Módulo Profesional: Big Data Aplicado

Scala

Llamada por Nombre o Llamada por Valor

```
def LlamadaporValor (x: Long): Unit = {
    println ("Por valor" + x)
    println ("Por valor" + x)
}

def LlamadaporNombre (x: => Long): Unit = {
    println ("Por nombre" + x)
    println ("Por nombre" + x)
}

LlamadaporValor(System.nanoTime())
LlamadaporNombre(System.nanoTime())
```

Llamada por Valor

```
def LlamadaporValor (x: Long): Unit = {
    println ("Por valor" + x)
    println ("Por valor" + x)
}
```

```
LlamadaporValor(System.nanoTime())
```

Devuelve el tiempo actual en nanosegundos. El resultado se calcula una vez antes de que la función comience a ejecutarse.

La función LlamadaporValor toma un argumento x que se evalúa inmediatamente cuando se llama a la función. Significa que el valor de x se calcula una vez, antes de que se ejecute el cuerpo de la función y el mismo valor se utiliza en todas las referencias a x dentro de la función.

Llamada por Nombre

```
def LlamadaporNombre (x: => Long): Unit = {
    println ("Por nombre" + x)
    println ("Por nombre" + x)
}
```

```
LlamadaporNombre(System.nanoTime())
```

System.nanoTime() no se evalua inmediatamente cuando se realiza la llamada a la función. Si no que se evalúa cuando es referenciado en el cuerpo de la función.

La función como parámetro x: => Long, significa que el argumento se pasa como una expresión sin evaluar y cada vez que x es utilizado en la función la expresión asociada se evalúa nuevamente.

Llamada por Valor / Llamada por Nombre

```
/*Ejercicio 1: Explica la diferencia de resultado de imprimePrimer*/

def Infinito(): Int = 1 + Infinito()
def imprimePrimer(x: Int, y : => Int) = println (x)

imprimePrimer(Infinito(), 34)
imprimePrimer(34, Infinito())
```

Parámetros por defecto o por nombre

Pasas todos los parámetros de la función o nombras los parámetros

```
def guardaPicture(formato: String = "jpg", ancho: Int = 1920, altura: Int = 1080): Unit =
    println("guarda picture")

guardaPicture(ancho = 800)
guardaPicture("bmp")
guardaPicture(800,600)
guardaPicture("bmp",600,800)
guardaPicture(altura = 600, ancho = 800, formato = "bmp")
```

Parámetros por defecto o por nombre

En este caso uno en la primera llamada pasamos sólo un parámetro y en la segunda llamada pasamos los dos parámetros.

```
def factorial (x: Int, acc: Int = 1): Int ={
....
}

val fact10 = factorial (10)
val fact11 = factorial (11,2)
```