



## Laboratórios de Programação

### Guia para Aula Laboratorial 8

Licenciatura em Engenharia Informática

#### Sumário

Geração automática de documentação a partir de comentários no código fonte utilizando a ferramenta Doxygen.

#### Pré-requisitos:

As tarefas propostas a seguir presumem o acesso a um sistema operativo com o *software* Doxygen ou com permissões para a sua instalação. Todas as tarefas foram testadas com sucesso em ambiente Linux mas devem funcionar noutros sistemas, assumindo que as ferramentas necessárias estão instaladas.

## 1 Introdução à Geração Automática de Documentação

### *Automatic Generation of Documentation*

Apesar da criação de documentação técnica de programas ser uma tarefa onerosa e morosa, facilita sobremaneira a sua interpretação por parte de outras pessoas, e permite a exposição do modo de funcionamento do programa sob a forma de um documento estruturado. Existem várias ferramentas que permitem gerar documentos a partir de um conjunto de comentários adicionados ao código fonte de programas, tornando essa tarefa menos onerosa e simples de gerir. Algumas delas permitem inclusive gerar documentação sem que o código esteja comentado, embora de uma forma mais limitada. Este guia procura explorar precisamente uma dessas ferramentas.

O gerador de documentação automático a utilizar neste guia é a ferramenta Doxygen, que suporta essa funcionalidade para implementações em C, C++, PHP, Python, entre outras.

#### Tarefa 1 *Task 1*

O Doxygen pode ser instalado seguindo as instruções de instalação na seguinte *hiperligação*. Por exemplo, nas distribuições baseadas em Debian e Arch Linux, o Doxygen

## Programming Laboratories

### Guide for Laboratory Class 8

Degree in Computer Science and Engineering

#### Summary

*Automatic generation of documentation from comments in the source code using the Doxygen tool.*

pode ser instalado utilizando, respetivamente, os comandos `sudo apt-get install doxygen` e `sudo pacman -S doxygen`

Assim, esta primeira tarefa consiste em verificar se o Doxygen está instalado no seu sistema, emitindo o comando `doxygen` no terminal e analisando o seu *output*.

Caso o sistema devolva um erro de *Command not found*, considere a instalação da ferramenta antes de evoluir no guia.

## 2 Geração de Documentação

### *Generating Documentation*

O Doxygen suporta vários métodos para gerar documentação, partindo de comentários no código. Dois modos de comentários que permitem a sua inclusão na documentação, em C, são:

```
/**  
... Comments ...  
*/
```

ou

```
/// Comments
```

**Repare no asterisco extra ( \* ),** na primeira linha do primeiro modo, e **na barra extra ( / ),** no segundo modo. Esta sintaxe é utilizada para que o

Doxygen reconheça que o comentário inserido no código necessita de ser extraído e devidamente processado.

As duas sintaxes previamente descritas devem ser **adicionadas antes de uma entidade**. Uma entidade no Doxygen pode ser, por exemplo, uma variável global, uma estrutura ou uma função.

### Tarefa 2 Task 2

Crie uma nova diretoria chamada `program1` com as seguintes sub-diretorias: `docs` e `src`. Na sub-diretoria `src` crie um ficheiro chamado `program1.c` e adicione o seguinte código:

```
/**
 * @file program1.c
 */
#include <stdio.h>

int a = 20;

int makeSum(int b){
    return(a + b);
}

int main(){
    int b = 20;
    int sum = makeSum(b);
    printf("%d+%d=%d\n", a, b, sum);
    return 0;
}
```

De seguida, crie o ficheiro de configuração que irá conter todos os parâmetros necessários para gerar a documentação do seu programa. Para esse efeito, deverá executar, na diretoria `program1`, o comando `doxygen -g`.

**Q1.: Quantos ficheiros foram criados após a execução do comando da tarefa anterior?**

- ☐ Ahh... Adoro perguntas com rasteira. Parece que foi criado só um, mas vou verificar com `$ ls -la`.
- ☐ Ahh... Adoro perguntas com rasteira. Parece que foi criado só um, mas vou verificar também dentro da diretoria `docs` que a mim ninguém m'engana!
- ☐ 0      ☐ 1      ☐ 2      ☐ 3      ☐ 4      ☐ 42

**Q2.: Qual foi o nome do ficheiro de configuração gerado pelo Doxygen?**

- ☐ Doxygen   ☐ Doxyfile   ☐ Doxydoc   ☐ Doxyconfig  
☐ Foxy

**Q3.: Qual seria o comando que utilizaria para gerar um ficheiro de configuração com um nome definido por si?**

- ☐ `doxygen <config_file_name>`

- ☐ `doxygen -u <config_file_name>`  
☐ `doxygen -g <config_file_name>`  
☐ `javadoc <config_file_name>`

### Tarefa 3 Task 3

Promova as seguintes alterações no ficheiro de configuração gerado pelo Doxygen (`nano Doxyfile`):

- Altere o nome do seu projeto, alterando a linha seguinte –  
`PROJECT_NAME = "Sum Values"`
- Altere o caminho da diretoria onde serão colocados os ficheiros de documentação gerados pelo Doxygen –  
`OUTPUT_DIRECTORY = "docs/"`
- Defina a diretoria onde se encontram os ficheiros fonte do seu programa –  
`INPUT = "src/"`
- Caso não esteja definido, especifique o tipo de documentação que será gerado –  
`GENERATE_HTML = Yes`

Precura rápida: **Q4.: Como se podem fazer pesquisas no editor de texto `nano`?**

- ☐ Basta fechar os olhos com muita força e pensar no que queremos... o `nano` avança o cursor para esse ponto.
- ☐ Através da combinação de teclas `CTRL+W`.
- ☐ Não dá... também já era pedir demais!

### Tarefa 4 Task 4

Execute, na diretoria `program1`, o comando `doxygen <config_file>` para gerar a documentação do programa `program1.c` e procure os ficheiros de documentação gerados. Caso o ficheiro de configuração tenha o nome por omissão (`Doxyfile`), pode usar apenas o comando `doxygen`.

**Q5.: Quais foram as sub-diretorias criadas na diretoria `docs`?**

- ☐ html   ☐ txt   ☐ png   ☐ latex   ☐ svg

Abra o ficheiro `index.html`, presente na diretoria `docs/html`, utilizando um navegador (`browser`).

**Q6.: Na página Web, onde se encontra a documentação do `program1.c`?**

- ☐ No separador *Files > File List*  
☐ No separador *Main Page*.

- ☐ Na página do Doxygen.
- ☐ Na página *preciso d'oxygen*.
- ☐ Procurei por `program1.c` na barra de pesquisa e encontrei a sua documentação.

**Q7.: Que secções são visíveis na documentação do `program1.c`?**

- ☐ Functions
- ☐ Detailed Description
- ☐ Variables
- ☐ Function Documentation

#### Tarefa 5 Task 5

Adicione o seguinte comentário à função `main` do programa `program1.c`:

```
...
/**
 * This is the main function of the program.
 */
int main() {
...
```

Não se esqueça que os comentários seguindo a estrutura acima sugerida devem ser colocados antes das entidades ao qual se referem. Após efetuar as alterações pedidas deverá executar novamente o Doxygen na raiz do projeto. **Se não efetuar este passo, a documentação do seu projeto não será devidamente atualizada.**

#### Tarefa 6 Task 6

Verifique a documentação gerada, abrindo, novamente, o ficheiro `index.html`. **Q8.: Notou alguma alteração?**

- ☐ Não existem alterações na documentação HTML.
- ☐ Surgiu um novo separador em *Files*, denominado *File Members*.
- ☐ Na página do `program1.c` existe agora uma secção chamada *Function Documentation*. Nesta secção existe o comentário que adicionei, associado à função principal (*i.e.*, `main`).
- ☐ Procurei por `main` na barra de pesquisa e obtive um resultado, coisa que antes não aconteceu!

#### Tarefa 7 Task 7

Altere novamente o `program1.c` e adicione o seguinte comentário à variável `a`:

```
...
/// Global variable with value 20.
int a = 20;
...
```

Verifique, novamente, a documentação gerada. **Q9.: Apercebeu-se de alguma alteração?**

- ☐ Houve algumas alterações mas nada digno de registo.
- ☐ Não, pois a tarefa não pediu para executar o Doxygen.

#### Tarefa 8 Task 8

Execute o Doxygen para gerar a documentação do programa e volte a dar uma vista de olhos.

**Q10.: E agora, já se deu alguma alteração na documentação gerada?**

- ☐ Ainda não. Comentários associados a variáveis não são incluídos na documentação.
- ☐ Agora sim! Na página de `program1.c`, na secção *Variables*, foi agora incluído o comentário que inseri associado à variável `a`. Que MA-RA-VI-(wait for it)-LHA!

#### Tarefa 9 Task 9

Por uma questão de **completude**, não se esqueça de incluir uma pequena descrição à função `makeSum`, seguindo eventualmente a seguinte sugestão:

```
...
/**
 * This function calculates the sum of the
 * parameter with a constant.
 */
int makeSum( int b) {
...
```

## 3 Utilização de Comandos Doxygen

### Using Doxygen Commands

No Doxygen, existem dois níveis de descrição para uma entidade: **breve** e **detalhada**. Os comandos `@brief` e `@details` podem ser utilizados para gerar, respectivamente, uma descrição breve e detalhada.

#### Tarefa 10 Task 10

Adicione uma descrição breve e detalhada ao seu programa `program1.c`. Para este fim, insira os seguintes comentários no ponto certo da implementação:

```
/**
 * @file program1.c
 * @brief Pretty nice program.
 * @details This program is used to sum two
 * values. One of them is a constant.
 */
#include <stdio.h>
...
```

No final, não se esqueça que deve gerar novamente a documentação.

Atente à documentação gerada. **Q11.: Registou alguma alteração?**

- ☐ Estranho, não existem alterações.
- ☐ No separador *Files > File List*, aparece a descrição breve.
- ☐ Na página do `program1.c`, na secção Detailed Description, existe uma descrição breve e uma detalhada do `program1.c`.
- ☐ Caso de polícia, a diretoria da documentação está VAZIA!

### Tarefa 11 Task 11

Existem outros comandos que podem ser utilizados num ambiente Doxygen.

Explore a seguinte *hiperligação* e registe quais os comandos a utilizar para adicionar uma **versão** e **autor** ao seu programa. Adicione o seu nome como autor e "1.0" como versão, usando os comandos registados, na mesma região onde definiu a descrição breve e detalhada do programa. No final, execute o Doxygen.

---

**Q12.: Após realizar a tarefa anterior, quais foram as alterações que identificou na documentação do programa?**

- ☐ Na secção Detailed Description foram adicionados dois novos campos: The Author Name e Version.
- ☐ Na secção Detailed Description foram adicionados dois novos campos: Author e Version.
- ☐ Na secção Detailed Description foram adicionados dois novos campos: Author e Final Version.
- ☐ Não detetei nenhuma alteração à documentação do programa mas também não me lembro se executei o Doxygen...

Os comandos `@file`, `@brief` e `@details`, usados em tarefas anteriores, são exemplos de **comandos especiais** do Doxygen. A sintaxe adotada para este tipo de comandos é a utilização do prefixo `@` ou `\`. Ambos podem ser usados de forma permutável. Alguns exemplos de comandos especiais são exibidos na listagem seguinte:

- `@file`: utilizado para definir o nome do ficheiro. Deve sempre existir no cabeçalho do programa. Se **não incluir** este comando, o ficheiro do programa não será processado pelo Doxygen.
- `@param`: descreve os parâmetros de uma função.
- `@return`: descreve o(s) valor(es) de retorno de uma função.
- `@mainpage`, `@page` e `@subpage`: definem, na documentação, uma página principal, uma página e uma sub-página. As páginas são geradas a partir de ficheiros Markdown.

