

## Lambda

Lambda é uma sintaxe especial que te permite simplificar o código de implementação de interfaces funcionais, já que elas têm um único método. Funciona basicamente escrevendo o comportamento que esse método único vai ter, dessa maneira podemos “esconder” todo o resto da implementação reduzindo na maioria das vezes o código inteiro pra uma linha.

Uma maneira de percorrer listas em java é utilizando o **for**, seja ele o **controlado por variáveis de interação** ou o **enhanced for (um pouco mais enxuto)**:

```
for(int i = 0; i < usuarios.size(); i++){  
    System.out.println(usuarios.get(i).getNome());  
}  
  
for (Usuario user: usuarios) {  
    System.out.println(user.getNome());  
}
```

No java 8 toda coleção ganhou um método chamado de **forEach()**, método esse que tem como parâmetro um **Consumer**, esse **consumer** é um tipo de objeto que entrou com o pacote de functions do java 8:

```
List<Usuario> usuarios = Arrays.asList(user1, user2, user3);  
  
usuarios.forEach();
```

Esse **Consumer** tem um único método chamado **accept()**, dá pra ler esse objeto como **Consumer<O que precisa ser consumido>**, aceite (**o que precisa ser consumido**) e faça alguma tarefa (a gente escolhe):

```
usuarios.forEach(new Consumer<Usuario>() {  
    @Override  
    public void accept(Usuario usuario) {  
        System.out.println(usuario.getNome());  
    }  
});
```

Em java, uma **Lambda** nada mais é que uma maneira mais simples de implementar uma Interface que **TEM UM ÚNICO MÉTODO** (nesse caso a **Consumer**).

Então podemos simplesmente **escrever o comportamento da interface** e já atribuir esse **comportamento** como o valor dela.

É como se o Java já entendesse que esse **comportamento** só pode entrar naquele único método que a **interface** tem (nesse caso o **accept**), então ele consegue inferir o **comportamento** na **interface** (também consegue inferir o tipo, nesse caso **Usuario**):

```
Consumer<Usuario> mostrar = usuario -> System.out.println(usuario.getNome());  
  
usuarios.forEach(mostrar);
```

Agora, se um **Consumer** (interface com um método único) consegue fazer toda essa inferência através do **lambda**, o método **forEach** não poderia entender a mesma inferência já que ele recebe a própria **interface(Consumer)**? SIM:

```
usuarios.forEach(usuario -> System.out.println(usuario.getNome()));
```