



Instituto Tecnológico Superior de Jerez

Ingeniería en Sistemas Computacionales

Programación Lógica y Funcional

10 semestre

Tema 2.- Modelo de Programación Funcional

Actividad 5.- Mapa Conceptual

Alumno: Esteban Faustino Muñoz Hidalgo

E-mail: faustino10.96@gmail.com

No. De Control: S15070115

MTI. Salvador Acevedo Sandoval

Jerez, Zacatecas a 20 de marzo de 2020.

Investigar las respuestas a las preguntas que se listan a continuación y posteriormente crear un MAPA CONCEPTUAL en alguna herramienta electrónica.

1.- Matemáticamente, que es el cálculo lambda:

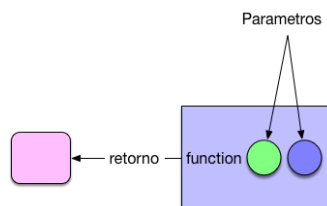
El lambda calculo (λ C) es un lenguaje simple que permite la descripción de las funciones matemáticas y de sus propiedades. (Blas Carlos Ruiz Jimenez, 2002)

2.- ¿Que son las ‘functional interfaces’ en Java?

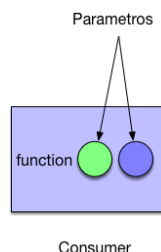
Las “functional interfaces” son aquellas interfaces que solo disponen de un método abstracto. (Caules, 2016)

3.- ¿Cuáles son las 6 interfaces funcionales del paquete java.util.function?

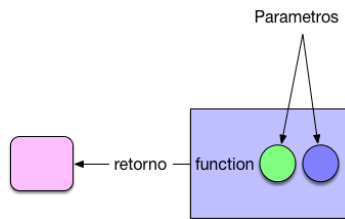
* **Java 8 Functional Interfaces:** Una función es un bloque de código que recibe varios parámetros y devuelve un resultado.



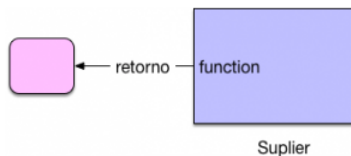
* **Java 8 Functional Interfaces y Consumers:** Ahora bien, existen variaciones sobre el concepto general de función. Por ejemplo, podemos disponer de una función que reciba parámetros pero que no devuelva nada.



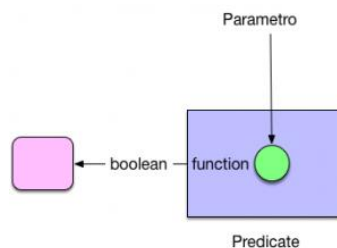
* **Java 8 Functions:** El siguiente tipo de interface son los que habitualmente se llaman Functions reciben uno o más parámetros y retornan un resultado:



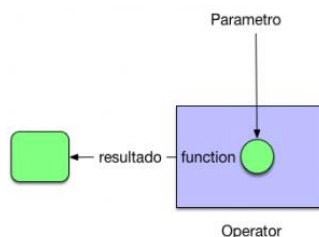
* **Java 8 Suppliers:** Este caso es algo más complejo ya que se trata de una función que no recibe parámetro alguno y devuelve un resultado. Nos puede parecer algo absurdo de entrada ya que una función debe recibir algo. Sin embargo, hay a veces que necesitamos generar un nuevo contenido prácticamente desde la nada.



* **Java 8 Predicates:** Los interfaces de tipo predicado son una especialización de los Functions ya que reciben un parámetro y devuelven un valor booleano.



* **Java 8 Operators:** El uso de Java 8 Operators es otro caso específico de los Function interfaces . Se trata de interfaces funcionales que reciben un tipo de parámetro y devuelven un resultado que es del mismo tipo. (Caules, arquitecturajava, 2018)



4.- ¿Que son las expresiones lambda?

Son subrutinas definidas que no están enlazadas a un identificador. Las expresiones lambda a menudo son argumentos que se pasan a funciones de orden superior, o se usan para construir el resultado de una función de orden superior que necesita devolver una función. (Wikipedia, 2020)

5.- Sintaxis de las expresiones lambda:

En el cálculo lambda, una *expresión* o *término* se define recursivamente a través de las siguientes reglas de formación:

- Toda variable es un término: $x, y, z, u, v, w, x_1, x_2, x_3, \dots$
- Si t es un término y x es una variable, entonces $(\lambda x.t)$ es un término (llamado una *abstracción lambda*).
- Si t y s son términos, entonces (ts) es un término (llamado una *aplicación lambda*).
- Nada más es un término. (Wikipedia, wikipedia, 2020)

