



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela

## **Generics en Java**

1. Restricciones
2. Clase genérica

# Generics en Java

# Generics en Java

---

1. Funcionalidad de java que provee **compile-time type checking**
2. Permite que por ejemplo, Integer, String, objetos propios puedan ser utilizados como parámetros de métodos, clases e interfaces.
3. Es decir, Generics permite crear métodos, clases e interfaces de tipo genérico.

- `LinkedList<String> list = new LinkedList<String>();`
- `LinkedList<Integer> list = new LinkedList<Integer>();`
- `LinkedList<Auto> list = new LinkedList<Auto>();`
- `LinkedList<E> list = new LinkedList<E>();`

# ¿Se ejecutará este programa?

---

```
import java.util.ArrayList;

public class Ejemplo1 {

    public static void main(String[] args) {
        int x = 5;
        ArrayList<Integer> arrayList = new ArrayList<Integer>();
        arrayList.add(x);
        arrayList.add(4);
        System.out.println(arrayList);
    }
}
```

# Restricciones en Generics

# Problemas con tipos

---

## Tipo Primitivo

- `int`
- `double`
- `boolean`
- `Char`

## Wrapper Class

- `Integer`
- `Double`
- `Boolean`
- `Character`

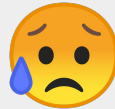
# Restricciones

```
import java.util.ArrayList;

public class Ejemplo2 {

    public static void main(String[] args) {
        // no se puede generalizar tipos primitivos
        ArrayList<int> arrayList = new ArrayList<int>>();
        ArrayList<char> arrayList = new ArrayList<int>>();

        // sólo se puede generalizar objetos
        ArrayList<Integer> arrayList = new ArrayList<Integer>>();
        ArrayList<Character> arrayList = new ArrayList<Character>();
    }
}
```





# Clase genérica

# Ejemplo clase Genérica

---

```
public class Elemento <T> {  
  
    public T dato;  
  
    public Elemento(){ }  
  
    public void setDato(T t){  
        this.dato = t;  
    }  
  
    public T getDato(){  
        return this.dato;  
    }  
}
```

```
public class LanzarElemento {  
    public static void main(String[] args) {  
        Elemento<Double> elEntero = new Elemento<Double>();  
        elEntero.setDato(23.3);  
        System.out.println(elEntero.getDato());  
  
        Elemento<String> elString = new Elemento<String>();  
        elString.setDato("hola");  
        System.out.println(elString.getDato());  
    }  
}
```





# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela