

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
CURSO DE GRADUAÇÃO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

EVANDSON DA COSTA MARANHÃO

**IMPORTÂNCIA DA DOCUMENTAÇÃO NO DESENVOLVIMENTO DE SISTEMAS:
UM ESTUDO DE CASO**

NATAL/RN
2014

EVANDSON DA COSTA MARANHÃO

**IMPORTÂNCIA DA DOCUMENTAÇÃO NO DESENVOLVIMENTO DE SISTEMAS:
UM ESTUDO DE CASO**

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário do Rio
Grande do Norte, em cumprimento às
exigências legais como requisito final para
obtenção do título de Bacharel em Sistemas
de Informação.

Orientador: Prof^o (MSC.) Joseane Alves
Pinheiro.

NATAL/RN
2014

EVANDSON DA COSTA MARANHÃO

IMPORTÂNCIA DA DOCUMENTAÇÃO NO DESENVOLVIMENTO DE SISTEMAS:
UM ESTUDO DE CASO

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário do Rio
Grande do Norte, em cumprimento às
exigências finais para
a obtenção do grau de Bacharel em Sistemas
de Informação

**Catálogo na Publicação – Biblioteca do UNI-RN
Setor de Processos Técnicos**

Maranhão, Evandson da Costa.

Importância da documentação no desenvolvimento de sistemas : um
estudo de caso. / Evandson da Costa Maranhão. – Natal, 2014.
47 f.

Orientador: Pro^{fa}. Joseane Alves Pinheiro.

Monografia (Graduação em Sistemas de Informação) – Centro
Universitário do Rio Grande do Norte.

1. Desenvolvimento de Sistemas – Monografia. 2. Documentação de
Software – Monografia. 3. Ferramentas de Documentação - Monografia. I.
Pinheiro, Joseane Alves. II. Título.

RN/UNI-RN/BC

CDU 004

UNIVERSIDADE
NATURAL

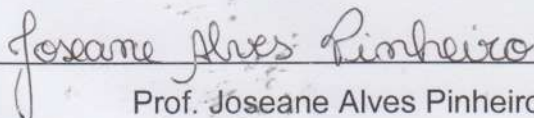
EVANDSON DA COSTA MARANHÃO ,

**IMPORTÂNCIA DA DOCUMENTAÇÃO NO DESENVOLVIMENTO DE
SISTEMAS: UM ESTUDO DE CASO**

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário do
Rio Grande do Norte, em cumprimento
às exigências legais como requisito
final para obtenção do título de
Bacharel em Sistemas de Informação.

Aprovado (a) em 08 / 12 / 14 .

BANCA EXAMINADORA



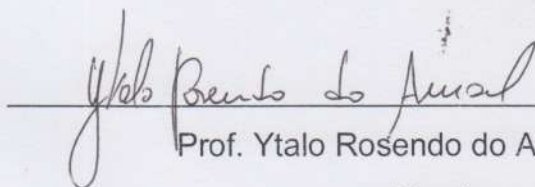
Prof. Joseane Alves Pinheiro (MSC.)

Orientador



Prof. Luiz Gonzaga Damasceno (MSC.)

Membro



Prof. Ytalo Rosendo do Amaral (MSC.)

Membro

Dedico este trabalho aos meus pais, por terem sempre acreditado na minha capacidade de realizar este sonho em minha vida, não medindo esforço para ajudar no que foi necessário para que essa caminhada chegasse ao fim da melhor forma possível.

AGRADECIMENTOS

Agradeço em primeiro lugar a DEUS por ter iluminado o meu caminho durante esta caminhada, me dando força e coragem para realizar essa conquista.

A minha família, principalmente aos meus pais, que foram os maiores incentivadores para que essa conquista fosse possível, ajudando durante toda caminhada a superar os obstáculos encontrados.

Aos educadores da instituição, que foram o canal responsável por todo conhecimento adquirido durante este período, ajudando superar todas as dificuldades no período de aprendizagem.

E, agradeço a todas as pessoas e colegas de classe pela amizade e companheirismo que recebi durante o período de curso, agora futuros amigos de profissão na área de tecnologia da informação.

"Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes."

(Martin Luther King)

RESUMO

O desenvolvimento da documentação do software de gerenciamento de Clínica-Escola do Centro Universitário do Rio Grande do Norte (UNI-SGCE), módulo recepção, busca facilitar a manutenção do sistema e a criação dos módulos futuros. Neste contexto, foi realizada uma pesquisa em busca do que era necessário para a criação de uma documentação de sistema, de forma a reunir os materiais de prioridade para a documentação, para facilitar o entendimento do funcionamento do sistema por partes dos programadores que participaram do projeto e os que participarão da criação de módulos futuros. Enfim, a documentação do UNI-SGCE módulo recepção, busca ser uma ferramenta facilitadora para o sistema e os desenvolvedores, já que no sistema ainda serão adicionados novos módulos referentes às clínicas específicas que constituem o grupo de Clínica-Escola do UNI-RN, por o sistema ser um projeto desenvolvido pelo grupo de extensão da instituição, projeto no qual existe a constante mudança de equipe de desenvolvimento, pois é formada por professores e alunos da instituição, ocorrendo sempre à saída de alunos após a conclusão do curso superior que realizam na instituição. Levando em consideração esses aspectos, a documentação do sistema é a única ferramenta que pode ser considerada eficaz para que um usuário que não conhece o sistema passe a entendê-lo, e que sua gerência de mudança possa ocorrer de forma simples. A respeito do uso da documentação e sua importância no processo de construção e manutenção do software, foi realizada uma pesquisa com profissionais que trabalham com desenvolvimento de sistemas, em busca de mostrar as ferramentas importantes para esses profissionais no entendimento de um software, o tipo de manutenção que mais realizam e se desenvolvem documentos de software na projeção de sistemas.

Palavras-chave: Documentação. Desenvolvimento. Sistema. Importância. Clínica.

ABSTRACT

The development of the documentation of the School Clinic management software University Center of Rio Grande do Norte (UNI-SGCE), reception module aims to facilitate system maintenance and the creation of future modules. In this context, a study in search of what was needed for the creation of a system of documentation was performed in order to meet the priority materials for documentation, to facilitate understanding of system operation in parts of the developers who participated in the project and those who participate in the creation of future modules. Finally, UNI-SGCE documentation module reception, seeks to be a facilitating tool for the system and developers, since the system will still be added new modules for the specific clinical constituting the clinic belonging to group UNI-RN, for the system is a project developed by the institution extension group project in which there is the ever-changing development team as it is formed by teachers and students of the institution, always occurring off students after completion of higher education holding in institution. Considering these aspects, the system documentation is the only tool that can be considered effective for a user who does not know the system password to understand it, and that his change of management can simply occur. Regarding the use of documentation and its importance in the software construction and maintenance process, a survey was conducted with professionals working with systems development, seeking to show the important tools for professionals in this understanding of a software, the type of maintenance that most realize and develop software documentation for system projection.

Keywords: Documentation. Development. System. Importance. Clinic.

