

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

Unidade Curricular: Desenvolvimento Web – Front-End

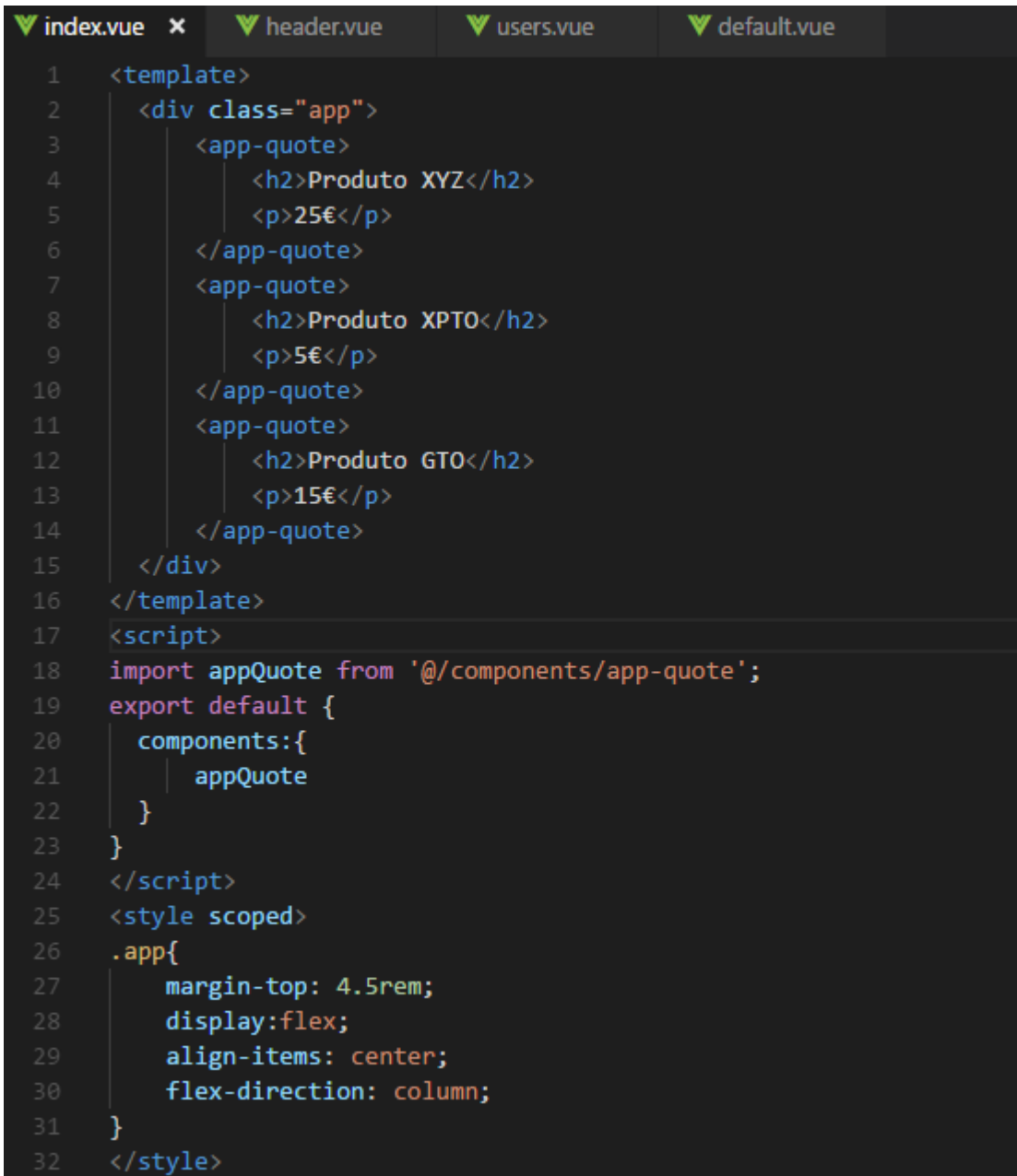
1º Ano/2º Semestre

Docente: Marco Miguel Olival Olim

Data 06/04/2018

ESTE EXERCÍCIO ABORDA A INTRODUÇÃO DE INFORMAÇÃO EM COMPONENTES E SUA GRAVAÇÃO EM BASES DE DADOS

Na continuação do exercício anterior, acrescentamos alguns produtos após formatar o ficheiro index.vue, correspondente à primeira página do site:



```
1 <template>
2   <div class="app">
3     <app-quote>
4       <h2>Produto XYZ</h2>
5       <p>25€</p>
6     </app-quote>
7     <app-quote>
8       <h2>Produto XPTO</h2>
9       <p>5€</p>
10    </app-quote>
11    <app-quote>
12      <h2>Produto GTO</h2>
13      <p>15€</p>
14    </app-quote>
15  </div>
16 </template>
17 <script>
18 import appQuote from '@components/app-quote';
19 export default {
20   components:{
21     appQuote
22   }
23 }
24 </script>
25 <style scoped>
26 .app{
27   margin-top: 4.5rem;
28   display: flex;
29   align-items: center;
30   flex-direction: column;
31 }
32 </style>
```

Atendendo a que utilizamos repetidamente o mesmo processo para descrever um produto, vamos converter este num componente que recebe o nome de um produto e um preço:

```
index.vue  app-ficha-produto.vue x
1  <template>
2    <app-quote>
3      <h2>{{produto}}</h2>
4      <p>{{valor}}€</p>
5    </app-quote>
6  </template>
7  <script>
8    import appQuote from '@components/app-quote';
9    export default {
10     components:{
11       appQuote
12     },
13     props:{
14       produto:{
15         type: String,
16         required: true
17       },
18       valor:{
19         type: Number,
20         required: true
21       }
22     }
23   }
24 </script>
```

Seguidamente temos de importar este novo componente para a primeira página:

```
index.vue x  app-ficha-produto.vue
1  <template>
2    <div class="app">
3      <app-ficha-produto produto="Produto XYZ" valor="25" />
4      <app-ficha-produto produto="Produto XPTO" valor="5" />
5      <app-ficha-produto produto="Produto GTO" valor="15" />
6    </div>
7  </template>
8  <script>
9    import appFichaProduto from '@components/app-ficha-produto';
10   export default {
11     components:{
12       appFichaProduto
13     }
14   }
15 </script>
16 <style scoped>
17   .app{
18     margin-top: 4.5rem;
19     display: flex;
20     align-items: center;
21     flex-direction: column;
22   }
23 </style>
```

Cofinanciado por:

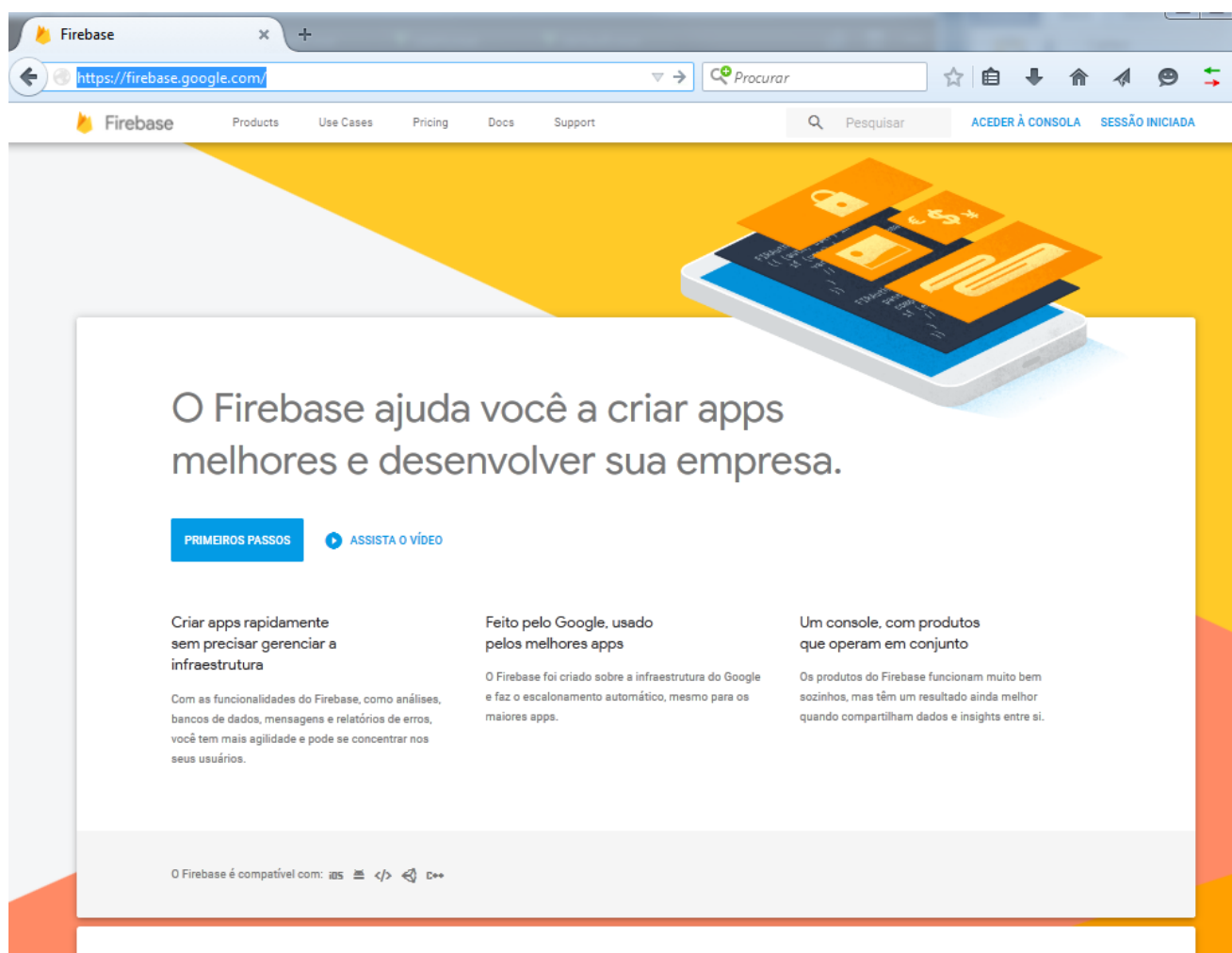
Separamos agora os dados do html, introduzindo uma diretiva v-for

```
index.vue x header.vue users.vue default.vue
1 <template>
2   <div class="app">
3     <app-ficha-produto v-for="(artigo, index) in artigos"
4                           :key="index"
5                           :produto="artigo.produto"
6                           :valor="artigo.valor" />
7   </div>
8 </template>
9 <script>
10 import appFichaProduto from '@components/app-ficha-produto';
11 export default {
12   components:{
13     appFichaProduto
14   },
15   data() {
16     return {
17       artigos:[
18         {
19           produto:"Produto XYZ",
20           valor:25
21         },
22         {
23           produto:"Produto XPTO",
24           valor:5
25         },
26         {
27           produto:"Produto GTO",
28           valor:15
29         }
30       ]
31     }
32   }
33 }
34 </script>
35 <style scoped>
36 .app{
37   margin-top: 4.5rem;
38   display:flex;
39   align-items: center;
40   flex-direction: column;
41 }
42 </style>
```

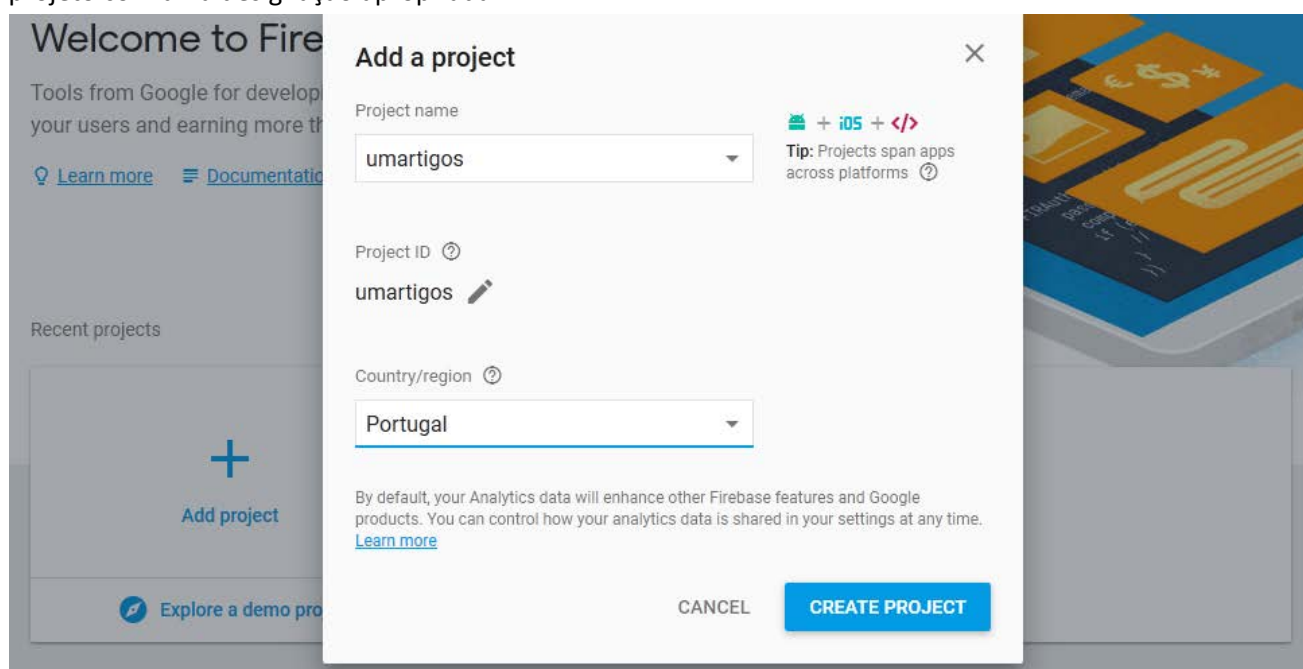
Cofinanciado por:



Os dados introduzidos desta maneira servem para testar a template mas não persistem, isto é, qualquer alteração que façamos perde-se ao reiniciarmos o browser. Uma vez que ainda não temos back-end neste projeto, vamos utilizar o Firebase, da Google, que disponibiliza uma base de dados como serviço.



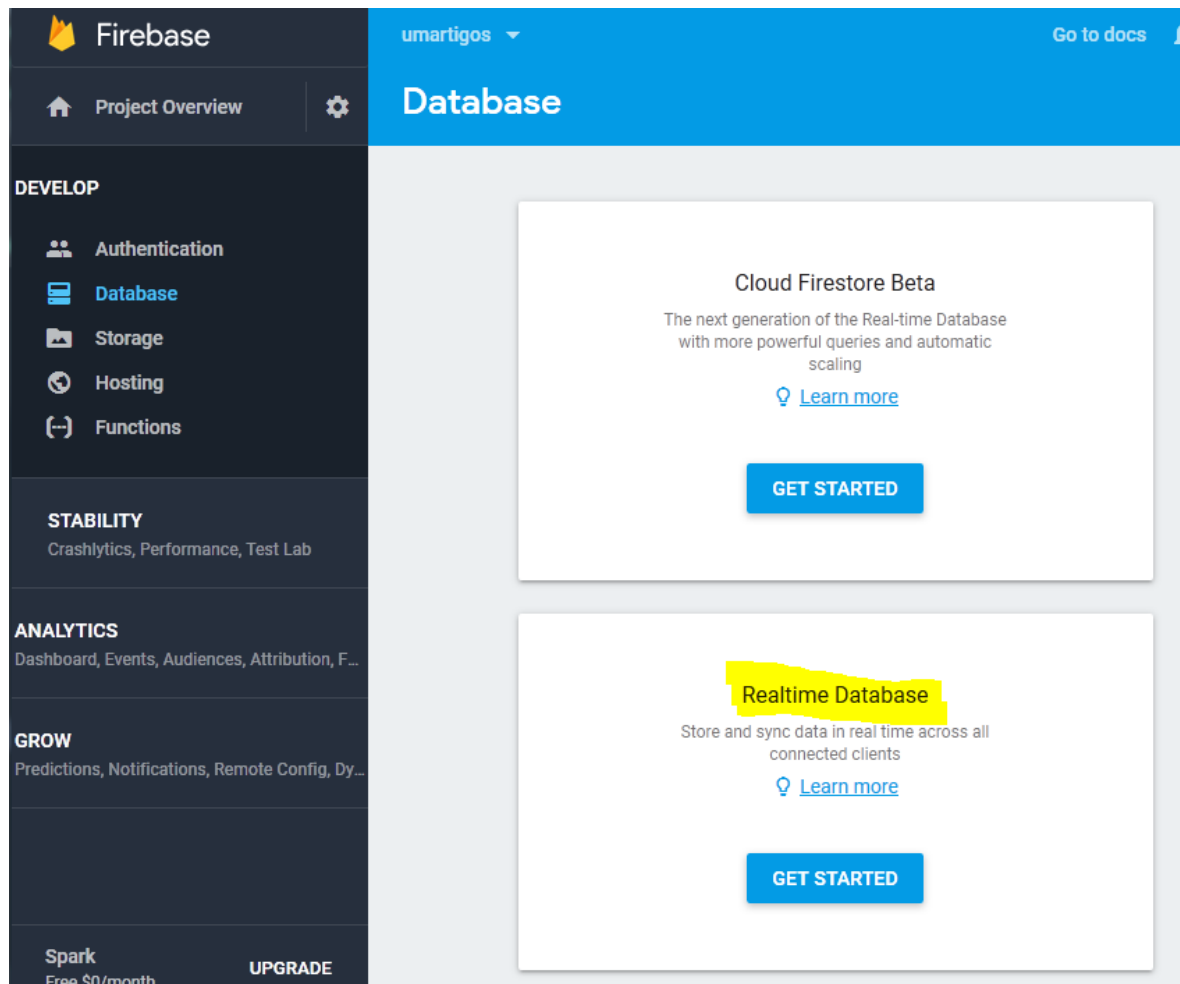
Crie uma conta, caso ainda não tenha uma conta Google e aceda à consola do Firebase. Inicie um novo projeto com uma designação apropriada



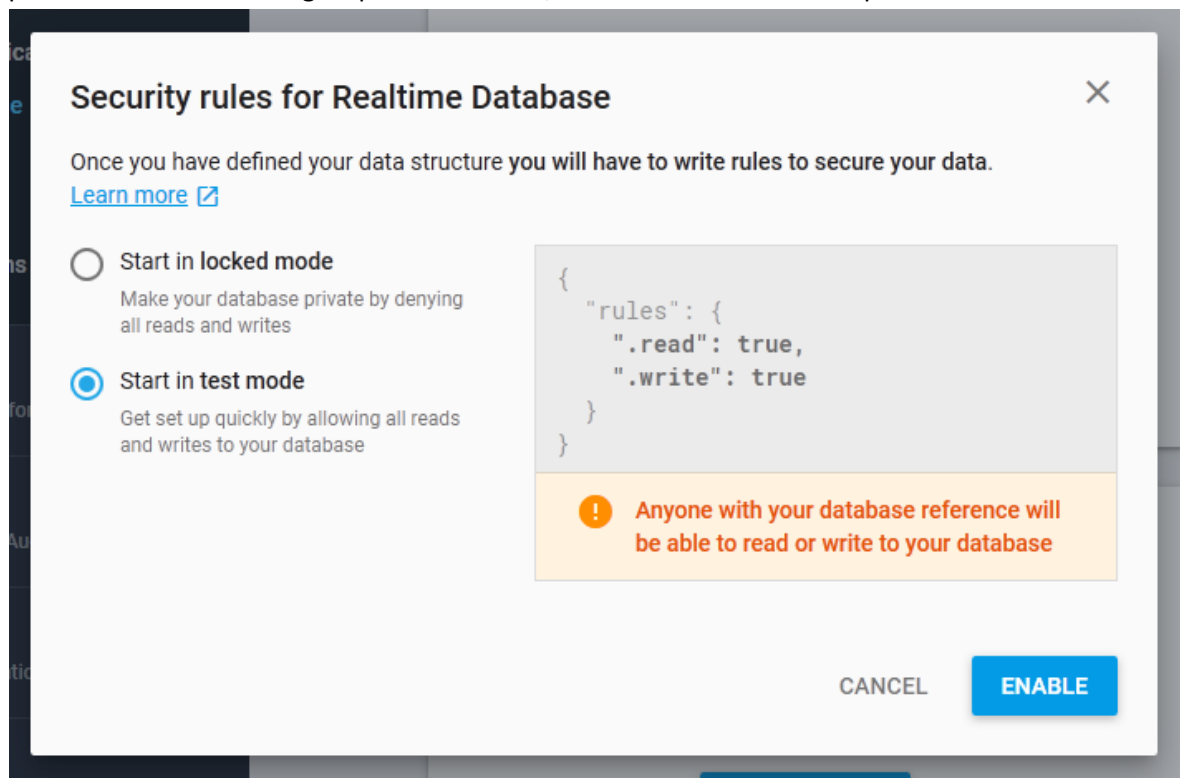
Cofinanciado por:



Das funcionalidades do firebase vamos seleccionar a Base de Dados, mais concretamente a **Realtime Database**



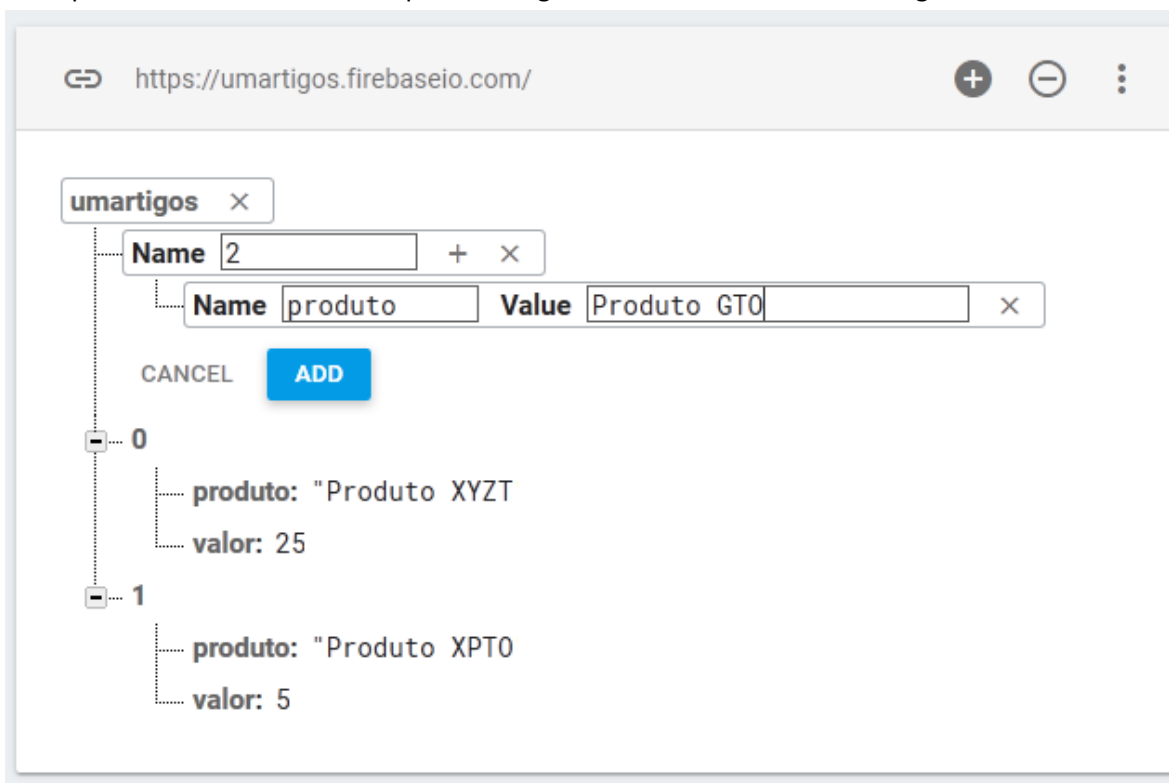
Inicie a base de dados em **test mode**, para não ser necessário definir regras para a autenticação. Caso só pretenda definir estas regras posteriormente, estas encontram-se no separador **rules**



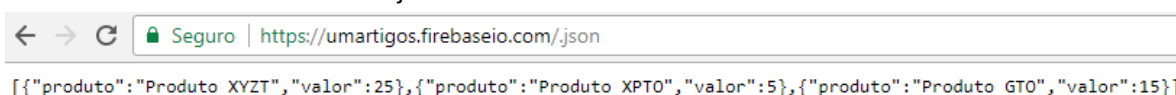
Cofinanciado por:



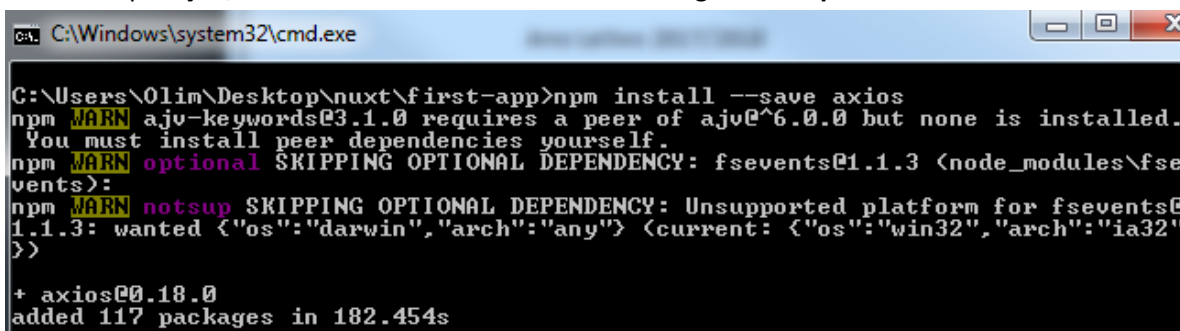
No separador **DATA** da **Database** podemos agora introduzir manualmente alguns dados



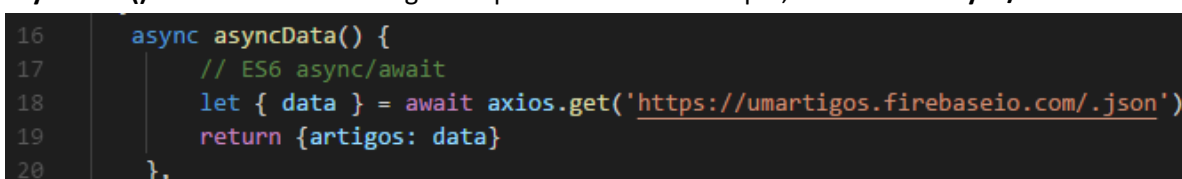
Estes dados já podem ser acedidos na web no endereço apresentado no separador **data**, sendo ainda necessário adicionar a extensão .json



Para a nossa aplicação aceder a dados por http torna-se necessário instalar o **axios** para o efeito. No diretório da nossa aplicação, iniciamos uma consola de sistema e digitamos: **npm install --save axios**



No caso particular do NUXT poderíamos querer carregar estes dados ainda no lado do servidor. Para tal é disponibilizado o método **asyncData** que permite gerir operações assíncronas antes dos dados estarem carregados na página (atenção: só funciona em **pages** e não pode ser usado em **components**). É possível utilizar o **asyncData** de três formas (https://nuxtjs.org/guide/async-data#__layout): **callback**, **promises** e **async/await**. Em qualquer dos casos, eliminamos os nossos dados estáticos de **data()** e adicionamos o **asyncData()** com uma das abordagens supracitadas. Por exemplo, usando **ES6 async/await**:

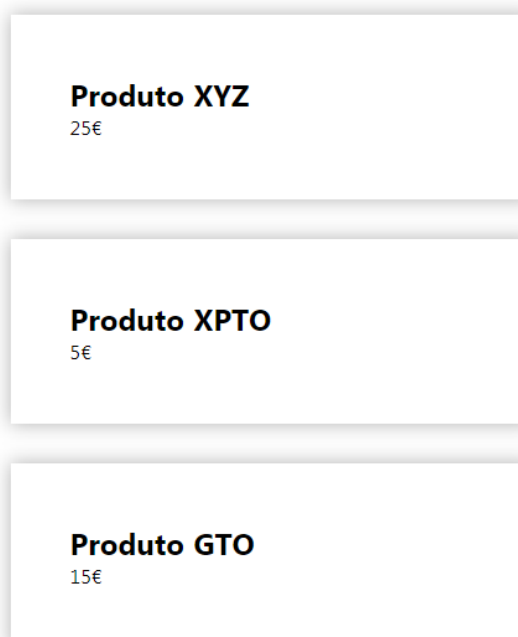


Cofinanciado por:

No caso de uma implementação com promessas:

```
index.vue header.vue users.vue default.vue
1 <template>
2   <div class="app">
3     <app-ficha-produto v-for="(artigo, index) in artigos"
4                       :key="index"
5                       :produto="artigo.produto"
6                       :valor="artigo.valor" />
7   </div>
8 </template>
9 <script>
10 import axios from 'axios';
11 import appFichaProduto from '@components/app-ficha-produto';
12 export default {
13   components: {
14     appFichaProduto
15   },
16   asyncData() {
17     // retornar uma Promessa
18     return axios.get('https://umartigos.firebaseio.com/.json')
19       .then((res) => {
20         return { artigos: res.data }
21       })
22   },
23 }
24 </script>
25 <style scoped>
26 .app{
27   margin-top: 4.5rem;
28   display: flex;
29   align-items: center;
30   flex-direction: column;
31 }
32 </style>
```

Em qualquer caso, a nossa página principal é renderizada da mesma forma (mesmo em SSR):



Cofinanciado por:

