Golden Raspberry Awards API

Esta aplicação é uma API RESTful para manipular dados de filmes do Golden Raspberry Awards, permitindo a leitura de indicados e vencedores da categoria Pior Filme.  
  
A API fornece endpoints para:  
1. Obter o produtor com o maior intervalo entre prêmios consecutivos.

2. Obter o produtor com o menor intervalo entre prêmios consecutivos.

3. Obter todos os dados salvos na tabela.

# Tecnologias Utilizadas

- .NET Core 9  
- Entity Framework Core com banco de dados em memória (H2 ou similar usando EF).  
- CsvHelper para leitura de arquivos CSV.  
- Swagger para documentação e testes da API.

# Configuração e Execução

## Pré-requisitos

- .NET SDK 9 ou superior instalado.  
- Editor de código, como Visual Studio, Rider ou Visual Studio Code.  
- Um arquivo CSV com os seguintes campos: 'year', 'title', 'studios', 'producers', 'winner'.

## Passos para Executar

1. Clone o repositório:  
 ```bash  
 git clone <https://github.com/lrenato12/GoldenRaspberryAwards>  
 cd golden-raspberry-api  
 ```  
  
2. Adicione o arquivo CSV:  
 - Certifique-se de que o arquivo CSV esteja localizado no diretório raiz do proejto com o nome 'movies.csv'.  
  
3. Compile o projeto:  
 ```bash  
 dotnet build  
 ```  
  
4. Execute a aplicação:  
 ```bash  
 dotnet run  
 ```  
  
5. Acesse a documentação do Swagger:  
 - Acesse https://localhost:7038/index.html no navegador para testar os endpoints.

# Endpoints da API

## 1. Obter os produtores com maior e menor intervalo entre prêmios consecutivos

- \*\*Rota\*\*: `GET /api/Movies/GetProducersInterval`  
  
 \*\*Exemplo de Resposta\*\*:  
 ```json  
 {  
 "min": [  
 {  
 "producer": "Producer 1",  
 "interval": 1,  
 "previousWin": 2008,  
 "followingWin": 2009  
 }  
 ],  
 "max": [  
 {  
 "producer": "Producer 2",  
 "interval": 10,  
 "previousWin": 2000,  
 "followingWin": 2010  
 }  
 ]  
 }  
 ```

# Testes

## Executar Testes de Integração

Os testes de integração garantem a consistência dos dados carregados e processados pela API. Para executá-los, use o comando:  
  
 ```bash  
 dotnet test  
 ```

# Estrutura do Projeto

API/  
├── Controllers/  
│ ├── MoviesController.cs # Endpoints da API.

├── Models/  
│ ├── MovieContext.cs # Contexto.

│ ├── MovieModel.cs # Modelo do filme.

│ ├── ProducerInterval.cs # Modelo do Producer.

│ ├── ProducerIntervalResponse.cs # Modelo do Producer Interval.

├── Services/  
│ ├── CsvLoaderService.cs # Leitura do CSV.

│ ├── MovieDataImporter.cs # Import do CSV.  
│ ├── MovieService.cs # Lógica para salvar os itens.

│ ├── ProducerIntervalService.cs # Lógica para recuperar os intervalor.

├── Resources/  
│ ├── movies.csv # Arquivo CSV com os dados.

├── Tests/  
│ ├── CsvLoaderServiceTests.cs # Testes unitários Load dos Movies.

│ ├── ProducerIntervalServiceTests.cs # Testes unitários para serviços.

├── Utils/  
│ ├── BooleanYesNoConverter.cs # Conversor de Boleano.

│ ├── MovieMap.cs # Map deMovie.

├── Program.cs

├── Movies.db

├── Movies.csv  
├── appsettings.json # Configurações.