



Título do Documento Relatório do Doxygen		Projeto
Escrito/Atualizado por Eric Sonagli Abbade		Data
Revisado por	Assinatura	Data
Aprovado por	Assinatura	Data

1) Objetivo

Este documento tem como objetivo demonstrar o funcionamento da ferramenta Doxygen que é capaz de gerar documentações de códigos. O grupo de Eletrônica e Microeletrônica (EMI) desenvolve vários códigos em seus projetos, igual a diversos outros grupos do CNPEM.

Documentar códigos é importante por facilitar que futuras mudanças neles sejam feitas, de forma mais ágil. A documentação, possivelmente, também pode evitar que os programas tenham que ser totalmente refeitos.

Portanto, entende-se como necessária o desenvolvimento de documentações do código. Esse relatório tem como objetivo discutir diversos aspectos do Doxygen, para que ele realize essa função. Tais aspectos são:

- Descrição;
- Necessidade de utilização;
- Vantagens e desvantagens;
- Tutorial;
- Conclusão;
- Referências

Com esses cinco tópicos acima, pretende-se gerar um bom aprendizado relativo o Doxygen. Assim, após a conclusão desse relatório, é esperado que o leitor tenha relativamente boas noções sobre quando o Doxygen deve ser usado, suas vantagens e desvantagens e do uso de suas principais funcionalidades.



Descrição

I. Descrição Doxygen (de seu próprio site):

Doxygen é uma ferramenta geradora de documentação amplamente utilizada no desenvolvimento de software. Ele automatiza a geração de documentação a partir de comentários de código-fonte, analisando informações sobre classes, funções e variáveis para produzir saída em formatos como HTML e PDF. Ao simplificar e padronizar o processo de documentação, a Doxygen aprimora a colaboração e a manutenção em diversas linguagens de programação e escalas de projeto. [1]

II. Estrutura Simplificada do Doxygen:

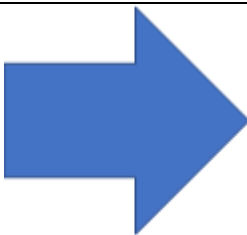
Documentar Arquivos (formatos):	Para ...	Documentos de saída (formatos):
C/C++/C#, D, IDL, Fortran, Java, Lex, Objective-C, PHP, Python, Splice e VHDL. [1]		HTML, PDF (via LaTeX), Word (via RTF) e XML. [1]
Usuário do Doxygen gera a documentação do código desenvolvido por ele		Leitor da Documentação

Tabela 1 – Funcionamento do Doxygen.

2) Necessidade de utilização

Códigos precisam muitas vezes serem atualizados, em um período que frequentemente ultrapassa o prazo inicial do projeto. Tais atualizações, por vezes, chegam a serem feitas por pessoas (tanto de dentro, como fora do CNPEM) que não participaram de sua elaboração inicial.

Além disso, existem casos em que as atualizações estão sendo feitas muito tempo após o desenvolvimento inicial do projeto. Por isso, mesmo que os membros tenham participado da elaboração inicial dos códigos, em vários casos, já se esqueceram de sua estrutura.

Diversos desenvolvedores consideram difícil trabalhar com códigos, sobre os quais não se conhece muito a respeito de suas estruturas. Por causa disso, vários desses códigos acabam sendo totalmente refeitos. Isso ocorre, por causa que é muitas vezes mais fácil escrever programas inteiros, do que entender outros códigos (os quais, por vezes, tem linguagens e bibliotecas não habituais à equipe).

Por tudo isso, é importante a documentação de códigos. Ela pode facilitar a compreensão de um novo código, de modo a reduzir o tempo de execução das manutenções. Além disso, em alguns casos, pode ser um fator decisivo para que a equipe opte por fazer uma manutenção simples no código (devido a sua melhor compreensão), ao invés de o reescrever por inteiro.

É claro que mesmo assim, haverá casos nos quais a equipe irá optar pela reescrita. Isso ocorre, já que muitas vezes, é difícil gerar uma documentação que explique todas as funcionalidades do código, de maneira concisa. Mesmo assim, a documentação ainda é vantajosa em muitos casos, já que, às vezes, evita que o código seja reescrito, além de reduzir o tempo das mudanças.



Todos esses pontos geram a necessidade do desenvolvimento de documentações para os códigos. Apesar disso, existe um custo temporal no desenvolvimento da documentação. Por isso, deve se balancear o desenvolvimento de uma documentação completa e concisa, de modo que ela possa ser feita no menor tempo possível.

É claro que o tempo empreendido para determinada documentação varia conforme características específicas do código. Além disso, diferentes projetos demandam níveis distintos de detalhamento na documentação. Isso ocorre, por exemplo, já que em alguns casos é sabido que está se projetando códigos de operações que podem durar anos e passar por novos desenvolvedores.

Por tudo isso, é importante um planejamento prévio para que se entenda o grau de detalhamento necessário do documento. Com base nesse planejamento, é possível se definir como será feita a documentação. Isso deve ser feito, tendo em mente a relação custo-benefício de gerar uma documentação mais detalhada, mas que demande mais tempo em seu desenvolvimento.

3) Vantagens e desvantagens

Serão apresentadas as principais vantagens, desvantagens, bem como possíveis alternativas para o uso do Doxygen:

I. Vantagens:

- Fácil uso da ferramenta em suas configurações mais básicas;
- Documentação com detalhamento razoável, somente com configurações padrões mencionadas (sem necessitar de comentários específicos nos códigos);
- Gratuito, código aberto e *cross-platform* [1];
- Amplamente usada na área de documentação de software [1];
- Vasta documentação e diversos fóruns de auxílio na internet (devido a sua popularidade);
- Versátil na questão da variedade de formatos de arquivos de saída (Tabela 2).

