**Descripción de Clases**

Este informe detalla el desarrollo de un software de escritorio encargado para gestionar un inventario de prendas de vestir, implementado en Java utilizando NetBeans. El proyecto se centra en la aplicación del patrón de diseño MVC (Modelo-Vista-Controlador) para garantizar una estructura organizada y modular.

# Clase Articulo

La clase Articulo representa un artículo con las siguientes características:

**Atributos**

1. **id (int):** Identificador único del artículo.
2. **precio (int):** Precio del artículo.
3. **nombre (String):** Nombre del artículo.

**Métodos**

**Constructor**

**public Articulo(int id, int precio, String nombre)**

Este constructor permite instanciar un nuevo objeto Articulo con los parámetros proporcionados.

**Métodos de Acceso**

**public int getId()**

Este método devuelve el identificador único del artículo.

**public int getPrecio()**

Este método devuelve el precio del artículo.

**public String getNombre()**

Este método devuelve el nombre del artículo.

**public void setId(int id)**

Este método establece el identificador único del artículo.

**public void setPrecio(int precio)**

Este método establece el precio del artículo.

**public void setNombre(String nombre)**

Este método establece el nombre del artículo.

# Clase Pantalon

La clase Pantalon representa un pantalón con las siguientes características:

## Atributos

1. **id (int):** Identificador único del pantalón.
2. **talla (String):** Talla del pantalón.
3. **material (String):** Material del pantalón.
4. **color (String):** Color del pantalón.

## Métodos

### Constructores

#### public Pantalon(String talla, String material, String color)

Este constructor permite instanciar un nuevo objeto Pantalon con la talla, material y color proporcionados.

#### public Pantalon(int id, String talla, String material, String color)

Este constructor permite instanciar un nuevo objeto Pantalon con el identificador, talla, material y color proporcionados.

### Métodos de Acceso

#### public int getId()

Este método devuelve el identificador único del pantalón.

#### public String getTalla()

Este método devuelve la talla del pantalón.

#### public String getMaterial()

Este método devuelve el material del pantalón.

#### public String getColor()

Este método devuelve el color del pantalón.

#### public void setId(int id)

Este método establece el identificador único del pantalón.

#### public void setTalla(String talla)

Este método establece la talla del pantalón.

#### public void setMaterial(String material)

Este método establece el material del pantalón.

#### public void setColor(String color)

Este método establece el color del pantalón.

# Clase Polera

La clase Polera representa una polera con las siguientes características:

## Atributos

1. **id (int):** Identificador único de la polera.
2. **talla (String):** Talla de la polera.
3. **material (String):** Material de la polera.
4. **color (String):** Color de la polera.

## Métodos

### Constructor

#### public Polera(int id, String talla, String material, String color)

Este constructor permite instanciar un nuevo objeto Polera con el identificador, talla, material y color proporcionados.

### Métodos de Acceso

#### public int getId()

Este método devuelve el identificador único de la polera.

#### public String getTalla()

Este método devuelve la talla de la polera.

#### public String getMaterial()

Este método devuelve el material de la polera.

#### public String getColor()

Este método devuelve el color de la polera.

#### public void setId(int id)

Este método establece el identificador único de la polera.

#### public void setTalla(String talla)

Este método establece la talla de la polera.

#### public void setMaterial(String material)

Este método establece el material de la polera.

#### public void setColor(String color)

Este método establece el color de la polera.

# Clase Vestido

La clase Vestido representa un vestido con las siguientes características:

## Atributos

1. **id (int):** Identificador único del vestido.
2. **tela (String):** Tipo de tela del vestido.
3. **talla (String):** Talla del vestido.
4. **diceno (String):** Diseño del vestido.

## Métodos

### Constructor

#### public Vestido(int id, String tela, String talla, String diceno)

Este constructor permite instanciar un nuevo objeto Vestido con el identificador, tipo de tela, talla y diseño proporcionados.

### Métodos de Acceso

#### public int getId()

Este método devuelve el identificador único del vestido.

#### public String getTela()

Este método devuelve el tipo de tela del vestido.

#### public String getTalla()

Este método devuelve la talla del vestido.

#### public String getDiceno()

Este método devuelve el diseño del vestido.

#### public void setId(int id)

Este método establece el identificador único del vestido.

#### public void setTela(String tela)

Este método establece el tipo de tela del vestido.

#### public void setTalla(String talla)

Este método establece la talla del vestido.

#### public void setDiceno(String diceno)

Este método establece el diseño del vestido.

# Clase Conexion

La clase Conexion proporciona métodos para establecer una conexión a la base de datos.

## Atributos

1. **con (Connection):** Objeto que representa la conexión a la base de datos.

## Métodos

### public static Connection getConexion() throws SQLException, ClassNotFoundException

Este método estático devuelve una instancia de la conexión a la base de datos. Si la conexión aún no se ha establecido, se crea una nueva conexión utilizando el controlador JDBC para MySQL.

* **Lanza Excepciones:**
  + SQLException: Si ocurre un error al interactuar con la base de datos.
  + ClassNotFoundException: Si no se encuentra la clase del controlador JDBC.

## Detalles de Implementación

* La conexión se realiza utilizando el controlador JDBC para MySQL.
* La URL de conexión es: "jdbc:mysql://localhost/bdarticulo".
* Se utiliza el usuario "root" y una contraseña vacía "" para acceder a la base de datos.

## Uso Recomendado

Para obtener una instancia de la conexión, se recomienda llamar al método estático getConexion().

# Clase ArticuloDAO

La clase ArticuloDAO se encarga de realizar operaciones de acceso a la base de datos para la entidad Articulo.

## Métodos

### public boolean ingresarArticulo(Articulo ar)

Este método ingresa un nuevo artículo a la base de datos.

### public boolean modificarArticulo(Articulo ar)

Este método modifica un artículo existente en la base de datos.

### public boolean eliminarArticulo(int id)

Este método elimina un artículo de la base de datos según su identificador.

### public ArrayList<Articulo> obtenerTodos()

Este método obtiene todos los artículos almacenados en la base de datos y los devuelve como un ArrayList de objetos Articulo.

### public Articulo buscarArticulo(int id)

Este método busca un artículo en la base de datos según su identificador y devuelve un objeto Articulo si se encuentra.

### public int contarArticulos()

Este método cuenta la cantidad total de artículos almacenados en la base de datos.

# Clase PantalonDAO

La clase PantalonDAO se encarga de realizar operaciones de acceso a la base de datos para la entidad Pantalon.

## Métodos

### public boolean ingresarPantalon(Pantalon pan)

Este método ingresa un nuevo pantalón a la base de datos.

### public boolean modificarPantalon(Pantalon pan)

Este método modifica un pantalón existente en la base de datos.

### public boolean eliminarPantalon(int id)

Este método elimina un pantalón de la base de datos según su identificador.

### public Pantalon buscarPantalon(int id)

Este método busca un pantalón en la base de datos según su identificador y devuelve un objeto Pantalon si se encuentra.

### public int contarPantalones()

Este método cuenta la cantidad total de pantalones almacenados en la base de datos.

# Clase PoleraDAO

La clase PoleraDAO se encarga de realizar operaciones de acceso a la base de datos para la entidad Polera.

## Métodos

### public boolean ingresarPolera(Polera pol)

Este método ingresa una nueva polera a la base de datos.

### public boolean modificarPolera(Polera pol)

Este método modifica una polera existente en la base de datos.

### public boolean eliminarPolera(int id)

Este método elimina una polera de la base de datos según su identificador.

### public Polera buscarPolera(int id)

Este método busca una polera en la base de datos según su identificador y devuelve un objeto Polera si se encuentra.

### public int contarPoleras()

Este método cuenta la cantidad total de poleras almacenadas en la base de datos.

# Clase VestidoDAO

La clase VestidoDAO se encarga de realizar operaciones de acceso a la base de datos para la entidad Vestido.

## Métodos

### public boolean ingresarVestido(Vestido ves)

Este método ingresa un nuevo vestido a la base de datos.

### public boolean modificarVestido(Vestido ves)

Este método modifica un vestido existente en la base de datos.

### public boolean eliminarVestido(int id)

Este método elimina un vestido de la base de datos según su identificador.

### public Vestido buscarVestido(int id)

Este método busca un vestido en la base de datos según su identificador y devuelve un objeto Vestido si se encuentra.

### public int contarVestidos()

Este método cuenta la cantidad total de vestidos almacenados en la base de datos.

# Clase Jose\_Miguel\_Lucas\_Baeza

La clase Jose\_Miguel\_Lucaz\_Baeza representa la clase principal del programa y contiene el método main que inicia la ejecución.

## Método main

### public static void main(String[] args)

Este método es el punto de entrada principal del programa. Realiza las siguientes acciones:

* Crea una instancia de la clase Ventana\_Inicio llamada ventInic.
* Configura la posición, título, capacidad de redimensionamiento y comportamiento de cierre de la ventana.
* Hace visible la ventana de inicio.

**Notas adicionales:**

* La ventana de inicio (Ventana\_Inicio) parece ser la interfaz principal de la aplicación.

