	Carátula para entrega de prácticas

Facultad de Ingeniería	Laboratorios de docencia

Laboratorio de Computación
Salas A y B

Profesor(a): ______________________________

Asignatura: ______________________________

Grupo: ______________________________

No de Práctica(s): Práctica 1 & 2

Integrante(s): 322246320

322267567

322012051

322236688

322330872

No. de lista o brigada: ______________________________

Semestre ______________________________

Fecha de entrega: ______________________________

Observaciones: ______________________________

CALIFICACIÓN: ________________

Índice

1. Introducción

- **Planteamiento del Problema:** Crear solución a la serie de problemas propuestos, y utilizar funciones recursivas para dicha solución.
- **Motivación:** Consideramos que es importante el realizar esta práctica para fortalecer las bases de la programación en Java, mediante la implementación de funciones recursivas para resolución de factorial, serie de Fibonacci y conjetura de Collatz, así mismo la correcta estructuración para un menú de opciones.
- **Objetivos:** Aprender a realizar funciones recursivas y un menú de opciones en Java.

2. Marco Teórico

Factorial: Es el producto de todos los números enteros positivos desde 1 hasta ese número n. Su representación matemática es: n! = n * (n-1)! [1]

Serie de Fibonacci: Es una serie de números en la que cada número es una suma de los dos anteriores, empezando siempre ésta en 0 y 1 y terminando en n. Su representación matemática es: F(n) = F(n-1) + F(n-2). [2]

Conjetura de Collatz: Es un algoritmo que asegura que todo número entero positivo siempre llegara a 1 mediante la recursión de dos casos, su representación matemática es: f(n) = n/2 si n es par ; 3n + 1 si n es impar [3]

3. Desarrollo

Solución teórica de los ejercicios:

Para los tres ejercicios realizados en clase, empezamos planteando la parte teórica, buscando el caso base y después la operación que va a permitir que la fución realice lo requerido y de forma recursiva.

1. **Factorial**

   El caso base del factorial, es cuando **n** es igual a **1**, la fórmula del factorial es **n(n-1)**, entonces en el código, se rompe la recursividad cuando el número es igual a 1, y cuando no lo es, se regresa el número multiplicado por la misma función factorial pero disminuido en 1.

2. **Serie de Fibonacci**

   En fibonacci los casos base son cuando **n** es igual a **0** ó **1**. Para que la función haga lo que queremos, necesitamos sumar el numero anterior con el número anterior a ese hasta llegar al caso base, y es lo que hacemos en el código, si no se rompe la recursividad, se hace la suma de los dos números anteriores hasta que la recursividad se rompa.

3. **Conjetura de Collatz**

   En la Conjetura de Collatz se realiza una evaluación dentro de la función para determinar si es par o impar y realizar su correspondiente operación; si **n** es par se divide entre 2 **(n/2)** y se llama de nuevo a la función de forma recursiva, en caso de ser **n** impar se multiplica por 3 y se suma 1 **(n*3+1)**, pero siempre evaluando si se llegó al caso base; el caso base es llegar al **1**, pero anteriormente a este número entra al ciclo infinito del **4,2,1,4,2,1...**, por ello es necesario cerrar ese ciclo con el **1**.

4. Resultados

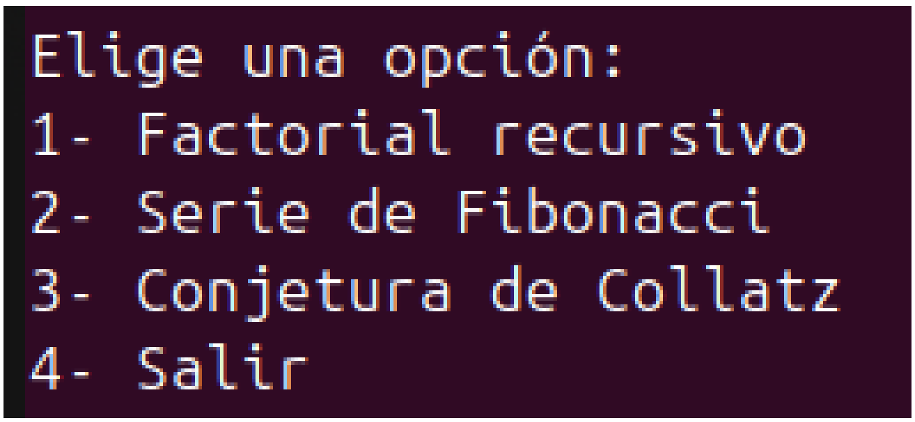

Figura 1: Todas las funciones integradas en un solo menú

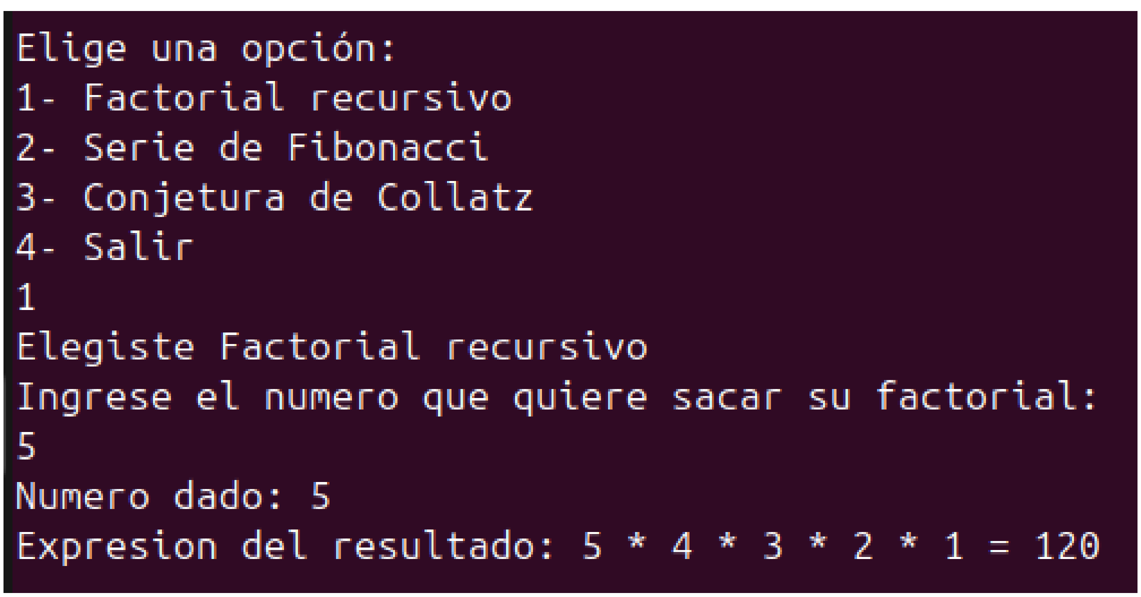

Figura 2: Opción 1, mostrando el factorial de un número ingresado y la operación que se realizo

```
Elige una opción:
1- Factorial recursivo
2- Serie de Fibonacci
3- Conjetura de Collatz
4- Salir
2
Elegiste Serie de Fibonacci
Cuantos terminos de la serie de Fibonacci quiere ver: 8
La serie de Fibonacci hasta el termino 8 es:
0 1 1 2 3 5 8 13
```

Figura 3: Opción 2, mostrando la serie de Fibonacci con los términos que el usuario solicito

```
Elige una opción:
1- Factorial recursivo
2- Serie de Fibonacci
3- Conjetura de Collatz
4- Salir
3
Elegiste Conjetura de Collatz
Ingrese el número con el que quiere hacer la conjetura de Collatz
15

 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1Fin
```

Figura 4: Opción 3, mostrando inmediatamente los resultados de las operaciones que realiza la conjetura

5. Conclusiones

La implementación del menú de opciones en Java permitió aplicar los conceptos teóricos vistos en clase mediante la resolución del factorial recursivo, la serie de Fibonacci y la conjetura de Collatz. Los objetivos de la práctica se cumplieron al desarrollar un menú con las funciones recursivas funcionales.

Referencias

[1] *¿Qué es un factorial? Cómo calcularlo con ejemplos*. 2025. URL: https://www-freecodecamp-org.translate.goog/news/what-is-a-factorial/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc (visitado 28-08-2025).

[2] *Secuencia de Fibonacci en Python: Aprende y explora técnicas de codificación*. 2025. URL: https://www.datacamp.com/es/tutorial/fibonacci-sequence-python (visitado 18-08-2025).

[3] *Conjetura de Collatz*. 2025. URL: https://geomaths-co-uk.translate.goog/2024/06/23/collatz-conjecture/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc (visitado 18-08-2025).

6. Reto para token

El triángulo de Pascal es una matriz de forma triangular, la cuál en cada una de sus filas empiezan y terminan en uno, y los dígitos son la suma de los dos dígitos encima de él.

Para manejar el triángulo de pascal en código, nos ayudamos usando el triángulo con 2 posiciones, una que mida las filas o pisos, y otra que mida las posiciones dentro del triángulo.

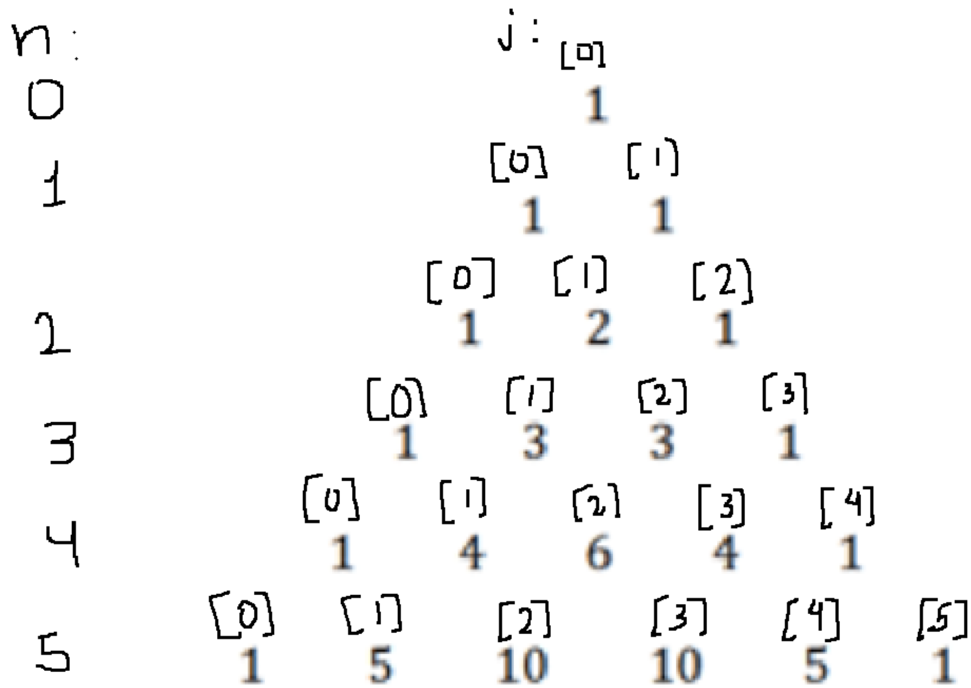

Figura 5: Forma gráfica de entender el código del Triángulo de Pascal

El caso base del triángulo de pascal es cuando llegamos a los bordes del triángulo, donde en esos casos siempre es uno, y si no se cumple el caso base, se hace la suma de la posición de **(n-1, j-1)** y **(n-1, j)**, lo que significa que hace la suma de los dos números que están en el piso de arriba justo encima del número a calcular.

Usamos una función recursiva, y lo que hace es calcular el número en la posición de **n** y **j** que es ingresada, usando la lógica explicada anteriormente, y para que se imprima la pirámide completa, usamos 2 ciclos **for** anidados que van iterando en las posiciones de la pirámide y calculando todos los números.

Usamos otro ciclo **for** que va imprimiendo espacios según el nivel en el que está para que la pirámide sí tenga forma de pirámide.

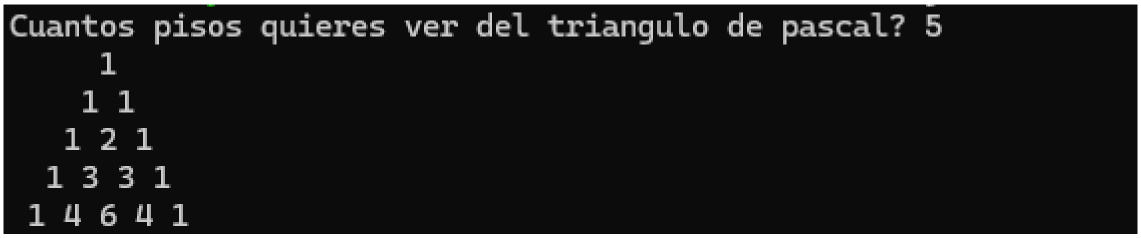

Figura 6: Código ejecutado