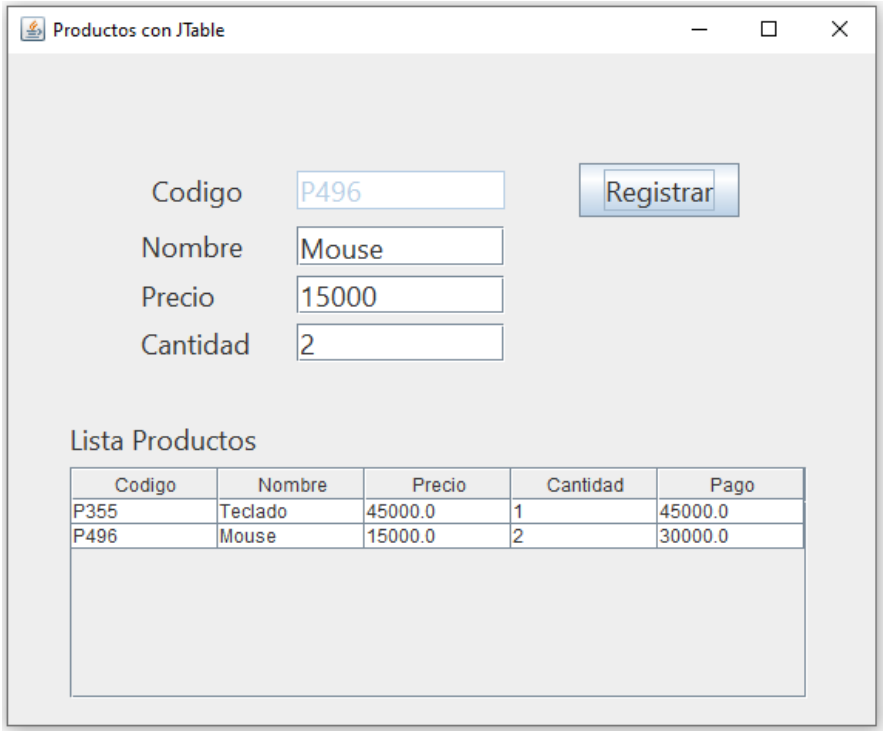
Universidad Distrital Francisco José de caldas
Tecnología en Sistematización de Datos



Aplicaciones POO y MVC
ArrayList
JTable

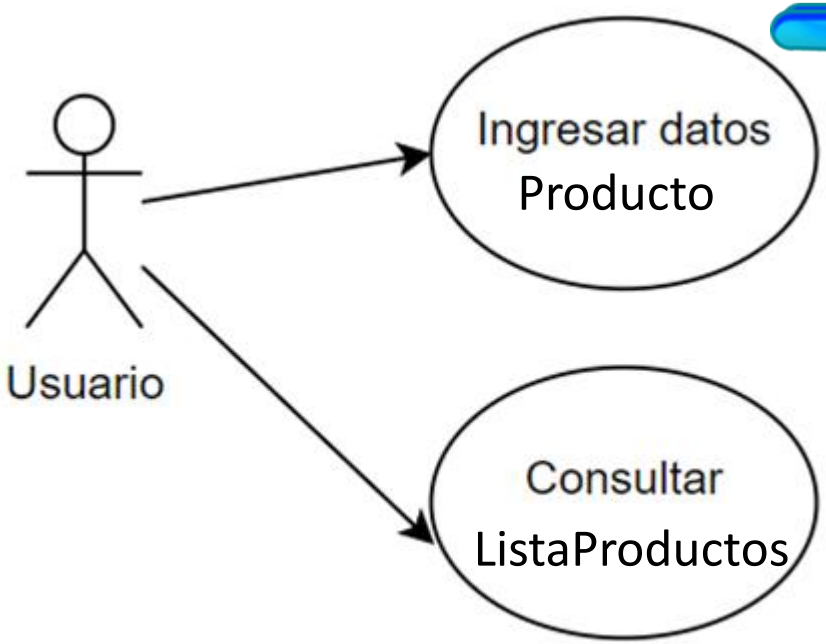
Programación Orientada a Objetos
Sonia Alexandra Pinzón Nuñez

