



INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

Ingeniería en Sistemas Computacionales

8° Semestre

Alumno: José Félix Alvarado Barrios

NC:16070124

Materia: Prog. Lógica y Funcional

Nombre del trabajo: Mapa conceptual

Docente: ISC Salvador Acevedo Sandoval

Jerez de García Salinas a 6 de Marzo 2020

1. ¿Qué es la EVALUACIÓN PEREZOSA?

En la teoría de lenguajes de programación, la evaluación perezosa (del inglés lazy evaluation) o llamada por necesidad es una estrategia de evaluación que retrasa el cálculo de una expresión hasta que su valor sea necesario, y que también evita repetir la evaluación en caso de ser necesaria en posteriores ocasiones. Esta compartición del cálculo puede reducir el tiempo de ejecución de ciertas funciones de forma exponencial, comparado con otros tipos de evaluación.

2. ¿Qué ventajas tiene la implementación de la evaluación perezosa?

Los beneficios de la evaluación perezosa son:

- El incremento en el rendimiento al evitar cálculos innecesarios, y en tratar condiciones de error al evaluar expresiones compuestas.
- La capacidad de construir estructuras de datos potencialmente infinitas.
- La capacidad de definir estructuras de control como abstracciones, en lugar de operaciones primitivas.

3. ¿Qué desventajas tiene la implementación de la evaluación perezosa?

La evaluación perezosa por otra parte es más inexacta y si se tienen errores en los resultados que da podría ser más confuso encontrar donde está el error ya que el programa simplemente deja de hacer lo que hace en vez de mostrar algún dialogo de error.

5. ¿Qué son las funciones de alto orden?

Una función es de orden superior si toma una función como argumento o devuelve una función como resultado.

Una función de orden superior, a diferencia de una función de primer orden, puede tener una de tres formas:

- Uno o más de sus parámetros es una función y devuelve algún valor.
- Devuelve una función, pero ninguno de sus parámetros es una función.

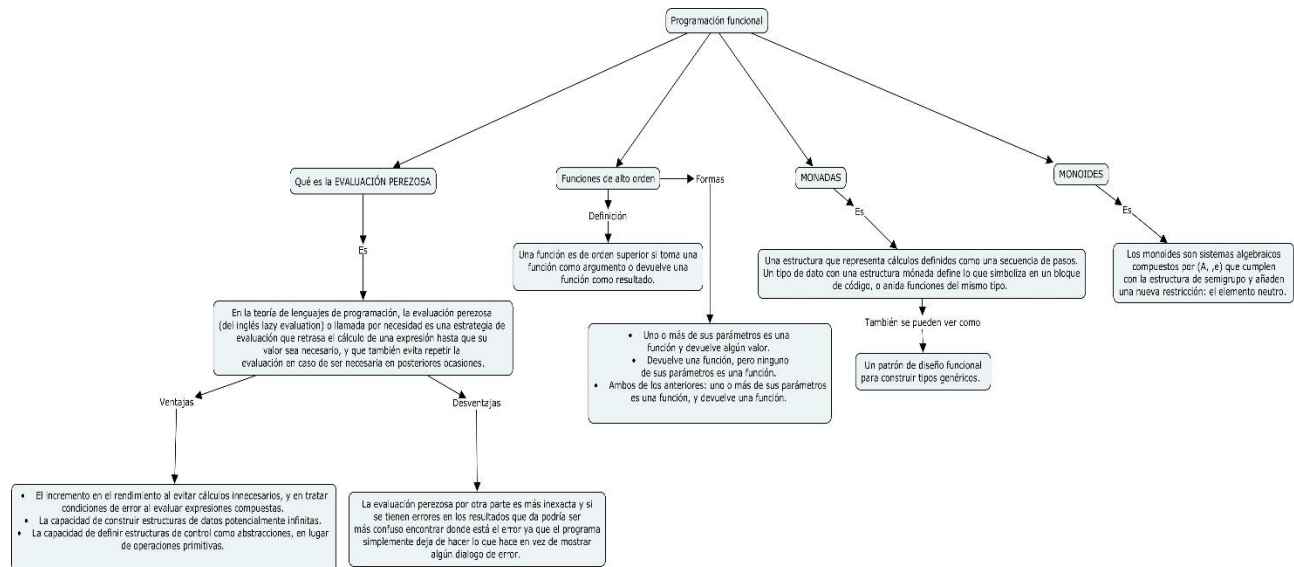
- Ambos de los anteriores: uno o más de sus parámetros es una función, y devuelve una función.

6. ¿Qué son las MONADAS en programación funcional?

Una mónada (monad en inglés), es una estructura que representa cálculos definidos como una secuencia de pasos. Un tipo de dato con una estructura mónada define lo que simboliza en un bloque de código, o anida funciones del mismo tipo. Esto le permite al programador crear tuberías informáticas que procesan datos en pasos, a los cuales se les asocia un decorador con reglas de proceso adicionales provistas por la mónada. Por lo tanto, las mónadas han sido descritas como un “punto y coma programable”; un punto y coma, siendo el operador usado para unir varias declaraciones en la programación imperativa, en consecuencia, la expresión implica que código extra será ejecutado entre las declaraciones de una tubería. Las mónadas también han sido explicadas con una metáfora física, donde se comportan como una línea de ensamblaje, en las que un objeto transporta datos entre unidades funcionales que la van transformando un paso a la vez. También se las puede ver como un patrón de diseño funcional para construir tipos genéricos.

7. ¿Qué son los MONOIDES en programación funcional?

Los monoides son sistemas algebraicos compuestos por (A, \odot, e) que cumplen con la estructura de Semigrupo y añaden una nueva restricción: el elemento neutro.



Referencias

Monadas. (s.f.). En Wikipedia. Recuperado el 3 de marzo de 2020 de: [https://es.wikipedia.org/wiki/M%C3%B3nada_\(programaci%C3%B3n_funcional\)](https://es.wikipedia.org/wiki/M%C3%B3nada_(programaci%C3%B3n_funcional))

Lazy evaluation. (s.f.). En Wikipedia. Recuperado el 3 de marzo de 2020 de: https://en.wikipedia.org/wiki/Lazy_evaluation

Programación Logica y funcional . (2016). En Blogspot. Recuperado el 3 de marzo de 2020 de: <http://plyfoact.blogspot.com/2016/06/tiposde-evaluacion.html>

Funciones de alto orden (función como parámetro) (s.f.). En RipTutorial. Recuperado el 3 de marzo de 2020 de: <https://riptutorial.com/es/scala/example/26649/funciones-de-alto-orden--funcion-como-parametro->

Álgebras y funciones: patrones en programación funcional (2018). En Adictos al Trabajo. Recuperado el 3 de marzo de 2020 de: <https://www.adictosaltrabajo.com/2018/08/14/algebras-y-funciones-patrones-en-programacion-funcional/#022>