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**

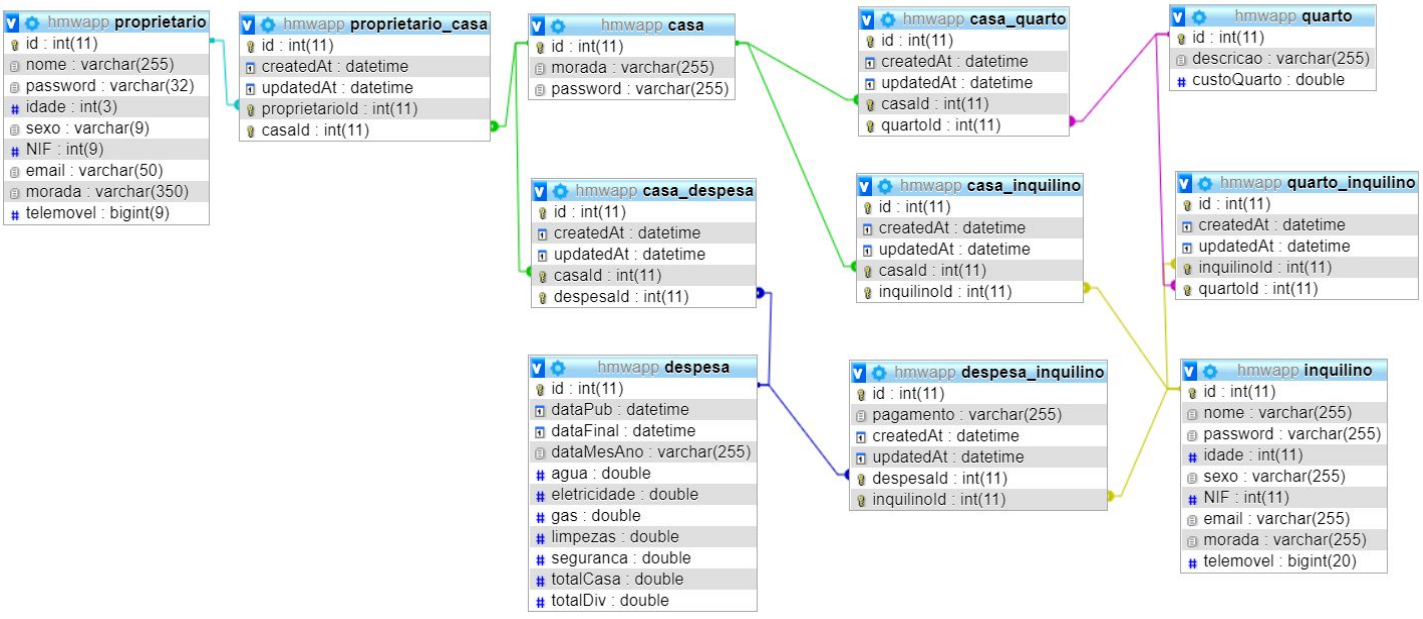
Como **Inquilino:**

- 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



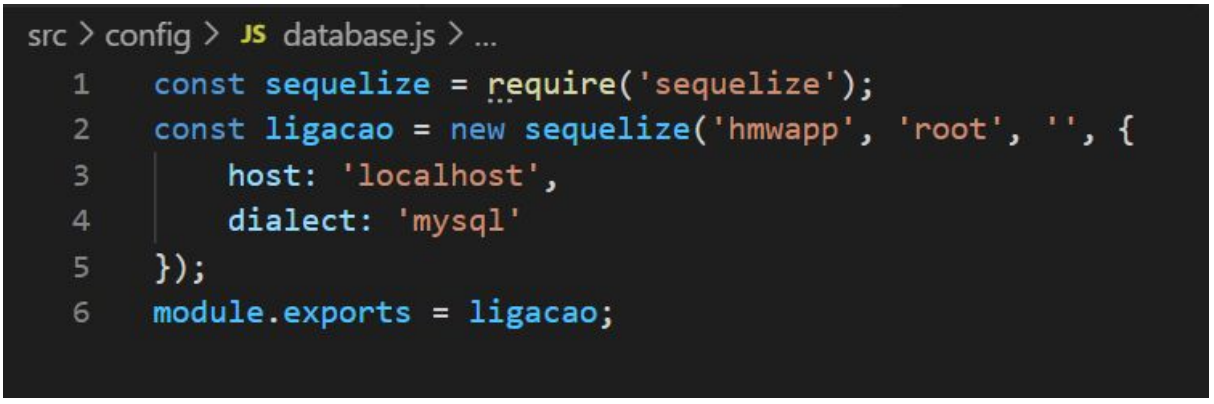
Os proprietários têm casas, que têm casas. As casas têm despesas e quartos. Os quartos têm inquilinos.
Eu não consegui criar a tabela “despesas_inquilino” em que pensava meter “totalDiv” e “pagamento”(para confirmar pagamento), porque me dava erro e não fazia sentido quando criava as chaves forasteiras.

