TALLER ENCUENTRO INTERMEDIO DESARROLLO DE SOFTWARE

Nombres: Johann Lopez, Oscar Velandia, Leydi Beltran, Alejandra Cruz y Sergio Cruz

Actividad 1

Determinar la cantidad de operaciones elementales de los siguientes pseudocódigos:

- **Hallar el área de un triángulo**

Inicio

Imprimir "Ingrese la base del triángulo:" Operación de salida

Leer base Operación de entrada

Imprimir "Ingrese la altura del triángulo:" Operación de salida

Leer altura Operación de entrada

area = (1 / 2) * base * altura Operación aritmética y asignación: 1 division, 2 multiplicaciones y 1 asignación

Imprimir "El área del triángulo es:", area Operación de salida

Fin

RTA/: Hay 4 operaciones elementales

- **Calcular la Calificación Definitiva**

Inicio

Imprimir "Ingrese la primera nota:" Operación de salida

Leer nota1 Operación de entrada

Imprimir "Ingrese la segunda nota:" Operación de salida

Leer nota2 Operación de entrada

Imprimir "Ingrese la tercera nota:" Operación de salida

Leer nota3 Operación de entrada

calificacionDefinitiva = (0.30 * nota1) + (0.35 * nota2) + (0.35 * nota3) Operación aritmética y asignación: 3 multiplicaciones, 2 suma y 1 asignación

Imprimir "La calificación definitiva del estudiante es:", calificacionDefinitiva Operación de salida

Fin

RTA/: Hay 6 operaciones elementales

- **Conversión de euros a pesos**

Inicio

Función convertirEurosAPesos(euros, tipoCambio)

pesos = euros * tipoCambio Multiplicación y asignación

Retornar pesos Operación de retorno

FinFunción

Imprimir "Ingrese la cantidad en euros:" Operación de salida

Leer euros Operación de entrada

Imprimir "Ingrese el tipo de cambio (euros a pesos):" Operación de salida

Leer tipoCambio Operación de entrada

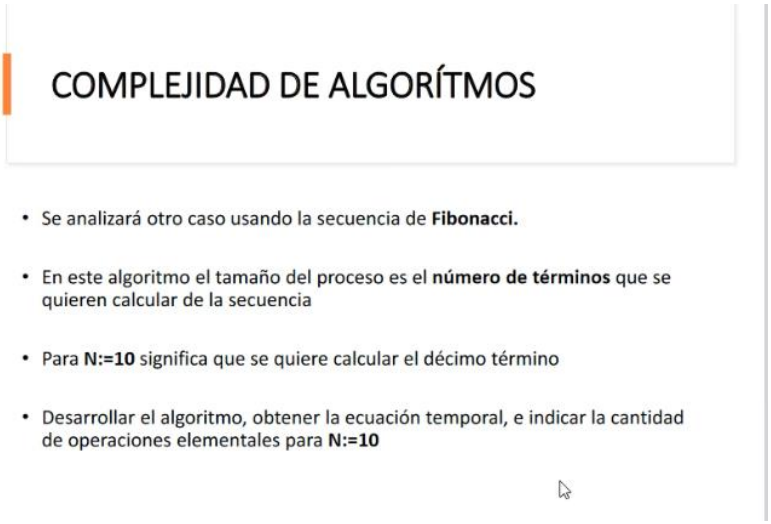
cantidadPesos = convertirEurosAPesos(euros, tipoCambio) Operación llamada función

Imprimir "La cantidad en pesos es:", cantidadPesos Operación de salida

Fin

RTA/: Hay 4 operaciones elementales

Actividad 2



- Pseudocódigo

Clase Fibonacci:
Definir mapa memo

Método fibonacci(n):
Si memo contiene n:
Retornar memo[n]
Si n <= 0:
Retornar 0
Si n == 1:
Retornar 1
Sino:
resultado = fibonacci(n - 1) + fibonacci(n - 2)
memo[n] = resultado
Retornar resultado

Método principal:
Crear instancia de Fibonacci
Definir n = 10
resultado = fibonacci(n)
Imprimir "El décimo término de la secuencia de Fibonacci es: " + resultado
Explicación:
1. Clase Fibonacci: Define un mapa memo para almacenar los resultados de los cálculos anteriores.
2. Método fibonacci(n):
○ Si memo contiene el valor de n, retorna el valor almacenado.
○ Si n es menor o igual a 0, retorna 0.
○ Si n es igual a 1, retorna 1.
○ De lo contrario, calcula el término de Fibonacci como la suma de los dos términos anteriores, almacena el resultado en memo y lo retorna.
3. Método principal:
○ Crea una instancia de la clase Fibonacci.
○ Define n como 10.
○ Calcula el décimo término de la secuencia de Fibonacci.
○ Imprime el resultado.
Este pseudocódigo sigue la misma lógica que el código en Java, pero está escrito de manera más abstracta para facilitar la comprensión.

- Código Java de la secuencia de Fibonacci

```
import java.util.HashMap;

import java.util.Map;


public class Fibonacci {

private Map<Integer, Integer> memo; public Fibonacci() {

this.memo = new HashMap<>(); }

public int fibonacci(int n) { if (memo.containsKey(n)) { return memo.get(n); }

if (n <= 0) { return 0; }

else if (n == 1) { return 1; }

else { int result = fibonacci(n - 1) + fibonacci(n - 2);

memo.put(n, result); return result; } }


public static void main(String[] args) { Fibonacci fib = new Fibonacci(); int n = 10; int resultado = fib.fibonacci(n);


System.out.println("El décimo término de la secuencia de Fibonacci es: " + resultado); } }
```

Ecuación temporal

$F(0)=0$

$F(1)=1$

$F(n)=F(n-1)+F(n-2)$

Operaciones elementales:

● **Acceso y almacenamiento en el mapa** memo: Cada vez que se accede o se almacena un

valor en memo, se realiza una operación elemental.
● **Sumas**: Cada vez que se suman dos términos de Fibonacci, se realiza una operación elemental.
Cantidad de operaciones para
Para n = 10, el algoritmo realiza las siguientes operaciones:
1. Accesos y almacenamiento en memo:
○ Se accede y almacena el valor de cada término de Fibonacci desde fibonacci(0) hasta fibonacci(10).
○ Esto implica 11 operaciones de acceso y almacenamiento (una por cada término).
2. Sumas:
○ Se realizan sumas para cada término desde fibonacci(2) hasta fibonacci(10).
○ Esto implica 9 operaciones de suma (una por cada término desde fibonacci(2) hasta fibonacci(10)).
Total de operaciones elementales:
● **Accesos y almacenamiento en** memo: 11 operaciones.
● **Sumas**: 9 operaciones.
● **Inicialización**: 1 operación. **Total**: 11 + 9 + 1 = 21 operaciones elementales