



Universidad  
Católica del Norte

# Taller 1

Elias Manque Olivares

20475385-7

Programación Avanzada

06/05/22



# Índice:

## Diagrama de Clases

----- Código ("documentación")

---- [Main\(\)](#)

---- [inicioSesion\(\)](#)

[Admin\(\)](#) **Completado 95%**

[bloquearUsuario\(\)](#)

[ventas\(\)](#)

[agregarProducto\(\)](#)

[addStock\(\)](#)

[txt.añadirATxtProductos\(\)](#)

[updatePrice\(\)](#)

[cerrarCiclo: boolean](#)

[Cliente\(\)](#) **Completado 60%**

[elegirProducto\(\)](#)

[cambiaContraseña\(\)](#)

[mostrarCatalogo\(\)](#)

[mostrarSaldo\(\)](#)

[recargarSaldo\(\)](#)

[verCarrito\(\)](#)

[quitarProducCarrito\(\)](#)

[pagarCarrito\(\)](#)

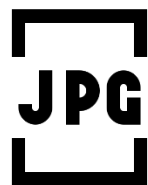
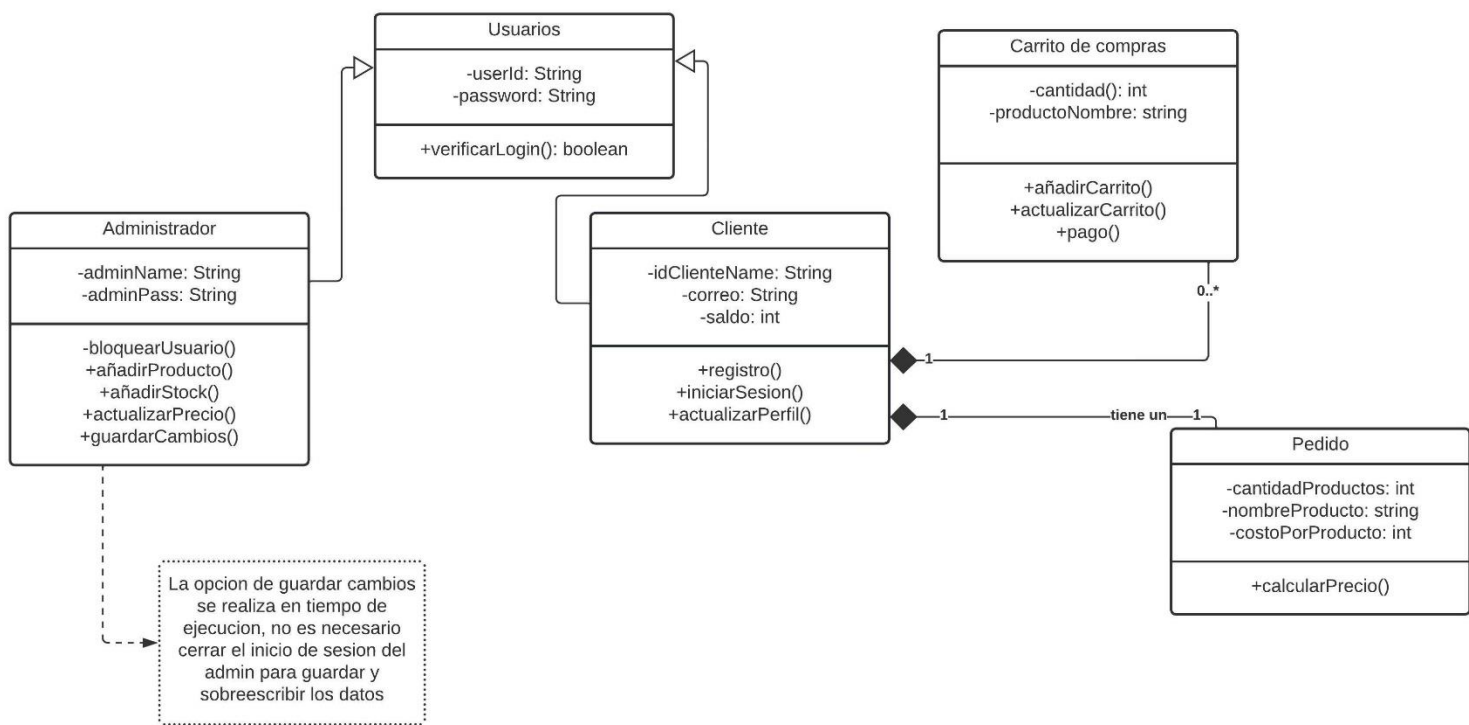
[txt.añadirATxtCientes\(\)](#)