Ejemplo Lista Productos:

Datos Producto

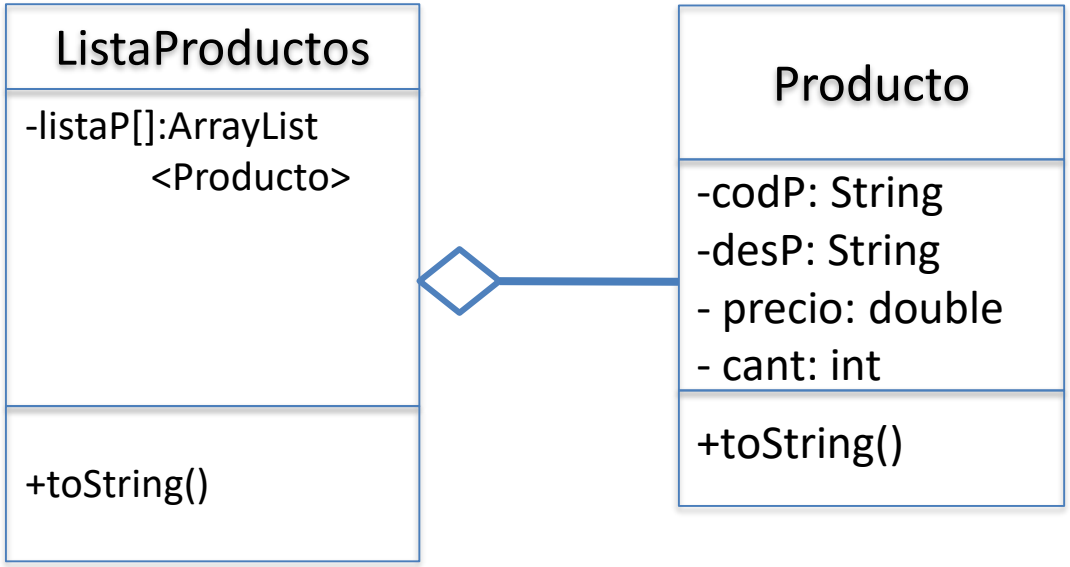
- Código
- Descripción
- Precio
- Cantidad

Ejemplo Lista Productos : Casos de Uso



Ejemplo Lista Productos : Composición de Clases

Relación en la cual una clase está compuesta o contiene como atributo un objeto de otra clase, por ejemplo, en la figura se puede observar que la capa Lógica contiene una clase denominada **ListaProductos**, que contiene un **ArrayList** con objetos de tipo **Producto** cuyos atributos son codP, desP, precio y cant.



Ejemplo Lista Productos : Objetos

Lista de Compras

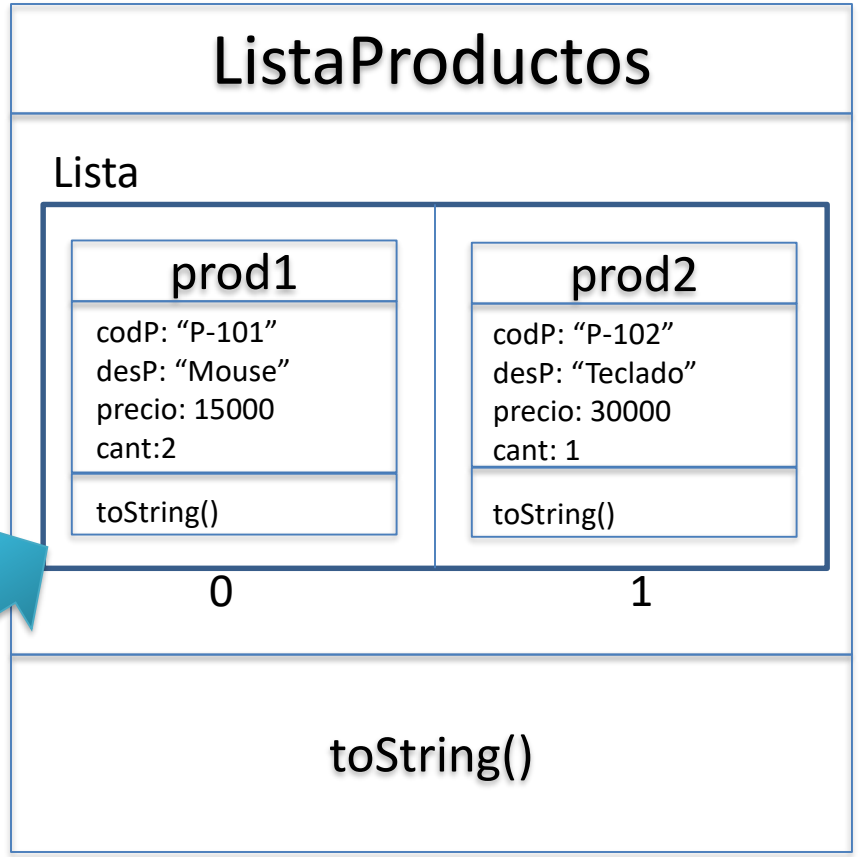
Hecho	Código	Descripción	Precio	Cantidad
☑	P-101	Mouse	15.000	2
☑	P-101	Teclado	30.000	1
☐				
☐				
☐				
☐				
☐				
☐				
☐				
☐				

TOTAL