LISTA DE FIGURAS

Figura 1 - Modelo de Dados	20
Figura 2 - Fluxograma	22
Figura 3 - Documentação de Código	23
Figura 4 - Ciclo de Vida do Desenvolvimento de Sistemas	24
Figura 5 - Modelo Físico de Dados	31
Figura 6 - Manutenção mais realizadas	44
Figura 7 - Documentos mais utilizados no entendimento do sistema	45

LISTA DE TABELAS

Tabela 1 - Tipos de Documentação	18
Tabela 2 - Dicionário de Dados.....	20
Tabela 3 - Questionário	29
Tabela 4 - Dicionário de Dados da Tabela pessoa.....	33
Tabela 5 - Dicionário de Dados da Tabela consulta.....	34
Tabela 6 - Dicionário de Dados da Tabela fila_espera	35
Tabela 7 - Dicionário de Dados da Tabela usuario.....	36
Tabela 8 - Dicionário de Dados da Tabela paciente.....	37
Tabela 9 - Dicionário de Dados da Tabela agenda	Erro! Indicador não definido.
Tabela 10 - Dicionário de Dados da Tabela alteracao_status_consulta	39
Tabela 11 - Dicionário de Dados da Tabela status_consulta.....	39
Tabela 12 - Dicionário de Dados da Tabela sala	40
Tabela 13 - Dicionário de Dados da Tabela chamado	40
Tabela 14 - Dicionário de Dados da Tabela clinica	41
Tabela 15 - Dicionário de Dados da Tabela profissional	41
Tabela 16 - Dicionário de Dados da Tabela status_pagamento.....	42
Tabela 17 - Dicionário de Dados da Tabela vinculo_instituicao	42
Tabela 18 - Dicionário de Dados da Tabela medico.....	42
Tabela 1 - Tipos de Documentação	18

LISTA DE ABREVIATURAS E SIGLAS

XP - Extreme Programming

SGCE - Software de Gerenciamento de Clínica-Escola

SDLC - Systems Development Life Cycle

UML – Unified Modeling Language

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVO	15
2 REFERENCIAL TEÓRICO	16
2.1 O QUE É DOCUMENTAÇÃO DE SOFTWARE	16
2.2 IMPORTÂNCIA DA DOCUMENTAÇÃO DE SOFTWARE	16
2.3 TIPOS DE DOCUMENTAÇÃO	18
2.4 FERRAMENTAS DE DOCUMENTAÇÃO	19
2.4.1 Modelo de Dados	19
2.4.2 Dicionário de Dados	20
2.4.3 Fluxogramas	21
2.4.4 Documentação de Código	22
2.5 O CICLO DE VIDA DO DESENVOLVIMENTO DE SISTEMAS	23
2.5.1 Levantamento das Necessidades	24
2.5.2 Análise de Alternativas	24
2.5.3 Projeto	25
2.5.4 Desenvolvimento	25
2.5.5 Implementação	25
2.5.6 Manutenção	25
3 METODOLOGIA	27
3.1 PESQUISA DE PADRÕES DE DOCUMENTAÇÃO	27
3.2 ORGANIZAÇÃO DO MATERIAL EXISTENTE	27
3.3 ELABORAÇÃO DE NOVOS ARTEFATOS	28
3.4 CONSTRUÇÃO DO PROJETO	28
3.5 PESQUISA SOBRE USO DE DOCUMENTAÇÃO DE SOFTWARE	28
4 RESULTADOS	30
4.1 ORGANIZAÇÃO DA DOCUMENTAÇÃO	30
4.2 MODELO FÍSICO DA BASE DE DADOS	30
4.3 DICIONÁRIOS DE DADOS	32
4.3.1 Tabela pessoa	33
4.3.2 Tabela consulta	34

4.3.3	Tabela fila_espera.....	35
4.3.4	Tabela usuario	36
4.3.5	Tabela paciente.....	37
4.3.6	Tabela agenda.....	38
4.3.7	<i>Tabela alteracao_status_consulta</i>	<i>39</i>
4.3.8	Tabela status_consulta	39
4.3.9	Tabela sala	40
4.3.10	Tabela chamado.....	40
4.3.11	Tabela clínica	41
4.3.12	Tabela profissional	41
4.3.13	Tabela status_pagamento.....	42
4.3.14	Tabela vinculo_instituicao	42
4.3.15	Tabela medico.....	42
4.3.16	Tabela cidade.....	43
4.3.17	Tabela estado.....	43
4.3.18	Tabela papel.....	43
4.4	RESULTADOS DA PESQUISA SOBRE USO DE DOCUMENTAÇÃO	43
4.4.1	Manutenções mais realizadas	44
4.4.2	Documentos mais utilizados no entendimento do sistema	45
4.4.3	Outros resultados.....	45
5	CONCLUSÃO.....	47
	REFERÊNCIAS	48

1 INTRODUÇÃO

Ao longo do tempo a documentação de software tem se caracterizado por ser o ponto fraco de muitas organizações da área de Tecnologia da Informação. Da perspectiva da Tecnologia da Informação, realizar a documentação de software é fator crítico para o sucesso, pois a documentação funciona como um componente central, para que o software suporte a gerência de mudança positivamente.

Cada vez mais a Engenharia de Software aprimora métodos, técnicas e ferramentas em busca de aumentar a qualidade e produtividade do desenvolvimento de software, neste contexto a documentação de software tem grande importância, pois pode influenciar tanto na qualidade quanto na produtividade. A falta ou excesso de documentação pode colocar em risco o desenvolvimento e a manutenção do software, por isso é fundamental encontrar um equilíbrio na quantidade de documentação gerada.

Muitas empresas têm dificuldades para desenvolverem seus softwares cumprindo os prazos e orçamentos estipulados, muitas vezes o prazo de entrega pode significar a sobrevivência ou não de uma organização, neste contexto a abordagem da documentação tem grande importância, no qual o prazo de entrega do projeto pode variar de acordo com a quantidade e qualidade dos artefatos a serem produzidos.

O Software de Gerenciamento de Clínica-Escola do UNI-RN (UNI-SGCE), sistema em desenvolvimento pelo Núcleo de Desenvolvimento de Software da instituição vem sendo desenvolvido desde o ano de 2010, atualmente conta com o módulo recepção já finalizado, onde são feitos os cadástrros de pacientes, discentes, horários de atendimento, salas para atendimento, marcação de consultas e filas de espera. Para o módulo já implantado, o módulo recepção do sistema, existiu a necessidade do desenvolvimento de uma documentação, a fim de auxiliar e agilizar as atividades de manutenção e também o desenvolvimento dos módulos posteriores.

Assim como em qualquer organização de Tecnologia da Informação, no Núcleo de Desenvolvimento de Software do UNI-RN acontece a constante mudança da equipe envolvida em projetos, pois se trata de um núcleo de extensão da instituição composta por discentes e docentes do curso de Bacharelado em sistemas de informação, no caso dos discentes que compõe o núcleo os mesmos finalizam a sua participação no grupo de extensão após a conclusão do curso superior, ter um

sistema documentado servirá como um grande apoio para os próximos ingressantes no curso de BSI que participarão do grupo, facilitando para os ingressantes o entendimento do sistema, o desenvolvimento dos módulos posteriores e manutenção de código.

O projeto do sistema UNI-SGCE conta com um vasto material para o desenvolvimento da documentação do sistema, materiais desenvolvidos por equipes que já participaram do núcleo e por equipes ainda participantes. O material restante continua em desenvolvimento, a fim de concluir o projeto.

