

## Two-way data binding

### Transcrição

O Vue disponibiliza a diretiva `v-model` que faz exatamente o que acabamos de fazer sem termos que explicitar os dois tipos de associações de dados que fizemos. Alterando nosso componente:

```
<!-- alurapic/src/components/cadastro/Cadastro.vue -->

<template>

  <div>
    <h1 class="centralizado">Cadastro</h1>
    <h2 class="centralizado">{{ foto.titulo }}</h2>

    <form @submit.prevent="grava()">
      <div class="controle">
        <label for="titulo">TÍTULO</label>
        <input v-model="foto.titulo" id="titulo" autocomplete="off">
      </div>

      <div class="controle">
        <label for="url">URL</label>
        <input v-model="foto.url" id="url" autocomplete="off">
        <imgem-responsiva/>
      </div>

      <div class="controle">
        <label for="descricao">DESCRIÇÃO</label>
        <textarea v-model="foto.descricao" id="descricao" autocomplete="off"></textarea>
      </div>

      <div class="centralizado">
        <meu-botao rotulo="GRAVAR" tipo="submit"/>
        <router-link to="/"><meu-botao rotulo="VOLTAR" tipo="button"/></router-link>
      </div>

    </form>
  </div>
</template>

<!-- código posterior omitido -->
```

Veja que escrevemos muito menos com a diretiva `v-model` e temos o mesmo resultado que antes. Tanto no código anterior quanto na diretiva `v-model` os dados são atualizados por padrão a cada entrada de dados que realizarmos nos elementos de entrada do formulário. Tanto isso é verdade que o subtítulo da página é atualizado a cada dígito no campo de título do formulário.

Agora que já temos uma maneira elegante de capturar a entrada do usuário, que tal passarmos as informações de que nosso componente `ImagenResponsiva` necessita para ser renderizado?

```
<!-- alurapic/src/components/cadastro/Cadastro.vue -->

<template>

<!-- código anterior omitido -->

<div class="controle">
  <label for="url">URL</label>
  <input v-model="foto.url" id="url" autocomplete="off">
  <imagem-responsiva :url="foto.url" :titulo="foto.titulo"/>
</div>

<!-- código posterior omitido -->

</template>

<!-- código posterior omitido -->
```

Excelente, mas essa solução deixa a desejar. A cada dígito no campo da URL atualizaremos `foto.url` com o valor digitado e, como realizamos um bind da propriedade de `foto.url` com a propriedade `url` de `ImagenResponsiva`, toda entrada fará com que o componente seja atualizado, ou seja, por debaixo dos panos o `src` de `img` será alterado, fazendo com que uma requisição seja disparada.

Em um mundo ideal, a atualização da URL entrada pelo usuário só deveria ser associada à `foto.url` quando ele deixasse o campo, ou seja, queremos postergar a aplicação da diretiva `v-model` apenas para o campo com o endereço da foto.

Isso é possível através do modificador `.lazy` suportado pela diretiva `v-model`. Alterando nosso template:

```
<!-- alurapic/src/components/cadastro/Cadastro.vue -->

<template>

<!-- código anterior omitido -->

<div class="controle">
  <label for="url">URL</label>
  <input v-model.lazy="foto.url" id="url" autocomplete="off">
  <imagem-responsiva :url="foto.url" :titulo="foto.titulo"/>
</div>

<!-- código posterior omitido -->

</template>

<!-- código posterior omitido -->
```

Podemos adicionar o modificador também no campo do título, fazendo com que o valor do título digitado só seja exibido na como subtítulo da página assim que o usuário sair do campo:

```
<!-- alurapic/src/components/cadastro/Cadastro.vue -->
```

```
<template>
<!-- código anterior omitido -->
<div class="controle">
  <label for="titulo">TÍTULO</label>
  <input v-model.lazy="foto.titulo" id="titulo" autocomplete="off">
</div>
<!-- código posterior omitido -->
```

Por fim, vamos exibir `ImagemResponsiva` apenas quando o valor de `foto.url` for diferente de branco:

```
<!-- alurapic/src/components/cadastro/Cadastro.vue -->

<template>
<!-- código anterior omitido -->

<div class="controle">
  <label for="url">URL</label>
  <input v-model.lazy="foto.url" id="url" autocomplete="off">
  <imagem-responsiva v-show="foto.url" :url="foto.url" :titulo="foto.titulo"/>
</div>
<!-- código posterior omitido -->
</template>

<!-- código posterior omitido -->
```