



El futuro digital  
es de todos

MinTIC

Hechos

QUE

CONECTAN



# Tutoría magistral #5

Interfaces, clases abstractas, colecciones y patrón contenedor

*Por equipo de tutores Misión TIC – UIS*

*Inicio: 6:05 PM*

Universidad  
Industrial de  
Santander



Misión  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Temario

- Clases abstractas
- Interfaces
- Listas
- Conjuntos
- Mapas
- Patrón contenedor

## Hechos

QUE

CONECTAN ✓

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Clases abstractas

Hechos

QUE

CONECTAN



Habrà ocasiones en las cuales necesitemos crear una clase **padre** donde únicamente coloquemos **la estructura** de una abstracción, una estructura muy general, dejando que sean **las clases hijas quienes definan los detalles**. En estos casos haremos uso de las clases abstractas.

Las clases abstractas pueden poseer atributos, métodos, constructores, etc. La principal diferencia entre una clase convencional y una clase abstracta es que la clase **abstracta debe poseer por lo menos un método abstracto**

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

Hechos  
QUE CONECTAN ✓

# Método abstracto

un método abstracto no es más que un **método vacío**, un método el cual **no posee cuerpo, por ende, no puede realizar ninguna acción**. La utilidad de un método abstracto es definir qué se debe hacer, pero no el cómo se debe hacer.

Universidad  
Industrial de  
Santander



Mision  
TIC 2022





# Ejemplo clase abstracta

```
public class Figura {  
  
    private int numeroLados;  
  
    public Figura() {  
  
        this.numeroLados = 0;  
    }  
  
    public float area() {  
  
        return 0f;  
    }  
  
}
```



```
public abstract float area();
```

```
public abstract class Figura {
```

Si el método área es un método abstracto la clase se convierte en una clase abstracta. Y las clases hijas deberán implementarlo.

```
public class Triangulo extends Figura {
```



# Características clases abstractas

- Definen parcialmente una clase
- En algunos de sus métodos no se conoce la implementación, así que es delegada a las subclases, mediante la definición de un método abstracto
- Una clase definida como abstracta puede contener métodos abstractos
- No se pueden crear objetos a partir de ellas
- Si una clase hereda de una clase abstracta debe implementar(lo quiera) todos los métodos que hayan sido definidos como abstractos, a menos que ella también sea una clase abstracta
- Si una clase define a lo menos un método abstracto, debe ser definida como abstracta





El futuro digital  
es de todos

MinTIC

# Interfaces

Hechos  
QUE CONECTAN ✓

A diferencia de una clase abstracta, **una interface no puede hacer nada por sí sola**, es prácticamente un contrato, en donde las clases que la implementen deben, obligatoriamente, **definir el comportamiento de todos los métodos abstractos**, contestando a la pregunta ¿Cómo se debe hacer?

Universidad  
Industrial de  
Santander



Mision  
TIC 2022





# Interfaces

```
public interface Canino{  
  
    public abstract void aullar();  
  
    public abstract void ladrar();  
  
}
```



Cómo podemos observar en la interfaz solo encontraremos métodos abstractos, métodos vacíos. Para poder implementar la interfaz basta con utilizar la palabra reservada **implements**.

```
public class Perro implements Canino {
```





# Características clases Interfaces

- Su objetivo está centrado en definir un API (Application Programming Interface) para una familia de objetos
- para su definición se cambia la palabra clave class por la palabra clave interface
- Todos sus métodos tienen visibilidad de tipo public Incluso si no se coloca el modificador de manera explícita
- No tienen atributos
- No tienen método constructor



El futuro digital  
es de todos

MinTIC

# Listas

Hechos

QUE

CONECTAN



Las listas **no son arreglos** (arrays), aunque ambos representan secuencias de elementos de un tipo, los arreglos tienen longitud fija; las listas, no; es decir, **las listas son flexibles** y permiten cambio de implementación.

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# ArrayList

Hechos

QUE

CONECTAN



ArrayList como su nombre indica la implementación de la lista en un array. Eso sí, un **array dinámico en tamaño** (es decir, de tamaño variable), pudiendo agrandarse el número de elementos o disminuirse. Implementa **todos los métodos de la interfaz List** y permite incluir elementos null.

Universidad  
Industrial de  
Santander

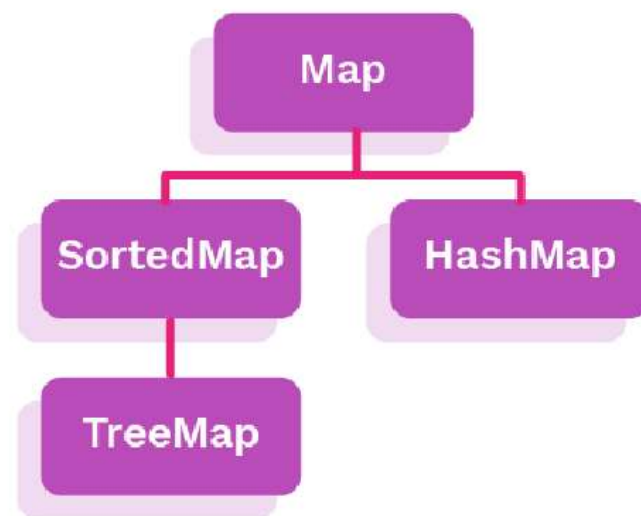
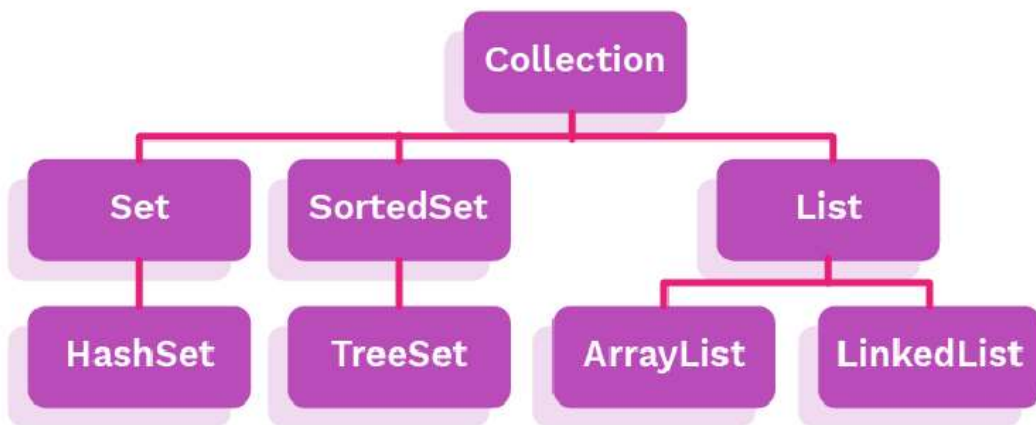


Misión  
TIC 2022





## Jerarquía de herencia de los mapas y las colecciones





El futuro digital  
es de todos

MinTIC

# ArrayList

Hechos

QUE

CONECTAN



## Declaración y creación

```
List<String> list = new ArrayList<String>();  
List<Integer> list = new ArrayList<>(Arrays.asList(1, 2, 3, 4));
```

## Agregar elementos

```
List<String> list = new ArrayList<String>();  
list.add("Pedro");  
list.add("Juan");  
list.add("Camila");
```

Universidad  
Industrial de  
Santander



Misión  
TIC 2022



El futuro digital  
es de todos

MinTIC

# ArrayList

Hechos

QUE

CONECTAN



Obtener un elemento y  
tamaño

```
ejemploLista.get(0);  
ejemploLista.size();
```

Eliminar elementos

```
ejemploLista.remove(0);  
ejemploLista.remove("Juan");
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022





El futuro digital  
es de todos

MinTIC

# ArrayList

Hechos

QUE

CONECTAN



Imprimir una lista

