

Ficha 1 – Markdown

Esta ficha tem como objetivo a exploração da tecnologia Markdown, a escrita de documentação tecnológica e a publicação da mesma utilizando o GitHub pages.

1 Markdown

O Markdown é uma linguagem de markup, à semelhança do HTML, com o objetivo focado na formatação de texto. Este é usado globalmente na escrita de ficheiros readme, na escrita de documentação de ferramentas tecnológicas, e também como conteúdo em páginas estáticas de muitos sites.

Na sua essência o markdown é traduzido para HTML, e é daí que advém a sua grande adoção. Cada ficheiro de markdown deve possuir a extensão **.md** ou **.markdown**.

Curiosidade:

Para além de ser a fonte de toda a documentação em repositórios **Git** (nomeadamente **GitHub**, Gitlab, Bitbucket, entre outros), também o **Jira** e o **Trello** possuem suporte para **markdown** nas descrições das suas *cards*!

Editores de markdown

Existem vários editores que permitem escrever markdown localmente e ver a sua preview em tempo real. Entre eles o **Visual Studio Code** (<https://code.visualstudio.com/>), o **Typora** (<https://typora.io/>), e o **Caret** (<https://caret.io/>).

1.1 Exercícios

1. Faça todos os exercícios contidos no tutorial *open source* da Common Mark (<https://commonmark.org/help/tutorial/>), clicando para começar em “Begin Lesson”. Estes cobrem todas as funcionalidades do markdown e permitem ganhar alguma prática no mesmo. Faça isso até ao capítulo “The End”.

Nota 1: durante os exercícios ative periodicamente a opção “Show Generated HTML” de forma a ver que elementos web são criados.

Nota 2: durante a execução dos mesmos, para efeitos de consulta, é recomendado verificar este guia no caso de dificuldades: <https://commonmark.org/help/>.

2. Explore os editores listados acima e instale o que achar mais agradável para utilização futura.

2 Docsify

O Docsify (<https://docsify.js.org/>) é um gerador de sites de documentação que utiliza ficheiros de markdown para conter e formatar o seu conteúdo.

Este interpreta os ficheiros de markdown em tempo de execução, possui pesquisa em todo o texto embutida, múltiplos temas, *plugins*, internacionalização e muito mais.

2.1 Instalar o node

Verificar se o nodejs está instalado no computador. Ao correr o seguinte comando no terminal, deverá obter como resposta a versão do mesmo, se instalado:

```
> node -v  
v12.14.1
```

Figura 1 – Verificação da versão do node

Caso não obtenha a versão do mesmo na resposta, deve fazer download do mesmo e instalá-lo através do seguinte site: <https://nodejs.org/>.

2.2 Instalar o docsify

Para instalar o docsify, apenas é necessário executar o seguinte comando:

```
~ > npm i docsify-cli -g
```

Figura 2 – Instalação do docsify

2.3 Exercício

1. Usando um editor de markdown, crie o ficheiro “**README.md**” (o nome do ficheiro deve estar obrigatoriamente em maiúsculas) com o seguinte conteúdo:

FP-Markdown

Nome do Estudante

Numero do estudante

Linguagens de programação conhecidas:

- C
- Javascript
- Python
- ...

Metodologias de Desenvolvimento de Software @ 2020



Figura 3 – Ficheiro readme.md

2. Crie um repositório no Github com o nome “FP-Markdown”.
3. Faça o clone do repositório para o disco.
4. Dentro do repositório local crie uma pasta com o nome “**docs**” na raiz do repositório.

2.4 Inicializar o Docsify

Para inicializar o docsify na pasta criada, é necessário executar o seguinte comando:

```
~/FP-Markdown > docsify init ./docs
```

Figura 4 – Inicialização do docsify

Após inicializado o docsify, a pasta docs irá conter dois novos ficheiros visíveis:

- **index.html** – ponto de entrada para o site gerado;
- **README.md** – ficheiro de conteúdo por omissão (podem ser adicionados mais).

2.5 Servir o Docsify

Para servir o docsify localmente, é necessário executar o seguinte comando na raiz do projeto:

```
~/FP-Markdown > docsify serve docs
```

Figura 5 – Servir o docsify local

Ao abrir o url <http://localhost:3000/>, será apresentada a seguinte página:

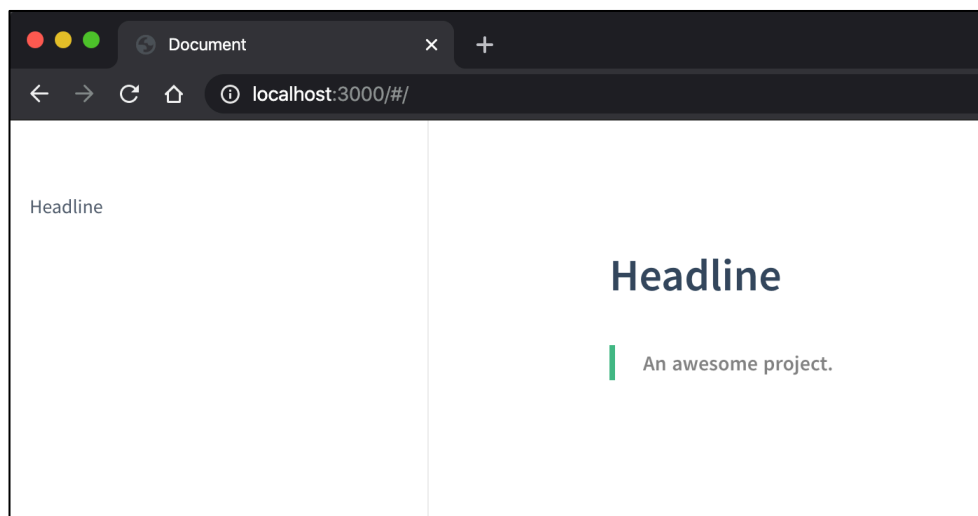


Figura 6 – Páginal inicial vazia

Este representa o ficheiro README.md que foi gerado dentro da pasta docs.

2.6 Exercício

1. Edite no ficheiro index.js o título da página para “FP1 – Markdown”.
2. Substitua o conteúdo do ficheiro README.md gerado, por aquele que foi criado no exercício 2.3.
Nota: assim que o ficheiro for atualizado, a página também o será.

2.7 Adicionar novas páginas

Assumindo a seguinte estrutura:

```
.
└─ docs
   ├── README.md
   └─ sobre.md
```

Estas serão as rotas para aceder às páginas:

- docs/README.md => http://dominio.com
- docs/sobre.md => http:// dominio.com/#!/sobre

2.8 Exercício

1. Crie a página **Sobre** com o seguinte texto “Site criado para aprender a usar Markdown na escrita de documentação”.

2.9 Adicionar novas páginas em subdiretórias

Assumindo que agora se pretende adicionar suporte para a língua inglesa no site, este pode ser feito através da criação de ficheiros .md dentro de uma pasta com o nome “**en**”.

```
.
└─ docs
   ├── README.md
   ├── sobre.md
   └─ en
      ├── README.md
      └─ about.md
```

Acesso através das seguintes rotas:

- docs/en/README.md => http://domain.com/#!/en/
- docs/en/about.md => http://domain.com/#!/en/about

2.10 Exercício

1. Crie a versão em inglês das páginas do seu site, criando para isso a pasta “**en**” e os respetivos ficheiros **README** e **about** dentro da mesma.

2.11 Configuração da *sidebar*

Antes de ser criada a *sidebar* esta precisa de ser habilitada no ficheiro index.html de forma a ser feito o *load*. Para tal, adicione **loadSidebar: true** a esse ficheiro, tal como mostra a seguinte imagem:

```

<script>
  window.$docsify = {
    loadSidebar: true
  }
</script>
<script src="//unpkg.com/docsify/lib/docsify.min.js"></script>

```

Figura 7 – Ativar o load da sidebar

De forma a criar a *sidebar*, é necessário criar um ficheiro **_sidebar.md** dentro da pasta docs. Insira o seguinte markdown de forma a que apareçam duas categorias na *sidebar*:

- **Home** – que aponta para o ficheiro README.md;
- **Sobre** – que aponta para o ficheiro sobre.md.

```

* [Home](./)
* [Sobre](./sobre.md)

```



Figura 8 – Página Home final

Para mais informações sobre o docsify visitar: <https://docsify.js.org/#/>.

3 Github Pages

O Github Pages permite que seja hospedado um website estático associado a um repositório ou utilizador. De seguida irá ser abordada a hospedagem do website de documentação criado com o docsify, associando-o ao repositório.

Notas:

- Neste ponto é necessário que a pasta **docs** e os seus ficheiros estejam no repositório do Github.
- É também recomendado que os ficheiros estejam no *branch* master.
- O repositório deve ou ser **público** ou a conta de utilizador **pro** (os alunos do IPL podem pedir em https://education.github.com/discount_requests/new).

3.1 Exercício

1. Ir aos **settings** do repositório e fazer *scroll* até encontrar o tópico relativo ao **GitHub Pages**:

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Save

Theme Chooser
Select a theme to publish your site with a Jekyll theme using the gh-pages branch. [Learn more.](#)

Choose a theme

Figura 9 – Github Pages vazio

2. Selecionar a opção “master branch /docs folder” como *source*:

Source
Your GitHub Pages site is currently being built from the /docs folder in the master branch. [Learn more.](#)

🔗 Branch: master ▾

📁 /docs ▾

Save

Figura 10 – Configuração da source

3. Ir a “<https://username.github.io/nome-do-repositorio/>” para ver o website de documentação publicado.

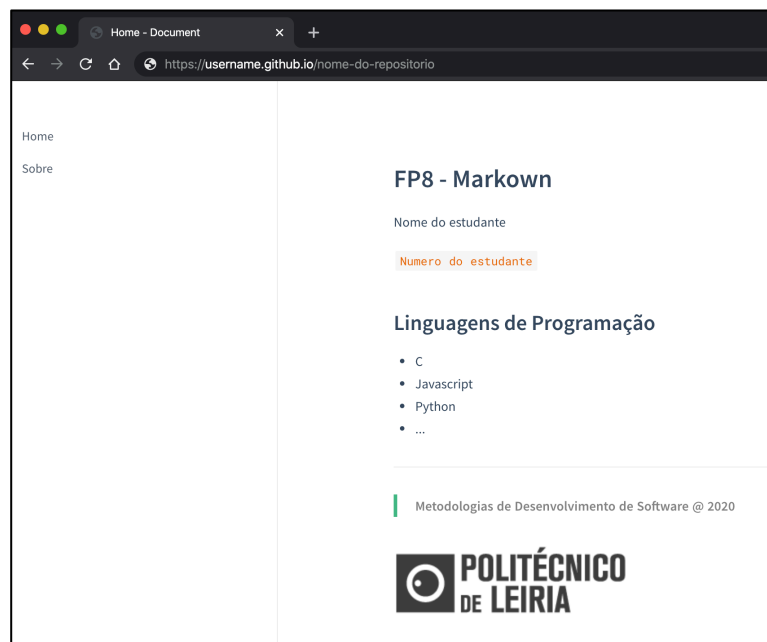


Figura 11 – Website publicado!