

[Página Principal](#) / [Mis cursos](#) / [131_15021482_ZSIFC02_MP0485_B](#) / [7. Colecciones II. Uso de las colecciones de la librería standard.](#)

/ [Ejercicios de interface list y queue en coderunner](#)

Pregunta **6**

Sin finalizar

Puntúa como 1,00

Combinaciones de números sin recursividad

Dada una lista de números se indica la longitud de cada combinación y se generan todas las combinaciones posibles. Se pide una solución NO basada en recursividad escribiendo el método combinar() que completa el trabajo del método combinaciones().

```
//escribir método static combinar y enviar con método combinaciones
static List combinaciones(int longitud, List numeros) {
    // en principio result almacena la combinación de longitud 1
    // es decir, es la copia de la lista numeros
    // List result= numeros; MAL, así modificaré el original
    if(longitud==0){
        return new ArrayList();
    }
    List result = new ArrayList<>(numeros);
    for (int i = 0; i < longitud - 1; i++) { // longitud-1 porque ya inicializamos con 1 combinacion
        result = combinar(result, numeros);
    }
    return result;
}
```

OJO: Se envía el método combinaciones del enunciado junta al método combinar, ambos static

Por ejemplo:

Test	Resultado
<pre>List<String> numeros = Arrays.asList("0", "1", "2"); System.out.println(combinaciones(3, numeros));</pre>	<pre>[000, 001, 002, 010, 011, 012, 020, 021, 022, 100, 101, 102, 110, 111, 112, 120, 121, 122, 200, 201, 202, 210, 211, 212, 220, 221, 222]</pre>

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

◀ 04D. Backtracking

Ir a...

05. Interface Set Y Collection ▶