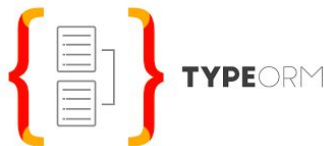




NodeJS + TypeScript + Express + TypeORM + MySQL



Tecnólogo em Análise e Desenvolvimento de
Sistemas - Faculdades SENAC/PE

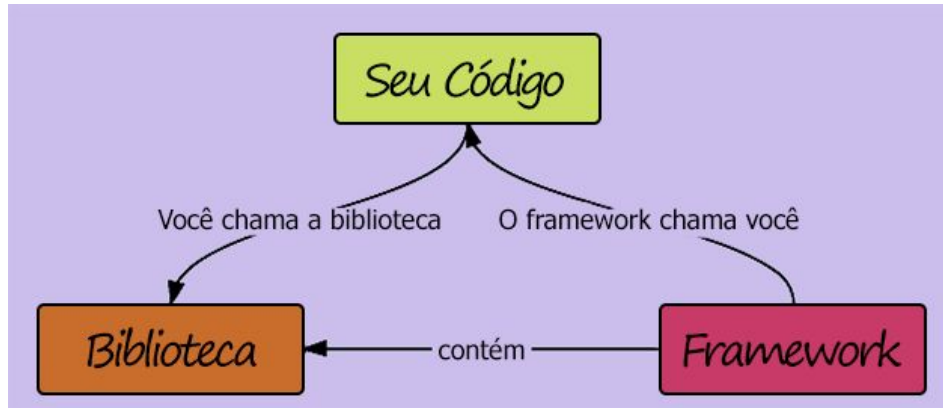


Professor: Danilo Farias

- Conhecer as tecnologias para API backend.
- Entender sobre a arquitetura MVC.
- Configurar um projeto do NodeJS + TypeScript + Express + TypeORM + MySQL no Visual Studio Code.
- Implementar API usando as tecnologias propostas.
- Testar as rotas com Postman.

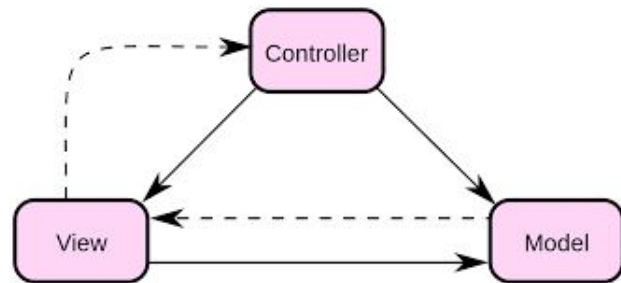
O que é um Framework?

- Framework é um termo em inglês que significa **estrutura**.
- Já em Computação, Framework é um conjunto de códigos genéricos capazes de ajudar na construção de um projeto de Software.

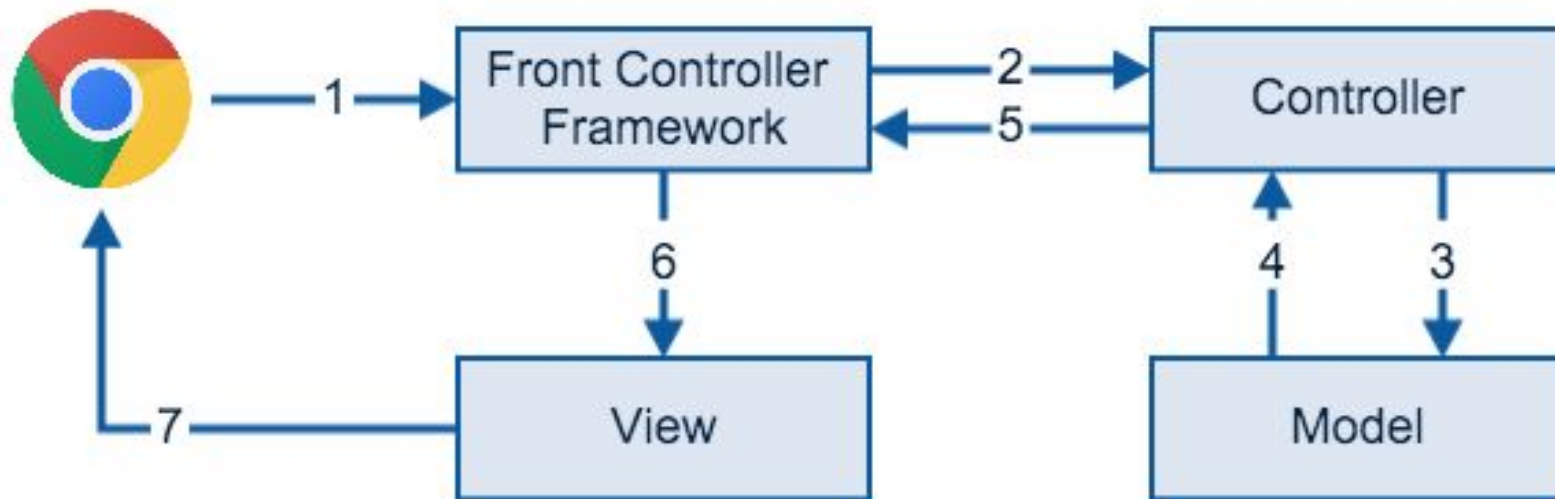


Entendendo a arquitetura MVC

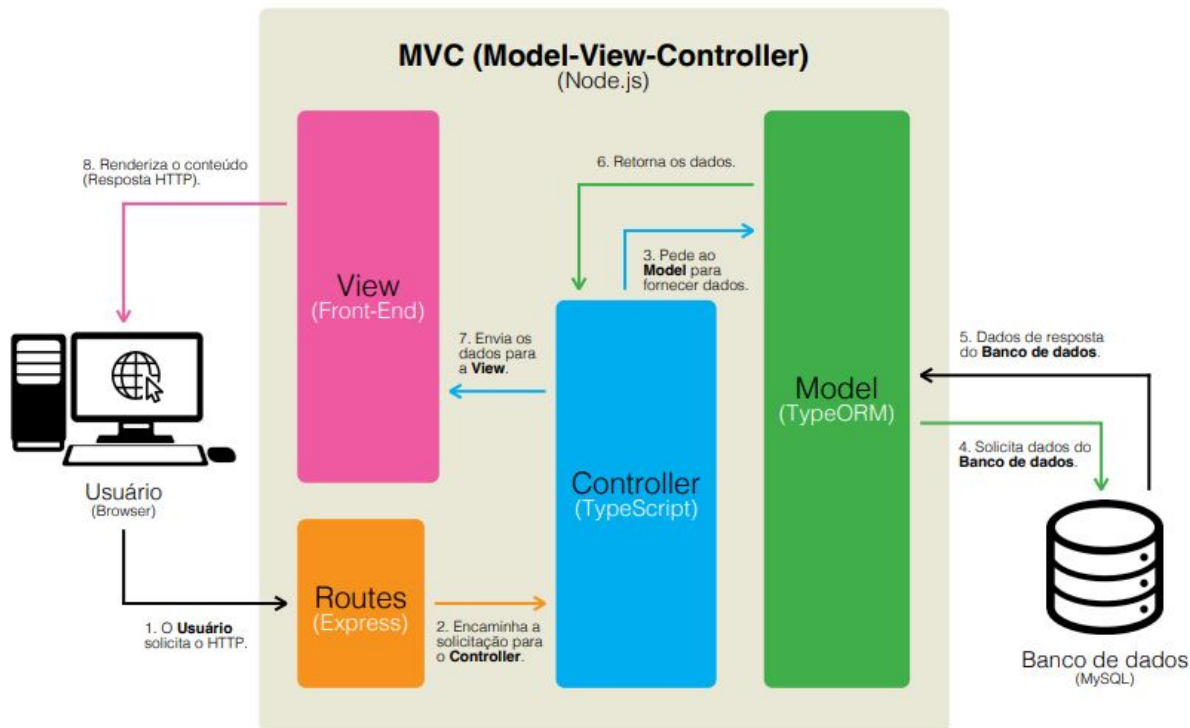
- **Model-View-Controller** (MVC) é um padrão de projeto de software, ou padrão de arquitetura de software focado no reuso de código e a separação de conceitos em três camadas interconectadas.
- Trazendo uma apresentação dos dados e interação dos usuários (front-end), separado dos métodos que interagem com o banco de dados (back-end)



