Sala 4
Maite Abril, Johan Alonso, Judy Heredia, Gloria Forero

Encuentro Intermedio Actividad 1

Determinar la cantidad de operaciones elementales de los siguientes pseudocódigos:

1. **Hallar el área de un triángulo**

```
Inicio
  Imprimir "Ingrese la base del triángulo:"
   Leer base
     Imprimir "Ingrese la altura del triángulo:"
   Leer altura
     area = (1 / 2) * base * altura      **/ aquí tenemos 4 operaciones (1 asignacion, 1 División y
2    multiplicaciones)
     Imprimir "El área del triángulo es:", area
Fin
```

En total se tendian 4 operaciones

2. **Calcular la Calificación Definitiva**

```
Inicio
   Imprimir "Ingrese la primera nota:"
   Leer nota1
     Imprimir "Ingrese la segunda nota:"
   Leer nota2
     Imprimir "Ingrese la tercera nota:"
   Leer nota3
     calificacionDefinitiva = (0.30 * nota1) + (0.35 * nota2) + (0.35 * nota3) **/ 1 asignacion 3
multiplicaciones y 2 sumas
     Imprimir "La calificación definitiva del estudiante es:", calificacionDefinitiva
Fin
```

En total tenemos 6 operaciones

3. **Conversión de euros a pesos**

```
Inicio
   Función convertirEurosAPesos(euros, tipoCambio)
     pesos = euros * tipoCambio   **/ 1 asignacion 1 multiplicacion
     Retornar pesos  **/ tenemos 1 “return”
   FinFunción
```

```
  Imprimir "Ingrese la cantidad en euros:"
  Leer euros

  Imprimir "Ingrese el tipo de cambio (euros a pesos):"
  Leer tipoCambio

  cantidadPesos = convertirEurosAPesos(euros, tipoCambio) **/ 1 asignacion

  Imprimir "La cantidad en pesos es:", cantidadPesos

Fin
```

Tenemos 4 Operaciones

Actividad 2

Se analizará otro caso usando la secuencia de Fibonacci.

• En este algoritmo el tamaño del proceso es el número de términos que se quieren calcular de la secuencia • Para N: =10 significa que se quiere calcular el décimo término

• Desarrollar el algoritmo, obtener la ecuación temporal, e indicar la cantidad de operaciones elementales para N: =10

Seformula el Algoritmo con el uso de recurrencia:

```
def fibonacci(n):
        if n == 0:
            return 0
        elif n == 1:
            return 1
        else:
        return fibonacci(n-1) + fibonacci(n-2)
```

considerando cuántas veces se llama a la función recursivamente. Sea T(n) el tiempo necesario para calcular F(n) La función recursiva realiza dos llamadas a sí misma para calcular F(n−1) y F(n−2). Así, la ecuación de recurrencia es :

T(n)=T(n−1) +T(n−2) +O (1)

Aquí, O (1) representa el tiempo constante para las operaciones adicionales como la suma. La ecuación es similar a la propia definición de la secuencia de Fibonacci, lo que indica que el tiempo de ejecución crece exponencialmente con n. La solución a esta ecuación de recurrencia es aproximadamente T(n)=O(2^n).

Por ende, el tiempo de ejecucion crece de manera exponencial, porporcionalmente al crecimiento de “n”.

De esta manera podemos decir que el calculo para N= 10 requiere del llamado de los n anteriores, por ejemplo, para ejecutar N=10 tendra que llamar a N=9 y a N=8 asi:

‘fibonacci (10) llama a fibonacci (9) y fibonacci (8)’

Deesta manera cada llamada realiza 2 comparaciones y 1 suma por llamado, pero al ser N=10 se podira decir que estas operaciones se repiten 2^9 veces

Entonces el número total de llamadas recursivas de la función Fibonacci:

T(n)=2n−1T(n) =2n−1

Para n=10

T (10) =2^ (10−1) =2(^9) =512.

Entonces:

- **Número total de Comparaciones**: 512 llamadas×2 comparaciones=1024
- **Número total de Sumas**: 512 llamadas×1