



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor(a): René Adrián Dávila Pérez

Asignatura: Programación Orientada a Objetos

Grupo: 7

No. de práctica(s): 1,2

Integrante(s): 320065570
425133840
423020252
322229916
425032224

No. de lista o brigada: 3

Semestre: 2026-1

Fecha de entrega: 27/08/2025

Observaciones:

CALIFICACIÓN: _____

Índice

1. Introducción	2
2. Marco Teórico	3
3. Desarrollo	4
4. Resultados	5
5. Conclusiones	8

1. Introducción

En cualquier lenguaje de programación, como Java, es fundamental comprender su estructura básica, métodos, sintaxis, estructuras de control, identificadores, operadores y tipos de datos para así poder hacer un uso óptimo del propio lenguaje de programación y poderle sacar provecho para realizar gran variedad de problemas que se presenten, desde matemáticos, pasando por problemas de gestión y procesamiento de datos; pudiendo seguir así con una larga lista de problemas por resolver.

Si no se realiza un primer acercamiento a la estructura general de cualquier lenguaje de programación, en este caso el lenguaje es Java, se puede caer muy fácil en un "bache", donde se desconozcan las herramientas necesarias para encontrar soluciones a problemas específicos, haciéndonos de esta manera obsoletos para poder realizar cualquier trabajo de una magnitud considerable.

2. Marco Teórico

Es importante tener un conocimiento general sobre las operaciones que son codificadas como parte de este trabajo, para así poder entender el significado de cada expresión para poder llevarla posteriormente al código.

El Factorial es una operación matemática, denotada por $n!$, donde n es un número natural; la operación consiste en el producto de todos los números naturales empezando por el 1 hasta llegar al número n , por ejemplo, $5! = 1*2*3*4*5$ [1]. Por su parte, la Serie o Sucesión de Fibonacci se trata de una secuencia infinita de números en la cual cada elemento es resultado de la suma de los elementos que le preceden [2].

Por otro lado, la Conjetura de Collatz es un problema matemático que aún no presenta solución; esta enuncia que todo número entero positivo siempre llegará a 1. Esto se logra a través de una serie de simples pasos: si el número de prueba es par, este se divide entre dos; por el contrario, si es impar, se multiplica por 3 y se le suma 1 para posteriormente volver a hacer la comparación de si es par o impar sucesivamente hasta llegar a 1 [3].

Para resolver estos problemas, aplicamos conceptos básicos de Java vistos en clase. Usamos la estructura de una clase con su método `main` como punto de entrada del programa. Implementamos métodos estáticos para cada algoritmo, lo que nos permitió organizar mejor el código. También utilizamos estructuras de control como el ciclo `do-while` para el menú y la sentencia `switch` para manejar las opciones, junto con la clase `Scanner` para la entrada de datos.

3. Desarrollo

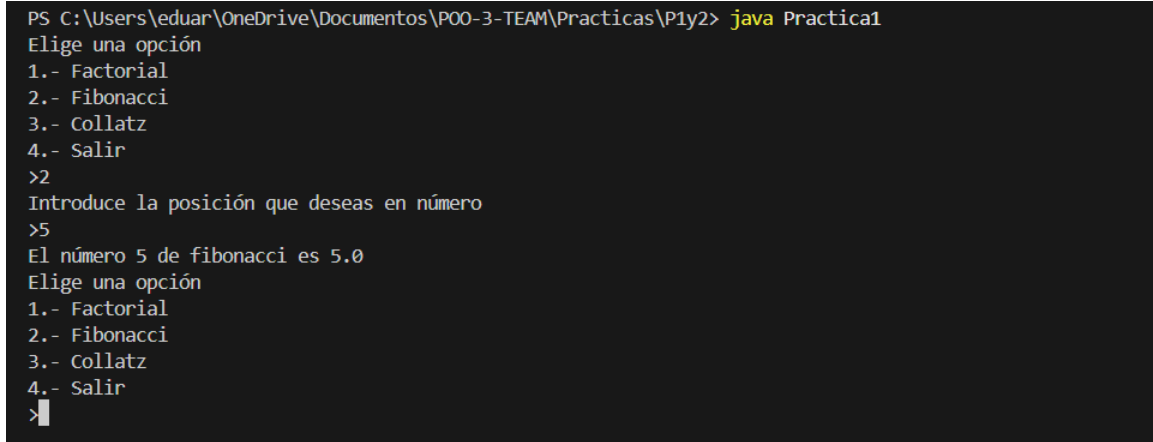
Para la implementación del código se creó un menú interactivo que deja al usuario elegir entre tres operaciones matemáticas. Se elaboró un bucle do-while, que asegura que el menú se repita hasta que el usuario elija salir. Cada algoritmo se desarrolló como un método independiente: Fibonacci calcula el valor en una posición dada mediante recursividad, es una función que al ser llamada usa un operador ternario que hace la pregunta Si $n == 0$, devuelve 0. si $n == 1$, devuelve 1. En cualquier otro caso, devuelve la suma de $\text{fibonacci}(n-1) + \text{fibonacci}(n-2)$.

El factorial resuelve el producto de un número y sus predecesores; igualmente, se ocupó un operador ternario que pregunta si $n == 1$, devuelve 1; si no, devuelve $n * \text{factorial}(n-1)$.

La conjetura de Collatz aplica las reglas de paridad hasta llegar a 1. En esta opción lo manejamos con un if else que establece que si es 1, se termina; si no, se ocupa el módulo entre 2. Si el módulo da 0, significa que es par y entonces se efectúa $\text{collatz}(n/2)$; de ser contrario, significa que es impar y se hace $\text{collatz}(3*n+1)$. Las pruebas se realizaron con entradas como 5 para Fibonacci (resultado: 5), 4 para factorial (resultado: 24) y 6 para Collatz (resultado: 1), validando que las salidas fueran consistentes con los resultados esperados. El programa respondió correctamente a opciones válidas e inválidas, finalizando únicamente cuando se seleccionó la opción de salida.

4. Resultados

Muestra de todos los algoritmos:



```
PS C:\Users\eduar\OneDrive\Documentos\POO-3-TEAM\Practicas\P1y2> java Practica1
Elige una opción
1.- Factorial
2.- Fibonacci
3.- Collatz
4.- Salir
>2
Introduce la posición que deseas en número
>5
El número 5 de fibonacci es 5.0
Elige una opción
1.- Factorial
2.- Fibonacci
3.- Collatz
4.- Salir
>
```

Figura 1: Muestra de Fibonacci

En la captura 1 mostramos la Opción 1 (Factorial) con entrada 4. Al introducir el valor de n , que en este caso será $n=4$, la función es $\text{factorial}(4)$, el código elabora sus operaciones que en este caso son $n * \text{factorial}(n-1)$, que nos da $4 * \text{factorial}(3)$, esto llama de nuevo a la función, pero esta vez como $\text{factorial}(3)$, se hacen las mismas operaciones pero esta vez son $3 * \text{factorial}(3-1)$, lo que disminuye la función a $\text{factorial}(2)$, pero sin olvidar todos los números anteriores por lo que se multiplicará, esto se reduce hasta que llegue a 1, donde al llegar al caso base (1), la recursión empieza a volver y a hacer las multiplicaciones de los números que se guardaron, como el número fue 4, esto da de regreso: $1 * 2 = 2$ a $2 * 3 = 6$ a $4 * 6 = 24$.

En la captura 2 donde se muestra la Opción 2 (Fibonacci) con entrada 5. Al insertar el valor de n , en este caso $n=5$, la función que se ejecuta es $\text{fibonacci}(5)$. Trabajamos las operaciones de la función que son $\text{fibonacci}(n-1) + \text{fibonacci}(n-2)$. Como el número ingresado fue 5, queda esta operación como $\text{fibonacci}(4) + \text{fibonacci}(3)$. Continúa la función con $n=4$, y es $\text{fibonacci}(3) + \text{fibonacci}(2)$, se reduce hasta los casos bases, y regresa, se empiezan a sumar todas las sumas que guardamos: $\text{fibonacci}(2) = 1 + 0 = 1$, $\text{fibonacci}(3) = 1 + 1 = 2$, $\text{fibonacci}(4) = 2 + 1 = 3$ y por último $\text{fibonacci}(5) = 3 + 2 = 5$. La última suma de esto da 5, por lo que $\text{fibonacci}(5)$ nos dio 5.

```

PS C:\Users\eduar\OneDrive\Documentos\P00-3-TEAM\Practicas\P1y2> java Practica1
Elige una opción
1.- Factorial
2.- Fibonacci
3.- Collatz
4.- Salir
>2
Introduce la posición que deseas en número
>5
El número 5 de fibonacci es 5.0
Elige una opción
1.- Factorial
2.- Fibonacci
3.- Collatz
4.- Salir
>

```

Figura 2: Caption

```

PS C:\Users\eduar\OneDrive\Documentos\P00-3-TEAM\Practicas\P1y2> java Practica1
Elige una opción
1.- Factorial
2.- Fibonacci
3.- Collatz
4.- Salir
>3
Introduce el número para la conjetura de Collatz
>6
6->3->10->5->16->8->4->2->1La conjetura de Collatz con 6 llega a 1
Elige una opción
1.- Factorial
2.- Fibonacci
3.- Collatz
4.- Salir
>4
Adioooooooooos
PS C:\Users\eduar\OneDrive\Documentos\P00-3-TEAM\Practicas\P1y2>

```

Figura 3: Muestra de Collatz

En la 3er Captura mostramos la Opción 3 del Menú (Collatz) con entrada 6, se imprime n que es 6 y después siguen las operaciones que trabaja esta función, son dos preguntas, como no es 1, las preguntas son si es par o impar el número n, esto lo hace usando $n \% 2 == 0$, como el número es 6, la operación es $6 \% 2 == 0$, entonces como es par, se divide entre 2, así que se llama a la función de nuevo pero ahora como $\text{collatz}(3)$, se efectúa la función, por lo que se imprime n, que es 3, prosigue con las operaciones del módulo, ya que es impar se hace la operación $\text{collatz}(3 * n + 1)$, en este caso es $\text{collatz}(3 * 3 + 1)$, esto es igual a $\text{collatz}(10)$, se llama la función de nuevo y así hasta que llegue eventualmente a 1. Como la función siempre imprime el número n en ese momento y después hace las operaciones, podemos observar todos los números por lo que pasó, el ejemplo que usamos fue 6, que nos dio: 6 -> 3 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1.

5. Conclusiones

Al final de la práctica, se logró el efectivo manejo de los conceptos básicos para codificar en el lenguaje Java, utilizando estructuras de control, bucles y funciones, e investigando sobre las palabras clave, reglas de sintaxis y estructura general de un programa en el lenguaje Java, para elaborar el programa abordado a lo largo de este reporte; el cual se trató de la codificación de 3 operaciones matemáticas, el factorial de un número, la sucesión de Fibonacci y la conjetura de Collatz, operaciones codificadas con éxito utilizando métodos iterativos para que las operaciones sean realizadas de una manera eficiente, además de permitir al usuario manipular las entradas de cada operación, en primer lugar accediendo a cada una de las operaciones a través de un menú y posteriormente permitiendo el ingreso de sus propias variables para cada una de las operaciones matemáticas.

De esta manera se puede constatar la correcta adquisición de los conocimientos para abordar futuros conceptos teóricos de la programación orientada a objetos y profundizar sobre nuestro conocimiento acerca de la plataforma Java para así poder tener las herramientas para afrontar desafíos futuros.

Referencias

- [1] Khan Academy. *La función factorial*. URL: <https://es.khanacademy.org/computing/computer-science/algorithms/recursive-algorithms/a/the-factorial-function>.
- [2] Universidad de Almeria. *Sucesión de Fibonacci*. URL: <https://www2.ual.es/jardinmatema/sucesion-de-fibonacci/>.
- [3] Geomathsblog. *Collatz conjecture*. URL: https://geomaths-co-uk.translate.google/2024/06/23/collatz-conjecture/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc.