

MovieRunner

User Stories

Introdução

Este documento tem como objetivo descrever as User Stories dos principais componentes que comporão o futuro sistema MovieRunner, que necessariamente deverá conter um protótipo, uma página específica para perguntar ao usuário seus gêneros preferidos de filmes e possíveis filmes que gosta(estilo REDDIT), de acordo com um DATASET de filmes lido pela LLM. Após isso o sistema deve realizar uma recomendação através de uma espécie de aprendizado de máquina de filmes com base no que o usuário escolher nesta predefinição.

Escopo

- Descrição geral - Cores e Contrastes(Provisória ou não)
- Descrição do Frontend - Por Gêneros
- Descrição do Frontend - Por Filmes
- Descrição do Backend - Por Gêneros
- Descrição do Backend - Por Filmes

Descrição geral - Tela única (Provisória ou não)

O sistema deve possuir uma interface inicial, apresentando diretamente a tela (única, talvez) que o sistema conterà.

O sistema deve possuir uma organização de cores mais voltada ao azul com com contraste no laranja e cinza, assim como descrito abaixo:

Primária - #2696D2 (Azul vibrante)

- Usada para botões principais, destaques, e áreas de foco.

Secundária - #1C6B9E (Azul Escuro)

- Para títulos, cabeçalhos ou para variações de tons no mesmo tema.

Acento - #F27A54 (Laranja Coral)

- A cor laranja complementa o azul, criando um contraste agradável para elementos de aviso, notificação, ou botões de ação secundária.

Neutro Claro - #F5F5F5 (Cinza Muito Claro)

- Usado como fundo para áreas de conteúdo para garantir que o azul e o laranja se destaquem.

Neutro Escuro - #2B2B2B (Cinza Escuro)

- Para texto principal e ícones, cria um bom contraste com o fundo claro sem ser tão pesado quanto o preto.

Apoio - #D2E1F2 (Azul Muito Claro)

- Uma variação leve e suave do azul primário, excelente para fundos de cards, caixas de entrada de dados, ou separadores.

Resumo:

Botões primários e destaques: #2696D2

Botões secundários e avisos: #F27A54

Fundo principal: #F5F5F5

Texto e ícones principais: #2B2B2B

Cards ou áreas de conteúdo: #D2E1F2

A imagem abaixo gerada pelo ChatGPT consegue ilustrar de maneira interessante essa montagem da tela:



Descrição do Frontend - Por gêneros

O sistema deve apresentar uma espécie de checkbox de maneira que ocupe o centro da tela para que caiba algumas opções que serão descritas mais abaixo.

O sistema deve apresentar um título chamativo, como: “Gêneros Disponíveis”, e ao lado ou em outra posição que fique coesa, a logo do sistema.

O sistema deve apresentar abaixo do título, um subtítulo, como: “Selecione seus gêneros de filmes favoritos”

O sistema deve começar a apresentar abaixo do subtítulo, as opções de gêneros de filmes, organizadas de maneira 5x5 ou 6x6, a depender do tamanho da tela e da responsividade que o sistema aplicar

O sistema deve deixar as opções como se fosse um checkbox, porém não um checkbox com ☒, mas sim um de seleção, que ao estar selecionado, o ícone torna-se circundado em azul ou laranja, a depender do contraste que for escolhido.

O sistema deve exibir nessa área da tela os gêneros listados:

- Adventure
- Animation
- Children
- Comedy
- Fantasy
- Romance

- Drama
- Action
- Crime
- Thriller
- Horror
- Mystery
- Sci-Fi
- IMAX
- Documentary
- War
- Musical
- Western
- Film-Noir

O sistema pode ser pensado para ser em inglês, com seus títulos também em inglês, visto que o DATASET obtido da internet é estrangeiro.

O sistema deve dar um jeito de possuir organizados em linhas e colunas essas opções, não necessariamente alinhados, mas organizados, de certa forma que dê para ler todos e que seja possível selecionar.

O sistema deve conter abaixo dessa parte, uma espécie de caixa de texto com o título acima dela: “Digite aqui alguma observação para o Modelo de Recomendação de Filmes”

O sistema deve colocar uma restrição caso a caixa não seja preenchida, que deve conter pelo menos 1 frase de 10 palavras. Caso não possua, não deve permitir que a requisição seja enviada.

