

## 12 | Recursividad - Máquina de cambio

### META

Que el alumno practique el uso de la programación recursiva para resolver un problema sencillo.

### OBJETIVOS

Al finalizar la práctica el alumno será capaz de:

1. Plantear un problema en forma recursiva.
2. Identificar el o los casos base en una solución recursiva.
3. Identificar el paso recursivo en una solución recursiva.
4. Programar funciones recursivas en Java.

### ANTECEDENTES

Una forma natural de almacenar soluciones a problemas planteados en forma recursiva es en forma de listas. En estos casos, cada llamada recursiva deberá agregar o quitar elementos a la lista que reciba como respuesta a una llamada recursiva, según el problema del que se trate. Si el problema recursivo es más complejo, es probable que necesites tener listas de listas.

### DESARROLLO

Esta práctica consiste en resolver un solo problema en forma recursiva, programar la solución en Java y ejecutar algunos ejemplos. El problema es el siguiente:

Digamos que se quiere dar  $n$  pesos de cambio a un cliente con monedas de \$10, \$5, \$2 y \$1. Calcular todas las formas posibles de dar el cambio.

En este ejercicio necesitas una lista de las posibles formas de dar cambio, donde cada forma está almacenada en un objeto, sin embargo existen otras alternativas. Otra forma de resolver este problema es tener una lista de formas de dar cambio, donde cada forma de dar el cambio es a su vez una lista de monedas. La redacción de los siguientes puntos asume que usarás un objeto para representar la forma de dar cambio.

### EJERCICIOS

1. Diseña el algoritmo para resolver este problema sin programarlo todavía. Escribe la receta.
2. Crea la clase `Cambio`, donde los atributos serán el número de monedas de cada denominación. Ej `numDiez`, `numCinco`, etc..
3. Crea la clase `MaquinaDeCambio`.
4. Agrega un método que reciba como parámetro el `precio` a pagar y devuelva una lista con todas las formas de dar el cambio. Utiliza la lista de `Object` que programaste para la práctica anterior.
5. Punto extra: si tu solución requería el uso de algún ciclo como `for`, has otra versión que no lo use, sino que sea puramente recursiva. Si tu método original ya era sólo recursivo, tienes este punto automáticamente.