### Tarefa 12 Task 12

Modifique o ficheiro `program1.c`, adicionando uma descrição dos valores de retorno das funções.

**Q13.: Qual foi o comando especial que utilizou?**

- ☐ `@point`      ☐ `@of`      ☐ `@no`      ☐ `@return`

### Tarefa 13 Task 13

Modifique, novamente, o `program1.c`, adicionando uma descrição dos parâmetros da função `makeSum`.

No final, execute o Doxygen. **Q14.: Qual foi o comando especial do Doxygen que utilizou?**

- ☐ `@return`      ☐ `@function`      ☐ `@param`      ☐ `@voz`

**Q15.: Em que secção da documentação foi apresentada a descrição dos parâmetros passados à função `makeSum`?**

- ☐ Function Documentation.
- ☐ Detailed Description.
- ☐ Todas as anteriores.
- ☐ Nenhuma das anteriores.

## 4 Páginas de Documentação Auxiliares

### Auxiliary Documentation Pages

O Doxygen permite gerar páginas *Web* em HTML a partir de ficheiros Markdown. Estas páginas serão posteriormente integradas na documentação final gerada pelo programa.

A utilização e integração destes ficheiros no Doxygen irá permitir adicionar trechos de código, informações relativas à arquitetura adotada pela aplicação ou outras informações que achar pertinente.

O Doxygen integra ficheiros Markdown em três categorias: **main**, **page**, e **subpage**.

## Tarefa 14 Task 14

Crie, na diretoria `src`, um novo ficheiro chamado `main.md` e adicione o seguinte código:

```
@mainpage The Sum Program
This is the main page of the documentation of
the project. Its contents can be
summarized as follows:
1. Listing of the files implementing the
software;
2. Listing and description of each function of
the program;
3. Indication of reference links.
```

A sintaxe do comando especial `mainpage` é a seguinte: `@mainpage [(title)]`.

### Q16.: Onde será exibido o conteúdo do ficheiro Markdown da tarefa anterior?

- ☐ Na página dos ficheiros (*Files > File List*) do projeto.
- ☐ Na página do `program1.c`, na secção *Detailed Description*.
- ☐ Na página principal (*Main Page*) do projeto.

## Tarefa 15 Task 15

Execute o Doxygen para confirmar a resposta à pergunta anterior.

### Q17.: Dada a sintaxe do comando `mainpage`, o que pode comentar relativamente ao título?

- ☐ É obrigatório.
- ☐ É opcional.
- ☐ Pode não estar presente mas se não estiver o Doxygen não executa. Ou seja, pode não estar mas tem de estar. É uma questão paradoxal!

## Tarefa 16 Task 16

Na diretoria `src`, crie um novo ficheiro chamado `page.md` e insira-lhe o seguinte excerto:

```
@page workflow Page – Description of the
Workflow

# The Workflow of the Program.
This is a text explaining the workflow of the
program.

# The Details of the Workflow.
This is a text explaining something more.
```

No final, execute o Doxygen. Observe que a sintaxe do comando especial `page` é a seguinte:  
`@page <reference_name/label> (Title)`.

### Q18.: Qual é a referência/etiqueta atribuída à página que acabou de criar?

- ☐ `workflow`
- ☐ `workflow Page`
- ☐ `page`
- ☐ `nao_sei`

### Q19.: Verificou alguma alteração na documentação após a criação do ficheiro `page.md`?

- ☐ A informação na página principal (*Main Page*) foi substituída pela informação presente no ficheiro `page.md`.
- ☐ Surgiu um novo separador, intitulado *Related Pages*, com uma hiperligação com o título definido em `page.md`.
- ☐ Foi acrescentado o ficheiro `page.md` na página dos ficheiros (*Files > File List*).
- ☐ Na página do `program1.c`, na secção *Detailed Description*, foi introduzida a informação presente no ficheiro `page.md`.

