



Universidad Distrital Francisco José de caldas

Tecnología en Sistematización de Datos

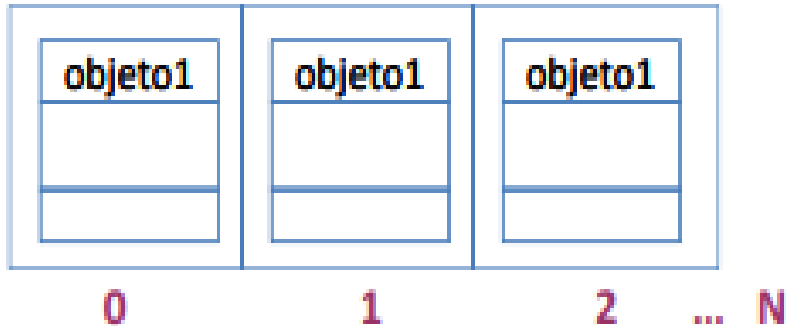


`ArrayList<Objeto> (dinámico)`

Aplicaciones POO y MVC

ArrayList

Lista Productos



Programación Multinivel
Sonia Alexandra Pinzón Nuñez



Ejemplo Lista Productos:

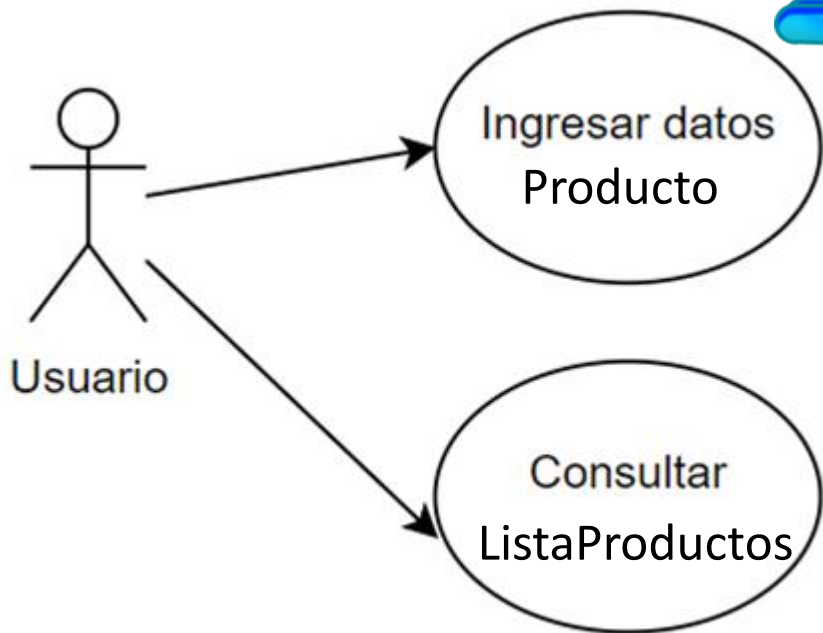
Datos Producto

Código
Descripción
Precio
Cantidad





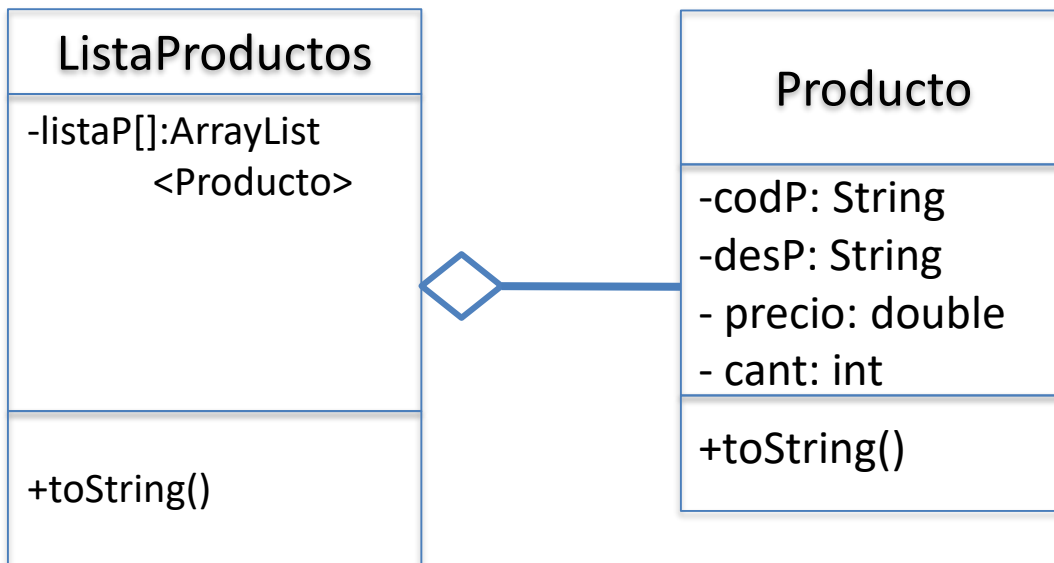
Ejemplo Agenda: Casos de Uso





Ejemplo Agenda: 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 Agenda : Objetos

Lista de Compras

Hecho	Código	Descripción	Precio	Cantidad
<input checked="" type="checkbox"/>	P-101	Mouse	15.000	2
<input checked="" type="checkbox"/>	P-101	Teclado	30.000	1
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

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);
```

ListaProductos

Lista

prod1

codP: "P-101"
desP: "Mouse"
precio: 15000
cant:2

toString()

0

prod2

codP: "P-102"
desP: "Teclado"
precio: 30000
cant: 1

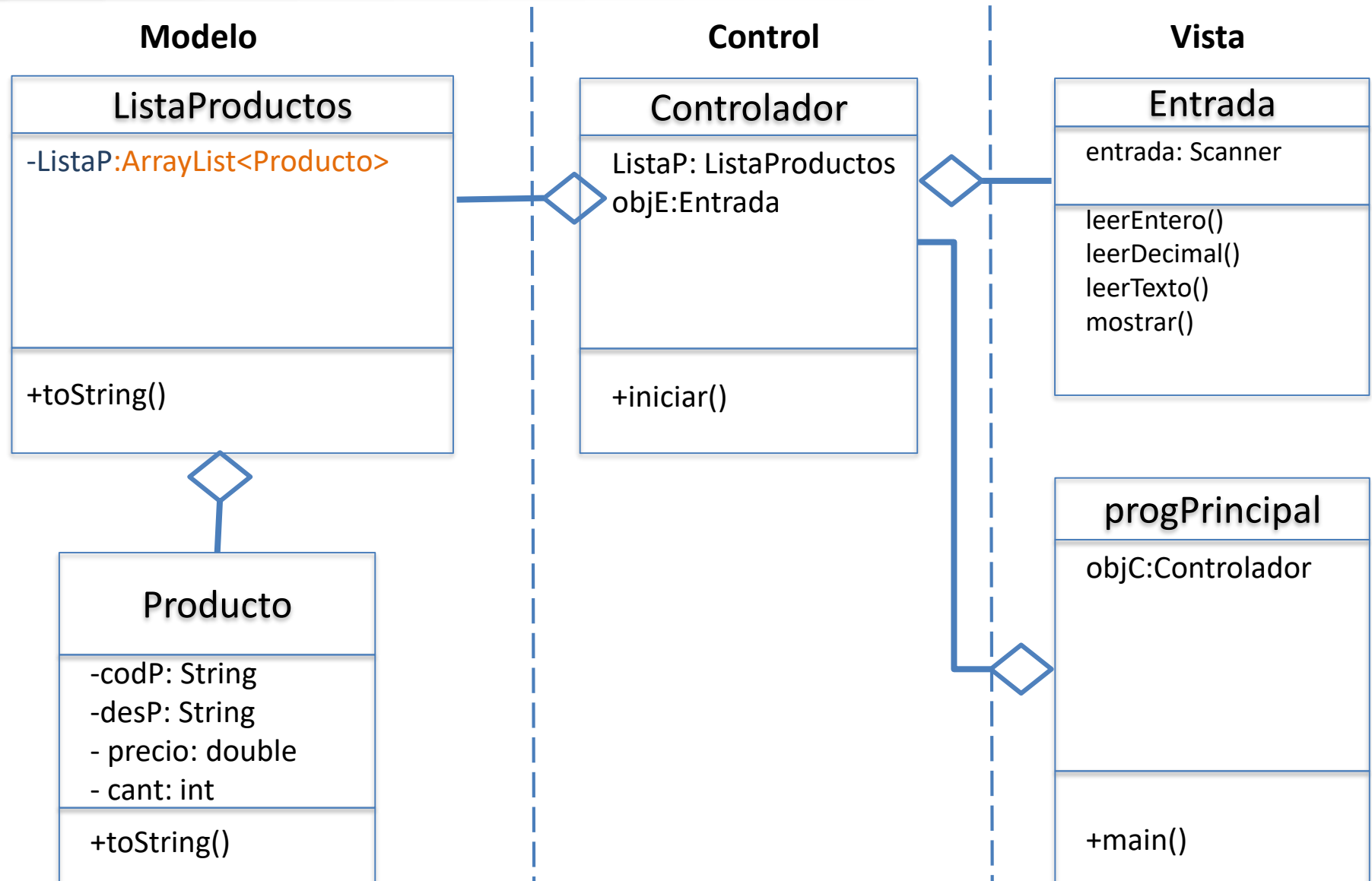
toString()

1

toString()



Ejemplo ListaProductos: Diagrama de Clases





Vista : Clase Entrada

Entrada

- entrada: Scanner
- msj:String

leerEntero()
LeerDecimal()
LeerTexto()
Mostrar()

