

Ingeniería en sistemas computaciones

Datos masivos



Practica 2

Alumnos:

Marquez Millan Seashell Vanessa

Galaviz Lona Oscar Eduardo

Desarrollo

La serie de Fibonacci es una serie numérica en la que cada número es la suma de los dos anteriores, por ejemplo 1, 1, 2, 3, 5, 8, 13, 21, 34, etc.

En el presente documento se realizaron algoritmos en el lenguaje de programación Scala, en donde se probaron diferentes tipos de algoritmos y cada uno de ellos cumple con el objetivo.

Algoritmo 1

Es la forma más común de realizar el algoritmo de Fibonacci en una función, se basa en un if y else, en donde se llama a la recursividad de la misma función ya que el problema en sí es recursivo.

```
def Serie1(num:Int): Int={
  if(num<2){
    return num
  }else{
    return Serie1(num-1)+Serie1(num-2)
  }
}

println(Serie1(10))
```

Algoritmo 2

Para este algoritmo en el caso de scala es necesario la libreria de math ya que se necesita de la raiz y la exponencial debido a que no hay recursividad, además que en este caso estamos trabajando con números en fracción

```
import scala.math.sqrt
import scala.math.pow
def Serie2(num: Double):Double ={
  var aux = 0.0
  if(num<2){
    return num
  }else{
    val aux2 = (1+sqrt(5))/2
    aux=pow(aux2,num)
    aux= aux-pow((1.0-aux2),num)
    aux = aux /sqrt(5)
  }
  return aux
}
println(Serie2(8))
```

Algoritmo 3

Otra de las formas es con la creación de variables auxiliares que nos ayuden a ir guardando los valores en las diferentes interacciones.

```
def Serie3(num: Int): Int={
  var a =0
  var b =0
  var c=1
  for(k<- 0 until num){
```

```
        b=c+a
        a=c
        c=b
    }
    return a
}
```

Algoritmo 4

Esta es una forma muy similar a la 3 donde utilizamos variables auxiliares para hacer nuestro recorrido y sumatoria, solo que en este caso de una forma diferente.

```
def Serie4(num: Int): Int={
    var a1 = 0
    var b1 = 0
    for(k<-0 until num){
        b1 = b1 + a1
        a1 = b1 - a1
    }
    return b1
}
println(Serie4(9))
```

Algoritmo 5

En este caso se utilizan variables auxiliares pero tambien se utiliza un for para llevar el control de las iteraciones.

```
def Serie5(num: Int): Int={
    var n = 0
    var v = 0
    var v1 = 0
    var v2 = 0
    if(k<-2){
        v = n+1
        for(k<-2 until num){
            v = v1(k-1)+v2(k-2)
        }
        return v(n)
    }
    return n
}
println(Serie5(9))
```