

Gabriel Gotardo Rosa

Igor Gustavo Matiello

VUE.JS

O que é?

É um framework JavaScript open source para a criação de aplicações web no front end. No lado do back-end, é comum encontrar implementações em tecnologias como Java, Node.js, Nest.js, entre outras..

O Vue.js possibilita criar aplicações de forma reativa. O Vue.js faz a utilização de um DOM virtual, o que faz com que seja extremamente performático na maioria das situações. Além disso, conta com uma arquitetura muito bem estruturada por meio da criação de componentes reusáveis.

Este framework é muito utilizado para criar aplicações SPA (Single Page Applications) e também para desenvolver vários outros tipos de interfaces, com foco na interação e experiência do usuário. Atualmente e não à toa, o Vue.js está entre os frameworks Javascript para criação de interfaces mais populares do mundo, devido à sua baixa curva de aprendizado, sua versatilidade e por oferecer uma solução bem completa – contando até mesmo com uma CLI, o Vue CLI.

página oficial com documentação

exemplos, como dar start, componentes, tutoriais, historia, quem usa

<https://vuejs.org/examples/#hello-world>

Vantagens e Desvantagens

Vantagens:

Acessível: com um conhecimento básico em HTML, CSS e JavaScript, você já consegue construir uma aplicação básica.

O Vue.js foi projetado para ser fácil de aprender. Mesmo que você esteja começando, se já tem uma noção de HTML, CSS e JavaScript, o Vue torna possível criar uma aplicação funcional de forma rápida.

A curva de aprendizado é suave porque o Vue permite que você vá usando seus conhecimentos atuais de web development sem precisar aprender muitos conceitos novos de uma só vez. Não há uma barreira enorme para começar, como em alguns outros frameworks mais complexos.

Versátil: Possui um ecossistema incremental e progressivo, oferecendo soluções prontas e padronizadas para a grande maioria das situações.

O Vue pode começar pequeno em um projeto e, à medida que o projeto cresce, você vai adicionando mais recursos do próprio framework ou ferramentas da comunidade. Isso significa que ele é adequado tanto para projetos simples quanto para os mais complexos.

O Vue oferece uma flexibilidade gradual, ou seja, você não precisa usar tudo de uma vez. Conforme suas necessidades aumentam, o Vue tem bibliotecas e pacotes adicionais que você pode usar, como o Vue Router (para navegação entre páginas) e o Vuex (para gerenciamento de estado).

Excelente desempenho: O Vue.js é rápido e leve, garantindo uma boa performance mesmo em aplicações mais robustas. Sua arquitetura foi feita pensando em otimização.

Como o Vue é um framework eficiente, ele faz renderizações rápidas de páginas e atualizações visuais com pouca sobrecarga. Isso significa que os usuários percebem uma resposta rápida e fluida, mesmo que a aplicação seja complexa.

Progressivo: Se você já tiver uma aplicação server-side, você pode utilizar o Vue em apenas uma parte da aplicação, utilizando inclusive somente os módulos necessários.

Ao contrário de outros frameworks, como Angular, o Vue pode ser adicionado a uma parte específica de um site já existente, sem precisar reformular toda a aplicação.

Isso é ótimo para quem quer começar a modernizar um site sem refazer tudo do zero. Se você quiser usar o Vue em apenas uma funcionalidade (por exemplo, um formulário), você pode, sem precisar adotar o framework em toda a aplicação.

Comunidade ativa e crescente: Apesar de ser mais jovem que o React ou Angular, o Vue tem uma comunidade muito ativa que contribui com soluções, pacotes e documentação.

Isso significa que, se você tiver um problema, provavelmente encontrará ajuda online, além de tutoriais e guias detalhados feitos pela própria comunidade. O suporte é forte e, como o Vue é popular, novos recursos e melhorias estão sendo desenvolvidos constantemente.

Documentação oficial clara e amigável: A equipe do Vue investe muito na documentação oficial, que é reconhecida como uma das melhores em frameworks JavaScript.

Isso facilita muito o aprendizado, já que a documentação é bem organizada e tem exemplos práticos, ajudando desenvolvedores de todos os níveis a entenderem como usar o Vue de maneira eficiente.

Desvantagens

Complexidade do paradigma reativo: Este framework "leva a sério" os princípios da programação reativa, o que pode trazer um pouco de complexidade no começo.

O Vue usa um modelo de reatividade para atualizar a interface de usuário automaticamente quando os dados mudam. Isso pode parecer confuso no início, porque esse "paradigma reativo" não é muito intuitivo para quem está acostumado com programação mais linear.

Programação reativa significa que as mudanças nos dados automaticamente "reagem" na interface do usuário, sem precisar dizer diretamente como atualizar cada coisa. O desafio é entender como e quando essas reações acontecem. Para quem não está acostumado com essa lógica, pode parecer mágico ou difícil de controlar.

Excesso de flexibilidade: o Vue.js é extremamente flexível e não muito opinativo, o que pode trazer problemas de falta de padronização de código em projetos grandes. O Vue permite que você escreva código de várias maneiras diferentes, o que é bom para flexibilidade, mas pode ser um problema quando muitas pessoas trabalham no mesmo projeto. Sem um guia de estilo ou convenções claras, cada desenvolvedor pode criar seu código de maneira própria.

Recursos mais limitados em relação a outros frameworks SPA: embora o Vue.js seja querido e conte com uma comunidade ativa, ele ainda não tem a mesma quantidade de recursos (como plugins) que seus concorrentes, Angular e React. Isso significa que, se você está trabalhando em um projeto que precisa de uma ferramenta ou funcionalidade muito específica, pode ser mais difícil encontrar uma solução pronta no Vue. Em comparação, no React, é mais provável que já exista um plugin ou biblioteca para resolver esse problema, devido ao seu tempo de maturação e tamanho da comunidade.

Menor suporte de grandes empresas.

O Vue.js não tem o apoio direto de gigantes da tecnologia como Google (que apoia o Angular) ou Facebook/Meta (que apoia o React), o que pode fazer algumas empresas grandes hesitarem em adotá-lo.

Isso não significa que o Vue não seja confiável, mas o fato de não ter um "patrocinador" tão grande como o Google ou o Meta faz com que algumas corporações prefiram usar frameworks com suporte corporativo mais garantido a longo prazo.

Os 10 principais provedores de hospedagem Vue.js

O Vue.js é um framework front-end que, geralmente, gera aplicativos estáticos que podem ser hospedados em qualquer serviço de hospedagem de conteúdo estático ou em um servidor web mais tradicional. Aqui estão algumas opções de provedores populares para hospedar projetos em Vue.js:

1. Netlify

- **Descrição:** Plataforma especializada em sites estáticos e aplicativos JAMstack, oferece integração contínua (CI/CD), HTTPS gratuito, e deploy fácil a partir de repositórios Git (como GitHub, GitLab e Bitbucket).
- **Vantagens:** Deploy rápido, CDN integrada, builds automáticos a partir de commits.
- **Ideal para:** Pequenos e médios projetos de Vue.js, SPAs (Single Page Applications).

2. Vercel

- Descrição: Plataforma que oferece hospedagem de aplicações estáticas e dinâmicas, com suporte excelente para frameworks como Vue.js, React e Next.js.
- Vantagens: CI/CD automático, integração com Git, otimizações de performance como serverless functions, CDN integrada.
- Ideal para: Projetos Vue.js de qualquer porte, com especial foco em projetos JAMstack.

3. GitHub Pages

- Descrição: Serviço de hospedagem estática gratuito da GitHub, ideal para hospedar repositórios de projetos Vue.js que geram páginas estáticas.
- Vantagens: Totalmente gratuito, fácil integração com repositórios GitHub.
- Ideal para: Pequenos projetos estáticos ou de código aberto.

4. Firebase Hosting

- Descrição: Parte da plataforma Google Firebase, fornece hospedagem para sites estáticos, com suporte para apps Vue.js e integração com outros serviços da Google Cloud.
- Vantagens: CDN integrada, suporte a HTTPS, integração com banco de dados em tempo real e funções serverless.
- Ideal para: Aplicações Vue.js que precisam de backend escalável ou serviços em tempo real.

5. AWS Amplify

- Descrição: Serviço da Amazon que oferece uma plataforma completa para desenvolvimento e hospedagem de aplicativos web front-end (como Vue.js) e mobile.
- Vantagens: Suporte a deploy contínuo, HTTPS, integração com outros serviços da AWS, como S3 e Lambda.
- Ideal para: Projetos Vue.js que precisam de uma infraestrutura escalável e integração com backend.

6. Heroku

- Descrição: Plataforma como serviço (PaaS) que permite o deploy fácil de apps Vue.js (especialmente com backends Node.js) com suporte a buildpacks e integração com Git.
- Vantagens: Facilidade de deploy de aplicações full-stack (frontend + backend), suporte a Node.js.
- Ideal para: Projetos Vue.js full-stack ou apps dinâmicos que exigem backend.

7. DigitalOcean App Platform

- Descrição: Plataforma da DigitalOcean que permite o deploy rápido de apps web. Oferece suporte a estáticos e backends como Node.js, HTTPS e auto-scaling.
- Vantagens: Integração com Git, deploy rápido e fácil, escalabilidade automática.
- Ideal para: Projetos Vue.js que precisam de um ambiente mais customizado ou de backend dedicado.

8. Surge

- Descrição: Plataforma de hospedagem estática simples e rápida, especializada em sites estáticos e aplicações como Vue.js.
- Vantagens: Super simples de usar, deploy via CLI, gratuito para pequenos projetos.
- Ideal para: Pequenos projetos Vue.js que não precisam de backends robustos.

9. Render

- Descrição: Plataforma de hospedagem que oferece suporte a apps estáticos e dinâmicos. Suporta frameworks como Vue.js e Node.js.
- Vantagens: Deploy simple e HTTPS gratuito.
- Ideal para: Pequenos e médios projetos de Vue.js.

10. Azure Static Web Apps

- Descrição: Serviço de hospedagem da Microsoft que oferece uma solução dedicada para aplicativos estáticos.
- Vantagens: Deploy contínuo, SSL gratuito, integração com Azure Functions para backend.
- Ideal para: Aplicações Vue.js que podem se beneficiar do ecossistema Azure.

Esses provedores variam em termos de complexidade, custo e recursos, e a escolha do melhor dependerá das necessidades específicas do seu projeto, como escalabilidade, integração com backends, e facilidade de deploy.

Fonte: <https://blog.back4app.com/pt/hospedagem-vue-js/>

<https://www.netlify.com/docs>

<https://vercel.com/docs>

<https://firebase.google.com/docs/hosting>

<https://aws.amazon.com/amplify/>

<https://docs.github.com/en/pages>

Configurações necessárias para rodar uma aplicação

Para rodar uma aplicação Vue.js, é necessário configurar o ambiente de desenvolvimento com os seguintes itens:

- Node.js: É uma plataforma de desenvolvimento backend com JavaScript. Recomenda-se instalar a versão LTS (Long Term Support), que é a mais estável.
- NPM: É um gerenciador de pacotes do Node.js que permite instalar os pacotes necessários para o desenvolvimento.
- Vue CLI: É uma ferramenta de linha de comando para criar projetos Vue.js. Para instalá-la, basta abrir o terminal ou o prompt de comando e digitar o comando `npm install -g @vue/cli`.

Além disso, é possível utilizar outros itens para o desenvolvimento, como:

- Editor de código-fonte, como o Visual Studio Code
- Plugin Vetur
- Extensões para o navegador, como VueDevtools e DejaVue
- Terminal do PC

Tipo de licença de software e responsáveis pelo desenvolvimento (proprietário ou comunidade).

Vue.js é um framework de JavaScript **open source** (código aberto). Ele foi criado por Evan You e lançado pela primeira vez em 2014. Por ser open source, o código-fonte do Vue.js está disponível publicamente, e desenvolvedores de todo o mundo podem contribuir para o projeto ou utilizá-lo livremente, sujeito aos termos da licença MIT sob a qual ele é distribuído.

- Suas conclusões sobre o uso do framework (facilidade para encontrar materiais, qualidade desses materiais; se é de fácil configuração, etc.)

Referencias:

<https://learn.microsoft.com/pt-br/windows/dev-environment/javascript/vue-beginners-tutorial>

<https://blog.back4app.com/pt/hospedagem-vue-js/>

<https://www.dio.me/articles/angular-react-ou-vue-ECQRXD>

<https://www.treinaweb.com.br/blog/o-que-e-o-vue-js>

REPOSITÓRIO DE BASE:

<https://github.com/leonardogbxv/vuejs-crud/tree/master/client/src/components>

npm install INSTALAÇÃO DO NODE

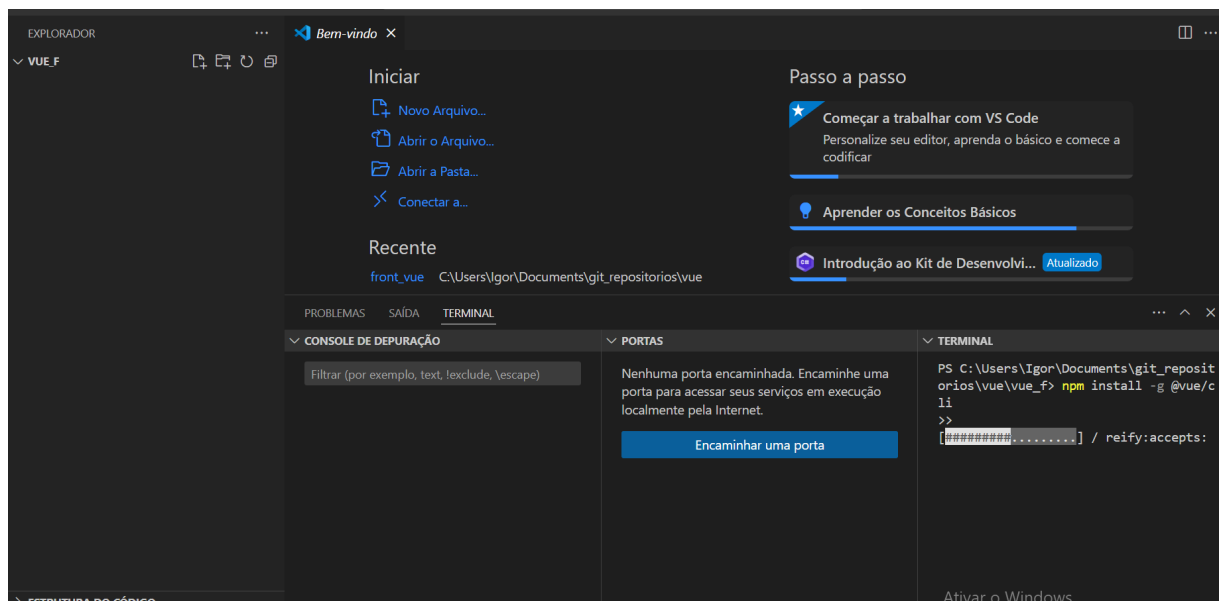
Instalar o Node.js:

- [Baixe e instale o Node.js.](#)

Após a instalação, verifique se ele está funcionando no terminal:

node -v

- npm -v



```
npm install -g @vue/cli
```

Instalar o Vue CLI:

- O Vue CLI é necessário para criar e gerenciar projetos Vue.js. Instale-o globalmente usando o NPM:

```
npm install -g @vue/cli
```

Quando o CLI pedir para escolher as configurações do projeto, escolher “Manually select features” (Selecionar recursos manualmente). Na opção **Choose Vue version** (Escolher versão do Vue), selecione **3.x**.

```

npm WARN deprecated shortid@2.2.16: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm WARN deprecated apollo-reporting-protobuf@3.4.0: The `apollo-reporting-protobuf` package is part of Apollo Server v2 and v3, which are no longer maintained. This package's functionality is now found in the `@apollo/usage-reporting-protobuf` package. See https://www.apollographql.com/docs/apollo-server/data/usage-reporting-protobuf/ for more details.
npm WARN deprecated apollo-server-env@4.2.1: The `apollo-server-env` package is part of Apollo Server v2 and v3, which are no longer maintained. This package's functionality is now found in the `@apollo/utils.fetcher` package. See https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws for more details.
npm WARN deprecated subscriptions-transport-ws@0.11.0: The `subscriptions-transport-ws` package is no longer maintained. We recommend using `graphql-ws` instead. See https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws for more details.
npm WARN deprecated vue@2.7.16: Vue 2 has reached EOL and is no longer actively maintained. See https://v2.vuejs.org/eol/ for more details.

changed 838 packages in 5m

75 packages are looking for funding
  run `npm fund` for details
PS C:\Users\Igor> vue --version
>>
vue : 0 arquivo C:\Users\Igor\AppData\Roaming\npm\vue.ps1 não pode ser carregado. O arquivo
C:\Users\Igor\AppData\Roaming\npm\vue.ps1 não está assinado digitalmente. Não é possível executar este script no
sistema atual. Para obter mais informações sobre como executar scripts e definir a política de execução, consulte
about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
No linha:1 caractere:1
+ vue --version
+ ~~~~~
+ CategoryInfo          : ErroDeSegurança: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\Igor>
>> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Alteração da Política de Execução
A política de execução ajuda a proteger contra scripts não confiáveis. A alteração da política de execução pode
implicar exposição aos riscos de segurança descritos no tópico da ajuda about_Execution_Policies em
https://go.microsoft.com/fwlink/?LinkID=135170. Deseja alterar a política de execução?
[S] Sim [A] Sim para Todos [N] Não [T] Não para Todos [U] Suspende [?] Ajuda (o padrão é "N"): s
PS C:\Users\Igor> vue --version
>>
@vue/cli 5.0.8
PS C:\Users\Igor> vue create front_vue

Vue CLI v5.0.8
? Please pick a preset: Default ([Vue 3] babel, eslint)

Vue CLI v5.0.8
✦ Creating project in C:\Users\Igor\front_vue.
⚙ Installing CLI plugins. This might take a while...
[ ] / idealTree:front_vue: sill idealTree buildDeps

```

vue create nome-do-projeto ()

Windows PowerShell não estava permitindo a execução de scripts externos.

Para resolver, executar o comando:

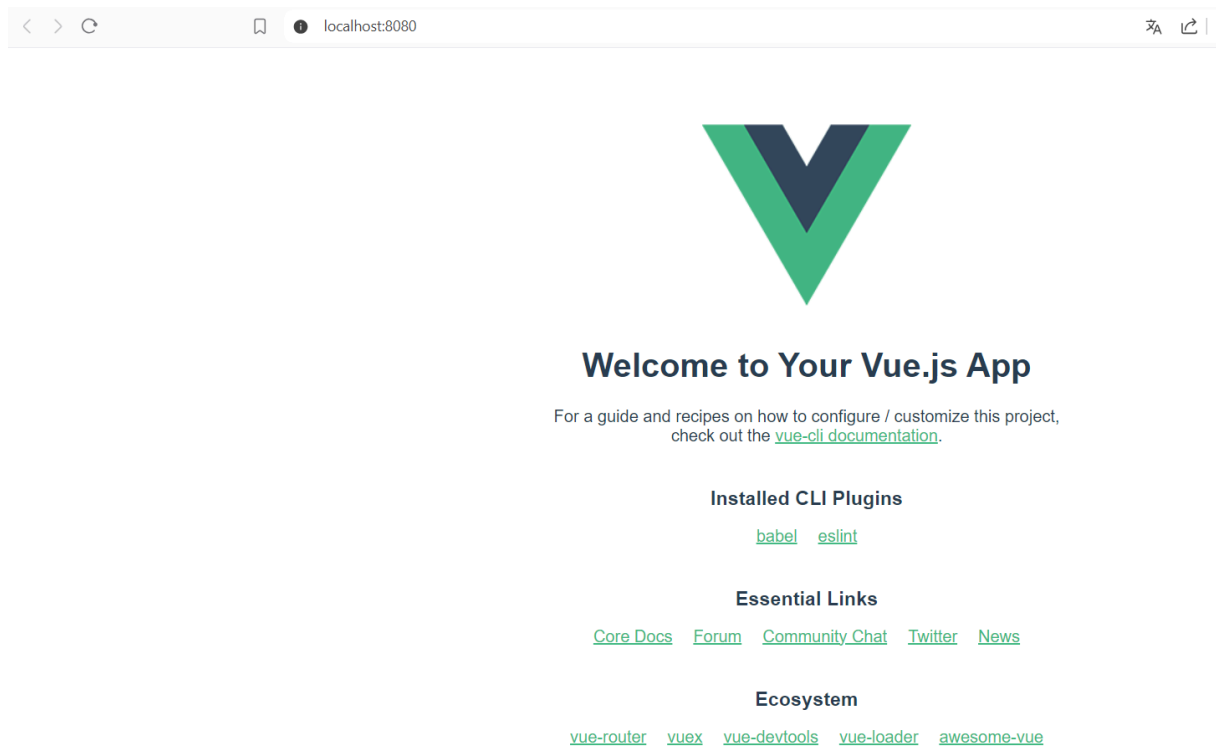
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Passo 2: Criar o Projeto com o Vue CLI

1. Inicie um novo projeto:

- No terminal, escolha o diretório onde deseja criar o projeto e execute:
vue create nome-do-seu-projeto

npm run serve - COMANDO PARA RODAR



Passo 3: Instalar Axios

Para realizar requisições HTTP, instalar o Axios:

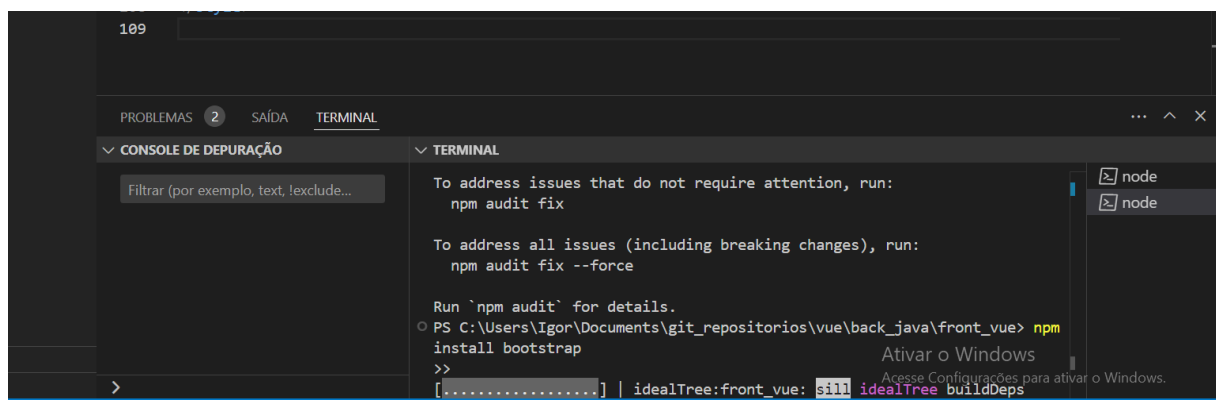
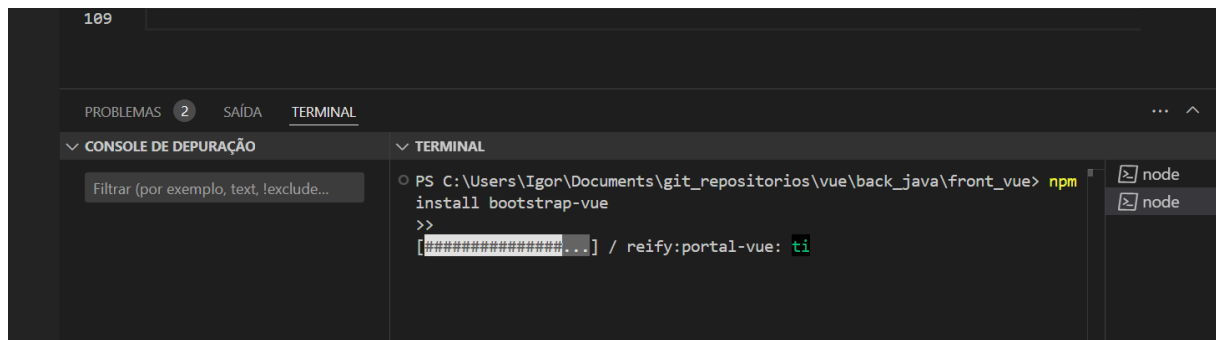
```
npm install axios
```

O **BootstrapVue** adapta o framework de design **Bootstrap** para Vue.js, permitindo que você use componentes do Bootstrap diretamente como componentes Vue (botões, cartões, modais, tabelas, etc.).

```
npm install bootstrap-vue bootstrap
```

O **Vue Router** é a biblioteca oficial de roteamento para Vue.js e é usada para gerenciar a navegação e as URLs da sua aplicação de forma eficiente.

