

UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGIA DE RECURSOS
RENOVABLES Y NO RENOVABLES

CARRERA DE COMPUTACION

INTEGRANTES: Juan Alverca

Jostin Jimenez

Jhostin Rojas

Geovanny Romero

INGENIERO: Édison Coronel

CICLO: Segundo "A"

1859

LOJA – ECUADOR 2023

RECURSIVIDAD

¿Qué es?

La recursividad en programación es un concepto que, normalmente, cuesta entender a los principiantes en este lenguaje. Se trata de algo que puede parecer bastante complejo a primera vista, pero que, en realidad, es relativamente sencillo.

La recursividad en programación Java es la capacidad que existe en este lenguaje de crear funciones que llamen a otras funciones. De esta manera se evita el uso de bucles u otros iteradores. Se trata de una técnica de programación que permite crear instrucciones que se repitan un número "n" de veces, por eso se trata de una forma de programación que permite evitar el uso de estructuras de datos repetitivas. [1]

¿Cómo implementarla en java?

En Java, puedes implementar la recursividad mediante la llamada a una función dentro de sí misma. Aquí tienes un ejemplo básico de cómo implementar la recursividad en Java:

```
public class RecursionExample {
    public static void main(String[] args) {
        int number = 5;
        int result = factorial(number);
        System.out.println("El factorial de " + number + " es: " + result);
    }

2usages
    public static int factorial(int n) {
        // Caso base: cuando n es igual a 0, el factorial es 1
        if (n == 0) {
            return 1;
        } else {
            // Llamada recursiva: se llama a la función factorial dentro de sí misma return n * factorial(m n - 1);
        }
    }
}
"C:\Program Files\Java\jdk-17.0.2\b.
El factorial de 5 es: 120
Process finished with exit code 0
```

En este ejemplo, se muestra cómo calcular la factorial de un número utilizando recursividad.

La función factorial se llama a sí misma dentro de su cuerpo hasta que se alcanza el caso base, donde "n" es igual a 0. En ese punto, la recursión se detiene y se comienza a deshacer, devolviendo el resultado final.

Es importante tener un caso base adecuado para evitar que la recursión se ejecute indefinidamente. En este caso, el caso base es cuando n es igual a 0, donde la factorial es 1.

BIBLIOGRAFIA

[1]"¿Qué es la recursividad en la programación con Java | Tokio". Tokio School. https://www.tokioschool.com/noticias/recursividad-programacion-java/#:~:text=La%20recursividad%20en%20programación%20Java,archivos,%20con%20carpetas%20y%20subcarpetas. (accedido el 13 de junio de 2023).