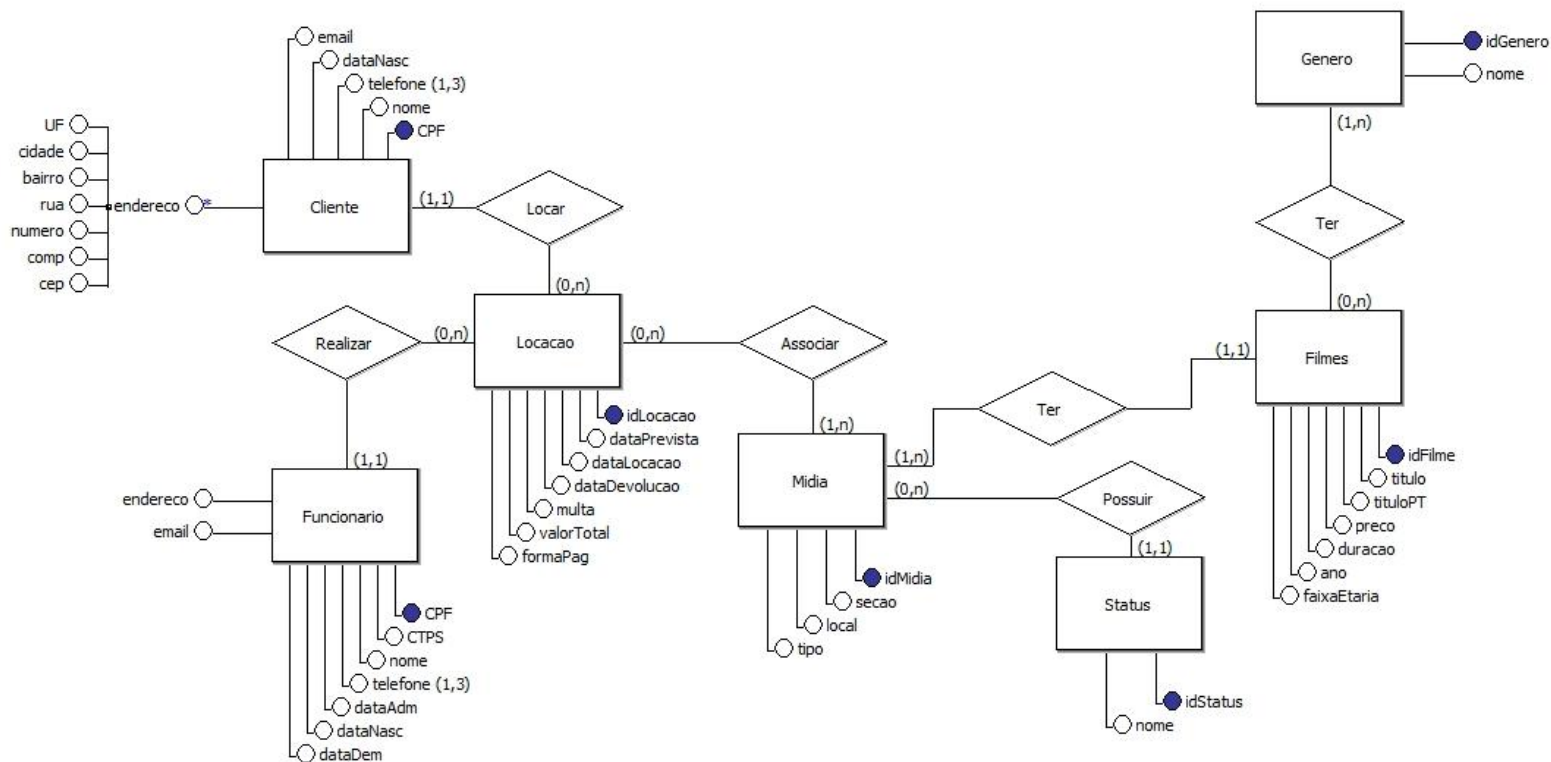
Entendendo a arquitetura MVC



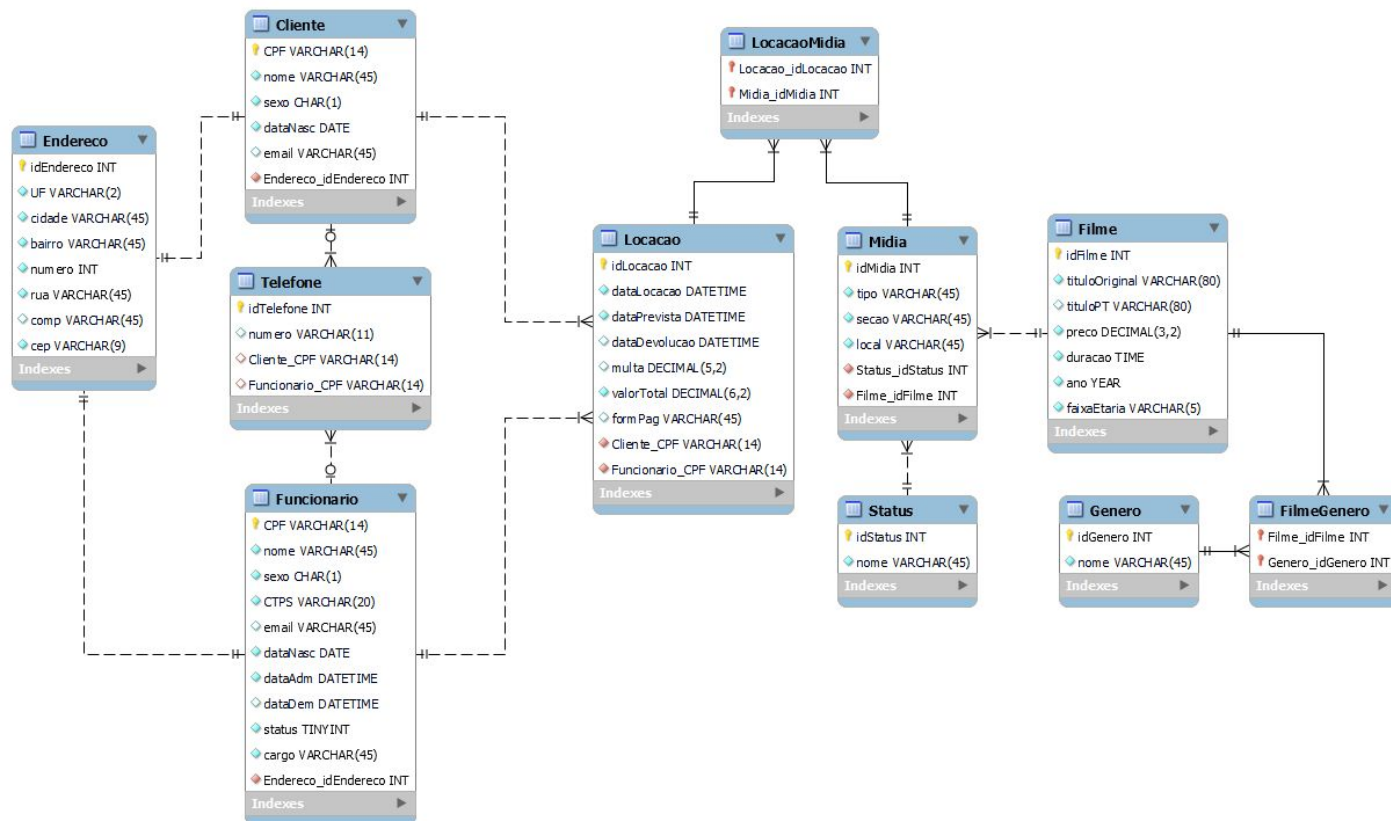
Entendendo a arquitetura MVC



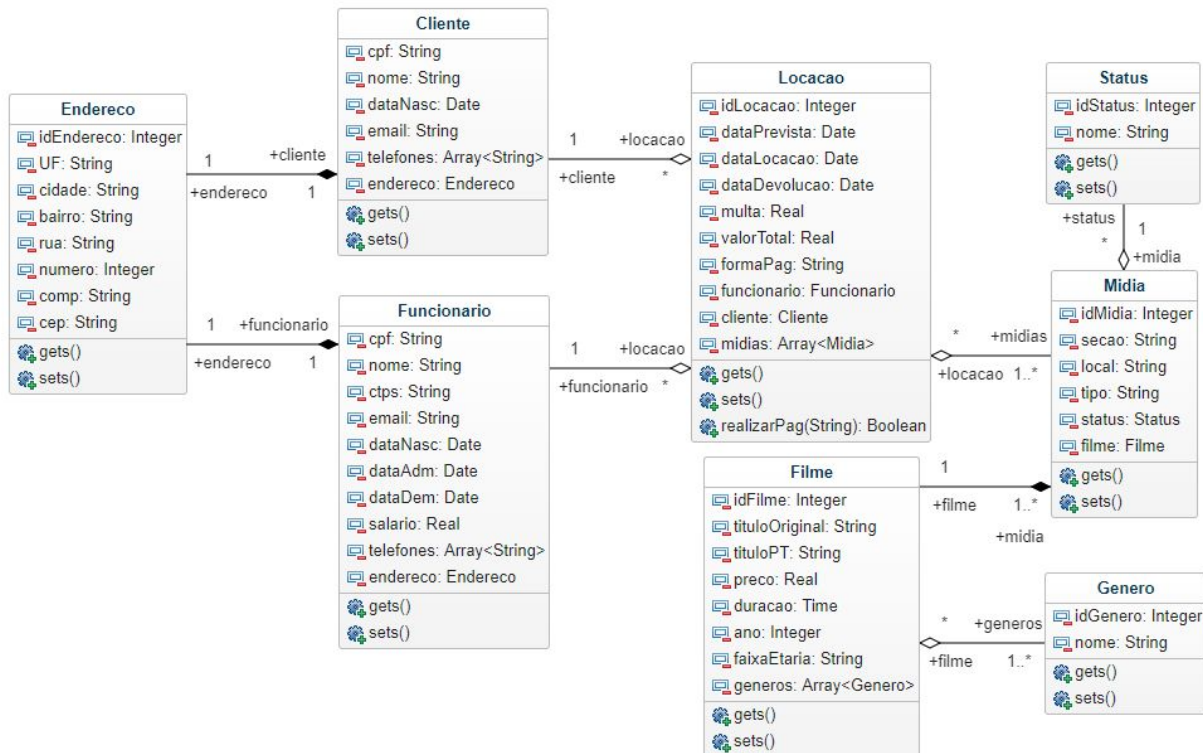
Entendendo o nosso negócio



Entendendo o nosso negócio



Entendendo o nosso negócio



Ambiente de trabalho

NodeJS



- O **Node.js**, uma biblioteca do JavaScript para aplicativos do lado do servidor, não é necessário para aprender o TypeScript. Mas, ao instalá-lo, você também obtém o Gerenciador de Pacotes do Node, também chamado de **npm**, o gerenciador de pacotes da linguagem JavaScript. Em seguida, você usará o npm para instalar o pacote TypeScript.
 - Vá para a página de [Downloads do Node.js](#).



- O TypeScript está disponível como um pacote no registro npm como typescript.
- Para instalar a versão mais recente do TypeScript:
 - Na janela do Prompt de Comando, insira
 - **npm init -y**
 - **npm install -G typescript**
 - **npm i -D tsx**
 - Insira **tsc** para verificar se o TypeScript está instalado. Se ele tiver sido instalado com êxito, esse comando mostrará uma lista de comandos e opções do compilador.

- Nem todo código **TypeScript** é código **JavaScript**.
- O TypeScript adiciona uma nova sintaxe ao JavaScript, o que torna o JavaScript mais fácil de ler e implementa alguns recursos, como a digitação estática.
- Embora o código TypeScript torne o desenvolvimento mais fácil e menos propenso a erros, os navegadores e a maioria dos outros runtimes não dão suporte nativo a TypeScript. Por esse motivo, o TypeScript requer uma etapa de build ([transcompilador](#)) para transformá-lo em JavaScript para que o aplicativo funcione.

- Teremos que criar um projeto TypeScript com o NodeJS para isso:
 - **npx tsc --init**
 - Crie na raiz do projeto a pasta **src**
 - No tsconfig atualizar:
 - No inicio inserir --> "include": ["./src"]
 - No "compilerOptions" descomentar a linha --> "outDir":
"./build"
 - **npx gitignore node**
 - **npm i -D ts-node-dev**

