



Universidad Nacional Autónoma de México

Ingeniería en Computacion

Programacion Orientada a objetos

REPORTE 1 y 2

ALUMNO:

322125337

322080869

322059179

323629814

322113536

Grupo:

Semestre:

2025-II

México,CDMX. Agosto 2025

Índice

1. Introducción	2
2. Marco Teórico	2
3. Desarrollo	2
4. Resultados	3
5. Conclusiones	6
6. Referencias bibliográficas	6

1. Introducción

En objetivo principal en esta practica es aplicar los conocimientos teóricos que hemos aprendido en clase como la aplicación de un menú de opciones en el código, con el fin de poderlo aplicarlo a problemas con los que ya hemos lidiado en otros lenguajes de programación. Esperamos poder implementar la mayoría de los conceptos en los ejercicios propuestos y analizar el como cada herramienta que está disponible en el lenguaje de java nos puede ser de ayuda para poder realizar los ejercicios de manera eficiente.

2. Marco Teórico

Para esta práctica se aplicaron conceptos de recursividad e iteración para resolver problemas matemáticos definidos.

Recursividad: Es una técnica donde una función se llama a sí misma para resolver un problema, dividiéndolo en partes más pequeñas hasta llegar a un caso base que detiene el proceso[1]. Se utilizó para los siguientes algoritmos, que tienen una definición naturalmente recursiva:

- **Factorial ($n!$):** Se define como $n! = n \cdot (n - 1)!$, con el caso base $0! = 1$.
- **Sucesión de Fibonacci (F_n):** Se define por la fórmula $F_n = F_{n-1} + F_{n-2}$, con los casos base $F_0 = 0$ y $F_1 = 1$.

Iteración (Bucles): Es la repetición de un bloque de código hasta que se cumple una condición[2]. Se usó para la **Conjetura de Collatz**, ya que el número de pasos es impredecible. La conjetura aplica dos reglas simples en un bucle para reducir cualquier número entero a 1: si es par, se divide entre 2; si es impar, se calcula $3n + 1$.

3. Desarrollo

Primero implementaremos tanto lo aprendido en clase sobre tipos de datos, condicionales, operadores e implementación de menús para poder realizar la estructura básica de nuestro programa esto lo realizamos mediante el uso de switch y la librería `import java.util.Scanner;` para poder obtener entradas desde la consola. Una vez que realizamos el menu con ayuda de las herramientas mencionadas y apoyándonos del ejemplo visto en clase comenzamos a analizar cada uno de los puntos solicitados en el código y nos pusimos de acuerdo de realizar una función a cada uno de los problemas independientemente de que se necesite recursividad o no en estos.

Comenzando con el ejemplo del factorial tomamos en cuenta la formula vista en clase o en anteriores cursos para poder obtenerlo $(n)(n-1)$ y el valor de base que seria nuestro return para parar la función recursiva que en este caso era `return n=1;`.

Para el ejercicio del triangulo de pascal seguimos un proceso muy parecido ya que de igual forma vimos en clase tanto la formula como el caso base por lo que tan solo empleamos la formula $(n-1)+(n-2)$ y tomando en cuenta los casos base vistos en la clase los colocamos en la función `return n=1;` y `return n=0;` dependiendo de en que punto de la recursividad nos encontremos ya que si la condición es `(numero == 0)` siendo numero la cantidad de la serie de Fibonacci que queremos ver pues tendríamos que devolver un 0 ya que es el inicio de la sucesión y en caso de que la condición sea `(numero == 1 || numero == 2)` devolvemos en 1 ya que para ambos casos base el Fibonacci es 1.

Para Collatz aunque se explico lo básico que seria la reducción de el numero par o impar a 1 a base de divisiones realizamos una investigación en internet para entender un poco mas su funcionamiento por lo que al momento de hacer la función utilizamos un bucle para que esta operación de división se realizara correctamente además de imprimir el proceso para poder apreciar una a una las operaciones realizadas.

