

Especificações do Software: MovieApp - Visualizador de Filmes em Cartaz

1. Introdução

O MovieApp é um aplicativo desenvolvido utilizando a tecnologia React.js, com a finalidade de proporcionar aos usuários uma experiência moderna para explorar filmes em cartaz. Conectando-se à API do TMDb (The Movie Database), o sistema possibilita a busca, visualização de detalhes, classificação e aplicação de filtros em filmes, oferecendo uma interface responsiva e intuitiva.

O principal objetivo é criar uma experiência fluida e agradável para amantes de cinema que desejam acompanhar os lançamentos de forma prática.

2. Levantamento de requisitos

Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.1. Requisitos funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 1 apresenta os requisitos funcionais deste projeto.

Tabela 1: Requisitos funcionais

Identificação	Requisito funcional	Prioridade
RF001	Listar filmes em cartaz	Essencial
RF002	Buscar filme por nome	Essencial
RF003	Visualizar detalhes do filme	Essencial
RF004	Aplicar filtros de exibição	Essencial

Fonte: Acervo do autor, 2025

2.1.1. Descrição dos requisitos funcionais

Nesta sessão estão descritas as especificações dos requisitos funcionais previstos na Tabela 1.

2.1.1.1. MovieApp React – Requisitos funcionais

Este requisito funcional descreve os casos de uso relacionados a inserção, alteração, exclusão e busca de um ou mais animais no banco de dados.

Tabela 2: RF001- Listar filmes em cartaz

RF001	
Nome	Listar filmes em cartaz
Descrição	Exibe a listagem dos filmes em cartaz, com pôster, título e classificação.
Plataforma	React.js Web
Atores	Usuário final
Prioridade	Essencial
Requisito não funcional associado	RNF001; RNF002; RNF004
Entradas e Restrições	Nenhuma
Fluxo de eventos	
Fluxo principal:	Ao acessar a aplicação, o usuário visualiza os filmes em cartaz.
Fluxo secundário:	Caso a API não responda, exibir mensagem de erro.

Fonte: Acervo do autor, 2025

Tabela 3: RF002: Buscar filme por nome

RF002	
Nome:	Buscar filme por nome
Descrição	Permite a pesquisa personalizada por nome de filme.
Plataforma	React.js Web
Atores	Usuário final
Prioridade	Essencial
Requisito não funcional associado	RNF001; RNF002
Entradas e Restrições	Nome do filme (texto, obrigatório)
Fluxo de eventos	
Fluxo principal:	O usuário digita o nome e visualiza os filmes correspondentes.
Fluxo secundário 1:	Caso não encontre resultados, exibir mensagem 'Filme não encontrado'.

Fonte: Acervo do autor, 2025

Tabela 4 RF003: Visualizar detalhes do filme

RF003	
Nome	Visualizar detalhes do filme
Descrição	Exibe sinopse, data de lançamento, avaliação, elenco e trailers do filme selecionado.
Plataforma	React.js Web
Atores	Usuário final
Prioridade	Essencial
Requisito não funcional associado	RNF001; RNF002
Entradas e Restrições	Selecionar filme na lista

Fluxo de eventos	
Fluxo principal:	O usuário clica no filme e visualiza os detalhes.
Fluxo secundário 1:	Caso falhe o carregamento dos detalhes, exibir mensagem de erro.

Fonte: Acervo do autor, 2025

Tabela 5 RF004: Aplicar filtros de exibição

RF001	
Nome	Aplicar filtros de exibição
Descrição	Permite filtrar os filmes pela popularidade, avaliação ou data de lançamento.
Plataforma	React.js Web
Atores	Usuário final
Prioridade	Essencial
Requisito não funcional associado	RNF001; RNF002
Entradas e Restrições	Opção de filtro selecionada (obrigatória)
Fluxo de eventos	
Fluxo principal:	O usuário seleciona um filtro e a lista de filmes é atualizada.
Fluxo secundário 1:	Caso a filtragem falhe, exibir mensagem de erro.

Fonte: Acervo do autor, 2025

2.2. Requisitos não funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 6 apresenta os requisitos não funcionais deste projeto.

2.2.1. MovieApp React – Requisitos não funcionais

Tabela 6: Requisitos Não Funcionais

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Interface responsiva para dispositivos móveis, tablets e desktop	Usabilidade	Essencial
RNF002	Consumo de dados em tempo real via API TMDb	Integração	Essencial
RNF003	Gerenciamento de estado eficiente usando Context API	Arquitetura	Essencial
RNF004	Estilização responsiva com Tailwind CSS	Padrões	Essencial

RNF005	Versionamento de código no GitHub	Gerenciamento	Essencial
--------	-----------------------------------	---------------	-----------

Fonte: Acervo do autor, 2025

2.2.2. Descrição dos requisitos não funcionais

Os requisitos não funcionais envolvem a escolha da arquitetura do sistema, a plataforma que o programa será implantado (*desktop*, intranet, web, etc), a linguagem de programação que será utilizada, o banco de dados apropriado para a aplicação, entre outras escolhas.

Tabela 7: RNF001 - Interface responsiva

RNF001	
Identificação	RNF001
Nome	Interface responsiva
Descrição	O MovieApp deve se adaptar a diferentes tamanhos de tela, garantindo a melhor experiência de uso independente do dispositivo.
Categoria	Usabilidade
Prioridade	Essencial

Fonte: Acervo do autor, 2025

Tabela 8: RNF002 - Consumo de dados em tempo real

RNF001	
Identificação	RNF002
Nome	Consumo de dados em tempo real
Descrição	Todas as informações exibidas devem ser consultadas em tempo real diretamente da API TMDb.
Categoria	Integração
Prioridade	Essencial

Fonte: Acervo do autor, 2025

Tabela 9: RNF003 - Gerenciamento de estado

RNF001	
Identificação	RNF003
Nome	Gerenciamento de estado
Descrição	A Context API será utilizada para gerenciar o estado da aplicação, mantendo a consistência de dados durante a navegação.
Categoria	Arquitetura
Prioridade	Essencial

Fonte: Acervo do autor, 2025

Tabela 10: RNF004 - Estilização responsiva

RNF001	
Identificação	RNF004
Nome	Estilização responsiva
Descrição	O layout e o design serão implementados utilizando o framework Tailwind CSS, garantindo padronização visual e responsiva.
Categoria	Padrões
Prioridade	Essencial

Fonte: Acervo do autor, 2025

Tabela 11: RNF004 - Estilização responsiva

RNF001	
Identificação	RNF004
Nome	Estilização responsiva
Descrição	O layout e o design serão implementados utilizando o framework Tailwind CSS, garantindo padronização visual e responsiva.
Categoria	Padrões
Prioridade	Essencial

Fonte: Acervo do autor, 2025

Tabela 12: RNF005 - Versionamento de código

RNF001	
Identificação	RNF005
Nome	Versionamento de código
Descrição	Todo o histórico de desenvolvimento do projeto será mantido e atualizado através da plataforma GitHub.
Categoria	Gerenciamento
Prioridade	Essencial

Fonte: Acervo do autor, 2025