TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Estructura de datos

Titulo:

Funciones Recursivas + trazos

Estudiante:

Kevin Girón

Fecha:

21/1/26

El Ministerio de Inclusión Económica y Social (MIES) necesita un sistema para calcular el subsidio mensual de familias beneficiarias del Bono de Desarrollo Humano. **El monto se calcula recursivamente según la estructura familiar:**

- Monto base: $55.00 por el titular del núcleo familiar (caso base).
- Por cada dependiente: Se agrega $15.00 adicionales, recorriendo recursivamente cada miembro del hogar (hijos, adultos mayores a cargo).
- Componente variable: Familias en zonas rurales de la Sierra o Amazonía reciben un factor multiplicador de 1.15.

La recursión es una herramienta poderosa para resolver problemas complejos (como el recorrido de árboles genealógicos o estructuras jerárquicas) descomponiéndolos en subproblemas idénticos más pequeños. Sin embargo, un diseño descuidado puede llevar a un desbordamiento de pila (Stack Overflow). En esta práctica, implementarás funciones recursivas robustas y analizarás su comportamiento en memoria paso a paso.

1. **Implementación Segura:**

Implementa en Java/Python la función calcularSubsidioBDH(dependientes, esZonaRural) que calcule recursivamente el monto del Bono de Desarrollo Humano para una familia.

```java
public class calcularrecursiva {

    private static double Monto_base= 55.0;
    private static double Monto_dependiente= 15.0;
    private static double Factor_rural= 1.15;
    private static double CalcularRecursivo(int dependientes, boolean esRural){
        // caso exitoso donde no hay dependedientes ni rural regresa solo se llama una vez
        if(dependientes == 0){
            return esRural ? Monto_base * Factor_rural : Monto_base;
        }

        // recursividad

        double montoAnterior = CalcularRecursivo(dependientes - 1, esRural);
        double adicional = esRural ? Monto_dependiente * Factor_rural : Monto_dependiente;

        return montoAnterior + adicional;
    }

    Run | Debug
    public void main(String [] args){
        System.out.println("Caso 1: Zona Urbana " + CalcularRecursivo(dependientes: 0, esRural: false));
        System.out.println("Caso 2: Zona Urbana " + CalcularRecursivo(dependientes: 3, esRural: false));
        System.out.println("Caso 3: Zona Urbana " + CalcularRecursivo(dependientes: 7, esRural: true));
    }
}

```

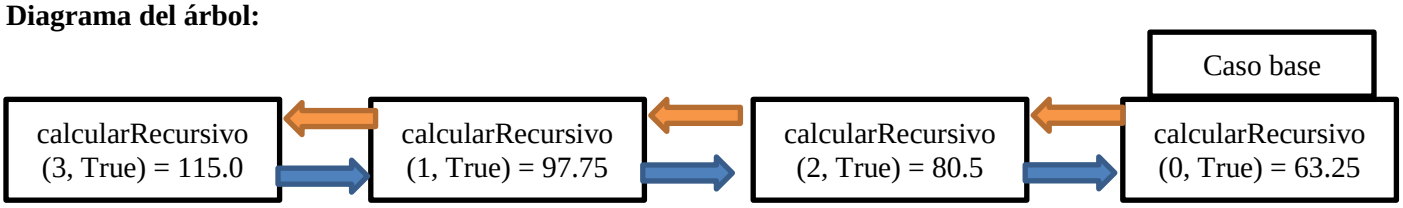
Resultados:

```
PS C:\Users\Kevin\Documents\P3.1_KevinGiron> java .\calcularrecursiva.java
Caso 1: Zona Urbana 55.0
Caso 2: Zona Urbana 100.0
Caso 3: Zona Urbana 184.0
```

2. **Traza Manual (Árbol)**

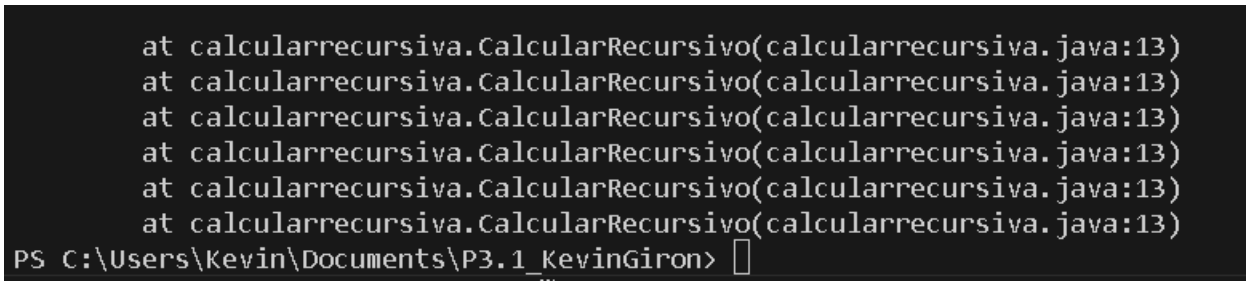
Para una familia con 3 dependientes (o n=5 en Fibonacci), dibuja a mano o digitalmente el Árbol de Recursión completo, mostrando cada llamada como un nodo y el orden de retorno de los valores.

Diagrama del árbol:



3. **Provocación de Error:**

Error StackOverflowError:



Justificación técnica:

El StackOverflowError se produce porque el método CalcularRecursivo se llama a sí mismo de forma infinita sin alcanzar nunca un caso base válido. En cada llamada recursiva, se reserva un nuevo frame en la pila de ejecución que contiene las variables locales y el contexto del método. Debido a que el valor de control no cambia o la condición de salida es inalcanzable, las llamadas se acumulan hasta que la pila se llena.

Reflexión: ¿Qué pasaría si el sistema del MIES tuviera este bug con millones de beneficiarios? Explica el impacto técnico y social.

Si un sistema del MIES tuviera este bug y procesara millones de beneficiarios, el impacto técnico sería crítico: caídas constantes del sistema, saturación de servidores y fallos en el cálculo de subsidios. Socialmente, miles de familias quedarían sin pagos o con retrasos, generando desconfianza en la institución, afectación económica directa y posibles conflictos sociales.