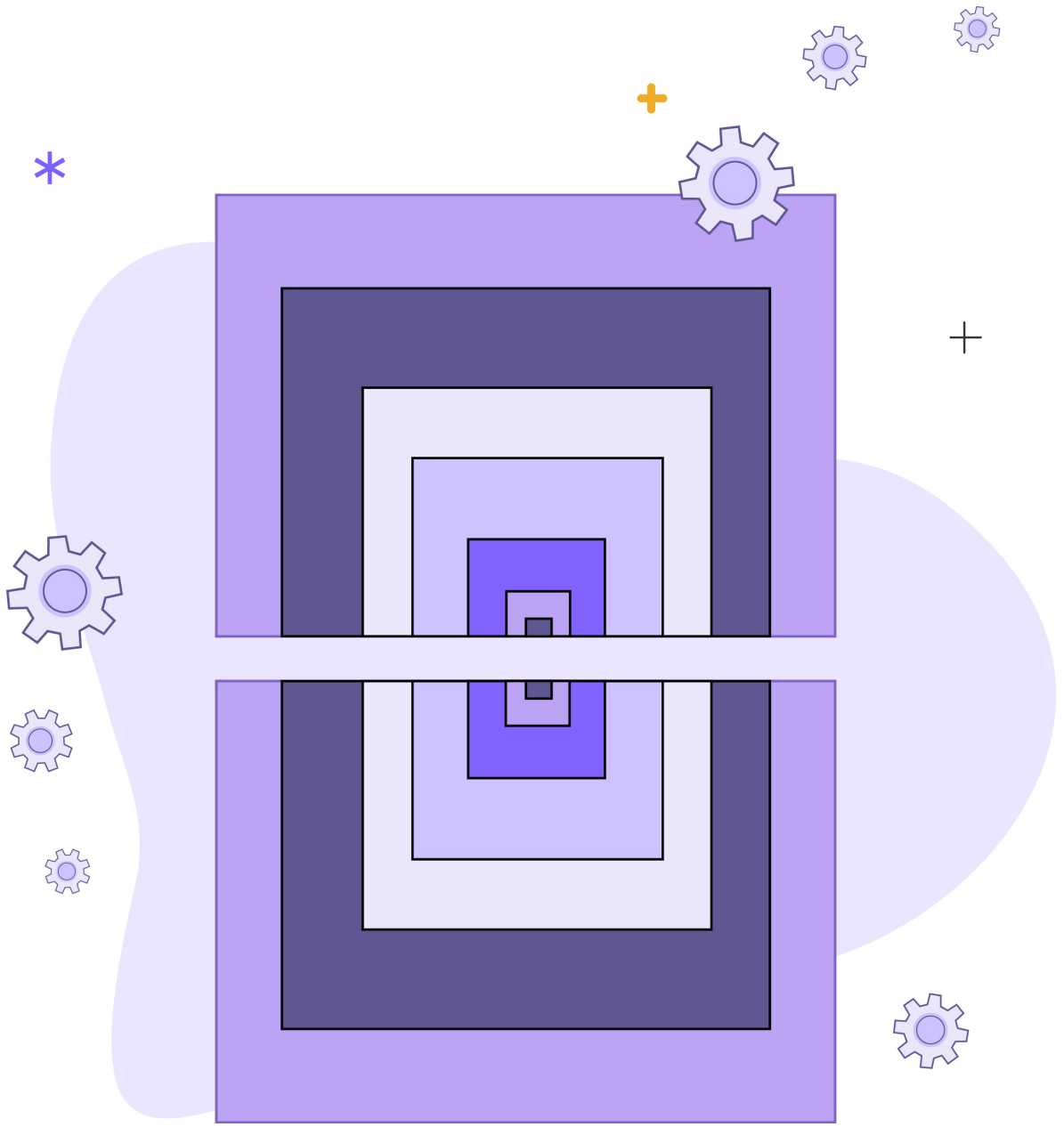
Capítulo 7

Recursión



Capítulo 7

Recursión

Objetivo

El capítulo tiene como objetivo enseñar la recursión, una técnica que nos permite resolver problemas de forma elegante y sencilla ante determinados escenarios. Aprenderemos a formular soluciones recursivas y comprender los beneficios y las limitaciones de este enfoque.

Al finalizar este capítulo, estarás capacitado para utilizar la recursividad de forma efectiva en tus programas. Habrás adquirido una comprensión profunda de los beneficios y las limi-taciones de la recursividad, así como la capacidad de identificar problemas que se pueden resolver de forma recursiva y diseñar algoritmos para su solución.

Recursión

La recursión es una técnica que se utiliza para resolver problemas del tipo evolutivo, digamos que la búsqueda de la solución es posible a través de la división del problema original en problemas cada vez más pequeños y similares al del comienzo.

Esta idea de resolución, pensada a través de la utilización de algoritmos, se materializa por medio de subproblemas, o más concretamente, en funciones que se llaman a sí misma.

Para lograr que la función resuelva el problema y además se llame a sí misma y no caiga-mos en un bucle que no podrá detenerse, se debe llamar a sí misma con los argumentos, aunque sea ligeramente, diferentes.

Veamos un ejemplo práctico de utilización. El factorial de un número natural n, denotado como n!, se define como el producto de todos los números naturales positivos menores o iguales a n.

Formalmente:

0! = 1 - por convención

1! = 1

n! = 1 * 2 * 3 * ... * (n - 1) * n

Por ejemplo el cálculo de 5! es igual a 1 * 2 * 3 * 4 * 5, lo que da como resultado 120.

De lo anterior podemos ver que:

5! = 5 * 4!

4! = 4 * 3!

3! = 3 * 2!

2! = 2 * 1!

1! = 1 ⇒

Esta situación nos lleva a observar que estamos frente a la definición de un problema recursivo, aquel problema original que puede resolverse con la división en subproblemas similares.

Entonces, siguiendo la definición de cómo se debe calcular el factorial podemos realizar el algoritmo.

Tenemos que considerar 2 escenarios posibles:

- si el valor del parámetro n es igual a 0 o es igual a 1, el resultado del factorial es 1.
- si el valor del parámetro es otro, dentro de los valores posibles, el resultado es n por el factorial de n-1.

> La recursión es un proceso, o conjunto de reglas, definido en términos de sí mismo.

A continuación vemos esta especificación en forma de código:

```
//
public static int factorial(int n) {

   //
   if (n == 0 || n == 1) {
       //
       return 1;
```