II. Desvantagens:

- Para uma documentação mais detalhada são necessárias novas configurações, além de comentários específicos nos códigos (o que pode ser trabalhoso);
- Necessita de considerável estudo para o uso de todas suas configurações.

III. Possíveis Alternativas de Softwares para Documentação:

→ **Sphinx (também gratuito):** [Welcome — Sphinx documentation \(sphinx-doc.org\)](http://www.sphinx-doc.org) :

- Citada por diversas referências [2], [3], [4] e, portanto, é uma das principais alternativas;
- Mais focada para Python [2], [4];
- Uso mais complexo [3];
- Gera arquivos de saídas com layouts melhores (segundo alguns usuários) [2], [3];



- A estética da saída é melhorada por instalações extras, conforme o tutorial (que também serve como um tutorial geral do Sphinx): [Auto-Generated Python Documentation with Sphinx \(See comments for update fix\) \(youtube.com\)](#).

→ **Lista com mais 9 opções:** [9 best alternatives to Doxygen as of 2024 - Slant](#)

4) Tutorial

É descrito a seguir um tutorial para o uso rápido do Doxygen, de modo a se usar suas configurações básicas. Apesar disso, ele gera um documento razoavelmente detalhado:

1. Download do Doxygen: <https://www.doxygen.nl/download.html>
2. Download do GraphViz: <https://graphviz.org/download/> (é opcional, mas fortemente recomendado para a criação dos diagramas de função).
3. Acesse o repositório do github: <https://github.com/cnpem-emi/Doxygen/blob/master/README.md> e faça o download de seus arquivos.
4. Para gerar a documentação, entre no diretório \doxygen\bin (sendo doxygen a pasta do download do software do tópico 1) do terminal, digite (substituindo o diretório local do arquivo doxyfile.txt, desse mesmo repositório do github, em seu computador): doxygen C:\Users\eric.abbade\Downloads\doxyfile.txt
5. Substitua os arquivos fontes (de entrada) dos códigos que se deseja documentar e o diretório de saída no qual a documentação será usada (encontre no doxyfile.txt: INPUT e OUTPUT_DIRECTORY).
6. Substitua os diretórios locais do GraphViz para geração dos diagramas de funções (encontre no doxyfile.txt: DIA_PATH e DOT_PATH).
7. Substitua os diretórios locais do logo e arquivo .css (doxygen-awesome, desse mesmo repositório do github) caso deseje maior personalização (encontre no doxyfile.txt: PROJECT_LOGO e HTML_STYLESHEET).
8. Edite outras configurações do doxyfile.txt conforme suas preferências.
9. Acesse a documentação em html/index no diretório de saída definido por você.

Outros tutoriais para que o leitor possa buscar mais formas de adquirir conhecimento sobre essa ferramenta podem ser encontrados abaixo:

- Tutorial, desse mesmo autor, mais detalhado sobre como documentar com o Doxygen: Tutorial Doxygen (sharepoint.com)
- Resumo **(próprio site do Doxygen):** <https://www.doxygen.nl/manual/starting.html>
- Vídeo tutorial com resumo do processo da geração da documentação: <https://www.youtube.com/watch?v=Rl50qI6e7HU>
- Resumo de uma documentação feita anteriormente: SIMAR: File List (cnpem-emi.github.io).



5) Conclusões

Por todos esses pontos destacados anteriormente, pode-se concluir que o Doxygen é uma ferramenta amplamente utilizada para documentação de projetos e é capaz de realizar seu propósito.

O processo de documentação é importante por facilitar que futuras mudanças no código sejam feitas, de forma mais ágil, além da tentativa de evitar que os programas tenham que ser totalmente refeitos. Por isso, é importante o estudo do funcionamento do Doxygen, já que ele é um software de boa qualidade nessa área.

É sugerido o seu uso, para um contato inicial com ferramentas de documentação. Para usos mais específicos, necessita-se uma melhor análise para averiguar a possibilidade da utilização de documentações diferentes.

Um estudo relativamente complexo é necessário, para que se domine todas as configurações do Doxygen. Caso, seja fundamental desenvolver uma **documentação muito detalhada, é preciso configurar diversos parâmetros e fazer comentários específicos nos códigos.** Esse processo é relativamente demorado, e demanda esse estudo prévio mencionado.

Apesar disso, **é possível gerar uma documentação com diversos detalhes a respeito do código somente com as configurações padrões mencionadas,** sem o uso (ou pouco uso) de comentários específicos nos códigos. Logo, é preciso um planejamento para entender a necessidade de detalhes da documentação e traçar a melhor estratégia.

Por tudo isso, podemos concluir que é importante o aprendizado da utilização do Doxygen. Essa conclusão foi tomada, já que essa é uma ferramenta capaz de gerar documentações de códigos com um bom grau de detalhes, sem muitos esforços. Mas, também gera documentos mais detalhados, caso empreendido um maior esforço.

6) Referências

- [1] [Doxygen homepage](#)
- [2] [gratis - A robust and flexible alternative to doxygen supporting C++ - Software Recommendations Stack Exchange](#)
- [3] [What is a viable replacement for Doxygen? : r/embedded \(reddit.com\)](#)
- [4] [Doxygen, What is it? How it works? \(embeddedinventor.com\)](#)