```
npm install vue-router
```



MAIN JS

`createApp` é uma função do Vue 3 que cria uma instância da aplicação Vue.

`App` é o componente raiz da aplicação, geralmente definido em `App.vue`.

`BootstrapVue3` é a biblioteca que adapta o Bootstrap 5 para o Vue 3, fornecendo componentes estilizados e uma grade de layout responsiva.

`createApp(App)` cria uma nova instância da aplicação Vue usando `App.vue` como componente raiz.

`app.use(BootstrapVue3)`: Adiciona o BootstrapVue 3 como um plugin, disponibilizando seus componentes em toda a aplicação.

- **BootstrapVue 3** para estilização e layout responsivo.
- **Vue Router** para gerenciar a navegação entre diferentes páginas.
- Montagem e exibição da aplicação na interface do navegador.

APP.VUE

- **<div id="app">**: Contém toda a interface da aplicação e aplica o ID **app** ao elemento raiz, permitindo estilização específica.
- **<AppHeader />**: Componente de cabeçalho
- **<router-view />**: É o local onde os componentes correspondentes às rotas serão renderizados. O **Vue Router** substitui **<router-view />** pelo componente da rota atual, permitindo navegação entre páginas sem recarregar a aplicação inteira.

ROUTER

createRouter: Função usada para criar a instância do roteador.

createWebHistory: Configura o roteador para usar o histórico de navegação do navegador, permitindo URLs amigáveis (sem hash #).

Componentes (**AddressesTable**, **AddressForm**, **EditForm**): São importados de outros arquivos e usados como componentes de página que serão renderizados nas rotas especificadas.

- **path**: O caminho URL associado à rota.
- **name**: Nome da rota (opcional, mas útil para navegar de forma programática).
- **component**: O componente Vue que será exibido quando o caminho correspondente for acessado.

CABEÇALHO

<header>: Elemento de contêiner principal para o cabeçalho.

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">: Usa classes do Bootstrap para criar uma barra de navegação responsiva e escura.

<div class="container">: Alinha o conteúdo da barra de navegação no centro da tela.

CRUD de Endereços: Logo ou nome da aplicação, que aqui é **CRUD de Endereços**.

LISTA DE ENDEREÇOS

data(): Define a propriedade reativa **addresses**, que armazenará os endereços recuperados da API.

O **@click** em Vue.js é uma forma de adicionar um **"escutador de eventos"** (event listener) a um elemento. Quando o usuário clica no botão, a função ou método associado a esse evento é chamado.

NOVO ENDEREÇO

@submit: Esta é uma forma abreviada para `v-on:submit`, que escuta o evento de submissão do formulário.

.prevent: Este modificador é usado para evitar o comportamento padrão do evento.

export default { ... }:

- Isso exporta o componente como um módulo padrão, permitindo que ele seja importado e utilizado em outras partes do seu aplicativo Vue.js.

name: xxx:

- Essa propriedade define um nome para o componente. É útil para identificação e depuração, especialmente ao usar ferramentas de desenvolvimento ou quando você está trabalhando com hierarquias de componentes.

data() { return { ... } }:

- O método `data` retorna um objeto que contém dados reativos do componente. Em Vue.js, qualquer propriedade definida dentro desse objeto se torna reativa, o que significa que, se você modificar um valor, a interface do usuário (UI) será atualizada automaticamente para refletir essas mudanças.

address: { ... }:

- Dentro do objeto `data`, você tem uma propriedade chamada `address`, que é um objeto que contém três atributos: `rua`, `cidade` e `cep`.
- Inicialmente, todos esses atributos são definidos como strings vazias (""). Esses atributos são utilizados para armazenar os valores que o usuário irá inserir nos campos do formulário (como Rua, Cidade e CEP).

Componente BootstrapVue

```
<b-button variant="primary" @click="submit">Salvar</b-button>
```

Componente HTML Padrão

```
<button class="btn btn-primary" @click="submit">Salvar</button>
```

- **BootstrapVue:** O `<b-button>` já vem com a lógica de Vue, além dos estilos do Bootstrap aplicados automaticamente.

- **HTML Padrão:** O botão HTML precisa de classes CSS definidas (como `btn` e `btn-primary`) para ter um estilo semelhante, e você pode precisar lidar com o comportamento do botão manualmente.
- **Interatividade Aprimorada:** Muitos componentes do BootstrapVue têm interações prontas, como popovers, tooltips, modais, entre outros, que podem ser facilmente implementados sem necessidade de scripts adicionais.

Como Funciona a Atualização da DOM no Vue.js

1. **Reatividade:** Vue.js utiliza um sistema reativo que "observa" as propriedades do estado do componente. Quando uma propriedade **reativa** é alterada, o Vue automaticamente detecta essa mudança e atualiza a **DOM** correspondente.
2. **Template:** Ao utilizar Vue, você geralmente define suas interfaces de usuário em arquivos de template, que podem ser `.vue` ou em strings HTML. O Vue **compila** esses templates em funções **renderizadoras** que gerenciam a **criação e atualização** da DOM.
3. **Componentes:** Cada componente em Vue tem sua própria instância e seu próprio estado **reativo**. Quando o estado de um componente muda (por exemplo, através de métodos ou bindings de eventos), o Vue atualiza **automaticamente** a DOM desse componente.

Curva de Aprendizado Suave: Vue é conhecido pela simplicidade de uso e pela acessibilidade, facilitando a adoção mesmo para desenvolvedores que não têm ampla **experiência** com frameworks JavaScript. Seu modelo intuitivo permite a criação de projetos robustos sem exigir uma estrutura complexa desde o início.

Flexibilidade e Integrabilidade: Vue pode ser utilizado tanto em projetos grandes quanto pequenos. É altamente adaptável, pois possibilita a utilização de componentes de forma isolada ou a integração gradual em partes específicas de uma aplicação já existente, sem precisar reescrever toda a base de código.

Juntar com outros projetos front

rapido para iniciar

Usabilidade e Flexibilidade

- **Vue.js:** É altamente **flexível** e pode ser usado em projetos pequenos e grandes, com uma abordagem que permite adotar características de forma gradual.
- **Angular:** É mais opinativo, exigindo que você siga certas convenções e estruturas, o que pode ser bom para aplicações corporativas que precisam de consistência.
- **React:** Oferece liberdade total, permitindo que os desenvolvedores escolham como estruturar suas aplicações, mas isso pode levar a uma falta de consistência se não for gerido adequadamente.

Estrutura: Vue utiliza uma arquitetura de **model-view-viewmodel (MVVM)**, o que facilita a separação da lógica de negócio e da interface do usuário.

A arquitetura MVVM no Vue.js permite:

- **Separação de Preocupações:** A lógica de negócios e a apresentação são mantidas separadas, facilitando manutenção e testes.
- **Reatividade Simples:** A atualização da View acontece automaticamente em resposta a mudanças no Model, graças ao sistema de reatividade do Vue.
- **Foco na Lógica:** O ViewModel pode conter toda a lógica necessária para manipular os dados, enquanto a View se concentra na exibição.

Componentes do MVVM no Vue.js

Model: Representa os dados e a lógica de negócios da aplicação. No Vue.js, o Model geralmente consiste nas propriedades do objeto `data` do componente.

View: É a parte visual da aplicação, onde os dados do Model são exibidos. No Vue, a View é definida em templates HTML que utilizam a sintaxe do Vue para binding de dados.

ViewModel: É a camada intermediária que atua como um "banco de dados" entre o Model e a View. O ViewModel contém a lógica que manipula os dados do Model e notifica a View sobre quaisquer alterações.

Curva de Aprendizado (comparações)

- **Vue.js:** Considerado o mais fácil para iniciantes, Vue tem uma curva de aprendizado suave. A sintaxe é simples e intuitiva, tornando-o acessível para novos desenvolvedores.
- **Angular:** Possui uma curva de aprendizado mais acentuada, devido à sua complexidade e à necessidade de entender conceitos como DI (Injeção de Dependência) e RxJS para programação reativa.
- **React:** A curva de aprendizado é moderada, especialmente se você se familiarizar com conceitos como JSX e o estado do componente. No entanto, a escolha de bibliotecas para gerenciamento de estado e roteamento pode adicionar complexidade.

Aqui estão as principais extensões no VISUAL STUDIO CODE que você pode baixar para trabalhar com Vue.js:

1. Volar

Suporte completo à sintaxe Vue.js (arquivos `.vue`).

Autocompletar e linting (ajuda a evitar erros de sintaxe).

3. ESLint

ESLint para garantir que o código siga as convenções e padrões definidos.

4. Prettier - Code formatter

Usar o Prettier para formatar o código, essa extensão ajuda a manter o código organizado e formatado de maneira consistente.

- Formatação automática do código ao salvar.
- Suporte para Vue, JavaScript, CSS, e HTML.

5. Vue 3 Snippets

Essa extensão fornece snippets (atalhos) para código Vue.js, o que pode aumentar a produtividade.

- **Funções:**
 - Snippets para componentes, métodos, e opções do Vue.js.