

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

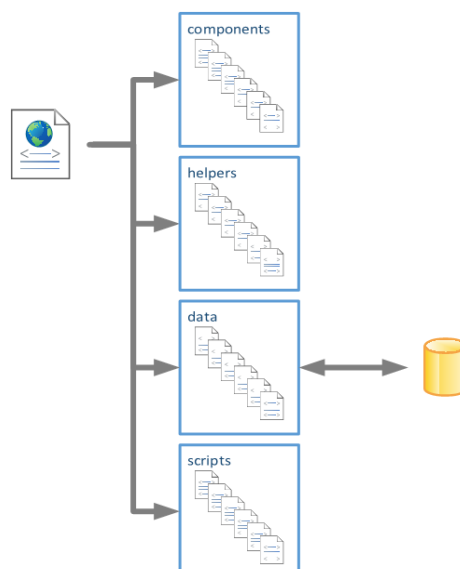
Pretende-se implementar um website em PHP a partir de um template de bootstrap e proceder ao alojamento num servidor Web.

1. Faça download do *template* de Bootstrap disponível em <https://startbootstrap.com/previews/agency/>
2. Descomprima o ficheiro zip e coloque essa pasta no seu diretório local de trabalho (se seguiu o 1º exercício prático da UC, essa pasta tem o nome “web”). Altere o nome da pasta do template para “avila”. Se estiver a trabalhar num servidor local deve utilizar o diretório Web no servidor Web local (pasta com o nome “htdocs”, no XAMPP).
3. Altere a extensão do ficheiro “index.html” para “index.php”.
4. Publique no servidor Web e experimente a página Web através do browser.

Organização do código

O código presente numa página deve ser organizado em pastas correspondentes a cada tipo de recurso (por exemplo, components, helpers, data, scripts, etc.). De maneira a manter tudo organizado e mais fácil de gerir, para além da divisão por pastas, é importante identificar cada recurso com um prefixo correspondente ao respetivo tipo (ex. cp_, help_, data_, sc_, etc.).

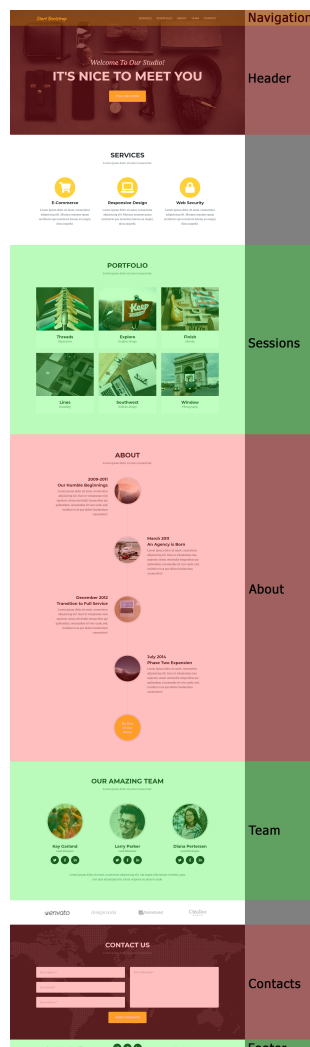
Ver figura seguinte:



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 conteúdo visível 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 os seguintes ficheiros PHP:
 - a. cp_navigation.php
 - b. cp_header.php
 - c. cp_about.php
 - d. cp_team.php
 - e. cp_contacts.php
 - f. cp_sessions.php
 - g. cp_footer.php
7. Os ficheiros anteriores utilizam o prefixo “cp_”, para facilitar a sua identificação no código que mais tarde iremos escrever. Cada um destes componentes corresponde a uma das áreas da página do template, tal como identificado na imagem seguinte.



8. Nesta parte pretende-se separar por componentes o código HTML que consta na página inicial do template. Para cada um dos ficheiros dos componentes, comece por apagar o código PHP que possa ter sido gerado automaticamente pelo seu IDE. De seguida, no ficheiro principal do website (index.php), encontre a parte do código HTML correspondente ao componente em questão e copie esse código para o ficheiro do componente correspondente.
- “cp_navigation.php”** – a tag <nav> e todos os conteúdos associados, correspondente à barra de navegação da página;
 - “cp_header.php”** – a tag <header> e todos os conteúdos associados, correspondente à primeira parte dos conteúdos da página;
 - “cp_about.php”** – a tag <section> com id “about” e todos os conteúdos associados, correspondente à linha do tempo da secção “about”;
 - “cp_team.php”** - a tag <section> com id “team” e todos os conteúdos associados, correspondente à linha do tempo da secção “Team”;
 - “cp_contacts.php”** - a tag <section> com id “contact” e todos os conteúdos associados, correspondente ao formulário de contatos do website;
 - “cp_sessions.php”** - a tag <section> com id “portfolio” e todos os conteúdos associados, correspondente portefólio do website. Deve também copiar para este ficheiro toda a área de modais que existe no “index.php”;
 - “cp_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.
9. Na página “index.php”, pretende-se mostrar apenas os componentes “navigation”, “header”, “about” e “footer”. Substitua os blocos de código correspondentes a estes componentes por chamadas ao método “include_once” do PHP, onde deve incluir o nome do ficheiro do respetivo componente. Deve apagar tudo o que estiver a mais no ficheiro “index.php” e, no final, dentro da tag <body> deve ter algo semelhante ao apresentado na imagem seguinte:

```
<body id="page-top">

  <!-- Navigation -->
  <?php include_once "components/cp_navigation.php"; ?>

  <!-- Header -->
  <?php include_once "components/cp_header.php"; ?>

  <!-- About -->
  <?php include_once "components/cp_about.php"; ?>

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

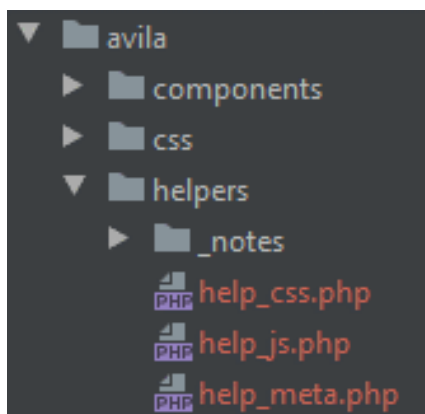
</body>
```

10. Publique as alterações e experimente a página. Deve conseguir ver uma página em tudo idêntica à inicial, mas contendo apenas as secções anteriormente incluídas.
11. Verifique a sua solução comparando com a página inicial do exemplo disponibilizado em https://labmm.clients.ua.pt/deca_21L4/deca_21L4_p/avila/
12. Se a sua solução não estiver correta, faça as alterações que considerar necessárias.

PARTE 3 - 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 omissão, são utilizados em todas as páginas do Website.

13. Dentro da pasta “avila” crie uma pasta “helpers”.
14. Dentro da nova pasta deve criar 3 ficheiros php de acordo com o exemplo que é mostrado na imagem seguinte.
(para mais fácil identificação dos helpers, recomenda-se o uso do prefixo “**help_**”)



15. Para cada um dos *helpers* deve copiar as partes correspondentes do código existente na página “index.php”.
 - a. “**help_meta.php**” – todas as linhas de metadados existentes no **head** da página;
 - b. “**help_css.php**” – todas as linhas de ligação a ficheiros CSS e fontes de texto existentes no **head** da página. Também pode incluir a definição do favicon do website;
 - c. “**help_js.php**” - todas as linhas de ligação a ficheiros JS existentes no **body** da página;
16. Na página “index.php”, substitua os blocos de código copiados no ponto anterior por chamadas ao método “include_once” do PHP.

17. Publique as alterações e experimente a página. Não deve existir qualquer tipo de alteração relativamente à versão testada no ponto 11.

```
<head>

<?php include_once "helpers/help_meta.php"; ?>

<title>AVILA Crew 2019</title>

<?php include_once "helpers/help_css.php"; ?>

</head>

<body id="page-top">

<!-- Navigation -->
<?php include_once "components/cp_navigation.php"; ?>

<!-- Header -->
<?php include_once "components/cp_header.php"; ?>

<!-- About -->
<?php include_once "components/cp_about.php"; ?>

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

<?php include_once "helpers/help_js.php" ?>

</body>
```

18. Num template Bootstrap é natural que precise de efetuar alterações ao CSS base. Essas alterações não devem ser feitas diretamente no ficheiro “styles.css”. Faça as seguintes alterações:

- Na pasta “css”, verifique que tem o ficheiro “styles.css”;
- Adicione um novo ficheiro “**custom.css**”, onde poderão ser colocados novos estilos ou introduzidas alterações aos existentes.
- Adicione a ligação para este ficheiro no respetivo *helper*.
- Teste o funcionamento adicionando código para, por exemplo, alterar a “line-height” dos parágrafos. Reponha o original.

PARTE 4 – PÁGINAS ADICIONAIS

Nesta parte pretende-se criar outras páginas que serão posteriormente adicionadas ao website.

19. Crie 4 cópias do ficheiro “index.php” e atribua os seguintes nomes:
- team.php
 - sessions.php

- c. `contacts.php`
 - d. `login.php`
20. Para as 3 primeiras novas páginas, altere o conteúdo para carregar o componente correspondente. Deve apagar os componentes “cp_header” e “cp_about” porque só fazem parte da página “index.php”;
 21. Crie um novo componente “**cp_available_soon.php**” que deve consistir num conteúdo que informa o utilizador que o conteúdo da página estará brevemente disponível.
Sugestão: Pode reutilizar o código do componente “cp_header”, alterando a mensagem que é dada ao utilizador;
 22. Altere a página “login.php” para mostrar o componente criado anteriormente;
 23. Altere o componente “cp_navigation.php” para incluir ligações para as novas páginas, pela seguinte ordem: “Equipa”, “Sessões”, “Contatos” e “Entrar”. Garanta que o logótipo da equipa faz ligação para a página “index.php”;
 24. Publique e teste o resultado!
 25. Desenhe o esquema com a estrutura do site, que inclua páginas, componentes e helpers, bem como as ligações entre eles.

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.

Algumas indicações:

- Altere a cor de fundo da barra de navegação para que tenha sempre a mesma cor mais escura;
- Para o componente “cp_header” escolha uma imagem de uma sessão da AVILA Crew e seja criativo no *copy*;
- No componente “cp_about” coloque algumas referências históricas sobre a equipa, por exemplo, quando começou, os temas de cada ano letivo, ...
- No componente “cp_team” coloque informação sobre um mínimo de 3 mentores. Pode escolher os que conhecer melhor;
- No componente “cp_sessions” coloque informação sobre um mínimo de 3 AVILA Sessions. Não se esqueça da informação a abrir nas janelas modais.