

Para saber mais: Tipo genéricos

Durante o curso vimos que a utilização deles serve para parametrizar classes ou interfaces que usamos com frequência e que com eles podemos deixar o código mais flexível e reutilizável! Inclusive, um caso famoso é o da interface `List<E>` e de toda a API de coleções do Java. Também vimos que com os tipos genéricos, podemos criar regras com os famosos **wildcards**, dentro os quais vimos os **Unbounded Wildcards** quando estamos trabalhando com tipos desconhecidos, como por exemplo `Class<?>`. Também vimos os **Upper Bounded Wildcards**, como fizemos no nosso container IoC ao restringir os tipos passados para o método registra fazendo `<K extends T>` indicando que o tipo `K` deveria ser um subtipo de `T`.

No entanto, tipos genéricos é um tema vasto! Por isso, deixo aqui para vocês, meus alunos, [esse link da documentação do Java](https://docs.oracle.com/javase/tutorial/java/generics/index.html) (<https://docs.oracle.com/javase/tutorial/java/generics/index.html>), com um tutorial da própria Oracle (em inglês) que irá se aprofundar e muito nesse assunto! Além disso, [esse outro link](https://docs.oracle.com/javase/tutorial/extra/generics/index.html) (<https://docs.oracle.com/javase/tutorial/extra/generics/index.html>) também da Oracle (em inglês) irá trazer conteúdo extra sobre o mesmo tema!

Espero que aproveitem esse conteúdo para aprofundar ainda mais o conhecimento de vocês!