# Clase Ventana\_Inicio

**Descripción General**

La clase Ventana\_Inicio es la interfaz principal de la aplicación, proporcionando una visión general de la cantidad de diferentes tipos de artículos almacenados en la base de datos.

**Componentes Principales**

1. **DAO (Data Access Object):**
   * Se crean instancias de los DAO para cada tipo de artículo (Articulo, Polera, Pantalon, Vestido).
2. **Timer y ActionListener:**
   * Utiliza un Timer para realizar consultas periódicas a la base de datos y actualizar la cantidad de artículos en la interfaz gráfica.
   * El ActionListener del Timer ejecuta el código dentro del actionPerformed, consultando las cantidades de cada tipo de artículo y actualizándolas en la interfaz.
3. **Actualización de la Interfaz:**
   * Los resultados de las consultas se reflejan en JLabels dentro de la interfaz gráfica.
   * La información se actualiza en tiempo real gracias a la clase Timer.
4. **Menú y Opciones:**
   * **MenuBar:**
     + **Archivo:**
       - Contiene submenús para agregar distintos tipos de artículos (Pantalón, Polera, Vestido). Cada opción del menú permite crear un artículo del tipo correspondiente.
     + **Mostrar:**
       - Permite a los usuarios ver la información de los artículos existentes.
       - Facilita acciones como modificar o eliminar artículos basados en su identificador (ID).
5. **Apartado de Datos:**
   * Muestra la cantidad de artículos en el inventario utilizando los DAO contar para obtener los números.
   * Se actualizan en tiempo real gracias a la clase Timer.

# Clase Ventana\_Agregar\_Pantalon

## Descripción

La clase Ventana\_Agregar\_Pantalon representa la interfaz gráfica que facilita al usuario la entrada de información para la creación de un nuevo artículo de tipo Pantalón en la aplicación.

## Interfaz de Usuario

La interfaz presenta al usuario un formulario que incluye campos para ingresar los siguientes detalles del artículo Pantalón:

* **ID:** Identificación única del artículo.
* **Nombre:** Nombre descriptivo del pantalón.
* **Talla:** Talla del pantalón seleccionada a través de un menú desplegable.
* **Color:** Opciones de color (Azul, Negro, Blanco) seleccionadas mediante botones de opción.
* **Material:** Material del pantalón elegido a través de un menú desplegable.
* **Precio:** Precio del pantalón.

## Funcionalidad de Botones

### Botón Ingresar

* **Descripción:** Este botón envía la información ingresada por el usuario a la base de datos para crear un nuevo artículo de tipo Pantalón.
* **Acciones:**
  1. Captura los valores ingresados en los campos.
  2. Realiza validaciones para asegurar que los datos sean correctos.
  3. Crea instancias de las clases Articulo y Pantalon.
  4. Verifica la unicidad del ID en la base de datos.
  5. Ingresa el nuevo pantalón en la base de datos si todos los datos son válidos.

### Botón Volver

* **Descripción:** Cierra la ventana actual, permitiendo al usuario regresar a la interfaz principal o a la ventana anterior.
* **Acciones:**
  + Cierra la ventana actual y regresa al usuario a la ubicación anterior en la interfaz.

# Clase Ventana\_Agregar\_Polera

## Descripción

La clase Ventana\_Agregar\_Polera representa la interfaz gráfica que facilita al usuario la entrada de información para la creación de un nuevo artículo de tipo Polera en la aplicación.

## Interfaz de Usuario

La interfaz presenta al usuario un formulario que incluye campos para ingresar los siguientes detalles del artículo Polera:

* **ID:** Identificación única del artículo.
* **Nombre:** Nombre descriptivo del pantalón.
* **Talla:** Talla de la polera seleccionada a través de un menú desplegable.
* **Material:** Material de la polera elegido a través de un menú desplegable.
* **Color:** Opciones de color (Blanco, Negro, Azul, Marrón, Gris, Verde, Naranja, Rojo,) seleccionadas un menú desplegable.
* **Precio:** Precio de la polera.

## Funcionalidad de Botones

### Botón Ingresar

* **Descripción:** Este botón envía la información ingresada por el usuario a la base de datos para crear un nuevo artículo de tipo Polera.
* **Acciones:**
  1. Captura los valores ingresados en los campos.
  2. Realiza validaciones para asegurar que los datos sean correctos.
  3. Crea instancias de las clases Articulo y Polera.
  4. Verifica la unicidad del ID en la base de datos.
  5. Ingresa la nueva polera en la base de datos si todos los datos son válidos.

### Botón Volver

* **Descripción:** Cierra la ventana actual, permitiendo al usuario regresar a la interfaz principal o a la ventana anterior.
* **Acciones:**
  + Cierra la ventana actual y regresa al usuario a la ubicación anterior en la interfaz.