Design

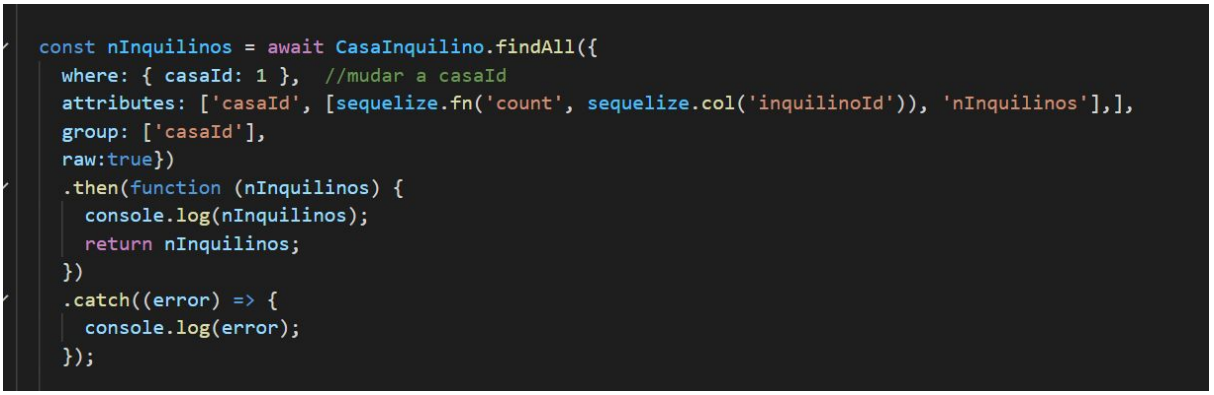
Nós começámos por fazer o design em Adobe XD, mas tivemos problemas a passar para android, por isso tivemos de optar por um design diferente. Ainda assim, disponibilizamos o documento XD para que possa ter uma noção do design inicialmente projectado para a aplicação.
Escolhemos diferentes azuis, pois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



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.



Este é o código para contar o número de inquilinos, neste caso, da casa 1. O “where” tem de ser melhorado.

```
  const totalCasa = await DespesaInquilino.findAll({
    attributes: {
      include: [
        [sequelize.literal(`(SELECT agua+eletricidade+gas+limpezas+seguranca FROM despesa)`),
            'totalDespesa'
        ]
      ]
  },
raw:true})
    .then(function (totalCasa) {
      console.log(totalCasa);
      return totalCasa;
    })
    .catch((error) => {
      console.log(error);
    });
```

Este código, faz a soma da despesa “agua”, “eletricidade”,...

```
let totalDiv = parseFloat(totalCasa[0]["totalDespesa"])/parseInt(nInquilinos[0]["nInquilinos"]);
res.json({
  success: true,
  nInquilinos: nInquilinos,
  despesa: despesa,
  totalCasa: totalCasa,  // ---> não descobri o motivo, mas ele não está a receber os valores. apa
  totalDiv: totalDiv,
});
```

Este código divide o total das despesas pelo número de inquilinos.

Aspetos a melhorar

O nosso primeiro objectivo, neste momento, é corrigir o código e melhorar a base de dados, completar a aplicação no Android Studio, criando as páginas restantes e ligando-a à base de dados. O projecto ainda está muito inacabado para outras melhorias serem mencionadas.
Conseguir com que o código relacionado às contas das despesas funcione, inserindo os resultados, das mesmas à base de dados na respectiva coluna da tabela. Fazer o código para enviar um pedido para uma casa e a confirmação da mesma para a pessoa ser um inquilino dessa residência. Aparecer o estado do pagamento “Por pagar”, e puder atualizá-lo para “Pago”.

Progressão

Começámos pelo Android Studio, mas surgiram complicações no computador de um de nós, que não conseguiram ser resolvidas. Então a outra pessoa ficou encarregue

do Android Studio. A outra parte, com os problemas, ficou encarregue de fazer o código, para assim, podermos entregar o trabalho a tempo.
A maior complicação, foi as contas das despesas, em que precisámos da ajuda do professor.
O estado do pagamento e o pedido para a casa, ficaram por fazer, devido ao tempo.

Conclusão

Infelizmente, o código ficou incompleto e com erros. A aplicação no Android Studio, ainda mais incompleta ficou. O relatório, deixado para a última hora, teve de ser feito à pressa pela pessoa responsável pelo código, que depois, também disponibilizou todas as etapas do trabalho no github.
A maior dificuldade no Android Studio foi conseguir converter o design criado no Adobe XD para o mesmo e ligar à base de dados e ao código.
As aulas atingiram o que foi proposto e o professor mostrou disponibilidade em ajudar e acompanhar os alunos. O real problema neste trabalho foi o trabalho de grupo.

Bibliografia

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