspecificacion
  1. Abrir archivo\_texto.txt en modo agregar.
  2. Escribir "Nombre Producto|Nueva Especificación".
  3. Cerrar el archivo.

#### ? Modificar Especificación

- Subalgoritmo: ModificarEspecificacion

1. Abrir archivo\_texto.txt en modo escritura.
2. Escribir las líneas modificadas para cada bebida:
  - Coca-Cola Original|Modificar|
  - Coca-Cola Zero|Modificar|
  - Coca-Cola Light|Modificar|
  - Sprite|Modificar|
  - Powerade|Modificar|
  - Monster Energy|Modificar|
3. Cerrar el archivo.

#### ? Eliminar Especificación

- Subalgoritmo: EliminarEspecificacion
  1. Abrir archivo\_texto.txt en modo lectura.
  2. Crear archivo\_texto\_copia.txt en modo escritura.
  3. Leer cada línea de archivo\_texto.txt.
    - Subalgoritmo: FiltrarEspecificacion
      1. Si la línea no contiene "Especificación" a eliminar:
        - Escribir la línea en archivo\_texto\_copia.txt.
  4. Cerrar ambos archivos.
  5. Sobrescribir archivo\_texto.txt con archivo\_texto\_copia.txt.

Fin

## Alta

Inicio

Mostrar Lista de Productos en Existencia

- Subalgoritmo: MostrarProductos
  1. Imprimir "Lista de productos en existencia:"
  2. Imprimir las categorías y productos:
    - Sodas carbonatadas:
      - Coca-Cola Original
      - Coca-Cola Zero
      - Sprite
    - Sodas:
      - Coca-Cola Light
    - Energizantes:
      - Monster Energy
      - Powerade

Leer Opción del Usuario

- Imprimir las opciones disponibles:

- 

1. Ver los registros

- 

2. Agregar registro

- Leer opcion.

#### ? Decisión

- Si opcion es 1, proceder a ver los registros.
- Si opcion es 2, proceder a agregar un registro.
- Si no, imprimir "Ingrese una opción válida."

#### ? Ver Registros

- Subalgoritmo: VerRegistros
  1. Abrir archivo\_texto.txt en modo agregar.
  2. Escribir los registros de productos:
    - Sodas carbonatadas:
      - Coca-Cola Original|Es una bebida alta en azúcar|2000
      - Coca-Cola Zero|Es una bebida sin azúcar|1000
      - Sprite|Es una bebida con sabor a limón|1000
    - Sodas:
      - Coca-Cola Light|Es una bebida con poca azúcar|500

- Energizantes:
  - Monster Energy|Es una bebida energética|1000
  - Powerade|Es una bebida hidratante|1000
- 3. Cerrar el archivo.
- 4. Imprimir "Archivo Guardado."

#### ? Agregar Registro

- Subalgoritmo: AgregarRegistro
  - 1. Abrir archivo\_texto.txt en modo escritura.
  - 2. Escribir el encabezado de los registros:
    - Categorías|Producto|Descripción|Cantidad de Stock
  - 3. Cerrar el archivo.

#### ? Fin

### **Baja**

#### ? Inicio

#### ? Mostrar Lista de Productos en Existencia

- Subalgoritmo: MostrarProductos
  - 1. Imprimir "Lista de productos en existencia:"
  - 2. Imprimir los productos:
    - Coca-Cola Original

- Coca-Cola Zero
- Sprite
- Coca-Cola Light
- Monster Energy
- Powerade

#### ? Leer Opción del Usuario

- Imprimir "Seleccione 1 para borrar"
- Leer opcion.

#### ? Decisión

- Si opcion es 1, proceder a eliminar un producto.
- Si no, imprimir "Opción no válida."

#### ? Eliminar Producto

- Subalgoritmo: EliminarProducto
  1. Abrir archivo\_texto.txt en modo lectura.
  2. Crear un archivo temporal archivo\_texto\_copia.txt en modo escritura.
  3. Leer cada línea del archivo original:
    - Si la línea no corresponde al producto a eliminar:
      - Escribir la línea en el archivo temporal.
  4. Cerrar ambos archivos.

5. Reemplazar el archivo original con el archivo temporal.

6. Imprimir "Producto eliminado."

❓ Fin

### **Asignación**

❓ Inicio

❓ Mostrar Opciones al Usuario

- Subalgoritmo: MostrarOpciones
  1. Imprimir "Ingrese el número 1 si desea crear un nuevo producto:"
  2. Imprimir "Ingrese el número 2 si desea agregar algún producto:"
  3. Imprimir "Ingrese el número 3 si desea cambiar algún producto:"

❓ Leer Opción del Usuario

- Leer opcion.

❓ Decisión

- Si opcion es 1, proceder a crear un nuevo producto.
- Si opcion es 2, proceder a agregar un producto.
- Si opcion es 3, proceder a cambiar un producto.
- Si no, imprimir "Opción no válida."

❓ Crear Nuevo Producto

- Subalgoritmo: CrearNuevoProducto

1. Abrir archivo\_texto.txt en modo escritura.
2. Escribir los nombres de los productos en el archivo:
  - Coca-Cola Original
  - Coca-Cola Zero
  - Coca-Cola Light
  - Sprite
  - Powerade
  - Monster Energy
3. Cerrar el archivo.

#### ? Agregar Producto

- Subalgoritmo: AgregarProducto
  1. Abrir archivo\_texto.txt en modo escritura (agregar).
  2. Escribir "Agregar producto" en el archivo.
  3. Cerrar el archivo.

#### ? Cambiar Producto

- Subalgoritmo: CambiarProducto
  1. Abrir archivo\_texto.txt en modo lectura.
  2. Crear un archivo temporal archivo\_texto\_copia.txt en modo escritura.
  3. Leer cada línea del archivo original:



- Si la línea corresponde al producto a cambiar, no escribirla.
  - Si no, escribir la línea en el archivo temporal.
4. Cerrar ambos archivos.
  5. Reemplazar el archivo original con el archivo temporal.
  6. Imprimir "Producto cambiado."

? Fin

## **Modificación**

? Inicio

? Mostrar Productos Existentes

- Subalgoritmo: MostrarProductos
  1. Imprimir "Lista de productos en existencia".
  2. Imprimir detalles de cada producto (nombre y descripción).

? Crear o Abrir Archivo

- Declarar el archivo archivo\_texto.txt.

? Mostrar Opciones al Usuario

- Subalgoritmo: MostrarOpciones
  1. Imprimir "Seleccione 1 para ver las modificaciones".
  2. Imprimir "Seleccione 2 para Modificar".

? Leer Opción del Usuario

- Leer opcion.

#### ? Decisión

- Si opcion es 1, proceder a ver las modificaciones.
- Si opcion es 2, proceder a modificar productos.
- Si no, imprimir "Opción no válida."

#### ? Ver Modificaciones

- Subalgoritmo: VerModificaciones
  1. Abrir archivo\_texto.txt en modo escritura.
  2. Escribir los detalles de cada producto en el archivo.
  3. Cerrar el archivo.
  4. Imprimir "Modificaciones guardadas."

#### ? Modificar Productos

- Subalgoritmo: ModificarProducto
  1. Abrir archivo\_texto.txt en modo agregar.
  2. Escribir "Producto|Descripción|Cantidad de stock" en el archivo.
  3. Cerrar el archivo.
  4. Imprimir "Modificación realizada."

#### ? Fin

## Registro de entrada

### ? Inicio

### ? Mostrar Códigos de Productos

- Subalgoritmo: MostrarCodigos
  1. Imprimir "Los códigos de cada producto son:".
  2. Imprimir los códigos correspondientes a cada producto:
    - CocaColaOriginal = 123
    - CocaColaZero = 456
    - Sprite = 789
    - CocaColaLight = 1011
    - monsterEnergy = 1213

### ? Mostrar Opciones al Usuario

- Subalgoritmo: MostrarOpciones
  1. Imprimir "Seleccione una de las opciones:".
  2. Imprimir "1. Actualizar la cantidad de Stock".

