



Documentação de Software: Proteção Legal e Boas Práticas

Kauã Oliveira Seixas ^{*1}, Rivaldo de Jesus Santos ^{†1}, Vinícius Scola Santana ^{‡1}, Eduardo Campos Aguiar ^{§1}, Cauã César Rodrigues Costa ^{¶ 1}, Orientador: Elton Figueiredo da Silva ^{||1*}

¹ 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

¹ *{kaua.seixas, rivaldo.santos, viniciusscola.santana, eduardo.aguiar, cauacesar.costa}@ucsal.edu.br*

^{1*} *{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

Este trabalho discute a importância da documentação de software e sua relação com os aspectos legais envolvidas, destacando as principais licenças e regulamentos. Além de ser essencial para o desenvolvimento, a documentação ajuda a reduzir esforços desnecessários, como a dependência excessiva de pessoas e o reaprendizado de código, além de prevenir problemas futuros. No cenário atual da tecnologia, sua relevância é ainda maior, pois o desenvolvimento de software está cada vez mais dinâmico, distribuído e colaborativo, exigindo maior organização e facilitando a integração de novos profissionais e reduzindo riscos legais, garantindo conformidade com as regulamentações vigentes. Além disso, a documentação contribui para a segurança do usuário ao garantir transparência no uso de dados e reforçar medidas de proteção.

Palavras-chaves: Documentação, software, licenças de software, questões legais.

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 descreve cada parte do código-fonte, uma função, uma classe, um trecho ou módulo. Podemos dizer que a documentação consiste em um conjunto de manuais gerais e técnicos, podendo ser organizado em forma de textos e comentários, utilizando ferramentas do tipo dicionários, diagramas e fluxogramas, gráficos, desenhos, dentre outros (COELHO, 2009). 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.

A documentação pode ser dividida em dois grandes grupos: a parte técnica é considerada mais simples, pois descreve o trabalho do desenvolvedor, enquanto que a parte para o usuário é a mais exigente e requer habilidades especiais para a redação de manuais, inserção de screenshots, desenhos e outros elementos gráficos (COELHO, 2009).

Dentro desses dois grupos existem diversos tipos de documentação, sendo cada tipo útil para situações diferentes, elas são:

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 arquitetura/design: 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

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.

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 Legal ou de Conformidade

É aquela que descreve os aspectos jurídicos ou regulatórios como política de privacidade, termos de uso, licença de software(ex.: GPL, MIT, Apache).

2.4 Documentação de Processos

Está relacionada ao gerenciamento do desenvolvimento do software como: Plano de projeto(Cronogramas, recursos e fases do projeto), relatórios de status, atas de reunião, etc.

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 Responsabilidade Civil e Contratos

Em um contexto legal, a documentação de software também pode ter implicações relacionadas à responsabilidade civil e aos contratos. A documentação pode estabelecer claramente os termos e condições de uso de um software, indicando responsabilidades e isenções

de responsabilidade. Isso é especialmente importante quando um software é utilizado em ambientes comerciais ou sensíveis. Exemplo Prático: Se um software é desenvolvido para uma empresa, o contrato de desenvolvimento pode exigir que a documentação seja fornecida de maneira clara e abrangente. Isso ajuda a prevenir disputas legais no futuro, como por exemplo, um cliente alegando que o software não está funcionando conforme o prometido.

5 Questões legais

A documentação de software e as questões legais estão profundamente interligadas. A documentação não é apenas um recurso técnico para os desenvolvedores, mas também desempenha um papel crucial nas implicações legais que envolvem a criação, distribuição e uso de software. A documentação bem elaborada pode servir como um ponto de referência legal em diversos contextos, ajudando a proteger direitos de propriedade intelectual, clarificar responsabilidades e fornecer garantias contratuais.

5.1 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 natural. Exigindo que documentações de software incluam políticas de privacidade e tratamento de dados (Presidência da República do Brasil, 2018).

6 Questões Legais Internacionais

6.1 GDPR (General Data Protection Regulation)

É uma legislação da União Europeia que garante a proteção de dados pessoais e defende os direitos dos indivíduos, se aplica a dados de pessoas localizadas na União Europeia (residentes ou não), impondo regras a organizações que tratam de dados pessoais (European Union, 2016). Ex.: Uma empresa do Brasil que atende e coleta dados de clientes da Alemanha deve seguir as regras do GDPR.

6.2 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 (International Organiza-

tion for Standardization (ISO) and International Electrotechnical Commission (IEC) and Institute of Electrical and Electronics Engineers (IEEE), 2022).

7 Licenças de software

A licença de uso é o instrumento com o qual o titular dos direitos do software ou o distribuidor licenciado delimitam o uso, a aplicação e a sua utilidade para o usuário final (NETTO,). Existem diversos tipos de licenças de software, cada uma com suas próprias restrições e obrigações. Alguns exemplos de licenças de software livre e de código aberto incluem:

7.1 GNU - General Public License (GPL)

Richard M. Stallman, criador da Free Software Foundation em 1984, lançou o projeto GNU que se trata de uma licença de software livre que dá a todos os usuários a liberdade de redistribuir e modificar o software GNU (Free Software Foundation, 2007).

Em vez de colocar o software em domínio público, o Projeto GNU o torna [...] "copy-left" que diz que qualquer um que distribui o software, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o programa. O copyleft garante que todos os usuários possuem liberdade (NETTO,).

7.2 MIT License

É uma licença de software livre que permite aos usuários usar, modificar e distribuir o software livremente, inclusive para fins comerciais além de oferecer proteção contra disputas de patentes. Ela exige apenas a inclusão de um aviso de direitos autorais (Open Source Initiative, 1988).

7.3 Apache License 2.0

É uma licença de software de código aberto 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 (Apache Software Foundation, 2004).

8 Desafios Legais na Documentação de Software

1 - Conformidade com Leis Internacionais: Como mencionado, a conformidade com regulamentações como o GDPR é um dos maiores desafios atuais. A documentação de

software precisa ser ajustada para garantir que todas as políticas de privacidade e de proteção de dados estejam adequadas à legislação vigente nos diversos países onde o software é utilizado.

2 - Licenciamento de Software: A documentação de software precisa ser rigorosa na especificação de licenças de uso. Com a proliferação de licenças open-source, como a Apache License ou GNU GPL, é importante que as implicações legais de cada tipo de licença sejam claras na documentação, para garantir que os desenvolvedores e usuários finais compreendam suas permissões e limitações.

3 - Propriedade de Código e Trabalhos Colaborativos: Em projetos colaborativos, especialmente aqueles envolvendo várias empresas ou desenvolvedores independentes, a documentação precisa abordar questões de propriedade intelectual, especificando quem detém os direitos sobre o código-fonte e os recursos criados. Contratos claros e documentação detalhada ajudam a evitar disputas sobre a propriedade do trabalho.

9 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.

Além disso, ela não é apenas uma ferramenta técnica, mas também um instrumento essencial para lidar com questões legais. Ela protege os direitos de propriedade intelectual, clarifica responsabilidades e estabelece a conformidade com regulamentações internacionais como o GDPR. A adoção de boas práticas na criação de documentação legalmente robusta é crucial para minimizar riscos legais e garantir que o software seja utilizado de forma segura e responsável.

Referências

- Apache Software Foundation. *Apache License, version 2.0*. 2004. Accessed: 2025-03-19. Disponível em: <<https://www.apache.org/licenses/LICENSE-2.0>>.
- COELHO, H. S. Documentação de software: uma necessidade. *Texto Livre: linguagem e tecnologia*, Universidade Federal de Minas Gerais, v. 2, n. 1, p. 17–21, 2009.
- European Union. *General Data Protection Regulation (GDPR)*. 2016. Accessed: 2025-03-21. Disponível em: <<https://gdpr-info.eu/>>.
- Free Software Foundation. *GNU General Public License, version 3 (GPL-3.0)*. 2007. Accessed: 2025-03-19. Disponível em: <<https://www.gnu.org/licenses/gpl-3.0.html>>.
- International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) and Institute of Electrical and Electronics Engineers (IEEE). *ISO/IEC/IEEE 26514:2022 - Systems and software engineering — Design and development of information for users*. 2022. Accessed: 2025-03-20. Disponível em: <<https://cdn.standards.iteh.ai/samples/77451/e161501288fd44f88c6fdd0f6e46b017/ISO-IEC-IEEE-26514-2022.pdf>>.
- NETTO, V. C. D. S. Questões legais sobre desenvolvimento, comercialização, direito autoral e uso de software.
- Open Source Initiative. *The MIT License*. 1988. Accessed: 2025-03-19. Disponível em: <<https://opensource.org/license/mit>>.
- Presidência da República do Brasil. *Lei Geral de Proteção de Dados Pessoais (LGPD) - Lei nº 13.709, de 14 de agosto de 2018*. 2018. Accessed: 2025-03-20. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>.