# Clase Ventana\_Agregar\_Vestido

## Descripción

La clase Ventana\_Agregar\_Vestido representa la interfaz gráfica que facilita al usuario la entrada de información para la creación de un nuevo artículo de tipo Vestido en la aplicación.

## Interfaz de Usuario

La interfaz presenta al usuario un formulario que incluye campos para ingresar los siguientes detalles del artículo Vestido:

* **ID:** Identificación única del artículo.
* **Nombre:** Nombre descriptivo del vestido.
* **Tela:** Material del vestido elegido a través de un menú desplegable.
* **Talla:** Talla del vestido seleccionada a través de un menú desplegable.
* **Diseño:** Opciones de diseño (Si, No) seleccionadas mediante botones de opción.
* **Precio:** Precio del vestido.

## Funcionalidad de Botones

### Botón Ingresar

* **Descripción:** Este botón envía la información ingresada por el usuario a la base de datos para crear un nuevo artículo de tipo Vestido.
* **Acciones:**
  1. Captura los valores ingresados en los campos.
  2. Realiza validaciones para asegurar que los datos sean correctos.
  3. Crea instancias de las clases Articulo y Vestido.
  4. Verifica la unicidad del ID en la base de datos.
  5. Ingresa el nuevo vestido en la base de datos si todos los datos son válidos.

### Botón Volver

* **Descripción:** Cierra la ventana actual, permitiendo al usuario regresar a la interfaz principal o a la ventana anterior.
* **Acciones:**
  + Cierra la ventana actual y regresa al usuario a la ubicación anterior en la interfaz.

# Clase Ventana\_Mostrar

## Descripción

La clase Ventana\_Mostrar representa la interfaz gráfica que facilita al usuario la visualización y modificación de la información dentro de la base de datos.

## Interfaz de Usuario

La interfaz presenta al usuario un formulario que incluye campos para ingresar los siguientes detalles del artículo Vestido:

* **Ingrese id Del Articulo:** es un JTextFiled en el que se ingresara el número de identificador(ID) del articulo.

## Funcionalidad de Botones

### Botón buscar

* **Descripción:** buscara el id ingresado en el panel de texto.
* **Acciones:**
  + Al presionar buscara el id en la tabla artículos con el método DAO buscarArticulo.

### Botón modificar

* **Descripción: Podrá modificar la columna precio y nombre de la tabla para actualizar la base de datos.**
* **Acciones:**
  + Luego de modificar los datos presione el botón y utilizara el método de articuloDAO Modificar.

### Botón limpiar

* **Descripción:** limpia los datos de la tabla.

### Botones Mostrar

* Mostrar Todo: Mostrara una vista que fue creada en la base de datos dentro de la tabla.
* Mostrar Pantalón: Mostrara la tabla pantalon que fue creada en la base de datos dentro de la tabla.
* Mostrar Polera: Mostrara la tabla polera que fue creada en la base de datos dentro de la tabla.
* Mostrar Vestido: Mostrara la tabla vestido que fue creada en la base de datos dentro de la tabla.

### Botón Volver

* **Descripción:** Cierra la ventana actual, permitiendo al usuario regresar a la interfaz principal o a la ventana anterior.
* **Acciones:**
  + Cierra la ventana actual y regresa al usuario a la ubicación anterior en la interfaz.

## Barra de menú

### Menú Eliminar

* **Descripción: en la barra de menú se encuentra la opción eliminar que está dividida según el tipo de articulo y al ingresar se le solicitara el id del articulo.**
* **Acciones:**
  + **Elimina el articulo de la base de datos**

# Anexo

Base de datos.

Tabla Articulo:

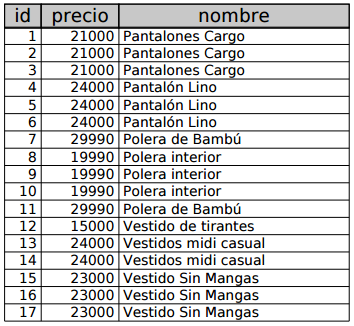


Tabla Pantalon:



Tabla Polera:

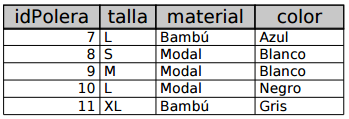
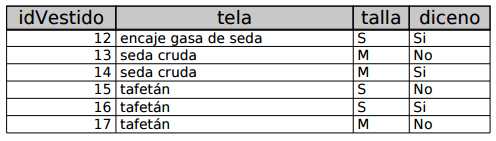


Tabla Vestido:



Vista vista1:

