2SEEORNOT

BILHETEIRA DE CINEMAS

PEDRO IGLÉSIAS 89318

WEI YE 93442

MARIANA LADEIRO 92964

DAVID MORAIS 93147

ROLES

Papel	Responsabilidades	Membro
Team Manager	Assegurar que existe uma distribuição justa das tarefas e que os membros trabalham de acordo com o planeado. Promover ativamente a colaboração entre os membros da equipa bem como tomar a iniciativa para endereçar os eventuais problemas que possam surgir. Assegurar que os resultados esperados do projeto são entregues dentro dos prazos.	Pedro Iglésias
Product Owner	Representa os interesses dos stakeholders. Tem um vasto conhecimento do produto e o seu domínio de aplicação. Pessoa a quem a equipa deverá colocar as dúvidas que tem sobre as features esperadas do produto.	Mariana Ladeiro
Arquiteto	Conhecimento vasto da arquitetura proposta bem como o suporte das tecnologias. Todas as dúvidas que possam surgir relativas ao comportamento de cada componente bem como a interação entre módulos deve ser esclarecida com este.	Wei Ye
DevOps Master	Responsável pelo desenvolvimento e produção de infraestruturas para as configurações requeridas. Assegurar que a ferramenta de desenvolvimento funciona de forma apropriada. Lidera a preparação do repositório git, operações da base de dados, entre outras.	David Morais
Developer	Contribuir para o desenvolvimento das tarefas.	Todos

IDEIA/OBJECTIVO

Conceito do produto

- Sistema de suporte a cinemas
- Venda e compra de bilhetes
- Informações sobre filmes
- Fazer reviews
- Adicionar, como dono de um cinema, o seu cinema

Personas e Cenários

- **Marta – procura plataforma para adicionar o seu cinema**
  - **Envia um request ao admin e, caso aceite, pode gerir os filmes.**
- **Lídia – procura um sistema intuitivo e fácil de usar**
  - **Efetua rapidamente a compra de bilhetes e, após o filme, faz uma review.**
- **Samuel – necessita de uma API com informações sobre filmes para desenvolver uma app**
  - **Utiliza a API do site para receber os dados que quer, em formato JSON, através de GETS.**

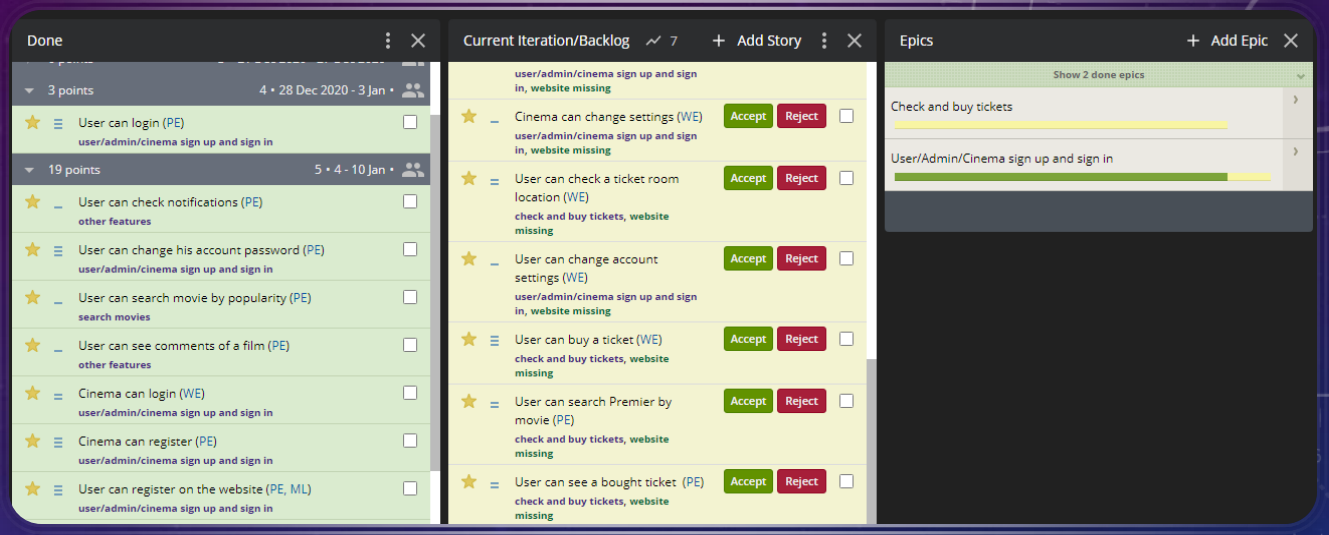
IDEIA/OBJECTIVO

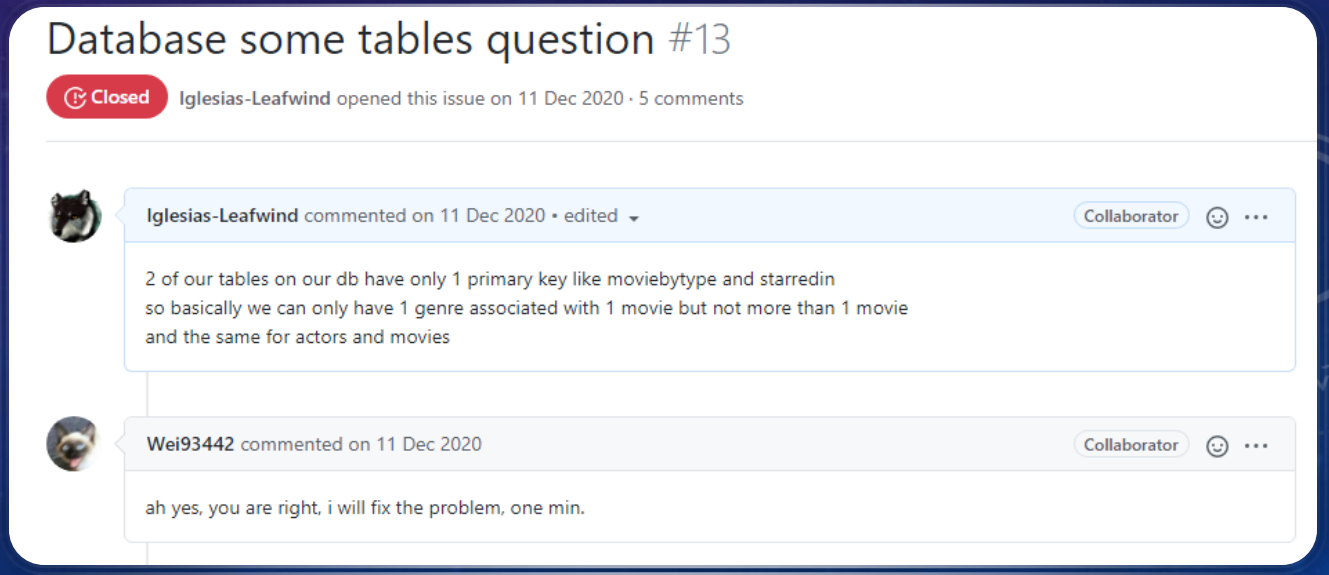
Limitações

- **Segurança na parte do cinema é bastante reduzida**
- **Criar uma release fácil de instalar**
- **O pagamento de um bilhete é meramente uma simulação**
- **Aplicação móvel não foi desenvolvida, como planeado inicialmente**

WORKFLOW / DEVOPS

- Como ferramente de gestão do backlog foi decidido usar Pivotal Tracker. Esta ferramenta permitirá acompanhar a evolução e o desenvolvimento do projeto
- Facilitou a visualização dos objetivos, permitiu gerir as entregas e as metas ao longo do trabalho e ajudou a dividir o projeto em partes mais simples.
- Para além disso a divisão do trabalho foi em módulos (epics) que por sua vez agrupam várias funcionalidades que estão desc ritas por user stories.
- As entregas devem ser revistas e analisadas antes de submetidas para que se tenha a certeza de que o objetivo foi concluído e que satisfaz os requisitos para isso no github para aceitar um pull request é necessário revisão e aprovação de pelo menos 50% dos membros neste projeto.
- Também foi criado uma dev branch no github para controlar e testar para verificar se tudo esta a funcionar como é suposto.
- Finalmente usamos github issues para discutir qualquer problema que ocoresse.





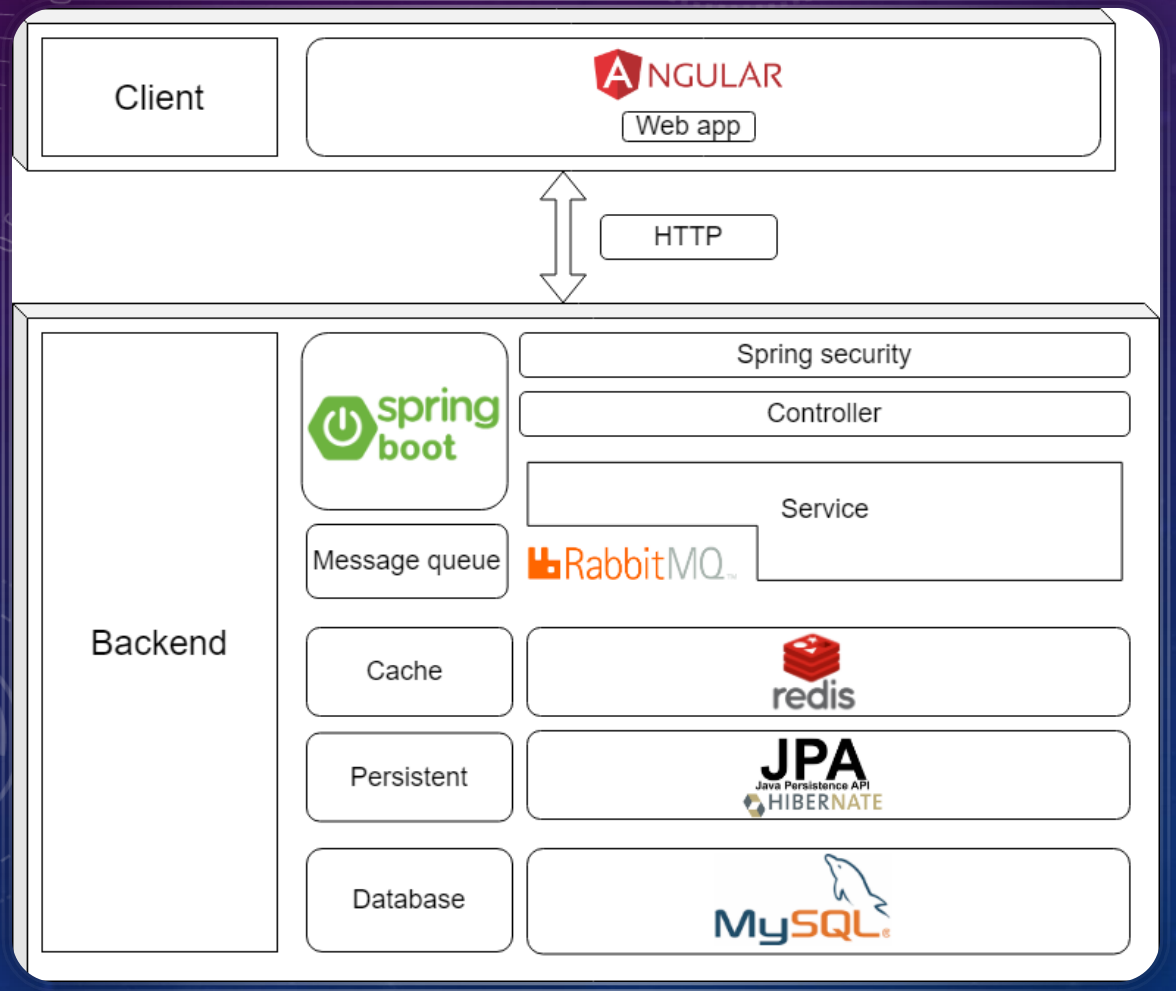
GERAÇÃO DE DADOS

Para a geração de dados foi usado um python script que faz download diretamente ao site (https://datasets.imdbws.com/) da imdb um dataset que após ser filtrado dá-nos todos filmes mais recentes (2020/2021) .

```
print("Downloading most recent imdb ids.")
url = 'https://datasets.imdbws.com/title.basics.tsv.gz'
r = requests.get(url, allow_redirects=True)
print("Loading dataset")
f_in = gzip.decompress(r.content)
with open('data.tsv', 'wb') as f_out:
    f_out.write(f_in)
with open('data.tsv', 'r', encoding="utf8") as file:
    print(file.readline())
    while(file.newlines):
        line = file.readline()
        try:
            arr = line.split("\t")
            imdbID,tipo,title,orgTitle,isadult,year,end,runtime,genres = arr
            if(isadult != "0" or tipo != "movie" or int(year) <= 2019 or "Documentary" in genres or runtime == "\\N"):
                continue
            print(imdbID,tipo,year)
            addMovie(imdbID)
        except:
            continue
```

Como esta geração de dados é lenta foi criado outro python script para mudar os dados constantemente para verificarmos as mudanças e utilização da nossa aplicação.

ARQUITETURA



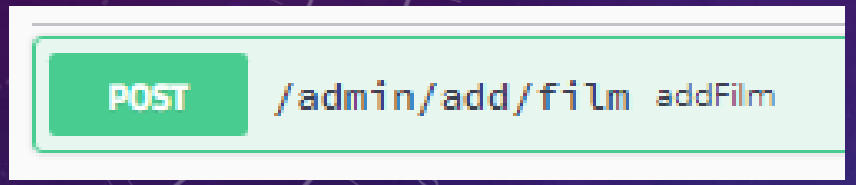
SERVIÇOS/FUNCIONALIDADES USER CONTROLLER

POST	/common/confirm/cinema/{code}	confirmRegisterCinema
POST	/common/login	login
POST	/common/register	register
POST	/common/register/cinema	registerCinema

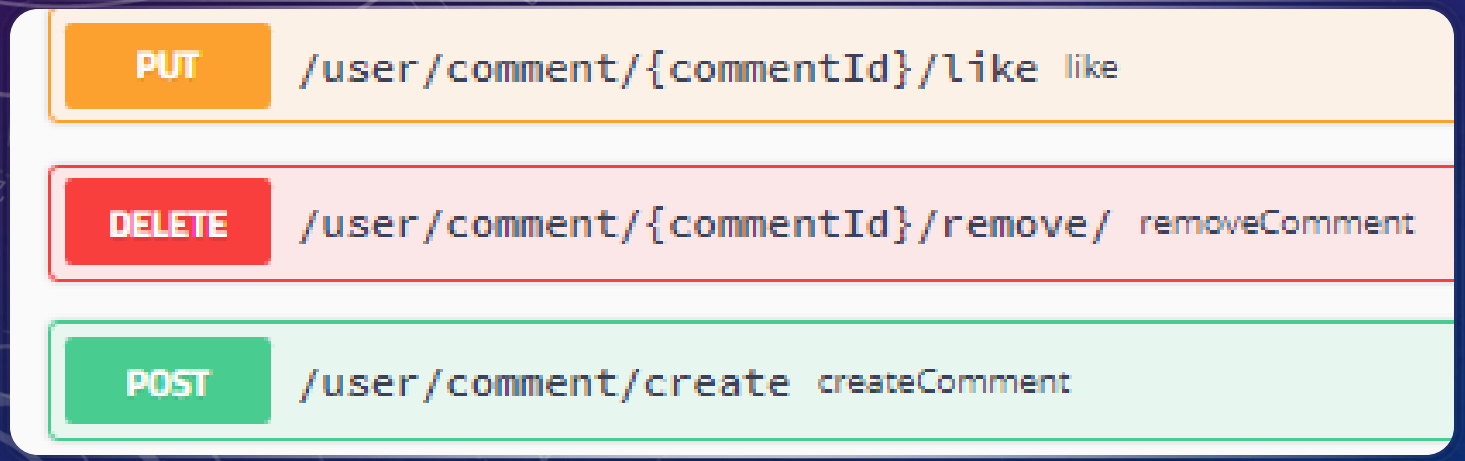
GET	/admin/requests	getRequests
GET	/admin/requests/{requestId}	getRequest
PUT	/admin/requests/accepted/{requestId}	acceptedRequest
PUT	/admin/requests/refuse/{requestId}	refuseRequest

POST	/user/add/favourite/cinema/{cinemaId}	addFavouriteCinema
POST	/user/add/favourite/film/{filmId}	addFavouriteFilm
POST	/user/buy/ticket	buyTicket
PUT	/user/change/avatar	change
PUT	/user/change/password	changePassword
GET	/user/favourite/cinema	getUserFavouriteCinema
GET	/user/favourite/films	getUserFavouriteFilms
GET	/user/notifications	getNotifications
GET	/user/payments	getPayments
DELETE	/user/remove/favourite/cinema/{cinemaId}	removeFavouriteCinema
DELETE	/user/remove/favourite/film/{filmId}	removeFavouriteFilm

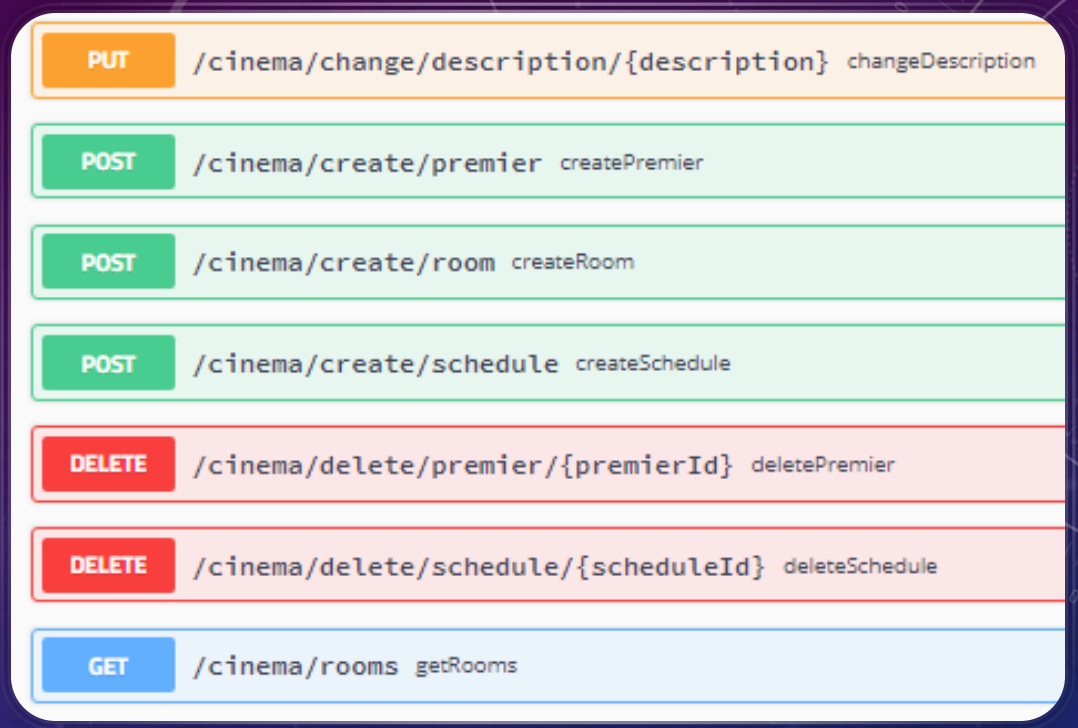
SERVIÇOS/FUNCIONALIDADES FILM CONTROLLER

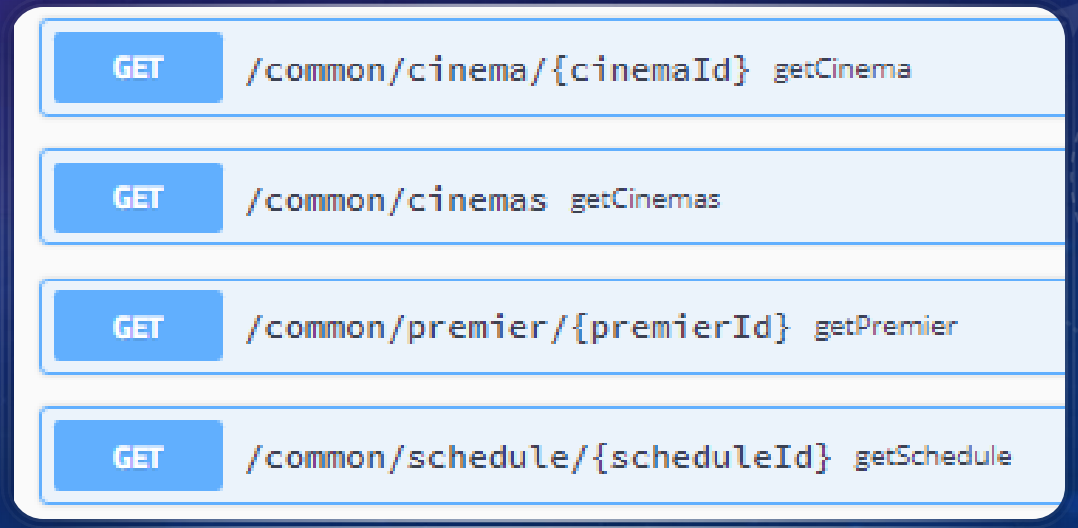


SERVIÇOS/FUNCIONALIDADES COMMENT CONTROLLER



SERVIÇOS/FUNCIONALIDADES CINEMA CONTROLLER





DEMO DA SOLUÇÃO