Orientações Gerais

O Projeto a ser desenvolvido deve considerar as melhores práticas da linguagem escolhida para ser realizado.
O Projeto deve ser estruturado considerando que será escalado de maneira **horizontal**.
Nós queremos que você desenvolva um serviço orientado à API que seja capaz de executar as tarefas citadas nos tópicos abaixo.
As tecnologias de persistência de dados devem ser **MySQL** e **MongoDB**.
O projeto deve ser desenvolvido tanto em **TypeScript**, o uso de frameworks e ORMs é livre.
Recomendamos Mongoose para MongoDB e Knex ou Prisma para o MySQL.
O código fonte deve estar em **inglês**.

O Projeto

O projeto consiste de uma api capaz de gerenciar usuários; assinaturas de pacotes por esses usuários; consumo da api do *movie database* para consumir e armazenar temas que vão compor as informações dos pacotes; listar filmes para os usuários de acordo com os temas permitidos por seus pacotes; e armazenar filmes assistidos por esses usuários e emitir um relatório com base nos filmes assistidos pelos usuários.

Parte 1 - Usuários

Crie um serviço que exponha as seguintes funcionalidades:

- Criação de usuários e autenticação (JWT)
- Recuperação de senha
  Critérios obrigatórios:
- Usuários devem estar armazenados no banco SQL
- Um usuário não deve ser capaz de alterar outro
- Melhores práticas de armazenamento de senha e autenticação

Parte 2 - Pacotes

Elabore um estrutura de dados capaz de mapear a funcionalidade de assinaturas de um pacote para um ou mais usuários;
É necessário que um pacote armazene os seguintes dados:

- nome
- temas habilitados (mais de um tema por pacote)
- data de criação

- data de última alteração
- versão
  Por favor, considere quaisquer possíveis problemas com as funcionalidades fornecidas e faça as suposições necessárias para evitá-los.

Parte 3 - Serviço Externo - Temas (Genre)

Rotina que busque temas disponíveis para poderem ser selecionados na hora de criar um pacote.
Criar uma rotina que mantenha uma lista atualizada de temas para serem selecionados pelos usuários na hora de criar pacotes.
Manter essa lista armazenada no mongodb.
Ao iniciar à aplicação é necessário ver se existe uma lista de temas já armazenada no banco, se não iniciar uma rotina que a cada 2 horas verifique os temas disponíveis no sistema externo e atualize eles se necessário.
Expor as informações de temas em um serviço com retorno paginado com filtros não obrigatórios porém pertinentes para auxiliar na criação de pacotes.

Parte 4 - Serviço Externo - Filmes

Expor um serviço capaz de listar filmes permitidos para o usuário de forma paginada.
Usuários podem pedir listas de filmes de acordo com os temas permitidos pelos pacotes que eles possuem;
Usuários que tentarem filtrar por temas que não são permitidos por eles não devem ver filmes que não estão presentes nos temas que ele não tem acesso;

Parte 5 - Filmes Assistidos

Expor um serviço capaz de armazenar filmes assistidos pelos usuários e emitir relatórios sobre os filmes assistidos;
Antes da inserção deve-se validar se o usuário tem acesso ao filme - Recomendado armazenar os filmes encontrados no serviço externo dentro do mongodb;
Os usuários devem ser capazes de marcar e desmarcar filmes vistos;
Deve existir um endpoint capaz de emitir um relatório com as seguintes informações dos usuários:

```
userId:number|string,
totalFilmsWatched:number,
mostWatchedTheme: {
    themeId:string|number,
    themeName:string,
    totalFilmsWatched: number
},
lastFilmeWatched: {
    movieId: number | string,
```

```
    movieName: string
}
```

Complementos

Interface de Paginação esperada:

```
const DEFAULT_PAGE_SIZE = 10;
export interface PaginationType<T>{
    data:Array<T>
    pagination:{
        pageNumber:number,
        pageSize:number,
        totalPages:number,
        lastPage:boolean,
        firstPage:boolean
    }
}
```

MovieDb

Documentação api: https://developer.themoviedb.org/reference

Entrega

- Repositório com o código fonte
- Estrutura dos bancos de dados
- Migrations (se necessário)
- Scripts para preparar e rodar o projeto (preferencialmente um docker compose)