

# Manual Técnico

## Caratula del codigo (resumido)

```
Proyecto1 > J Inventario.java > Inventario
1 > import java.io.FileWriter; //Sirve para crear el archivo .txt...
7
8 public class Inventario {
9
10 //Menu de opciones
11 > public static void main(String[] args) { ...
71
72 //Variables
73 public static String[][] invmatriz = new String[50][5]; //[#Espacio de almacenaminto][datos de producto], invmatriz = matriz de inventario
74 public static int contarfila = 0; //conteo de la fila
75 public static String[][] bitacoraM = new String[100][4]; // Matriz para la bitacora, [100 acciones][4 datos / accion]
76 public static int ContarBitacora = 0; // Conteo de registros de la bitacora
77
78 // Agrega producto
79 > public static void Aproducto(Scanner scanner, String estudiante){...
115 // Buser producto
116 > public static void Bproducto(Scanner scanner, String estudiante){...
180 // Eliminar producto
181 > public static void Eproducto(Scanner scanner, String estudiante) { ...
213 // Registro de venta
214 > public static void Rventa(Scanner scanner, String estudiante) { ...
263 // Generar reporte -- No funcional --
264 > public static void Greporte(Scanner scanner, String estudiante) { ...
268
269 //Método para guardar la venta en un archivo de texto
270 > public static void guardar(String codigo, int cantidad, String fecha, float total) { ...
277 // Ver datos de estudiante
278 > public static void Vestudiante(Scanner scanner, String estudiante) { ...
285 //Bitacora
286 > public static void Bitacora(Scanner scanner) { ...
302
303 //Metodo para la registrar la bitacora
304 > public static void RBitacora(String accion, String estado, String estudiante) { //RBitacora = Registrar Bitacora...
320
321
322 }
```

## Ideas

- Al principio del Proyecto se quería realizar un archivo .java para cada método para un mejor orden pero se descarto la idea después de tener complicaciones con las variables al tener que ser importadas y no se reconocidas.
- Para las variables de la matriz y bitácora se opto por hacer las public static que permite que se pueda acceder desde cualquier clase ya que esta es llamada en la mayor parte del código en los métodos.
- Al no lograr generar el pdf se opto como alternativa genera un archivo .txt para mostrar el registro de ventas

## Diccionario de Métodos

### Librerías

**java.io.FileWriter:** Permite escribir datos en un archivo.

**java.io.IOException:** Es una excepción que se usa para manejar errores que pueden ocurrir al trabajar con archivos.

**java.io.PrintWriter:** Ofrece métodos convenientes para escribir texto en un flujo de salida.

**java.text.SimpleDateFormat:** Se usa para dar formato a fechas y horas.

**java.util.Date:** Representa una fecha y hora específicas. Se usa junto con **SimpleDateFormat** para obtener la fecha y hora actual.

**java.util.Scanner:** Permite leer la entrada del usuario desde la consola.

## Métodos del Código

**main(String[] args):** Es el punto de entrada del programa. Aquí se inicia el menú principal, se pide la entrada del usuario y se llama a los otros métodos según la opción elegida.

**Aproducto(Scanner scanner, String estudiante):** Se encarga de agregar un nuevo producto al inventario. Pide al usuario los datos del producto (nombre, categoría, precio, etc.) y los guarda en la matriz invmatriz.

**Bproducto(Scanner scanner, String estudiante):** Permite buscar productos en el inventario por nombre, categoría o código. Muestra los datos de los productos que coinciden con la búsqueda.

**Eproducto(Scanner scanner, String estudiante):** Elimina un producto del inventario. Pide el código del producto.

**Rventa(Scanner scanner, String estudiante):** Registra una venta. Pide el código del producto y la cantidad vendida, actualiza el stock, calcula el total de la venta y llama al método guardar para registrar la información.

**guardar(String codigo, int cantidad, String fecha, float total):** Un método auxiliar que guarda los detalles de una venta en el archivo ventas.txt.

**Vestudiante(Scanner scanner, String estudiante):** Muestra los datos del estudiante.

**Bitacora(Scanner scanner):** Muestra un registro de todas las acciones que se han realizado en el programa.

**RBitacora(String accion, String estado, String estudiante):** Un método auxiliar que registra una acción en la matriz de la bitácora (bitacoraM) con su fecha, tipo de acción, estado (correcto/erróneo) y el nombre del estudiante.