



**EDUCACIÓN
EN LÍNEA**



Computación Paralela y Distribuida

Expresiones de Lambda

Nombre: Anderson Lucero, Diego Vaca, Cristian Quilumbaquin

Expresiones de Lambda

- La expresión Lambda es una nueva característica que se introduce en Java 8. Una expresión lambda es una función anónima. Una función que no tiene un nombre y no pertenece a ninguna clase. El concepto de expresión lambda se introdujo por primera vez en el lenguaje de programación LISP.

Sintaxis de expresión de Lambda de Java

- Para crear una expresión lambda, especificamos los parámetros de entrada (si los hay) en el lado izquierdo del operador lambda \rightarrow y colocamos la expresión o el bloque de instrucciones en el lado derecho del operador lambda. Por ejemplo, la expresión lambda $(x, y) \rightarrow x + y$ especifica que la expresión lambda toma dos argumentos x e y y devuelve la suma de estos.

```
//Syntax of lambda expression  
(parameter_list) -> {function_body}
```

- El operador lambda (\rightarrow) separa la declaración de parámetros de la declaración del cuerpo de la función.
- Parámetros:
 1. Cuando se tiene un solo parámetro no es necesario utilizar los paréntesis.
 2. Cuando no se tienen parámetros, o cuando se tienen dos o más, es necesario utilizar paréntesis.
- Cuerpo de lambda:
 1. Cuando el cuerpo de la expresión lambda tiene una única línea no es necesario utilizar las llaves y no necesitan especificar la cláusula return en el caso de que deban devolver valores.
 2. Cuando el cuerpo de la expresión lambda tiene más de una línea se hace necesario utilizar las llaves y es necesario incluir la cláusula return en el caso de que la función deba devolver un valor .

Expresión Lambda vs método en Java

- ❖ Un método (o función) en Java tiene estas partes principales:
 - Nombre.
 - Lista de parámetros.
 - Cuerpo.
 - Tipo de devolución.
- ❖ Una expresión lambda en Java tiene estas partes principales: la expresión Lambda solo tiene body y parameter list.
 - No hay nombre – la función es anónima por lo que no nos importa el nombre.
 - Lista de parámetros.
 - Cuerpo – Esta es la parte principal de la función.
 - Sin tipo de valor devuelto: el compilador java 8 puede inferir el tipo de valor devuelto comprobando el código. no es necesario mencionarlo explícitamente.

Dónde utilizar los Lambda en Java

- Para utilizar la expresión lambda, debe crear su propia interfaz funcional o utilizar la interfaz funcional predefinida proporcionada por Java. Una interfaz con **un solo método abstracto** se denomina interfaz funcional (o interfaz de método Single Abstract), por ejemplo: Runnable, callable, ActionListener, etc.
- **Para utilizar la interfaz de función:**
Pre Java 8: Creamos clases internas anónimas.
Post Java 8: Puede utilizar la expresión lambda en lugar de las clases internas anónimas.

Tipos

Las expresiones lambda puede clasificarse de la siguiente manera:

- Consumidores.
- Proveedores.
- Funciones.
 - Operadores Unarios.
 - Operadores Binarios.
- Predicados.
- A continuación iremos detallando uno a uno y contando en que consisten.

Tipos

Consumidores

- Se trata de aquellas expresiones lambda que aceptan un solo valor y no devuelven valor alguno.

```
1 String message -> System.out.println(message);
```

Proveedores

- En este caso se trata de expresiones que no tienen parámetros pero devuelven un resultado.

```
1 () -> return createRandomInteger()
```


Tipos

Funciones

- Aquellas expresiones que aceptan un argumento y devuelven un valor como resultado y cuyos tipos no tienen porque ser iguales.

```
1 | Order persistedOrder -> persistedOrder.getIdentifier();
```

Operadores Unarios

- Caso especial de funciones en las que tanto el parámetro como el valor devuelto son del mismo tipo.

```
1 | String message -> message.toLowerCase()
```

Tipos

Operadores Binarios

- Igual que en el caso de los Operadores Unarios, se trata de un caso especial de funciones en las que los dos argumentos y el resultado son del mismo tipo.

```
1 (String message, String anotherMesssage) -> message.concat(anotherMessage);
```

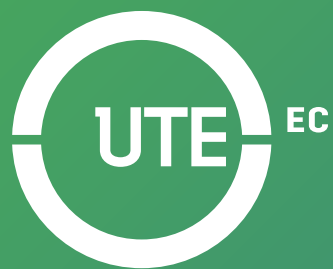
Predicados

- Se trata de expresiones que aceptan un parámetro y devuelven un valor lógico.

```
1 String message -> message.length > 50
```

Expresión Lambda vs método en Java

- Bibliografía:
- Singh, C. (2017, 2 octubre). *Java Lambda Expressions Tutorial with examples*. beginnersbook.com. <https://beginnersbook.com/2017/10/java-lambda-expressions-tutorial-with-examples/>.
- Villapecellín, J. L. R. (2015, 4 diciembre). *Expresiones Lambda con Java 8*. Adictos al trabajo. <https://www.adictosaltrabajo.com/2015/12/04/expresiones-lambda-con-java-8/>



¡GRACIAS!

**TRAS
CENDE
MOS**

A white curved line graphic, resembling a stylized 'C' or a partial arc, positioned to the right of the text.