

Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Desenvolvimento Web – Front-End

1.º Ano/2.º Semestre

Docente: Marco Miguel Olival Olim

Data 01/02/2019

ESTE EXERCÍCIO ABORDA OS CONCEITOS PRINCIPAIS DA FRAMEWORK DE FRONT-END VUEJS

Vue (pronuncia-se /vju:/, como **view**, em inglês) é um **framework progressivo** para a construção de interfaces para o utilizador (UI). Ao contrário de outros *frameworks* monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal foca exclusivamente a camada visual (*view layer*), sendo relativamente fácil adotar e integrar com outras bibliotecas ou projetos existentes. Por outro lado, o Vue é também perfeitamente capaz de servir de base para o desenvolvimento de sofisticadas *Single-Page Applications* quando utilizado conjuntamente com **ferramentas modernas e bibliotecas de apoio**.

É bastante simples começar a utilizar o VueJS, ao contrário de outras frameworks, visto que basta incluí-lo numa qualquer página de html. Para verificar este procedimento, começamos por criar vue-1.html

```
<> vue-1.html ✕  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  
4  <head>  
5    <meta charset="UTF-8">  
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
8    <title>Document</title>  
9  </head>  
10  
11 <body>  
12  
13 </body>  
14  
15 </html>
```

Introduzimos uma tag div com o id “app” onde irá funcionar o Vue e, temporariamente colocamos uma tag h1 com o texto que será interpolado posteriormente pelo Vue:

```
11 </body>  
12 <div id="app">  
13   <h1>Desenvolvimento WEB</h1>  
14 </div>  
15 </body>
```

Importamos o Vue a partir de um CDN com recurso a uma tag script

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.6/dist/vue.js"></script>
```

Cofinanciado por:

Agora que já dispomos do Vue, abrimos mais uma tag script onde é criada uma nova instância do Vue. Nesta é feita a passagem de um objeto com a configuração do Vue. Por exemplo, precisamos indicar ao Vue onde irá persistir na página de html, e para tal usamos a chave **el** que terá como valor o id app que criamos anteriormente:

```
<div id="app">
  <h1>Desenvolvimento WEB</h1>
</div>
<script src="https://cdn.jsdelivr.net/npm/vue@2.6.6/dist/vue.js"></script>
<script>
  new Vue({
    el: '#app'
  })
</script>
```

Indicamos agora os dados que queremos utilizar com o Vue, adicionando o elemento **data** ao objeto de configuração.

```
16 <script>
17   new Vue({
18     el: '#app',
19     data: {
20       texto: ' com VueJS'
21     }
22   })
23 </script>
```

Para usarmos estes dados utilizamos interpolação de string com a notação **mustache**, caracterizada por chavetas duplas no início e fim dos dados a interpolar

```
12 <div id= app >
13   <h1>Desenvolvimento WEB {{texto}}</h1>
14 </div>
```

Devemos obter o seguinte resultado na página web:



Desenvolvimento WEB com VueJS

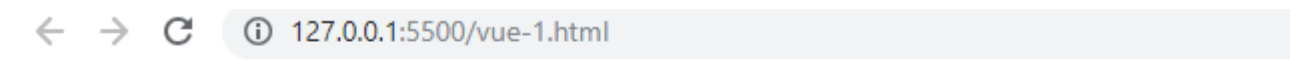
Cofinanciado por:



Desta forma podemos constatar que o Vue funciona corretamente na página. No entanto, estas frameworks caracterizam-se mais pela **reatividade** que proporcionam, pelo que podemos usar a diretiva **v-model** associado a uma simples tag de input para exemplificar este funcionamento

