



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 100%**

[bloquearUsuario\(\)](#)

[ventas\(\)](#)

[agregarProducto\(\)](#)

[addStock\(\)](#)

[txt.añadirATxtProductos\(\)](#)

[updatePrice\(\)](#)

[cerrarCiclo: boolean](#)

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

[elegirProducto\(\)](#)

[cambiaContraseña\(\)](#)

[mostrarCatalogo\(\)](#)

[mostrarSaldo\(\)](#)

[recargarSaldo\(\)](#)

[verCarrito\(\)](#)

[quitarProducCarrito\(\)](#)

[pagarCarrito\(\)](#)

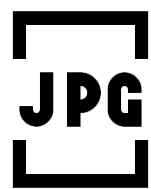
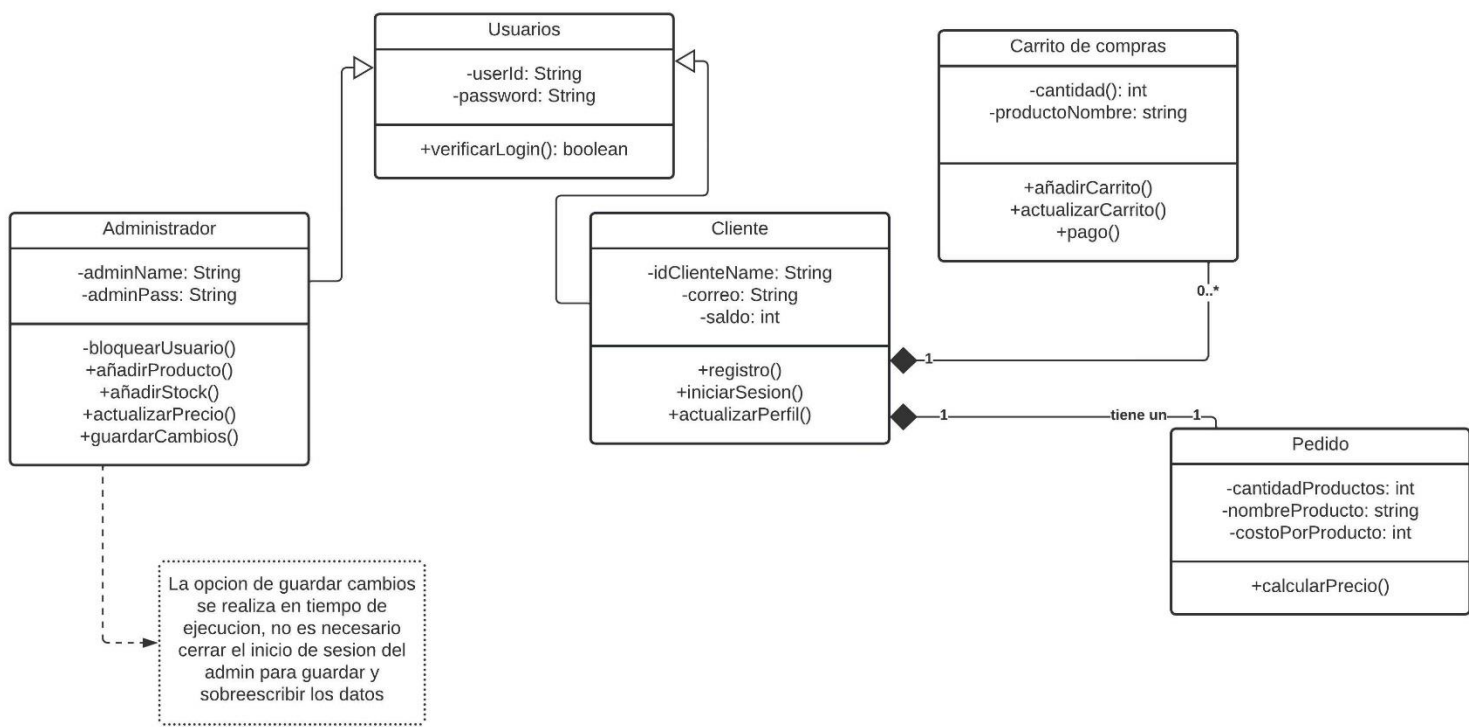
[txt.añadirATxtCientes\(\)](#)

[Registro\(\)](#)

---- funcionesComplementarias()

----- Notas Importantes

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();
    }
}
```

Orden del código(inicioSesion(): linea 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");
}
}
```

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!");
        }
    }
}
```

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

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

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

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

    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;

            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++;
                }
            }
        }
    }
    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 100%)

```
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]);
    }
}
```

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();
}
```

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

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

    //aer borrrar = 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;
        }
    }
}
```

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();
}
```