

Programação III

Home Managing WebApp

Ema Guedes, nº 11280.

Manuel Morais, nº 11337

2º ano de Multimédia

Instituto Superior Miguel Torga

Para testar (No Postman porque o Android não está ligada à base de dados, nem está completo)

Como Proprietário:

- Id do proprietário: **1**
- Senha: **pro1**

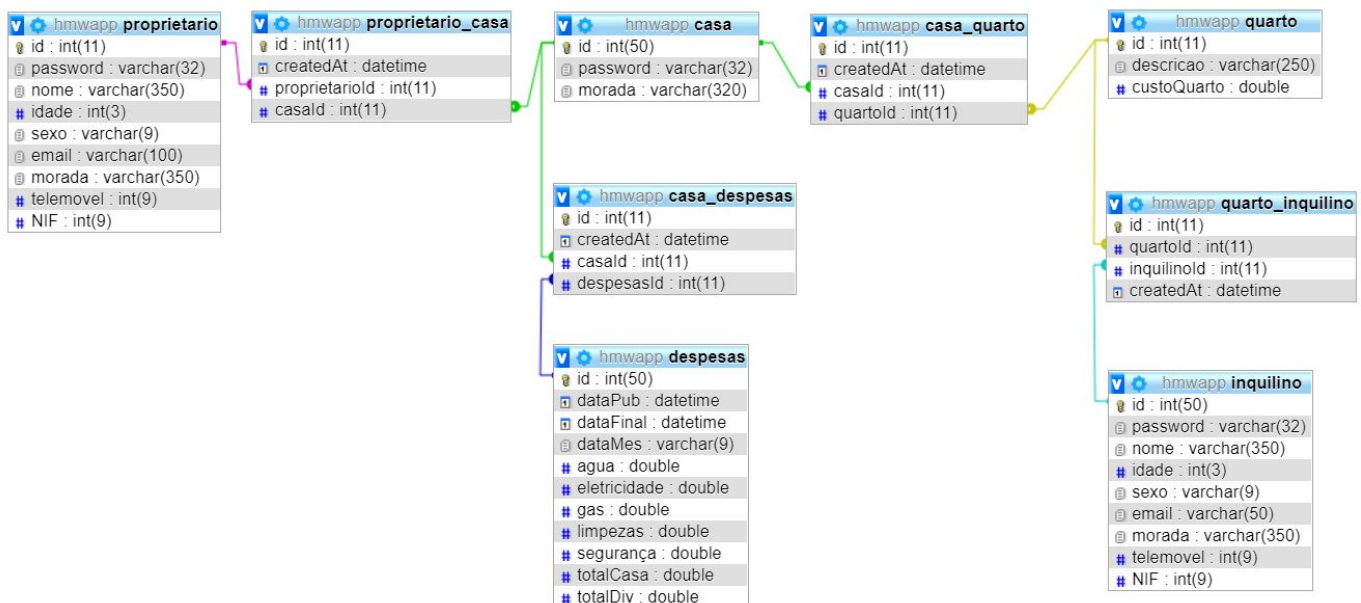
Como Inquilino:

- Id do inquilino: **1**
- Senha: **inq1**

Tema

A nossa aplicação tem como função ajudar os proprietários de residências partilhadas na gestão das despesas. Os inquilinos dessas residências, também podem aceder ao site, estando sempre notificados das despesas que têm de pagar ou verem o histórico de todas as despesas associadas a essa residência tal como uma lista de nomes de todas as pessoas que lá habitam. Se quiserem, têm acesso às informações do administrador para que o consigam contactar ou fazer uma transferência bancária.

Base de dados



Os proprietários têm casas, que têm quartos. As casas têm despesas e quartos. Os quartos têm inquilinos.

Eu não consegui criar a tabela “despesas_inquilino” em que pensava meter “totalDiv” e “pagamento”(para confirmar pagamento), porque me dava erro e não fazia sentido quando criava as chaves forasteiras.

Design

Nós começámos por fazer o design em Adobe XD, mas tivemos problemas a passar para android, por isso tivemos de optar por um design diferente. Ainda assim, disponibilizamos o documento XD para que possa ter uma noção do design inicialmente projectado para a aplicação.

Escolhemos diferentes azuis, pois esta cor simboliza confiança, cooperatividade, competência e alta qualidade. Queremos que quem utilize a nossa aplicação, se sinta seguro e que acredite na nossa competência.

Código

Usamos todos os métodos usados em aula: GET, POST, PUT e DELETE.

O GET para buscar valores à base de dados(ex: buscar informação de um inquilino), o POST para inserir valores à base de dados(ex: para criar um inquilino); PUT para atualizar valores na base de dados(ex: para mudar a idade de um inquilino) e DELETE para eliminar valores da base de dados(ex: para eliminar um quarto).

```
src > config > JS database.js > ...
1  const sequelize = require('sequelize');
2  const ligacao = new sequelize('hmwapp', 'root', '', {
3    host: 'localhost',
4    dialect: 'mysql'
5  });
6  module.exports = ligacao;
```

Este código é o que permite ligar a base de dados à aplicação. O « 'hmwapp' » é o nome da nossa base de dados.

Os ficheiros estão organizados de acordo com as normas dadas pelo professor.

Bibliografia

<https://sequelize.org/v3/docs/models-definition/> - para saber as definições dos models. Depois aplicadas nos ficheiros da pasta models.