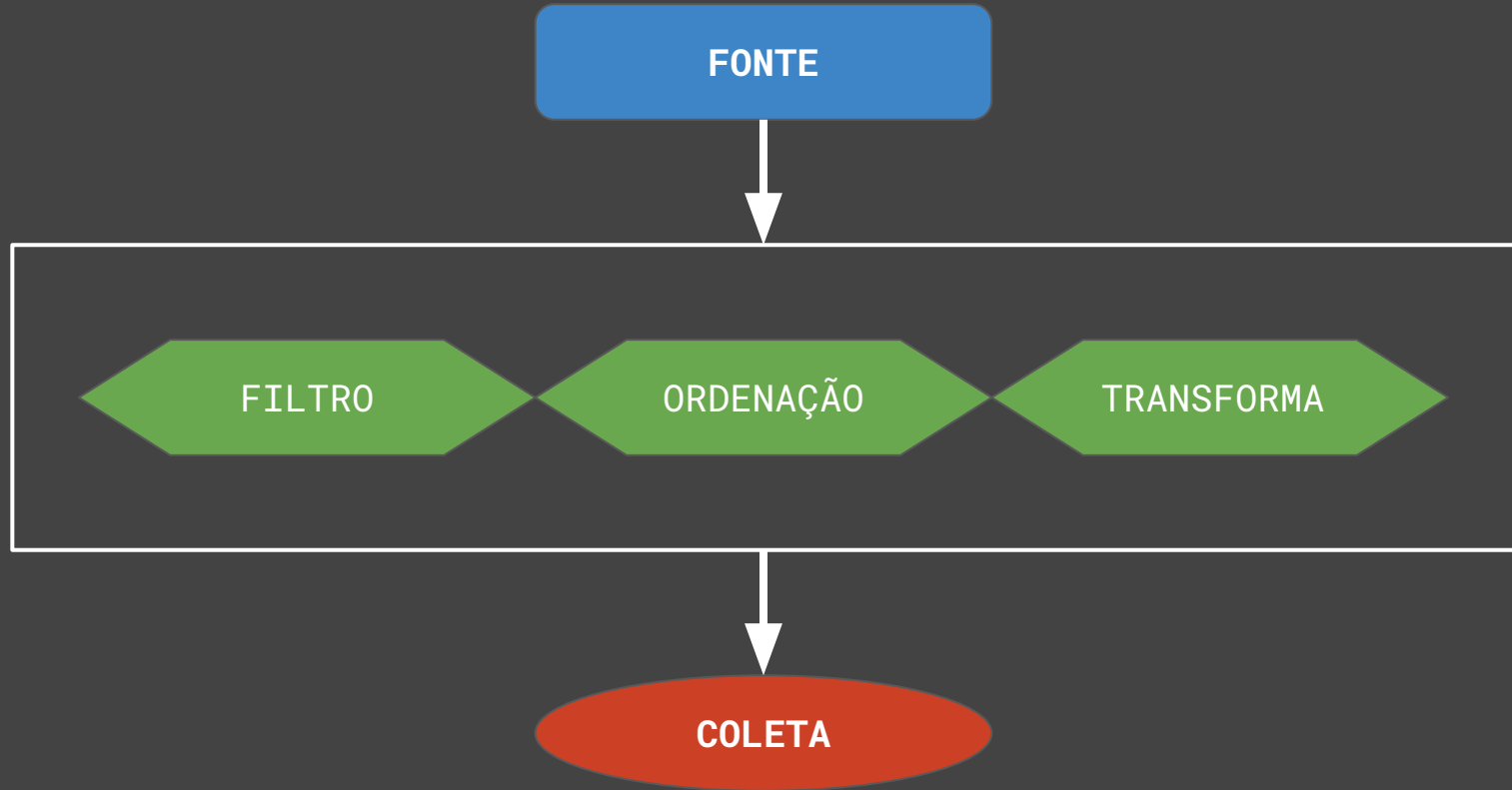


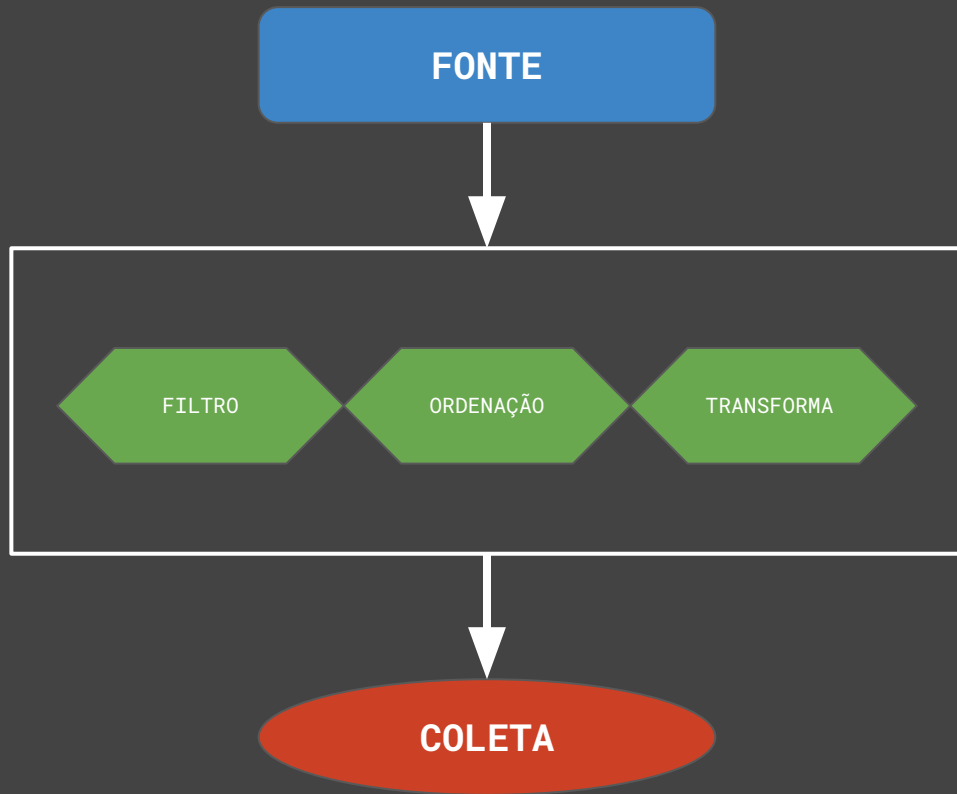
Streams

Vantagens do Streams:

- Fornece mais eficiência no desenvolvimento em Java.
- Com paralelismo fica muito fácil utilizar multi-threads.
- Utiliza muito Lambdas.

