

# Manual Técnico

## 1. Método Main

Aquí se da la bienvenida y se hace la llamada al método menu.

```
public static void main(String[] args) throws IOException {  
    System.out.println(";Bienvenido a la Tienda de Ropa Jujutsu Kaisen!");  
    System.out.println("");  
    Menu();  
}
```

## 2. Método Menu

En él se encuentran las opciones que tiene el usuario a seleccionar. Es aquí donde los demás métodos se adjuntan.

```
public static void Menu() throws IOException {  
    //Entrada de texto  
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);  
    BufferedReader buffer = new BufferedReader(capturarTeclado);  
  
    boolean salir = false;  
  
    do {  
        System.out.println("1.Ingresar a inventario ");  
        System.out.println("2.Salir ");  
        String entrada = buffer.readLine();  
        Integer Entrada = Integer.parseInt(entrada);  
  
        if (Entrada == 1) {  
            //Solicitando y guardando nombre de la persona  
            try { //Para evitar que el sistema colapse cuando se haya llegado al límite de personas registradas  
                Nombre();  
                int N = contador - 1;  
                //System.out.println(N);  
                //System.out.println(Persona[0][N]);  
                //System.out.println(Persona[0][1]);  
  
                boolean salir_n = false;  
                do {  
                    //Entrada al menu de opciones  
  
                    if (Persona[0][N] != null) {  
                        //agregar algo  
                        String a = "correcta";  
                        String b = "Ingreso al inventario";  
                        Acciones(a, b); //Se registra esta acción
```

```
System.out.println("");  
System.out.println("Hola! " + Persona[0][N] + " Aquí está el menú del inventario (Recuerda cada acción se registra en bitácora)");  
System.out.println("");  
System.out.println("1.Agregar Producto");  
System.out.println("2.Buscar Producto");  
System.out.println("3.Eliminar Producto");  
System.out.println("4.Registrar Venta");  
System.out.println("5.Bitácora");  
System.out.println("6.Generar Reportes");  
System.out.println("7.Ver Datos del Estudiante");  
System.out.println("8.Salir");  
  
//Menú  
try {  
    String opcion = buffer.readLine();  
    Integer OPCION = Integer.parseInt(opcion);
```

```

switch (OPCION) {
    case 1 -> { //AGREGAR PRODUCTO

        Producto();
        Categoria();
        Precio();
        Stock();
       Codigo();
        String c = "correcta";
        String d = "Se agrego el producto: " + Nombre Producto[0][contador1 - 1];
        Acciones(c, d); //Se registra esta acción
        //continue;
        break;

    }
    case 2 -> { //MOSTRAR PRODUCTO
        Opciones_Mostrar();
        break;
    }
}

```

```

case 3 -> { //BORRAR PRODUCTO
    E Codigo();
    break;
}
case 4 -> { //REGISTRAR VENTA
    Venta();
    break;
}
case 5 -> { //BITÁCORA
    Bitacora();
    break;
}
case 6 -> { //GENERAR REPORTES
    PDF();
    break;
}
case 7 -> { //VER DATOS DEL ESTUDIANTE
    Estudiante();
    break;
}
case 8 -> { //SALIR
    String c = "correcta";
    String d = "Saliendo del inventario";
    Acciones(c, d); //Se registra esta acción
    salir_n = true;
}
default -> { //Cualquier opción que no este en el menú
    System.out.println("Opción no valida→selecciona una opción del menú");
    String e = "incorrecta";
    String f = "Opción no valida→selecciona una opción del menú";
    Acciones(e, f); //Se registra esta acción
}
}

```

### 3. Método Nombre

Su función es poder registrar el nombre de cada usuario cada que se ingresa y sale del inventario.

```
public static void Nombre() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    boolean salir2=false;
    do{
        System.out.println("Ingresa tu nombre para poder acceder al menú");
        String nombre = buffer.readLine();

        //Se agrega a la matriz
        Persona[0][contador] = nombre;
        contador++;
        salir2=true;
        if (Persona[0][contador-1].equals("")){
            Persona[0][contador-1]=null;
            contador=contador-1;
            System.out.println("Porfavor acceda un nombre válido");
            salir2=false;
            //break;
        }
    }while(!salir2);

    //System.out.println(Persona[0][contador - 1]);

    System.out.println("Se agrego tu nombre a la bitacora");
}
```

