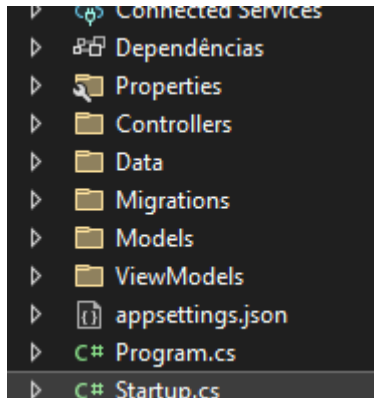


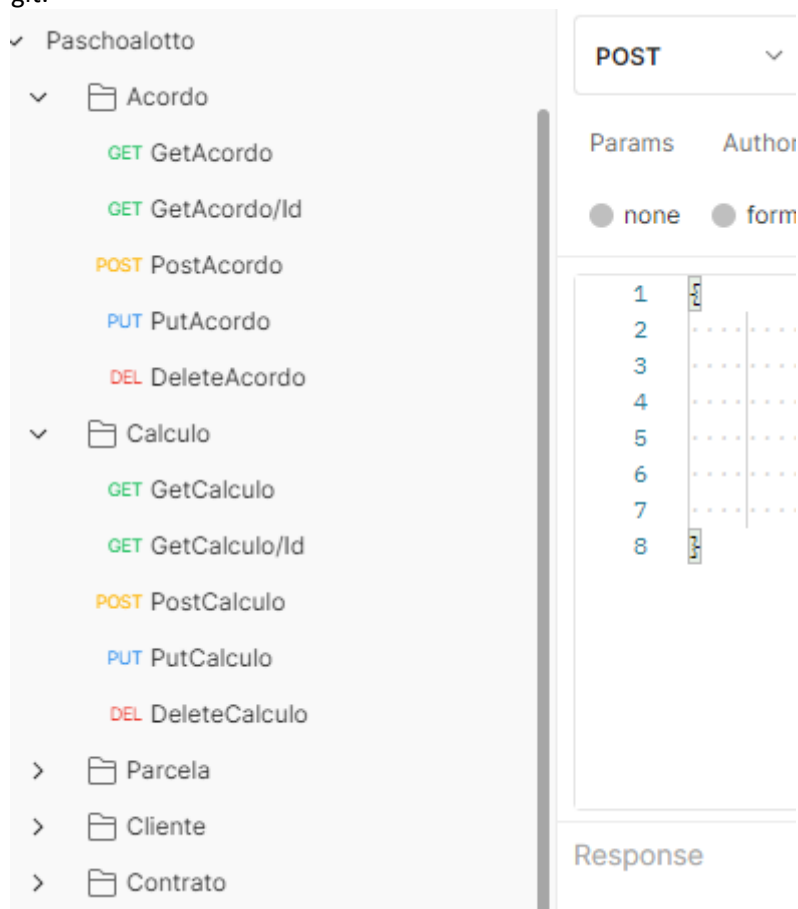
Inicialmente a intenção foi trabalhar com front e backend, porém ao decorrer do caminho me deparei com diversos problemas que me tomaram muito mais tempo do que o previsto, impedindo assim a finalização, mesmo assim estou enviando o que consegui fazer.

Criei um CRUD no backend , padrão web (mvc) para ser consumida no front e somente após o manuseio do cliente, e então aplicar a regra para salvar os dados e mostrar.

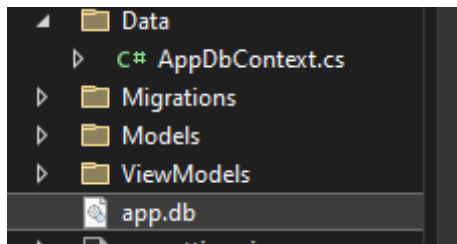
Estrutura do Backend



Para validação das rotas foi utilizado o Postman, que deixarei a biblioteca disponível junto no git.

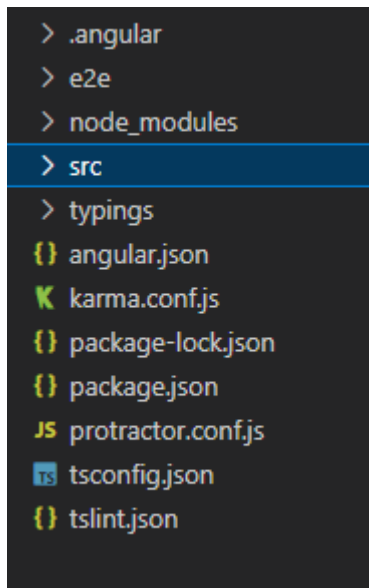


Estou trabalhando com um banco de dados sqlite que é um banco leve e a apropriado para pequenos projetos, em um modelo relacional.



Para o front usei Angular um layout com base no creative tim que é gratuito(<https://www.creative-tim.com/>) com typescrip

Estrutura do Frontend



Onde utilizamos o padrão de interpolação (double mustache) para o manuseio dos dados do backend.

```

<mat-form-field class="example-full-width">
  <input matInput placeholder="#{{NumeroContrato}}" disabled>
</mat-form-field>
</div>
<div class="col-md-3">
  <mat-form-field appearance="legacy">
    <mat-label>Nome</mat-label>
    <input matInput disabled placeholder="" name="name" value="{{NomeUser}}">
  </mat-form-field>
</div>
<div class="col-md-3">
  <mat-form-field appearance="legacy">
    <mat-label>Data vencimento</mat-label>
    <input matInput disabled placeholder="" name="venc" value="{{DataVencCont}}">
  </mat-form-field>
</div>
<div class="col-md-3">
  <mat-form-field appearance="legacy">
    <mat-label>Valor R$ </mat-label>
    <input matInput disabled placeholder="" name="valor" value="{{ValorContrato}}">
  </mat-form-field>
</div>

```

```

> app > negociacao > TS negociacao.component.ts
1  import { Component, OnInit } from
2
3  @Component({
4    selector: 'app-negociacao',
5    templateUrl: './negociacao.component.html',
6    styleUrls: ['./negociacao.component.css'],
7  })
8  export class NegociacaoProfileComponent implements OnInit {
9
10   constructor() { }
11
12   ngOnInit() {
13   }
14
15   public NumeroContrato='101010';
16   public NomeUser='Caique Lopes';
17   public ValorContrato='110,55';
18   public DataVencCont='01/03/2019';
19
20 }
21

```

Sendo com isso conseguimos trabalhar o retorno do backend.

Tecnologias utilizadas:

FRONT-----
BOOTSTRAP 4 ANGULAR 13

BACK-----
.NetFramework 5.0