O fluxo do stream consiste em uma fonte, que pode sofrer nenhuma ou muitas operações intermediárias, e tem uma operação terminal.

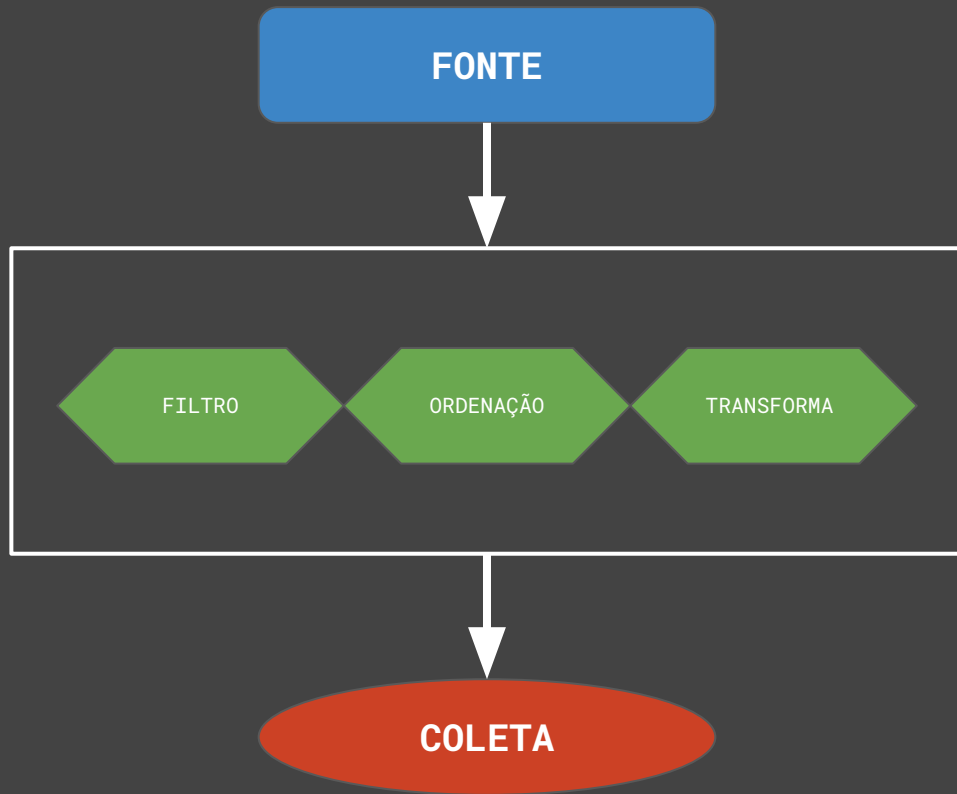




Source

Podem ser:

- Collections como lists e sets.
- Arrays como ints, longs, doubles.
- Linhas de arquivos.

