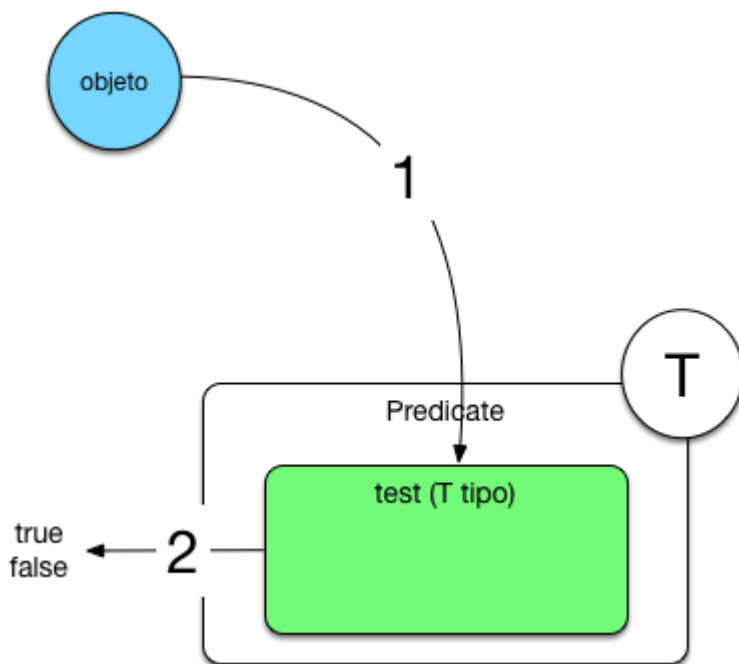


## CURSO JAVA 8 GRATIS APUNTATE!!

Crear un Java 8 Predicate , es una de las operaciones que más realizaremos cuando trabajemos con expresiones Lambda y Streams en Java 8. ¿Qué es un Predicado? , un Predicado es un interface funcional que define una condición que un objeto determinado debe cumplir . ¿Por ejemplo es una Persona mayor de 20 años?.



El Predicado dispone de un único método denominado test que recibe el objeto y comprueba si cumple la condición. Vamos a construir en ejemplo apoyándonos en la clase Persona.

```
package com.arquitecturajava;
```

```
public class Persona {

    private String nombre;
    private String apellidos;
    private int edad;
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public Persona(String nombre, String apellidos, int edad) {
        super();
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.edad = edad;
    }
    public String getApellidos() {
        return apellidos;
    }
    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }
    public int getEdad() {
        return edad;
    }
    public void setEdad(int edad) {
        this.edad = edad;
    }

}
```

Vamos a construir una lista de Personas para más adelante aplicar un Predicado sobre ella.

```
public class Principal {  
  
    public static void main(String[] args) {  
  
        List<Persona> lista = new  
        ArrayList<>();  
        Persona p1 = new Persona("pepe", "perez", 20);  
        Persona p2 = new Persona("angel", "sanchez", 30);  
        Persona p3 = new Persona("pepe", "blanco", 40);  
        lista.add(p1);  
        lista.add(p2);  
        lista.add(p3);  
  
    }  
  
}
```

**TODOS LOS CURSOS  
PROFESIONALES  
25\$/MES  
APUNTATE!!**

Finalmente convertiremos la lista en un Stream de datos y la recorreremos:



```
@Override
public boolean test(Persona p) {

    return p.getNombre().equals("pepe");
}

};
```

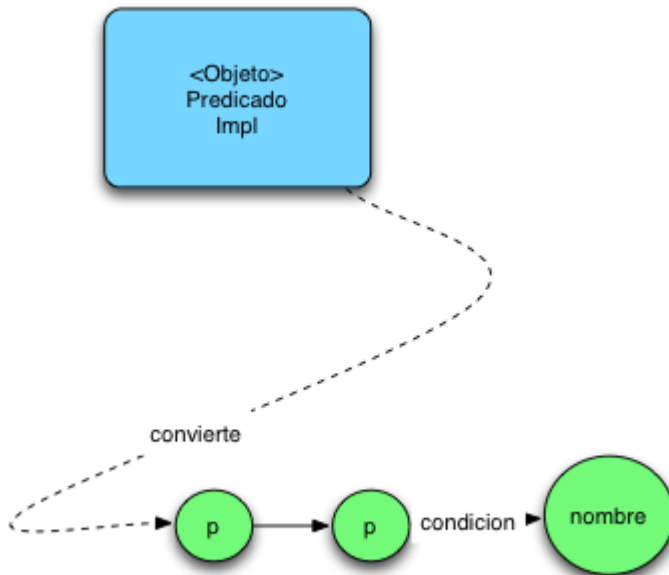
En este caso acabamos de crear un Predicado que solo cumplirán las personas que se llamen “pepe”. Usamos la función filter a nivel de Streams para pasar este objeto Predicado y obligar a que se filtre en base a la condición que el Predicado define.

```
lista.stream().filter(predicadoNombre).forEach(p -& & & >
System.out.println(p.getApellidos()));
```

El resultado en pantalla será:

```
*****
perez
blanco
*****
```

El Stream se ha apoyado en el Predicado para realizar el filtrado y quedarnos con las Personas que se llaman “pepe”. Recordemos que los interfaces funcionales se pueden ver como expresiones Lambda y su gestión es más sencilla.



Es momento de convertir nuestro Java 8 Predicate a una expresión lambda para ganar claridad:

```
lista.stream()  
    .filter(p -> p.getNombre().equals("pepe"))  
    .forEach(p -> System.out.println(p.getApellidos()));
```

El resultado será idéntico:

```
*****
perez
blanco
*****
```

# CURSO MAVEN GRATIS

## APUNTATE!!

[Predicados de Java 8](#)

Otros artículos relacionados:

[Java Lambda](#) [Java Streams](#)