



# Tecnológico Nacional de México Instituto Tecnológico de Tijuana

## Departamento de Sistemas y Computación Ingeniería en Sistemas Computacionales

Semestre:

Febrero - Junio 2022

Materia:

Datos Masivos

Docente:

JOSE CHRISTIAN ROMERO HERNANDEZ

Unidad: U1

Título del trabajo:

Práctica 4

Alumnos:

Manuel Angel Real Castro 18212253

Anahi Del Carmen Hernandez Pablo 18210486

```
//Given the pseudocode of the Fibonacci sequence in the provided link, implement  
//Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4, Algorithm 5 with Scala
```

## ALGORITHM 1

**Algoritmo 1** Versión recursiva descendente (Complejidad  $O(\varphi^n)$ )

**función** fib( $n$ )

**si**  $n < 2$  **entonces**

**devuelve**  $n$

**en otro caso**

**devuelve** fib( $n - 1$ ) + fib( $n - 2$ )

```
//Algorithm 1  
def fib1(n : Int) : Int = {  
    if(n < 2) {  
        return n  
    } else {  
        return fib1(n - 1) + fib1 (n - 2)  
    }  
}
```

```
scala> fib1(8)  
res2: Int = 21
```

## ALGORITHM 2

**Algoritmo 2** Versión con fórmula explícita (6) (Complejidad  $O(\log_2(n))$ )

**función** fib( $n$ )

**si**  $n < 2$  **entonces**

**devuelve**  $n$

**en otro caso**

$\varphi \leftarrow ((1 + \sqrt{5}) \div 2)$

$j \leftarrow ((\varphi^n - (1 - \varphi)^n) \div (\sqrt{5}))$

**devuelve**  $j$

```
//Algorithm 2
def fib2(n : Int) : Double = {
  if(n < 2) {
    return n
  } else {
    var p = ((1 + math.sqrt(5)) / 2)
    var j = ((math.pow(p, n) - (math.pow((1 - p), n))) / (math.sqrt(5)))
    return j
  }
}
```

```
scala> fib2(8)
res3: Double = 21.000000000000004
```

## ALGORITHM 3

**Algoritmo 3** Versión iterativa  
(Complejidad  $O(n)$ )

función fib( $n$ )

$a \leftarrow 0$

$b \leftarrow 1$

$c$

para  $k$  desde 0 hasta  $n$  hacer

$c \leftarrow b + a$

$a \leftarrow b$

$b \leftarrow c$

devuelve  $a$

```
//Algorithm 3
def fib3(n : Int) : Int = {
  var a = 0
  var b = 1
  var c = 0
  var k = 0
  while (k < n){
    c = b + a
    a = b
    b = c
    k = k + 1
  }
  return a
}
```

```
scala> fib3(8)
res4: Int = 21
```

## ALGORITHM 4

**Algoritmo 4** Versión iterativa  
2 variables (Complejidad  $O(n)$ )

```
función fib( $n$ )  
   $a \leftarrow 0$   
   $b \leftarrow 1$   
  para  $k$  desde 0 hasta  $n$  hacer  
     $b \leftarrow b + a$   
     $a \leftarrow b - a$   
  devuelve  $b$ 
```

```
//Algorithm 4  
def fib4( $n$  : Int) : Int = {  
  var  $a$  = 0  
  var  $b$  = 1  
  var  $k$  = 0  
  while ( $k < n$ ){  
     $b = b + a$   
     $a = b - a$   
     $k = k + 1$   
  }  
  return  $b$   
}
```

```
scala> fib4(8)  
res5: Int = 34
```

## ALGORITHM 5

**Algoritmo 5** Versión iterativa vector  
(Complejidad  $O(n)$ )

```
función fib( $n$ )  
  si  $n < 2$  entonces  
    devuelve  $n$   
  en otro caso  
    vector[0... $n + 1$ ]  
    vector[0]  $\leftarrow 0$   
    vector[1]  $\leftarrow 1$   
    para  $k$  desde 2 hasta  $n + 1$  hacer  
      vector[ $k$ ]  $\leftarrow$  vector[ $k - 1$ ] + vector[ $k - 2$ ]  
    devuelve vector[ $n$ ]
```



```
//Algorithm 5
def fib5(n : Int) : Int = {
  if (n < 2){
    return n
  } else {
    var m = n + 1
    var arr = Array.range(0, m)
    arr(0) = 0
    arr(1) = 1
    for(k <- 2 to (n + 1) ){
      arr(k) = arr(k - 1) + arr(k - 2)
    }
    return arr(n)
  }
}
```

```
scala> fib5(8)
res6: Int = 21
```