



INSTITUTO TECNOLÓGICO DE TIJUANA
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
PERIODO: Agosto - Diciembre 2018

Carrera:

Ingeniería en Sistemas Computacionales

Materia:

Datos Masivos

Alumno:

Hernández Barrios Eric Harim

Fecha:

11 de Septiembre de 2018

Código utilizado

```
//Ejercicio 1

def isEven(number: Int) = number % 2 == 0

def EJ1(number: Int) : Boolean = {

    if(isEven(number)) {
        return true
    }
    false
}
EJ1(4)
EJ1(7)


// Ejercicio 2

def isEven(number: Int) = number % 2 == 0
def isOdd(number: Int) = !isEven(number)

def EJ2(test:List[Int]) : String = {
    val n = 0

    for (n <- tests) {
        print(n)
        if (isEven(n)) {
            println(" -> Par" )

        }
        if (isOdd(n)) {

            println(" -> Impar" )
        }
    }
    return "Listo"
}
```

```
}  
val LISTA = List.range(-2, 10)  
EJ2(LISTA)
```

```
//Ejercicio 3
```

```
val milist = List(1,3,3,4,6,7,3,7)  
def EJ3(lista:List[Int]) : Int = {  
  
    val x = lista.map(e => if(e==7) 14 else e)  
    val TT = x.sum  
    return TT  
}  
EJ3(milist)
```

```
//Ejercicio 4
```

```
def EJ4(Listap:List[Int], Seccion:Int): Boolean = {  
    var sup = 0  
    var inf = 0  
    var x = 0  
    while(x< Listap.size){  
        if(x <= Seccion-1){  
            sup = sup + Listap(x)  
            println(sup)  
        }  
        else{  
            inf= inf + Listap(x)  
            println(inf)  
        }  
        x = x+1  
    }  
    if(sup == inf){  
        return true  
    }else{  
        return false  
    }  
}  
var milista= List(1,3,3,4,6,7,3,7)  
var partir = milista.size/2
```

```
EJ4(milista,partir)
```

```
//Ejercicio 5
```

```
def EJ5 (Cadena:String) : Boolean = {
```

```
  val alreves=Cadena.reverse
```

```
  if (Cadena == alreves)
```

```
    {
```

```
      return true
```

```
  } else {
```

```
    return false
```

```
  }
```

```
}
```

```
val texto = "popu"
```

```
EJ5(texto)
```

Explicación

```
//Ejercicio 1

def isEven(number: Int) = number % 2 == 0

def EJ1(number: Int) : Boolean = {

    if(isEven(number)) {
        return true
    }
    false
}

EJ1(4)
EJ1(7)
```

Hice un metodo llamado IsEven (si es par) en el cual mandamos un int, y si ese int es divisible entre 2 y su cociente es 0 entonces me hacemos un if diciendo, si el número es par entonces devolverá un true, de lo contrario devolverá un false Se probó con : un 4 y un 7

```
scala> :load Examen1.scala
Loading Examen1.scala...
isEven: (number: Int)Boolean
EJ1: (number: Int)Boolean
res247: Boolean = true
res248: Boolean = false
```

```
// Ejercicio 2
```

```
def isEven(number: Int) = number % 2 == 0
def isOdd(number: Int) = !isEven(number)

def EJ2(test: List[Int]) : String = {
  val n = 0

  for (n <- tests) {
    print(n)
    if (isEven(n)) {
      println(" -> Par" )

    }
    if (isOdd(n)) {

      println(" -> Impar" )
    }
  }
  return "Listo"
}
val LISTA = List.range(-2, 10)
EJ2(LISTA)
```

Hicimos el mismo procedimiento, solo que aquí hicimos otra función llamada isOdd (es impar) en el cual dice: si es un número diferente a isEven(es par) entonces se le asignará como impar a ese número. Ahora solo mandamos la lista al método, en el cual un ciclo for fuimos recorriendo la lista y a la vez evaluando cada número si es par e impar después regresamos toda la lista y al final un string que diga "listo" significado que se ha concluido con éxito.

```
isEven: (number: Int)Boolean
isOdd: (number: Int)Boolean
EJ2: (test: List[Int])String
LISTA: List[Int] = List(-2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
-2 -> Par
-1 -> Impar
0 -> Par
1 -> Impar
2 -> Par
3 -> Impar
4 -> Par
5 -> Impar
6 -> Par
7 -> Impar
8 -> Par
9 -> Impar
res249: String = Listo
```

```
//Ejercicio 3
val milist = List(1,3,3,4,6,7,3,7)
def EJ3(lista:List[Int]) : Int = {

val x = lista.map(e => if(e==7) 14 else e)
val TT = x.sum
return TT
}
EJ3(milist)
```

En este ejercicio creamos un método para que si en una lista hay uno o más elementos que sea 7 se reemplacen con el número 14 y que al final de evaluarlo, se sumen todos los elementos de la lista, devolviendo este valor. Para eso simplemente convertí la lista en un mapa e hice la condición que si hay un 7 en los elementos de la lista se reemplacen con 14 de lo contrario que sólo sea su valor original.

```
milist: List[Int] = List(1, 3, 3, 4, 6, 7, 3, 7)
EJ3: (lista: List[Int])Int
res250: Int = 48
```

//Ejercicio 4

```
def EJ4(Listap:List[Int], Seccion:Int): Boolean = {  
  var sup = 0  
  var inf = 0  
  var x = 0  
  while(x< Listap.size){  
    if(x <= Seccion-1){  
      sup = sup + Listap(x)  
      println(sup)  
    }  
    else{  
      inf= inf + Listap(x)  
      println(inf)  
    }  
    x = x+1  
  }  
  if(sup == inf){  
    return true  
  }else{  
    return false  
  }  
}  
var milista= List(1,3,3,4,6,7,3,7)  
var partir = milista.size/2  
  
EJ4(milista,partir)
```

En este ejercicio se ocupo ayuda de un compañero XD

Primero debemos crear la lista, una vez creado, se hace que la lista se parta en la mitad para tener ambas partes.

A continuación en el método declaramos la variable superior (sup) que es la primera sección recortada, y después la segunda sección recortada (inf), seguido de un contador. A continuación usamos un while que diga que mientras el contador este en dentro del número del tamaño de la lista vea si x pertenece en el tamaño menos 1 despues el superior se empieza a sumar con los numeros de esa parte e imprimiendo el resultado ya sumado, es igual con la parte inferior al final se evaluan la parte superior con la inferior y si son iguales devuelve un verdadero, si no es falso.


```

EJ4: (Listap: List[Int], Seccion: Int)Boolean
milista: List[Int] = List(1, 3, 3, 4, 6, 7, 3, 7)
partir: Int = 4
1
4
7
11
6 //Ejercicio 4
13
16 def EJ5 (Cadena:String) : Boolean = {
23
res251: Boolean = false
252: (Cadena: String)Boolean

```

```

//Ejercicio 5

def EJ5 (Cadena:String) : Boolean = {

val alreves=Cadena.reverse

if (Cadena == alreves)
{
    return true
} else {
    return false
}
}
val texto = "popu"
EJ5(texto)

```

En este ejercicio evaluamos una cadena, en la cual con el método usamos esa cadena y usamos el reverse, el resultado la guardamos en esa variable, después comprobamos si ambas cadenas son los mismos, si lo cumplen es verdadero y si no es falso.

```

res251: Boolean = false
EJ5: (Cadena: String)Boolean
texto: String = popu
res252: Boolean = false

```