

# Ciência, Tecnologia e Cultura na Praça

## VOLUME 2

### Organizadores:

Carlos Eduardo Schuster  
Claudia Aparecida Longatti  
Claudio Luis Vieira Oliveira  
Galileo de Souza Schioser  
Peter Jandl Júnior



Série Ciência na Praça  
ISBN 978-65-5104-062-7



# ***Documentando o Código: boas práticas e ferramentas essenciais***

## **Antony Fernando Ribas Rocha Silva**

Graduando em Tecnologia em Análise e Desenvolvimento de Sistemas, Fatec Jundiaí  
antony.silva3@fatec.sp.gov.br

## **Matheus Henrique Santos e Silva**

Graduando em Tecnologia em Análise e Desenvolvimento de Sistemas, Fatec Jundiaí  
matheus.silva450@fatec.sp.gov.br

## **Thais Priscila Lopes**

Graduanda em Tecnologia em Análise e Desenvolvimento de Sistemas, Fatec Jundiaí  
thais.lopes13@fatec.sp.gov.br

## **Adani Cusin Sacilotti**

Docente do CST em Análise e Desenvolvimento de Sistemas, Fatec Jundiaí  
adani.sacilotti@fatec.sp.gov.br

## **INTRODUÇÃO**

A documentação de código é uma parte fundamental e essencial do processo de desenvolvimento de software. Ela tem o propósito de descrever e explicar o funcionamento do código, bem como fornecer instruções sobre como utilizar adequadamente as funcionalidades implementadas. Uma documentação bem redigida e estruturada desempenha um papel crucial ao ajudar outros desenvolvedores a compreenderem o código, facilitando assim a manutenção do software e promovendo uma colaboração mais eficiente em projetos de equipe. No entanto, nem sempre é uma tarefa fácil para os desenvolvedores criar uma documentação de código clara, precisa e útil. Por isso, este artigo abordará algumas das melhores práticas que podem ser adotadas para produzir uma documentação de código mais eficaz. Vale ressaltar que a qualidade da documentação de código está diretamente ligada à clareza e à organização das informações fornecidas. Portanto, é importante dedicar tempo e esforço para criar uma documentação abrangente, completa e de fácil compreensão. Além disso, a documentação deve ser atualizada regularmente, acompanhando as alterações e evoluções do código-fonte. Dessa

forma, assegura-se que a documentação permaneça relevante e útil ao longo do tempo. Este artigo explora as principais diretrizes e práticas para documentar o código de maneira eficaz, ajudando a melhorar a compreensão, a manutenção e a colaboração no desenvolvimento de software.

## DESENVOLVIMENTO

Foi utilizado, como por exemplo, a linguagem de programação C e o ambiente de desenvolvimento Visual Studio Code para desenvolver diversos códigos na linguagem e fizemos uma análise para documentar o código da melhor forma.

Uma das melhores formas de documentar o código em linguagem C é utilizando a técnica de documentação comentário barra-barra. É importante ressaltar que uma boa prática para documentar é comentar trechos em texto que explicam o que o código faz. Eles devem ser escritos em linguagem clara e concisa e devem descrever a função, o objetivo e a lógica do código. Os nomes de variáveis e funções devem ser significativos e descritivos. Isso ajuda a tornar mais fácil de entender e facilitar a manutenção.

Também podemos fazer a utilização de ferramentas de documentação como javadoc, que é uma ferramenta para a criação de documentação de pacotes, classes, atributos e métodos Java a partir do processamento do código fonte com comentários em formato adequado, também podem ajudar a criar documentação para o código automaticamente, segundo (Luiz, 2000).

Lembrando que a documentação de API também muito utilizada pelos desenvolvedores, e capaz de fornecer informações sobre como usar uma API (*Application Programming Interface*). Isso pode incluir informações sobre os parâmetros de entrada e saída, os métodos disponíveis e exemplos de uso.

A documentação de usuário que é um tipo de documentação que fornece informações sobre como usar um programa. Isso pode incluir instruções passo a passo, exemplos de uso e soluções para problemas comuns que os usuários podem encontrar.

Diagramas de fluxo também são uma forma visual de representar a lógica de um programa. Eles podem ser úteis para entender a estrutura geral do programa e a ordem em que as diferentes partes do programa são executadas.

Documentar um código é uma prática essencial para garantir a compreensão e manutenção de um software ao longo do tempo. Existem diversas modalidades e ferramentas disponíveis para esse propósito, como comentários inline, documentação em arquivos externos, testes automatizados, entre outras. No entanto, é importante lembrar que a escolha da modalidade de documentação deve levar em consideração a necessidade do desenvolve-

dor e do projeto em questão. Por exemplo, se o código for uma biblioteca que será utilizada por outros desenvolvedores, uma documentação completa e bem estruturada é essencial para facilitar o uso da biblioteca e evitar dúvidas ou erros na utilização.

A importância da documentação de código pode ser percebida quando precisamos entender a funcionalidade do código ou realizar alterações necessárias futuramente. A seguir, temos a função `max` que calcula o valor de um elemento máximo de  $v[0..n-1]$ . (Feofiloff, 2018).

**Imagem 1** – Projeto de Algoritmos em linguagem C.

```
int max(int n, int v[]){
    int x = v[0];
    for(int j=1; j<n; ++j)

        if(x < v[j])
            x = v[j];
    return x;
}
```

Fonte: Feofiloff, 2018.

Embora seja fácil para muitos desenvolvedores entenderem o código, pessoas que possuem conhecimento técnico frágil podem ter dificuldades para realizar alterações necessárias sem entender as modificações anteriores no código. Para ilustrar a importância da documentação, vejamos como seria o mesmo código se fosse documentado, usando uma linguagem clara e utilizando a modalidade comentários inline, mais conhecida como barra-barra.

**Imagem 2** – Projeto de Algoritmos em Linguagem C.

```
int max (int n, int v[]) {
    int x = v[0];
    for (int j = 1; j < n; ++j)
        // neste ponto, x é um
        // elemento máximo de v[0..j-1]
        if (x < v[j])
            x = v[j];
    return x;
}
```

Fonte: Feofiloff, 2018.

Ao comparar o código sem documentação com o código documentado, fica evidente o impacto positivo da documentação na compreensão e manutenção do código. Através dos comentários inline, é possível descrever o propósito de cada linha de código, explicar a lógica por trás das operações e fornecer insights adicionais para os desenvolvedores que precisam trabalhar com esse código no futuro.

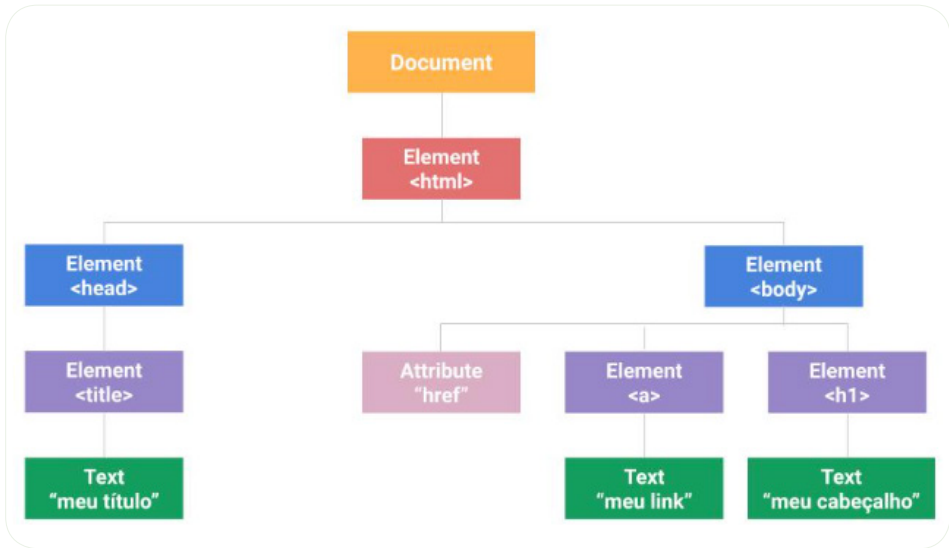
Prosseguindo para documentação JavaScript, ela é importante para páginas web e aplicativos web, pois o JavaScript desempenha um papel fundamental na interatividade e na dinâmica dessas aplicações. Além das práticas gerais de documentação mencionadas anteriormente, aqui estão alguns aspectos específicos a serem considerados ao documentar JavaScript para o desenvolvimento web (Barro, 2023).

Quando esse modelo é alterado através da linguagem de script, se altera a página da web — seja sua estrutura, estilo ou elementos. Consequentemente, o DOM facilita a manipulação e alteração de elementos de um documento da web ao conectar as linguagens de programação à página em questão. Ele é independente em relação à linguagem, então pode ser usado com qualquer uma delas, embora seja mais comum utilizá-lo com o JavaScript.

Dado que toda página da web é um documento, os navegadores precisam ler e interpretar esses documentos para exibir seu conteúdo na tela do seu dispositivo em um formato padrão. O DOM é a interface que representa a forma ou estrutura pela qual o documento será lido e exibido em todos os navegadores da internet.