Finalmente nos aseguramos de llamar correctamente a las funciones dentro del menú para que funcionara bien y cerramos a la variable de Scanner que llamamos al principio

4. Resultados

Los ejercicios propuestos funcionaron correctamente, se logró sin el uso de bibliotecas externas y con base a lo aprendido en clase

Caso 1: Calculo del Factorial

•

```
PS C:\Users\edgar\OneDrive\Documentos\Practicas EDA\Practica2> java Practica12
Elige una opcion:
Elige una opcion:
1 - Calcular factorial
2 - Calcular serie de Fibonacci
3 - Calcular conjetura de Collatz
4 - Salir
1
Dame el numero para calcular su factorial:
5
El factorial es: 120
```

Se ejecuta el código y se despliega un menú para elegir que tipo de operación queremos realizar. En este caso es el calculo factorial del numero 5. Al final se muestra el resultado 120 usando la función "factorial(numero)", que está definida de forma recursiva.

Caso 2: Serie de Fibonacci

```
eDrive\Documentos\Practicas EDA\Practica2> java Practica12
Elige una opcion:
1 - Calcular factorial
2 - Calcular serie de Fibonacci
3 - Calcular conjetura de Collatz
4 - Salir
2
¿Que numero de la serie de Fibonacci quieres? :
7

0
1
1
2
3
5
8
El numero de fibonacci es 8
```

En este caso de igual forma se despliega el menú, esta vez escogemos el calculo de la serie de Fibonacci. El programa pregunta: “¿Qué número de la serie de Fibonacci quieres?”

En este caso el numero 7 fue el que escogimos para el calculo. Luego imprime toda la serie hasta ese número, usando la función recursiva "fibonacci(numero)".

-El programa considera que el primer término es 0.

-Si se introduce el numero 1, devuelve 0.

Caso 3: Conjetura de Collatz.

```
PS C:\Users\edgar\OneDrive\Documentos\Practicas EDA\Practica2> java Practica12
Elige una opcion:
1 - Calcular factorial
2 - Calcular serie de Fibonacci
3 - Calcular conjetura de Collatz
4 - Salir
3
Dame el numero sobre el que calcular la conjetura de Collatz:
6
La conjetura de collatz de este numero es:
3
10
5
16
8
4
2
1
```

Para este calculo Se pide un número entero positivo y se aplica la regla de Collatz:

-Si el número es par: lo divide entre 2.

-Si es impar: lo multiplica por 3 y le suma 1.

Este proceso se repite hasta que llega a 1, imprimiendo cada paso. En este caso el numero que introducimos fue el 6 y al final se muestra el resultado del calculo.

Caso 4: Salir.

```
Elige una opcion:
1 - Calcular factorial
2 - Calcular serie de Fibonacci
3 - Calcular conjetura de Collatz
4 - Salir
4
Saliendo...
PS C:\Users\edgar\OneDrive\Documentos\Practicas EDA\Practica2> |
```

Al final el cuarto caso es la opcion salir donde se imprime el texto "Saliendo.."
Y se finaliza el programa.

5. Conclusiones

Esta práctica nos dejó aplicar los conceptos teóricos de recursividad e iteración para resolver problemas matemáticos.

Al implementar los algoritmos como factorial, sucesión de Fibonacci y la conjetura de Collatz demostramos que, más allá de conocer las definiciones, es necesario comprender la lógica detrás de cada técnica para utilizarlas de manera adecuada dependiendo del problema.

La recursividad mostró su eficiencia en la mayoría de los problemas gracias a que se puede autorreferenciar, mientras que la iteración resultó más conveniente en procesos de comportamiento impredecible, como en Collatz.

Esta práctica resalta la importancia de relacionar la teoría con la programación aplicada, consolidando así un entendimiento más profundo sobre cómo los fundamentos computacionales permiten modelar y resolver problemas de manera sistemática.

6. Referencias bibliográficas

[1] Facultad de Estudios Superiores Cuautitlán, UNAM, Recursividad. "[En línea]. Disponible en: https://virtual.cuautitlan.unam.mx/intar/?page_id=185.

[2] Universitat Carlemany, "¿Qué es la iteración en programación?," 29 de nov. de 2023. [En línea]. Disponible en: <https://www.universitatcarlemany.com/actualidad/blog/que-es-la-iteracion/>.