

## Módulo 2 - PHP | Exercício 1 – AVILA Crew Website

### Introdução à arquitetura de um website em PHP

#### Temas abordados

- Template de Twitter Bootstrap
- Identificação de módulos funcionais e construção de componentes
- Otimização adicional com *helpers*
- Criação de páginas

#### PARTE 1 – INSTALAÇÃO E PUBLICAÇÃO

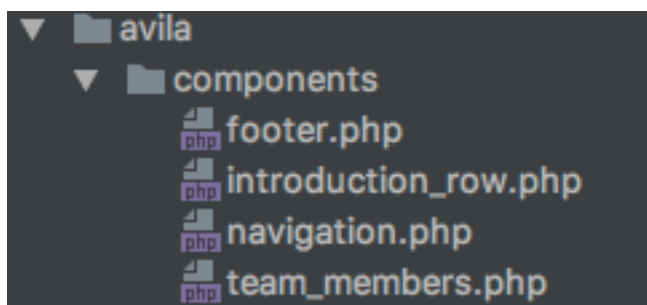
Pretende-se implementar um website em PHP a partir de um template de bootstrap e proceder ao alojamento num servidor Web.

1. Faça download do *template* de Twitter Bootstrap disponível em <https://startbootstrap.com/template-overviews/round-about/>
2. Descomprima o ficheiro zip e coloque essa pasta no seu diretório local de trabalho. Altere o nome da pasta para “avila”. Se estiver a trabalhar num servidor local deve utilizar o diretório Web no servidor Web local (“htdocs” no XAMPP).
3. Altere o nome do ficheiro “index.html” para “index.php”.
4. Publique no servidor Web e experimente a página Web através do browser.

#### PARTE 2 - COMPONENTES

Nesta parte pretende-se otimizar o código do *template* em componentes que poderão facilmente ser adicionados ou removidos das diversas páginas que vão constituir o website. Os componentes são partes lógicas do Website com conteúdos visíveis para o utilizador final.

5. Dentro da pasta “avila” crie uma pasta “components”. (Se preferir pode utilizar designações em Português)
6. Dentro da nova pasta deve criar 4 ficheiros php de acordo com o visível na imagem seguinte.



7. Para cada um dos componentes deve eliminar os conteúdos de php do ficheiro e copiar as partes correspondentes do código existente na página “index.php”.
  - a. “**navigation.php**” – a tag <nav> e todos os conteúdos associados;
  - b. “**introduction\_row.php**” – a linha do Bootstrap (“row”) correspondente à primeira parte dos conteúdos da página;
  - c. “**team\_members.php**” - a linha do Bootstrap (“row”) correspondente à listagem dos membros da equipa;
  - d. “**footer.php**” - a tag <footer> e todos os conteúdos associados. Edite posteriormente a informação do footer para conter o seu nome e identificação da UC.
8. Na página “index.php”, substitua os blocos de código copiados no ponto anterior por chamadas ao método “include\_once” do PHP. No final deve ter algo semelhante ao apresentado na imagem seguinte:

```
<body>
  <!-- Navigation -->
  <?php include_once "components/navigation.php"; ?>

  <!-- Page Content -->
  <div class="container">

    <!-- Introduction Row -->
    <?php include_once "components/introduction_row.php"; ?>

    <!-- Team Members Row -->
    <?php include_once "components/team_members.php"; ?>

    <hr>

    <!-- Footer -->
    <?php include_once "components/footer.php"; ?>

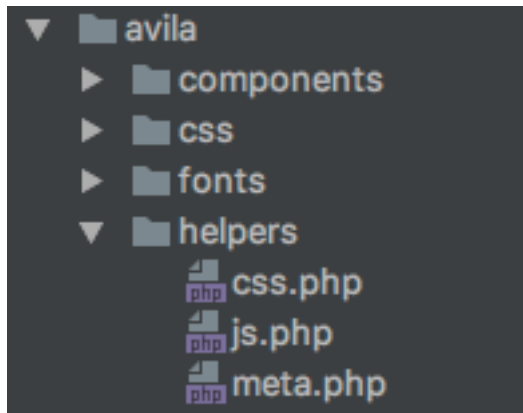
  </div>
  <!-- /.container -->
```

9. Publique as alterações e experimente a página. Não deve existir qualquer tipo de alteração relativamente à versão testada no ponto 4.
10. Altere os conteúdos dos componentes de acordo com o exemplo disponibilizado em [http://labmm.clients.ua.pt/deca\\_17L4/deca\\_17L4\\_p/avila/](http://labmm.clients.ua.pt/deca_17L4/deca_17L4_p/avila/)
11. Publique e verifique as alterações!

## PARTE 2 - HELPERS

Nesta parte pretende-se dar mais um passo na otimização do código do *template* através da utilização de helpers. Neste contexto, define-se um helper como um módulo que não contém conteúdos visíveis para o utilizador final e que, por defeito, são utilizados em todas as páginas do Website.

12. Dentro da pasta “avila” crie uma pasta “helpers”.
13. Dentro da nova pasta deve criar 3 ficheiros php de acordo com o visível na imagem seguinte.



14. Para cada um dos *helpers* deve copiar as partes correspondentes do código existente na página “index.php”.
  - a. “**meta.php**” – todas as linhas de metadados existentes no *head* da página;
  - b. “**css.php**” – todas as linhas de ligação a ficheiros CSS existentes no *head* da página;
  - c. “**js.php**” - todas as linhas de ligação a ficheiros JS existentes no *body* da página;
15. Na página “index.php”, substitua os blocos de código copiados no ponto anterior por chamadas ao método “include\_once” do PHP.
16. Publique as alterações e experimente a página. Não deve existir qualquer tipo de alteração relativamente à versão testada no ponto 11.
17. Num template Bootstrap é natural que precise de efetuar alterações ao CSS base. Essas alterações não devem ser feitas diretamente no ficheiro “bootstrap.css”! Para preparar alterações futuras deve:
  - a. Adicionar um ficheiro “custom.css” na pasta “css”;
  - b. Incluir uma ligação a esse ficheiro no respetivo *helper*.

### PARTE 3 – PÁGINAS ADICIONAIS

Nesta parte pretende-se criar as restantes páginas desta versão inicial do website.

18. Copie o ficheiro “index.php” para criar as páginas “**sessoes.php**” e “**contactos.php**”.
19. Crie um novo componente “**available\_soon.php**” que deve consistir num conteúdo que informa o utilizador que o conteúdo da página estará brevemente disponível. Sugestão: Pode seguir a estrutura base do componente “introduction\_row.php”.
20. Adicione esse componente às páginas anteriores e remova todos os componentes que não são necessários.
21. Altere o componente “navigation.php” para que os elementos do menu liguem às páginas corretas.
22. Publique e teste o resultado!

**T.P.C. – A AVILA CREW MERECE MUITO MAIS!!!**

**A AVILA CREW merece ter um website muito mais bonito. Altere o aspeto gráfico da página a seu gosto, mas não altere a estrutura do HTML porque será a base para os próximos exercícios. Essas alterações devem seguir as boas práticas de alteração de um template de Bootstrap. Não sabe como? E que tal combinar uma “Hora extra” com o NENTC? ;)**