



## Mapa Conceptual

**Nombre del alumno:** Yahir Aguilar Sicalhua.

**Nombre del tema:** Unidad III. Evaluación Perezosa.

**Parcial:** 1.

**Nombre de la materia:** Programación Lógica.

**Nombre del profesor:** Andrés Alejandro Reyes Molina.

**Nombre de la licenciatura:** Ingeniería en Sistemas Computacionales.

**Cuatrimestre:** 8.

# EVALUACIÓN PEREZOSA

Evaluación mediante paso de parámetros por nombre usando punteros para compartir valores de expresiones.

## 3.7 Implementación

En muchos otros idiomas, la evaluación puede ser retrasado mediante la suspensión de forma explícita el cálculo utilizando la sintaxis especial (como ocurre con el esquema de "delay" y "force" y OCaml 's 'lazy' y "Lazy.force") o, más generalmente, envolviendo la expresión en un golpe seco.

## 3.8 Pereza y afán Controlar el entusiasmo en lenguajes perezosos

En los lenguajes de programación perezosos como Haskell, aunque el valor predeterminado es evaluar las expresiones sólo cuando se exigen, en algunos casos es posible hacer que el código sea más ansioso o, por el contrario, hacerlo más perezoso nuevamente después de que se haya hecho más ansioso.

## 3.9 Simular la pereza en idiomas ávidos

Java.  
En Java, la evaluación diferida se puede realizar mediante el uso de objetos que tienen un método para evaluarlos cuando se necesita el valor. El cuerpo de este método debe contener el código necesario para realizar esta evaluación. Desde la introducción de expresiones lambda en Java SE8, Java ha admitido una notación compacta para esto.

## 3.10 La estrategia de evaluación perezosa.

La Evaluación impaciente evalúa cada expresión en el momento exacto del tiempo en que ésta es encontrada dentro del código fuente. La evaluación perezosa, por el contrario, posterga la evaluación de la expresión hasta que su valor es realmente demandado por el programa en ejecución.

## 3.11 Técnicas de programación funcional perezosa.

La evaluación perezosa puede también reducir el consumo de memoria de una aplicación, ya que los valores se crean solo cuando se necesitan. Sin embargo, es difícil de combinar con las operaciones típicas de programación imperativa, como el manejo de excepciones o las operaciones de entrada/salida, porque el orden de las operaciones puede quedar indeterminado.