

Ficha 2

Objectivos: Prática com `Stream<T>`, `Splitterator<T>` e o idioma *internal iteration*.

Data limite de entrega: 7de Maio de 2017

NOTA: Implementar **os testes unitários necessários** para validar o funcionamento das funcionalidades pedidas.

Parte 1

No âmbito da biblioteca `movlazy` pretende-se substituir a utilização de `Iterator<T>` por `Stream<T>` e de `Iterable<T>` por `Supplier<Stream<T>>`. Deverá alterar a implementação de `movlazy` de acordo com os seguintes requisitos:

- Remover todas as utilizações de `Queries`.
- Remover todas as utilizações de `Iterable<T>` e `Iterator<T>`.
- Campos do tipo `Iterable<T>` devem passar a ser do tipo `Supplier<Stream<T>>`.
- Propriedades (*getters*) ou métodos que retornam `Iterable<T>` passam a retornar `Supplier<Stream<T>>`.
- Em `MovService` deve substituir a cache `Map<Integer, List<CastItem>>` cast por `Map<Integer, Supplier<Stream<CastItem>>>`.

Para fazer cache de `Supplier<Stream<CastItem>>>` pode usar algumas das abordagens propostas em: <https://stackoverflow.com/q/49877005/1140754>

O projecto `movapi` disponibilizado em <https://github.com/isel-leic-mpd/movapi> foi actualizado com as alterações acima descritas e podem ser consultadas no seguinte [commit](#).

Parte 2

O nome da entidade `CastItem` deve ser corrigido para `Credit` e o nome da entidade de domínio `Actor` deve ser corrigido para `Person`. Ou seja `Credit` e `Person` deixam de representar apenas personagens (*"character"*) mas passam também a incluir realizadores, argumentistas, etc.

Deste modo, a classe `Credit` além do campo `character` usado para identificar uma personagem passa a ter mais dois campos: `department` e `job`, usados para identificarem outras funções como realizador, produtor, etc.

Para tal, o método de `MovWebApi` que obtém os créditos de um filme deve passar a retornar também a equipa (*"crew"*) que trabalhou nesse filme.

Os dois arrays obtidos de `MovWebApi`: `cast` e `crew`, devem ser juntos por `MovService` numa única sequência de `Credit` (i.e. `Supplier<Stream<Credit>>>`). Repare que algumas pessoas aparecem simultaneamente nos dois arrays (`cast` e `crew`) devendo ser "fundidas" num mesmo objecto `Credit`. Por exemplo, no filme *Good Will Hunting* (O Bom Rebelde) `Matt Damon` consta como actor (`cast`) e como argumentista (`crew` com `job`: *"Writer"*), devendo constar uma única vez na sequência numa instância de `Credit` com: `"id": 1892, "character": "Will Hunting", "department": "Writing", "job": "Writer", "name": "Matt Damon"`.

Assuma que a mesma pessoa não consta mais que uma vez em `cast` e `crew`.

Implemente um, ou mais, métodos genéricos semelhantes aos realizados nas aulas para `Stream` (e.g. `collapse()` ou `oddLines()` ou outro baseado em `Splitterator<T>`) que ajudem a construir a *query* que realiza a junção de duas sequências conforme os requisitos descritos.