**1 - Streams Infinitos**

Streams infinitos, ou “infinite Streams”, são streams que não têm um tamanho definido. Eles são úteis quando queremos gerar uma sequência de números ou valores. Aqui está um exemplo de como criamos um Stream infinito com o método iterate:

Stream.iterate(0, n -> n + 1)

.limit(10)

.forEach(System.out::println);

No exemplo acima, começamos com o número 0 e adicionamos 1 a cada iteração para gerar uma sequência numérica. Utilizamos o método limit() para restringir o Stream infinito a 10 elementos e usamos forEach para imprimi-los.

Lembrar que: o Stream pega dados de uma lista para fazer uma nova coleção, então não podemos usar somente uma variável.

ist<String> numeros = new ArrayList<>();  
  
 numeros.add("1");  
 numeros.add("2");  
  
 numeros.forEach(System.*out*::println);  
  
 numeros.stream()  
 .filter(n -> n.contains("2"))  
 .forEach(n -> System.*out*.println("Numero localizado: " + n));  
  
  
 // Lembrar que se eu quiser usar o forEach eu nao posso colocar em uma lista

// Aqui estamos somente fazendo a iteração e imprimindo  
 Stream.*iterate*(0, n -> n + 1)  
 .limit(10)  
 .forEach(System.*out*::println);  
  
 // Agora se eu nao quiser e somente guarda os dados eu posso fazer assim

// Aqui estamos passando toda a nossa coleção para uma nova lista, por isso estamos usando o collect para coletar os dados e passar para uma nova lista.  
  
 List<Integer> numero = Stream.*iterate*(0, n -> n +1)  
 .limit(10)  
 .collect(Collectors.*toList*());  
  
 numero.forEach(System.*out*::println);  
  
}