

UNIVERSIDADE SALVADOR

YGOR PINTO GAMA

GUSTAVO RAFAEL VIEIRA GOES

WESLEY WENDEL MELO DOS SANTOS

SERGIO RICARDO SANTOS

CRUD DE DADOS DE UMA ESCOLA

Salvador, BA

2022

SUMÁRIO

1. INTRUDOÇÃO.....	3
2. MODELOS DE PROJETO.....	4
3. DESENVOLVIMENTO COM DISCURSSÃO ACERCA DO CÓDIGO FONTE.....	9
4. RESULTADOS.....	10
5. CONCLUSÕES.....	11
6. BIBLIOGRAFIA.....	12

INTRODUÇÃO

O presente trabalho foi exigido pelo professor Adailton de Jesus, da disciplina de Usabilidade, desenvolvimento web, mobile e jogos, nesse trabalho vai ser demonstrado como foram feitos o front-end, construído em reactjs, e o backend, construído usando nodejs e postgresql, de uma página web. O front-end dessa página vai estar consumindo dados fornecidos por um backend, API REST, fazendo um CRUD, CREATE, DELETE, READ e UPDATE, na aplicação.

MODELOS DE PROJETO

Nessa sessão serão apresentados os modelos do projeto, como por exemplo, a modelagem do banco de dados e as telas de cadastro de turma, entre outros modelos.

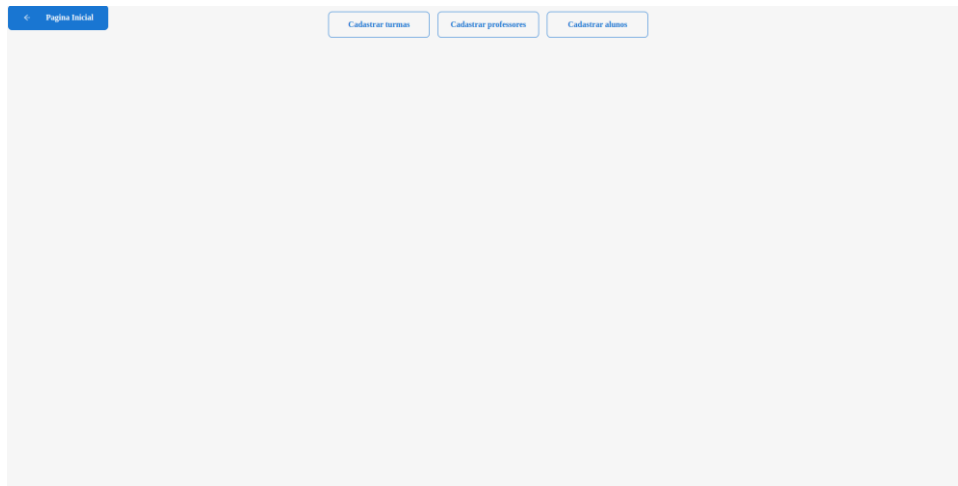
Tela inicial nela o usuário vai escolher pra qual ação ele que ver



Tela que apresenta os dados da escola

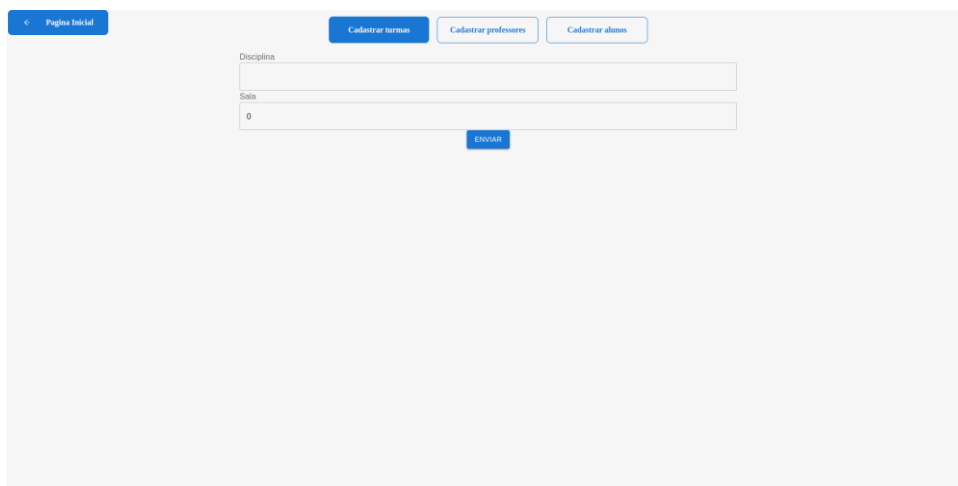
← Retornar		
NOME	CNPJ	ENDEREÇO
CoopRandom	22.307.454/0001-42	R. das Pitangueiras, 42 - Pôrto, Lauro de Freitas - BA, 42700-000

Nessa tela o usuário escolhe qual das informações abaixo ele vai querer cadastrar



The screenshot shows a web interface with a top navigation bar. On the left, there is a blue button labeled "Página Inicial" with a left-pointing arrow. On the right, there are three light blue buttons: "Cadastrar turmas", "Cadastrar professores", and "Cadastrar alunos". The main content area is a large, empty light gray rectangle.

Nessa tela o usuário escolheu cadastrar a turma



The screenshot shows the same top navigation bar as the previous screen. The "Cadastrar turmas" button is now highlighted in dark blue. Below the buttons, there is a form with two input fields. The first field is labeled "Disciplina" and is empty. The second field is labeled "Sala" and contains the number "0". Below these fields is a dark blue button labeled "Enviar".

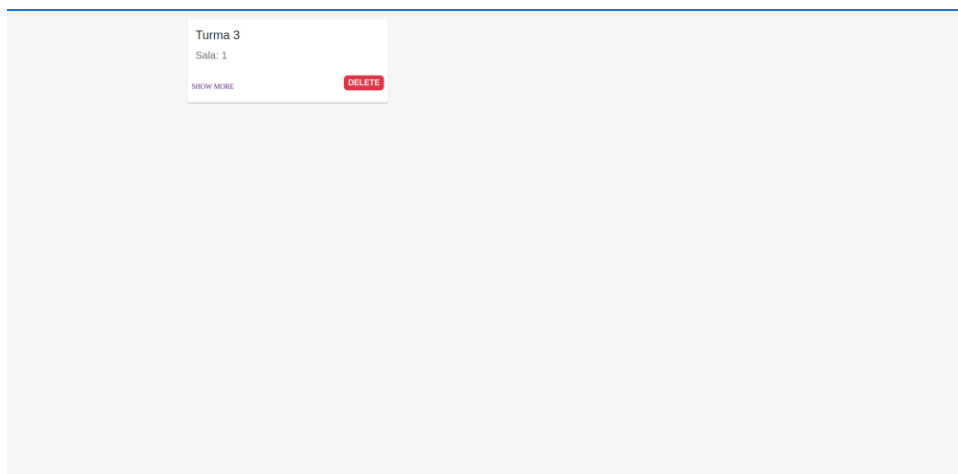
Nessa tela o usuário escolheu cadastrar o professor

The screenshot shows a web interface with a top navigation bar. On the left, there is a blue button labeled "Página Inicial" with a left-pointing arrow. On the right, there are three buttons: "Cadastrar turmas", "Cadastrar professores", and "Cadastrar alunos". The "Cadastrar professores" button is highlighted in blue, while the others are white with blue borders. Below the navigation bar, there is a registration form with the following fields: "Nome" (text input), "Cpf" (text input), "Titulação" (text input), "Disciplina" (dropdown menu), and a blue "ENVIAR" button at the bottom right.

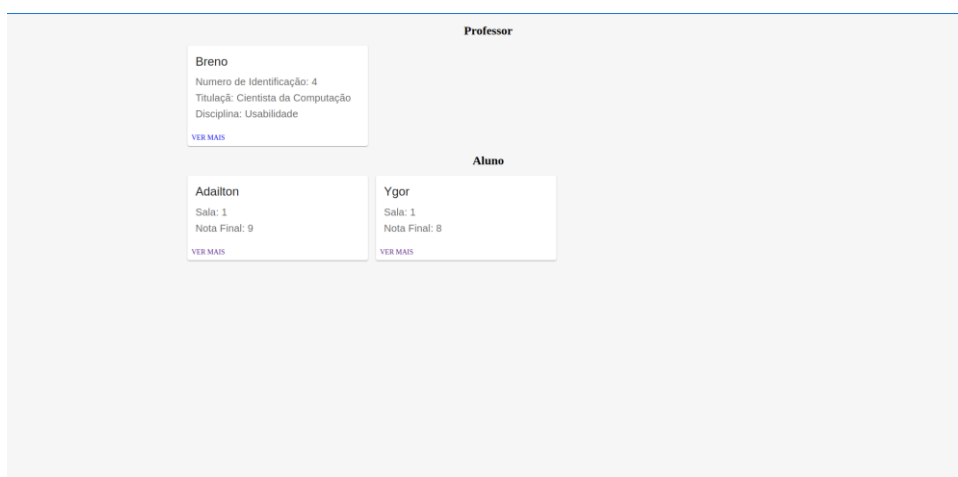
Nessa tela o usuário escolheu cadastrar o aluno

This screenshot shows the same web interface as the first one, but with the "Cadastrar alunos" button highlighted in blue. The "Cadastrar turmas" and "Cadastrar professores" buttons are now white with blue borders. The registration form below remains unchanged, with the "Nome" and "Cpf" fields visible and the "ENVIAR" button at the bottom.

Ao clicar na opção turmas na tela inicial o usuário vai ser levado a tela de turmas



Ao escolher a turma o usuário vai ver os aluno e os professores cadastrados nas turmas



Ao clicar em ver mais em uma das duas telas o usuário vai ser direcionado as informações sobre aquele aluno ou professor.

←

Retornar

Nome	Cpf	Título	Disciplina
Breno	461.457.850-04	Cientista da Computação	Usabilidade

Atualizar título

DELETE

←

Retornar

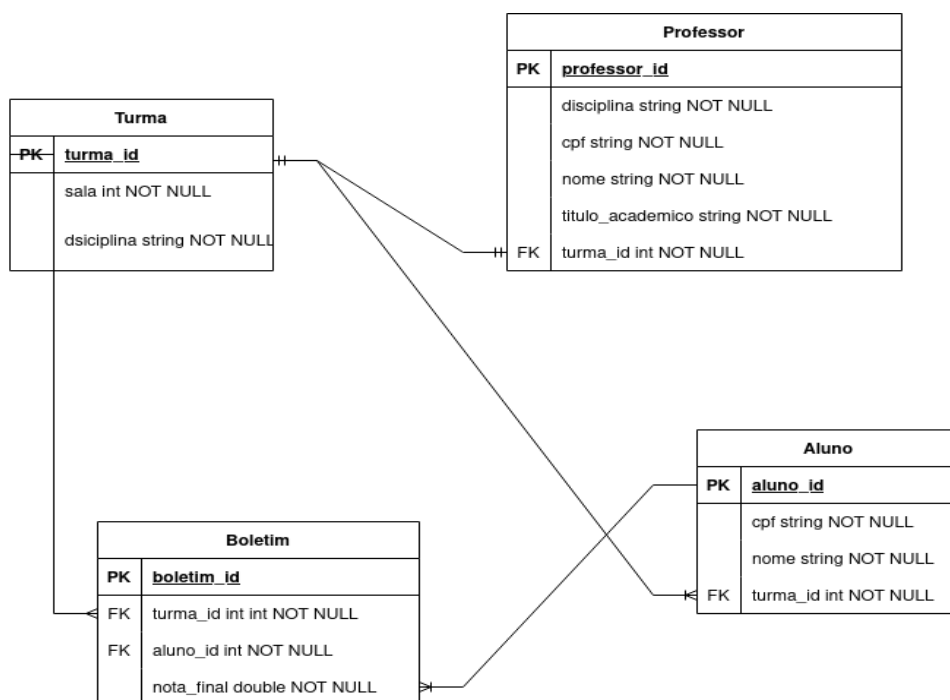
Nome	Cpf	Nota Final	Sala
Vigor	461.457.850-04	8	1

Trocar de sala

Atualizar nota

DELETE

Essa última imagem mostra como ficou a modelagem do banco de dados



Desenvolvimento com discussão acerca do código fonte

O código fonte do projeto desenvolvido foi feito utilizando a linguagem JavaScript, sendo o backend desenvolvido usando a runtime nodejs e para o front-end foi usado a biblioteca reactjs.

Foi também usado como forma de armazenar os dados da aplicação o banco de dados postgresql na localhost, para utilizar o banco de dados na localhost é necessário baixar uma extensão no navegador que vai ser utilizado, pois, as Cross-Origin Request (CORS) estão por padrão bloqueadas, pelo menos nos navegadores que eu usei, a extensão usada para burlar essa proibição foi a CORS Unblock.

O projeto também faz uso de uma ORM para trabalhar com o banco de dados, no caso, a ORM escolhida foi o sequelize, com essa ORM, foi possível simplificar a criação da base de dados, o acesso e os comandos necessários para realizar as exigências do projeto.

Para cada um dos itens exigidos foram criadas, no back-end, rotas que executariam as funções disponibilizadas pelos controladores, neles seria usados as funções disponibilizadas pelo sequelize para realizar o CRUD na aplicação. Essas rotas seriam usadas no front-end da aplicação junto do axios, um dos pacotes instalados, para consumir os dados disponibilizados e modificá-los.

O front-end da aplicação foi feito totalmente com react-js, foi usado também a biblioteca de componentes module ui para agilizar o processo de desenvolvimento do front-end e não ter que construir vários componentes do zero, também foi usado module css para não haver risco de classes do css não ficarem aplicando estilos onde não devem.

A página web também recebeu funções que alertam o usuário se caso algum erro ocorreu na aplicação, por exemplo, um usuário não digitou o nome de um aluno vai então aparecer um alert falando o que aconteceu e o local que aconteceu o erro vai aparecer em vermelho.

Para executar a aplicação de maneira correta na máquina é preciso primeiramente ter o banco de dados postgresql na máquina, depois disso vai ser necessário instalar os node_modules, caso os arquivos tenham sido pegos pelo github, por isso no terminal nos paths dos arquivos de front-end e back-end digite npm install.

Após instaladas as dependências é preciso conferir o seu usuário e senha do postgresql, na pasta config, no arquivo database.js, depois vai ser preciso digitar os comandos npx sequelize db:create e em seguida npx sequelize db:migrate, isso vai criar no seu computador o banco de dados e as tabelas dele, pois foi usado sequelize na aplicação.

Com tudo isso pronto o usuário só precisa inicializar o front-end e o back-end, usando o comando `npm start` no terminal, nos paths das pastas.

RESULTADOS

Como resultados do projeto foi possível realizar o CRUD dos itens, atendendo todos as funcionalidades essenciais, aprovação dos alunos caso a média fosse maior ou igual a 6.

CONCLUSÃO

Ao fim do presente trabalho pode-se concluir que o desenvolvimento de uma aplicação de uma página web, mesmo que a página seja simples, vai ter diversas complexidades, por exemplo, a construção do banco de dados, sem dúvidas a parte do projeto que consumiu mais tempo e recursos, isso se deveu por uma falta de conhecimento inicial de como se deveria seguir com ele, no caso desse projeto foi usando a ORM sequelize.

Com tudo isso dito foi possível aprender novos meios de mexer com uma base dados, e também deu para estabelecer uma base sólida sobre o que é página web e como ela pode ser criada.

REFERÊNCIAS

Sequelize v6 disponível em : <https://sequelize.org/docs/v6/>

Orlandi, Claudio. Configurando o ORM Sequelize no NodeJS com ExpressJS. In: Elias Gabriel. **Rocktseat**. 14 de Junho de 2018. Disponível em: <https://blog.rocktseat.com.br/nodejs-express-sequelize/>

Node.js v18.12.1 documentation. Disponível em: <https://nodejs.org/dist/latest-v18.x/docs/api/>

Introdução. Disponível em: <https://pt-br.reactjs.org/docs/getting-started.html>