



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®



# **Tecnológico Nacional De México**

**Instituto Tecnológico De Tijuana**

**Subdirección Académica**

**Departamento de Sistemas y Computación**

**Semestre Feb - Jun 2022**

**Ingeniería Informática**

**Datos Masivos**

**Práctica 4**

**Unidad 1**

**Hilares Lazareno Raymundo**

**No. 17212339**

**Ramos Verdín Paula Andrea**

**No. 18210721**

**Romero Hernandez Jose Christian**

**09 de Marzo del 2022**

## 1.- Algoritmo 1 Recursiva descendente

Declaramos la función llamada fibonacci con una variable n, si n llega a ser menor a 2 el valor asignado se retorna, si este no es retornado la suma entre n - 1 y n - 2 llama a la función de cada una.

```
//1. Algoritmo 1 Recursiva descendente

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

## 2.- Algoritmo Fórmula explícita

Declaramos las variables p y j estos deben ser de tipo Double junto con la función fibonacci1 y la variable n, primero se realiza la función if donde n es menor a 2, este retorna el valor de n, si no se realiza este las siguientes dos operaciones, primero p donde debe suma 1 son la raíz de 5 dividido entre 2 el resultado y j donde resta la elevación de exponentes entre la variable p y n, donde p se eleva a la n y 1 - p elevado a n para ser dividido entre la raíz de 5 y retornar a j.

```
//2. Algoritmo 2 Formula explicita

var p: Double = 0
var j: Double = 0

def fibonacci1(n:Double):Double =
{
    if(n<2)
    return n
    else
    {
        p = ((1 + Math.sqrt(5))/2)
        j = ((Math.pow(p,n)- Math.pow((1-p), n))/ Math.sqrt (5))
        return j
    }
}
```

### 3.- Algoritmo versión Iterativa

Declaramos la función Fibonacci2 con la variable n este debe ser tipo entero, se declaran las variables a, b siendo 1 y c, realizamos un ciclo desde 0 hasta el valor de n sea asignado, se realizan las 3 operaciones y al final se retorna el valor de a.

```
//3. Algoritmo 3 Version Iterativa

def fibonacci2(n2:Int): Int =
{
  var a = 0
  var b = 1
  var c = 0

  for (k <- Range(0, n2))
  {
    c = b + a
    a = b
    b = c
  }

  return a
}
```

### 4.- Algoritmo Iterativa con 2 Variables

```
//4. Algoritmo Iterativa con 2 variables

def fibonacci3(n:Int): Int =
{
  var a = 0;
  var b = 1;

  for(k <- Range(0, n))
  {
    b = b + a;
    a = b - a;
  }

  return b;
}
```



## 5.- Algoritmo Iterativo con Arreglos

```
//5. Algoritmo Iterativo con Arreglos

def fibonacci4(n: Int): Int =
{
  if(n < 2)
  {
    return n;
  }
  else
  {
    var arre = Array.ofDim[Int](n + 1);
    arre(0) = 0;
    arre(1) = 1;

    for(k <- Range(2, n + 1))
    {
      arre(k) = ar(k - 1) + arre(k - 2);
    }

    return arre(n);
  }
}
```