

Programação Visual

Trabalho de Laboratório nº 10

Objetivo	Angular – Introdução à utilização de Angular em projetos ASP.NET Core MVC.
Programa	Pretende-se criar uma aplicação cliente dos serviços da aplicação ESTeSoccer usando angular.
Regras	Implementar o código necessário e testar no fim de cada nível. Use as convenções de codificação adotadas para as linguagens usadas.
Descrição	
Nível 1	<ul style="list-style-type: none">• Descarregue e descompacte o ficheiro “Lab 10 – Materiais” fornecido com este enunciado. Depois de descompactar, abra a solução que lá está. Nessa solução irá encontrar 2 projetos:<ol style="list-style-type: none">1. ESTeSoccer – É uma aplicação MVC que disponibiliza uma API com ações CRUD para as ligas de futebol guardadas na base de dados..2. ESTeSoccerClient – É uma aplicação de consola que gere os dados das ligas de futebol usando os serviços disponibilizados pelo controlador LeaguesAPIController da aplicação anterior.• Para poder usar a base de dados desta solução comece por criar a migração inicial e a base de dados para o projeto ESTeSoccer. Atenção, confirme que o <i>Nuget</i> tem selecionado este projeto antes de executar os comandos.• A aplicação ESTeSoccer usa a base de dados criada e quando for executada pela primeira vez irá colocar alguns dados iniciais. Sendo assim, execute esta aplicação usando a opção Debug-> Start new instance e verifique que está a funcionar. Neste caso, confirme que a utilização dos serviços de consulta das ligas de futebol do controlador LeaguesAPIController retorna a informação das 3 ligas de futebol que lá foram colocadas inicialmente.• Sem parar a execução anterior, arranque com o projeto ESTeSoccerClient da mesma forma e verifique que também está a funcionar. Deverá ver na listagem das ligas, as 3 ligas adicionadas antes.• Pretende-se agora, dentro da solução que está usada, criar uma aplicação Frontend usando a framework Angular. Sendo assim, crie dentro da solução um novo projeto com o nome ESTeSoccerAngularClient do tipo ASP.NET Core Web Application e depois use o <i>template Angular</i>.• Teste a aplicação criada pelo template Angular. Atenção que para esta aplicação funcionar poderá ter que instalar o NodeJs (Se for necessário, encontra este servidor em https://nodejs.org/en/). Nota: Pode demorar algum tempo a iniciar.
Nível 2	<ul style="list-style-type: none">• Pretende-se agora criar uma aplicação cliente semelhante à aplicação de consola existente mas utilizando uma interface desenvolvida em Angular. Comece então por alterar o componente Angular Home para que passe a mostrar o texto “ESTeSoccer” no título e “Gestor de Ligas” no subtítulo. Na barra de menu em cima também deverá aparecer “ESTeSoccer”.• Crie um novo componente Angular com o nome “Acerca” e adicione-o ao projeto. Este componente deve estar associado a uma nova opção do menu do lado esquerdo. Quando selecionado deve mostrar no painel central os nomes dos elementos do grupo que estão a resolver o laboratório.

Programação Visual

Trabalho de Laboratório nº 10

Nível 3

- Precisamos agora de um componente para mais uma opção de menu que mostre as ligas de futebol, usando o serviço correspondente da aplicação **ESTeSoccer**. Uma vez que a utilização de um URL num servidor diferente implica uma mudança de servidor vamos ter de fazer algumas configurações na aplicação Angular:
- Vamos criar um proxy que quando se acede a um determinado prefixo de URL nos redireciona para um URL diferente. Sendo assim crie um ficheiro com o nome **proxyconfig.json** na pasta onde estiver o ficheiro **package.json** do Angular e coloque lá o seguinte código:

```
{
  "/api": {
    "target": "https://localhost:7040",
    "secure": false,
    "changeOrigin": true
  }
}
```

Neste caso, sempre que um URL comece com **/Api** será utilizado o URL <https://localhost:7040/Api> seguido da restante informação. Para que este proxy seja usado deve ainda acrescentar no arranque do servidor do Angular, comandos **ng serve**, a opção **--proxy-config proxyconfig.json**. Faça-o na utilização do comando referida no ficheiro **package.json**.

- Agora já pode utilizar o serviço que devolve as ligas de futebol existentes. Para isso, defina um novo componente **Leagues** que irá corresponder a uma nova opção de menu com o título Ligas. Sugestão: baseie-se no código do componente **FetchDataComponent**.
Notas: Dentro do código do componente Leagues, na interface **League** das ligas não necessita de incluir a propriedade para as equipas e não necessita de receber o BASE_URL, basta usar apenas `get<League[]>('api/leaguesapi')`.
- Depois de adicionar a opção de menu para a listagem das ligas e desta estar a funcionar elimine as opções **Fetch-Data** e **Counter**.

Nível 4

- Crie agora opções para mostrar uma liga a partir do id da liga e criar uma liga.

Nível 5

- Para completar a aplicação crie as opções que faltam para apagar e editar uma liga.

Notas

Para os identificadores do C# siga as convenções adotadas, nomeadamente:

- A notação camelCase para o nome das variáveis locais e identificadores privados.
- A notação PascalCase para os nomes públicos dos métodos e classes
- Não utilize o símbolo de qualquer forma '_' nos identificadores
- Não use abreviaturas