1.1 OBJETIVO

O objetivo do projeto foi desenvolver a documentação de software do módulo recepção do UNI-SGCE contendo todas as regras de negocio desenvolvidas a respeito do sistema, desenvolvimento de novos artefatos de software necessários e informações sobre o sistema para complementar a estrutura da documentação, a fim de agilizar o processo de manutenção de código e entendimento do funcionamento do sistema por parte de futuros integrantes que irão desenvolver módulos posteriores necessários para a conclusão do sistema por completo e realizar uma pesquisa a respeito da documentação de software, a fim de mostrar se é algo importante para os desenvolvedores de sistemas, qual o documento importante para o entendimento do software por parte do desenvolvedor e a manutenção mais realizada por estes profissionais.

2 REFERENCIAL TEÓRICO

Para melhor compreensão da pesquisa bibliográfica que compreende o foco deste trabalho, faz-se necessário o conhecimento de alguns conceitos e idéias as quais fundamentam os aspectos abordados, sobre o desenvolvimento do projeto em questão. Sendo assim, este segundo capítulo objetiva explicar, de forma sucinta, alguns fundamentos centrais envolvidos na construção da documentação do módulo recepção do UNI-SGCE e a pesquisa realizada sobre documentação de software com profissionais desenvolvedores.

2.1 O QUE É DOCUMENTAÇÃO DE SOFTWARE

Conforme Ambler (2001) a documentação de *software* é qualquer artefato externo ao código fonte cujo propósito seja transmitir informação de uma maneira persistente.

Forward (2002) define a documentação como um artefato cuja finalidade seja comunicar a informação sobre o sistema de *software* ao qual ele pertence.

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.

2.2 IMPORTÂNCIA DA DOCUMENTAÇÃO DE SOFTWARE

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 os passos de seu desenvolvimento e isto inclui o conhecimento de seus usuários, expondo a eles "os requisitos reais do sistema" Booch (2006, p.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.

(Booch, 2006; Rumbaugh, 2006; Jacobsen 2006) destacam a importância do planejamento de 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 (2006, p.5).

Ambler (2001) destaca duas razões importantes para documentar um sistema: auxiliar a comunicação no período de desenvolvimento do *software* e auxiliar o entendimento nas atividades de manutenção e atualização quando se fizerem necessárias.

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 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*.

Diversos 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*, a documentação tem um efeito significativo no entendimento do programa.

Outro aspecto relevante é o cumprimento do prazo de entrega do software, fator essencial para a sobrevivência de uma organização, e que está diretamente

ligada à quantidade e qualidade da documentação. Alguns métodos ágeis vêm ganhando bastante importância, como é o caso do Extreme Programming (XP), que tem uma visão muito diferente, confia exclusivamente na comunicação oral, testes e código fonte para comunicar intenção e estrutura, trazendo para outras palavras, se aplicado, não existe nenhum documento de análise e projeto.

Entretanto, a falta de documentação pode representar um risco para os projetos de software, principalmente nas atividades de manutenção. Segundo (Freeman, 1992; Munro, 1992), a falta de documentação ou documentação incorreta afeta a compreensão do sistema, principalmente na atividade de análise de código.

2.3 TIPOS DE DOCUMENTAÇÃO

Segundo Michelazzo (2006) a documentação de software é dividida em dois grandes grupos, apresentados na tabela 1.

Tabela 1 - Tipos de Documentação

Documentação Técnica	Documentação de Uso
• Voltada ao desenvolvedor ou pessoa de TI. • Compreendem dicionários e modelos de dados; fluxogramas de processos e regras de negócios, dicionários de funções e comentários de código.	• Voltada para o usuário final quanto para o administrador do sistema • Formada por apostilas ou manuais que apresentam como o software deve ser usado, o que esperar dele e como receber as informações que se deseja.

Fonte: dados da pesquisa (2006).

A documentação técnica é considerada mais simples, pois descreve o trabalho do desenvolvedor e também é utilizada pelo mesmo como ferramenta para o desenvolvimento do sistema. Já a documentação de uso é mais exigente, pois é voltada para usuários e administradores do sistema, sendo composta por redação de manuais, screenshots, desenhos e outros elementos gráficos, buscando facilitar o entendimento do sistema para todos os tipos de usuários.

2.4 FERRAMENTAS DE DOCUMENTAÇÃO

Segundo Michelazzo (2006) apresentaremos as partes da documentação, resultado da divisão em duas grandes áreas e conheceremos algumas ferramentas que ajudam ou facilitam o desenvolvimento da documentação de sistemas.

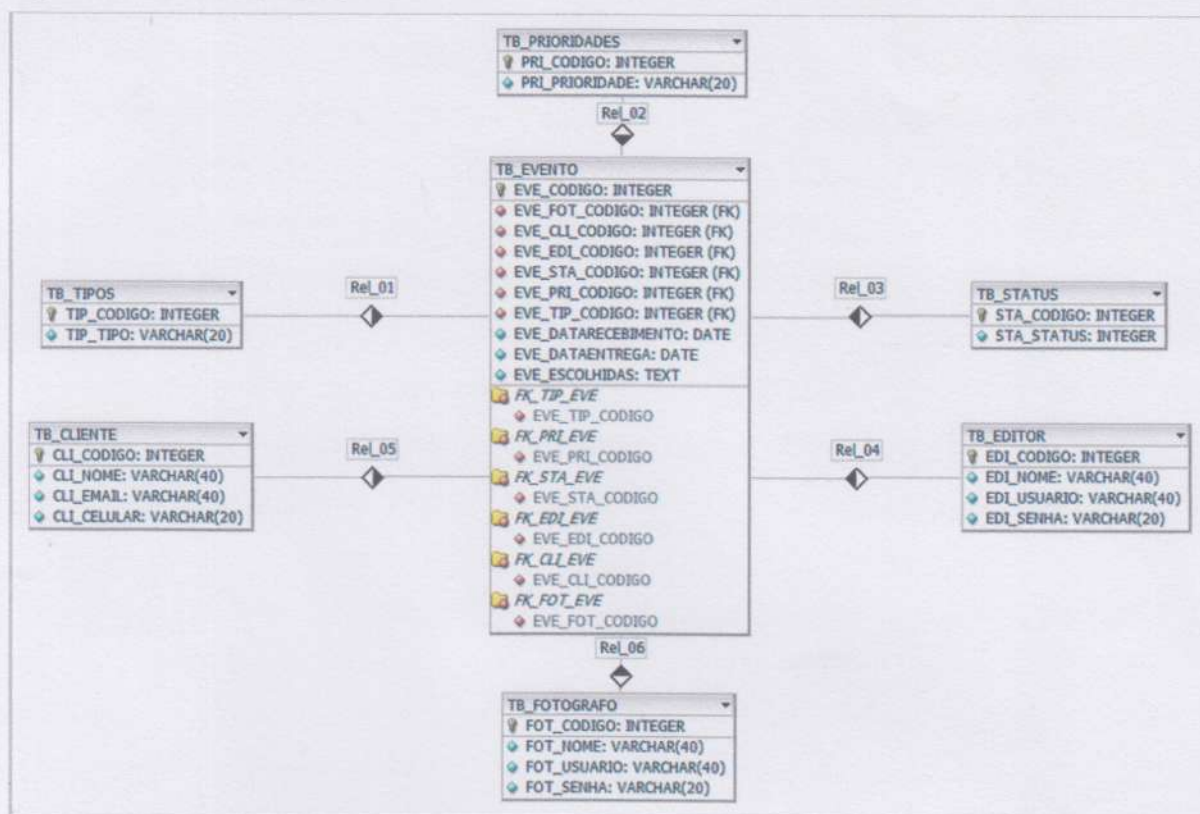
2.4.1 Modelo de Dados

De acordo com Michelazzo (2006) o modelo de dados reflete de forma gráfica (e lógica) a base de dados de um sistema, seus relacionamentos, entidades, chaves e tudo aquilo que é referente aos dados em si.

