



Universidad Tecmilenio

Campus Las Torres

Materia: Estructura de Datos

Actividad 3 - Explicación

Alumno: Patricio Calvo

Matricula: 07097795

Septiembre / 2025

### **Actividad 3: Recursividad y Algoritmos de Divide y Vencerás**

#### **Explicación de uso y explicación de las opciones del programa**

En esta actividad se desarrolló un programa en Java que permite al usuario interactuar con tres problemas clásicos de programación: la serie de Fibonacci, la suma de subconjuntos y la resolución de Sudoku mediante backtracking. El objetivo es que el usuario comprenda cómo funcionan los algoritmos recursivos y de divide y vencerás, y pueda experimentar con ellos de manera sencilla desde la consola.

A continuación, se explica detalladamente cómo utilizar cada una de las opciones del programa, acompañando la explicación con ejemplos prácticos y recomendaciones para el correcto uso.

#### **Opción 1: Calcular término de Fibonacci**

La primera opción del menú permite calcular el término  $n$  de la serie de Fibonacci. El usuario debe ingresar el valor de  $n$ , y el programa le mostrará el resultado correspondiente.

Por ejemplo, si el usuario selecciona la opción 1 y escribe el número 7, el programa responderá:

```
Seleccione una opción: 1
Ingrese el valor de n: 7
El término 7 de Fibonacci es: 13
```

Esta opción es útil para entender cómo la recursividad puede utilizarse para calcular términos de una serie matemática de manera sencilla.

## **Opción 2: Verificar suma de subconjuntos**

La segunda opción permite al usuario verificar si existe un subconjunto de números que sumen un valor objetivo. El usuario debe escribir los números del conjunto separados por espacio y luego indicar el valor objetivo que desea buscar.

Por ejemplo, si el usuario ingresa los números 3 34 4 12 5 2 y busca la suma 9, el programa responderá:

Seleccione una opción: 2

Escriba los números separados por espacio (ejemplo: 1 2 3 4):

3 34 4 12 5 2

¿Qué suma objetivo desea buscar?: 9

¡Sí existe un subconjunto que suma 9!

Si no existe ningún subconjunto que sume el valor objetivo, el programa lo indicará claramente:

No existe subconjunto que sume 100.

Esta opción ayuda a visualizar cómo los algoritmos recursivos pueden explorar todas las combinaciones posibles de un conjunto para encontrar una solución.

### Opción 3: Resolver Sudoku

La tercera opción es una de las más interesantes, ya que permite al usuario resolver un Sudoku utilizando algoritmos de backtracking. El programa muestra un tablero de Sudoku vacío, representado con ceros, y le pide al usuario que lo copie, reemplace los ceros por los números conocidos de su propio Sudoku y lo pegue tal cual en la consola.

El tablero que se muestra es el siguiente:

```
+-----+-----+-----+
| 0 0 0 | 0 0 0 | 0 0 0 |
| 0 0 0 | 0 0 0 | 0 0 0 |
| 0 0 0 | 0 0 0 | 0 0 0 |
+-----+-----+-----+
| 0 0 0 | 0 0 0 | 0 0 0 |
| 0 0 0 | 0 0 0 | 0 0 0 |
| 0 0 0 | 0 0 0 | 0 0 0 |
+-----+-----+-----+
| 0 0 0 | 0 0 0 | 0 0 0 |
| 0 0 0 | 0 0 0 | 0 0 0 |
| 0 0 0 | 0 0 0 | 0 0 0 |
+-----+-----+-----+
```

El usuario debe copiar este tablero, reemplazar los ceros por los números de su Sudoku y pegarlo en la consola, incluyendo los símbolos y espacios exactamente igual.

Por ejemplo, si el usuario tiene el siguiente Sudoku:

```
+-----+-----+-----+
| 5 3 0 | 0 7 0 | 0 0 0 |
| 6 0 0 | 1 9 5 | 0 0 0 |
| 0 9 8 | 0 0 0 | 0 6 0 |
+-----+-----+-----+
| 8 0 0 | 0 6 0 | 0 0 3 |
| 4 0 0 | 8 0 3 | 0 0 1 |
| 7 0 0 | 0 2 0 | 0 0 6 |
+-----+-----+-----+
| 0 6 0 | 0 0 0 | 2 8 0 |
| 0 0 0 | 4 1 9 | 0 0 5 |
| 0 0 0 | 0 8 0 | 0 7 9 |
+-----+-----+-----+
```

El programa procesará el tablero y, si es resoluble, mostrará la solución con el mensaje:

Solución del Sudoku:

```
+-----+-----+-----+
| 5 3 4 | 6 7 8 | 9 1 2 |
| 6 7 2 | 1 9 5 | 3 4 8 |
| 1 9 8 | 3 4 2 | 5 6 7 |
+-----+-----+-----+
| 8 5 9 | 7 6 1 | 4 2 3 |
| 4 2 6 | 8 5 3 | 7 9 1 |
| 7 1 3 | 9 2 4 | 8 5 6 |
+-----+-----+-----+
| 9 6 1 | 5 3 7 | 2 8 4 |
| 2 8 7 | 4 1 9 | 6 3 5 |
| 3 4 5 | 2 8 6 | 1 7 9 |
+-----+-----+-----+
```

Esta opción permite al usuario experimentar con la resolución automática de Sudokus, observando cómo el algoritmo encuentra la solución paso a paso.

#### **Opción 4: Salir**

La última opción simplemente finaliza el programa.

#### **Recomendaciones generales**

- Si se ingresa una letra o símbolo donde se espera un número, el programa solicitará que se intente de nuevo.
- Para el Sudoku, es fundamental pegar el tablero exactamente como se muestra, incluyendo los símbolos | y +, para que el programa lo procese correctamente.
- El programa está diseñado para ser intuitivo y guiar al usuario en cada paso, mostrando mensajes claros en caso de error.

#### **Conclusión**

Este programa es una herramienta didáctica para comprender y experimentar con algoritmos recursivos y de backtracking. Permite al usuario interactuar con problemas clásicos y observar cómo la recursividad y el divide y vencerás pueden aplicarse a la resolución de problemas reales.