



## Seu Título Aqui

Kauã Oliveira Seixas <sup>\*1</sup>, Rivaldo de Jesus Santos <sup>†1</sup>, Vinícius Scola Santana <sup>‡1</sup>, Eduardo Campos Aguiar <sup>§1</sup>, Cauã César Rodrigues Costa <sup>¶1</sup>, Orientador: Elton Figueiredo da Silva <sup>||1\*</sup>

<sup>1</sup> Bacharelado em Engenharia de Software

Escola de Tecnologias

Universidade Católica do Salvador (UCSAL)

Av. Prof. Pinto de Aguiar, 2589 Pituaçu, CEP: 41740-090

Salvador/BA, Brasil

<sup>1</sup> {kaua.seixas, rivaldo.santos, viniciusscola.santana, eduardo.aguiar, cauacesar.costa}@ucsal.edu.br

<sup>1\*</sup> {elton.figueiredo}@pro.ucsal.edu.br

Março 2025

---

\*kaua.seixas@ucsal.edu.br

†rivaldo.santos@ucsal.edu.br

‡viniciusscola.santana@ucsal.edu.br

§eduardo.aguiar@ucsal.edu.br

¶cauacesar.costa@ucsal.edu.br

||elton.figueiredo@pro.ucsal.br

## Resumo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Palavras-chaves:** Transformação digital, software, inovação.

# 1 Introdução

Muitas pessoas pensam que software é simplesmente outra palavra para programas de computador. No entanto, quando falamos de engenharia de software, não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente. A documentação nada mais é que um conjunto de informações que descreve as características, funcionamento, o uso e os requisitos de um software, dando apoio ao processo de construção dele. Desse modo, temos que, a documentação é de suma importância para o desenvolvimento de um software, servindo para diversos propósitos como:

- Facilitar o entendimento e a comunicação entre os envolvidos no projeto, como desenvolvedores, clientes, usuários, testadores, gerentes, etc;
- Auxiliar na definição do escopo, dos objetivos e das funcionalidades do software;
- Orientar o desenvolvimento, o teste e a implantação do software;
- Permitir a verificação e a validação da conformidade do software com os requisitos especificados;
- Fornece instruções e orientações para o uso adequado do software pelos usuários finais;
- Apoiar a manutenção e a evolução do software ao longo do tempo.

## 2 Fundamentos da Documentação de Software

Quando falamos sobre documentação de software, estamos nos referindo a qualquer material textual que times de engenharia, teste, produto e demais profissionais utilizam para realizar o seu trabalho. a documentação deve ser uma descrição precisa sobre um sistema de software. Quanto maior a precisão desses documentos, maior o status de autoridade que eles podem ter. Documentações devem abstrair a tecnicidade de um assunto, seja de implementações, configurações, e focar no que é essencial para quem a utilizar, trazendo informações completas e imprescindíveis para que se possa alcançar um objetivo e realizar uma ação. Existem diversos tipos de documentação, sendo cada tipo útil para situações diferentes:

### 2.1 Documentação Técnica

É aquela voltada para os profissionais que participam do projeto, como desenvolvedores, testadores, gerentes, etc. Ela inclui:

Documentação de requisitos: descreve as necessidades, as expectativas e as restrições dos clientes e dos usuários em relação ao software. Ela pode ser feita em diferentes níveis de detalhamento, como visão geral, especificação funcional, especificação técnica, casos de uso, histórias de usuário.

Documentação de projeto: descreve a arquitetura, o design e a estrutura do software. Ela pode incluir diagramas, modelos, padrões, componentes, interfaces, etc.

Documentação de código: descreve o funcionamento interno do código-fonte do software. Ela pode incluir comentários, anotações, cabeçalhos, etc.

Documentação de teste: descreve os métodos, as ferramentas, os cenários, os casos e os resultados dos testes realizados no software. Ela pode incluir planos de teste, roteiros de teste, relatórios de teste, evidências de teste, etc.

Documentação de implantação: descreve os procedimentos e as configurações necessárias para instalar e executar o software em um ambiente específico. Ela pode incluir guias de instalação

## 2.2 Documentação de usuário

É um conjunto de materiais criados para ajudar os usuários a entenderem, instalarem e configurarem um software. Ela é voltada a pessoas que não necessariamente têm conhecimento técnico. ela inclui: -Guia de início rápido: Passos básicos para a instalação e configuração de um software. -Tutoriais Interativos: Passo a passo para tarefas específicas.

## 2.3 Documentação de arquitetura/design

A documentação de arquitetura ou design fornece um modelo da estrutura do software, detalhando os componentes de alto nível, suas interações e os padrões de design subjacentes. É crucial para a integração de novos desenvolvedores e para manter a consistência em grandes projetos.

## 2.4 Documentação da API

A documentação da API é uma forma especializada de documentação técnica que fornece detalhes sobre como interagir com a API do software. Inclui descrições de métodos, parâmetros de entrada, formatos de saída e exemplo.

# 3 Boas práticas de documentação de software

Por fim é necessário ter uma boa prática de documentação para a construção de um software como:

- Definir o público-alvo e o propósito da documentação. A documentação deve ser adaptada às necessidades e ao nível de conhecimento dos leitores, seja eles desenvolvedores, usuários finais, gerentes ou clientes. O propósito da documentação também deve ser claro, seja ele instruir, informar, persuadir ou avaliar.
- Escolher o formato e a ferramenta adequados para a documentação. A documentação pode ser apresentada em diferentes formatos, como texto, diagramas, vídeos, tutoriais ou exemplos de código. A escolha do formato depende do tipo de informação que se quer transmitir, da complexidade do assunto e da preferência do público.
- Seguir um padrão e uma estrutura para a documentação. A documentação deve seguir um padrão de estilo, linguagem e formatação que seja consistente, claro e objetivo. Isso facilita a leitura, a compreensão e a busca pela informação.
- Manter a documentação atualizada e revisada. A documentação deve refletir o estado atual do projeto, as mudanças realizadas e as decisões tomadas
- Obter feedback e melhorar a documentação. A documentação deve ser testada e avaliada pelos leitores para verificar se ela atende às suas expectativas, necessidades e dúvidas.

São, também, boas práticas para criar um documentação: Clareza e consistência, Abordagem centrada no público, Controle de versão e gerenciamento de mudanças, Colaboração entre equipes. Assim a documentação de um software trás diversos benefícios como a melhor qualidade e confiabilidade de um software, a redução dos erros e falhas no funcionamento do software, o aumento de produtividade e a eficiência do desenvolvimento e entrega do software, além de aumentar a satisfação e a fidelização dos clientes e dos usuários. Por fim, cabe ao time de desenvolvimento decidir qual tipo de documentação faz mais sentido ser adotado. Essa decisão deve ser tomada de acordo com o contexto do projeto, as habilidades do time e o perfil de quem vai consumir esse conteúdo

## 4 Questões legais

-LGPD(Lei Geral de Proteção de Dados - Brasil):Esta Lei dispõe sobre o tratamento de dados pessoais, inclusive nos meios digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado, com o objetivo de proteger os direitos fundamentais de liberdade e de privacidade e o livre desenvolvimento da personalidade da pessoa natura. Exigindo que documentações de software incluam políticas de privacidade e tratamento de dados. 2-GDPR(Regulamento Geral de Proteção de Dados):é crucial para o desenvolvimento de software, pois garante a proteção de dados pessoais e defende os direitos dos

indivíduos, impacta na documentação de sistemas que coletam dados de cidadão europeus. 3-Apache License 2.0: é uma licença de software de código aberto permissiva que permite aos usuários usar, modificar e distribuir software livremente, inclusive para fins comerciais. Ele exige menção clara na documentação sobre termos de uso modificações e redistribuição. 4-Creative Commons recomenda: usa licenças de software livre e de código aberto para software, e para licenciar documentação técnica. 5-ISO/IEC 26514:Ele descreve e define requisitos para documentação e como estabelecer quais informações os usuários precisam, como determinar o caminho que essas informações devem ser apresentadas e como preparar as informações e disponibilizá-las

## 5 Desafios e soluções

1-MANTER A DOCUMENTAÇÃO ATUALIZADA:Um dos maiores desafios é garantir que a documentação reflita o estado atual do software. Ferramentas automatizadas e auditorias regulares da documentação podem ajudar a manter as coisas atuais. 2- INCENTIVAR A PARTICIPAÇÃO DOS DEVS:Os desenvolvedores frequentemente veem a documentação como uma tarefa. Incentivar a participação através de ferramentas colaborativas e integrar a documentação no processo de desenvolvimento pode ajudar a aliviar esse problema. 3-GERENCIAR A "DÍVIDA"DA DOCUMENTAÇÃO:Assim como com o código, a documentação pode acumular "dívida"ao longo do tempo. Revisar e refatorar regularmente a documentação pode evitar que ela fique desatualizada ou redundante.

## 6 Conclusão

A documentação de software é uma parte vital do processo de desenvolvimento de software. Ela garante que todas as partes interessadas tenham as informações necessárias para entender, usar e manter o software de forma eficaz. Boulic e Renault (1991) teste.

## Referências

BOULIC, R.; RENAULT, O. 3d hierarchies for animation. *New Trends in Animation and Visualization*, John Wiley & Sons ltd., 1991.