

# Prova de Conceito #1

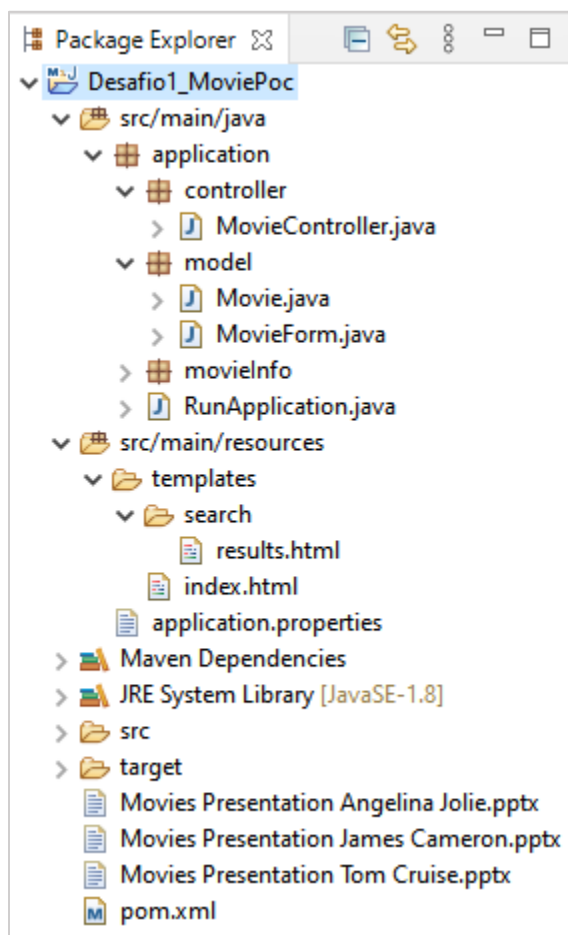
## Descrição do desafio

Criar uma aplicação onde seja possível pesquisar pelo nome de um realizador ou de um ator. Essa pesquisa gera, a partir de várias fontes que disponibilizam informação sobre filmes, a obtenção de uma lista com dados de filmes onde o realizador/ator pesquisado participa.

A aplicação deve comparar a informação das diferentes fontes e fundi-la numa lista única, com os campos “Nome do filme”, “Realizador”, “Protagonistas”, “Ano de Lançamento”.

A aplicação deve também permitir, num só click/pedido, passar a lista para uma apresentação PowerPoint onde cada slide apresenta o “nome do filme”, o “realizador” e o “ano de lançamento”.

## Estrutura do projeto



## Ferramentas utilizadas

- Eclipse IDE 2020-09
- Java 8
- Maven
- Framework Spring com Thymeleaf
- Apache POI para geração do documento Powerpoint
- JSON
- Apache Commons Text, para manipulação de Strings

## Descrição da estrutura do projeto

### Páginas HTML

- **Index.html** – página inicial da aplicação, onde são apresentados os campos de pesquisa referentes ao ator e realizador.
- **Search.html** – página que apresenta a lista de filmes obtida a partir de várias fontes, onde o ator/realizador pesquisado participa. Também é apresentado um botão para gerar o Powerpoint e outro botão que permite voltar à página inicial.

### Classes Java

- **Movie.java** – define o objeto Movie, que é representado pelo nome, uma lista de protagonistas, uma lista de realizadores e o ano de lançamento
- **MovieForm.java** – define os campos de pesquisa ator e realizador apresentados na página inicial
- **MovieInfoCompilation.java** – classe que, dado o nome de um realizador/ator, obtém listas de filmes a partir de várias fontes de informação e compila-as numa lista única.
- **MovieController.java** – controlador que faz ligação com as páginas HTML e processa os dados apresentados e submetidos em cada página. Também trata da criação do documento Powerpoint.
- **RunApplication.java** – classe responsável pela execução da aplicação.

