# Imutabilidade o que é?

Conceito na programação de um objeto ou valor que não pode ser alterado.

Vamos imaginar uma ficha cadastral

Nome, cpf, peso, altura.

Imaginamos que nome e cpf tem que ser imutalvel, porque não altera.

Peso e altura a pessoa engordam, emagrece, cresce...

Além disso, a imutabilidade promove a concorrência segura em ambientes multithread, pois cada thread pode trabalhar com cópias dos dados imutáveis sem interferir nas outras. A imutabilidade também torna o código mais legível e facilita a manutenção, pois o fluxo de informações é mais claro e previsível.

Ao utilizar streams, temos como uma das operações finais a possibilidade de colocar os dados gerados em outra lista. Vimos também que se utilizarmos diretamente o toList()para esse fim, é gerada uma lista imutável. Sendo assim, quando precisamos que a nova lista possa ser modificada, temos que utilizar o collect(Collectors.toList()).

* String: A classe java.lang.String é imutável. Uma vez criada, uma instância de String não pode ser modificada.
* Classes Wrapper: As classes que representam os tipos primitivos, como Integer, Double, Character, etc., também são imutáveis.
* Enum: As enumerações (enum) em Java são imutáveis. As instâncias de um enum são criadas apenas uma vez e não podem ser modificadas.

Além das classes nativas, a API de coleções do Java também oferece algumas classes imutáveis na classe utilitária Collections. Essas classes são usadas para criar coleções imutáveis, como listas, conjuntos e mapas. Alguns exemplos incluem:

* Collections.unmodifiableList: Cria uma lista imutável a partir de uma lista existente.
* Collections.unmodifiableSet: Cria um conjunto imutável a partir de um conjunto existente.
* Collections.unmodifiableMap: Cria um mapa imutável a partir de um mapa existente.

Essas classes permitem que você crie versões imutáveis de coleções existentes, tornando-as seguras para compartilhar e evitar a modificação acidental.

**public** **class** **ImutabilidadeExemplo** {

**public** **static** void **main**(String[] args) {

**List**<**String**> listaOriginal = **new** **ArrayList**<>();

listaOriginal.**add**("A");

listaOriginal.**add**("B");

**List**<**String**> listaImutavel = **Collections**.**unmodifiableList**(listaOriginal);

// Tentativa de adicionar um elemento na lista imutável resultará em exceção

listaImutavel.**add**("C"); // Lançará UnsupportedOperationException

}

}