### ? Leer Opción del Usuario

- Leer opcion.

### ? Decisión

- Si opcion es 1, proceder a actualizar la cantidad de stock.

- Si no, imprimir "Opción no válida".

#### ? Actualizar Cantidad de Stock

- Subalgoritmo: ActualizarStock
  1. Abrir archivo\_texto.txt en modo lectura.
  2. Crear un archivo copia archivo\_texto\_copia.txt.
  3. Leer línea por línea del archivo original.
  4. Si la línea contiene "Stock", actualizar la línea con el nuevo stock:
    - Escribir "Soda Carbonatada|CocaColaOriginal|2000" en el archivo copia.
  5. Escribir otras líneas sin cambios en el archivo copia.
  6. Cerrar ambos archivos.
  7. Reemplazar el archivo original con el archivo copia.
  8. Imprimir "Stock actualizado".

#### ? Fin

### **Registro Salida**

#### ? Inicio

#### ? Inicializar Scanner y Mapas

- Crear un Scanner para la entrada del usuario.
- Crear un mapa productos que asocie códigos de productos con sus nombres.
- Crear un mapa stock que asocie códigos de productos con sus cantidades en stock.

#### ? Solicitar Código del Producto

- Subalgoritmo: SolicitarCodigoProducto
  1. Imprimir "Ingrese código del producto:".
  2. Leer el codigo del producto desde la entrada del usuario.

#### ? Verificar Código del Producto

- Si codigo está en productos, proceder al siguiente paso.
- Si no, imprimir "Código de producto no válido." y terminar.

#### ? Mostrar Información del Producto

- Subalgoritmo: MostrarInformacionProducto
  1. Imprimir "Producto: " junto con el nombre del producto.
  2. Imprimir "Cantidad de stock disponible: " junto con la cantidad en stock.

#### ? Solicitar Cantidad a Vender

- Subalgoritmo: SolicitarCantidadVenta
  1. Imprimir "Ingrese la cantidad a vender:".
  2. Leer la cantidad desde la entrada del usuario.

#### ? Verificar Stock Suficiente

- Si cantidad es menor o igual a stock.get(codigo), proceder al siguiente paso.
- Si no, imprimir "No hay suficiente stock para esa cantidad." y terminar.

#### ? Registrar Venta

- Subalgoritmo: RegistrarVenta

1. Imprimir "Ingrese la fecha de la venta (formato YYYY-MM-DD):".
2. Leer la fecha de la venta.
3. Actualizar stock restando cantidad de stock.get(codigo).
4. Imprimir "Producto vendido con éxito. Quedan " junto con la nueva cantidad de stock.
5. (Opcional) Llamar a guardarRegistroVenta para registrar la venta en un archivo.

❓ Fin

## **Alertas**

❓ Inicio

❓ Inicializar Scanner y Variables

- Crear un Scanner para la entrada del usuario.
- Definir los códigos de productos.
- Inicializar variables punto, cantidad, y codigo.

❓ Solicitar Código del Producto

- Subalgoritmo: SolicitarCodigoProducto
  1. Imprimir "ingresa el codigo del producto".
  2. Leer el codigo desde la entrada del usuario.

❓ Verificar Producto y Generar Alerta

- Subalgoritmo: VerificarProducto

1. Si codigo es igual a 123, llamar a GestionarProducto con el nombre "CocaColaOriginal" y su stock.
2. Si codigo es igual a 456, llamar a GestionarProducto con el nombre "CocaColaZero" y su stock.
3. Si codigo es igual a 789, llamar a GestionarProducto con el nombre "Sprite" y su stock.
4. Si codigo es igual a 1011, llamar a GestionarProducto con el nombre "CocaColaLight" y su stock.
5. Si codigo es igual a 1213, llamar a GestionarProducto con el nombre "MonsterEnergy" y su stock.
6. Si codigo es igual a 1415, llamar a GestionarProducto con el nombre "Powerade" y su stock.
7. Si codigo no coincide con ninguno, imprimir "Código de producto no válido."

