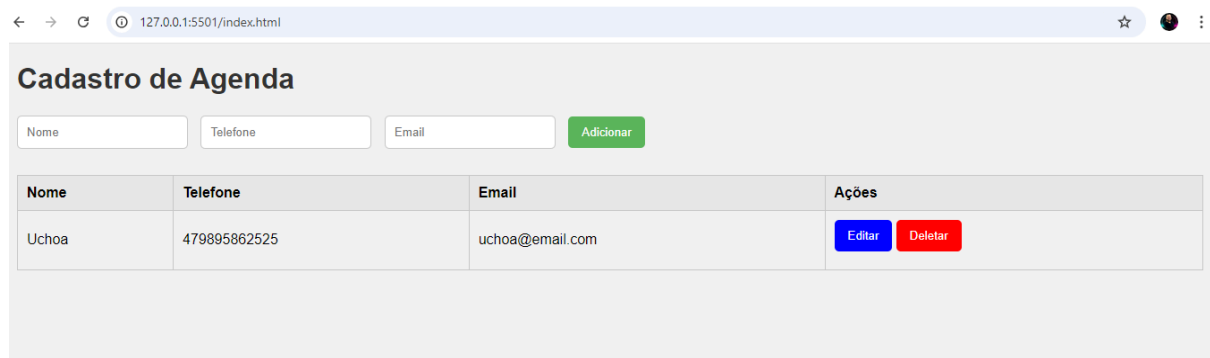


Prática: Crud com VueJS 2

Apresentando a versão finalizada:



Nome	Telefone	Email	Ações
Uchoa	479895862525	uchoa@email.com	<button>Editar</button> <button>Deletar</button>

Vamos adicionar uma mensagem de confirmação ao adicionar um contato, você pode usar uma nova propriedade de estado para controlar a visibilidade da mensagem. Vamos fazer algumas alterações para incluir essa funcionalidade.

Primeira modificação iremos adicionar um novo css:

```
.message {  
  display: none; /* Inicialmente oculta */  
  background-color: #dfff0d8; /* Verde claro */  
  color: #3c763d; /* Verde escuro */  
  padding: 10px;  
  margin: 20px 0;  
  text-align: center;  
  border: 1px solid #d6e9c6; /* Borda verde */  
  border-radius: 5px;  
}  
</style>
```

E uma div no código da seguinte forma:

```
</form>  
  
<div v-if="messageVisible" class="message">{{ messageText }}</div> <!-- Mensagem de confirmação -->  
  
<table>  
  <thead>
```

Adicionaremos agora essa parte do código onde terão as variáveis da mensagem de forma reativa:

```

    },
    contacts: [], // Array que armazenará todos os contatos
    editIndex: -1, // Índice para saber se estamos editando um contato
    messageVisible: false, // Controla a visibilidade da mensagem
    messageText: '' // Texto da mensagem
  ],
  methods: {

```

O que foi Adicionado

1. Novo Estado:

- `messageVisible`: controla se a mensagem de confirmação deve ser exibida.
- `messageText`: armazena o texto a ser exibido na mensagem.

2. Elemento da Mensagem:

- Adicionamos um `<div>` que aparece quando `messageVisible` é verdadeiro.

3. Método `showMessage(text)`:

- Este método exibe a mensagem e a oculta após 4 segundos usando `setTimeout`.

4. Chamadas ao `showMessage`:

- Chamamos `showMessage` após adicionar ou atualizar um contato para exibir uma mensagem apropriada.

Teste a Funcionalidade

Agora, ao adicionar ou atualizar um contato, você verá uma mensagem de confirmação no centro da tela por 4 segundos. Isso ajuda a melhorar a experiência do usuário, informando sobre ações bem-sucedidas. Se precisar de mais ajustes, é só avisar!