O mais importante é realizar todos os ajustes antes da requisição ser enviada, e montá-la da maneira mais polida possível, pois o que será retornado é o que fará do sistema o que ele é.

O sistema deve conter abaixo desse título, a caixa de texto em si, que permitirá ao usuário que esteja procurando uma boa recomendação de filmes, digitar alguma observação relevante para que a LLM faça a leitura e retorne uma melhor resposta de recomendação dos filmes, como por exemplo:

- Selecione os gêneros Comedy, War, Drama, e na caixa de texto digite “Quero que dê preferências a um filme de comédia” -> nesse caso, a LLM poderá recomendar os outros gêneros, mas dará preferência a um filme de comédia, trazendo curiosidades sobre ele, principalmente.
- Selecione somente o gênero de ação, e na caixa de texto digite “Recomende apenas filmes a partir dos anos 2000” -> A LLM irá filtrar por todos os filmes de ação, e recomendar os melhores filmes de ação a partir dos anos 2000.
- É possível ter outras configurações disponíveis, que serão pensadas ao longo do desenvolvimento do projeto.

IDEIA:

O sistema pode utilizar de uma técnica de Engenharia de Prompt para treinamento da LLM, chamado COSTAR, do qual será definido um documento de configuração dividido em passos: Contexto, Objetivo, Estilo, Tom, Público-Alvo e Tipo da resposta. Onde havendo configurado o tipo de prompt desejado previamente, as respostas serão muito mais precisas e coesas para o usuário.

O sistema deve exibir abaixo da caixa de texto, um botão de “Enviar”, que irá enviar a requisição para o back-end, que processará todas as informações e retornará em outra página da sessão.

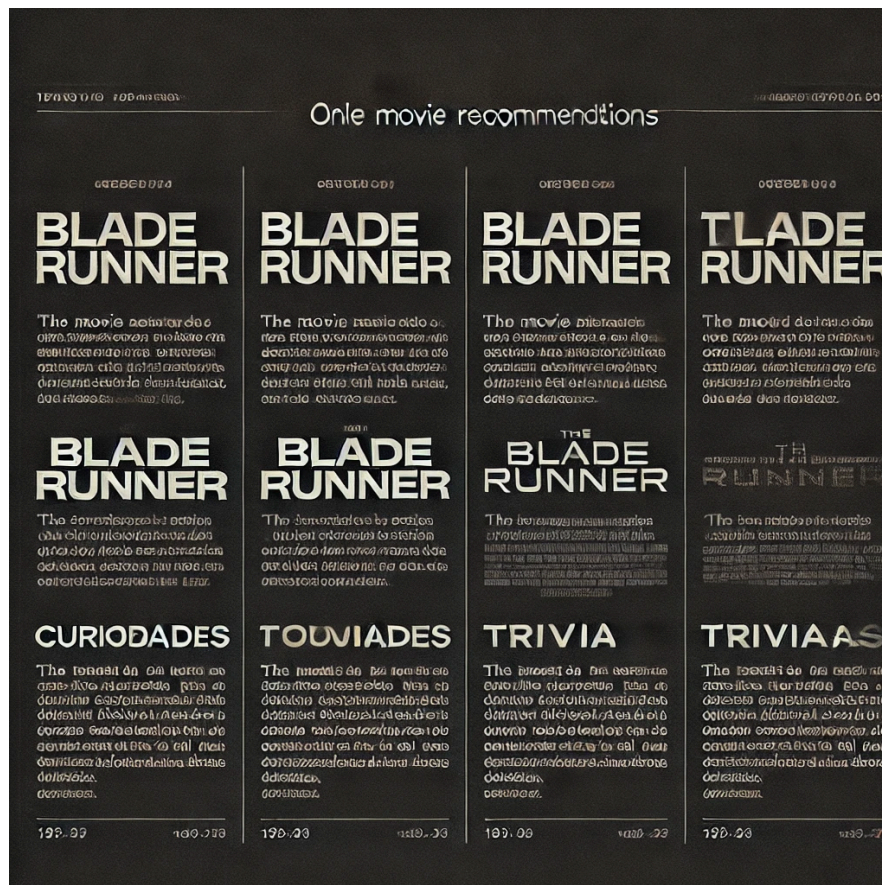
O sistema deve exibir essa nova página como sendo uma página de exibição de conteúdo simples, como por exemplo contendo o título do filme e algumas curiosidades sobre ele.

