

## DOCUMENTAÇÃO DE SOFTWARE: UMA NECESSIDADE

COELHO, Hilda Simone<sup>1</sup>

<sup>1</sup>Universidade Federal de Minas Gerais, Faculdade de Letras, MG, Brasil

\* e-mail: [hildacoelho2003@yahoo.com.br](mailto:hildacoelho2003@yahoo.com.br)

---

### RESUMO

Este trabalho tem como objetivo mostrar a importância da documentação em software e apresentar algumas ferramentas mais comuns bem como trazer os resultados de uma pesquisa com profissionais desenvolvedores de programas de software sobre as ferramentas mais utilizadas para a documentação.

**Palavras-chave:** documentação de software; ferramentas de documentação.

### ABSTRACT

This work aims to discuss about the importance of software documentation and to show some examples of tools for this purpose. It also brings the results of a research done with professionals of the area about the most used tools for software documentation.

**Keywords:** software documentation; tools for software documentation.

---

### Introdução

Fazer uma pesquisa sobre ferramentas de documentação em software foi um grande desafio. Sendo da área de Linguística Aplicada, me vi na obrigação e necessidade de fazer uma busca detalhada nas bibliotecas com a expectativa de encontrar referencial teórico para dar subsídio a este trabalho. Qual não foi minha surpresa ao encontrar pouco material, incluindo pesquisas, com foco em documentação e ferramentas para a documentação de software.

Continuei minha busca e parei em uma empresa de programação de software, especificamente o desenvolvimento de jogos pedagógicos<sup>1</sup>. Outra surpresa: os técnicos deram o depoimento de que, apesar de saberem da importância da documentação, reconhecem a dificuldade para realizá-la de forma adequada e eficiente para o usuário; conseguem construir uma documentação que registre os passos da construção do software que venha facilitar sua manutenção, mas afirmaram que é de extrema dificuldade a redação de uma documentação que facilite o uso e a compreensão do software para o usuário, principalmente se pensar a possibilidade de ser este um leigo nos caminhos da informática.

Segui minha busca pela internet e encontrei disponibilizados alguns trabalhos de colegas de outras instituições, a maioria material em *power point*, outros pequenos artigos disponibilizados em sites para o desenvolvimento de documentação de software, e outros poucos artigos em pdf que discorriam especificamente sobre o assunto.

Desta forma, acredito na relevância deste trabalho que, além de compartilhar o sentimento dos pesquisadores e profissionais da área sobre a escassez de material sobre documentação em software, tem como objetivo discutir a importância da documentação de software e trazer a exemplificação de algumas ferramentas básicas de documentação, como, também, mostrar os resultados de uma pesquisa sobre as ferramentas mais utilizadas para documentação.

Este trabalho está organizado da seguinte forma: primeiro apresento a definição de documentação, a seguir trago exemplos de algumas ferramentas de documentação e, por último, mostro os resultados de uma pesquisa sobre as ferramentas mais utilizadas (SOUZA et al 2004).

## 1. O que é documentação?

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.

As empresas de software bem sucedidas sabem que precisam desenvolver seus produtos cumprindo prazos, garantindo a qualidade e a eficiência de seus recursos. O produto final que garante a satisfação do cliente é o mais importante, mas para alcançar o sucesso de um software devem-se seguir todos os passos de seu desenvolvimento e isto inclui o conhecimento de seus usuários, expondo a eles “os requisitos reais do sistema” (BOOCH et al 2006:3). A evolução de um software e seu sistema operacional devem ser claramente informados, estabelecendo procedimentos para registrar e resolver problemas, facilitando sua manutenção posterior e, assim, garantindo sua durabilidade. Para que isto aconteça é preciso desprender um tempo precioso em documentação, designando as pessoas habilitadas especificamente para esta tarefa e escolhendo as ferramentas que possibilitem o enfoque correto de cada passo a ser documentado.

A documentação deve ser feita ao longo do desenvolvimento do software. Problemas de custos, imprecisão dos serviços fornecidos e dificuldades de manipulação podem ser causadas pela ausência de documentação ou quando esta é construída após o desenvolvimento do software. A documentação exige seu planejamento, isto é, a identificação dos documentos, o estabelecimento de sua organização, a definição da linguagem a ser utilizada e um estudo das diferentes tecnologias para tal.

Ao explicar sobre o uso da linguagem de modelagem UML<sup>ii</sup>, BOOCH, RUMBAUGH e JACOBSON (2006) afirmam que a documentação, a modelagem de um software, é “uma parte central de todas as atividades que levam à implantação de um bom software.” Sua construção deve:

- Comunicar a estrutura e o comportamento do sistema;
- Visualizar e controlar a arquitetura do sistema;

- Expor oportunidades de simplificação e reaproveitamento;
- Gerenciar riscos.

Os mesmo autores reforçam a importância do planejamento do software e de sua documentação comparando sua construção a construção de casas e prédios. Assim como o projeto arquitetônico e o de engenharia de um imóvel deve guiar os passos da construção, o segredo de um software bem sucedido é “uma questão de arquitetura, processo e ferramentas” (BOOCH et al 2006: 5).

Tendo exposto o que é documentação e tratado de justificar sua importância, apresentaremos a seguir os tipos de documentação e algumas ferramentas mais comuns.

## 2. Tipos de documentação

Segundo Michelazzo (2006), a documentação pode ser dividida em dois grandes grupos, conforme a tabela 1.

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.

DOCUMENTAÇÃO TÉCNICA	DOCUMENTAÇÃO DE USO
<ul style="list-style-type: none"> <li>■ Voltada ao desenvolvedor</li> <li>■ Compreende dicionários e modelos de dados, fluxogramas de processos e regras de negócios, dicionários de funções e comentários de códigos</li> </ul>	<ul style="list-style-type: none"> <li>■ Voltada para o usuário final e o administrador do sistema</li> <li>■ Formada por apostilas ou manuais que apresentam como o manual deve ser usado, o que esperar dele e como receber as informações que se deseja</li> </ul>

### 3. Ferramentas de documentação

Seguindo a exemplificação de Michelazzo (2006), apresentaremos alguns tipos de ferramentas de documentação.

**3.1. Documentação de código:** feita basicamente de duas formas – comentários dentro do próprio código e geração de documentação online. O desenvolvedor deve ter a consciência de que não será o único a “colocar a mão no sistema” e por isso deve fazer o comentário dos códigos de maneira bastante clara. Um exemplo de um código bem comentado pode ser visto a seguir:

```
$database->setQuery( $query );
$rows = $database->loadObjectList(); // establish the
hierarchy of the menu
$children = array();
// first pass - collect children
if ($rows) foreach ($rows as $v ) {
    $pt = $v->parent;
    $list = @$children[$pt] ? $children[$pt] : array();
    array_push( $list, $v );
    $children[$pt] = $list;
}
// second pass - get an indent list of the items
$list = mosTreeRecurse( 0, "", array(), $children,
max( 0, $levellimit-1 ) );
// eventually only pick out the searched items.
if ($search) {
```

(fonte: <http://www.michelazzo.com.br/node/123>)

Michelazzo (2006) ressalta que não é o número de linhas de comentários que garante a eficácia da documentação, mas a ausência de comentários poderá acarretar gastos desnecessários de horas de trabalho à procura de erros e possivelmente provocará estresse em todos que irão lidar com o software.

**3.2. Modelos de dados:** refletem de forma gráfica (e lógica) a base de dados de um sistema, seus relacionamentos, entidades, chaves e tudo aquilo que se refere aos dados em si. É, portanto, considerado peça fundamental para o desenvolvimento de um sistema e por isso são pensados e criados *antes* do início do desenvolvimento. Podem ser criados por meio de engenharia reversa ou ainda baseando-se nas necessidades do aplicativo que está sendo envolvido. As ferramentas mais conhecidas para a modelagem dentro do mundo livre são DBDesigner, MySQL Workbench e PGDesigner, as quais possuem funcionalidades

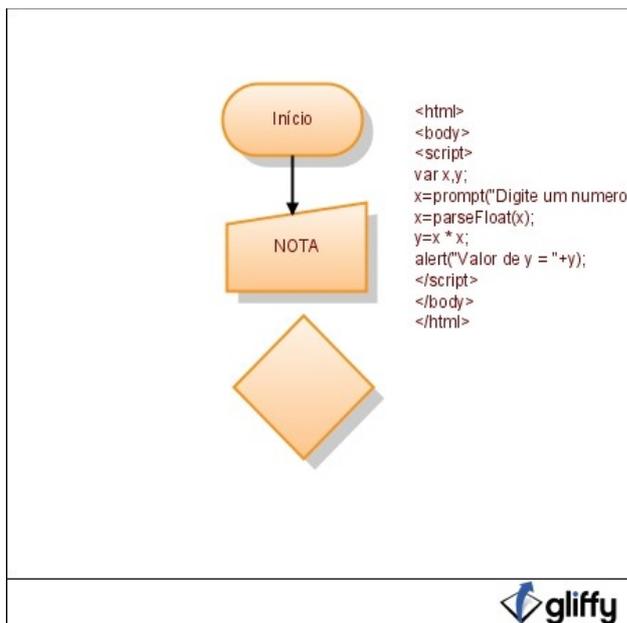
diferentes e dispõem de versões tanto para Linux quanto para Windows.

**3.3. Dicionários de dados:** arquivo ou documento que define a organização básica dos dados do banco. Nele são informadas as tabelas, os campos, suas definições, tipos e descrições. Um exemplo simples de um dicionário de dados pode ser visto a seguir:

Campo	Tipo	Tam.	Propriedades	Desc.
id_cadastro	Int	5	Auto-increment	ID do registro do cadastro
na_nome	Varchar	50	Not Null	Nome do dono da empresa
na_empresa	Varchar	50	Null	Nome da empresa
ad_email	Varchar	75	Not Null	Endereço de e-mail principal
nu_fone	Varchar	14	Not Null	Telefone de contato

(fonte: <http://www.michelazzo.com.br/node/123>)

**3.4. Fluxogramas:** apresentam graficamente a seqüência lógica das informações de um processo ou sistema, utilizando vários elementos de geometrias diferentes que indicam uma das partes do processo. Visualmente conseguem passar a lógica de todo um sistema desde os níveis mais altos de processamento até pequenas partes, permitindo, assim, uma visão geral do que realmente precisa ser feito dentro de um sistema. Um exemplo de fluxograma pode ser visto a seguir:



(fonte: <http://www.gliffy.com/pubdoc/1366615/L.jpg>)

Existem algumas ferramentas para criar fluxogramas como, por exemplo, DIA e o OpenModeling, disponíveis em várias plataformas e livros.

Existem várias outras ferramentas que facilitam a vida do programador na hora de gerar a documentação (vide lista disponível em Souza et al, 2005). VAILATI (2006), por exemplo, aborda em seu artigo a utilização do PHPDocumentor, uma excelente ferramenta para a linguagem PHP que gera automaticamente a documentação, baseada em tags próprias contidas nos códigos fonte. O desenvolvedor do software deve procurar pelas ferramentas que mais o ajudam na descrição de determinada parte do documento.

Uma vez conhecido algumas ferramentas de documentação, passaremos a tratar do uso e da importância da documentação, apresentando mais adiante, os resultados de duas pesquisas realizadas para o conhecimento de ferramentas mais relevantes no momento da manutenção em software.

#### 4. Usos da documentação

Segundo AMBLER (2001) há duas razões básicas para documentar: auxiliar a comunicação durante o desenvolvimento do software e auxiliar o entendimento nas atividades de manutenção e atualização quando se fizerem necessárias.

Os especialistas em manutenção afirmam que mais de 50% do trabalho de evolução de um software é dedicado ao entendimento do programa e a tarefa mais difícil da manutenção é o entendimento da estrutura do sistema. Por isso, o desenvolvedor deve encontrar um ponto de equilíbrio na quantidade de documentação gerada e encontrar as ferramentas que facilitarão o entendimento da documentação, tanto para o administrador quanto para o futuro usuário do software.

Alguns problemas são apontados pela falta de documentação adequada. Dentre eles podemos citar as dificuldades para a atualização e manutenção, dificuldades de acesso do usuário e alto custo para o software.

Outra questão relevante é o cumprimento do prazo de entrega do software, fator essencial para a sobrevivência de uma organização, e que está intimamente ligada à quantidade e qualidade da documentação. Alguns métodos ágeis vêm ganhando popularidade, como, por exemplo, o Extreme Programming (XP).

#### 5. Ferramentas mais utilizadas

No intuito de trazer neste trabalho as ferramentas de documentação mais recorrentes, apresentaremos os resultados de duas pesquisas discutidas por SOUZA, ANQUETIL & OLIVEIRA (2005). Ambas foram realizadas com profissionais da área de desenvolvimento de software, sendo 65% analistas de sistemas; 25% gerentes; 7% programadores e 3% consultores.

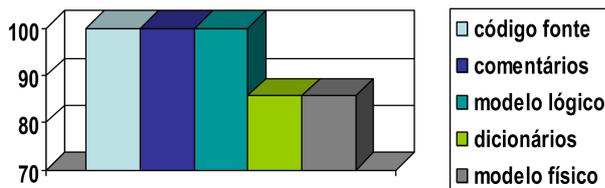
A primeira pesquisa buscou identificar as ferramentas de software que auxiliam no entendimento do sistema na fase de manutenção. A tabela a seguir mostra estas ferramentas e traz a definição para cada tipo.

ferramentas que auxiliam no entendimento dos sistemas	DEFINIÇÃO DA FERRAMENTA
Requisitos	Referentes às ferramentas lista de requisitos e sua descrição na análise estruturada, e diagrama de caso de uso e sua especificação na análise orientada a objetos

Modelos de dados	Referentes às ferramentas modelo lógico e físico de dados na análise estruturada, as mesmas ferramentas associadas ao diagrama de classe que é a base para a definição desses modelos na análise orientada a objetos
Código	Referente ao código fonte e comentários no mesmo nas duas abordagens

Tabela 2.

A segunda pesquisa teve como objetivo verificar a real utilização de cada ferramenta no dia a dia da manutenção. Os dados foram coletados através de dois questionários respondidos pelos profissionais da área. As cinco ferramentas mais utilizadas podem ser vistas no gráfico abaixo:



Souza et al (2005) apontam que outras oito ferramentas foram assinaladas pelos profissionais entrevistados, mostrando uma recorrência entre 85,7 % (manual do usuário, plano de implantação unitário, plano de testes) a 42,9% (plano de testes de homologação, diagramas de fluxo de dados, diagrama hierárquico de funções).

Dentre as 24 ferramentas apresentadas no questionário, 14 foram utilizadas pelos mantenedores. Isto implica a importância da documentação e também a necessidade de se ter claro o propósito para a qual ela é criada e mantida. Os autores destacam a importância do código fonte juntamente com seus comentários, daí o cuidado que é preciso ter com a adoção de padrões de codificação e de organização interna do código.

Souza et al (2005) argumentam que os resultados destas pesquisas podem ser úteis para a análise em torno de outras ferramentas de documentação de software visando buscar o equilíbrio na produção de ferramentas adequadas.

## 6. Conclusão

Os resultados da pesquisa feita para a realização deste trabalho indicam a necessidade de produzir outros materiais sobre documentação em software bem como divulgar a importância do planejamento da documentação, as escolhas de seus redatores e das ferramentas mais adequadas a cada passo a ser documentado, servindo como apoio para os seus desenvolvedores, ou, para apoio de seus usuários. Embora compreendamos a limitação deste trabalho, esperamos ter contribuído para o entendimento do uso e da importância da documentação de software e de algumas ferramentas utilizadas para este fim.

## REFERÊNCIAS BIBLIOGRÁFICAS

AMBLER, S. W. Agile Documentation. 2001-2004, The Official Agile Modeling (AM) Site, 2001, Disponível em: <http://www.agilemodeling.com/essays/AgileDocumentation.htm> Acesso em: nov. 2008.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. Rio de Janeiro: Elsevier, 2005 – 2ª reimpressão, 2006.

MICHELAZZO, P. A Documentação de software, 2006. Disponível em: <http://www.michelazzo.com.br/node/123>. Acesso em: nov, 2008.

SOUZA, S. C. B. de; NEVES, W. C. G. das; ANQUETIL, N.; OLIVEIRA, K. M. de. Documentação Essencial para a Manutenção de Software, 2005. Disponível em: <http://www.web.br/prg/professores/anqueteil/Publicacoes/pbqp04.pdf>. Acesso em: Nov, 2008.

VAILATI, A. T. Documentação de Software , 2006, disponível em: [HTTP://imasters.uol.com.br/artigo/4923/des\\_de\\_software/documentacao\\_de\\_software](HTTP://imasters.uol.com.br/artigo/4923/des_de_software/documentacao_de_software). Acesso em nov. 2008.

- i Meus agradecimentos aos profissionais Páblisson, Héber, Igor e Henrique, da *Jungle Digital Games*, que muito me auxiliaram neste trabalho.
- ii Unified Modeling Language, desenvolvida por Grady Booch, James Rumbaugh e Ivar Jacobson por volta do ano de 1996.