



## AULA 05 - Video 5

### Introdução

Nesta aula vamos estar vendo um pouco mais sobre a comunicação entre os componentes feitos nas aulas passadas.

### Desenvolvimento

No VueJS, podemos passar um atributo em uma tag e puxar ela na interpolação. Para fazermos isso, primeiro vamos atribuir o atributo. Na aula, de exemplo vamos fazer isso no botão:

```
<app-btn texto ="Abacaxi"> </app-btn>
```

Feito isso, vamos deixar o nosso return vazio e vamos definir um atributo chamado “props” que vai receber o nosso atributo feito acima:

```
props: ["texto"],
data: () => {
  return { }
},
template: `
  <button style="background-color:blue; color:white;
border:none; padding: 8px 10px;border-radius:5px;">
    {{ texto }}
  </button>
`
```

Para verificarmos se tudo ocorreu bem, basta voltarmos no navegador e visualizar se o nome do botão mudou:

## Introdução ao Vuejs

Digite o nome do cliente Adicionar

Jão da Silva Remove

Maria da Silva Remove

Zé da Silva Remove

Pafúncio Silveira Remove

Abacaxi

Também podemos definir valor padrão da propriedade se não passarmos nada, e se ela é obrigatório ou não

### AppBtn.js

```
props: {
  texto: {
    type: String,
    required: true,
    default: "Sem texto"
  }
},
```

html:

```
<app-btn texto="Abacaxi"></app-btn>
<app-btn texto="Banana"></app-btn>
<app-btn texto="Maçã"></app-btn>
<app-btn></app-btn>
```

podemos visualizar no navegador

## Introdução ao Vuejs

Digite o nome do cliente Adicionar

Jão da Silva Remove

Maria da Silva Remove

Zé da Silva Remove

Pafúncio Silveira Remove

Abacaxi Banana Maçã Sem texto

Por fim, basta visualizarmos a documentação, posto que, ela é de suma importância para visualizarmos as outras comunicações entre componentes.

<https://br.vuejs.org/v2/guide/components-props.html>

Para acabar a aula, vamos estar vendo agora um pouco mais sobre a comunicação entre filho e pai, isso depende de evento, oque chamamos de Event Emitter. Nesse exemplo, vamos novamente fazer isso com o botão. Na nossa const AppBtn, vamos adicionar um método no qual ele vai receber uma função com o nome btnClicado e dentro dessa função vamos ter um alert:

```
methods: {  
  btnClicado() {  
    alert("Botão Clicado")  
  }  
},
```

Feito isso, em nosso template vamos adicionar o:

```
template: '<button @click ="btnClicado() "'
```

Por fim, basta verificarmos se funcionou na página.

Caso queira fazer uma comunicação para Cima, pro Componente que esta me utilizando.

adicionar também qual atributo que foi clicado, basta adicionarmos os seguintes códigos:

```
methods: {  
  btnClicado() {  
    this.$emit("clicado", "Botão" + this.texto + "foi clicado")  
  }  
},
```

**A cima emitimos o evento pro Pai, então temos que interceptar no pai, captar o evento e tratar nos 'index'**

em nosso index.html:

```
<app-btn texto="Abacaxi" v-on:clicado="btnClicado"></app-btn>  
<app-btn texto="Banana" @clicado="btnClicado"></app-btn>  
<app-btn texto="Maçã" @clicado="btnClicado"></app-btn>  
<app-btn @clicado="btnClicado"></app-btn>
```

em nosso index.js:

```
btnClicado (msg) {  
  alert(msg)  
}
```

Resultado:



Principais comunicações que usaremos:

Comunicação Pai para Filho: Props.

Comunicação Filho para Pai: Event Emitter.