

Tecnológico Nacional De México Instituto Tecnológico De Tijuana

Subdirección Académica

Departamento De Sistemas Y Computación

**SEMESTRE:**

Enero – junio 2020

**CARRERA:**

Ing. Tecnologías De Información Y Comunicación

**NOMBRE DEL TRABAJO:**

Ejercicios De Fibonacci

**UNIDAD A EVALUAR:**

Unidad 1

**NOMBRE DEL ALUMNO:**

Garcia Bautista Ana Laura # 15210793

Enciso Maldonado Aileen Yurely #15210329

**MAESTRO (A):**

Romero Hernández José Christian

```
//5 algoritmos de la secuencia de Fibonacci.
```

```
//Garcia Bautista Ana Laura #15210793
```

```
//1.-
```

En este quinto algoritmo se realiza una función que solicita un valor entero (Int) luego devuelve un valor entero

//con decimales (Doble) Se crea una matriz que comienza de 0 a (n + 1) si la variable (n) es menor que 2, esa misma

///variable se devuelve como resultado de lo contrario, el vector con espacio (0) tendrá un valor de cero (0) y

//el vector con espacio (1) tendrá un valor de uno (1) Comience a pedalear con a para el vector, el

//resultado será la variable (n) de acuerdo con el vector de

```
def función4 ( n : Int ) : Doble =  
  {  
  
    val vector = Array .range ( 0 , n + 1 )  
    si (n < 2 )  
    {  
      volver (n)  
    }  
    mas  
    {  
      vector ( 0 ) = 0  
      vector ( 1 ) = 1  
  
      para (k < - 2 a n)  
      {  
        vector (k) = vector (k - 1 ) + vector (k - 2 )  
      }  
  
      retorno del vector establecido (n)  
    }  
  }  
función4 ( 10 )
```

```
// 5 algoritmos de la secuencia de Fibonacci.
```

```
//Garcia Bautista Ana Laura #15210793
```

```
//2.- Versión con fórmula explícita
```

// Si el número ingresado en la función es menor que 2, el número ingresado será devuelto

// si no es menor que 2, se hará lo siguiente la fórmula se divide en partes para finalmente crear el resultado

```
def función1 ( n : Doble ) : Doble =
```

```

{
  si (n < 2 )
  {
    volver n
  }

  {
    var p = (( 1 + ( Math .sqrt ( 5 ))) / 2 )
    var a = Math .pow (p, n)
    var b = Math .pow ( ( 1 - p), n)
    var j = ((a - (b))) / ( Math .sqrt ( 5 ))
    return (j)
  }
}

function1 ( 10 )

```

```

//Garcia Bautista Ana Laura #15210793
// 5 algoritmos de la secuencia de Fibonacci.

//3.-
En este cuarto trimestre se agregó una función que después de haber realizado las operaciones correspondientes,
//la función nos dará un resultado (retorno) este debe ser un valor entero (Int) Un ciclo (para) comienza donde k = 1,
//comenzará a ciclar hasta que se convierta en (n), (n) representa el valor que se ingresará en la función
//dependiendo del ciclo (para) las variables (b, a) comenzarán a cambiar su resultado hasta el final del ciclo (para)
//el resultado se devolverá con (return)
def function3 ( n : Int ) : Int = {
var a = 0
var b = 1

para (k < - 1 a n)
{
  b = b + a
  a = b - a
}

return (a)
}

función3 ( 10 )

```

```
//5 algoritmos de la secuencia de Fibonacci.
```

```
//Garcia Bautista Ana Laura #15210793
```

```
//4.-
```

En este sexto algoritmo se agregó una función que después de haber realizado las operaciones correspondientes,

//la función nos dará un resultado (retorno) este debe ser un valor entero con puntos decimales (Doble)

//si el valor ingresado es menor o igual a 0, entonces ese valor se devolverá De lo contrario, tendrá

//que realizar una serie de operaciones de iniciar un ciclo ( while) donde las variables comenzarán a cambiar el valor

//dependiendo de la iteración del ciclo si la variable (i) es impar, se realizarán diferentes operaciones

// Si la variable (i) es par, diferentes operaciones se harán la variable (i) comenzará a cambiar el valor cada

// vez que se ingrese el ciclo hasta que salga del ciclo y se devuelva la suma de (a + b)

```
def función5 ( n : Doble ) : Doble =  
{
```

```
    si (n <= 0 )  
{  
        volver (n)  
    }
```

```
otra persona
```

```
{  
    var i : Doble = n - 1  
    var auxOne : Doble = 0  
    var auxTwo : Doble = 1  
    var a : Doble = auxTwo  
    var b : Doble = auxOne  
    var c : Doble = auxOne  
    var d : Doble = auxTwo
```

```
    mientras (i > 0 )  
{
```

```
        si (i % 2 == 1 )  
{
```

```
            auxOne = (d * b) + (c * a)
```

```

    auxTwo = ((d + (b * a)) + (c * b))
    a = auxOne
    b = auxTwo
}

otra cosa
{
    var pow1 = Math.pow (c, 2 )
    var pow2 = Math.pow (d, 2 )
    auxOne = pow1 + pow2
    auxTwo = (d * (( 2 * (c)) + d))
    c = auxOne
    d = auxTwo
}
i = (i / 2 )
}
retorno (a + b)
}
}
Función5 ( 9 )

```

```

// Práctica 2
//Garcia Bautista Ana Laura #15210793

// 1. Crea una lista llamad "lista" con los elementos "rojo", "blanco", "negro"
import scala.collection.mutable.ListBuffer
var lista = collection.mutable.ListBuffer("rojo","blanco","negro")

// 2. Añadir 5 elementos mas a "lista" "verde", "amarillo", "azul", "naranja", "perla"
lista += "verde"
lista += "amarillo"
lista += "azul"
lista += "naranja"
lista += "perla"

// 3. Traer los elementos de "lista" "verde", "amarillo", "azul"
lista slice (3,6)

// 4. Crea un arreglo de número en rango del 1-1000 en pasos de 5 en 5
Array.range(1, 1000, 5)

```

```
// 5. Cuales son los elementos unicos de la lista Lista (1,3,3,4,6,7,3,7) usan conversion a
    conjuntos
lista.toSet

// 6. Crea un mapa mutable llamado nombres que contenga los siguientes "José", 20, "Luis",
24, "Ana", 23, "Susana", 27"
var names = collection.mutable.Map(("Jose", 20), ("Luis", 24), ("Ana", 23) ("Susana", 27))

// 6 a. Imprime todas las llaves del mapa
nombres.claves

// 7 b. Agrega el siguiente valor al mapa ("Miguel", 23)
nombres += ("Miguel" -> 23)
```