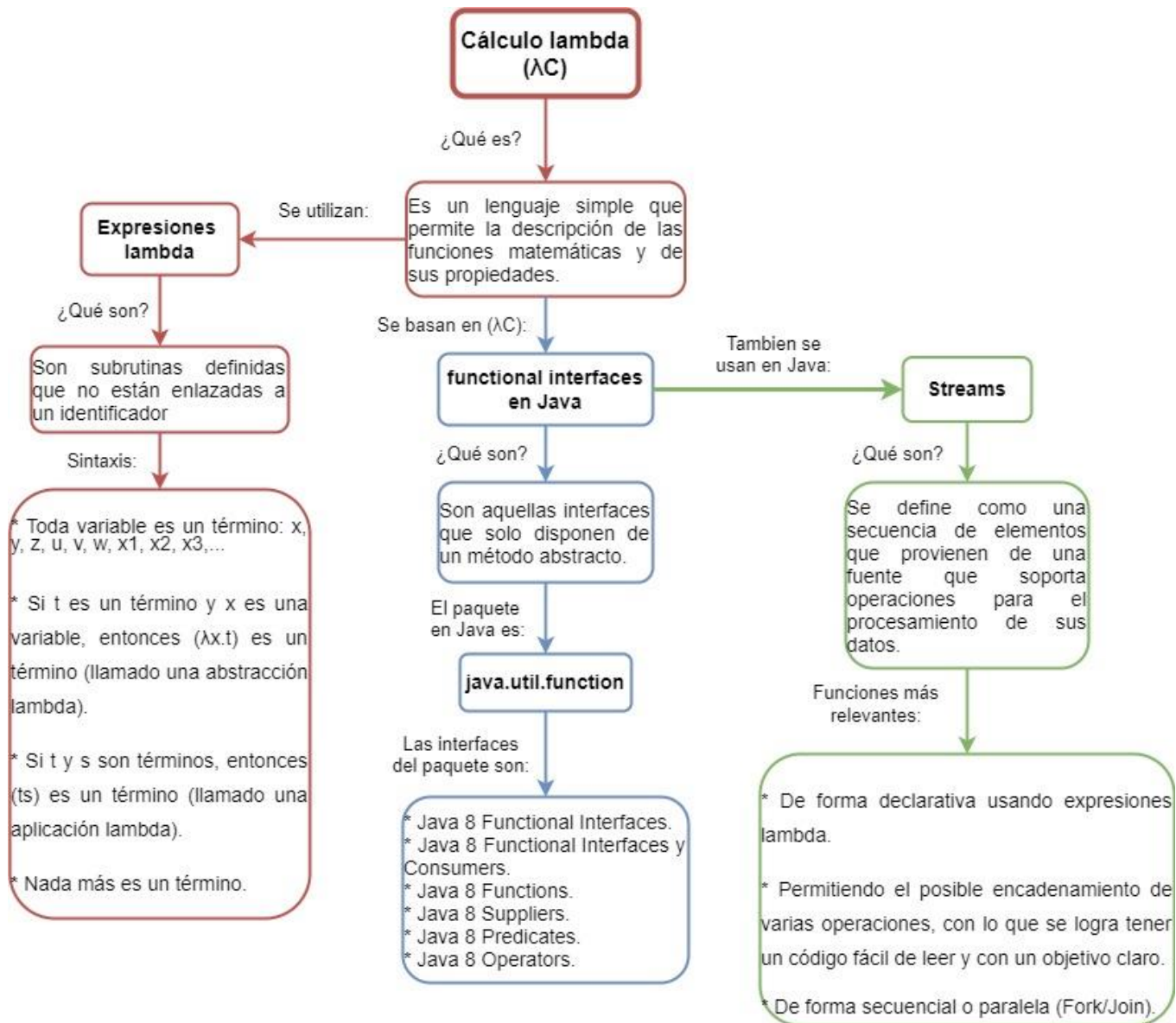
6.- ¿Que son los streams y para qué sirven?

Stream se define como una secuencia de elementos que provienen de una fuente que soporta operaciones para el procesamiento de sus datos.

Permite realizar operaciones sobre colecciones de datos usando el modelo filtro/mapeo/reducción, en el cual se seleccionan los datos que se van a procesar (filtro), se convierten a otro tipo de dato (mapeo) y al final se obtiene el resultado deseado (reducción).

7.- Funciones más relevantes de la clase Stream:

- De forma declarativa usando expresiones lambda.
- Permitiendo el posible encadenamiento de varias operaciones, con lo que se logra tener un código fácil de leer y con un objetivo claro.
- De forma secuencial o paralela (Fork/Join). (Lopez, 2015)



Bibliografía

Blas Carlos Ruiz Jimenez, B. C. (06 de marzo de 2002). *uma*. Obtenido de uma:

<http://www.lcc.uma.es/~blas/apuntes/PDAv/lambdaC.pdf>

Caules, C. Á. (26 de febrero de 2016). *arquitecturajava*. Obtenido de arquitecturajava:

<https://www.arquitecturajava.com/java-functional-interface/>

Caules, C. Á. (18 de abril de 2018). *arquitecturajava*. Obtenido de arquitecturajava:

<https://www.arquitecturajava.com/java-8-functional-interfaces-y-sus-tipos/>

Lopez, A. (octubre de 2015). *oracle*. Obtenido de oracle:

<https://www.oracle.com/technetwork/es/articles/java/expresiones-lambda-api-stream-java-2737544-esa.html>

Wikipedia. (18 de marzo de 2020). *wikipedia*. Obtenido de wikipedia:

https://es.wikipedia.org/wiki/Expresi3n_lambda

Wikipedia. (18 de marzo de 2020). *wikipedia*. Obtenido de wikipedia:

https://es.wikipedia.org/wiki/C3lculo_lambda#Sintaxis