❓ Fin

## Historial

❓ Inicio

❓ Inicializar Scanner y Variables

- Crear un Scanner para la entrada del usuario.
- Definir la variable usuario para capturar el rol en la empresa.
- Inicializar la variable producto para seleccionar un producto.

❓ Solicitar Rol de Usuario

- Subalgoritmo: SolicitarRolUsuario
  1. Imprimir "Ingrese su rol en la empresa".
  2. Leer el rol del usuario.

❓ Mostrar Opciones de Productos

- Subalgoritmo: MostrarProductos

1. Imprimir lista de productos disponibles (CocaColaOriginal, CocaColaZero, etc.).

#### ? Seleccionar Producto

- Subalgoritmo: SeleccionarProducto

1. Leer el número del producto.

2. Llamar a MostrarDetallesProducto con el número del producto seleccionado y el rol del usuario.

#### ? Fin

### **Creación de Pedidos de Compra**

#### ? Inicio

#### ? Inicializar Scanner y Variables

- Crear un Scanner para la entrada del usuario.
- Definir la variable opcion para seleccionar la acción (crear pedido o agregar pedido).

#### ? Mostrar Productos en Existencia

- Subalgoritmo: MostrarProductos

1. Imprimir lista de productos disponibles (CocaColaOriginal, CocaColaZero, etc.).

#### ? Solicitar Opción

- Subalgoritmo: SolicitarOpcion

1. Imprimir opciones para crear o agregar un pedido.

2. Leer la opción seleccionada por el usuario.



## ? Ejecutar Acción según Opción Seleccionada

- Subalgoritmo: EjecutarAccion

1. Usar una estructura de control (switch) para determinar qué acción tomar según la opción elegida.

2. Si opción es 1 (Crear Pedido)

- Subalgoritmo: CrearPedido

1. Crear y escribir en un archivo con información del pedido.

2. Calcular total.

3. Si opción es 2 (Agregar Pedido)

- Subalgoritmo: AgregarPedido

1. Abrir el archivo en modo de adición.

2. Escribir en el archivo información del nuevo pedido.

## ? Fin

### **Gestión de Pedidos de Compra:**

## ? Inicio

## ? Inicializar Variables

- Crear un Scanner para la entrada del usuario.

## ? Capturar Datos del Usuario

- Subalgoritmo: CapturarDatos

1. Imprimir "Ingresa el precio:" y leer el precio.

2. Imprimir "Ingresa la cantidad:" y leer la cantidad.

3. Imprimir "Seleccione el número del producto del 1 al 6:" y leer el número del producto.

? Generar Fechas de Creación y Entrega

- Generar valores aleatorios para fechaDeCreacion y fechaEntrega.

? Calcular el Total

- Subalgoritmo: CalcularTotal

1.  $\text{Total} = \text{cantidad} * \text{precio}$ .

? Mostrar Información del Producto Seleccionado

- Subalgoritmo: MostrarInformacionProducto

1. Según el producto seleccionado, imprimir la información correspondiente.

2. Preguntar si desea cancelar el pedido.

3. Subalgoritmo: ProcesarCancelacion

- Leer la respuesta del usuario y mostrar el mensaje correspondiente.

? Fin

## **Recepción de Pedidos de Compra**

? Inicio

? Inicializar Variables

- Crear un Scanner para la entrada del usuario.

? Capturar Datos del Usuario

- Subalgoritmo: CapturarDatos

1. Imprimir "Seleccione el número del producto del 1 al 6:" y leer el número del producto.
2. Imprimir "Ingresa el precio:" y leer el precio.
3. Imprimir "Ingresa la cantidad:" y leer la cantidad.

❓ Generar Fechas de Creación y Entrega

- Generar valores aleatorios para fechaDeCreacion y fechaEntrega.

❓ Calcular el Total

- Subalgoritmo: CalcularTotal

1.  $\text{Total} = \text{cantidad} * \text{precio}.$

❓ Mostrar Información del Producto Seleccionado

- Subalgoritmo: MostrarInformacionProducto

1. Según el producto seleccionado, imprimir la información correspondiente.
2. Mostrar detalles del pedido (fecha de creación, fecha de entrega, cantidad, total).
3. Preguntar si desea cancelar el pedido.
4. Subalgoritmo: ProcesarCancelacion
  - Leer la respuesta del usuario y mostrar el mensaje correspondiente.

❓ Fin

**Generación de Informes de Inventario Actual**

❓ Inicio

#### ? Inicializar Variables

- Crear un Scanner para la entrada del usuario.
- Definir un objeto File para el archivo "archivo.CSV".
- Inicializar opcion en 0.

#### ? Mostrar Catálogo de Productos

- Subalgoritmo: MostrarCatalogoProductos
  1. Imprimir "Catálogo de los productos".
  2. Imprimir productos y sus detalles (nombre, stock y valor).

#### ? Solicitar Opción al Usuario

- Imprimir opciones:
  - Opción 1: Ver registros.
  - Opción 2: Agregar registro.

#### ? Leer Opción del Usuario

- Subalgoritmo: LeerOpcion
  1. Leer la opción elegida por el usuario.

#### ? Ejecutar Acción según Opción Elegida

- Subalgoritmo: EjecutarOpcion
  1. Si la opción es 1:
    - Subalgoritmo: GuardarRegistros

- Abrir archivo en modo de escritura.
- Escribir los registros de productos.

2. Si la opción es 2:

- Subalgoritmo: AgregarRegistro
  - Abrir archivo en modo de adición.
  - Agregar nueva línea con encabezados.

Fin

### **Generación de Informes de Movimientos de Stock**

Inicio

Inicializar Variables

- Crear un Scanner para la entrada del usuario.
- Definir un objeto File para el archivo "archivo.CSV".
- Inicializar opcion en 0.

Mostrar Opciones al Usuario

- Imprimir opciones:
  - Opción 1: Ver movimientos.
  - Opción 2: Agregar movimientos.

Leer Opción del Usuario

- Subalgoritmo: LeerOpcion

1. Leer la opción elegida por el usuario.

❓ Ejecutar Acción según Opción Elegida

- Subalgoritmo: EjecutarOpcion

1. Si la opción es 1:

- Subalgoritmo: GuardarMovimientos
  - Abrir archivo en modo de escritura.
  - Escribir los registros de movimientos.

2. Si la opción es 2:

- Subalgoritmo: AgregarMovimiento
  - Abrir archivo en modo de adición.
  - Agregar nueva línea con encabezados.

❓ Fin

## **Generación de Informes de Compras**

❓ Inicio

❓ Inicializar Variables

- Crear un Scanner para la entrada del usuario.
- Definir un objeto File para el archivo "archivo\_texto.txt".
- Inicializar opcion en 0.

❓ Mostrar Opciones al Usuario

- Imprimir opciones:
  - Opción 1: Ver las compras realizadas.
  - Opción 2: Agregar una nueva compra.

#### ? Leer Opción del Usuario

- Subalgoritmo: LeerOpcion
  - 1. Leer la opción elegida por el usuario.

#### ? Ejecutar Acción según Opción Elegida

- Subalgoritmo: EjecutarOpcion
  - 1. Si la opción es 1:
    - Subalgoritmo: GuardarCompras
      - Abrir archivo en modo de escritura.
      - Escribir los registros de compras.
  - 2. Si la opción es 2:
    - Subalgoritmo: AgregarCompra
      - Abrir archivo en modo de adición.
      - Agregar nueva línea con encabezados.

#### ? Fin ? Inicio

#### ? Inicializar Variables

- Crear un Scanner para la entrada del usuario.

- Definir un objeto File para el archivo "archivo\_texto.txt".
- Inicializar opcion en 0.

#### ? Mostrar Opciones al Usuario

- Imprimir opciones:
  - Opción 1: Ver las compras realizadas.
  - Opción 2: Agregar una nueva compra.

#### ? Leer Opción del Usuario

- Subalgoritmo: LeerOpcion
  - 1. Leer la opción elegida por el usuario.

#### ? Ejecutar Acción según Opción Elegida

- Subalgoritmo: EjecutarOpcion
  - 1. Si la opción es 1:
    - Subalgoritmo: GuardarCompras
      - Abrir archivo en modo de escritura.
      - Escribir los registros de compras.
  - 2. Si la opción es 2:
    - Subalgoritmo: AgregarCompra
      - Abrir archivo en modo de adición.
      - Agregar nueva línea con encabezados.



## ? Fin ? Inicio

### ? Inicializar Variables

- Crear un Scanner para la entrada del usuario.
- Definir un objeto File para el archivo "archivo\_texto.txt".
- Inicializar opcion en 0.

### ? Mostrar Opciones al Usuario

- Imprimir opciones:
  - Opción 1: Ver las compras realizadas.
  - Opción 2: Agregar una nueva compra.

### ? Leer Opción del Usuario

- Subalgoritmo: LeerOpcion
  1. Leer la opción elegida por el usuario.

### ? Ejecutar Acción según Opción Elegida

- Subalgoritmo: EjecutarOpcion
  1. Si la opción es 1:
    - Subalgoritmo: GuardarCompras
      - Abrir archivo en modo de escritura.
      - Escribir los registros de compras.
  2. Si la opción es 2:

- Subalgoritmo: AgregarCompra
  - Abrir archivo en modo de adición.
  - Agregar nueva línea con encabezados.

❓ Fin

## **Visualización de Gráficas y Estadísticas**

❓ Inicio

❓ Inicializar Variables

- Crear un Scanner para la entrada del usuario.
- Definir un arreglo de productos y otro de cantidadesProductos.
- Definir un arreglo de categorías y otro de cantidadesCategoria.
- Definir un arreglo de fechas y otro de cantidadesFecha.

❓ Mostrar Productos en Existencia

- Subalgoritmo: MostrarProductos
  1. Imprimir la lista de productos.

❓ Agregar Datos de Productos al Dataset

- Subalgoritmo: AgregarDatosProductos
  1. Para cada producto, agregar su cantidad al dataset.

❓ Agregar Datos de Categorías al Dataset

- Subalgoritmo: AgregarDatosCategorias

1. Para cada categoría, agregar su cantidad al dataset.

#### ? Agregar Datos de Fechas al Dataset

- Subalgoritmo: AgregarDatosFechas

1. Para cada fecha, agregar su cantidad al dataset.

#### ? **Fin**

### **Seguridad**

#### ? Inicio

#### ? Inicializar Variables

- Crear un Scanner para la entrada del usuario.
- Definir las contraseñas para cada puesto (jefe, administrador, vendedor).

#### ? Solicitar puesto del usuario

- Subalgoritmo: SolicitarPuesto

1. Imprimir opciones de puesto.

#### ? Leer opción ingresada

- Subalgoritmo: LeerOpcion

1. Leer el número ingresado por el usuario.

#### ? Verificar acceso según opción

- Subalgoritmo: VerificarAcceso

1. Si la opción es 1 (jefe), llamar a VerificarContraseña con la contraseña del jefe.
2. Si la opción es 2 (administrador), llamar a VerificarContraseña con la contraseña del administrador.
3. Si la opción es 3 (vendedor), llamar a VerificarContraseña con la contraseña del vendedor.

❓ Fin

**Diagramas de Flujo:**

**Diagramas.pdf**

## Instrucciones de configuración del entorno de trabajo:

### 1. Instalación de Herramientas Necesarias

- Asegúrate de tener instalado el JDK. Puedes descargarlo desde [Oracle](#) o usar un gestor de paquetes como apt en Ubuntu

```
sudo apt update
sudo apt install openjdk-17-jdk
```

- Verifica la instalación:

```
java -version
```

- Si tu proyecto utiliza **Maven** para la gestión de dependencias, asegúrate de instalarlo:

```
sudo apt install maven
```

### 2. Estructura del Proyecto

- Asegúrate de que tu proyecto Java tenga la siguiente estructura:

```
/sistema-ventas
  /src
    /main
      /java
        /com/miempresa/sistemaventas
          Main.java
          VentasController.java
          ProductosController.java
      /resources
        productos.csv
        ventas.csv
    /libs
    /target
    pom.xml (si usas Maven)
```

### 3. Configuración del Proyecto en el IDE

#### a. Configuración del JDK

- Abre tu IDE y configura la ruta del JDK en las preferencias del proyecto:
  - En **IntelliJ**, ve a File -> Project Structure -> Project, y selecciona la versión del JDK.

#### b. Agregar Dependencias

- Si no usas Maven, asegúrate de agregar manualmente los archivos .jar (como JFreeChart o JavaFX) a tu proyecto. En **Eclipse**, ve a:
  - Project -> Properties -> Java Build Path -> Libraries -> Add External JARs.

### 4. Conexión a Base de Datos (Opcional)

Si deseas usar una base de datos para almacenar información de productos, ventas, etc., puedes configurar un motor de base de datos como **MySQL** o **SQLite**.

- Instala MySQL en tu sistema:

```
sudo apt install mysql-server
```

### 5. Ejecución del Proyecto

Para ejecutar tu proyecto:

#### a. Desde el IDE

- Selecciona la clase Main.java y ejecuta el proyecto desde el IDE.

#### b. Desde la Línea de Comandos

Si usas Maven:

```
mvn clean install  
mvn exec:java -Dexec.mainClass="com.miempresa.sistemaventas.Main"
```

## 6. Pruebas y Depuración

Asegúrate de realizar pruebas unitarias para comprobar el correcto funcionamiento de cada parte del sistema, usando frameworks como **JUnit**:

```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <version>4.13.2</version>  
  <scope>test</scope>  
</dependency>
```



## **Tecnologías utilizadas:**

1. Lenguaje de programación: Java.
2. GitHub

## Sección de solución de problemas comunes:

### Error: ClassNotFoundException al intentar ejecutar el proyecto

asegúrate de que los archivos .jar están en la ruta de construcción (build path).

### Error de compilación: incompatible types

Verifica que el tipo de datos que intentas asignar es compatible con el tipo de la variable. Por ejemplo, si una variable es de tipo int, asegúrate de que no estás asignando un valor de tipo double sin realizar una conversión.

### Problemas con el control de versiones (Git)

Si enfrentas un conflicto en Git, puedes resolverlo manualmente editando los archivos en conflicto, y luego ejecutar:

```
git add <archivo_resuelto>
git commit
```

### Aplicación Java lenta o con problemas de rendimiento

- Optimiza las consultas a la base de datos, por ejemplo, utilizando índices en las tablas.
- Revisa tu código para evitar el uso innecesario de bucles anidados o procesamiento innecesario de grandes volúmenes de datos en el lado de la aplicación.

## Información de contacto:

**Nombre:**

Luis Alejandro Berreondo Florián

**Número telefónico:**

3399-9914

**Correo electrónico:**

[luisalejandroberreondoflorian@gmail.com](mailto:luisalejandroberreondoflorian@gmail.com)

**archivo tipo jar**

**"C:\Users\luisa\OneDrive\Escritorio\Proyecto1\dist\Proyecto1.jar"**