```
package vista;
import java.util.Scanner;
public class Entrada {
    private String titulo;
    private Scanner entrada;
    public Entrada(String titulo) {
        entrada= new Scanner(System.in);
        this.titulo = titulo;
    }
    public Entrada() {
        entrada= new Scanner(System.in);
        this.titulo = "";
    }
    @Override
    public String toString() {
        return "titulo=" + titulo ;
    }
    public int leerEntero(String msj){
        System.out.println(msj);
        return entrada.nextInt();
    }
    public double leerDecimal(String msj){
        System.out.println(msj);
        return entrada.nextDouble();
    }
    public String leerTexto(String msj){
        System.out.println(msj);
        return entrada.next();
    }
    public void mostrar(String msj){
        System.out.println(titulo+"\n"+msj);
    }
    public String getTitulo() {...3 lines }
    public void setTitulo(String titulo) {...3 lines }
}
```



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

objE: Entrada

listaP: ListaProductos

+iniciar()

```
/**
 *
 * @author Sonia Pinzón
 */
public class Controlador {
    Entrada objE;
    ListaProductos listaP;
    public Controlador(Entrada objE, ListaProductos listaP) {...4 lines }
    public Controlador() {
        this.objE = new Entrada();
        this.listaP = new ListaProductos();
    }
    public void iniciar(){
        int resp=0;
        objE.mostrar("Lista de Productos");
        do{
            objE.mostrar("Datos de Producto:");
            listaP.getListP().add(new Producto(objE.leerTexto("Código: "),
                                                                    objE.leerTexto("Nombre: "),
                                                                    objE.leerDecimal("Precio: "),
                                                                    objE.leerEntero("Cantidad: ")));
            resp= objE.leerEntero("Desea registrar otro producto? 1.Si");
        }while (resp==1);
        objE.mostrar("Datos registrados: \n"+ listaP.toString());
    }
}
```



Vista : Clase Programa Principal

progPrincipal

objC:Controlador

+main()

```
import control.Controlador;
```

```
public class progPrincipal {
```

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

```
} Output - proyArrayProducto (run) x
```



```
run:  
Lista de Productos  
Datos de Producto:  
Código:  
101  
Nombre:  
mouse  
Precio:  
15000  
Cantidad:  
2  
Desea registrar otro producto? 1.Si  
1  
Datos de Producto:  
Código:  
102  
Nombre:  
teclado  
Precio:  
30000  
Cantidad:  
1  
Desea registrar otro producto? 1.Si  
2
```

```
Datos registrados:  
ListaProductos :  
Producto 1:  
codigo: 101  
nombre: mouse  
precio: 15000.0  
cantidad:2  
Pago: 30000.0  
Producto 2:  
codigo: 102  
nombre: teclado  
precio: 30000.0  
cantidad:1  
Pago: 30000.0
```

```
BUILD SUCCESSFUL (total time: 26 seconds)
```



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.