

Taller 02

Elias Manque Olivare

Programación Avanzada

Paralelo C2

Índice

Diagrama de interfaces graficas

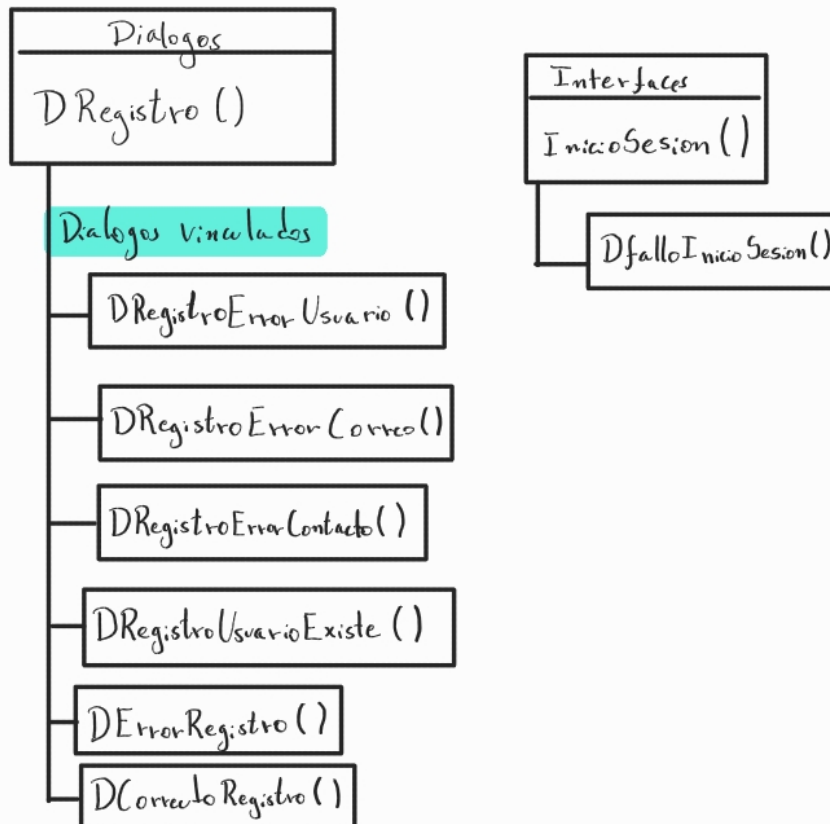
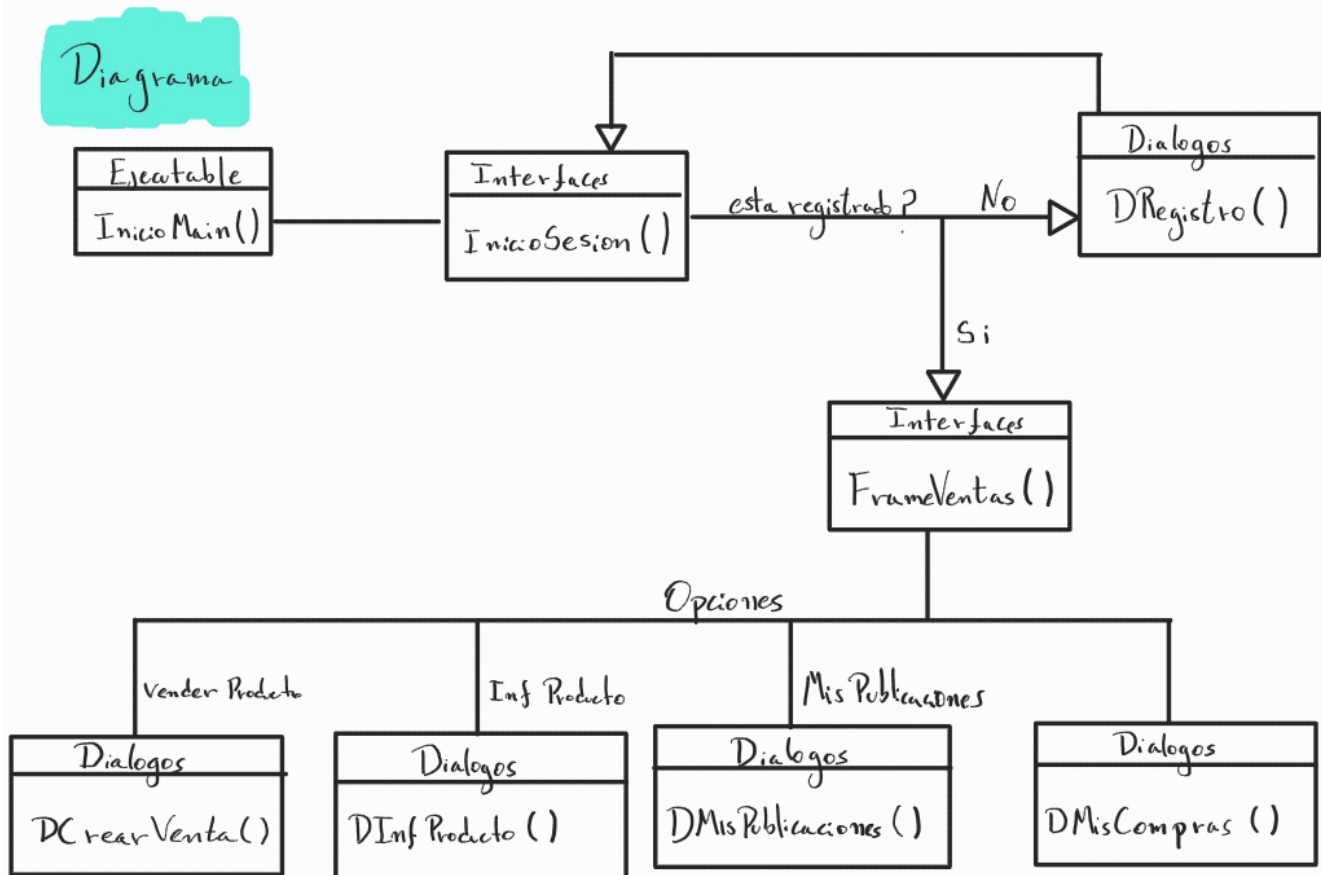
Contratos

Diagrama de Clases

Modelo de Dominio

Código

Diagrama de interfaces graficas



Contratos

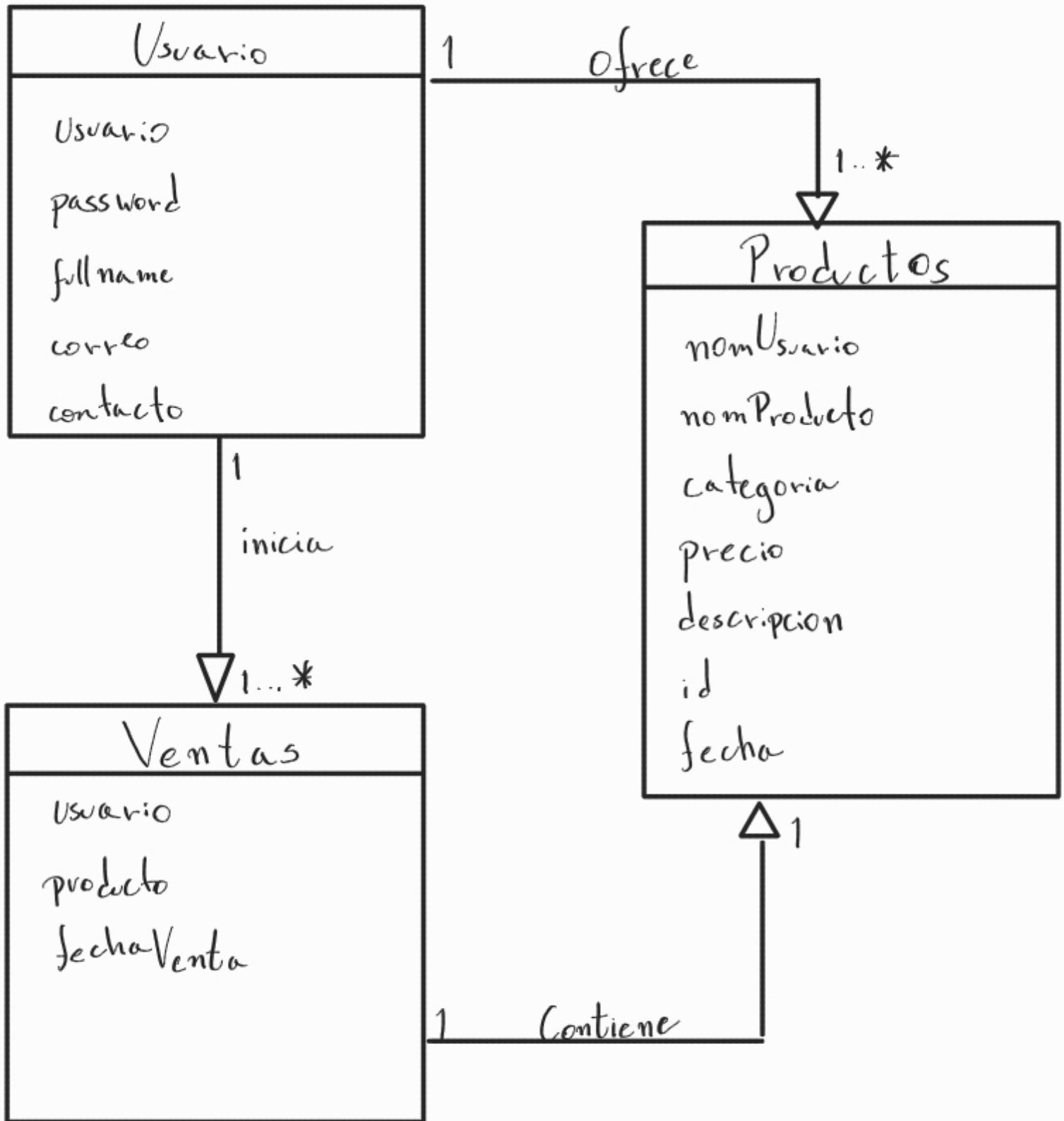
DRegistro()

Nombre	registro() Parámetros de entrada: Nombre Usuario, Nombre Completo, Correo, Contacto , Contraseña, Confirmar Contraseña
Descripción	El sistema debe permitir ingresar los datos para registrar a un nuevo usuario
Precondición	1. Que el usuario no este registrado antes
Postcondición	1. Existe un nuevo usuario en el sistema

InicioSesion()

Nombre	inicioSesion() Parámetros de entrada: (ninguno)
Descripción	El sistema busca al usuario y da acceso
Precondición	1. Que el usuario esté registrado antes 2. Que la contraseña coincida con la registrada
Postcondición	1. Existe un nuevo usuario en el sistema

Diagrama de Clases



Archivos .txt Usados



Forma de leer el archivo “usuarios.txt”

Este archivo contiene todos los datos de los usuarios registrados:

Nombre de usuario [0], contraseña [1], nombre completo [2], email[3] y su contacto[4]



Forma de leer el archivo “productos.txt”

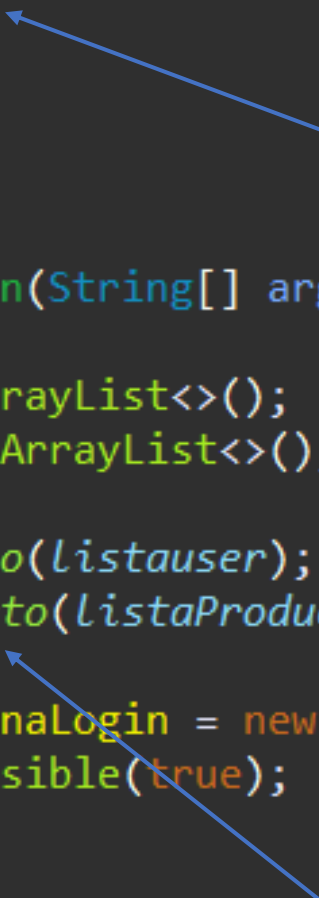
Este archivo contiene todos los datos de los productos publicados por los usuarios registrados:

Nombre de usuario que vende le producto [0], nombre del producto [1], categoría a la que pertenece el producto [2], valor del producto [3], una descripción del producto [4], fecha de publicación (esta fecha es sacada del computador que se este usando [5] y por ultimo la referencia a la imagen que suba el vendedor (esta última opción no fue implementada)



Código (package "Ejecutable", Clase "InicioMain")

```
public class InicioMain {  
  
    private static ArrayList<Productos> listaProduc;  
    private static ArrayList<Usuario> listauser;  
  
    public static void main(String[] args) throws IOException {  
  
        listauser = new ArrayList<>();  
        listaProduc = new ArrayList<>();  
  
        agregarListaUsuario(listauser);  
        agregarListaProducto(listaProduc);  
  
        InicioSesion ventanaLogin = new InicioSesion();  
        ventanaLogin.setVisible(true);  
  
    }  
}
```



Creación de ArrayList de tipo Usuario y Productos

Procedimiento para llenar las listas leyendo un arch .txt

```
public static void agregarListaUsuario(ArrayList<Usuario> lista) throws IOException {
```

```
    File arch = new File("usuarios.txt");
```

```
    if (!arch.exists()) {  
        arch.createNewFile();  
    }
```

```
    Scanner leer = new Scanner(arch);
```

```
    while (leer.hasNextLine()) {  
        String linea = leer.nextLine();  
        String[] datos = linea.split(",");
```

```
        String usuario = datos[0];  
        String password = datos[1];  
        String fullName = datos[2];  
        String correo = datos[3];  
        String contacto = datos[4];
```

```
        Usuario u = new Usuario(usuario, password, fullName, correo, contacto);  
        lista.add(u);
```

```
    }  
    leer.close();
```

```
}
```

```
public static void agregarListaProducto(ArrayList<Productos> lista) throws IOException {
```

```
    File arch = new File("productos.txt");
```

```
    if (!arch.exists()) {  
        arch.createNewFile();  
    }
```

```
    Scanner leer = new Scanner(arch);
```

```
    while (leer.hasNextLine()) {  
        String linea = leer.nextLine();  
        String[] datos = linea.split(",");
```

```
        String usuario = datos[0];  
        String producto = datos[1];  
        String categoria = datos[2];  
        int precio = Integer.parseInt(datos[3]);  
        String descripcion = datos[4];  
        int id = Integer.parseInt(datos[5]);  
        String fecha = datos[6];
```

```
        Productos p = new Productos(usuario, producto, categoria, precio, descripcion, id, fecha);  
        lista.add(p);
```

```
    }  
    leer.close();
```

```
}
```

Se abre el archivo y se crea una variable de tipo String llamada "línea" la cual contiene toda la línea de texto del archivo (arch) para luego separar en un arreglo

Se instancia una variable de tipo Usuario que use esos parámetros para ser agregado a la ArrayList (listauser)

Se abre el archivo y se crea una variable de tipo String llamada "línea" la cual contiene toda la línea de texto del archivo (arch) para luego separar en un arreglo

Codigo (package "Interfaces", JFrame "InicioSesion")

```
public class InicioSesion extends JFrame{

    private JPanel contentPane;
    private JTextField user;
    private JPasswordField passwordField;

    protected static ArrayList<Productos> listaProduc = InicioMain.getListaProduc();
    protected static ArrayList<Usuario> listauser = InicioMain.getListauser();
    private static String usuarioInicio;
    private static int index = 0;
```

La variable usuarioInicio e index, de tipo String e int respectivamente, son usados como referencia de la posición en que se encuentra el usuario para no volverlo a buscar en los siguientes JFrame

```
JButton botonIngresar = new JButton("Ingresar");
botonIngresar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        String usuario = user.getText();

        @SuppressWarnings("deprecation")
        String password = passwordField.getText();

        if(procesoInicioSesion.inicioSesionI(usuario,password)) {
            usuarioInicio = usuario;
            index = procesoInicioSesion.indiceUsuario(usuario);
            FrameVentas ventanaPrincipal = new FrameVentas();
            FrameVentas.llenarTabla();
            ventanaPrincipal.setVisible(true);
            dispose();
        }
    }
});
```

Se recuperan los datos ingresados por los usuarios los cuales son enviados a la clase "procesolnicioSesion"

"procesolnicioSesion" devuelve un boolean, si es encontrado y si su contraseña coincide se le da acceso al sistema

Acá se le da el valor de "usuario" a "usuarioInicio" y a "index" se le da la posición de donde se encuentra el usuario en la lista "listauser"

Código (package "Lógica", clase "procesoInicioSesion")

```
public class procesoInicioSesion {  
  
    protected static ArrayList<Productos> listaProduc = InicioMain.getListProduc();  
    protected static ArrayList<Usuario> listauser = InicioMain.getListuser();
```

Se referencian las listas
"listasProduc" y "listauser"

```
public static boolean inicioSesionI(String usuario, String password) {
```

```
    boolean inicio = false;
```

```
    Usuario u;
```

```
    if(buscarUsuarioBoolean(usuario)) {
```

```
        u = procesoInicioSesion.buscarUsuario(usuario);
```

```
    }else {
```

```
        DfalloInicioSesion ventanaError = new DfalloInicioSesion();
```

```
        ventanaError.setVisible(true);
```

```
        return inicio;
```

```
    }
```

```
    String nombreUsuarios = u.getUsuario();
```

```
    String passwordConfirmar = u.getPassword();
```

```
    if (usuario.equals(null)) {
```

```
        DfalloInicioSesion ventanaError = new DfalloInicioSesion();
```

```
        ventanaError.setVisible(true);
```

```
    }
```

```
    else if (nombreUsuarios.equals(usuario)) {
```

```
        if(password.equals(passwordConfirmar)) {
```

```
            inicio = true;
```

```
        }else {
```

```
            DfalloInicioSesion ventanaError = new DfalloInicioSesion();
```

```
            ventanaError.setVisible(true);
```

```
        }
```

```
    }else {
```

```
        DfalloInicioSesion ventanaError = new DfalloInicioSesion();
```

```
        ventanaError.setVisible(true);
```

```
    }
```

```
    return inicio;
```

```
}
```

Valor boolean que devolverá el
método, se instancia en "false" ya
que cambiará su valor cuando los
datos ingresados coincidan con
los ingresados por el usuario

Esta primera condicional sirve
para que no lase errores al
comenzar a preguntar si la
contraseña y usuario ingresado
coinciden, y que sino existe la
variable "u" será null

Si es que existe el usuario se
consiguen las variables del
objeto "u" para instanciar las
variables "nombreUsuarios" y
passwordConfirmar", las
cuales serán usadas para
compararlas y así dar acceso al
sistema

Cambio del valor de la variable
"inicio" a true cuando "usuario"
coincide con "nombreUsuarios" y
"password" coincide con
"passwordConfirmar"

Sino llega a cambiar el valor el
boolean que se devolverá será
false pero si cambia se devolverá
true lo que hará que [package
"Interfaces", JFrame "InicioSesion"]
ejecute lo que está dentro del if