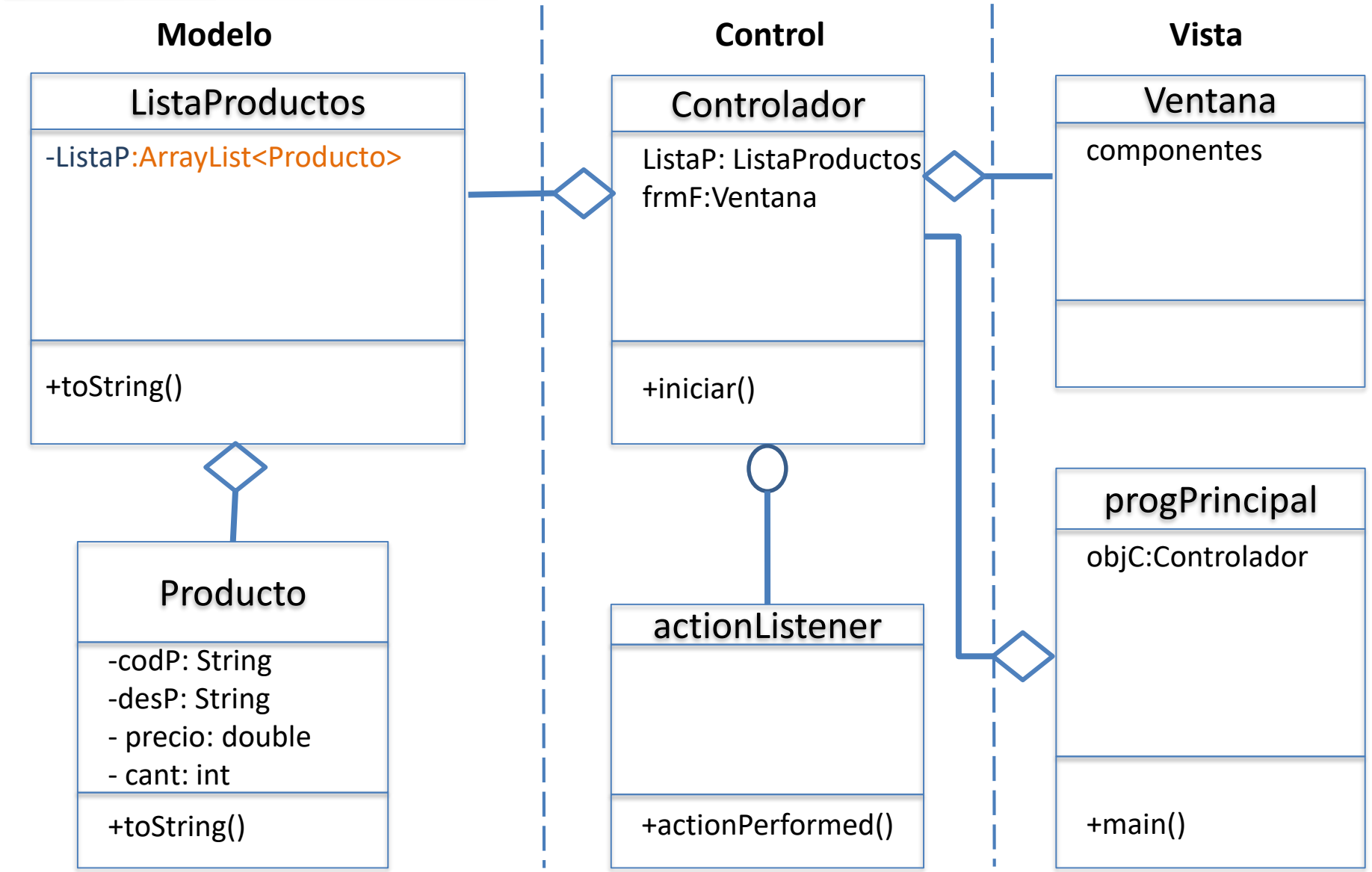
ArrayList, es una clase que permite crear un arreglo de objetos donde los elementos se almacenan de forma dinámica añadiéndose al final del arreglo, además no es necesario definir su tamaño como pasa con los Vectores.

Ej:

```
lista.add(Producto);
```



Ejemplo Lista Productos : Diagrama de Clases

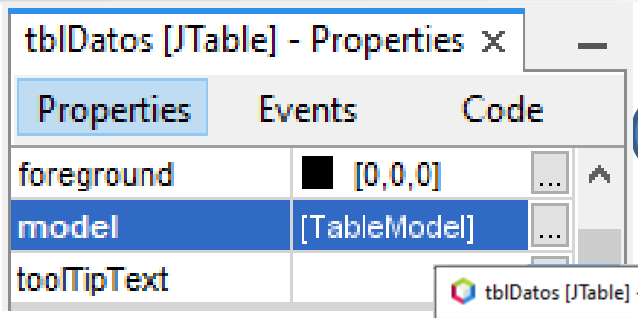


Ejemplo Lista Productos : Diseño de Interfaz -Ventana

Crear Formulario y

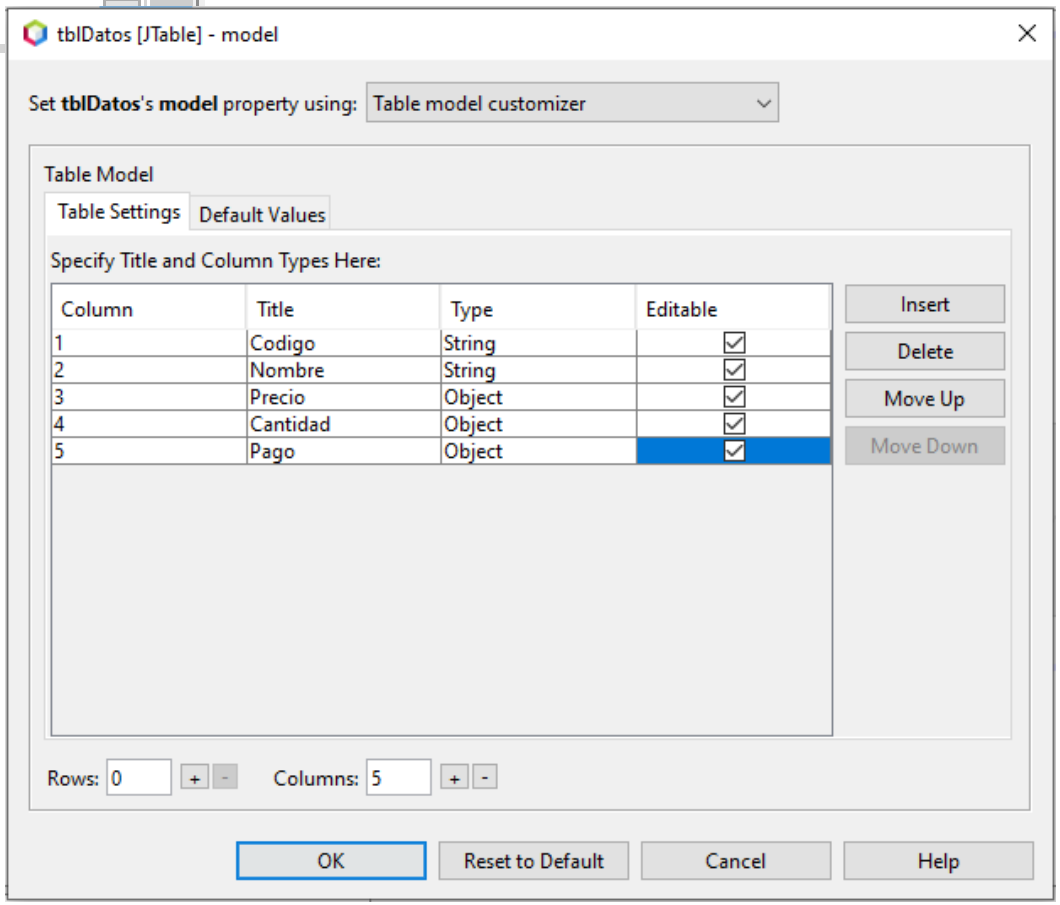
Generar Get y Set de controles

Ejemplo Lista Productos : Control JTable



1 Modificar el Modelo del JTable

2 Establecer los Títulos y tipos de valores de las celdas y cantidad de filas y columnas



Modelo : Clase Producto

Producto
-codP: String -desP: String - precio: double - cant: int
+toString()

```
public class Producto {
    protected String cod, nom;
    protected double precio;
    protected int cant;
    public Producto(String cod, String nom, double precio, int cant) {
        this.cod = cod; this.nom = nom;
        this.precio = precio;this.cant = cant;
    }
     public Producto() {
        int cod= (int)(Math.random()*999 + 100);
        this.cod = "P"+cod; this.nom = "";
        this.precio = 0;    this.cant = 0;
    }
    public double IVA(){...3 lines }
    public double valorPago(){...3 lines }
    public String getCod() {...3 lines }
    public void setCod(String cod) {...3 lines }
    public String getNom() {...3 lines }
    public void setNom(String nom) {...3 lines }
    public double getPrecio() {...3 lines }
    public void setPrecio(double precio) {...3 lines }
    public int getCant() {...3 lines }
    public void setCant(int cant) {...3 lines }
    @Override
    public String toString() {
        return "\ncodigo: " + cod + "\n nombre: " + nom +
                "\n precio: " + precio + "\n cantidad:" + cant ;
    }
}
```

Modelo : Clase ListaProductos

ListaProductos
-ListaP:ArrayList<Producto>
+toString()

```
public class ListaProductos {
    private ArrayList <Producto> ListaP;
     public ListaProductos(ArrayList<Producto> ListaP) {
         this.ListaP = ListaP;
     }
     public ListaProductos() {
         this.ListaP = new ArrayList<Producto>();
     }
     @Override
     public String toString() {
         String productos="";
         for (int i = 0; i < ListaP.size(); i++) {
             productos+= "Producto "+ (i+1)+":"+ ListaP.get(i).toString()+
                         "\nPago: "+ListaP.get(i).valorPago()+"\n";
         }
         return "ListaProductos :\n " + productos;
     }
     public ArrayList<Producto> getListaP() {...3 lines }
     public void setListaP(ArrayList<Producto> ListaP) {...3 lines }
}
```

Control : Clase Controlador

Controlador
listaP: ListaProductos frmF:Ventana
+iniciar() +agregarProductos()

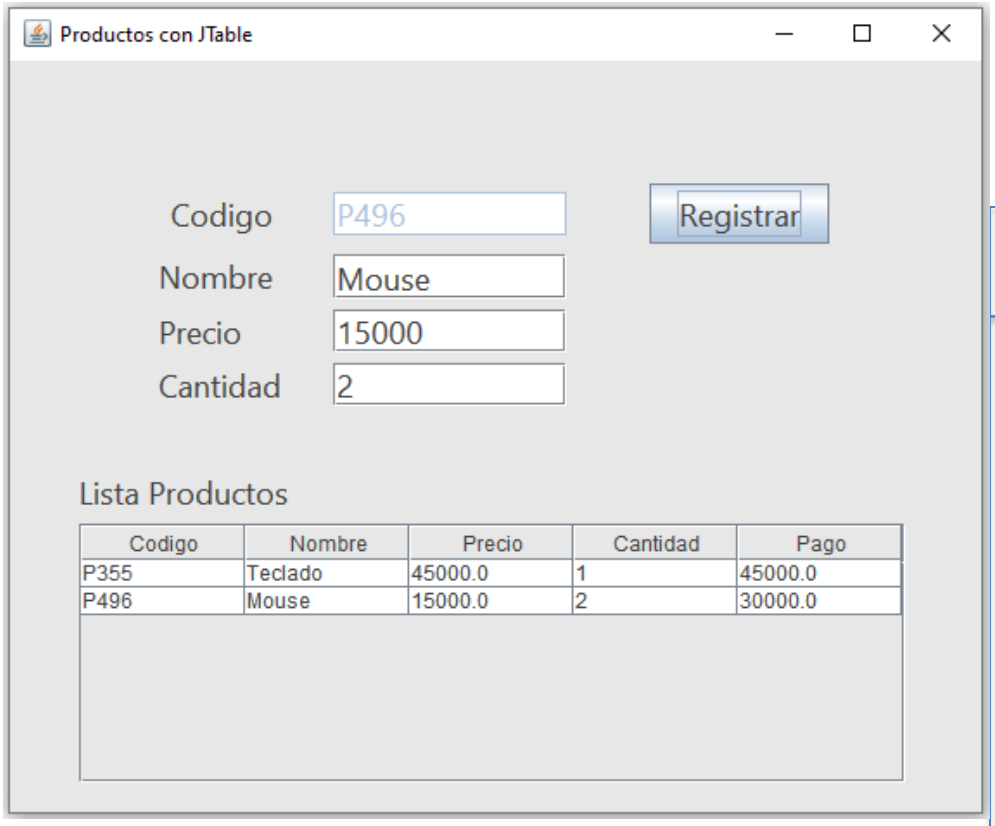
```
public class Controlador implements ActionListener{
    ListaProductos listaP;
    Ventana frmP;
    public Controlador() {...4 lines }
    public void iniciar(){
     frmP.setTitle("Productos con JTable");
     frmP.setLocationRelativeTo(null);
     frmP.getTxtCod().setEnabled(false);
     frmP.getBtnRegistrar().addActionListener(this);
     frmP.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(frmP.getBtnRegistrar())){
       Producto objP = new Producto();
       objP.setNom(frmP.getTxtNom().getText());
       objP.setPrecio(Double.parseDouble(frmP.getTxtPre().getText()));
       objP.setCant(Integer.parseInt(frmP.getTxtCant().getText()));
       frmP.getTxtCod().setText(objP.getCod());
       listaP.getListaP().add(objP);
        agregarProductos(objP, frmP.getTblDatos());
       JOptionPane.showMessageDialog(frmP,objP.toString()+
                                     "\nSub Total: "+ objP.valorPago()+
                                     "\nIVA: "+objP.IVA());
    }
    }
    public void agregarProductos(Producto prod, JTable tabla){
    Object datos[]= {prod.getCod(),prod.getNom(),prod.getPrecio(),
                     prod.getCant(), prod.valorPago()};
        DefaultTableModel plantilla= (DefaultTableModel) tabla.getModel();
        plantilla.addRow(datos);
    }
}
```

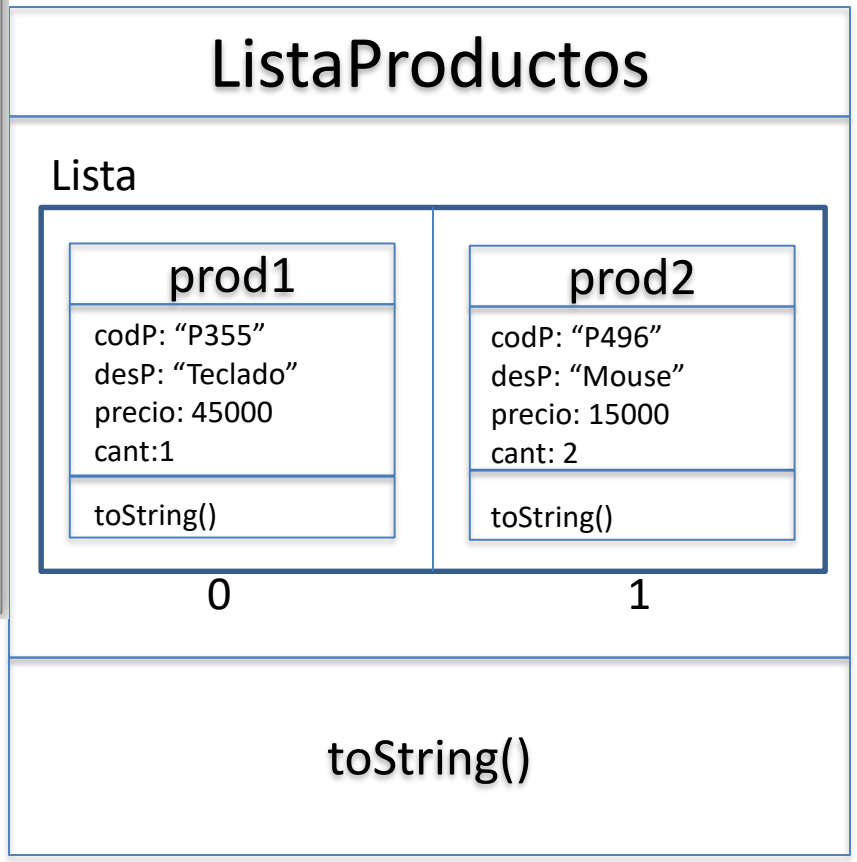
Control : Clase Programa Principal

progPrincipal
objC:Controlador
+main()

```
import control.Controlador;

public class progPricipal {

    public static void main(String[] args) {
        Controlador objC= new Controlador();
        objC.iniciar();
    }

}
```

Ejemplo Lista Productos Ejecución





Bibliografía

- Pinzón, Sonia Alexandra. Material de Clase Moodle y Drive.
- Pinzón, Sonia Alexandra. Rodríguez Guerrero, Rocío. Vanegas, Carlos Alberto. Java y el patrón Modelo- Vista – Controlador (MVC).Editorial Universidad Distrital F.J.D.C. 2021
- LADRÓN, Jorge Martínez. Fundamentos de programación en Java - 4ed. Ed. EME.Universidad Complutense de Madrid. Madrid(España), formato Digital
- Deitel y Deitel. Programación Java. Editorial Mc Graw Hill.