# AULA 02 - Vídeo 2

## Introdução

Dando continuidade a nosso aprendizado de vue

Senha escrito simultaneamente, mas isso nao e nada de novo

```
<body>
    <div id="app">
        <h1>{{title}}</h1>
         <input v-model="title" type="text" placeholder="Digite o titulo
da pagina">
        <br>
         <input v-model="senha" type="password" placeholder="digite sua
senha">
        {{senha}}
    </div>


                                                                <script
src="https://cdnjs.cloudflare.com/ajax/libs/vue/3.2.33/vue.global.min.j
s"></script>
    <script>
        Vue.createApp({
            data: function(){ // ou Arrow function () =>
                return {
                    title: "Introducao ao VueJS",
                    senha: ""
                }
            }
        }).mount("#app")
    </script>
```

Atributo nativamente elemento do Html5 seja compativel com área de dados do Vue:

**“v-bind:”**

```
<body>
    <div id="app">
        <h1>{{title}}</h1>
         <input v-model="title" type="text" placeholder="Digite o titulo
da pagina">
        <br>
                        <input   v-model="senha"   v-bind:type="tipoSenha"
placeholder="digite sua senha">
        {{senha}}
    </div>

                                                                    <script
src="https://cdnjs.cloudflare.com/ajax/libs/vue/3.2.33/vue.global.min.j
s"></script>
    <script>
        Vue.createApp({
            data: function(){ // ou Arrow function () =>
                return {
                    title: "Introducao ao VueJS",
                    senha: "",
                    tpoSenha: "password"
                }
            }
        }).mount("#app")
    </script>
```

serve para outros tipos também

```
tipoSenha: "date"
```

```
tipoSenha: "number"
```

Ou seja consegue fazer elementos nativamente html leiam áreas de dados do Vue

De forma simples podemos Utilizar métodos que ja conhecemos como onclick, mas utilizaremos a versão do Vue que tem pequenas diferenças assim podemos mudar o valor da variável de data do  vue

```
<body>
    <div id="app">
        <h1>{{title}}</h1>
         <input v-model="title" type="text" placeholder="Digite o titulo
da pagina">
        <br>
                        <input   v-model="senha"   v-bind:type="tipoSenha"
placeholder="digite sua senha">
        <button v-on:click=" tipoSenha ='text' ">Exibir Senha</button>
               <button  v-on:click="  tipoSenha  ='password'  ">Esconder
Senha</button>
    </div>

                                                                    <script
src="https://cdnjs.cloudflare.com/ajax/libs/vue/3.2.33/vue.global.min.j
s"></script>
    <script>
        Vue.createApp({
            data: () => {
                return {
                    title: "Introducao ao VueJS",
                    senha: "",
                    tipoSenha: "password"
                }
            },

        }).mount("#app")
    </script>
</body>
</html>
```

Versão “Short”/Curta de se escrever “v-bind:type” e “v-on:click” ,
Mais comumente utilizado:

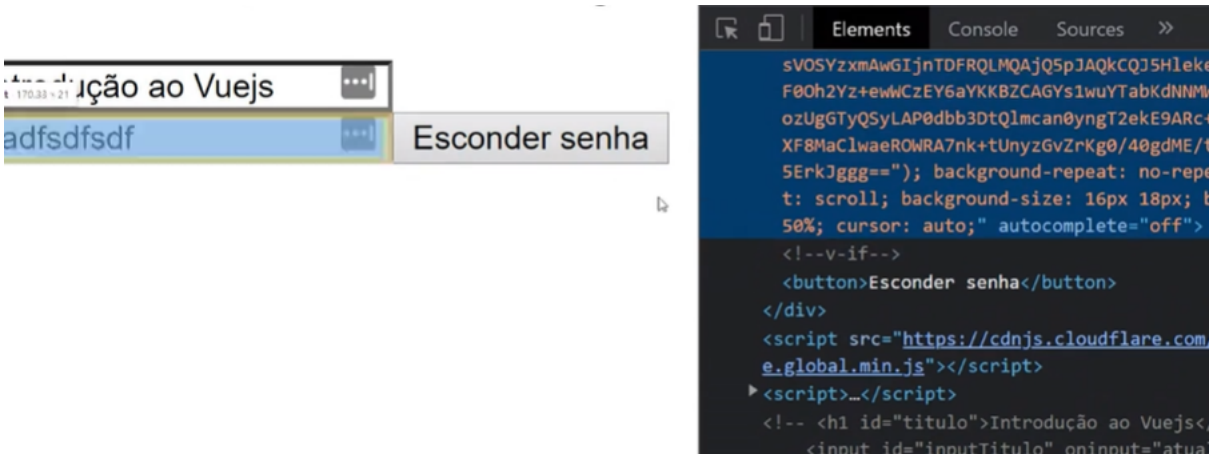
```
  <input   v-model="senha"   :type="tipoSenha"   placeholder="digite   sua
senha">
```

```
<button @click=" tipoSenha ='text' ">Exibir Senha</button>
                  <button   @click="   tipoSenha   ='password'   ">Esconder
Senha</button>
```

**Para aparecer Somente um botão, utilizando condicionais do vue:**

```
        <input v-model="title" type="text" placeholder="Digite o titulo
da pagina">
        <br>
           <input v-model="senha" :type="tipoSenha" placeholder="digite
sua senha">
            <button v-if="tipoSenha == 'password'" @click=" tipoSenha
='text' ">Exibir Senha</button>
              <button  v-if="tipoSenha  ==  'text'"  @click="  tipoSenha
='password' ">Esconder Senha</button>
```

**v-if remove os botões, remove a Tag do DOM, então temos que ter mais cautela:**




**Já o v-show, não remove a tag, ele muda a estilização dos botões no css, display: none, fica exibida no inspetor de elementos.**

```
    <button v-show="tipoSenha == 'password'" @click=" tipoSenha ='text'
">Exibir Senha</button>
```

**Podemos trabalhar com If e Else também:**
**Lembrando que eles tem que respeitar a Ordem do else ser embaixo do If:**

```
<input   v-model="senha"   :type="tipoSenha"   placeholder="digite   sua
senha">
            <button  v-if="tipoSenha  ==  'password'"  @click="  tipoSenha
='text' ">Exibir Senha</button>
             <button  v-else  @click="  tipoSenha  ='password'  ">Esconder
Senha</button>
```

e v-else-if , tambem funciona..

**Atalho: (Alt + seta) na linha desejada, a move a linha para cima ou para baixo**

v-show, não possui else..

```
  <div id="app">
        <h1>{{title}}</h1>
         <input v-model="title" type="text" placeholder="Digite o titulo
da pagina">
        <br>
           <input  v-model="senha"  :type="tipoSenha"  placeholder="digite
sua senha">
          <button  v-show="tipoSenha  ==  'password'"  @click="  tipoSenha
='text' ">Exibir Senha</button>
            <button  v-show="tipoSenha  ==  'text'"  @click="  tipoSenha
='password' ">Esconder Senha</button>
    </div>
```

**Da mesma maneira que a área de Dados, há também área de métodos onde você escreve suas funções:**

```
methods: {
            exibirSenha (){
                this.tipoSenha = 'text';
            }
           }
```

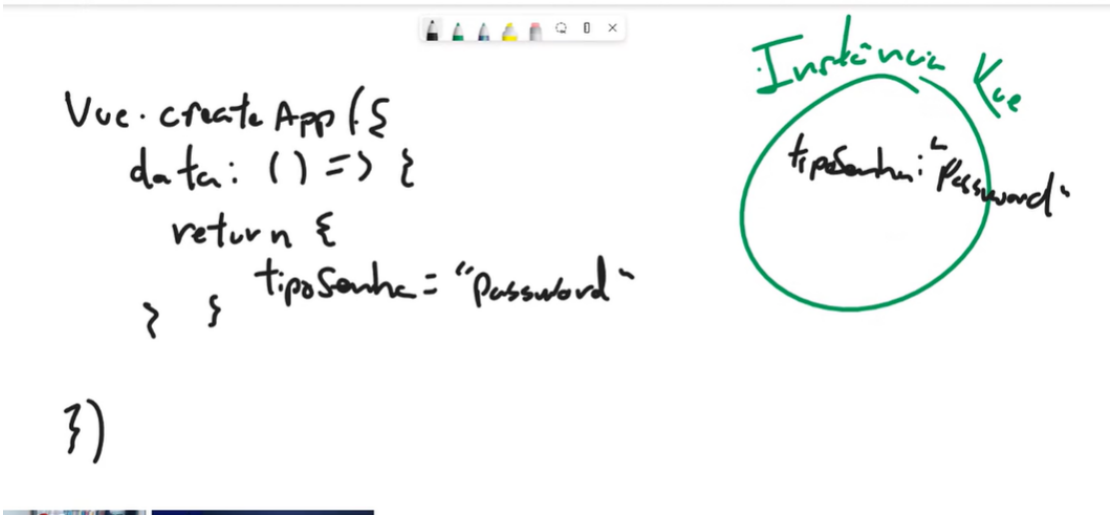
**Para acessar area de datas do vue precisa se utilizar 'this': ^**

**No ‘template’ , área de exibição do Html, não necessita do ‘this’ para acessar as variáveis e ao chamar funções o parêntese é opcional() , a não ser que haja parâmetros, ai é obrigatório:**

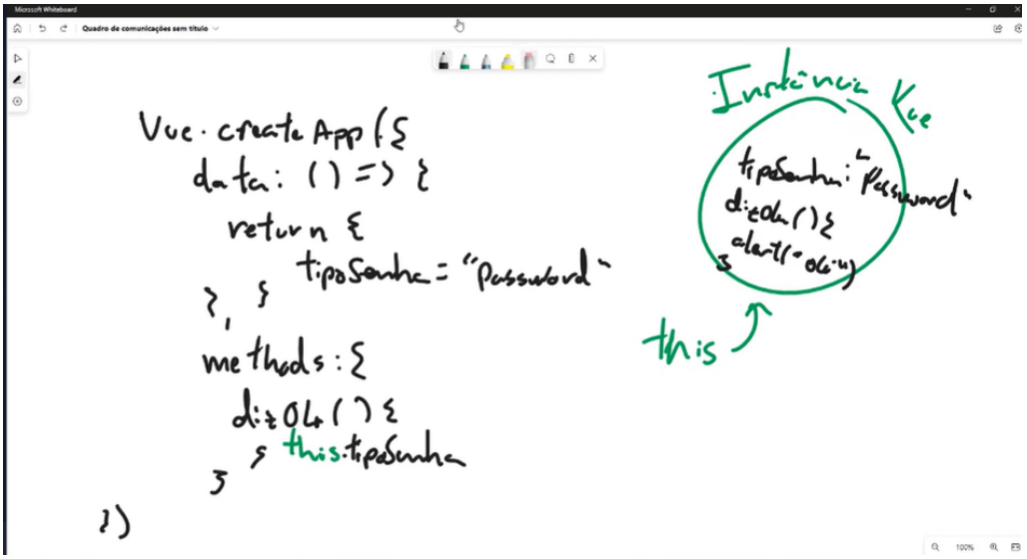
```
<div id="app">
        <h1>{{title}}</h1>
         <input v-model="title" type="text" placeholder="Digite o titulo
da pagina">
        <br>
           <input v-model="senha" :type="tipoSenha" placeholder="digite
sua senha">
                          <button   v-show="tipoSenha   ==   'password'"
@click="exibirSenha()">Exibir Senha</button>
                             <button   v-show="tipoSenha   ==   'text'"
@click="esconderSenha">Esconder Senha</button>
    </div>

                                                                    <script
src="https://cdnjs.cloudflare.com/ajax/libs/vue/3.2.33/vue.global.min.j
s"></script>
    <script>
        Vue.createApp({
            data: () => {
                return {
                    title: "Introducao ao VueJS",
                    senha: "",
                    tipoSenha: "password"
                }
            },
           methods: {
            exibirSenha (){
                this.tipoSenha = 'text';
            },
            esconderSenha(){
                this.tipoSenha = 'password'
            }
           }
        }).mount("#app")
```

**Porque utilizamos o ‘this’?**

o Vue.create App , cria uma instância do Vue, e copia nossa área de data, e os métodos, tudo que está dentro do return, ele copia para nova instância




E também a área de métodos:

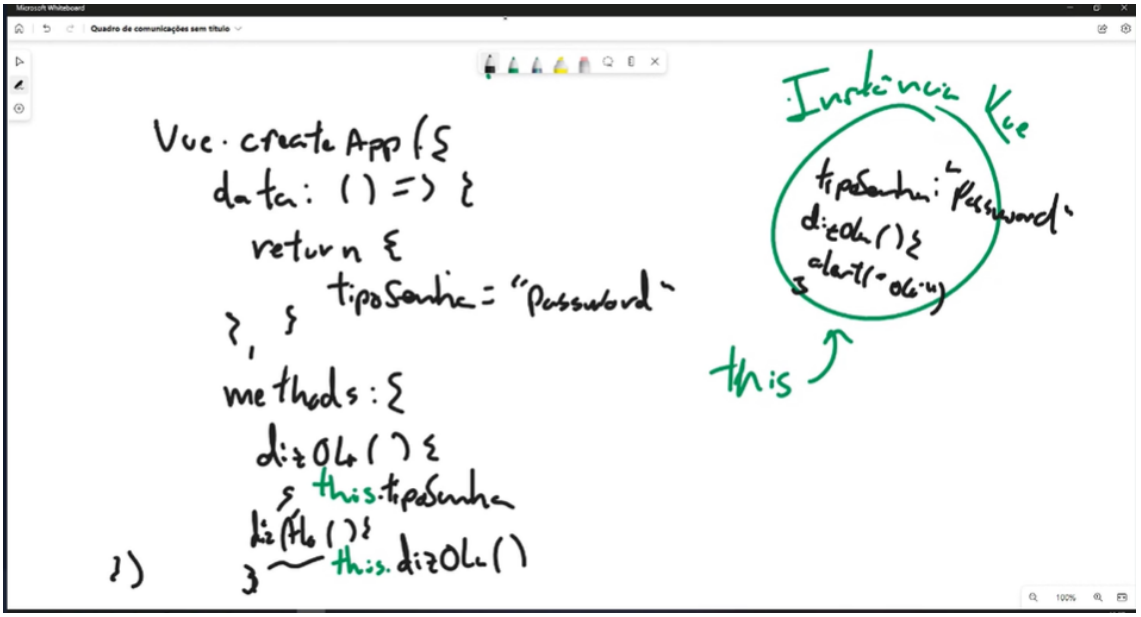



Então o **This**, é esta **instância..**

na criação de data não precisa

pois você não está fazendo uma ação, está só informando quais são os atributos, atribuição dos valores entao nao precisa do this,

já na área dos métodos necessita, Isso também vai servir para hooks, e ciclo de vida do vue




**“No Html, o Template”** , tambem nao necessita, pois o html já acessa direto a instância, a instância já está feita, o html já acessa direto a instância, sem precisa do this.