## Tarefa 17 Task 17

Crie, na diretoria `src`, um novo ficheiro chamado `subpage.md` e adicione o seguinte código:

```
@page sum Subpage – Description of the Sum
Function

# How to Sum Two Numbers
The mathematical explanation of summing two
numbers.
```

No final, execute o Doxygen.

### Q20.: Qual é a referência atribuída à sub-página que acabou de criar?

- ☐ `sum`
- ☐ `sum Subpage`
- ☐ `page`
- ☐ `nao_respondo`

Atente ao separador *Related Pages*, na documentação gerada.

### Q21.: O que pode comentar relativamente à disposição dos ficheiros?

- ☐ Apenas é visível uma hiperligação para *Subpage - Description of the Sum Function*.
- ☐ Apenas é visível uma hiperligação para *Page - Description of the Workflow*.
- ☐ São visíveis duas hiperligações, ambas com a mesma indentação relativamente à margem esquerda.
- ☐ São visíveis duas hiperligações, em que *Subpage - Description of the Sum Function* é uma sub-página de *Page - Description of the Workflow*.

Admita que tinha como objetivo a adição de `subpage.md` como sub-página de `page.md`. Para tal, teria que fazer uso do comando `@subpage`, na página `page.md`.  
**Q22.: Para atingir o seu objetivo, qual era o comando completo a introduzir**

em `page.md`?

- |   |   |
|---|---|
| <input type="checkbox"/> <code>@subpage Subpage</code>  | <input type="checkbox"/> <code>@subpage sum</code>  |
| <input type="checkbox"/> <code>@subpage workflow</code> | <input type="checkbox"/> <code>@subpage Page</code> |

### Tarefa 18 *Task 18*

Tendo em conta a resposta anterior, modifique o ficheiro `page.md` e execute o Doxygen.

Atente, novamente, ao separador `Related Pages`, na documentação gerada.

**Q23.: A tarefa anterior promoveu alguma alteração?**

- ☐ Nenhuma. A tarefa não produziu nenhum efeito na disposição das páginas.
- ☐ Subpage - Description of the Sum Function é uma sub-página de Page - Description of the Workflow.
- ☐ Page - Description of the Workflow é uma sub-página de Subpage - Description of the Sum Function.
- ☐ O separador `Related Pages`, que outrora estava povoado com duas páginas, encontra-se agora desprovido de recursos. Precipitou-se algo de inexplicável e a música dos *X-Files* ocorreu-se-me.

### Tarefa 19 *Task 19*

O Doxygen disponibiliza a opção de adicionar um índice de conteúdos.

Registe o comando que usaria para adicionar um índice de conteúdos ao ficheiro `page.md`. Para tal, use o comando que usaria em  $\text{\LaTeX}$  ou pesquise no manual do Doxygen, usando a seguinte *hiperligação*.

---

Adicione o comando ao ficheiro `page.md`, abaixo de `@subpage sum`, e execute o Doxygen.

Atente ao conteúdo da hiperligação `Page - Description of the Workflow`, no separador `Related Pages`.

**Q24.: Consegue visualizar o índice de conteúdos?**

- ☐ Sim, sem quaisquer problemas!
- ☐ Creio que não. Onde deveria estar o dito índice?

### Tarefa 20 *Task 20*

Modifique o ficheiro de configuração do Doxygen, alterando o conteúdo da variável `TOC_INCLUDE_HEADINGS` para 1. Confirme que a variável `MARKDOWN_SUPPORT` tem como conteúdo `Yes`. No final, execute o Doxygen.

**Q25.: Face a estas alterações, consegue visualizar o índice de conteúdos?**

- ☐ Agora sim, consegui.
- ☐ Raios e coriscos, ainda não foi desta!

### Tarefa 21 *Task 21*

Tendo como exemplo o `Makefile` implementado em aulas anteriores, execute todas as alterações que achar necessárias para automatizar a geração de documentação, após a compilação do programa `program1.c`.