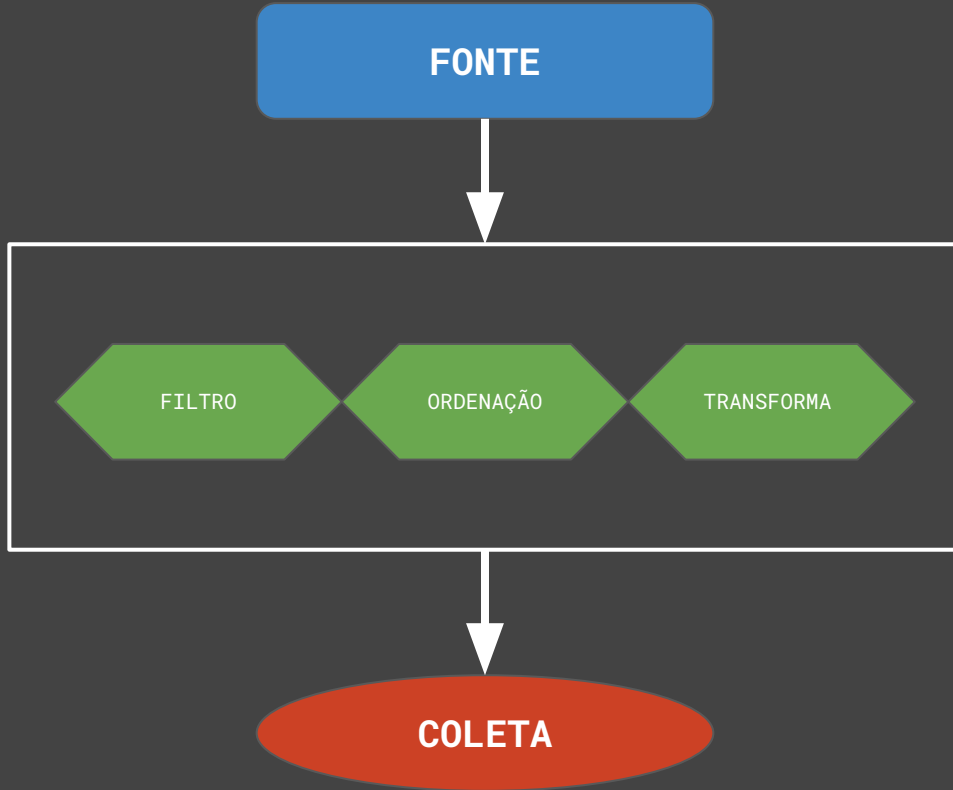


Operações Intermediárias

Como:

- Filters
- Maps
- Sort

Retorna um stream então suporta múltiplas operações intermediárias.



Operações Terminais

Como:

- For Each
- Collect
- Reduce

Retorna void ou um não stream.

Operações Intermediárias

- `filter(Predicate)` realiza a filtragem dos elementos através da validação.
- `map(Function)` aplica a função em cada elemento, pode provocar a transformação.
- `flatMap(Function)` semelhante a `map`, porém usada em valores "envelopados" em coleções ou `Optional`'s.
- `distinct()` elimina elementos repetidos, garante ocorrências únicas.
- `sorted()` ordena os elementos, com a variação `sort(Comparator)` que permite personalizar o critério de ordenação.
- `peek(Consumer)` método voltado para o debug.
- `limit(long)` limita a quantidade máxima de elementos.
- `skip(long)` salta uma quantidade de elementos.

Operações Terminais

- `allMatch/anyMatch/noneMatch(Predicate)` verifica se todos/algum/nenhum atendem a condição de validação do predicado.
- `count()` conta a quantidade total de elementos.
- `min()` obtém e fornece um `Optional` do menor elemento.
- `max()` obtém e fornece um `Optional` do maior elemento.
- `reduce(BinaryOperator)` realiza a operação de redução para um único elemento a partir do processamento de todos elementos em pares.
- `forEach(Consumer)` faz uma determinada operação em cada um dos elementos.
- `collect(Collector)` coleta todos os elementos em uma nova estrutura final, gerando por exemplo uma `Collection`.
- `findFirst/findAny()` coleta o primeiro / ou qualquer elemento.
- `sum()` soma todos os elementos.
- `average()` a média dos elementos.