Para que a mensagem de confirmação apareça corretamente, você precisa chamar o método `showMessage` após adicionar ou atualizar um contato. Isso não foi feito nas funções `addContact` e `updateContact`.

```

methods: {
  addContact() {
    if (this.currentContact.name && this.currentContact.phone && this.currentContact.email) {
      this.contacts.push({ ...this.currentContact });
      this.showMessage('Contato adicionado com sucesso!'); // Exibe a mensagem
      this.resetForm();
    }
  },
  editContact(index) {
    this.currentContact = { ...this.contacts[index] };
    this.editIndex = index;
  },
  updateContact() {
    this.$set(this.contacts, this.editIndex, this.currentContact);
    this.showMessage('Contato atualizado com sucesso!'); // Exibe a mensagem
    this.resetForm();
    this.editIndex = -1;
  },
  deleteContact(index) {
    this.contacts.splice(index, 1);
  },
  resetForm() {
    this.currentContact = { name: '', phone: '', email: '' };
  },
  showMessage(text) {
    this.messageText = text;
    this.messageVisible = true;
    setTimeout(() => {
      this.messageVisible = false;
    }, 4000);
  }
}

```

← → ↺ 127.0.0.1:5501/index.html ☆ ⬇️ 🌐 ⋮

Cadastro de Agenda

Nome Telefone Email Adicionar

Contato adicionado com sucesso!

Nome	Telefone	Email	Ações
uchoa	47878787	carlos@hotmail.com	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>

Modificações

1. Chamadas para showMessage:

- Adicionei a chamada `this.showMessage('Contato adicionado com sucesso!');` em `addContact`.
- Adicionei a chamada `this.showMessage('Contato atualizado com sucesso!');` em `updateContact`.

2. Estilo da Mensagem:

- Removi `display: none;` do estilo da classe `.message`, pois a visibilidade agora é controlada pela diretiva `v-if`.

E para finalizar a melhoria vamos colocar um contador de usuários cadastrados centralizado em cima da tabela.

Para adicionar um contador de contatos cadastrados centralizado acima da tabela, você pode usar uma propriedade computada no Vue.js.

Vamos adicionar um css:

```
.counter {  
  text-align: center; /* Centraliza o contador */  
  font-size: 18px;  
  margin: 20px 0;  
  font-weight: bold;  
}  
</style>
```

```
<div v-if="messageVisible" class="message">{{ messageText }}</div> <!-- Mensagem de confirmação -->  
<div class="counter">Total de contatos: {{ contactsCount }}</div> <!-- Contador de contatos -->  
  
<table>  
  <thead>
```

Adicionamos uma div com a class counter e a variável que iremos por na propriedade computada

```
<script>  
  new Vue({  
    el: '#app',  
    data: {  
      currentContact: {  
        name: '',  
        phone: '',  
        email: ''  
      },  
      contacts: [],  
      editIndex: -1,  
      messageVisible: false,  
      messageText: ''  
    },  
    computed: {  
      contactsCount() {  
        return this.contacts.length; // Contador de contatos  
      }  
    },  
    methods: {  
      addContact() {
```

O que foi Adicionado

1. Contador de Contatos:

- Adicionamos uma nova `div` com a classe `counter` logo acima da tabela, que exibe o total de contatos cadastrados.

2. Propriedade Computada:

- `contactsCount`: uma propriedade computada que retorna o número de contatos atualmente no array `contacts`.

3. Estilo do Contador:

- Estilizamos o contador para que ele fique centralizado e em destaque.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5501/index.html'. The page title is 'Cadastro de Agenda'. Below the title, there is a form with three input fields labeled 'Nome', 'Telefone', and 'Email', followed by a green 'Adicionar' button. A green success message 'Contato adicionado com sucesso!' is displayed below the form. Below the message, it says 'Total de contatos: 3'. A table with four columns: 'Nome', 'Telefone', 'Email', and 'Ações' is shown. The table contains three rows of contact data. Each row has two buttons in the 'Ações' column: 'Editar' (blue) and 'Deletar' (red).

Nome	Telefone	Email	Ações
carlos	87878787872	carlos@email.com	<button>Editar</button> <button>Deletar</button>
silvio	54548778787	silvio@email.com	<button>Editar</button> <button>Deletar</button>
sestito	54545445445	sestito@gmail.com	<button>Editar</button> <button>Deletar</button>

