forEach percorre geralmente uma lista, sem precisar fazer o FOR + índice.

temporadas.forEach(t -> t.episodios().forEach(e -> System.*out*.println(e.titulo())));  
// vou pegar temporada  
 // Vou chamar temporada de T  
 // Vou dizer que quero entrar dentro de temporada.episodio  
 // vou fazer outro forEach em episodios.  
 // vou chamar episodios de E

**Esse código está fazendo o seguinte, esta acessando a lista de temporadas e esta retornando apenas o nome dos episódios.**

Temporadas é uma lista que contêm as temporadas da serie

Observa a facilidade e vou deixar o código antigo em baixo.

Olhe como estava o código antes disso.

for (int i = 1; i < dados.totalTemporadas() ; i++) { // Fazendo um for para percorrer a lista  
 List<DadosEpisodios> episodiosTemporada = temporadas.get(i).episodios(); // Pega a temporada e episodios e faz outra lista.  
 for(int j = 0; j < episodiosTemporada.size(); j++) { // Faz outro for contando a quantidade de temporadas.  
 System.*out*.println(episodiosTemporada.get(j).titulo()); // Imprime a lista com os epsodios.  
 }  
}

Outra maneira de converter e deixar mais eficiente

temporadas.forEach(t -> System.*out*.println(t));  
temporadas.forEach(System.*out*::println);

Observa que na primeira linha precisamos passar o T -> .....

Quando estamos trabalhando somente com um parâmetro, no caso estamos trabalhando somente com a temporada e nada mais, podemos usar a segunda linha.

Segue outro exemplo da função Lambda

List<Integer> lista = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9);

lista.stream().filter(i -> i % 2 == 0).forEach(System.out::println);

i = Os números que esta dentro da lista

Aqui vamos pegar os números I dividir por 2 e retornar os números que tem o resto == 0

E depois vamos usar o forEach para dar o printa em todos os elementos que atingiram essa condição