

AVALIAÇÃO PRÁTICA

Unidade Curricular: Tecnologias Para Desenvolvimento de Sistemas (2025)

Professor: Leonardo Medeiros

VALOR: máximo de 100 PONTOS

RELATÓRIO - AP04 - IMPLEMENTAÇÃO DE APLICAÇÃO WEB COM NODE.JS E VUE.JS

Alunos: Gustavo Chiossi, Gabriel Senkovski

Repositório:

O trabalho foi desenvolvido via [WSL2](#).

Este relatório descreve o desenvolvimento de uma aplicação web de gestão de tarefas (To-Do List), desenvolvida como parte da avaliação prática da disciplina de Tecnologias Para Desenvolvimento de Sistemas. O objetivo principal foi a aplicação de conceitos fundamentais de desenvolvimento frontend moderno, utilizando o framework progressivo Vue.js e a biblioteca de estilos Bootstrap. Primeira etapa: instalação das tecnologias

Instalar NVM:

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | bash
$ nvm install --lts
```

Para verificar se a instalação deu certo:

```
$ npm -v
$ node -v
```

Instalar Bootstrap:

```
$ npm install bootstrap
```

Desenvolvimento da aplicação

O esqueleto do projeto foi iniciado com o comando `npm create vue@latest`. Após isso, não foi adicionada nenhuma feature (tecla enter), e foi iniciado um projeto

em branco. Depois, foram usados os comandos `cd application`, `npm install` e `npm run dev`. É possível acessar a aplicação via navegador:

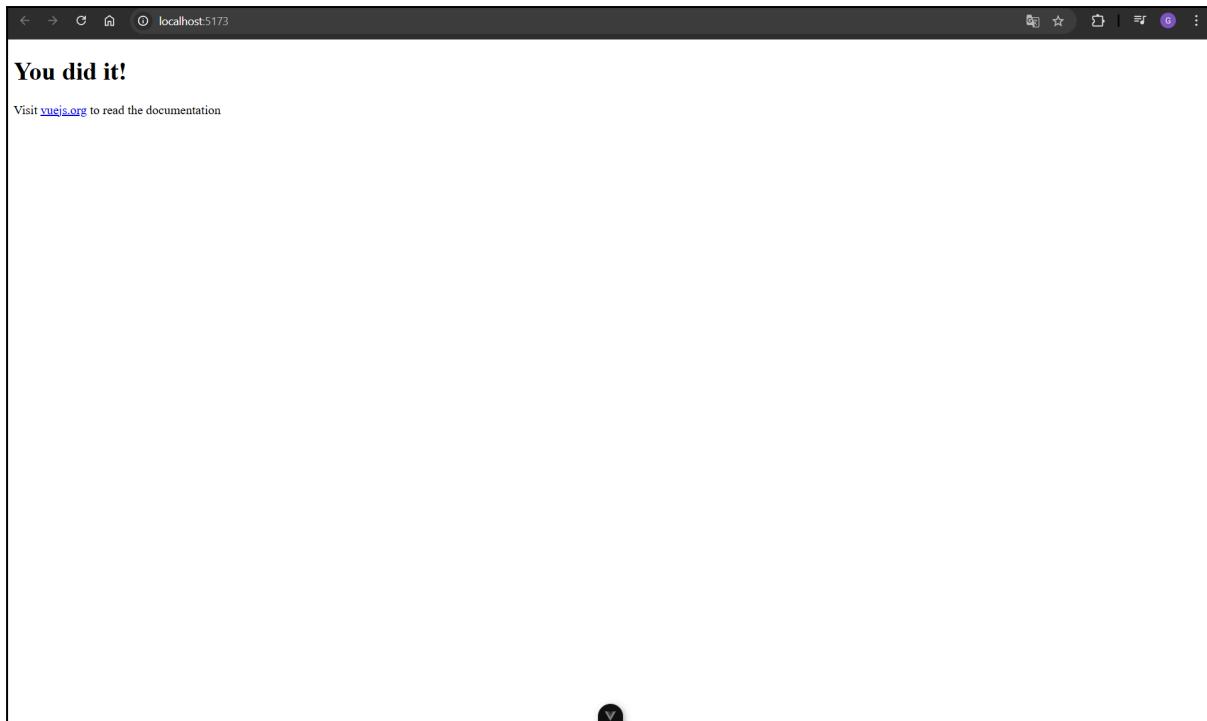


Figura 1: teste inicial da aplicação.

Em `~/application/src/` foi criado a pasta `components/`, e dentro dela, criado o arquivo `ListaDeTarefas.vue`. Nesse arquivo, a lógica do código foi a seguinte:

```
<template>
  <div class="container mt-5">
    <h2 class="mb-3">Lista de Tarefas</h2>

    <div class="input-group mb-3">
      <input
        v-model="novaTarefa"
        type="text"
        class="form-control"
        placeholder="Digite uma tarefa"
      />
      <button class="btn btn-primary" @click="adicionarTarefa">
        Adicionar
      </button>
    </div>

    <ul class="list-group">
      <li>
```

```

    v-for="(tarefa, index) in tarefas"
    :key="index"
    class="list-group-item d-flex justify-content-between align-items-center"
  >
  {{ tarefa }}
  <button class="btn btn-sm btn-danger" @click="removerTarefa(index)">
    Remover
  </button>
</li>
</ul>
</div>
</template>

<script>
export default {
  name: 'TodoList',
  data() {
    return {
      novaTarefa: '',
      tarefas: []
    }
  },
  methods: {
    adicionarTarefa() {
      if (this.novaTarefa.trim() !== '') {
        this.tarefas.push(this.novaTarefa)
        this.novaTarefa = ''
      }
    },
    removerTarefa(index) {
      this.tarefas.splice(index, 1)
    }
  }
}
</script>

<style scoped>
h2 {
  text-align: center;
}
</style>

```

No arquivo `App.vue`, foi usado o seguinte código para conectar o componente `ListaDeTarefas.vue`:

```

<template>
  <ListaDeTarefas />
</template>

<script>
import ListaDeTarefas from './components/ListaDeTarefas.vue'

export default {
  components: {
    ListaDeTarefas
  }
}
</script>

```

A estrutura apresentada utiliza o framework **Vue.js** e o **Bootstrap** para implementar uma aplicação de gerenciamento de tarefas fundamentada no conceito de reatividade, onde o **Node.js** provê o ambiente necessário para a gestão de dependências e a compilação do projeto. A interface é controlada por diretivas como *v-model*, que estabelece o sincronismo bidirecional de dados entre o campo de entrada e o estado *novaTarefa*, enquanto as funções *adicionarTarefa* e *removerTarefa* manipulam o array *tarefas* por meio de eventos *@click*; simultaneamente, a diretiva *v-for* garante a renderização dinâmica e a atualização automática do DOM conforme a modificação dos dados, encapsulando toda a lógica, estilização e comportamento em um componente modular e reutilizável.

Para usar o bootstrap, no arquivo *main.js*, foi utilizado o seguinte código:

```

import { createApp } from 'vue'
import App from './App.vue'
import 'bootstrap/dist/css/bootstrap.min.css'

createApp(App).mount('#app')

```

Este trecho de código configura a inicialização da aplicação, onde o *Node.js* gerencia a resolução de módulos e a importação de dependências globais. Por meio da função *createApp*, o componente raiz *App.vue* é instanciado e os estilos do *Bootstrap* são aplicados, finalizando com o método *mount* que acopla a estrutura reativa ao elemento *#app* do DOM.

Após isso, foi utilizado o comando *npm run dev* para verificar a funcionalidade da aplicação.

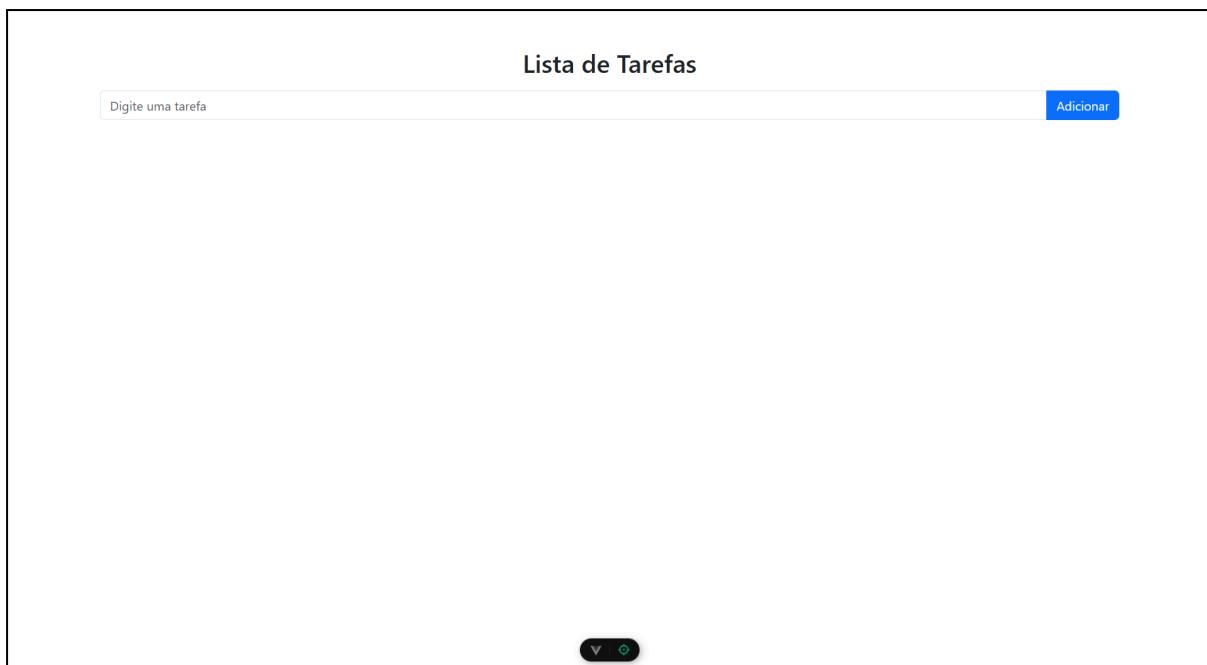


Figura 2: aplicação finalizada e funcionando.

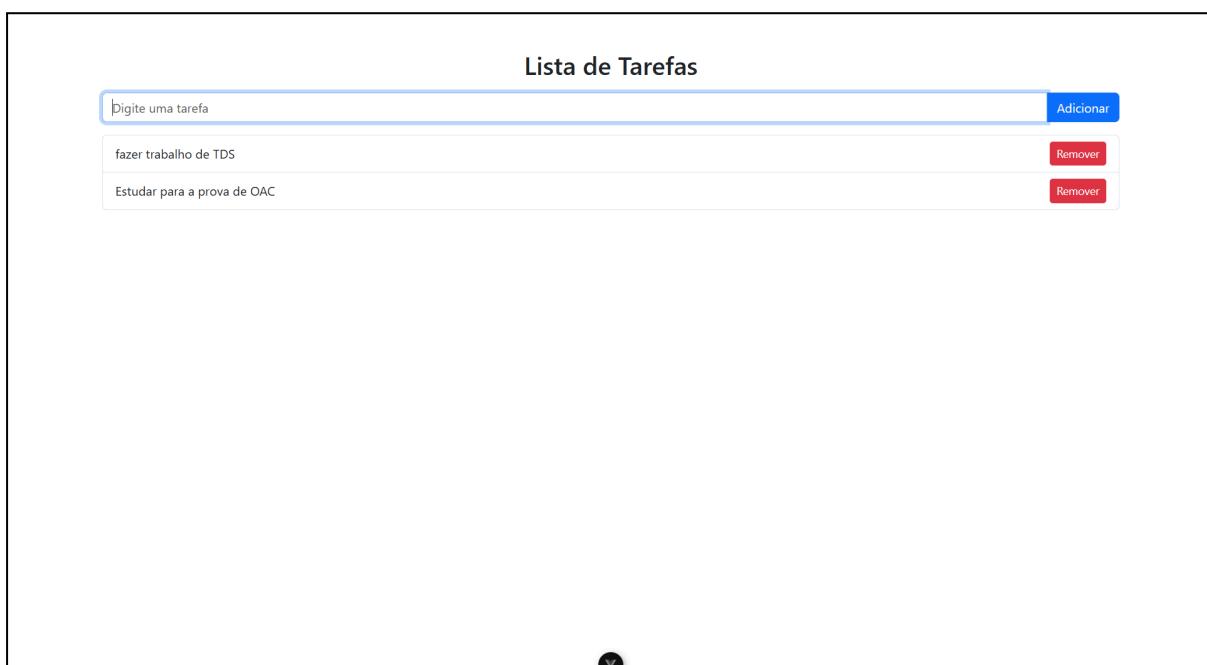


Figura 3: adição de duas tarefas na lista.



Figura 4: remoção de uma das tarefas da lista.

Por fim, a execução deste trabalho permitiu consolidar conhecimentos sobre o ecossistema Node.js e a agilidade proporcionada pelo Vue.js na criação de interfaces dinâmicas. O ambiente de desenvolvimento configurado via WSL2 demonstrou ser eficiente para a gestão de pacotes via NPM e execução do servidor de desenvolvimento Vite, resultando numa aplicação funcional que atende aos requisitos propostos para a interface do utilizador.