O sistema deve exibir, caso seja recomendado mais de um filme no retorno da requisição, dividi-los de maneira harmoniosa.

O sistema deve separar as curiosidades sobre os filmes em tópicos utilizando de bullets ou até mesmo parágrafos para cada curiosidade, ou separados por partes, assim como na imagem contida mais abaixo. O melhor é que seja da maneira que ficar mais interessante.

O sistema deve limitar a recomendação de até 10 filmes, configurando para que sejam 10 filmes de 1 gênero só, ou sendo definido da maneira que a LLM recomendar (se possível). Porém para diminuir a complexidade, deve ser possível limitar isso na configuração do sistema.

O sistema deve exibir, caso o usuário selecione o número máximo de gêneros (10), e esteja configurado para retornar apenas 5 filmes, a LLM deve recomendar apenas os cinco primeiros que forem retornados. **Essa questão de complexidade pode ficar para depois, o simples vem primeiro.**



Exemplo muito interessante do retorno da tela de recomendação.

Descrição do Frontend - Por filmes

O sistema deve exibir em sua tela inicial, uma listagem de todos os filmes disponíveis na plataforma, a depender do dataset utilizado e de sua quantidade de filmes disponíveis.

O sistema deve exibir esses filmes ordenados através de seus IDs, contendo até 20 filmes por página ou ordenados através do ano de lançamento, do mais atual para o mais antigo. Por exemplo: Top Gun: Maverick (2022) ficaria antes de Spider-Man: No Way Home.

O sistema deve exibir uma opção de paginação, ou um sistema de rolagem infinita, carregando a página infinitamente até o limite de filmes.

O sistema deve permitir filtrar os filmes por ID, Ordem Alfabética ou mesmo “nota”, caso tenha essa opção no DATASET escolhido para o site.

O sistema deve exibir na página dividido por alguns parâmetros:

- Título do filme (com maior fonte)
- Ano de lançamento do filme, logo abaixo do título
- Gênero (s) do filme,
- Nota do filme (SE EXISTIR)
- Sinopse (SE EXISTIR)

O sistema deve exibir a sinopse apenas se o usuário clicar no filme, exibindo um texto em uma tela destacada, uma espécie de modal com o conteúdo da sinopse.

O sistema deve exibir apenas o título da sinopse com um botão “Exibir” para que se clicado, apareça o modal de sinopse, a depender da organização que o filme será exibido.

O sistema deve exibir uma espécie de caixa de seleção com um símbolo de “+”, para que se o usuário selecionar, ele estará adicionando o filme à uma lista de assistidos.

O sistema deve conter, nesse aspecto, uma funcionalidade de recomendação dos filmes, agora baseado no seu perfil de filmes assistidos. Tendo uma página separada apenas para receber as recomendações baseadas nos filmes assistidos.

O sistema deve armazenar os filmes dessa lista em uma tabela própria, contendo a ligação para esse usuário, ao ser adicionado à lista, que deve ser exibida nesta página.

O sistema teria agora que ao realizar a recomendação, verificar quais foram assistidos e retornar os filmes da recomendação, com exceção dos assistidos.

O sistema deve ser implementado dessa maneira, para evitar possíveis problemas futuros e garantir possível expansibilidade.

É mais simples na parte de configuração da LLM, porém mais complicado na parte da implementação, pois são mais pontos.

Provavelmente será usado o dataset do MovieLens com 87.000 filmes, que já possui parâmetros separados corretamente através de manipulações feitas pelo Gustavo. O dataset está salvo em um banco de dados POSTGRES.

Descrição do Backend - Por Gêneros

Segue dois capítulos de maneiras para fazer o backend do sistema, ao realizar a recomendação por gêneros ou apenas pelos filmes.

Havendo a filtragem pelos gêneros disponíveis, temos o seguinte contexto:
Os gêneros de filmes foram selecionados através dos checkbox, então os que foram selecionados, serão enviados no payload através de uma requisição para a API.