[Registro\(\)](#)

---- funcionesComplementarias()

----- Variables y arreglos usados

# Diagrama de Clases



## Orden del código (main(): línea 476 - 527)

```
public static void main(String[] args) throws Exception {

    //Inicializacion de Variables
    boolean vueltaWhile = true;
    tamañoArch = conteoLineas(archClientes);
    tamañoArchProductos = conteoLineas(archProductos);
    tamañoArchVentas = conteoLineas(archVentas);

    //Variables para Arch Ventas
    productosVentas = new String[tamañoArchVentas];
    unidadesVendidas = new int[tamañoArchVentas];

    productosVentas = pasoTXTaListaStr(archVentas, productosVentas, 0);
    unidadesVendidas = pasoTXTaListaint(archVentas, unidadesVendidas, 1);

    //Variables para Arch Productos
    productosStock = new String[tamañoArchProductos];
    precioProducto = new int[tamañoArchProductos];
    unidadesDisponibles = new int[tamañoArchProductos];

    tamañoFijoProductos = productosStock.length;

    productosStock = pasoTXTaListaStr(archProductos, productosStock, 0);
    precioProducto = pasoTXTaListaint(archProductos, precioProducto, 1);
    unidadesDisponibles = pasoTXTaListaint(archProductos, unidadesDisponibles, 2);

    //Variables para Arch Cliente
    usuario = new String[tamañoArch];
    password = new String[tamañoArch];
    saldo = new int[tamañoArch];
    correos = new String[tamañoArch];

    tamAnteriorCliente = usuario.length;

    usuario = pasoTXTaListaStr(archClientes, usuario, 0);
    password = pasoTXTaListaStr(archClientes, password, 1);
    saldo = pasoTXTaListaint(archClientes, saldo, 2);
    correos = pasoTXTaListaStr(archClientes, correos, 3);

    //Inicializar Scanner
    Scanner leer = new Scanner(System.in);

    //Ingreso While
    while(vueltaWhile){
        //mostrarCATalogo(archProductos);
        inicioSesion(leer, password, archProductos, archClientes, productosVentas, unidadesVendidas,
            productosStock, unidadesDisponibles, precioProducto);
        vueltaWhile = detener();
    }
}
```

En el main se rellenan todas las listas principales, además la función "iniciarSesion" contendrá todo el programa, recibe el "Scanner leer" que se usará para ingresar datos por consola

## Orden del código(inicioSesion(): línea 340 - 395)

```
public static void inicioSesion(Scanner leer , String[] password, File productos, File clientes,
                                String[] nombProductosVendidos, int [] vecesVendidos, String [] nomProductos,
                                int[] stockProducto , int [] precio) throws Exception{

    String user;
    String pass;
    boolean validarWhile = true;
    boolean encontrado = false;
    int indice = 0;
    System.out.println("#####");
    System.out.println("");
    System.out.println("Bienvenido");
    System.out.println("");
    System.out.println("#####");
    while (validarWhile){
        System.out.println("Ingrese su nombre de Usuario");
        user = leer.next();
        if(user.equals("ADMIN")){
            System.out.println("Ingrese su Contraseña");
            pass = leer.next();
            if(pass.equals("NYAXIO")){
                encontrado = true;
                admin(Leer,productos,nomProductos,stockProducto, precio);
                break;
            }else{
                System.out.println("Contraseña incorrecta!");
                validarWhile = detener();
            }
        }else{
            for (int i = 0; i < usuario.length; i++){
                if (user.equals(usuario[i])){
                    encontrado = true;
                    System.out.println("Ingrese su Contraseña");
                    pass = leer.next();
                    if (pass.equals(password[i])){
                        System.out.println("#####");
                        System.out.println(" ");
                        indice = i;
                        userMenu(user, leer, productos, password, clientes, indice);
                        validarWhile = false;
                    }else{
                        System.out.println("Contraseña incorrecta!");
                        validarWhile = detener();
                    }
                }
            }
        }
    }
    if (!encontrado){
        validarWhile = false;
    }
    if (!encontrado) {
        System.out.println("Usuario no encontrado");
        Registro(leer,clientes);
        System.out.println("Para hacer efectivo el Registro cierre el programa y vuelvalo a Abrir");
    }
}
```

Se genera el while donde se repetirá hasta que sea encontrado un usuario o hasta que se registre, además para ingresar como "ADMIN" es con un simple if y dentro del segundo "if" se envía al usuario a "admin()" el que recibirá el Scanner, lista productos, lista noProductos, lista stockProductos, lista precio (en realidad solo se llega a usar el Scanner, los demás es solo para llevar un orden de que datos se usaran

En esta parte se niega la variable "encontrado" ya que cuando se encuentra al usuario o al Admin el valor cambia a falso, y para que entre al un if debe ser verdadera

## Orden del código(admin() linea 55 – 115 \_ COMPLETADO 100%)

```
public static void admin(Scanner leer, File productos,
                        String[] nomProductos, int[] totalStock, int[] precio) throws Exception{

    int opcion;
    boolean ciclo = true;//Boolean que sirve para detener el while
    int tamAnteriorProducto = tamañoArchProductos;
    //variable para verificar si se hicieron cambios en el arch y así actualizar los datos

    System.out.println("#####");
    System.out.println("Bienvenido Admin");
    System.out.println("#####");

    while(ciclo){
        System.out.println("")

        Opciones Disponibles
        1) Bloquear Usuario
        2) Ver Historial de Compras
        3) Agregar Producto
        4) Agregar Stock
        5) Actualizar Datos
        6) Actualizar Precios
        7) Salir

        """);
        System.out.print("Ingrese su opcion ==> ");
        opcion = leer.nextInt();

        if (opcion == 1){

        }
        else if (opcion == 2){
            ventas();
        }
        else if (opcion == 3){
            tamañoArchProductos++;
            agregarProducto(leer, productos);
            tamañoFijoProductos++;
        }
        else if (opcion == 4){
            addStock(leer);
        }
        else if (opcion == 5){
            if(tamAnteriorProducto < precioProducto.length){
                txt.añadirATxtProductos(archProductos, productosStock, precioProducto, unidadesDisponibles);
                tamañoFijoProductos = tamañoArchProductos;
            }
        }
        else if (opcion == 6){
            updatePrice(leer);
            System.out.println(Arrays.toString(precioProducto));
        }
        else if (opcion == 7){
            ciclo = false;
        }
        else{
            System.out.println("Opcion no valida!");
        }
    }
}
```

El menú de Admin se divide en 7 opciones:

Primera	es	bloquearUsuarios()
Segunda	es	ventas()
Tercera	es	agregarProducto()
Cuarta	es	addStock()
Quinta	es	actualizar archTxt()
Sexto	es	updatePrice()
Séptimo	es	salir del menu Admin()

Si se ingresa un numero distinto de x<1 o x>7 se mostrará por pantalla que la opción no es valida

## Orden del código(admin())\_bloquearUsuario() COMPLETADO 100%)

```
public static void bloquearUsuario(Scanner leer){
```

```
    usuarioAux = usuario;  
    passwordAux = password;  
    saldoAux = saldo;  
    correosAux = correos;
```

Antes de empezar a manipular los arreglos las listas son guardadas en arreglos auxiliares

```
    System.out.println("USUARIOS: "+ Arrays.toString(usuarioAux));  
    int tamañoLista = usuarioAux.length;  
    int indice = 0, agregar = 0;
```

Es mostrado por pantalla los usuarios existentes

```
    System.out.println("Ingrese el nombre del usuario que bloqueará: ");  
    String userBloqueado = leer.next();
```

```
    boolean encontrado = true;
```

```
    for (int i = 0; i < usuarioAux.length; i++){  
        if(usuarioAux[i].equals(userBloqueado)){
```

```
            encontrado = false;
```

```
            usuario = new String[tamañoLista - 1];  
            password = new String[tamañoLista - 1];  
            saldo = new int [tamañoLista - 1];  
            correos = new String[tamañoLista - 1];  
            indice = i;
```