```
System.out.println(ejemploLista);  
  
for (int i = 0; i < ejemploLista.size() ; i++) {  
    System.out.println(ejemploLista.get(i));  
}
```

```
Iterator i = ejemploLista.iterator();  
while(i.hasNext()) {  
    System.out.println(i.next())  
}
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# ArrayList

Hechos

QUE

CONECTAN



Eliminar todos los  
elementos

```
ejemploLista.clear();
```

Sublista

```
ejemploLista.subList(0, 2)
```

Está vacía o contiene

```
ejemploLista.isEmpty();  
ejemploLista.contains("José");
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Conjuntos

Hechos  
QUE CONECTAN ✓

Un conjunto es una clase que almacena elementos sin repetir (únicos) y no respeta el orden de inserción

```
Set<String> conjunto = new HashSet<>();  
conjunto.add("Conjuntos");
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022





El futuro digital  
es de todos

MinTIC

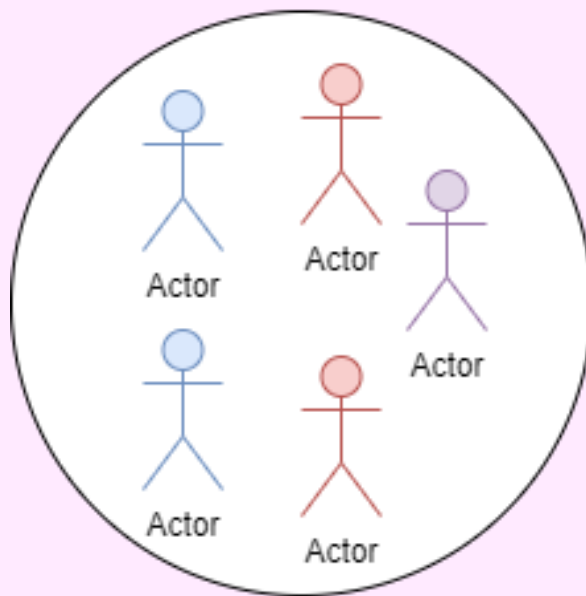
# Conjuntos

Hechos

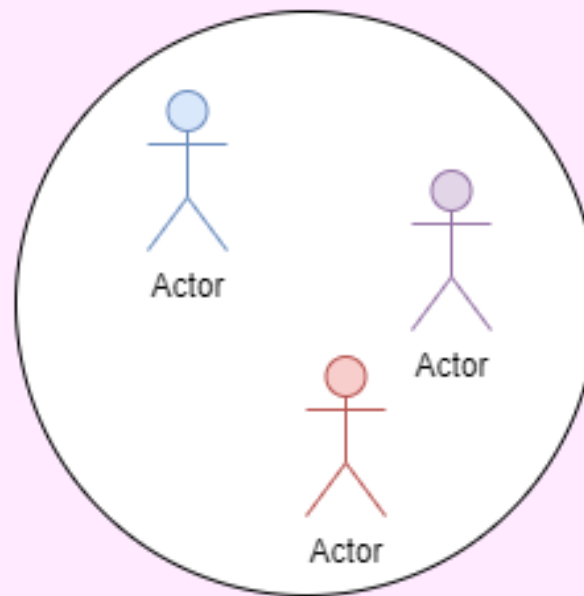
QUE

CONECTAN ✓

Grupo de elementos



Conjunto



Universidad  
Industrial de  
Santander



Misión  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Conjuntos

Hechos  
QUE CONECTAN ✓

Declarar e instanciar un conjunto

```
Set<String> frutasConjunto = new HashSet<>();  
Set<String> frutasConjunto2 = new HashSet<>(Arrays.asList(frutas));
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Conjuntos

Hechos

QUE

CONECTAN



Añadir un elemento

```
frutasConjunto.add("Banano");
```

Eliminar un elemento

```
frutasConjunto.remove("Banano");
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022





El futuro digital  
es de todos

MinTIC

# Conjuntos

Hechos  
QUE CONECTAN ✓

Obtener elementos

```
for(String fruta: frutasConjunto2){  
    System.err.println(fruta);  
}
```

Obtener un elemento

```
String[] tmpFrutas = frutasConjunto.toArray(new String[0]);  
System.out.println(tmpFrutas[0]);
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Mapas

## Hechos

QUE

CONECTAN



Clase que almacena elementos mediante claves/valor (Conocido como diccionarios en Python)

```
Map<String, Curso> mapas = new HashMap<>();  
mapas.put("Español", new Curso());  
mapas.get("Español");
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Mapas

## Hechos

QUE

CONECTAN



Clave	Valor
santander	objeto: Departamento
queso	34
null	null

Universidad  
Industrial de  
Santander



Mision  
TIC 2022





El futuro digital  
es de todos

MinTIC

# Mapas

Hechos

QUE

CONECTAN



Crear e instanciar mapa

```
Map<String,String> mapaFrutas = new HashMap<>();
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Mapas

**Hechos**  
QUE CONECTAN ✓

Insertar clave/valor

```
mapaFrutas.put("Fresa", "Malteada de fresa");
```

Obtener valor mediante clave

```
mapaFrutas.get("Fresa");  
mapaFrutas.getDefault("Fresa", "Valor por defecto");
```

Universidad  
Industrial de  
Santander



Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC

# Patrón contenedor

Hechos

QUE

CONECTAN



El patrón contenedor hacer referencia a una clase diseñada específicamente para almacenar otras clases.

## Por ejemplo:

- Productos en un carrito de compras
- Ítems en una lista
- Estudiantes en un colegio
- Etc.



Universidad  
Industrial de  
Santander



Misión  
TIC 2022

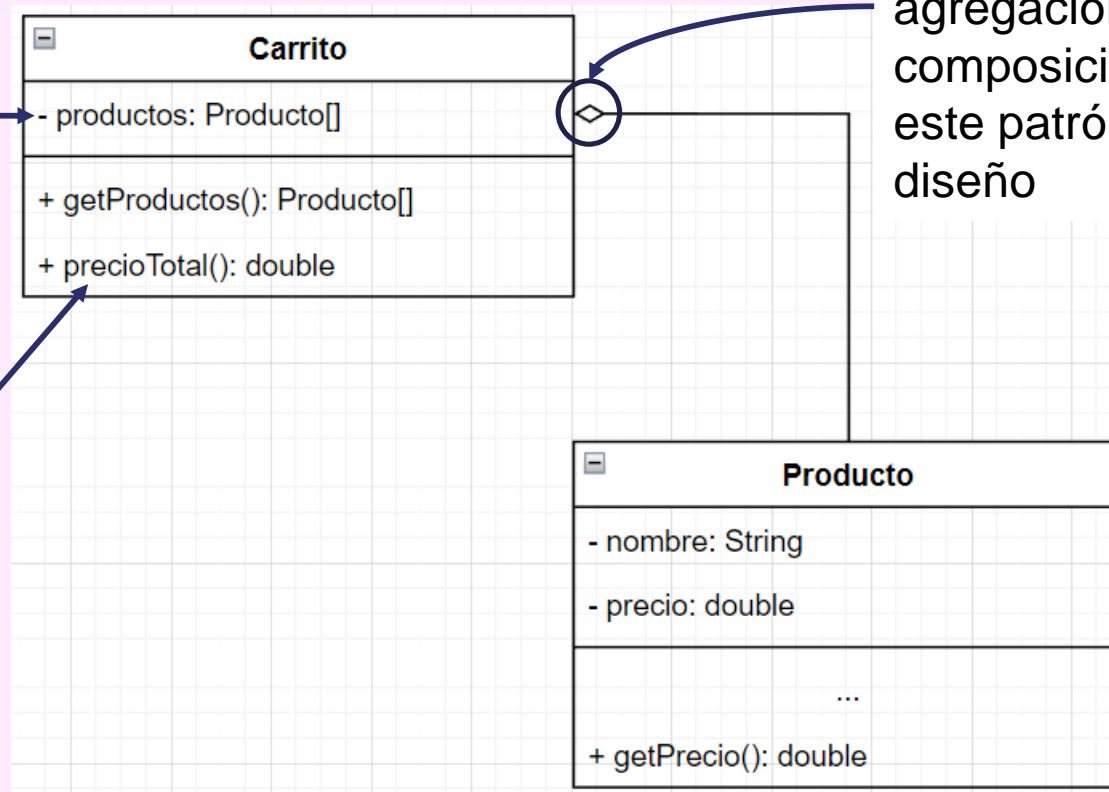




## Patrón contenedor

Nótese que se usa un vector  
o lista para almacenar  
instancias de la clase  
contenida

Las clases contenedoras suelen  
tener métodos que interactúan con  
los objetos de las clases contenidas.  
Por ejemplo, calcular el total del  
precio del carrito de compras a partir  
de los precios de los productos allí  
añadidos.



Se usa la  
agregación o la  
composición para  
este patrón de  
diseño



El futuro digital  
es de todos

MinTIC

Hechos

QUE

CONECTAN ✓

# Ejercicios prácticos

Universidad  
Industrial de  
Santander

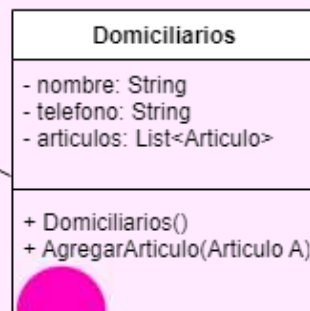
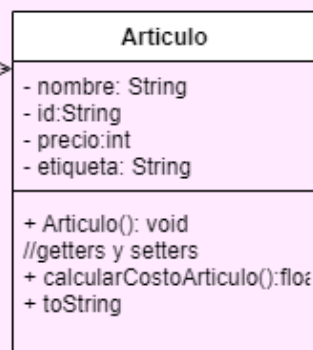
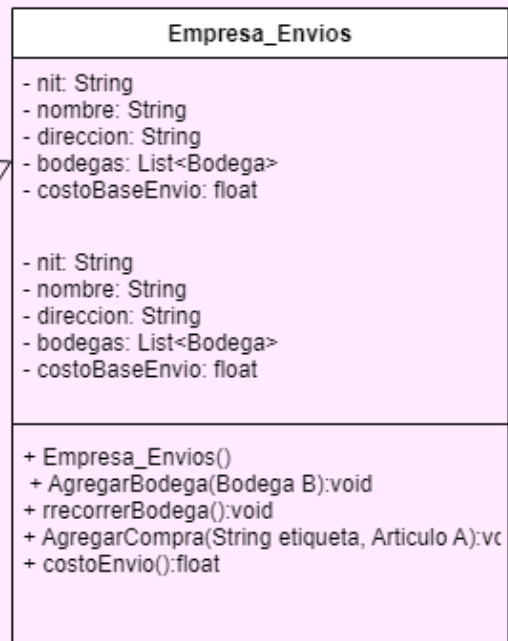
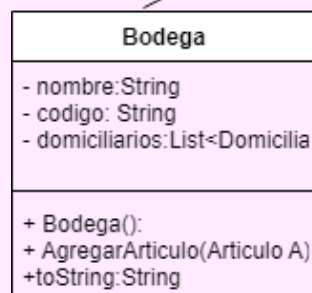
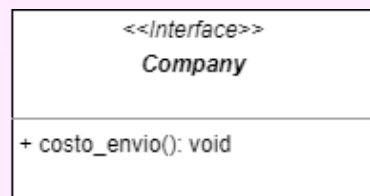


Mision  
TIC 2022



El futuro digital  
es de todos

MinTIC



# Hechos

QUE

CONECTAN ✓

Universidad  
Industrial de  
Santander



Misión  
TIC 2022