Lembrando que há pouco mais de 7000 filmes que não contém gênero especificado, para este dataset em específico.

Digamos que os filmes estão em um arquivo JSON, contendo um dataset de pouco mais de 87.000 filmes, divididos da seguinte maneira:

```
{
  "movieId": 1954,
  "title": "Rocky",
  "genres": "Drama",
  "year": "1976"
```



```
} ,
```

Levando em conta que se necessita de algo relacionado à aprendizagem de máquina, podemos fazer da seguinte maneira, ao dividirmos por gêneros:

- O usuário selecionaria seus gêneros preferidos, que irão na requisição para onde estão armazenados os filmes, podendo filtrar de alguma forma os gêneros pelo ID do filme, seja no banco ou em um JSON.
- A LLM iria receber aqueles filmes contendo apenas os gêneros selecionados, e iria processá-los.
- O modelo usa sua capacidade de compreensão para filtrar filmes com base no gênero e gerar uma resposta personalizada.
- A LLM após processá-los, iria retornar em um formato específico, talvez JSON, e essa resposta para a requisição seria exibida na tela do navegador, listado na tela de uma maneira que fosse visível para o usuário.

Descrição do Backend - Por Filmes

Havendo a filtragem pelos filmes, teríamos uma listagem de todos os filmes organizados em um sistema de paginação, onde todos esses que apareceriam teríamos a opção de adicioná-los à uma lista de “Filmes que gosto”, e a partir dessa lista separada, teríamos um botão em algum lugar da página onde seria selecionado.

Quando o botão for selecionado, o que deve acontecer é:

- Tendo em vista que os filmes foram adicionados a uma lista específica em algum arquivo ou banco de dados, ao selecionar o botão, digamos que têm 20 filmes na lista.
- O botão de “Recomendar-me filmes” irá enviar uma requisição à API que será processada pela LLM que analisará todos os filmes da lista, ou o máximo de filmes em um limite estabelecido.
- A LLM irá processar toda aquela quantidade de filmes e retornar em um formato específico ou armazenar em um banco através do retorno da requisição.
- O retorno da requisição irá fazer exibir em uma nova tela gerada através dessa mesma requisição, exibindo os filmes recomendados de acordo com o que a LLM analisou e aprendeu, com todos os filmes dentro do mesmo contexto da lista do “usuário”.
- Deve-se definir se tudo será de maneira local ou se realmente deverá existir um banco de dados (provavelmente sim).

A existência de um banco de dados se vê importante, devido aos filmes serem armazenados, e a partir de uma query, ter retornada uma quantidade específica de filmes para um formato JSON que será lido pela LLM, processada, e retornada para outro arquivo, e este arquivo será processado para ser exibido em uma página separada do Frontend.

Descrição de qual é a melhor decisão

Levando em conta a simplicidade de implementação, a abordagem dos gêneros é mais atrativa e simples, pois basta filtrar todos os filmes a partir dos gêneros selecionados, podendo ser até mesmo 1 gênero, e fazer uma consulta para filtrar apenas os filmes daquele gênero, e enviar à LLM, para essa poder processar e retornar o resultado de acordo com a configuração imposta nela.

Levando em conta a abordagem pelos filmes, se torna mais complicada pela questão da paginação, onde teremos que armazenar todos os filmes em um banco, e exibi-los na página inicial, separando em diversas páginas e tentando organizar de uma forma boa de visualizar, além de ter o botão de adicionar à lista de “Filmes que gosto”, que deverá ter em todos os milhares de filmes possíveis, no entanto a recomendação será feita com a LLM recebendo os títulos de cada filme, retornando recomendações de acordo com o que o usuário adicionou a aquela lista.

Conclusão, ambos têm uma implementação semelhante com relação ao aprendizado de máquina, com a parte mais complicada sendo apenas a questão das queries e exibição das informações no frontend. E mesmo havendo filmes com gêneros não exibidos, é possível preenchê-los de alguma forma para a LLM recomendar certos filmes, com base no nome e gênero ao mesmo tempo.

Fica a critério da equipe decidir qual vai ser a melhor opção.