Este modelo é fundamental para o desenvolvimento de um sistema, sendo inclusive pensado antes do início do desenvolvimento do sistema. Um bom modelo de dados bem pensado e estruturado impacta em um bom código, um bom desempenho da aplicação e reduz horas de desenvolvimento equivocado.

Para criar o modelo de dados são utilizadas ferramentas de modelagem de dados, as quais geram de forma gráfica as tabelas, índices, relacionamentos e tudo aquilo que tem a ver com a base de dados em si, podendo ser criados por meio de engenharia reversa ou ainda baseando-se nas necessidades do aplicativo que está sendo desenvolvido. As ferramentas mais conhecidas para modelagem de banco de dados dentro do mundo livre são o DBDesigner, MySQL Workbench e PGDesigner, as quais possuem funcionalidades diferentes e dispõem de versões tanto para Linux quanto para Windows.

Figura 1 – Modelo de Dados



Fonte: Resultado da pesquisa do próprio autor.

2.4.2 Dicionário 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:

Tabela 2 - Dicionário de Dados

Campo	Tipo	Tamanho	Propriedades	Descrição
id_cadastro	int	5	auto_increment	ID do registro de 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 email
nu_fone	varchar	14	not null	Telefone de contato

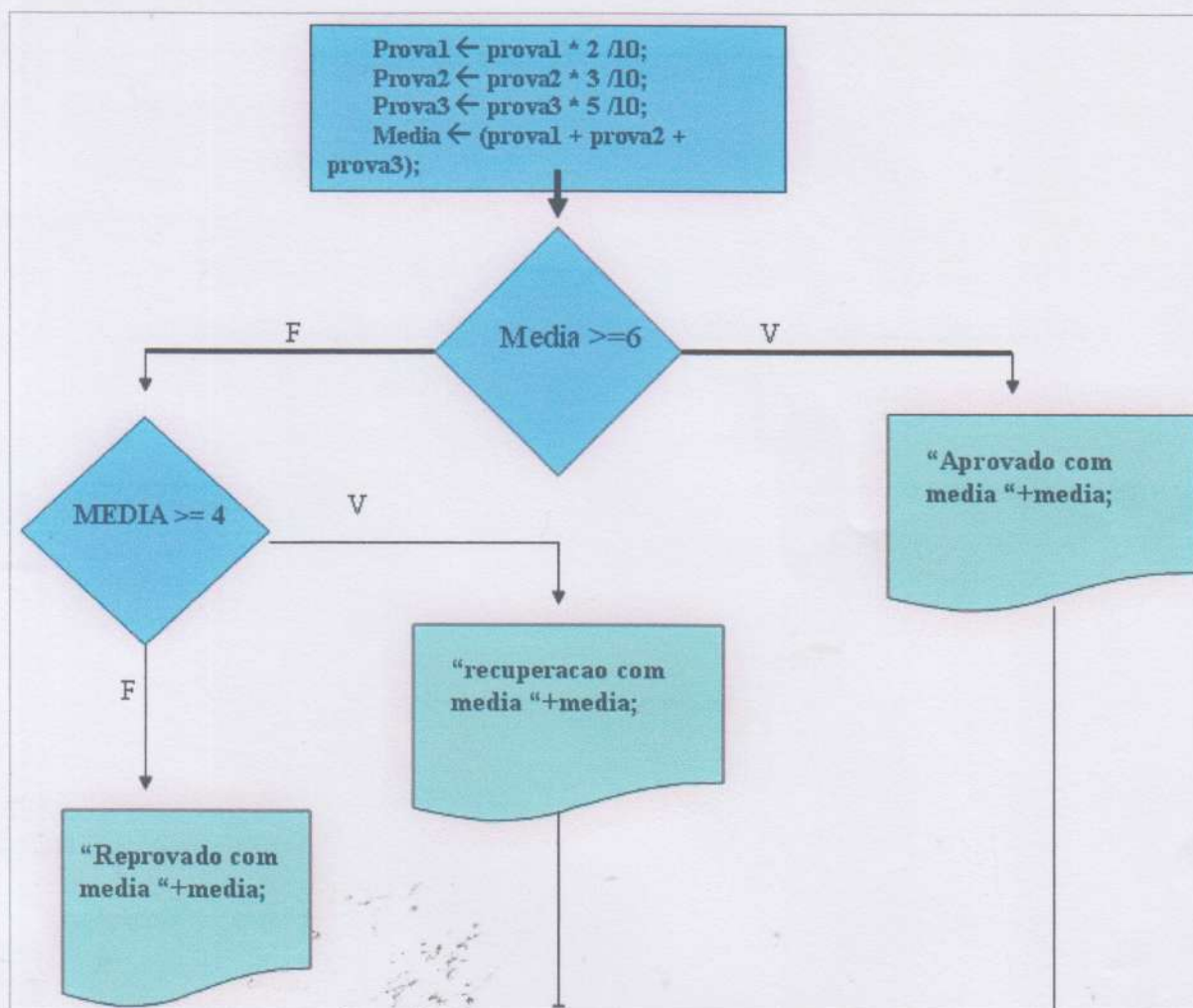
Fonte: <http://www.michelazzo.com.br/textos/dicas-para-documentacao-de-sofware>.

O dicionário de dados, conforme a tabela 2, juntamente com o modelo de dados, a possibilidade de erro na hora do desenvolvimento fica extremamente reduzida, levando em consideração o acesso à dados na base.

2.4.3 Fluxogramas

Apresentam graficamente a sequência lógica das informações de um processo ou sistema, utilizando para isso vários elementos de geometrias diferentes que indicam cada 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 do sistema. A seguir um exemplo de fluxograma:

Figura 2 - Fluxograma



Fonte: Disponível em: < <http://www.informaticon.com.br/j/index.php/algoritmos-e-logica.html> >

Acesso em: 27 nov. 2014.

Para criar fluxogramas, existem algumas ferramentas como, por exemplo, *DIA* e o *OpenModeling*, ambas disponíveis em varias plataformas e livres.

