



# Java Avançado

## Exercícios Propostos

Expressões Lambda

## 1 Exercício

Crie uma aplicação que ordena de forma decrescente uma lista de números compreendidos entre 0 e 9. Para fazer esta ordenação, utilize uma instância de um `Comparator` criado a partir de uma expressão lambda.

## 2 Exercício

Crie um método `transform()` que recebe como parâmetros um `List<Integer>` e um `Function<Integer, Integer>`. Este método deve aplicar o objeto `Function` em cada um dos elementos da lista e retornar uma nova lista, resultante desta aplicação.

Para testar o funcionamento do método `transform()`, crie uma lista com alguns números e faça chamadas ao método usando dois objetos `Function`: um deles deve multiplicar os elementos da lista por 2 e o outro deve inverter o sinal dos números.

## 3 Exercício

Crie um método `transform()` semelhante ao que foi feito no Exercício 2, mas dessa vez este método deve receber um objeto `List` com um tipo parametrizado e um objeto `Function` com este mesmo tipo parametrizado. Por exemplo, se o usuário decidir que a lista contém objetos `String`, o objeto `Function` deve ser capaz de transformar uma `String` em outra.

Para testar o funcionamento do método `transform()`, crie uma lista com algumas strings e faça uma chamada ao método para que ele retorne uma nova lista com os mesmos elementos, mas todos eles compostos apenas por letras maiúsculas. Crie também uma lista de números decimais e faça uma chamada ao `transform()` para que ele retorne uma nova lista onde os elementos da lista inicial sejam arredondados para baixo.

**Dica:** Para fazer o arredondamento de um número decimal para baixo é possível utilizar o método `Math.floor()`.