### Ficheiro de Dependências (pom.xml)

- Inclui as dependências Spring, Thymeleaf, Apache POI, Maven, JSON e Apache Commons Text

### Ficheiro de propriedades (application.properties)

- Define o nível de detalhe apresentado nos logs da Consola

### Ficheiro Powerpoint (\*.pptx)

- Documento gerado automaticamente a partir de um clique num botão apresentado na página de resultados. Este documento expõe, no primeiro slide, informação do ator/realizador pesquisado e cada slide seguinte apresenta detalhes sobre cada filme da lista única.

## Solução implementada

Para implementar esta Prova de Conceito, foi desenvolvida uma aplicação onde a página inicial apresenta dois campos de input, um para ser inserido o nome de um ator e outro o nome de um realizador.

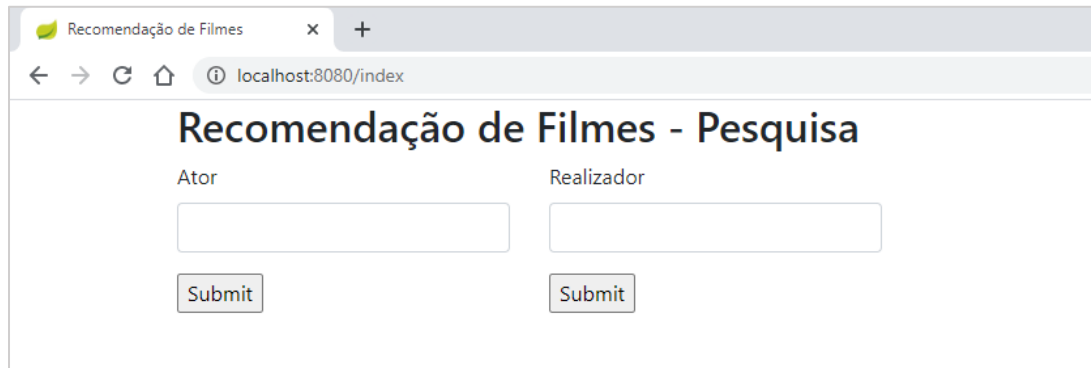


Figura 1 - página inicial

Após submissão de um dos inputs, a aplicação é reencaminhada para a página de resultados onde é apresentada uma lista única de filmes, obtida a partir da compilação de algumas fontes de informação de filmes. Essa compilação é feita na classe *MovieInfoCompilation.java*.

A obtenção de informação de filmes foi feita a partir de 2 APIs de filmes (TMDB e OMDB) e uma lista de filmes mock, definida manualmente. Para aceder à API de cada fonte, foi necessário obter KEYS nos sites respetivos.

No método `List<Movie> tmdbMovieList (String name)` é retornada uma lista de filmes que é obtida através da API da TMDB. A pesquisa é feita pelo nome de um ator ou realizador, obtendo, numa resposta JSON, uma lista de trabalhos onde ele tenha participado. Dos vários pedidos que fiz à API, verifiquei que os resultados obtidos através desta API não apresentam informações sobre os atores que protagonizam o filme nem os realizadores que participam nele, por isso os filmes da lista retornada neste método apenas têm definidos o nome do filme e o ano de lançamento. Também reparei que é retornada uma pequena amostra de filmes por ator/realizador pesquisado.

Em relação ao método `List<Movie> mockMovieList(String name)`, foi definida manualmente uma lista de filmes mock para cruzar com as listas obtidas a partir de fontes reais de informação de filmes. Nesta lista, foram adicionados alguns nomes de filmes da Angelina Jolie e do Tom Cruise, mantendo os outros campos dos Objeto do filme por preencher.

Sobre o método `List<Movie> omdbMovieList (List<Movie> movieList)`, a API da OMDB não permite pesquisar diretamente através de um ator ou realizador, por isso a pesquisa foi feita através do nome do filme. A lista de filmes retornada é obtida a partir do conjunto de nomes dos filmes das duas listas referidas anteriormente. Na resposta JSON dos pedidos feitos a esta API, conseguimos obter todas as informações que precisamos sobre um determinado filme, ou seja, o nome, lista de protagonistas, lista de realizadores e ano de lançamento.

Depois de obtidas as listas de filmes a partir das várias fontes, é feita a compilação numa lista única no método `List<Movie> movieListCompilation(List<Movie> moviesList)`. Aqui, à medida que vamos adicionando um filme novo à lista que queremos retornar, é feita uma verificação para validar se o nome do filme já existe na lista. Caso exista, então as listas de protagonistas e realizadores no novo filme a adicionar são comparadas às do filme que já existe na lista e são concatenadas, sem Strings duplicadas. Caso exista e o ano de lançamento ainda não tenha sido definido, esta campo é definido pela informação que vem no filme a ser comparado. Se o nome do filme não existe na lista, então o filme é adicionado à lista final.

Após a compilação dos filmes numa lista única, esta lista é apresentada numa tabela na página de resultados:



Nome do Filme	Realizadores	Protagonistas	Ano de Lançamento
Maleficent	Robert Stromberg	Angelina Jolie, Sharlto Copley, Elle Fanning, Lesley Manville	2014
Kung Fu Panda	John Stevenson, Mark Osborne	Angelina Jolie, Dustin Hoffman, Jack Black, Ian McShane	2008
Mr. & Mrs. Smith	Doug Liman	Brad Pitt, Angelina Jolie, Vince Vaughn, Adam Brody	2005
By The Sea	Angelina Jolie	Brad Pitt, Angelina Jolie, Mélanie Laurent, Melvil Poupaud	2015
Lara Croft: Tomb Raider	Simon West	Angelina Jolie, Jon Voight, Noah Taylor, Iain Glen	2001

Criar Powerpoint Voltar

Figura 2 - página de resultados

Nesta situação, as queries chamadas foram:

[http://api.themoviedb.org/3/search/person?query=angelina+jolie&api\\_key=b8837d02522fd74d2a462226eb912eaf](http://api.themoviedb.org/3/search/person?query=angelina+jolie&api_key=b8837d02522fd74d2a462226eb912eaf)  
<http://www.omdbapi.com/?t=maleficent&apikey=d14484e4>  
<http://www.omdbapi.com/?t=by+the+sea&apikey=d14484e4>  
<http://www.omdbapi.com/?t=lara+croft:+tomb+raider&apikey=d14484e4>  
<http://www.omdbapi.com/?t=kung+fu+panda&apikey=d14484e4>  
<http://www.omdbapi.com/?t=mr.+&+mrs.+smith&apikey=d14484e4>

O utilizador pode também gerar um Powerpoint com estas informações, clicando no botão para o efeito, apresentado na página de resultados.

O conteúdo do documento Powerpoint expõe, no primeiro slide, informação do ator/realizador pesquisado e cada slide seguinte apresenta detalhes sobre cada filme da lista única:

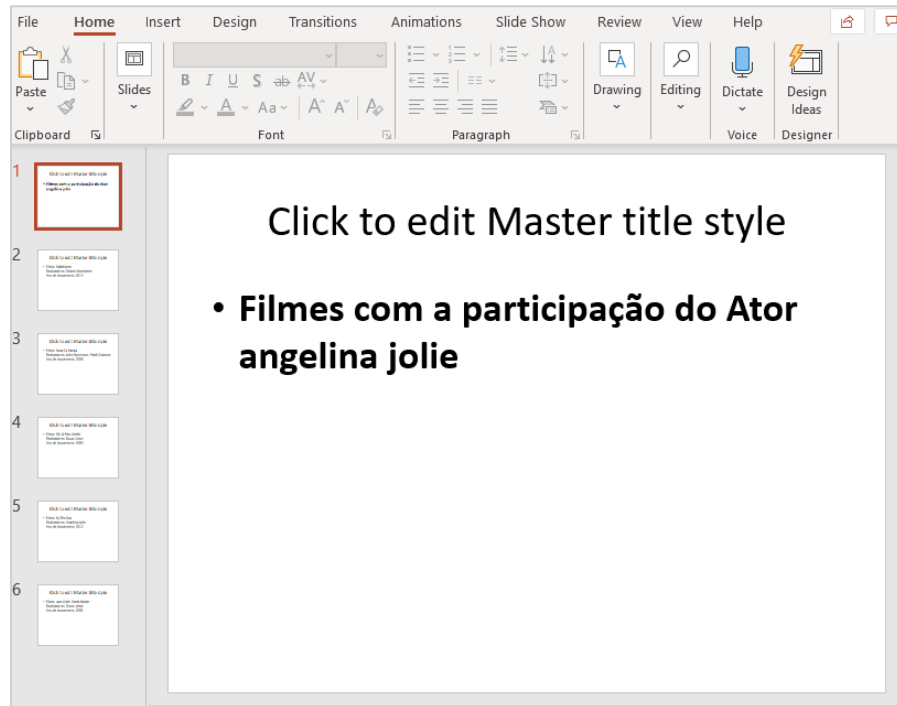


Figura 3 - primeiro slide do documento Powerpoint

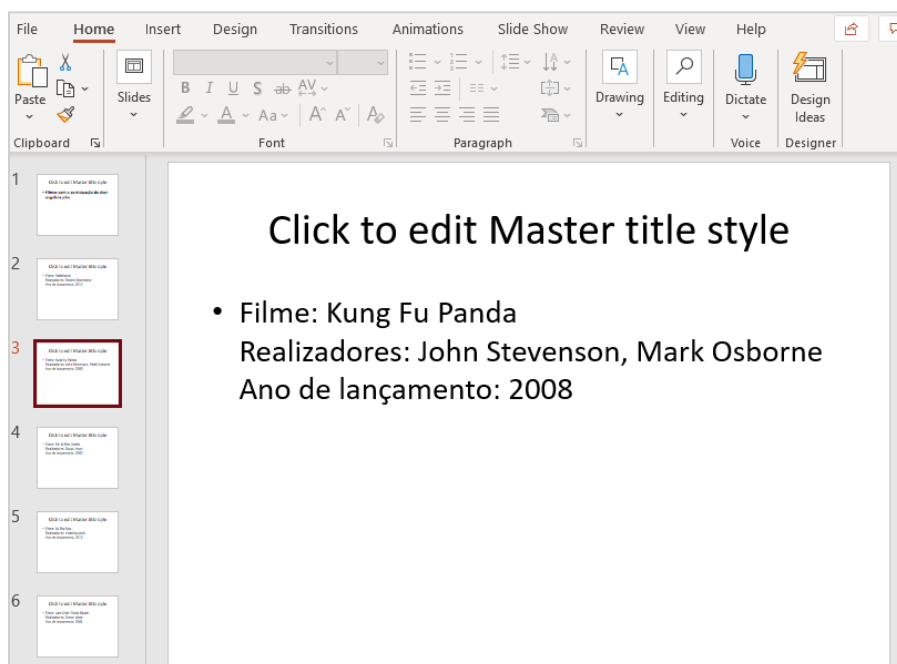


Figura 4 - terceiro slide do documento Powerpoint

## Conclusão

Na descrição desta Prova de Conceito, foi pedido para obter uma lista única de filmes, resultante da compilação de várias fontes de informação de filmes, e passar a lista para uma apresentação Powerpoint. A implementação desta aplicação passou por apresentar uma página inicial para inserção do ator/realizador a pesquisar, sendo de seguida apresentada uma lista única com informações sobre os filmes onde ele participou. Também é possível passar essa lista para um ficheiro Powerpoint.

Como pontos de melhoria, podiam ser mostradas, na página de resultados, as listas de filmes obtidas a partir de cada fonte de informação. Também seria interessante definir um layout mais composto no documento Powerpoint, de forma a organizar melhor a informação de cada slide e tornar o documento mais cativante visualmente.