Si es encontrado el usuario buscado recién son creadas las listas vacías en donde se guardarán todos los usuarios menos el que se eliminará, el tamaño de la nueva lista es el tamaño de la lista auxiliar menos 1, además se guarda la posición del usuario buscado en la variable "índice"

```
            for (int listaNueva = 0; listaNueva < usuarioAux.length; listaNueva++){  
                if(indice != listaNueva){  
                    usuario[agregar] = usuarioAux[listaNueva];  
                    password[agregar] = passwordAux[listaNueva];  
                    saldo[agregar] = saldoAux[listaNueva];  
                    correos[agregar] = correosAux[listaNueva];
```

```
                    agregar++;
```

Se agregarán todos menos el que coincida entre la variable listaNueva e indice, la lista dominante es "usuarios[ ]" dependiendo de ella es por la que las demás eliminaran su dato vinculado al usuario. Se usa la variable agregar para poder añadir de 0 hasta "N"

```
                }  
            }  
        }  
    }
```

```
    if (encontrado){  
        System.out.println("Usuario no encontrado!! ");  
    }else{  
        System.out.println("Nueva lista de usuarios: "+ Arrays.toString(usuario));  
    }  
}
```

## Orden del código(admin().\_ventas() línea 172 - 180 COMPLETADO 80%)

```
public static void ventas(){
    System.out.println("");
    System.out.println("Nombre Producto | Veces vendido");
    for (int i = 0; i< productosVentas.length;i++){
        System.out.println("#####");
        System.out.println(productosVentas[i]+" :      "+unidadesVendidas[i]);
    }
}
```

Se muestra por pantalla los productos vendidos, pero no está vinculado con el menuUsuario, por lo que solo mostrará lo que hay en arch txt

## Orden del código(admin().\_agregarProducto() línea 201- 232 COMPLETADO 100%)

```
public static void agregarProducto(Scanner leer, File productos) throws Exception{
    System.out.println("Ingrese el nombre del producto: ");
    String debugg = leer.nextLine();
    String nomProducto = leer.nextLine();
    System.out.println("Ingrse el precio (USD) del producto: ");
    int precio = leer.nextInt();
    System.out.println("Ingrese el stock: ");
    int stock = leer.nextInt();

    ///////////////////////////////////
    productosStockaux = new String[tamañoFijoProductos - 1];
    productosStockaux = productosStock;
    ///////////////////////////////////
    precioProductoaux = new int[tamañoFijoProductos - 1];
    precioProductoaux = precioProducto;
    ///////////////////////////////////
    unidadesDisponiblesaux = new int[tamañoFijoProductos - 1];
    unidadesDisponiblesaux = unidadesDisponibles;
    ///////////////////////////////////

    productosStock = new String[tamañoArchProductos];
    precioProducto = new int[tamañoArchProductos];
    unidadesDisponibles = new int[tamañoArchProductos];

    productosStock = pasoTXTSts(productosStockaux,productosStock);
    precioProducto = pasoTXTint(precioProductoaux,precioProducto);
    unidadesDisponibles = pasoTXTint(unidadesDisponiblesaux,unidadesDisponibles);

    productosStock[tamañoArchProductos-1] = nomProducto;
    precioProducto[tamañoArchProductos-1] = precio;
    unidadesDisponibles[tamañoArchProductos-1] = stock;
}
```



## Orden del código(admin().\_addStock() línea 183 - 198 COMPLETADO 100%)

```
public static void addStock(Scanner leer) throws Exception {  
  
    System.out.println("Ingrese el nombre del producto");  
    leer.nextLine();  
    String nameProducto = leer.nextLine();  
    int canti = 0, suma;  
    for (int i = 0; i < productosStock.length; i++){  
        if(nameProducto.equals(productosStock[i])){  
            System.out.println("Ingrese cuanto quiere agregar: ");  
            canti = leer.nextInt();  
            suma = canti + unidadesDisponibles[i];  
            unidadesDisponibles[i] = suma;  
            break;  
        }  
    }  
}
```

## Orden del código(admin())\_txt.añadirATxtProductos() línea 13 - 30 COMPLETADO 100%)

```
public static void añadirATxtProductos(File productos, String [] nomProductos,
                                     int[] precio, int[] stock) throws IOException
{
    String ruta = productos.getAbsolutePath();
    FileWriter arch = new FileWriter(ruta);

    int ultimaLinea = 0;
    ultimaLinea = nomProductos.length - 1;

    for (int i = 0; i < nomProductos.length; i++){

        if (ultimaLinea == i){
            arch.write(nomProductos[i]+","+precio[i]+","+stock[i]);
        }else{
            arch.write(nomProductos[i]+","+precio[i]+","+stock[i]+"\\n");
        }
    }
    arch.close();
}
```

Este módulo sobrescribe el archivo txt productos.

El if que esta dentro del for sirve para poder escribir la última línea sin salto de línea y así poder agregar más datos la siguiente vez

## Orden del código(admin())\_updatePrice() línea 236 - 251 COMPLETADO 100%)

```
public static void updatePrice(Scanner leer) throws Exception {

    //aer borrrrar = new aer();
    System.out.println("Ingrese el nombre del producto");
    leer.nextLine();
    String nameProducto = leer.nextLine();
    int nuevoPrecio = 0;
    for (int i = 0; i < productosStock.length; i++){
        if(nameProducto.equals(productosStock[i])){
            System.out.println("Ingrese el nuevo Precio: ");
            nuevoPrecio = leer.nextInt();
            precioProducto[i] = nuevoPrecio;
            break;
        }
    }
}
```

Esta función recibe el arreglo productosStock[] y el arreglo precioProducto[ ], luego busca por la lista productosStock y cuando es encontrado cambia el valor de precio gracias a la variable "i"

**Orden del código**(admin().\_cerrarCiclo: boolean, COMPLETADO 100%)

```
else if (opcion == 7){  
    ciclo = false;  
}
```

## Orden del código(userMenu() línea 118 - 170 COMPLETADO 60%)

```
public static void userMenu (String usuarioEntrada, Scanner leer , File productos,
                             String[] pass          , File clientes, int indice    ) throws Exception{
    System.out.println("Bienvenido [%s]".formatted(usuarioEntrada));
    System.out.println("");
    System.out.println("#####");
    boolean confirmacionWhile = true;
    int opcion;
    while(confirmacionWhile){
        System.out.println("""

        Ingresa tu opcion:
        1)Elige un Producto
        2)Cambia tu contraseña
        3)Ver catalogo de Productos
        4)Ver tu Saldo [USD]
        5)Recargar Saldo
        6)Ver Carrito
        7)Quitar productos del Carrito
        8)Pagar Carrito
        9)Cerrar Sesion
        """);
        System.out.print("==> ");
        opcion = leer.nextInt();
        if (opcion >= 1 && opcion<=9){
            if(opcion == 1){

            }
            if(opcion == 2){
                cambiaContraseña(leer,indice);
            }
            if(opcion == 3){
                mostrarCATalogo(productos);
            }
            if(opcion == 4){
                System.out.println("Tu saldo es: $" +saldo[indice]);
            }
            if(opcion == 5){
                saldo = recargarSaldo(leer, saldo, indice);
                System.out.println("Saldo recargado");
            }
            if(opcion == 6){}
            if(opcion == 7){}
            if(opcion == 8){}
            if(opcion == 9){
                txt.añadirATxtClientes(archClientes, usuario, password, saldo, correos);
                confirmacionWhile = false;
            }
        }else{
            System.out.println("#####");
            System.out.println("Opcion no Valida! intenta denuevo");
            System.out.println("#####");
        }
    }
}
```

**Orden del código**(userMenu().\_elegirProducto () NO COMPLETADO )

## Orden del código(userMenu()\_cambiaContraseña () línea 453 - 475 COMPLETADO 100%)

```
public static void cambiaContraseña(Scanner leer,int indice){
    boolean valorTrue = true;
    String anteriorPASS="", nuevaContra = "";
    System.out.println(Arrays.toString(password));
    while(valorTrue){
        System.out.println("Ingrese su anterior contraseña: ");
        anteriorPASS = leer.next();
        if(anteriorPASS.equals(password[indice])){
            System.out.println("Ingrese la nueva contraseña (Menor a 10 caracteres) ");
            nuevaContra = leer.next();
            if(nuevaContra.length() > 0 && nuevaContra.length() <= 10){
                password[indice] = nuevaContra;
                System.out.println("Contraseña guardada ");
                valorTrue = false;
            }else{
                System.out.println("Contraseña no valida, intentalo denuevo");
            }
        }else{
            System.out.println("Contraseña no coincide con la anterior!!");
        }
    }
    System.out.println(Arrays.toString(password));
}
```

