

# Formulários e validação de formulários



DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

# AGENDA

- Introdução
- Apresentação do VeeValidate
- Instalação do VeeValidate
- Aplicação de validações

Imagine o seguinte cenário, existe um formulário com dois campos, um chamado nome e o outro chamado valor. Ao preencher o formulário um determinado usuário se engana e preenche o nome no campo valor e não preenche o campo nome. Logo após isso, o mesmo clica no botão submeter e envia os dados para o servidor. Se o servidor estiver validando os dados, ok o sistema irá apresentar um erro. Mas vamos partir do pressuposto de que essas informações irão ser armazenadas no banco. Agora vou acrescentar mais uma informação, esse sistema é responsável pelo cadastro de produtos de uma grande rede de supermercados. Imagine o caos ao tentar vender um produto e este produto está com o preço nulo, zerado ou com um valor errado.

# Introdução

Então, a fim de evitar erros como o citado anteriormente, no ciclo de desenvolvimento devemos adotar algumas medidas de prevenção. Dentre várias medidas possíveis uma delas é a validação dos campos de input do formulário.



The image shows a web form titled "Adicionar Movimentação" with a close button (X) in the top right corner. The form contains five input fields: "Título", "Descrição", "Valor", "Tipo", and "Data". The "Título" field is currently empty and has a red error message "Este campo é obrigatório!" displayed below it. The "Data" field contains the placeholder text "dd/mm/aaaa" and a calendar icon on the right. At the bottom of the form is a blue button labeled "Salvar".

Existem diversas bibliotecas que podemos utilizar para fazer nossas validações. Mas irei apresentar aqui neste material a biblioteca VeeValidate.

Podemos utilizar o VeeValidate adicionando a tag ao nosso html:

```
<script src="https://unpkg.com/vee-validate"></script>
```

Ou fazendo a instalação via npm:

```
npm i vee-validate --save
```

Para validar nossos inputs utilizando VeeValidate é easy! Basta importar os componentes de validação em nosso componente e passar como propriedade o nosso método de validação. Vamos ver um exemplo.

# Aplicação das validações

```
<script>  
import { Form, Field, ErrorMessage } from "vee-validate";  
import validations from "../validations";
```



No script do nosso componente devemos primeiro fazer a importação dos componentes do VeeValidate.



# Aplicação das validações

```
components: {  
  Form,  
  Field,  
  ErrorMessage,  
},
```



Em seguida, devemos declarar os componentes que foram importados dentro do escopo components do nosso componente.

# Aplicação das validações

Vimos que são importados basicamente três componentes: Form, Field e ErrorMessage. Vamos ver em detalhes agora a aplicação desses três componentes.

# Aplicação das validações

Form:

- Envolve todo o formulário que se deseja fazer a validação.
- Possui um evento chamado submit que ao ser lançado, deve chamar o método responsável por submeter o formulário do nosso componente.

```
<Form @submit="salvarMovimentacao">
  <div class="mb-3">
    <label for="titulo" class="form-label">Título</label>
    <Field
      type="text"
      id="titulo"
      name="titulo"
      class="form-control"
      :rules="validations.isRequired"
      v-model="movimentacao.titulo"
    />
    <span class="mt-2 error-message">
      <ErrorMessage name="titulo" />
    </span>
  </div>
</Form>
```



# Dúvidas?





# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>