

## Relatório sobre o Trabalho com Webpack e Vue.js

### 1. Passos para Executar o Projeto

#### 1.1 Configuração Inicial

Para iniciar o projeto, é necessário criar uma estrutura básica de arquivos e diretórios. O projeto é dividido em duas partes principais: frontend e backend.

#### 1.2 Instalação das Dependências

Foram utilizadas as seguintes dependências no projeto:

1. Webpack: Para empacotar e agrupar os módulos JavaScript do frontend.
2. Vue.js: Um framework JavaScript progressivo para construir interfaces de usuário interativas.
3. Vue Loader: Um loader para Webpack que permite carregar arquivos Vue no projeto.
4. Babel Loader: Um loader para Webpack que permite transpilar o código JavaScript moderno para uma versão compatível com navegadores mais antigos.
5. Express: Um framework para criação de servidores web em Node.js.

Todas essas dependências foram instaladas usando o npm (Node Package Manager) e registradas no arquivo `package.json`.

### 2. Configuração do Webpack

O Webpack foi configurado no arquivo `webpack.config.js`. Neste arquivo, definimos a entrada (`entry`) e a saída (`output`) do processo de empacotamento. Além disso, foram adicionadas regras de carregamento (loaders) para processar arquivos JavaScript usando o Babel Loader e arquivos Vue usando o Vue Loader.

### 3. Configuração do Vue.js

O Vue.js foi configurado no arquivo `main.js`, que é o ponto de entrada da aplicação Vue. Neste arquivo, importamos e configuramos o Vue, definimos o componente principal e montamos o aplicativo Vue na página.

### 4. Implementação do Frontend:

O frontend foi desenvolvido utilizando Vue.js e consiste em um componente Vue chamado `App.vue`. Neste componente, foi criada uma aplicação simples que exibe um contador de tempo e se comunica com o backend para obter informações.

### 5. Implementação do Backend

O backend foi implementado usando o framework Express em conjunto com o Node.js. O servidor Express foi configurado no arquivo `server.js`. Neste arquivo, definimos as rotas para lidar com as requisições do frontend.

## 6. Descrição dos Arquivos do Projeto

### 6.1. package.json

O arquivo `package.json` é um arquivo de configuração do projeto Node.js. Ele lista todas as dependências do projeto, bem como scripts que podem ser executados para diferentes fins, como iniciar o servidor, empacotar o frontend, entre outros. Além disso, contém metadados sobre o projeto, como o nome, versão e autor.

### 6.2. package-lock.json

O arquivo `package-lock.json` é gerado automaticamente pelo npm e serve para garantir que todas as dependências do projeto sejam instaladas com as mesmas versões em todos os ambientes. Ele é usado para garantir a reprodutibilidade das instalações de pacotes em diferentes máquinas.

### 6.3. webpack.config.js

O arquivo `webpack.config.js` é responsável por configurar o Webpack, uma ferramenta de empacotamento de módulos JavaScript. Neste arquivo, são definidas as entradas, saídas e regras de carregamento (loaders) para processar os diferentes tipos de arquivos do projeto.

### 6.4. babelrc

O arquivo `.babelrc` é um arquivo de configuração para o Babel, uma ferramenta de transposição de código JavaScript. Neste arquivo, são especificadas as configurações para transpilar o código moderno do JavaScript para uma versão compatível com navegadores mais antigos.

### 6.5. src/main.js

O arquivo `main.js` é o ponto de entrada da aplicação Vue.js. Neste arquivo, é importado e configurado o Vue, definido o componente principal (`App.vue`) e montada a aplicação Vue na página.

### 6.6. src/App.vue

O arquivo `App.vue` é o componente principal da aplicação Vue.js. Ele contém a estrutura do template HTML, a lógica do script JavaScript e o estilo CSS. Neste componente, é criada a aplicação Vue que exibe um contador de tempo e faz requisições para o servidor Express.

#### 6.7. src/data.json:

O arquivo `data.json` contém dados fictícios utilizados para simular informações obtidas do servidor. Neste caso, ele contém uma estrutura de objeto com informações do tempo.

#### 6.8. server.js

O arquivo `server.js` é responsável por configurar o servidor Express. Neste arquivo, são definidas as rotas para lidar com as requisições do frontend e é criado um servidor web que escuta na porta 3000.

#### 6.9. node\_modules

O diretório `node\_modules/` é criado após a instalação das dependências do projeto. Ele contém todas as bibliotecas e pacotes externos que foram instalados pelo npm.

#### 6.10. dist

O diretório `dist/` é criado após a execução do Webpack e contém o resultado final do empacotamento do frontend. Ele contém o arquivo `bundle.js`, que é o código JavaScript final que será carregado pelo navegador.

#### 6.11. index.html:

O arquivo `index.html` é a página principal da aplicação. Ele contém o elemento `

` com o id `app`, onde a aplicação Vue será montada pelo `main.js`. Além disso, é neste arquivo que o `bundle.js` é carregado para o frontend da aplicação.

#### 6.12. style.css

O arquivo `style.css` é um arquivo de estilo global para a aplicação. Ele contém estilos CSS, com o bootstrap que serão aplicados a toda a página.

### 7. Executando o Projeto

Para executar o projeto, primeiro, é necessário instalar todas as dependências. Isso pode ser feito executando o seguinte comando no terminal:

```
npm install  
npx webpack --mode development  
node server.js  
abrir `http://localhost:3000`.
```