

**FACULDADES INTEGRADAS DE BAURU**

CIÊNCIA DA COMPUTAÇÃO

TRABALHO MVC EM NODE E MONGODB

Marcos Leandro da Cruz Barboza

Trabalho de Engenharia de software do curso de Ciência da Computação das Faculdades Integradas de Bauru – FIB.

BAURU - SP

2019

**Sumário**

**1.** **Introdução………………………………………………………………………... 7**

1.1 Objetivos Gerais………………………………………………………………………….7

**2.** **Desenvolvimento………………………………………………………………...7**

2.1 Interface do Sistema……………………………………………………………………...7

2.2.1 Tela de Login……………………………………………………………………….…….7

2.2.2 Tela de Cadastro……………………………………………………………………….....7

2.2.2 Dashboard………………………………………………………………………………...7

**3.** **Tecnologias utilizadas…………………………………………………………..8**

3.1 Node.js…………………………………………………………………………………....8

3.2 React……………………………………………………………………………………....8

3.3 Express…………………………………………………………………………………....8

3.4 MongoDB………………………………………………………………………………...8

3.5 Webpack………………………………………………………………………………....8,

3.6 Babel……………………………………………………………………………………...8,

3.7 Docker…………………………………………………………………………………...8,

3.8 Redux……………………………………………………………………………………..8

**4.** **Referências 10**

**1. Introdução**

Resumidamente pode-se entender como padrão de projeto, como a solução recorrente para um problema em um contexto, mesmo que em projetos e áreas distintas. O mais importante sobre os padrões é que eles são soluções aprovadas. Cada catálogo inclui apenas padrões que foram considerados úteis por diversos desenvolvedores em vários projetos.

Os principais padrões de projeto são:

MVC, Abstract Factory, Factory Method, Singleton, Padrões estruturais, Adapter, Padrões de comportamento, Template Method.

O padrão MVC é dividida em 3 partes, são elas:

Model que é é responsável pela leitura e escrita de dados, e também de suas validações.

View ela apenas faz a exibição dos dados, sendo ela por meio de um html, xml ou jsx.

Controller O responsável por receber todas as requisições do usuário. Seus métodos chamados actions são responsáveis por uma página, controlando qual model usar e qual view será mostrado ao usuário.

**2. Desenvolvimento**

Foi criado uma schema de usuário com os campos abaixo:

nome, email, senha, dataNascimento e telefone.

Foi criado algumas controllers para realizar as operações de inclusão, alteração, exclusão e consulta.

Foi utilizado o docker-compose para estar gerando as imagens do mongodb e mongoExpress.

O projeto view com react e redux foi criado separadamente da API, para facilitar o entendimento e a legibilidade do código.

**2. Interface do sistema**

2.2.1 Tela de login

A tela de login é a primeira interface que o usuário terá acesso, se tiver cadastrado basta informar o e-mail e a senha para realizar o login. Caso ele não esteja cadastrado terá uma opção para se cadastrar no lado do botão Login.

2.2.2 Tela de cadastro

A tela de cadastro contém 5 campos para serem preenchidos, nome, email, senha, data de nascimento e telefone. Depois de realizado todo o processo basta clicar no botão cadastrar e em seguida será redirecionado para a tela de login.

2.2.3 Dashboard

A dashboard é a tela para onde o usuário irá logo após ter realizado o login. Nela ele terá as opções de alterar os dados cadastrais e excluir a conta cadastrada.

**3. Tecnologias utilizadas**

3.1 Node.js

Node.js é uma plataforma construída sobre o motor JavaScript do Google Chrome para facilmente construir aplicações de rede rápidas e escaláveis. Node.js usa um modelo de I/O direcionada a evento não bloqueante que o torna leve e eficiente, ideal para aplicações em tempo real com troca intensa de dados através de dispositivos distribuídos.

3.2 React

O React é uma biblioteca JavaScript de código aberto para criar interfaces de usuário. É mantido pelo Facebook, Instagram e uma comunidade de desenvolvedores individuais e outras empresas.

3.3 Express

Express é uma estrutura de aplicativo da web para o Node.js, lançada como software livre e de código aberto sob a licença MIT. Ele é projetado para construir aplicativos da Web e APIs. Ele foi chamado de estrutura de servidor padrão de fato para o Node.js.

3.4 MongoDB

MongoDB é um software de banco de dados orientado a documentos livre, de código aberto e multiplataforma, escrito na linguagem C++. Classificado como um programa de banco de dados NoSQL, o MongoDB usa documentos semelhantes a JSON com esquemas.

3.5 WebPack

O Webpack é um bundler de módulo JavaScript de código aberto. Seu principal objetivo é agrupar arquivos JavaScript para uso em um navegador, mas também é capaz de transformar, empacotar ou empacotar qualquer recurso ou ativo.

3.6 Babel

O Babel é um compilador JavaScript gratuito e de código aberto e transpiler configurável usado no desenvolvimento web.

3.7 Docker

Docker é uma tecnologia de software que fornece contêineres, promovido pela empresa Docker, Inc. O Docker fornece uma camada adicional de abstração e automação de virtualização de nível de sistema operacional no Windows e no Linux.

3.8 Redux

Redux é uma biblioteca JavaScript de código aberto para gerenciar o estado do aplicativo.

**Referências**

[**Redux**](https://tableless.com.br/bem-vindo-ao-redux/)[**Docker**](https://www.mundodocker.com.br/o-que-e-docker/)[**React**](https://tableless.com.br/react-o-que-e-e-como-funciona-essa-ferramenta/)[**Express**](https://expressjs.com/pt-br/)[**Babel**](https://desenvolvedor.expert/o-que-eh-es6-66c8d7631a0b)[**WebPack**](https://medium.com/rocketseat/entendendo-e-dominando-o-webpack-4b2e8b3e02da)[**MongoDB**](https://blog.umbler.com/br/boas-praticas-com-mongodb/)[**Node.js**](https://nodejs.org/en/)