É desta forma que estabelece um padrão na internet. Essa interface possibilita que programas e scripts de sites sejam acessados de maneira dinâmica para efetuar adições, remoções ou atualizações no conteúdo, na estrutura e no estilo de um documento. O DOM delineia a estrutura lógica dos documentos em formato de árvore, que orienta o navegador na representação e manipulação dos componentes da página. A seguir podemos ver a “Árvore DOM” e como ela é estruturada:

**Imagem 3 – Árvore DOM.**



Fonte: Hostinger Tutoriais (2023).

Sem a linguagem de programação, que pode ser qualquer uma, não apenas JavaScript, não teria um modelo para representar seus objetos. Afinal, ela não teria uma noção da página web e de seus componentes. Portanto, enquanto o DOM representa e organiza os elementos da página, o JavaScript acessa e controla esse conteúdo.

Temos outro tipo de documentação que é a docstrings, que são comentários de documentação inseridos diretamente no código Python. Uma docstring é uma forma de explicar o funcionamento de uma classe, função ou método. É especialmente valiosa para outros desenvolvedores que precisam entender como usar e interagir com seu código.

As docstrings em Python são colocadas entre três aspas duplas (""") na primeira linha de uma classe, função ou método. Elas devem ser concisas, diretas e focadas no propósito da classe ou função. As docstrings também podem conter informações sobre parâmetros, valores de retorno e exemplos de uso. Manter as docstrings atualizadas à medida que o código evolui é fundamental para garantir que a documentação seja precisa e útil (Evandro, 2020).

Vamos utilizar um exemplo prático utilizando docstrings na linguagem Python. Considere que você está desenvolvendo um jogo em Python, onde é necessário implementar os movimentos, a detecção de colisões, o controle de velocidade e outros elementos essenciais. No processo, você cria várias classes e funções para manter seu código organizado e seguir as melhores práticas de programação. No entanto, em algum momento, percebe que al-

gumas dessas funções podem ser úteis em outros projetos e decide criar uma base com uma biblioteca que realize essas tarefas de forma reutilizável. Nesse contexto, surge a necessidade de documentar integralmente essa biblioteca, de modo que outros desenvolvedores que a utilizem compreendam a finalidade e saibam como empregar as funções, classes e métodos disponíveis.

**Imagem 4** – Documentação docstring na Linguagem Python.

```
def _sum(a, b):  
    """  
    Esta função recebe dois parâmetros inteiros  
    para calcular a soma entre eles.  
    :param a: recebe um valor inteiro  
    :param b: recebe um valor inteiro  
    :return: retorna a soma entre eles  
    """  
  
    calc = a + b  
    return calc  
  
print(_sum(8, 10))  
help(_sum)
```

Fonte: DEV Community (2023).

É crucial identificar as áreas que demandam explicações detalhadas para assegurar uma documentação simples, eficaz e de fácil compreensão.

Analisando as linguagens e suas documentações anteriores, chegou a hora de ver a documentação na linguagem de programação PHP. O PHPDoc é uma convenção de comentários de documentação projetada para uso em código PHP. É uma ferramenta que permite a inclusão de informações e metadados diretamente nos comentários do código, tornando-o mais claro e documentado.

A função principal do PHPDoc é facilitar o desenvolvimento em equipe, tornando o código mais compreensível para outros desenvolvedores. Além disso, muitas ferramentas de desenvolvimento e ambientes integrados de desenvolvimento (IDEs) usam o PHPDoc para oferecer recursos adicionais, como preenchimento automático, sugestões de código e verificação de tipos. Em resumo, o PHPDoc é uma prática que aprimora a qualidade da documentação e a colaboração em projetos PHP (Aires, 2023).

A seguir, podemos ver um exemplo utilizando o PHPDoc é um texto de auxílio para ajudar no entendimento da documentação. Os comentários com “@param” descrevem os parâmetros da função. Neste contexto, estamos lidando com três parâmetros:

“texto”: Representa o texto que será resumido.

“limite”: Refere-se ao número máximo de caracteres permitidos no resumo.

“continue” (opcional): Indica o texto que será acrescentado ao final do resumo, sendo que o valor padrão é ‘...’.

Além disso, o “@return” é usado para explicar o valor que a função retornará. Nesse caso, a função fornecerá uma string contendo o resumo do texto.

O uso de “@var” é empregado para especificar o tipo de dados dos parâmetros e do valor retornado. No exemplo mencionado:

“texto” é do tipo “string”.

“limite” é do tipo “int”.