(Otros métodos de "procesoInicioSesion")

```
public static Usuario buscarUsuario(String user) {
    int i = -1;
    boolean encontrado = false;

    for(Usuario valor : listauser ) {
        i++;
        if (valor.getUsuario().equals(user)) {
            encontrado = true;
            break;
        }
    }
    if (encontrado) {
        return listauser.get(i);
    }else {
        return null;
    }
}
```

```
public static boolean buscarUsuarioBoolean(String user) {
    boolean encontrado = false;

    for(Usuario valor : listauser ) {
        if (valor.getUsuario().equals(user)) {
            encontrado = true;
            break;
        }
    }
    if (encontrado) {
        return encontrado;
    }else {
        return encontrado;
    }
}
```

```
public static int indiceUsuario(String usuario) {
    int index;
    Usuario u = procesoInicioSesion.buscarUsuario(usuario);
    index = listauser.indexOf(u);
    return index;
}
```

Codigo (package "Interfaces", JFrame "FrameVentas")

```
public class FrameVentas extends JFrame {  
  
    private JPanel contentPane;  
    private JTable listaProducto;  
    private static DefaultTableModel miTabla;  
    public static ArrayList<Productos> listaP = InicioMain.getListaproductos();  
}
```

Tabla que muestra los productos
que se venden

VentasCoquimbo - Perfil

Bienvenido [elias]

Lista de Productos disponibles

ID	Nombre	Categoria	Fecha	Visto
1	Vasos	Cocina u Hogar	2022-07-06	No
2	Mesa	Cocina u Hogar	2022-07-06	No
3	Cocina	Cocina u Hogar	2022-07-06	No
4	Zapatillas	Moda	2022-07-06	No
5	Taladro	Herramienta	2022-07-06	No
6	Poly Station	Juego	2022-07-06	No
7	Carreta antigua	Vehiculo	2022-07-07	No

Vender Producto Inf. Producto -----

Mis Publicaciones Mis Compras Filtrar

Interfaces
FrameVentas()

Dialogos
DCrearVenta()

Dialogos
DInfProducto()

Dialogos
DMisPublicaciones()

Dialogos
DMisCompras()

Dialogos

DCrearVenta()

```
public void actionPerformed(ActionEvent e) {  
  
    String nombreProducto = nomProducto.getText();  
    String nombreUsuario = InicioSesion.getUsuarioInicio();  
    String categoriaProducto = categoria.getSelectedItem();  
    int precio = Integer.parseInt(precioProducto.getText());  
    String descripcion = descripcionProducto.getText();  
    int ultimoValor = lista.size() + 1;  
    LocalDate fechaObtenida = LocalDate.now();  
    String fecha = fechaObtenida.toString();  
    System.out.println(fecha);  
  
    try {  
        if(categoriaProducto.equals("-----")) {  
            System.out.println(nombreUsuario);  
            System.out.println("Elija una categoria");  
        }  
        else if(nombreProducto.equals("")) {  
            System.out.println("Ingrese un nombre");  
        }  
        else if (!procesoRegistroProducto.agregarProducto(nombreProducto,nombreUsuario,categoriaProducto,precio,descripcion,ultimoValor,fecha)) {  
            procesoRegistroProducto.guardarTxtProducto(lista);  
            dispose();  
            DRegistroProductoCorrecto v = new DRegistroProductoCorrecto();  
            v.setVisible(rootPaneCheckingEnabled);  
        }  
    } catch (Exception e2) {  
        System.out.println(e2.getMessage());  
    }  
}
```

Se recupera toda la información que ingresa el usuario, además se usa un método get para obtener la fecha de creación de la publicación

Si se ingresan todos los datos requeridos se guarda la publicación en el archivo "productos.txt", luego se cierra automáticamente le JFrame

```
categoria = new Choice();  
categoria.setFont(new Font("Dialog", Font.PLAIN, 14));  
categoria.setBounds(204, 60, 171, 40);  
  
categoria.add("-----");  
categoria.add("Juego");  
categoria.add("Cocina u Hogar");  
categoria.add("Electrodomesticos");  
categoria.add("Herramienta");  
categoria.add("Moda");  
categoria.add("Oficina");  
categoria.add("Deporte y Fitness");  
categoria.add("Telefonia");  
categoria.add("Audio o Video");  
categoria.add("Computacion");  
categoria.add("Vehiculo");  
  
contentPanel.add(categoria);
```

Opciones disponibles en el selector de "categoría"

Nombre

Categoria

Precio

Descripcion



Registrar/Editar Publicacion **Subir Foto**

```
String nombreProducto = nomProducto.getText();
String nombreUsuario = InicioSesion.getUsuarioInicio();
String categoriaProducto = categoria.getSelectedItem();
int precio = Integer.parseInt(precioProducto.getText());
String descripcion = descripcionProducto.getText();
int ultimoValor = lista.size() + 1;
LocalDate fechaObtenida = LocalDate.now();
String fecha = fechaObtenida.toString();
System.out.println(fecha);
```

Esto nos dará el id del producto

Diálogos

DMisPublicaciones()

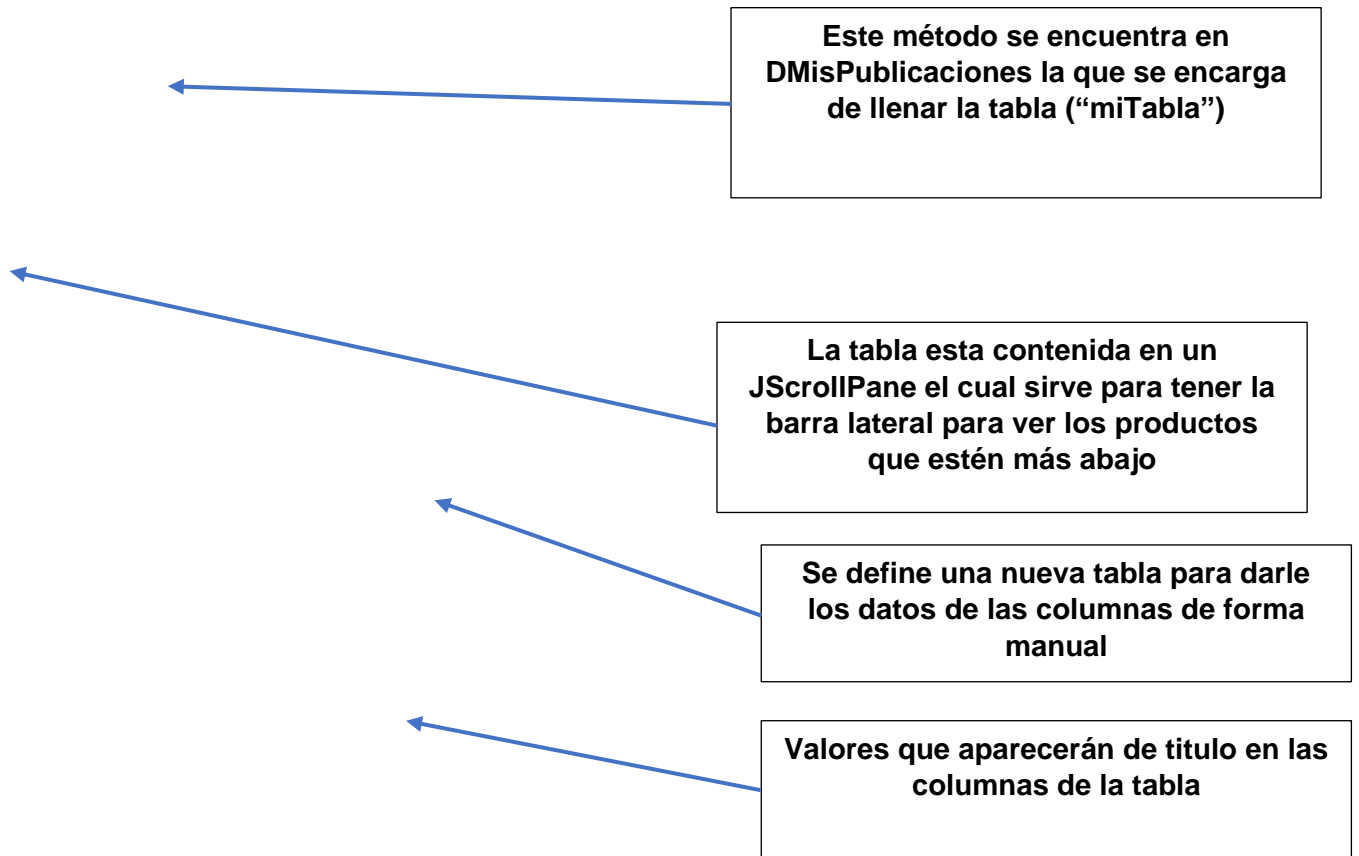
```
JButton ventanaMisProductos = new JButton("Mis Publicaciones");
ventanaMisProductos.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        DMisPublicaciones ventanaMisPublicaciones = new DMisPublicaciones();
        DMisPublicaciones.llenarTabla();
        ventanaMisPublicaciones.setVisible(rootPaneCheckingEnabled);
    }
}
```

```
public static void llenarTabla() {

    String nombre = InicioSesion.getUsuarioInicio();

    for (Productos valores : listaP) {
        Object[] filas = new Object[5];
        if (nombre.equals(valores.getNomUsuario())) {
            filas[0] = valores.getId();
            filas[1] = valores.getNomProducto();
            filas[2] = valores.getCategoria();
            filas[3] = valores.getFecha();
            filas[4] = "No";
            miTabla.addRow(filas);
        }
    }
}
```

Botón que activa
"ventanaMisPublicaciones", antes de
ingresar a la ventana primero se llena
la tabla para mostrarla lista



```
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 97, 635, 178);
contentPanel.add(scrollPane);
{

    tableMisPublicaciones = new JTable();

    miTabla = new DefaultTableModel();

    tableMisPublicaciones.setModel(miTabla);

    miTabla.addColumn("ID");
    miTabla.addColumn("Nombre");
    miTabla.addColumn("Categoria");
    miTabla.addColumn("Fecha");
    miTabla.addColumn("VedidasS_N");

    scrollPane.setViewportView(tableMisPublicaciones);
}
```

Numero: +56956789245

Nombre: elias Nicolas

Publicaciones

ID	Nombre	Categoria	Fecha	VedidasS_N
7	Carreta antigua	Vehiculo	2022-07-07	No
8	Violin	Juego	2022-07-08	No

Productos que nosotros hemos publicado

Editar Producto

Borrar Producto

Cantidad de Ventas: 2

Nos muestra el total de publicaciones realizadas tanto si se han vendido o no

```
public int totalVentas(int i) {  
    String nombre = InicioSesion.getUsuarioInicio();  
    for (Productos valores : listaP) {  
        if (nombre.equals(valores.getNomUsuario())) {  
            i++;  
        }  
    }  
    return i;  
}
```

Se busca la cantidad de veces que esta el usuario en la lista (en la lista de los productos) y la "i" sirve de contador

```
JLabel cantVentas = new JLabel("Cantidad de Ventas: " + totalVentas(totalProductos));  
cantVentas.setFont(new Font("Tahoma", Font.BOLD, 12));  
cantVentas.setBounds(216, 340, 194, 31);  
contentPanel.add(cantVentas);
```

Numero: +56956789245

Pe

ID	Nombre	
7	Carreta antigua	Vehi
8	Violin	Jueg

```
int i = InicioSesion.getIndex();
System.out.println(i);
JLabel numeroContacto = new JLabel("Numero: "+listaU.get(i).getContacto());
numeroContacto.setBounds(10, 16, 231, 25);
contentPanel.add(numeroContacto);
```

×

Nombre: elias Nicolas

	Fecha	VedidasS_N
	2022-07-07	No
	2022-07-08	No

```
JLabel nombreCompleto = new JLabel("Nombre: "+ procesoInicioSesion.buscarUsuario(InicioSesion.getUsuarioInicio()).getFullName());
nombreCompleto.setHorizontalAlignment(SwingConstants.CENTER);
nombreCompleto.setBounds(394, 10, 251, 25);
contentPanel.add(nombreCompleto);
```

