Como trabalhar com Streams

Operações intermediárias: são as operações que retornam a Stream modificada (filtrada, ordenada, reorganizada)

- distinct(): Retorna a Stream eliminando elementos repetidos
- filter(Predicate filtro): Retorna a Stream com os elementos que cumprem a condição do filtro
- limit(long N): Retorna a Stream com os primeiros N elementos
- map(Lambda mapper): Retorna a Stream com a versão mapeada de cada elemento
- peek(Consumer ação): Retorna a Stream após executar a ação que é passada como parâmetro (existe por motivos de debugging, para poder ver como os elementos estão dispostos no Stream num determinado momento)
- skip(long N): Remove os primeiros N elementos e retorna o restante
- sorted(Comparator comparador):Retorna o Stream classificado pelo Comparador que é passado como um parâmetro
- unordered (): Retorna a Stream potencialmente fora de ordem (interessa mais para execução paralela, potencialmente mais eficiente)

Tokio.

Como trabalhar com Streams

Operações finais: são as operações que retornam como resultado algo diferente do próprio Stream

- boolean allMatch(Predicate predicado): Compara se todos os elementos cumprem o predicado
- boolean anyMatch(Predicate predicado): Compara se algum elemento da Stream cumpre o predicado
- boolean noneMatch(Predicate predicado): Compara se nenhum elemento da Stream cumpre o predicado
- Optional findAny(): Devolve um elemento qualquer da Stream
- Optional findFirst(): Devolve o primeiro elemento da Stream
- R collect(Collector colector): Devolve a Stream como um tipo de coleção
- long count(): Devolve o número de elementos da Stream
- void forEach(Consumer ação): Executa uma operação sobre cada elemento da Stream
- void forEachOrdered(Consumer ação): Executa uma operação sobre cada elemento da Stream ordenado
- Optional max(Comparator comparador): Devolve o maior dos elementos segundo o comparador
- Optional min(Comparator comparador): Devolve o menor dos elementos segundo o Comparador
- Object[] toArray(): Devolve a Stream com um array de Objects
- Iterator iterator(): Devolve a Stream como um Iterator