



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 gera documentações de códigos. O grupo de Eletrônica e Microeletrônica (EMI) desenvolve diversos códigos em seus projetos, de forma semelhante a vários grupos do CNPEM.

Esse processo de documentação é importante por facilitar a realização de futuras mudanças nos arquivos, de forma mais ágil. Além disso, possivelmente, pode evitar que os programas tenham que ser totalmente refeitos.

Portanto, entende-se como necessário o desenvolvimento de documentações do código que será feito pelo Doxygen. Este relatório tem como objetivo discutir diversos aspectos do Doxygen, para que ele realize essa função, tais como:

- 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.

Por fim, vale ressaltar, que a **documentação completa desenvolvida sobre o uso do Doxygen está no seguinte repositório do Github: [cnpem-emi/Doxygen \(github.com\)](https://github.com/cnpem-emi/Doxygen)**.



2) Descrição

I. Descrição Doxygen (de seu próprio site):

Doxygen é uma ferramenta geradora de documentação amplamente utilizada no desenvolvimento de softwares. 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:

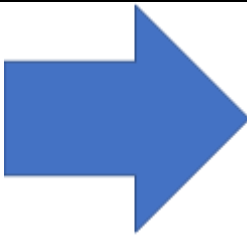
Documentar Arquivos (formatos):	Para ...	Documentos de saída (formatos):
C/C++/C#, D, IDL, Fortran, Java, Lex, Objective-C, PHP, Python, Splice e VHDL. [1]		HTML, PDF (via LaTeX), Word (via RTF) e XML. [1]
Usuário do Doxygen gera a documentação do código desenvolvido por ele		Leitor da Documentação

Tabela 1 – Funcionamento do Doxygen.

3) 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. Também existem casos, nos quais as mudanças serão feitas muito tempo após o desenvolvimento inicial do código, de modo que os membros originais já se esqueceram de sua estrutura.

Diversos desenvolvedores consideram difícil trabalhar com códigos, sobre os quais não tem pleno conhecimento a respeito de suas estruturas. Por isso, vários desses códigos acabam sendo totalmente refeitos. Isso ocorre, já que muitas vezes é mais fácil escrever programas inteiros, do que entender outros códigos (que podem 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 na elaboração da documentação. Por isso, deve se



balancear o esforço para o desenvolvimento de uma documentação completa, com o tempo empreendido. Diferentes projetos demandam níveis distintos de detalhamento na documentação. Por isso, o tempo empreendido para determinada documentação, varia em cada caso. Todas essas análises devem ser feitas para cada cenário, de modo a se otimizar a relação custo-benefício.

Essa documentação pode ser feita por um software que a gere de forma automática (como o Doxygen), ou pela escrita total do programador (como arquivos README.md ou PDFs), ou até mesmo por ambos os métodos. Usar um README.md ou PDF tem a vantagem de permitir que se descreva o código, da forma que se julgue a mais didática possível. Porém, o uso do Doxygen é bem mais rápido. Por isso, deve-se analisar qual estratégia é mais vantajosa, em cada caso. Como o uso do Doxygen é bem mais complexo e precisa de mais instruções, do que a elaboração de um README.md ou PDF, além de ser mais rápido, ele será o foco desse relatório.

4) Vantagens, desvantagens e alternativas

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 as configurações padrões (sem necessitar de comentários específicos nos códigos);
- Gratuito, código aberto e *cross-platform* [1];
- Amplamente usada na área de documentação de software [1], e por isso tem vasta documentação e diversos fóruns de auxílio na internet;
- Versátil na questão da variedade de formatos de arquivos de saída (Tabela 2);
- Pode usar, também, em códigos mal documentados recebidos de outras pessoas.

II. Desvantagens:

- Para uma documentação mais detalhadas 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 \(sphinx-doc.org\)](https://www.sphinx-doc.org/en/master/) :

- Uma das principais alternativas ao Doxygen [2], [3], [4];
- Mais focada para Python [2], [4];
- Uso mais complexo [3];
- Pode gerar arquivos de saídas com layouts melhores [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 \(See comments for update fix\) \(youtube.com\)](https://www.youtube.com/watch?v=...).

→ **Lista com mais 9 opções:** [9 best alternatives to Doxygen as of 2024 - Slant](https://www.slant.co/community-top-picks/item/622/doxygen-vs-sphinx/alternatives)



5) Tutorial

É descrito a seguir um tutorial para o uso rápido do Doxygen, com as suas configurações básicas (9 tópicos). Apesar disso, ele gera um documento significativamente 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. Baixe o doxyfile.txt: <https://github.com/cnpem-emi/Doxygen/blob/master/doxyfile.txt>
4. Substitua os arquivos fontes (de entrada) dos códigos quais se desejam documentar e o diretório de saída no qual a documentação será gerada (encontre no doxyfile.txt: INPUT e OUTPUT_DIRECTORY).
5. Substitua os diretórios locais do GraphViz para geração dos diagramas de funções (encontre no doxyfile.txt: DIA_PATH e DOT_PATH).
6. Opcional: substitua os diretórios locais da logo e do arquivo .css (encontre no doxyfile.txt: PROJECT_LOGO e HTML_STYLESHEET). Exemplo de .css: <https://github.com/cnpem-emi/Doxygen/blob/master/doxygen-awesome.css>.
7. Opcional: edite outras configurações do doxyfile.txt conforme suas preferências.
8. No terminal entre no diretório: \doxygen\bin (sendo doxygen a pasta do software baixado do tópico 1), digite: doxygen C:\Users\eric.abbade\Downloads\doxyfile.txt (substituindo o diretório local do doxyfile.txt do tópico 3 de seu computador).
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).



6) 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 (e relativamente demorado) 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.**

Apesar disso, **é possível gerar uma documentação com diversos detalhes a respeito do código somente com as já mencionadas configurações padrões**, 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.

7) 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? \(embeddedinventor.com\)](#)