## Orden del código(userMenu()\_mostrarCatalogo() línea 419 - 441 COMPLETADO 100%)

```
public static void mostrarCAtalogo(File archProductos) throws Exception{
    @SuppressWarnings("resource") // Se usa solo para retirar el aviso del Scanner
    Scanner arch = new Scanner(archProductos);
    System.out.println("#####");
    System.out.println(" ");
    System.out.println("Lista de Productos en Stock: ");
    System.out.println(" ");
    while (arch.hasNextLine()){
        String linea = arch.nextLine();
        String partes[] = linea.split(",");
        //Definiendo partes para mostrarlas por pantalla
        String producto    = partes[0],
               precio      = partes[1],
               unidadesDis  = partes[2];
        //Salida por Pantalla
        System.out.print("""
            #####
            Producto : %s
            Precio   : $%s USD
            Unidades : %s
            """).formatted(producto,precio,unidadesDis));
    }
}
```

**Orden del código**(userMenu().\_mostrarSaldo() línea 151 - 153 COMPLETADO 100%)

```
if(opcion == 4){  
    System.out.println("Tu saldo es: $" + saldo[indice]);  
}
```

**Orden del código**(userMenu().\_addStock() línea 183 - 198 COMPLETADO 100%)

```
public static int[] recargarSaldo (Scanner leer, int[] saldo, int indice){  
    int agregar;  
    System.out.println("Cuanto quiere recargar? ");  
    agregar = leer.nextInt();  
    saldo[indice] += agregar;  
    return saldo;  
}
```

**Orden del código**(userMenu().\_verCarrito() NO COMPLETADO)

**Orden del código**(userMenu().\_quitarProductoCarrito() NO COMPLETADO)

**Orden del código**(userMenu().\_pagarCarrito() NO COMPLETADO)



## Orden del código(userMenu().\_cerrarSesion() COMPLETADO 100%)

```
if(opcion == 9){  
    txt.añadirATxtClientes(archClientes, usuario, password, saldo, correos);  
    confirmacionWhile = false;  
}
```

## Orden del código(registro() COMPLETADO 100%)

```
public static void Registro(Scanner leer, File archCliente) throws IOException{

    System.out.println("#####");
    System.out.println("Registro: ");
    System.out.println("#####");
    boolean finCiclo = true;
    String contra = " ", user, correo;
    System.out.println("Cree su nombre de usuario: ");
    user = leer.next();
    System.out.println("Cree su contraseña (Maximo 10 caracteres)");

    while(finCiclo){
        contra = leer.next();
        if (contra.length() > 0 && contra.length() <= 10){
            System.out.println("Contraseña Designada");
            finCiclo = false;
        }else{
            System.out.println("Contraseña no valida ingrese una nuevamente!!");
            System.out.println("Cree su contraseña (Maximo 10 caracteres)");
        }
    }
    BufferedWriter addUser = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(archCliente,true)));
    System.out.println("Por ultimo ingrese su correo electronico: ");
    correo = leer.next();
    addUser.write("\n");
    addUser.write(user + "," + contra + ",0," + correo );
    addUser.close();
}
```

## Orden del código(Variables y arreglos usados)

```
// Arch Clientes
public static File archClientes = new File("Clientes.txt");
// Arch Productos
public static File archProductos = new File("Productos.txt");
// Arch Ventas
public static File arhVentas = new File("Ventas.txt");

//Variable que nos sirven para darle tamaño a las listas que contendran los datos del arch txt
public static int tamañoArch;//tamaño de lista usuario
public static int tamañoArchProductos;//tamaño de lista producto
public static int tamañoArchVentas;//tamaño de lista ventas

//Listas dediacadas para el arch txt Productos
public static String productosStock [];
public static int precioProducto[];
public static int unidadesDisponibles[];

public static String productosStockaux [];
public static int precioProductoaux[];
public static int unidadesDisponiblesaux[];

public static int tamañoFijoProductos;
public static int tamAnteriorCliente ;

//Listas dediacadas para el arch txt Clientes
public static String usuario [];
public static String usuarioAux[];
public static String password [];
public static String passwordAux[];
public static int saldo [];
public static int saldoAux[];
public static String correos [];
public static String correosAux[];

//Listas dediacadas para el arch txt Ventas
public static String productosVentas [];
public static int unidadesVendidas [];
```