```
12 <div id= app >
13   <h1>Desenvolvimento WEB {{texto}}</h1>
14   <input type="text" v-model="texto">
15 </div>
```

Neste momento, o que digitar na caixa de texto irá se refletir na interpolação de string obtida com o Vue



Desenvolvimento WEB com HTML e VueJS

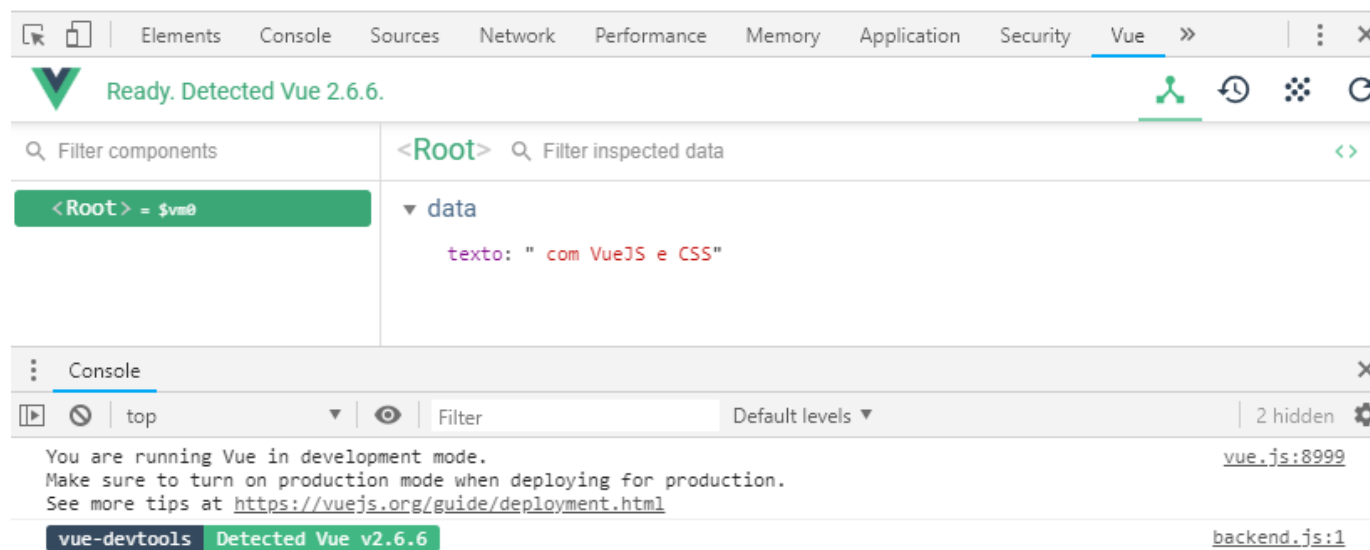
com HTML e VueJS

Em suma, o que digitar reflete-se no DOM e no Código, mas também qualquer alteração no Javascript reflete-se no DOM. Esta característica das frameworks é classificada de **dual binding**. Para testar podemos usar a extensão do chrome “Vue DevTools” e alterar os dados diretamente:



Desenvolvimento WEB com VueJS e CSS

com VueJS e CSS



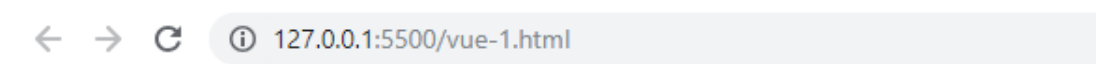
Cofinanciado por:



Caso não tenha esta extensão instalada, pode simplesmente recorrer à consola do browser. No entanto, para aceder ao Vue precisa de efetuar uma alteração ao código, nomeadamente atribuir uma variável à instância

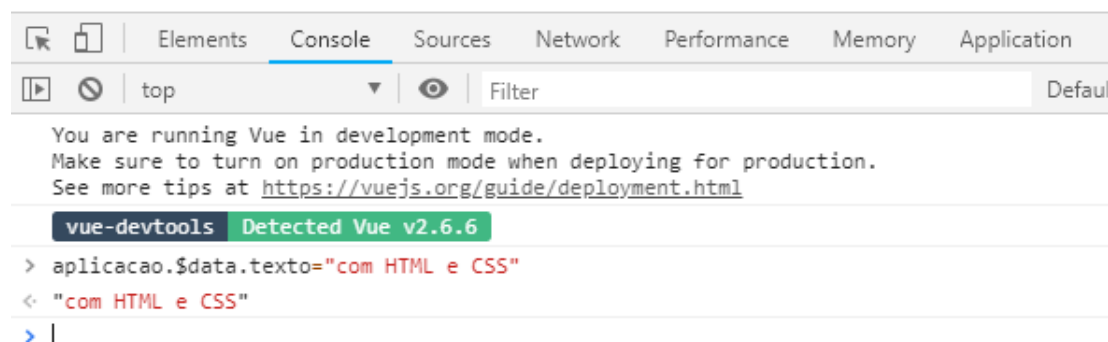
```
<> vue-1.html ●
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <meta http-equiv="X-UA-Compatible" content="ie=edge">
8    <title>Document</title>
9  </head>
10
11 <body>
12   <div id="app">
13     <h1>Desenvolvimento WEB {{texto}}</h1>
14     <input type="text" v-model="texto">
15   </div>
16   <script src="https://cdn.jsdelivr.net/npm/vue@2.6.6/dist/vue.js"></script>
17   <script>
18     var aplicacao = new Vue({
19       el: '#app',
20       data: {
21         texto: ' com VueJS'
22       }
23     })
24   </script>
25 </body>
26
27 </html>
```

Poderá agora aceder à variável global “aplicacao” a partir da consola



Desenvolvimento WEB com HTML e CSS

com HTML e CSS



Cofinanciado por:

