

Páginas Web & Publicação

Introdução Engenharia Informática

Mário Antunes

November 17, 2025

Exercícios

Guia Prático: Construir & Publicar uma Página Web Estática

Este guia irá acompanhá-lo por todo o processo de criação de um site estático simples, começando de um único ficheiro HTML e construindo-o progressivamente. Terminaremos por servir este site a partir de um servidor web Nginx de nível profissional, a correr dentro de um contentor Docker.

Passo 0: Instalar Software Necessário (Debian)

Precisamos de configurar o nosso sistema Debian (ou baseado em Debian, como o Ubuntu). Precisamos de um editor de texto, do Docker e do seu plugin 'compose'.

1. **Atualizar Listas de Pacotes:** Abra um terminal e execute:

```
sudo apt update; sudo apt full-upgrade -y; \
sudo apt autoremove -y; sudo apt autoclean
```

2. **Instalar um Editor de Texto:** Recomendamos o Visual Studio Code pelas suas funcionalidades. Alunos a usar WSL podem instalar o VSCode no Windows e conectá-lo à VM.

```
# Instalar pré-requisitos
sudo apt install curl wget gpg
wget -qO- [https://packages.microsoft.com/keys/microsoft.asc](https://packages.microsoft.com/keys/microsoft.asc) | gpg --dearmor > microsoft.gpg
sudo install -D -o root -g root -m 644 microsoft.gpg \
/usr/share/keyrings/microsoft.gpg
rm -f microsoft.gpg
```

Crie um ficheiro `/etc/apt/sources.list.d/vscode.sources` com o seguinte conteúdo para adicionar uma referência ao repositório de pacotes upstream:

```
Types: deb
URIs: [https://packages.microsoft.com/repos/code](https://packages.microsoft.com/repos/code)
Suites: stable
Components: main
Architectures: amd64,arm64,armhf
Signed-By: /usr/share/keyrings/microsoft.gpg

# Instalar o VS Code
sudo apt install apt-transport-https
sudo apt update
sudo apt install code # ou code-insiders
```

Os Exercícios

Crie uma pasta para o seu projeto.

```
mkdir o-meu-portfolio
cd o-meu-portfolio
```

Se instalou o VS Code, abra esta pasta escrevendo code .

Passo 1: A Sua Primeira Página “Olá, Mundo”

Crie o seu primeiro ficheiro HTML:

```
nano index.html
```

Dentro do editor, escreva:

```
<h1>Olá, Mundo!</h1>
```

Guarde o ficheiro (no nano, pressione Ctrl+O, Enter, e depois Ctrl+X). Encontre este ficheiro index.html no seu explorador de ficheiros, clique com o botão direito nele e abra-o com o Firefox. Verá o seu texto! O “URL” no seu browser será um caminho de ficheiro local (ex: file:///home/seu-utilizador/o-meu-portfolio/index.html).

Passo 2: Adicionar um Esqueleto HTML5 Real

Um ficheiro HTML real precisa de um “esqueleto”. Substitua o conteúdo do index.html por isto:

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>O Meu Portfólio</title>
</head>
<body>
  <h1>Olá, Mundo!</h1>
</body>
</html>
```

- <head>: Contém informação “meta” sobre a página.
- <body>: Contém o *conteúdo visível* da página.
- <meta charset="UTF-8">: Garante que todos os caracteres de texto (como emojis) são exibidos corretamente.
- <meta name="viewport">: Garante que a sua página tem bom aspeto em dispositivos móveis.

Guarde e atualize o seu browser. A página parecerá a mesma, mas agora o separador (tab) do seu browser dirá “O Meu Portfólio”.

Passo 3: Adicionar Estrutura Semântica

“Semântico” significa “com significado”. Usamos tags HTML que descrevem o seu conteúdo. Isto é ótimo para acessibilidade e SEO.

Substitua a secção <body> por isto:

```
<body>
  <header>
    <h1>O Meu Portfólio</h1>
    <nav>
      <a href="index.html">Início</a>
    </nav>
  </header>

  <main>
    <h2>Bem-vindo</h2>
    <p>Este é o conteúdo principal do meu site.</p>
  </main>

  <footer>
    <p>&copy; 2025 O Seu Nome</p>
```

```
</footer>
</body>
```

Atualize o seu browser. Agora tem um cabeçalho, uma área de conteúdo principal e um rodapé.

Passo 4: Adicionar Conteúdo “Sobre Mim”

Vamos adicionar uma secção apropriada à página. Dentro da sua tag <main>, logo abaixo da primeira tag <p>, adicione um novo <article>:

```
<main>
  <h2>Bem-vindo</h2>
  <p>Este é o conteúdo principal do meu site.</p>

  <article>
    <h3>Sobre Mim</h3>
    <p>Eu sou um estudante de primeiro ano a aprender sobre desenvolvimento web.
    <p>Nos meus tempos livres, gosto de [0 Seu Hobby 1] e [0 Seu Hobby 2].</p>
  </article>
</main>
```

Exercício: Altere o texto de placeholder e preencha os seus próprios passatempos.

Passo 5: Adicionar um Link mailto:

Um link mailto: abrirá o cliente de email padrão do utilizador. Vamos adicionar um ao rodapé.

Edite a sua secção <footer> para ficar assim:

```
<footer>
  <p>Contacte-me: <a href="mailto:seu.email@exemplo.com">seu.email@exemplo.com</a></p>
  <p>&copy; 2025 O Seu Nome</p>
</footer>
```

Guarde e atualize. Clique no link para vê-lo a funcionar.

Passo 6: Ligar um Ficheiro CSS Externo

O nosso HTML está a ficar cheio. É altura de separar os nossos estilos (CSS) para um novo ficheiro.

1. Crie uma pasta css:

```
mkdir css
```

2. Crie um novo ficheiro para os nossos estilos:

```
nano css/style.css
```

3. Adicione uma regra de teste ao css/style.css:

```
body {
  background-color: #f0f0f0; /* Um fundo cinzento claro */
}
```

4. **Ligue o ficheiro CSS.** Abra o index.html e adicione esta linha dentro da secção <head>:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>O Meu Portfólio</title>
  <link rel="stylesheet" href="css/style.css"> </head>
```

Guarde ambos os ficheiros e atualize o seu browser. A sua página deve ter agora um fundo cinzento claro.

Passo 7: Adicionar CSS Básico (Fontes & Cores)

Agora podemos adicionar mais estilos. Abra o `css/style.css` e adicione estas regras:

```
/* Esta regra já cá estava */
body {
    background-color: #f0f0f0;

    /* ADICIONE ESTES */
    font-family: Arial, sans-serif;
    line-height: 1.6;
    color: #333;
}

header h1 {
    color: #004a8c; /* Um bonito azul escuro */
}

a {
    color: #005a9c;
    text-decoration: none; /* Remover o sublinhado */
}

a:hover {
    text-decoration: underline; /* Adicionar sublinhado de volta ao passar o rato */
}
```

Guarde e atualize. A sua página terá agora novas fontes e cores.

Passo 8: Adicionar Estilização de Layout (Box Model)

Vamos fazer com que a nossa página pareça menos “colada” ao lado esquerdo. Vamos adicionar alguns estilos de layout ao nosso ficheiro `css/style.css`. Isto irá centrar o conteúdo em “cartões”.

```
/* Adicione estas novas regras ao fundo do css/style.css */

header, main, footer {
    max-width: 800px; /* Impede o conteúdo de ficar demasiado largo */
    margin: 20px auto; /* 20px cima/baixo, 'auto' esquerda/direita centra-o */
    padding: 20px;
    background-color: #fff;
    border-radius: 8px; /* Cantos arredondados */
    box-shadow: 0 2px 5px rgba(0,0,0,0.1); /* Uma sombra subtil */
}

nav a {
    margin-right: 15px;
    font-weight: bold;
}

footer {
    text-align: center;
    font-size: 0.9em;
    color: #777;
}
```

Guarde e atualize. O seu site tem agora um layout moderno, centrado e baseado em cartões.

Passo 9: Adicionar Media (Imagem)

Uma secção “Sobre Mim” precisa de uma foto.

1. Crie uma pasta `images`:

```
mkdir images
```

2. Encontre uma foto de perfil, nomeie-a `profile.jpg`, e guarde-a na pasta `images`. Se quiser apenas um placeholder, execute este comando:

```
wget "[https://placeholder.co/400x400/004a8c/fff?text=Eu](https://placeholder.co/400x400/004a8c/fff?text=Eu)"
```

3. No `index.html`, encontre o seu `<article>` "Sobre Mim" e adicione a tag ``:

```
<article>
  <h3>Sobre Mim</h3>
  
  <p>Eu sou um estudante de primeiro ano ... </p>
  <p>Nos meus tempos livres, gosto de ... </p>
</article>
```

4. Vamos fazer com que a imagem fique bonita. Adicione isto ao `css/style.css`:

```
article img {
  float: left; /* Faz o texto envolver a imagem */
  margin-right: 15px;
  border-radius: 50%; /* Torna a imagem um círculo */
  width: 150px;
  height: 150px;
}
```

Guarde e atualize. Tem agora uma foto de perfil circular e flutuante.

Passo 10: Adicionar Media (Vídeo)

Vamos adicionar um pequeno clipe de vídeo.

1. Crie uma pasta `videos`:

```
mkdir videos
```

2. Vamos descarregar um pequeno vídeo de amostra gratuito:

```
wget "[https://www.w3schools.com/html/mov_bbb.mp4](https://www.w3schools.com/html/mov_bbb.mp4)"
```

(Este é um clipe de 10MB do "Big Buck Bunny", um vídeo de teste comum).

3. No `index.html`, adicione uma nova `<section>` dentro da sua tag `<main>`:

```
<main>
  <h2>Bem-vindo</h2>
  <section>
    <h3>A Minha Curta-Metragem Favorita</h3>
    <video controls width="100%"> <source src="videos/my-clip.mp4" type="video/mp4" />
    O seu browser não suporta a tag de vídeo.
  </video>
</section>
</main>
```

Guarde e atualize. A sua página tem agora um leitor de vídeo incorporado.

Passo 11: Reorganizar para Publicação (Deployment)

Para uma publicação profissional, precisamos de separar o nosso **código-fonte** da nossa **configuração de publicação**. Vamos colocar todos os nossos ficheiros públicos do site numa única pasta `src`.

1. Crie a pasta `src`:

```
mkdir src
```

2. Mova todos os ficheiros do seu site para dentro dela:

```
mv index.html src/  
mv css src/  
mv images src/  
mv videos src/
```

A estrutura do seu projeto deve agora parecer-se com isto:

```
o-meu-portfolio/  
├── src/  
│   ├── index.html  
│   ├── css/  
│   │   └── style.css  
│   ├── images/  
│   │   └── profile.jpg  
│   └── videos/  
│       └── my-clip.mp4
```

Importante: Volte ao seu browser e atualize. **A sua página está agora partida (broken)!** Porquê? Porque o caminho `file:///` está errado. Não há problema! Estamos prestes a corrigir isto de vez, executando um servidor real.

Passo 12: Publicar com Docker e Nginx

Este é o passo final. Vamos servir a nossa pasta `src` usando um servidor web Nginx real.

1. Na raiz da sua pasta `o-meu-portfolio` (a que *contém* a pasta `src`), crie um novo ficheiro chamado `docker-compose.yml`:

```
nano docker-compose.yml
```

2. Adicione esta “receita” ao ficheiro:

```
services:  
  webserver:  
    image: nginx:alpine  
    container_name: o_meu_portfolio_site  
    ports:  
      - "8080:80"  
    volumes:  
      - ./src:/usr/share/nginx/html
```

O que isto significa:

- `image: nginx:alpine`: Usa o servidor web Nginx oficial e pequeno.
- `ports: - "8080:80"`: Mapeia a porta **8080** no nosso computador para a porta **80** no contentor.
- `volumes: - ./src:/usr/share/nginx/html`: Esta é a magia. Diz ao Docker para pegar na nossa pasta `./src` e torná-la disponível *dentro* do contentor em `/usr/share/nginx/html`, que é a pasta exata de onde o Nginx serve os ficheiros.

3. **Execute-o!** Certifique-se de que está no diretório `o-meu-portfolio` (aquele com o `docker-compose.yml`). Execute:

```
docker compose up -d
```

(`-d` significa “detached”, para que corra em segundo plano).

4. **Visite o Seu Site Ativo!** Abra o seu browser e vá a:

```
http://localhost:8080
```

Deverá ver o seu site completo e estilizado, com imagens e vídeo, a ser servido por um servidor web Nginx de nível profissional!

Para parar o servidor, basta executar:

`docker compose down`

Parabéns!

Conseguiu com sucesso:

- Configurar um ambiente Debian com ferramentas de desenvolvimento modernas.
- Escrever uma página web HTML5 válida e semântica.
- Adicionar conteúdo de texto, um link `mailto:`, imagens e um vídeo.
- Estilizar essa página com um ficheiro CSS externo e responsivo.
- Empacotar e publicar (deploy) o seu site dentro de um contentor Docker usando Nginx.

Este é o fluxo de trabalho fundamental para o desenvolvimento web estático moderno.