- No projeto TypeScript com o NodeJS, por fim vamos fazer esse ultimo passo:
 - Ajustar o package.json em "scripts" para:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "dev": "ts-node-dev src/index.ts",  
  "start": "node build/index.js",  
  "build": "rimraf .build && tsc"  
}
```

- Agora vamos criar um arquivo **index.ts** na raiz do **src**
- Vamos escrever um “Hello Word” com TypeScript
 - No arquivo index.ts, escreva: `console.log("Olá Mundo!");`
- Por fim, no terminal execute o código TypeScript para sua compilação e execução usando o comando:
 - Modo de Desenvolvimento: **npm run dev**
 - Build do Projeto: **npm run build**
 - Executar o Build: **npm start**

Ambiente de trabalho

TypeORM



- TypeORM é um ORM que pode ser executado em plataformas NodeJS, Browser, Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo e Electron e pode ser usado com TypeScript e JavaScript (ES2021).
- Seu objetivo é oferecer suporte aos recursos mais recentes do JavaScript e fornecer recursos adicionais que ajudem você a desenvolver qualquer tipo de aplicativo que use bancos de dados de pequenos aplicativos com algumas tabelas a aplicativos corporativos de grande escala com vários bancos de dados.

Ambiente de trabalho

MySQL



- O MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface. É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo.
 - Documentação do MySQL:
<https://dev.mysql.com/doc/refman/8.4/en/>



Ambiente de trabalho

Instalando o TypeORM e MySQL



- No terminal execute os códigos npm para instalação do TypeORM e MySQL:
 - **npm install typeorm --save**
 - **npm install reflect-metadata --save**
 - **npm install @types/node --save-dev**
 - **npm install mysql2 --save**

Ambiente de trabalho

Configurando o acesso ao MySQL



- Crie a pasta **config**;
- Depois crie o arquivo **db.config.ts**
- No arquivo **db.config.ts** insira o código ao lado:

```
export const config = {  
  HOST: "localhost",  
  PORT: 3306,  
  USER: "root",  
  PASSWORD: "root",  
  DB: "alohaFilmes",  
  pool: {  
    max: 5,  
    min: 0,  
    acquire: 30000,  
    idle: 10000  
  }  
};
```

```
export const dialect = "mysql";
```

Ambiente de trabalho

Configurando o acesso ao MySQL



- Crie a pasta **db**;
- Depois crie o arquivo **data-source.ts**
- No arquivo **data-source.ts** insira o código ao lado:

```
import "reflect-metadata"
import { DataSource } from "typeorm"
import { config, dialect } from
"../config/db.config"
import { Genero } from "../models/genero"

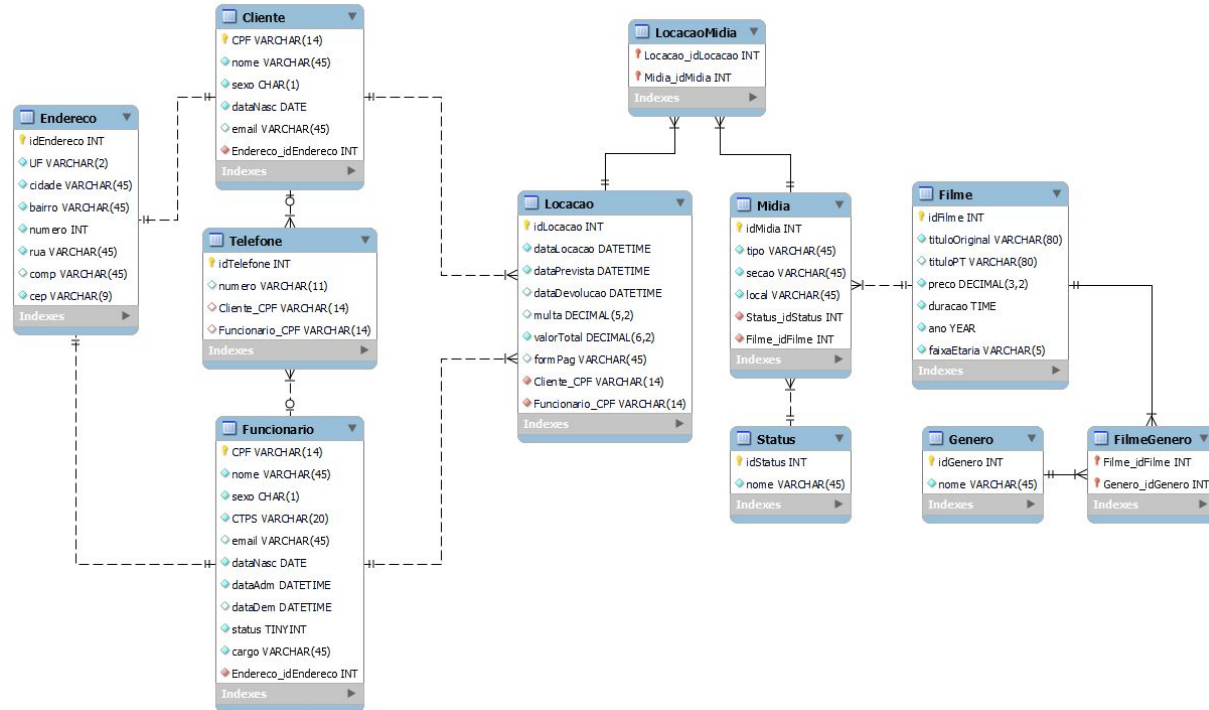
export const AppDataSource = new DataSource({
  type: dialect,
  host: config.HOST,
  port: config.PORT,
  username: config.USER,
  password: config.PASSWORD,
  database: config.DB,
  entities: [Genero],
  synchronize: true,
  logging: false,
})
```

Ambiente de trabalho

Criando nossas classes model

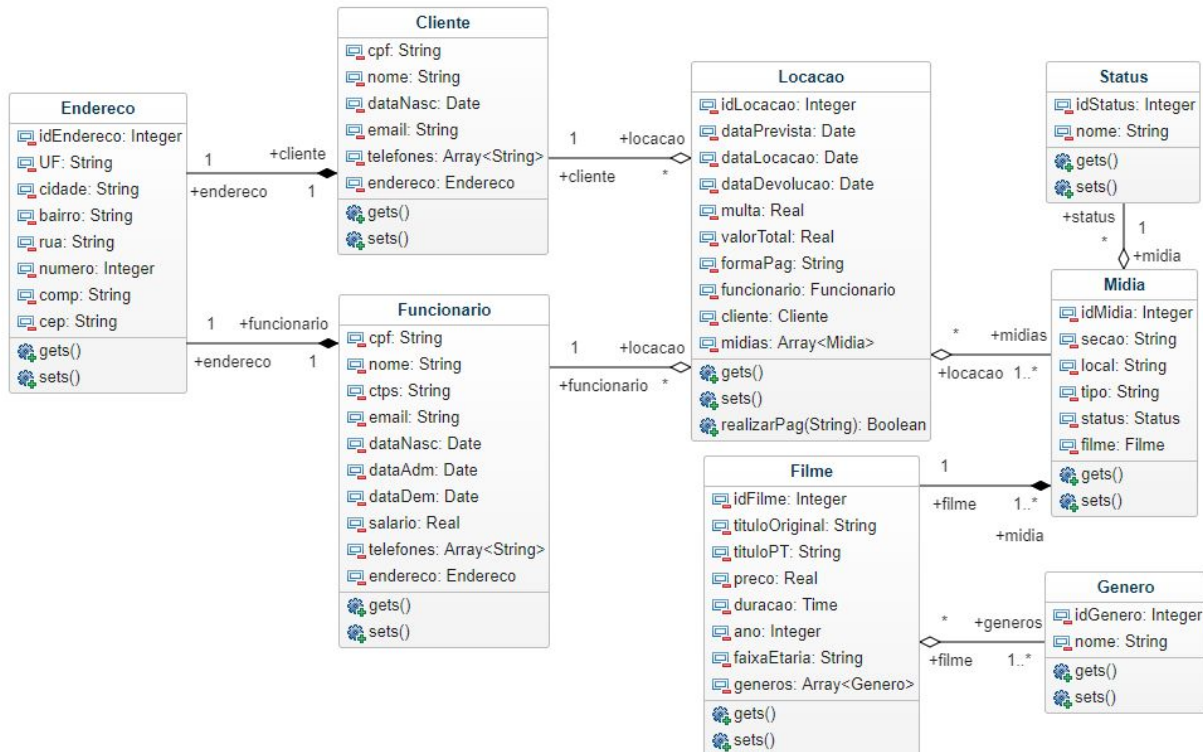


- Crie a pasta **model**;
- Depois vamos solicitar ao **ChatGPT** que crie os arquivos model do nosso projeto da locadora de acordo com nosso MER.



Ambiente de trabalho

Criando nossas classes model



Dúvidas?

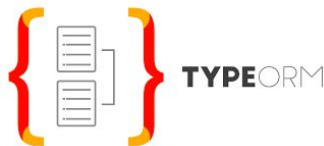




NodeJS + TypeScript + Express + TypeORM + MySQL



Tecnólogo em Análise e Desenvolvimento de
Sistemas - Faculdades SENAC/PE



Professor: Danilo Farias