“continue” é do tipo “string”.

**Imagem 5** – Documentação em código PHPDoc.

```
/**
 * Resumir um texto para um limite de caracteres.
 *
 * @param string $texto O texto a ser resumido.
 * @param int $limite O limite de caracteres para o resumo.
 * @param string $continue O texto que será adicionado ao final do resumo (opcional, padrão: '...').
 * @return string O texto resumido.
 */

public static function resumirTexto(string $texto, int $limite, string $continue = '...'): string
{
    $textolimpo = trim(strip_tags($texto));
    if (mb_strlen($textolimpo) <= $limite) {
        return $textolimpo;
    }
    $resumirTexto = mb_substr($textolimpo, 0, mb_strrpos(mb_substr($textolimpo, 0, $limite), ''));
    return $resumirTexto . $continue;
}
```

Fonte: UnSet PHP, (2023).

Você também tem a opção de inserir comentários suplementares para explicar a lógica da função ou apresentar exemplos de utilização, caso seja sua vontade. Esses comentários têm a possibilidade de ser incorporados antes das linhas que iniciam com \*.

Tal documentação auxiliará outros programadores a compreenderem a finalidade da função, os parâmetros esperados, o valor retornado e as alternativas disponíveis. Além disso, ferramentas de desenvolvimento e IDEs compatíveis com o PHPDoc podem oferecer funcionalidades extras, como auto preenchimento e verificação de tipos, com base nessas informações do-



cumentais.

Agora entrando na linguagem orientada a banco de dados, vamos falar sobre o O SQL Server, que é um sistema de gerenciamento de banco de dados relacional (RDBMS) desenvolvido pela Microsoft. Ele é amplamente utilizado para armazenar, recuperar e gerenciar dados em aplicativos de software, websites e sistemas empresariais. O SQL Server utiliza a linguagem SQL (Structured Query Language) como sua linguagem de consulta principal. Essa linguagem é usada para criar, consultar, modificar e gerenciar bancos de dados SQL Server (Microsoft, 2023).

A documentação no SQL Server pode ser feita de várias maneiras, sendo duas das mais comuns o uso de comentários delimitados por `/* */` (comentário de bloco), e o uso de comentários iniciados por `--` (hifens duplos).

No SQL Server, você pode inserir comentários delimitados por `/* */` para documentar seu código. Qualquer texto inserido entre `/` e `*/` será tratado como um comentário e não será interpretado como código SQL. Isso é útil para adicionar explicações e notas ao seu script SQL.

**Imagem 6** - Comentário no código SQL Server em `/* */`.

```
SQL

DECLARE @comment AS VARCHAR(20);
GO
/*
SELECT @comment = '/*';
*/ */
SELECT @@VERSION;
GO
```

Fonte: Microsoft Learn.

Outra forma comum de documentar código SQL no SQL Server é usando comentários iniciados por `--`. Tudo após `--` na mesma linha é considerado um comentário e não afeta a execução do código.

### Imagem 7 - Comentário no código SQL Server em --.

SQL

```
-- Choose the AdventureWorks2022 database.
USE AdventureWorks2022;
GO
-- Choose all columns and all rows from the Address table.
SELECT *
FROM Person.Address
ORDER BY PostalCode ASC; -- We do not have to specify ASC because
-- that is the default.
GO
```

Fonte: Microsoft Learn (2023).

A documentação adequada permite que até mesmo aqueles com conhecimento técnico limitado possam entender as funcionalidades do código e realizar as alterações necessárias com confiança. Além disso, a documentação contribui para uma colaboração mais efetiva entre os membros da equipe, pois proporciona um entendimento comum do código e evita mal-entendidos.

No geral, o resultado é um código mais legível, compreensível e fácil de manter. A documentação adequada reduz a curva de aprendizado para novos desenvolvedores que se juntam ao projeto e torna as atividades de manutenção e depuração mais eficientes. Portanto, investir tempo e esforço na documentação de código é uma prática valiosa que traz benefícios significativos ao longo do ciclo de vida do software.

## CONSIDERAÇÕES FINAIS

Foi enfatizado nesse artigo a importância da documentação de código foi destacada neste artigo, ressaltando sua natureza indispensável para aprimorar a colaboração e facilitar a manutenção em projetos de equipe, bem como garantir o sucesso do software desenvolvido. Recomendou-se a adoção de uma linguagem clara e concisa na documentação, por meio de comentários que detalhem minuciosamente a função, o objetivo e a lógica do código. Além disso, é fundamental atribuir nomes significativos e descritivos às variáveis e funções, contribuindo para a compreensão do código.

Foi mencionada também a viabilidade de utilizar ferramentas de documentação automatizada, como o Javadoc, que permite gerar a documentação a partir de comentários inseridos no código-fonte, agilizando e mantendo a documentação atualizada de maneira eficiente.

Adicionalmente, foram apresentadas diversas modalidades de documentação amplamente utilizadas pelos desenvolvedores, tais como a documentação de usuário, a criação de diagramas, o mapeamento de fluxo e a descrição de APIs. Essas diferentes abordagens podem ser adotadas de acordo com as necessidades específicas do projeto, ampliando as opções disponíveis para a documentação eficaz.

Em síntese, fica evidente que a documentação de código é uma prática essencial e de suma importância para o êxito de um projeto de software. Ao seguir as melhores práticas mencionadas neste artigo e explorar as diversas modalidades de documentação, os desenvolvedores podem garantir uma comunicação clara, a compreensão do código-fonte, facilitar a manutenção contínua e promover a colaboração efetiva entre os membros da equipe.

Investir tempo e esforço na documentação de código é um passo significativo em direção a um código mais legível e a um desenvolvimento de software mais eficiente e eficaz.

Fica claro que a documentação de código é uma habilidade valiosa para qualquer desenvolvedor. Quanto melhor a documentação, mais fácil é para você, sua equipe e outros desenvolvedores compreenderem, manterem e colaborarem no código. Portanto, não subestime o poder da documentação, pois ela desempenha um papel crucial em projetos de desenvolvimento de software bem-sucedidos.

## REFERÊNCIAS

BARRO, B. B. O que é DOM? Descubra o Coração do Desenvolvimento Web. Disponível em: <<https://www.hostinger.com.br/tutoriais/dom-o-que-e#:~:text=Trata%2Dse%20de%20uma%20interface>>. Acesso em: 5 nov. 2023.

DAITYARI, Shaumik. 13 Melhores Ferramentas de Revisão de Código para Desenvolvedores Edição 2023. Kinsta. Disponível em: <<https://kinsta.com/pt/blog/ferramentas-de-revisao-de-codigo/>> Acesso em: 11 maio 2023.

DEV.TO. O segredo de uma documentação simples em Python. Disponível em: <<https://dev.to/suspir0n/o-segredo-de-uma-documentacao-simples-em-python-4ik9>>. Acesso em: 5 nov. 2023.

FEOFILOFF, Paulo, Projeto de Algoritmos. IME.USP. Disponível em: <<https://www.ime.usp.br/~pf/algoritmos/aulas/docu.html>> Acesso em: 11 maio 2023.

LUIZ, Ivan. Documentação de código com Javadoc. DCA/FEEC/UNICAMP. Disponível em: <<https://www.dca.fee.unicamp.br/cursos/PooJava/javaenv/javadoc.html#:~:text=javadoc%20%C3%A9%20uma%20ferramenta%20para,com%20coment%C3%A1rios%20em%20formato%20adequado./>> Acesso em: 12 maio 2023.

RALHA, Cláudio. Aprendendo a documentar o seu código. Linhadecódigo.

Disponível em: <<http://www.linhadecodigo.com.br/artigo/1494/aprendendo-a-documentar-o-seu-codigo.aspx/>> Acesso em: 11 maio 2023.

UNSET. Disponível em: <<https://unset.com.br/post/curso-de-php-8/aula-023-documentacao-do-codigo-com-phpdoc>>. Acesso em: 5 nov. 2023.

RWESTMSFT. Barra estrela (comentário de bloco) (Transact-SQL) - SQL Server. Disponível em: <<https://learn.microsoft.com/pt-br/sql/t-sql/language-elements/slash-star-comment-transact-sql?view=sql-server-ver16>>. Acesso em: 5 nov. 2023.

RWESTMSFT. -- (Comentário) (Transact-SQL) - SQL Server. Disponível em: <<https://learn.microsoft.com/pt-br/sql/t-sql/language-elements/comment-transact-sql?view=sql-server-ver16>>. Acesso em: 5 nov. 2023.

“O conteúdo expresso neste capítulo é de inteira responsabilidade dos autores”

Os autores autorizam a publicação e distribuição gratuita deste e-book por meio das editoras e da Fatec de Jundiaí ou canais de distribuição indicados pelas referidas entidades.

# Ciência, Tecnologia e Cultura na Praça

## VOLUME 2

### Organizadores:

Carlos Eduardo Schuster  
Claudia Aparecida Longatti  
Claudio Luis Vieira Oliveira  
Galileo de Souza Schioser  
Peter Jandl Júnior



ISBN 978-65-5104-062-7