#### 4. Método Producto

Aquí se encuentran todos los siguientes métodos hasta el métodos código, dado que son todos los atributos que solicita un producto para poder agregarse al inventario.

```
public static void Producto() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    boolean retirada = true;
    while (retirada == true) {
        System.out.println("Ingresa el nombre del producto: ");
        String nombreproducto = buffer.readLine();
        retirada = false;
        //Nombre_Producto[0][0]="Posición de muestra";
        //Se agrega a la matriz
        Nombre_Producto[0][contador1] = nombreproducto;
        contador1++;
        //System.out.println(Nombre_Producto[0][contador1]);

        //Se revisa que el nombre no este repetido
        for (int i = 0; i < (contador1 - 1); i++) {
            if (Nombre_Producto[0][i].toLowerCase().equals(Nombre_Producto[0][contador1 - 1].toLowerCase())) {
                System.out.println("El nombre ya existe,ingresa otro nombre");
                String g = "incorrecta";
                String f = "El nombre ya existe,ingresa otro nombre";
                Acciones(g, f); //Se registra esta acción
                Nombre_Producto[0][contador1 - 1] = null;
                contador1 = contador1 - 1;
                retirada = true;
                break;
            }
        }
        break;
    }
}
```

#### 5. Método Categoría

Guarda el dato de categoría al ingresar un producto.

```
public static void Categoria() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa la categoría a la cual pertenece");
    String cat = buffer.readLine();

    //Se agrega a la matriz
    Categoria_Producto[0][contador2] = cat;
    contador2++;
    System.out.println(Categoria_Producto[0][contador2 - 1]);

    System.out.println("Se agrego la categoría");
}
```

## 6. Método Precio

Guarda el dato de precio al ingresar un producto.

```
public static void Precio() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    boolean retirada = true;
    while (retirada == true) {
        System.out.println("Ingresa el precio del producto");
        String pre = buffer.readLine();
        float PRE = Float.parseFloat(pre);
        if (PRE > -1) {
            //Se agrega a la matriz
            Precio[0][contador5] = PRE;
            contador5++;
            System.out.println(Precio[0][contador5 - 1]);
            System.out.println("Se agrego el precio");
            retirada = false;
        } else {
            System.out.println("No puede agregar números negativos, intentelo de nuevo");
            String g = "incorrecta";
            String f = "No puede agregar números negativos, intentelo de nuevo";
            Acciones(g, f); //Se registra esta acción
            retirada = true;
        }
    }
}
```

## 7. Método Stock

Guarda el dato de stock al ingresar un producto.

```
public static void Stock() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    boolean retirada = true;
    while (retirada == true) {
        System.out.println("Ingresa el stock del producto (en números enteros)");
        String sto = buffer.readLine();
        Integer STO = Integer.parseInt(sto);
        if (STO > -1) {
            //Se agrega a la matriz
            Stock[0][contador3] = STO;
            contador3++;
            System.out.println(Stock[0][contador3 - 1]);
            System.out.println("Se agrego el stock");
            retirada = false;
        } else {
            System.out.println("No puede agregar números menos a 0, intentelo de nuevo");
            String g = "incorrecta";
            String f = "No puede agregar números menos a 0, intentelo de nuevo";
            Acciones(g, f); //Se registra esta acción
            retirada = true;
        }
    }
}
```

## 8. Método Codigo

Guarda el dato de código al ingresar un producto.

```
public static void Codigo() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    boolean retirada = true;
    while (retirada == true) {
        System.out.println("Ingresa el código único del producto: ");
        String cod = buffer.readLine();
        Integer COD = Integer.parseInt(cod);
        if (COD > -1) {
            retirada = false;
            //Nombre_Producto[0][0]="Posición de muestra";
            //Se agrega a la matriz
            Codigo[0][contador4] = COD;
            contador4++;
            //System.out.println(Codigo[0][contador4]);
            //Se revisa que el nombre no este repetido

            for (int i = 0; i < (contador4 - 1); i++) {
                if (Objects.equals(Codigo[0][i], Codigo[0][contador4 - 1])) {
                    System.out.println("El código ya existe");
                    String g = "incorrecta";
                    String f = "El código ya existe";
                    Acciones(g, f); //Se registra esta acción
                    Codigo[0][contador4 - 1] = null;
                    contador4 = contador4 - 1;
                    retirada = true;
                    break;
                }
            }
        } else {
            System.out.println("No puede agregar números negativos, porfavor intentelo de nuevo");
            String g = "incorrecta";
            String f = "No puede agregar números negativos, porfavor intentelo de nuevo";
            Acciones(g, f); //Se registra esta acción
            retirada = true;
        }
    }
}
```

## 9. Método Opciones\_Mostrar

Es el primer filtro que tiene el usuario para buscar un producto.

```
public static void Opciones_Mostrar() throws IOException {
    //Entrada de texto
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    boolean salir = false;

    //Solicitando y guardando nombre de la persona
    try { //Para evitar que el sistema colapse cuando se haya llegado al límite de personas registradas

        boolean salir_n = false;
        do {
            //Entrada al menu de opciones
            System.out.println("");
            System.out.println("Selecciona el filtro para ver el/los productos:");
            System.out.println("1.Nombre del producto ");
            System.out.println("2.Categoría ");
            System.out.println("3.Precio");
            System.out.println("4.Cantidad en stock ");
            System.out.println("5.Código único del producto ");
            System.out.println("6.Salir ");
            System.out.println("");

            try {
                String opcion = buffer.readLine();
                Integer OPCION = Integer.parseInt(opcion);

                switch (OPCION) {
                    case 1 -> { //Por Nombre del producto
                        M_Nombre();
                        break;
                    }
                    case 2 -> { // Por categoría
                        M_Categoria();
                        break;
                    }
                    case 3 -> { // Por precio
                        M_Precio();
                        break;
                    }
                    case 4 -> { // Por cantidad en stock
                        M_Stock();
                        break;
                    }
                    case 5 -> { // Por Código único del producto
                        M_Codigo();
                        break;
                    }
                    case 6 -> { //Salir
                        salir_n = true;
                    }
                    default -> {
                        System.out.println("Opció no valida-selecciona una opción del menú");// Este es por si le persona no escoge ninguna opción
                        String g = "incorrecta";
                        String f = "Opció no valida-selecciona una opción del menú";
                        Acciones(g, f); //Se registra esta acción
                    }
                }
            } catch (NumberFormatException e) {
                System.out.println("Opció no valida-selecciona una opción del menú");
            }
        } while (!salir_n);
    }
}
```

## 10. Método M\_Nombre

Es uno de los filtros que tiene el usuario para buscar un producto.

```
public static void M_Nombre() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa el nombre del producto");
    String nm = buffer.readLine();
    boolean respuesta = false;
    for (int i = 0; i < Nombre_Producto[0].length; i++) {
        if (Nombre_Producto[0][i] != null && nm.toLowerCase().equals(Nombre_Producto[0][i].toLowerCase())) {
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " +Codigo[0][i]);
            System.out.println("");

            String c = "correcta";
            String d = "Se mostro por opción de nombre al producto: " + nm;
            Acciones(c, d); //Se registra esta acción
            respuesta = true;
        }
    }
    if (respuesta == false) {
        System.out.println("No existe el producto");
        String g = "incorrecta";
        String f = "No existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 11. Método M\_Categoria

Es uno de los filtros que tiene el usuario para buscar un producto.

```
public static void M_Categoria() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa la categoría del producto");
    String nm = buffer.readLine();
    boolean respuesta = false;
    for (int i = 0; i < Categoria_Producto[0].length; i++) {
        if (Categoria_Producto[0][i] != null && nm.toLowerCase().equals(Categoria_Producto[0][i].toLowerCase())) {
            System.out.println("-----");
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " +Codigo[0][i]);
            System.out.println("-----");
            System.out.println("");

            String c = "correcta";
            String d = "Se mostro producto(s) con categoría: " + nm;
            Acciones(c, d); //Se registra esta acción
            respuesta = true;
        }
    }
    if (respuesta == false) {
        System.out.println("Categoría no encontrada,no existe el producto");
        String g = "incorrecta";
        String f = "Categoría no encontrada,no existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```



## 12. Método M\_Precio

Es uno de los filtros que tiene el usuario para buscar un producto.

```
public static void M_Precio() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa el precio");
    String nc = buffer.readLine();
    float NC = Float.parseFloat(nc);
    boolean respuesta = false;
    for (int i = 0; i < Precio[0].length - 1; i++) {
        if (Objects.equals(Precio[0][i], NC)) {
            System.out.println("-----");
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " +Codigo[0][i]);
            System.out.println("-----");
            System.out.println("");
            String c = "correcta";
            String d = "Se mostro producto(s) con precio de: " + nc;
            Acciones(c, d); //Se registra esta acción
            respuesta = true;
        }
    }
    if (respuesta == false) {
        System.out.println("Precio no encontrado,no existe el producto");
        String g = "incorrecta";
        String f = "Precio no encontrado,no existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 13. Método M\_Stock

Es uno de los filtros que tiene el usuario para buscar un producto.

```
public static void M_Stock() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa el stock");
    String nc = buffer.readLine();
    Integer NC = Integer.parseInt(nc);
    boolean respuesta = false;
    for (int i = 0; i < Stock[0].length - 1; i++) {
        if (Objects.equals(Stock[0][i], NC)) {
            System.out.println("-----");
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " +Codigo[0][i]);
            System.out.println("-----");
            System.out.println("");
            String c = "correcta";
            String d = "Se mostro producto(s) con stock de: " + nc;
            Acciones(c, d); //Se registra esta acción
            respuesta = true;
        }
    }
    if (respuesta == false) {
        System.out.println("Stock no encontrado,no existe el producto");
        String g = "incorrecta";
        String f = "Stock, no encontrado,no existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 14. Método M\_Codigo

Es uno de los filtros que tiene el usuario para buscar un producto.

```
public static void M_Codigo() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa el código del producto");
    String nc = buffer.readLine();
    Integer NC = Integer.parseInt(nc);
    boolean respuesta = false;
    for (int i = 0; i < Codigo[0].length - 1; i++) {
        if (Objects.equals(Codigo[0][i], NC)) {
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " + Codigo[0][i]);
            System.out.println("");
            String c = "correcta";
            String d = "Se mostro producto con código de: " + nc;
            Acciones(c, d); //Se registra esta acción
            respuesta = true;
        }
    }
    if (respuesta == false) {
        System.out.println("Código no encontrado,no existe el producto");
        String g = "incorrecta";
        String f = "Código no encontrado,no existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 15. Método E\_Codigo

Es la función que permite borrar cualquier producto previamente registrado.

```
public static void E_Codigo() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);
    System.out.println("Ingresa el código del producto");
    String nc = buffer.readLine();
    Integer NC = Integer.parseInt(nc);
    boolean respuesta = false;
    for (int i = 0; i < Codigo[0].length - 1; i++) {
        if (Objects.equals(Codigo[0][i], NC)) {
            Nombre_Producto[0][i] = null;
            Categoria_Producto[0][i] = null;
            Precio[0][i] = 0; //por ser de tipo primitivo no admite null
            Stock[0][i] = null;
            Codigo[0][i] = null;
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " + Codigo[0][i]);
            System.out.println("");
            String c = "correcta";
            String d = "Se borro producto con código de: " + nc;
            Acciones(c, d); //Se registra esta acción
            respuesta = true;
        }
    }
    if (respuesta == false) {
        System.out.println("Código no encontrado,no existe el producto");
        String g = "incorrecta";
        String f = "Código no encontrado,no existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 16. Método Venta

Aquí se realizan las ventas por producto, cada que el usuario lo necesite.

```
public static void Venta() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    System.out.println("Escribe el código del producto de venta");
    String nc = buffer.readLine();
    Integer NC = Integer.parseInt(nc);

    boolean respuesta = false;
    for (int i = 0; i < Codigo[0].length - 1; i++) {
        if (Objects.equals(Codigo[0][i], NC)) {
            System.out.println("1.Nombre del producto: " + Nombre_Producto[0][i]);
            System.out.println("2.Categoría: " + Categoria_Producto[0][i]);
            System.out.println("3.Precio :Q" + Precio[0][i]);
            System.out.println("4.Cantidad en stock: " + Stock[0][i]);
            System.out.println("5.Código único del producto: " + Codigo[0][i]);
            System.out.println("");
            System.out.println("¿Cuánto quiere vender?(escriba la cantidad entera)");
            String ca = buffer.readLine();
            Integer CA = Integer.parseInt(ca);

            //validación de existencia y stock suficiente
            if (CA <= Stock[0][i]) {
                System.out.println("Si hay stock suficiente");
                System.out.println("");
                //Resta
                int Resta = Stock[0][i] - CA;
                Stock[0][i] = Resta;
                //Registro de CA
                Cantidad[0][contador8] = CA;
                contador8++;
                //Registro de P_V
                P_V[0][contador9] = Nombre_Producto[0][i];
                contador9++;

                //Registro
                System.out.println("Registrando venta con los siguientes datos");
                System.out.println("Nombre del producto " + P_V[0][contador9 - 1]);
                System.out.println("Código del producto: " + Codigo[0][i]);
                System.out.println("Cantidad vendida: " + Cantidad[0][contador8 - 1]);
                //Tiempo de transacción
                Tiempo[0][contador7] = LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"));
                System.out.println("Fecha y hora de transacción: " + Tiempo[0][contador7]);
                contador7++;
                //Se agrega el total
                Total[0][contador6] = Float.parseFloat(ca) * Precio[0][i]; //Esto para que no se elimine el float del precio a la hora de m
                System.out.println("Total de venta: " + Total[0][contador6]);
                contador6++;
                //para que no se repita
                String c = "correcta";
                String d = "Se registro una venta del producto " + Nombre_Producto[0][i] + " con un total de: " + Total[0][contador6 - 1];
                Acciones(c, d); //Se registra esta acción
            } else {
                System.out.println("Stock insuficiente");
                String g = "incorrecta";
                String f = "Stock insuficiente";
                Acciones(g, f); //Se registra esta acción
            }
            respuesta = true;
        }
        //break de prueba
    }
    if (respuesta == false) {
        System.out.println("Código no encontrado,no existe el producto");
        String g = "incorrecta";
        String f = "Código no encontrado,no existe el producto";
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 17. Método Bitacora

Es el encargado de mostrar todas las acciones que realice el usuario dentro del inventario.

```
public static void Bitacora() throws IOException {  
  
    System.out.println("ACCIONES CORRECTAS:");  
    for (int j = 0; j < gatito3; j++) {  
        System.out.println("-----");  
        System.out.println(Tiempos[0][j]);  
        System.out.println(TAC[0][j]);  
        System.out.println(Acciones_Correctas[0][j]);  
        System.out.println(Persona1[0][j]);  
    }  
    System.out.println("-----");  
    System.out.println("-----");  
    System.out.println("");  
    System.out.println("ACCIONES INCORRECTAS:");  
    for (int k = 0; k < gatito33; k++) {  
        System.out.println("-----");  
        System.out.println(Tiempos[0][k]);  
        System.out.println(TAI[0][k]);  
        System.out.println(Acciones_Incorrectas[0][k]);  
        System.out.println(Persona2[0][k]);  
    }  
}
```

## 18. Método Acciones

Guarda las acciones correctas e incorrectas de todos los usuarios, para el método bitacora.

```
public static String Acciones(String a, String b) throws IOException {  
  
    if (a == "correcta") {  
        Tiempos[0][gatito3] = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd-MM-yyyy HH-mm-ss"));  
        gatito3++;  
        TAC[0][gatito4] = b;  
        gatito4++;  
        Acciones_Correctas[0][gatito1] = "Correcta";  
        gatito1++;  
        Persona1[0][contp1] = Persona[0][contador - 1];  
        contp1++;  
    }  
    if (a == "incorrecta") {  
        Tiempos[0][gatito33] = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd-MM-yyyy HH-mm-ss"));  
        gatito33++;  
        TAI[0][gatito5] = b;  
        gatito5++;  
        Acciones_Incorrectas[0][gatito2] = "Incorrecta";  
        gatito2++;  
        Persona2[0][contp2] = Persona[0][contador - 1];  
        contp2++;  
    }  
    return null;  
}
```

## 19. Método PDF

Esta función únicamente le presenta al usuario qué tipo de reportes se puede generar.

```
public static void PDF() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    boolean salir = false;

    //Solicitando la opción
    try { //Para evitar que el sistema colapse cuando se haya llegado al límite de personas registradas

        boolean salir_n = false;
        do {
            //Entrada al menu de opciones
            System.out.println("");
            System.out.println("¿Qué quieres generar?");
            System.out.println("1.Reporte de Stock");
            System.out.println("2.Reporte de Ventas");
            System.out.println("3.Salir");
            System.out.println("");
```

```
        try {
            String opcion = buffer.readLine();
            Integer OPCION = Integer.parseInt(opcion);

            switch (OPCION) {
                case 1 -> { //Reporte de Stock
                    R_Stock();
                    break;
                }
                case 2 -> { //Reporte de Ventas
                    R_Venta();
                    break;
                }
                case 3 -> { // Salir
                    salir_n = true;
                }

                default -> {
                    System.out.println("Opción no valida-selecciona una opción del menú");// Este es por si le persona no escoge ninguna opción
                    String g = "incorrecta";
                    String f = "Opción no valida-selecciona una opción del menú";
                    Acciones(g, f); //Se registra esta acción
                }
            }
        }
    }
}
```

```
    } catch (NumberFormatException c) { //Este por si la persona ingresa un salto o espacio, letras (LETRAS)
        System.out.println("No puedes ingresar saltos de línea o letras ¡Recuerda! solo puedes ingresar las opciones del menú" + " Error= " + c);
        String g = "incorrecta";
        String f = "No puedes ingresar saltos de línea o letras ¡Recuerda! solo puedes ingresar las opciones del menú"+" Error= "+c;
        Acciones(g, f); //Se registra esta acción
    }

    while (!salir_n);

    catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Ups! el sistema llego a su limite de ingresos" + " Error= " + e);
        String g = "incorrecta";
        String f = "Ups! el sistema llego a su límite de ingresos"+" Error= "+e;
        Acciones(g, f); //Se registra esta acción
        //salir = true;
    }
}
}
```

## 20. Método R\_Venta

Se encarga de generar el pdf correspondiente a las ventas que ha realizado el usuario.

```
public static void R_Venta() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    if (contador6 == 0) {
        System.out.println("No hay productos para exportar al PDF.");
        String g = "incorrecta";
        String f = "No hay productos para exportar al PDF.";
        Acciones(g, f); //Se registra esta acción
        return;
    }

    System.out.print("Ruta/nombre del PDF (por defecto 'DD_MM_YYYY_HH_mm_ss_Venta.pdf'): ");
    String path = buffer.readLine();

    if (path.isEmpty()) {
        String Fecha = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd-MM-yyyy HH-mm-ss"));
        path = Fecha + "_Venta.pdf";
    }

    try (PdfWriter writer = new PdfWriter(path); PdfDocument pdf = new PdfDocument(writer); Document doc = new Document(pdf)) {

        doc.add(new Paragraph("Reporte de Venta")
            .setBold()
            .setFontSize(14));
        doc.add(new Paragraph("Reporte Generado: "
            + LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"))));

        Table table = new Table(new float[]{3, 6, 3, 3});
        table.useAllAvailableWidth();
        table.addHeaderCell(new Cell().add(new Paragraph("Productos Vendidos")));
        table.addHeaderCell(new Cell().add(new Paragraph("Cantidad")));
        table.addHeaderCell(new Cell().add(new Paragraph("Fecha")));
        table.addHeaderCell(new Cell().add(new Paragraph("Total de cada venta")));

        double total = 0.0;
        for (int i = 0; i < contador6; i++) {
            //Nombre_Producto p = Stock[0][i];
            table.addCell(new Cell().add(new Paragraph(String.valueOf(P V[0][i]))));
            table.addCell(new Cell().add(new Paragraph(String.valueOf(Cantidad[0][i]))));
            table.addCell(new Cell().add(new Paragraph(Tiempo[0][i])));
            table.addCell(new Cell().add(new Paragraph(String.valueOf("Q" + Total[0][i]))));
            total += Total[0][i];
        }

        // Fila resumen
        table.addCell(new Cell(1, 3).add(new Paragraph("Total productos: " + contador6)));
        table.addCell(new Cell().add(new Paragraph(String.format("Q" + "%.2f", total)))); //Esto puede producir un error sino lo cambiamos
        table.addCell(new Cell(1, 5).add(new Paragraph("Persona que realiza la venta: " + Persona[0][contador - 1]))); // Así siempre imp

        doc.add(table);
        System.out.println("PDF de productos vendidos: " + path);

        String c = "correcta";
        String d = "Se creo un documento de venta " + path;
        Acciones(c, d); //Se registra esta acción
    } catch (Exception e) {
        System.err.println("Error exportando PDF: " + e.getMessage());
        String g = "incorrecta";
        String f = "Erro exportando PDF: " + e.getMessage();
        Acciones(g, f); //Se registra esta acción
    }
}
```

## 21. Método R\_Stock

Su funcionalidad es parecida a la de venta con la diferencia que registra únicamente el stock de productos.

```
public static void R_Stock() throws IOException {
    InputStreamReader capturarTeclado = new InputStreamReader(System.in);
    BufferedReader buffer = new BufferedReader(capturarTeclado);

    if (contador1 == 0) {
        System.out.println("No hay productos para exportar al PDF.");
        String g = "incorrecta";
        String f = "No hay productos para exportar al PDF.";
        Acciones(g, f); //Se registra esta acción
        return;
    }

    System.out.print("Ruta/nombre del PDF (por defecto 'DD_MM_YYYY_HH_mm_ss_Stock.pdf'): ");
    String path = buffer.readLine();

    if (path.isEmpty()) {
        String Fecha = LocalDateTime.now().format(DateTimeFormatter.ofPattern("dd-MM-yyyy HH-mm-ss"));

        path = Fecha + "_Stock.pdf";
    }

    try (PdfWriter writer = new PdfWriter(path); PdfDocument pdf = new PdfDocument(writer); Document doc = new Document(pdf)) {

        doc.add(new Paragraph("Reporte de Stock")
            .setBold()
            .setFontSize(14));
        doc.add(new Paragraph("Reporte Generado: "
            + LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"))));

        Table table = new Table(new float[]{3, 6, 3, 3, 3});
        table.useAllAvailableWidth();
        table.addHeaderCell(new Cell().add(new Paragraph("Nombre")));
        table.addHeaderCell(new Cell().add(new Paragraph("Código")));
        table.addHeaderCell(new Cell().add(new Paragraph("Categoría")));
        table.addHeaderCell(new Cell().add(new Paragraph("Precio")));
        table.addHeaderCell(new Cell().add(new Paragraph("Cantidad disponible")));

        //double total = 0.0;
        for (int i = 0; i < contador1; i++) {
            //Nombre_Producto p = Stock[0][i];
            if (Nombre_Producto[0][i] != null) {
                table.addCell(new Cell().add(new Paragraph(String.valueOf(Nombre_Producto[0][i]))));
            }
            if (Codigo[0][i] != null) {
                table.addCell(new Cell().add(new Paragraph(String.valueOf(Codigo[0][i]))));
            }
            if (Categoria_Producto[0][i] != null) {
                table.addCell(new Cell().add(new Paragraph(Categoria_Producto[0][i])));
            }
            if (Precio[0][i] != 0) {
                table.addCell(new Cell().add(new Paragraph(String.valueOf("Q" + Precio[0][i]))));
            }
            if (Stock[0][i] != null) {
                table.addCell(new Cell().add(new Paragraph(String.valueOf(Stock[0][i]))));
            }
            //total += Codigo[0][i];
        }

        // Fila resumen
        contprueba = contador1;
        for (int i = 0; i < contador1; i++) {
            if (Nombre_Producto[0][i] == null) {
                contprueba = contprueba - 1;
            }
        }
        table.addCell(new Cell(1, 5).add(new Paragraph("Total productos registrados: " + contprueba)));
        table.addCell(new Cell(1, 5).add(new Paragraph("Persona que ingreso el inventario: " + Persona[0][contador - 1]));

        //table.addCell(new Cell().add(new Paragraph(String.format("%.2f", total))));
        doc.add(table);
        System.out.println("PDF de productos en stock: " + path);
    }
}
```

```

        //table.addCell(new Cell().add(new Paragraph(String.format("%.2f", total))));
        doc.add(table);
        System.out.println("PDF de productos en stock: " + path);

        String c = "correcta";
        String d = "Se creo un documento de stock " + path;
        Acciones(c, d); //Se registra esta acción
    } catch (Exception e) {
        System.err.println("Error exportando PDF: " + e.getMessage());
        String g = "incorrecta";
        String f = "Error exportando PDF: " + e.getMessage();
        Acciones(g, f); //Se registra esta acción
    }
}

```

## 22. Método Estudiante

Muestra los datos del estudiante que desarrolló el inventario.

```

//Método para ver dato del estudiante
public static void Estudiante() throws IOException {
    System.out.println("Facultad de Ingeniería");
    System.out.println("Laboratorio de Introducción a la Programación y Computación 1");
    System.out.println("");

    System.out.println("Nombre: Hany Clara Jannette Alvarez Bulux");
    System.out.println("CUI: 3263543281401 ");
    System.out.println("Carnet: 202401231");
    System.out.println("Correo: 3263543281401@ingenieria.usac.edu.gt ");

    String c = "correcta";
    String d = "Se ingreso a los datos de estudiante";
    Acciones(c, d); //Se registra esta acción
}

```