

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 16/03/2018

ESTE EXERCÍCIO PRETENDE ILUSTRAR O FUNCIONAMENTO DO WATCH - PROPRIEDADE DA INSTÂNCIA DO VUE JS

Iniciar uma nova aplicação VueJS com esta template fornecida:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Exercício 6</title>
</head>

<body>
  <script src="https://npmcdn.com/vue/dist/vue.js"></script>
  <div id="exercicio6">

    </div>
    <script>
      new Vue({
        el: '#exercicio6',
        data: {}
      });
    </script>
  </body>

</html>
```

Inicializar uma variável **contador** a zero na propriedade de data da instância do VueJS

```
data: {
  contador: 0
}
```

Cofinanciado por:

Adicionar à template de html dois botões para incrementar/decrementar o contador, além de interpolar essa string numa tag div

```
<div id="exercicio6">
  <button @click="contador++">Incrementa</button>
  <button @click="contador--">Decrementa</button>
  <div>{{ contador }}</div>
</div>
```

Queremos agora reinicializar automaticamente, após 2 segundos, o valor de **contador** novamente para zero. Utilizamos para o efeito a propriedade **watch**, a qual executa código sempre que há uma alteração nos dados

```
<script>
  new Vue({
    el: '#exercicio6',
    data: {
      contador: 0
    },
    watch:{

    }
  });
</script>
```

Esta propriedade executa código sempre que há uma alteração nos dados, passando esta alteração por argumento (seguidamente designado por **valor**)

```
watch:{
  contador: function (valor) {

  }
}
```

Utilizamos agora a função de javascript **setTimeout()** para temporizar os dois segundos (2000 milissegundos)

```
watch:{
  contador: function (valor) {
    setTimeout(function(){}, 2000)
  }
}
```

Cofinanciado por:



Como será utilizado uma função em `setTimeout` para passar o valor ao contador (ver closure de um callback de uma função), é necessário inicializar uma variável `dados` para poder aceder ao valor do contador da propriedade `data` do VueJS

```
watch: {  
  contador: function(valor) {  
    var dados = this;  
    setTimeout(function() {  
      dados.contador = 0  
    }, 2000)  
  }  
}
```

Após utilizar um dos botões, o valor do contador deve agora voltar a zero após dois segundos:



Cofinanciado por:

