

CineBuy – Sistema de Reserva de Ingressos de Filmes

Documentação do Projeto Final para a Academia.Net

Por: Rebeca Paulino Feitosa Silva

CineBuy – Sistema de Reserva de Ingressos de Filmes	1
1. Visão Geral do Projeto	2
1.1 Introdução	2
1.2 Funcionalidades.....	2
2. Configuração	2
2.1 Requisitos.....	2
2.2 Dependências.....	2
2.3 Tecnologias:	3
3. Banco de Dados	3
3.1 Modelo de Dados	3
3.2 Diagrama ER (Entidade-Relacionamento).....	4
4. API	4
4.1 Descrição	4
4.2 Endpoints	5
5. Estrutura do Projeto.....	10
5.1 Pasta Models	10
5.2 Pasta Images	10
5.3 Pasta Content	10
5.4 Pasta Views	10
6. GitHub.....	11

1. Visão Geral do Projeto

1.1 Introdução

Objetivo: O CineBuy é um sistema de reserva de ingressos para sessões de filmes que permite aos usuários visualizar filmes, reservar assentos, gerar confirmação de reserva e cancelar reservas.

1.2 Funcionalidades

Lista de Funcionalidades:

- Visualização de filmes
- Seleção de horários disponíveis
- Reserva de assentos
- Geração de confirmações de reserva
- Cancelamento de reservas

2. Configuração

2.1 Requisitos

Ambiente de Desenvolvimento:

- Visual Studio 2022
- SQL Server v20.1

2.2 Dependências

API:

- ASP.NET Core 8.0
- Pacotes Instalados:
 - Dapper
 - System.Data.SqlClient

WebForms:

- Aplicativo Web ASP.NET (.NET Framework) 4.7.2
- Pacotes Instalados:
 - System.Net.Http
 - System.Net.Http.Formatting.Extension

2.3 Tecnologias:

- C#
- ASP.NET
- ASP.NET Web Forms
- API Web ASP.NET Core
- HTML
- CSS
- Bootstrap

3. Banco de Dados

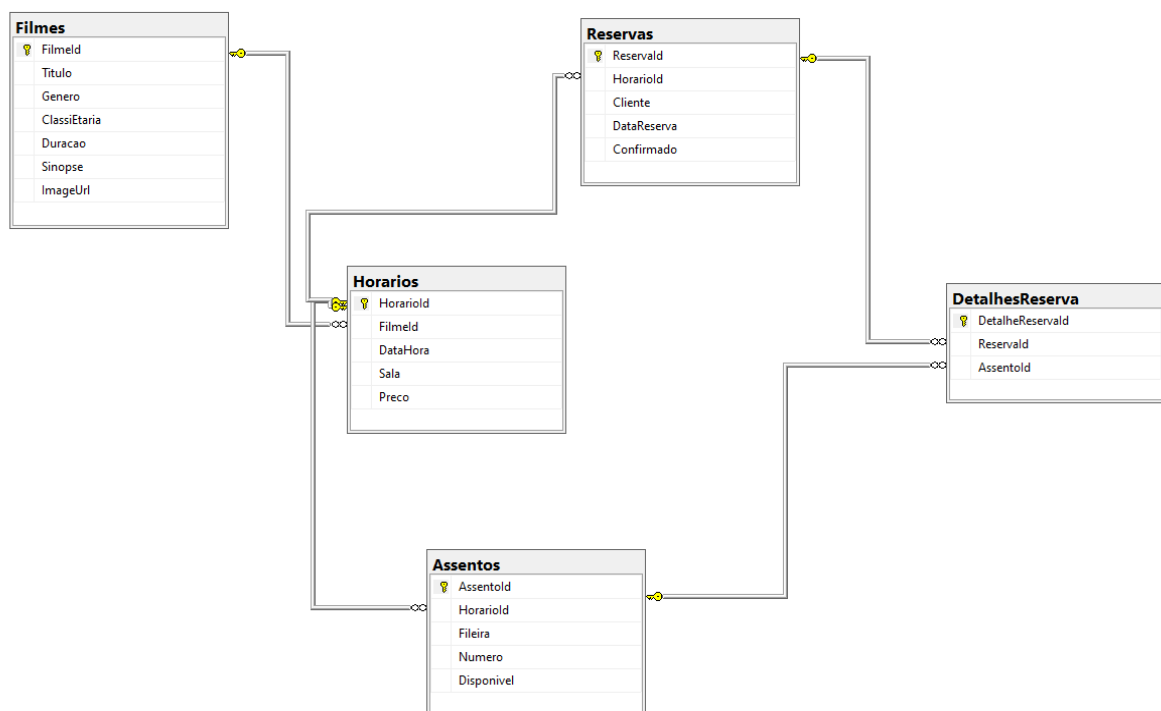
3.1 Modelo de Dados

O banco de dados contém as seguintes tabelas:

- **Filmes:** Responsável por armazenar informações do filme, contendo as colunas:
 - FilmId
 - Titulo
 - Genero
 - ClassiEtaria
 - Duração
 - Sinopse
- **Horários:** Responsável por armazenar horários disponíveis para cada filme, contendo as colunas:
 - HorarioId
 - FilmId
 - DataHora
 - Sala
 - Preço
- **Assentos:** Responsável por armazenar os assentos referentes aos horários, contendo as colunas:
 - AssentoId
 - HorarioId
 - Fileira
 - Numero
 - Disponivel
- **Reservas:** Responsável por armazenar as reservas feitas, contendo as colunas:
 - ReservaId
 - HorarioId

- Cliente
- DataReserva
- Confirmado
- **Detalhes da Reserva:** Responsável por armazenar os detalhes da reserva, contendo as colunas:
 - DetalheReservald
 - Reservald
 - Assentold

3.2 Diagrama ER (Entidade-Relacionamento)



4. API

4.1 Descrição

A API do CineBuy é responsável por fornecer uma interface para interação com o sistema de reserva de ingressos. Ela permite que clientes e o front-end se comuniquem com o back-end para realizar operações como a visualização de filmes, a seleção de horários, a reserva de assentos e o gerenciamento de reservas.

4.2 Endpoints

1. Filmes

Base URL: <http://localhost:5109/api/Filme>

- **GET /api/Filme/BuscarTodosFilmes**
 - **Descrição:** Retorna uma lista de todos os filmes disponíveis no sistema.
 - **Resposta:**
 - 200 OK: Retorna a lista de filmes.
 - 500 Internal Server Error: Erro no servidor ao buscar os filmes.
- **GET /api/Filme/BuscarFilme/{codigo}**
 - **Descrição:** Retorna detalhes de um filme específico com base no Filmeld.
 - **Parâmetro(s):** codigo (int): O ID do filme.
 - **Resposta:**
 - 200 OK: Retorna os detalhes do filme.
 - 404 Not Found: Filme não encontrado.
 - 500 Internal Server Error: Erro no servidor ao buscar o filme.
- **POST /api/Filme/AdicionarFilme**
 - **Descrição:** Adiciona um novo filme ao sistema.
 - **Parâmetro(s):** Corpo da requisição: FilmeRequest contendo os dados do filme a ser adicionado.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas adicionadas com o filme.
 - 500 Internal Server Error: Erro no servidor ao adicionar o filme.
- **PUT /api/Filme/AlterarFilme/{codigo}**
 - **Descrição:** Atualiza os dados de um filme existente com base no Filmeld.
 - **Parâmetro(s):** codigo (int): O ID do filme.
 - **Corpo da requisição:** FilmeRequest com os novos dados do filme.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas alteradas.
 - 500 Internal Server Error: Erro no servidor ao alterar o filme.
- **DELETE /api/Filme/ExcluirFilme/{codigo}**
 - **Descrição:** Exclui um filme existente com base no Filmeld.
 - **Parâmetro(s):** codigo (int): O ID do filme.

- **Resposta:**
 - 200 OK: Retorna o total de linhas deletadas.
 - 500 Internal Server Error: Erro no servidor ao excluir o filme.

2. Horários

Base URL: <http://localhost:5109/api/Horario>

- **GET /api/Horario/BuscarTodosHorarios**
 - **Descrição:** Retorna uma lista de todos os horários disponíveis no sistema.
 - **Resposta:**
 - 200 OK: Retorna a lista de horários.
 - 500 Internal Server Error: Erro no servidor ao buscar os horários.
- **GET /api/Horario/BuscarHorariosFilme/{codigo}**
 - **Descrição:** Retorna todos os horários de um filme específico com base no Filmeld.
 - **Parâmetro(s):** codigo (int): O ID do filme.
 - **Resposta:**
 - 200 OK: Retorna a lista de horários do filme.
 - 404 Not Found: Nenhum horário encontrado para o filme especificado.
 - 500 Internal Server Error: Erro no servidor ao buscar os horários.
- **GET /api/Horario/BuscarHorario/{codigo}**
 - **Descrição:** Retorna detalhes de um horário específico com base no Horariold.
 - **Parâmetro(s):** codigo (int): O ID do horário.
 - **Resposta:**
 - 200 OK: Retorna os detalhes do horário.
 - 404 Not Found: Horário não encontrado.
 - 500 Internal Server Error: Erro no servidor ao buscar o horário.
- **POST /api/Horario/AdicionarHorario**
 - **Descrição:** Adiciona um novo horário ao sistema.
 - **Parâmetro(s):** Corpo da requisição: HorarioRequest contendo os dados do novo horário a ser adicionado.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas adicionadas com o novo horário.

- 500 Internal Server Error: Erro no servidor ao adicionar o horário.
- **PUT /api/Horario/AlterarHorario/{codigo}**
 - **Descrição:** Atualiza os dados de um horário existente com base no Horariold.
 - **Parâmetro(s):** codigo (int): O ID do horário.
 - **Corpo da requisição:** HorarioRequest com os novos dados do horário.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas alteradas.
 - 500 Internal Server Error: Erro no servidor ao alterar o horário.
- **DELETE /api/Horario/ExcluirHorario/{codigo}**
 - **Descrição:** Exclui um horário existente com base no Horariold.
 - **Parâmetro(s):** codigo (int): O ID do horário.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas deletadas.
 - 500 Internal Server Error: Erro no servidor ao excluir o horário.

3. Assentos

Base URL: <http://localhost:5109/api/Assento>

- **GET /api/Assento/BuscarTodosAssentos**
 - **Descrição:** Retorna uma lista de todos os assentos disponíveis.
 - **Resposta:**
 - 200 OK: Retorna a lista de assentos.
 - 500 Internal Server Error: Erro no servidor ao buscar os assentos.
- **GET /api/Assento/BuscarAssento/{codigo}**
 - **Descrição:** Retorna detalhes de um assento específico com base no Assentold.
 - **Parâmetro(s):** codigo (int): O ID do assento.
 - **Resposta:**
 - 200 OK: Retorna os detalhes do assento.
 - 404 Not Found: Assento não encontrado.
 - 500 Internal Server Error: Erro no servidor ao buscar o assento.
- **GET /api/Assento/BuscarAssentosPorHorario/{codigo}**
 - **Descrição:** Retorna os assentos disponíveis para um determinado horário.

- **Parâmetro(s):** codigo (int): O ID do horário.
- **Resposta:**
 - 200 OK: Retorna os assentos para o horário especificado.
 - 404 Not Found: Nenhum assento encontrado para o horário especificado.
 - 500 Internal Server Error: Erro no servidor ao buscar os assentos.
- **POST /api/Assento/AdicionarAssento**
 - **Descrição:** Adiciona um novo assento ao sistema.
 - **Parâmetro(s):** Corpo da requisição: AssentoRequest contendo os dados do novo assento.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas adicionadas com o novo assento.
 - 500 Internal Server Error: Erro no servidor ao adicionar o assento.
- **PUT /api/Assento/AlterarAssento/{codigo}**
 - **Descrição:** Atualiza os dados de um assento existente com base no Assentold.
 - **Parâmetro(s):** codigo (int): O ID do assento.
 - **Corpo da requisição:** AssentoRequest com os novos dados do assento.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas alteradas.
 - 500 Internal Server Error: Erro no servidor ao alterar o assento.
- **DELETE /api/Assento/ExcluirAssento/{codigo}**
 - **Descrição:** Exclui um assento existente com base no Assentold.
 - **Parâmetro(s):** codigo (int): O ID do assento.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas deletadas.
 - 500 Internal Server Error: Erro no servidor ao excluir o assento.

4. Reservas

Base URL: <http://localhost:5109/api/Reserva>

- **GET /api/Reserva/BuscarTodasReservas**
 - **Descrição:** Retorna uma lista de todas as reservas feitas.
 - **Resposta:**
 - 200 OK: Retorna a lista de reservas.

- 500 Internal Server Error: Erro no servidor ao buscar as reservas.
- **GET /api/Reserva/BuscarReserva/{codigo}**
 - **Descrição:** Retorna detalhes de uma reserva específica com base no ReservaId.
 - **Parâmetro(s):** codigo (int): O ID da reserva.
 - **Resposta:**
 - 200 OK: Retorna os detalhes da reserva.
 - 404 Not Found: Reserva não encontrada.
 - 500 Internal Server Error: Erro no servidor ao buscar a reserva.
- **POST /api/Reserva/AdicionarReserva**
 - **Descrição:** Adiciona uma nova reserva ao sistema.
 - **Parâmetro(s):** Corpo da requisição: ReservaRequest contendo os dados da nova reserva.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas adicionadas com a nova reserva.
 - 500 Internal Server Error: Erro no servidor ao adicionar a reserva.
- **PUT /api/Reserva/AlterarReserva/{codigo}**
 - **Descrição:** Atualiza os dados de uma reserva existente com base no ReservaId.
 - **Parâmetro(s):** codigo (int): O ID da reserva.
 - **Corpo da requisição:** ReservaRequest com os novos dados da reserva.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas alteradas.
 - 500 Internal Server Error: Erro no servidor ao alterar a reserva.
- **DELETE /api/Reserva/ExcluirReserva/{codigo}**
 - **Descrição:** Exclui uma reserva existente com base no ReservaId.
 - **Parâmetro(s):** codigo (int): O ID da reserva.
 - **Resposta:**
 - 200 OK: Retorna o total de linhas deletadas.
 - 500 Internal Server Error: Erro no servidor ao excluir a reserva.

5. Estrutura do Projeto

5.1 Pasta Models

Define a estrutura dos dados da aplicação e facilita a interação com o banco de dados e a lógica de negócios.

- **Classes:**
 - Filme.cs
 - Horário.cs
 - Reserva.cs
 - ReservaRequest.cs
 - Assentos.cs
 - AssentoRequest.cs

5.2 Pasta Images

Armazena as imagens de capa dos filmes usadas na aplicação.

5.3 Pasta Content

Armazena arquivos relacionados à estilização e design da aplicação.

5.4 Pasta Views

Contém as classes e arquivos responsáveis pela visualização e interação das páginas da aplicação.

- **Filmes.aspx:** Define a estrutura da página que exibe a lista de filmes em cartaz.
- **Filmes.aspx.cs:** Contém a lógica por trás da página Filmes.aspx.
- **ComprarFilme.aspx:** Define o layout e os componentes visuais da página de compra de ingressos.
- **ComprarFilme.aspx.cs:** Contém a lógica de interação da página ComprarFilme.aspx.
- **ConfirmaçãoReserva.aspx:** Define o layout da página que mostra os detalhes da reserva feita pelo usuário.
- **ConfirmaçãoReserva.aspx.cs:** Contém a lógica para exibir as informações da reserva confirmada e processar ações dos botões.
- **Contact.aspx:** Fornece informações de contato sobre o projeto, incluindo o e-mail para suporte e um link para o repositório do projeto no GitHub.

6. GitHub

O repositório do projeto CineBuy no GitHub contém o passo a passo para rodar o projeto, além de todos os arquivos necessários.

- **Link para o repositório:** [GitHub - ProjetoFinal-CineBuy](#)