2.4.4 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 conjunto de comentários bem feito é tão importante quanto uma lógica bem estudada. Um exemplo de um código bem comentado pode ser visto a seguir:

Figura 3 - Documentação de Código

```

$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 ) {

```

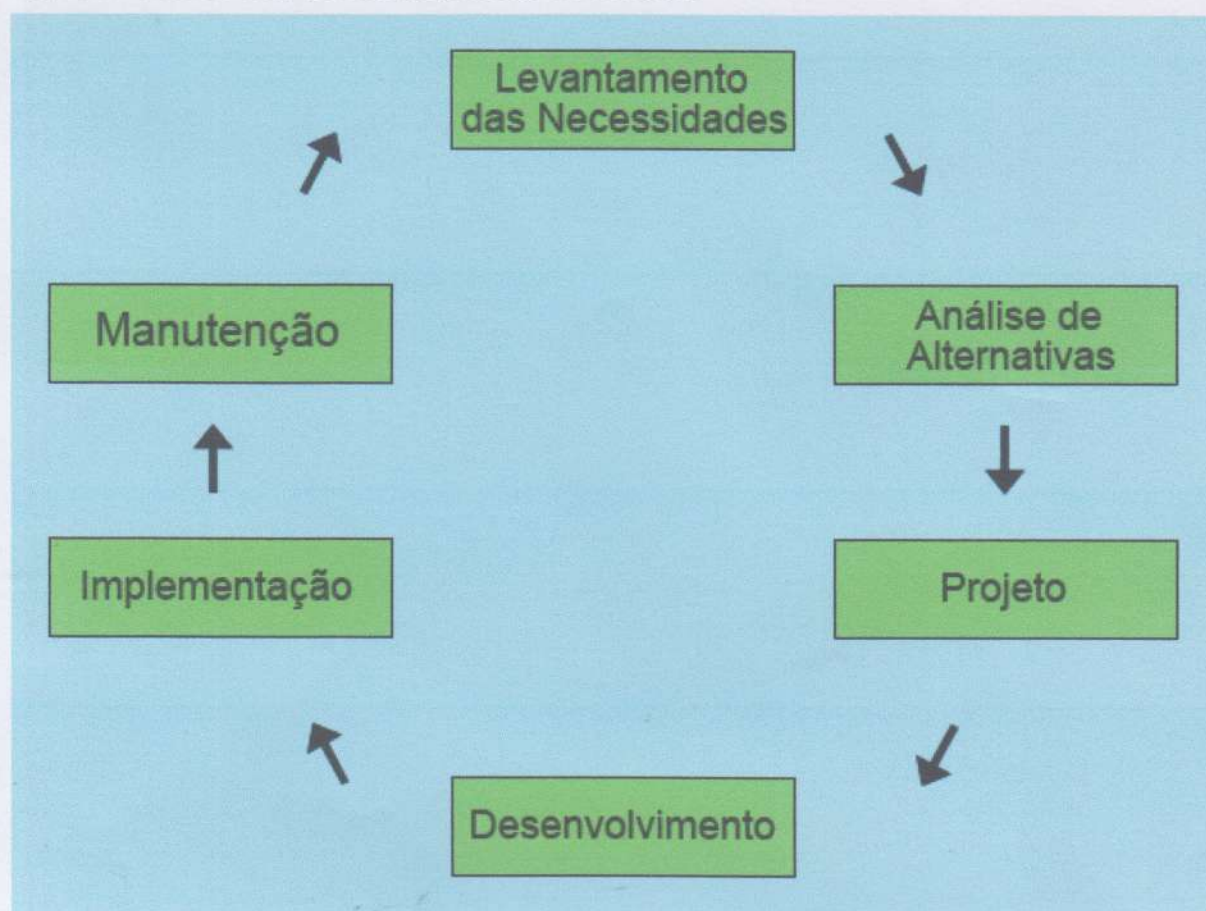
Fonte: Disponível em: < <http://www.michelazzo.com.br/textos/dicas-para-documentacao-de-sofware> > Acesso em: 27 nov. 2014.

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.

2.5 O CICLO DE VIDA DO DESENVOLVIMENTO DE SISTEMAS

O Ciclo de Vida do Desenvolvimento de Sistemas (SDLC – Systems Development Life Cycle), também conhecido como o “ciclo de vida do software” refere-se aos estágios de concepção, projeto, criação e implementação de Sistema de Informação.

Figura 4 - Ciclo de Vida do Desenvolvimento de Sistemas



Fonte: dados da pesquisa (2006).

2.5.1 Levantamento das Necessidades

Também chamado de análise de requisitos, identifica as necessidades de informações da organização, é uma das partes mais importantes do processo que resultará no desenvolvimento de um sistema, os requisitos descreve aquilo que o cliente deseja ou que o cliente acredita que precisa e as regras de negócio.

2.5.2 Análise de Alternativas

Consiste na identificação e avaliação de sistemas alternativos. O gerente de projetos deve analisar todas as diferentes alternativas para executar as atividades do projeto.

2.5.3 Projeto

Trata da construção das especificações detalhadas para o projeto selecionado. Essas especificações incluem o projeto de interfaces, banco de dados, características físicas do sistema, tais como número, e localizações das estações de trabalho, hardware de processamento, o cabeamento e os dispositivos de rede. Deve especificar os procedimentos para testar o sistema completo antes da instalação. Esta etapa é de bastante influência na etapa de manutenção, pois a qualidade dos artefatos produzidos facilitará ou não as atividades de manutenção.

2.5.4 Desenvolvimento

Nesta etapa inclui o desenvolvimento ou aquisição do software, a provável aquisição do hardware e o teste do novo sistema. Envolve atividades de concepção, especificação, design da interface, prototipação, design da arquitetura, codificação e verificação, dentre outras.

2.5.5 Implementação

Ocorre após o sistema ter passado satisfatoriamente por testes de aceitação. O sistema é transferido do ambiente de desenvolvimento para o ambiente de produção. O sistema antigo (caso exista) deve migrar para o novo.

2.5.6 Manutenção

Refere-se a todas as atividades relacionadas a um sistema depois que ele é implementado. Deve incluir atividades tais como a correção de software que não funcione corretamente, a adição de novos recursos aos sistemas em resposta às novas demandas dos usuários. A documentação de software é bastante importante nas atividades de manutenção, o desenvolvimento da documentação com artefatos de software de qualidade, deve acontecer na etapa de desenvolvimento do projeto do sistema.

Segundo Gordon (2006, p.356):

Não há modelo de SDLC uniformemente aceito. Alguns combinam desenvolvimento e implementação em uma única etapa. Outros combinam o levantamento e a análise das necessidades também em uma única etapa. Alguns modelos dividem o projeto em projeto lógico e projeto físico.

3 METODOLOGIA

Foi realizado um planejamento, e ficou dividido em quatro etapas o processo para desenvolvimento da documentação do sistema do módulo recepção do UNI-SGCE, também foi realizada uma pesquisa com desenvolvedores e mantenedores de sistemas, com o intuito de levantar algumas informações sobre o uso da documentação de software.

3.1 PESQUISA DE PADRÕES DE DOCUMENTAÇÃO

Inicialmente, precisava-se de uma forma para estruturar de maneira organizada o documento do software desejado, foram feitas pesquisas com relação a padrões utilizados para o desenvolvimento de documentação de software e os principais artefatos presentes numa documentação.

Com a pesquisa realizada, não foi possível obter sucesso com fontes que fornecessem uma estrutura adequada para ser adaptada conforme a desejada para a documentação do módulo recepção do UNI-SGCE, concluiu-se que seria elaborada uma estrutura de uma documentação de um produto de software baseado no material existente do núcleo e adaptado com novos que chegassem a ser elaborados, baseado em notações UML.

3.2 ORGANIZAÇÃO DO MATERIAL EXISTENTE

O projeto do sistema da Clínica-Escola do UNI-RN foi iniciado em 2010, com isso, para o desenvolvimento da documentação, já contava com alguns materiais, estes desenvolvidos por equipes que anteriormente participaram e por equipes atualmente participantes do projeto de extensão do UNI-RN, o núcleo de desenvolvimento de software.

Materiais como, casos de uso do sistema; diagrama de classe, diagrama de sequencia, modelo de domínio e arquitetura do sistema, foram utilizados para o desenvolvimento da documentação.

3.3 ELABORAÇÃO DE NOVOS ARTEFATOS

Por ser um documento bastante importante para um sistema de software, a documentação precisa conter os artefatos mínimos necessários para compreender todo o fluxo do sistema, em casos de futuras manutenções de código e adições de funcionalidades. Levando em consideração que uma equipe desenvolvedora nem sempre continua no mesmo projeto ou empresa, sendo assim, a documentação é o manual do sistema para o programador que entra no projeto após ser concluído.

Neste contexto, ficou clara a necessidade do desenvolvimento de mais recursos para a documentação, assim, foi focado o desenvolvimento de documentos que tratasse os aspectos relevantes ao banco de dados do sistema, que era de maior necessidade e possibilidade, considerando a existência dos outros materiais suficientes para a confecção da documentação, pois o sistema já se encontra em funcionamento, a elaboração de certos artefatos e documentos de software chega a ser dificultoso em sistemas já concluídos, pois deveriam ser desenvolvidos na fase de projeto, segundo as fases do ciclo de vida do desenvolvimento de sistemas.

3.4 CONSTRUÇÃO DO PROJETO

A partir do padrão de modelo inicialmente decidido, com todo material para a confecção selecionado, foi dado o início a adaptação de toda modelagem existente para o desenvolvimento da documentação do software do módulo recepção do UNISGCE.

3.5 PESQUISA SOBRE USO DE DOCUMENTAÇÃO DE SOFTWARE

Na pesquisa realizada, foram feitos alguns questionamentos a respeito de documentação, destacando a sua importância, manutenção realizada com o auxílio da documentação, sistemas mantidos com o auxílio da documentação e as ferramentas que facilitam no entendimento de uma futura manutenção. Segue o modelo na tabela:

Tabela 3 - Questionário

1. Considera a documentação de software necessária? () sim () não
2. Costuma criar algum tipo de modelagem para iniciar o desenvolvimento do sistema? () sim () não
3. Considera as modelagens desenvolvidas importantes para futuras manutenções ou adição de recursos no sistema? () sim () não
4. Tipo de manutenção que costuma realizar em algum sistema? () corretiva () adaptativa () perfectiva () preventiva
5. Abordagem dos sistemas mantidos? () estruturada () orientada a objetos
6. Marcar entre as ferramentas, a mais utilizada por você no processo de entendimento para a manutenção do sistema? () código fonte () comentários () modelo lógico de dados () modelo físico de dados () modelo de arquitetura do sistema () modelo de domínio () diagrama de sequencia () outro Qual?

Fonte: pesquisa do próprio autor.

4 RESULTADOS

Seguindo as etapas estabelecidas, foi possível obter a documentação do módulo recepção do sistema UNI-SGCE, com os principais artefatos de software possível a um sistema em pleno funcionamento, e com a pesquisa realizada foi possível desenvolver gráficos com base nos resultados.

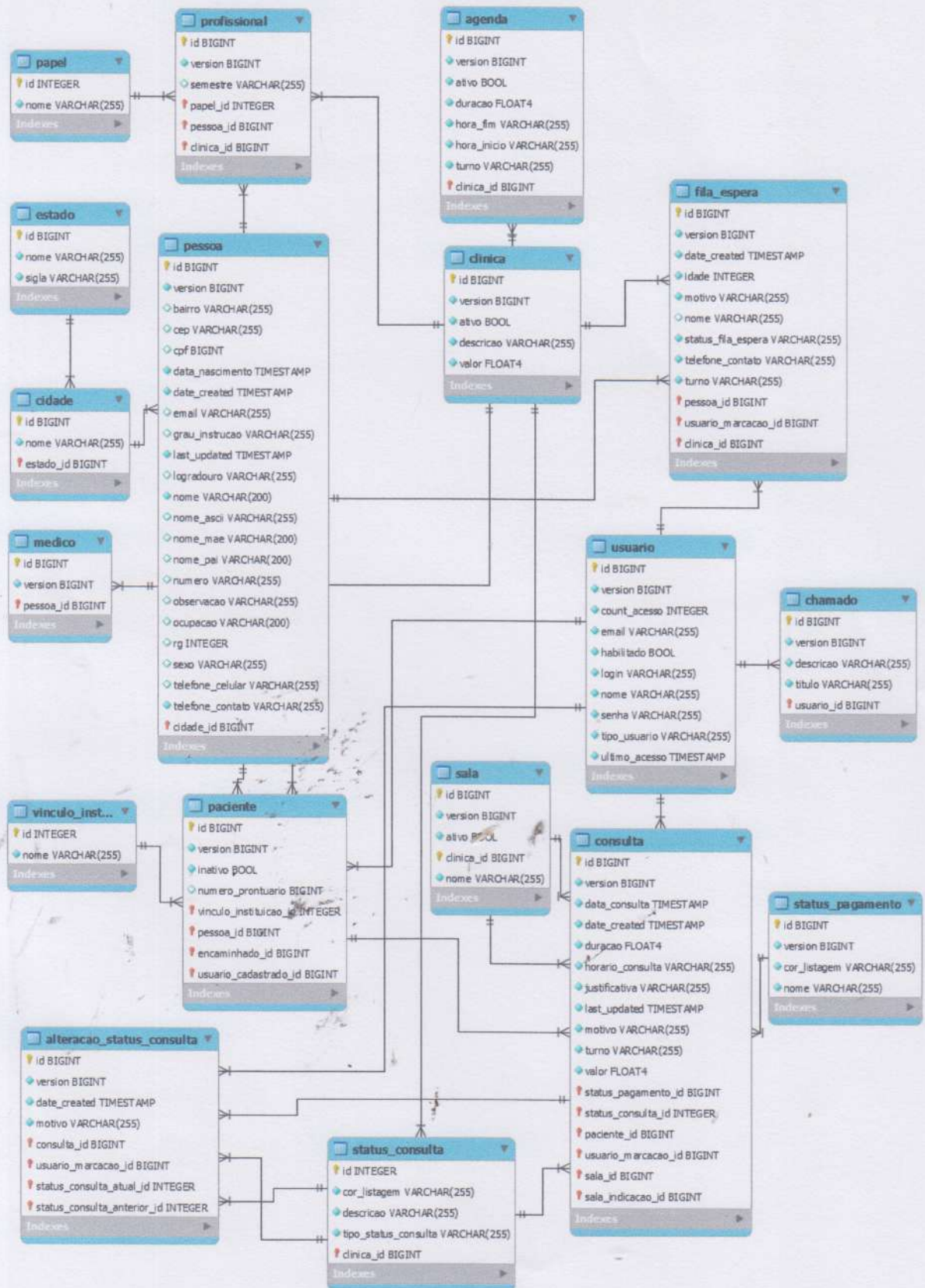
4.1 ORGANIZAÇÃO DA DOCUMENTAÇÃO

A documentação foi focada em três grandes tópicos, que contém os principais documentos do sistema, sendo esses tópicos, a Descrição Geral do Sistema, que descreve de forma geral o sistema, o escopo e as principais funções. Os requisitos do sistema, que tem o objetivo de descrever todos os requisitos do sistema e o terceiro grande tópico abordado na organização da documentação, é a parte de Análise e Design, que tem o objetivo de analisar e detalhar a solução do sistema de acordo com os requisitos levantados.

4.2 MODELO FÍSICO DA BASE DE DADOS

Modelo físico do banco gerado a partir de engenharia reversa usando a ferramenta *MicroOLAP Database Designer for PostgreSQL* e modelado no *MySQL Workbench*.

Figura 5 – Modelo Físico da Base de Dados



Fonte: pesquisa do próprio autor.

4.3 DICIONÁRIOS DE DADOS

Nos dicionários de dados referentes às tabelas do módulo recepção do UNI-SGCE, ficam descritos as informações detalhadas de cada tabela, identificando o campo, que consta o nome da coluna na tabela, o tipo, que especifica o tipo de dado referente à coluna, o tamanho, que informa o valor que o dado pode armazenar para a determinada coluna, a propriedade, informando a restrição para a coluna, e a descrição, que informa qual a função da coluna na tabela.

4.3.1 Tabela pessoa

Tabela 4 - Dicionário de Dados da Tabela pessoa

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
bairro	varchar	255	null	Bairro da pessoa cadastrada
cep	varchar	255	null	CEP da pessoa cadastrada
cpf	bigint	-	null	CPF da pessoa cadastrada
data_nascimento	timestamp	-	null	Nascimento da pessoa cadastrada
date_crated	timestamp	-	not null	Data do cadastro
email	varchar	255	null	Endereço de email
grau_instrução	varchar	255	null	O grau de estudo
last_update	timestamp	-	not null	Ultima atualização do cadastro
logradouro	varchar	255	null	Endereço do cadastrado
nome	varchar	200	not null	Nome do cadastrado
nome_ascii	varchar	255	null	-
nome_mae	varchar	200	null	Nome da mãe
nome_pai	varchar	200	null	Nome do pai
numero	varchar	255	null	Numero residência
observacao	varchar	255	null	Observações do cadastro
ocupado	varchar	200	null	status
rg	integer	-	null	RG do cadastrado
sexo	varchar	255	null	Sexo do cadastrado
telefone_celular	varchar	255	not null	Celular para contato
telefone_contato	varchar	255	null	Fixo para contato
cidade	bigint	-	null	Cidade (FK da tabela cidade)

Fonte: Resultado da pesquisa do próprio autor.

4.3.2 Tabela consulta

Tabela 5 - Dicionário de Dados da Tabela consulta

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
data_consulta	timestamp	-	not null	Data da consulta
data_created	timestamp	-	not null	Data de criação da consulta
duracao	float4	-	not null	Tempo de duração da consulta
horario_consulta	varchar	-	not null	Horário da consulta marcada
justificativa	varchar	255	not null	Justificativa da consulta
Last_update	timestamp	-	not null	Ultima atualização
motivo	varchar	255	not null	Motivo da consulta
turno	varchar	255	not null	Turno de realização
valor	float4	-	not null	Valor a ser pago
status_pagamento_id	bigint	-	null	Andamento do pagamento (FK tabela status_pagamento)
status_consulta_id	bigint	-	null	Status da consulta (FK da tabela status_consulta)
paciente_id	bigint	-	not null	Nome do paciente (FK da tabela paciente)
usuario_marcacao_id	bigint	-	not null	Usuário que criou a consulta (FK da tabela usuario)
sala_id	bigint	-	not null	Sala da consulta (FK da tabela sala)
sala_indicacao_id	float4	-	null	Local da sala (FK da tabela sala)

Fonte: Resultado da pesquisa do próprio autor.

4.3.3 Tabela fila_espera

Tabela 6 - Dicionário de Dados da Tabela fila_espera

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
data_created	timestamp	-	not null	Data de criação
idade	integer	-	not null	Idade do paciente que se encontra na fila de espera
motivo	varchar	255	not null	Motivo da consulta
nome	varchar	255	null	Nome paciente
status_fila_espera	varchar	255	not null	Andamento na fila de espera
telefone_contato	varchar	255	not null	Telefone para contato
turno	varchar	255	not null	Turno da consulta
pessoa_id	varchar	255	not null	ID da pessoa (FK da tabela pessoa)
usuario_marcacao	float4	-	not null	Usuário que marcou a consulta
clinica_id	bigint	-	not null	ID da clinica (FK da tabela clinica)

Fonte: Resultado da pesquisa do próprio autor.

4.3.4 Tabela usuario

Tabela 7 - Dicionário de Dados da Tabela usuario

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
count_acesso	integer	-	not null	Contador de acessos do usuário
email	varchar	255	not null	Email do usuário
habilitado	bool	-	not null	Indicador de ativo ou não no sistema
login	varchar	255	not null	Nome de acesso ao sistema
nome	varchar	255	not null	Nome do usuário
senha	varchar	255	not null	Senha de acesso ao sistema
tipo_usuario	varchar	255	not null	Tipo de usuário no sistema
ultimo_acesso	timestamp	-	null	Registro de ultimo acesso

Fonte: Resultado da pesquisa do próprio autor.

4.3.5 Tabela paciente

Tabela 8 - Dicionário de Dados da Tabela paciente

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
inativo	bool	-	not null	Ativo ou não no sistema
numero_prontuario	bigint	-	null	Numero do prontuário do paciente
vinculo_instituicao_id	integer	-	not null	Matricula na instituição
pessoa_id	bigint	-	not null	ID do paciente (FK na tabela pessoa)
encaminhado_id	bigint	-	null	Encaminhamento para uma clinica (FK da tabela clinica)
usuario_cadastrado_id	bigint	-	not null	ID do usuario do sistema (FK da tabela usuario)

Fonte: Resultado da pesquisa do próprio autor.

4.3.6 Tabela agenda

Tabela 9 - Dicionário de Dados da Tabela agenda

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
ativo	bool	-	not null	Ativo ou não no sistema
duracao	float4	-	not null	Duração da consulta registrada na agenda
hora_fim	varchar	255	not null	Fim para término da consulta na agenda
hora_inicio	varchar	255	not null	Hora de início da consulta marcada na agenda
turno	varchar	255	not null	Turno registrado na agenda
clinica_id	bigint	-	not null	ID da clinica agendada (FK da tabela clinica)

Fonte: Resultado da pesquisa do próprio autor.

4.3.7 Tabela alteracao_status_consulta

Tabela 10 - Dicionário de Dados da Tabela alteracao_status_consulta

Campo	Tipo	Tamanho	Propriedades	Descrição
id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
date_created	timestamp	-	not null	Data de cadastro no sistema
motivo	varchar	255	not null	Motivo da alteração
consulta_id	bigint	-	not null	ID da consulta (FK da tabela consulta)
usuario_marcao	bigint	-	not null	Usuario que fez a alteração
status_consulta_atual_id	integer	-	not null	Status da consulta que vai acontecer (FK da tabela status_consulta)
status_consulta_anterior	integer	-	not null	Status da consulta que aconteceu anteriormente (FK da tabela status_consulta)

Fonte: Resultado da pesquisa do próprio autor.

4.3.8 Tabela status_consulta

Tabela 11 - Dicionário de Dados da Tabela status_consulta

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	Integer	-	auto_increment	ID do registro de cadastro
cor_listagem	varchar	255	not null	Listagem da consulta
descricao	varchar	255	not null	Descricao dos status
tipo_status_consulta	varchar	255	not null	Tipo do status
clinica_id	bigint	-	not null	ID da clinica (FK da tabela clinica)

Fonte: Resultado da pesquisa do próprio autor.

4.3.9 Tabela sala

Tabela 12 - Dicionário de Dados da Tabela sala

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
ativo	bool	-	not null	Ativo ou não no sistema
clinica_id	bigint	-	not null	ID da clinica (FK da tabela clinica)
nome	varchar	255	not null	Nome da sala

Fonte: Resultado da pesquisa do próprio autor.

4.3.10 Tabela chamado

Tabela 13 - Dicionário de Dados da Tabela chamado

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
descricao	varchar	255	not null	Descricao do chamado
titulo	varchar	255	not null	Titulo do chamado
usuario_id	varchar	-	not null	ID do usuario (FK da tabela usuario)

Fonte: Resultado da pesquisa do próprio autor.

4.3.11 Tabela clinica

Tabela 14 - Dicionário de Dados da Tabela clinica

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
ativo	bool	-	not null	Ativo ou não
descricao	varchar	255	not null	Descrição da clinica
valor	float4	-	not null	Valor de consulta na clinica especifica

Fonte: Resultado da pesquisa do próprio autor.

4.3.12 Tabela profissional

Tabela 15 - Dicionário de Dados da Tabela profissional

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
semestre	varchar	255	not null	Situação acadêmica do profissional
papel_id	integer	-	null	ID papel (FK da tabela papel)
pessoa_id	bigint	-	not null	ID da pessoa (FK da tabela pessoa)
clinica_id	bigint	-	null	ID da clinica (FK da tabela clinica)

Fonte: Resultado da pesquisa do próprio autor.

4.3.13 Tabela status_pagamento

Tabela 16 - Dicionário de Dados da Tabela status_pagamento

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
cor_listagem	varchar	255	not null	Listagem do status de pagamento
nome	varchar	255	not null	Descricao do status

Fonte: Resultado da pesquisa do próprio autor.

4.3.14 Tabela vinculo_instituicao

Tabela 17 - Dicionário de Dados da Tabela vinculo_instituicao

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	integer	-	auto_increment	ID do registro de cadastro
nome	varchar	255	not null	Descricao do vinculo com a instituição (matricula)

Fonte: Resultado da pesquisa do próprio autor.

4.3.15 Tabela medico

Tabela 18 - Dicionário de Dados da Tabela medico

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	integer	-	auto_increment	ID do registro de cadastro
version	bigint	-	not null	Controlar manipulação do objeto no banco, tem função de versionar
pessoa_id	bigint	-	not null	ID da pessoa (FK da tabela pessoa)

Fonte: Resultado da pesquisa do próprio autor.

4.3.16 Tabela cidade

Tabela 19 - Dicionário de Dados da Tabela cidade

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
nome	varchar	255	not null	Nome da cidade
estado_id	bigint	-	not null	Estado referente a cidade (FK da tabela estado)

Fonte: Resultado da pesquisa do próprio autor.

4.3.17 Tabela estado

Tabela 20 - Dicionário de Dados da Tabela estado

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	bigint	-	auto_increment	ID do registro de cadastro
nome	varchar	255	not null	Nome do estado
sigla	varchar	255	not null	Sigla do estado

Fonte: Resultado da pesquisa do próprio autor.

4.3.18 Tabela papel

Tabela 21 - Dicionário de Dados da Tabela papel

Campo	Tipo	Tamanho	Propriedades	Descrição
Id	integer	-	auto_increment	ID do registro de cadastro
nome	varchar	255	not null	Nome do estado

Fonte: Resultado da pesquisa do próprio autor.

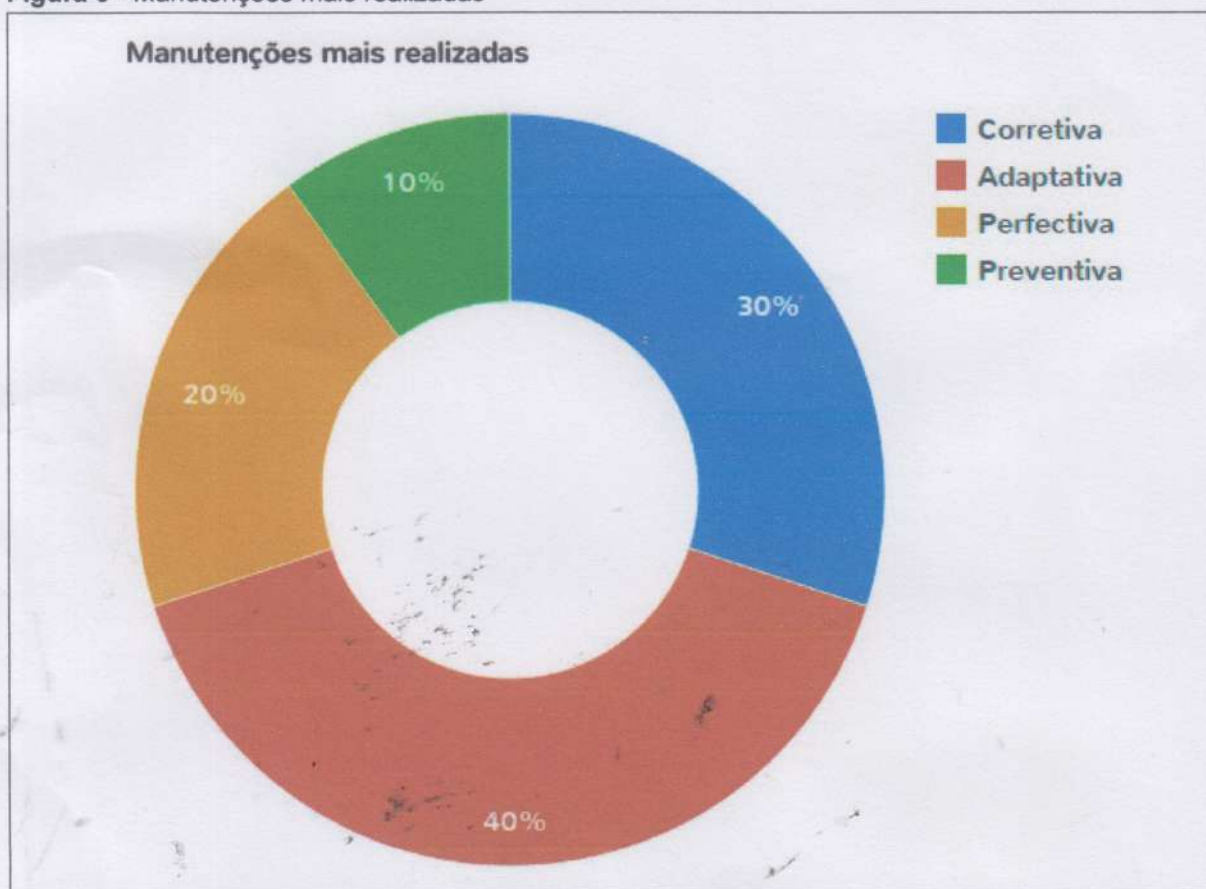
4.4 RESULTADOS DA PESQUISA SOBRE USO DE DOCUMENTAÇÃO

Os resultados da pesquisa foram obtidos através da resolução do questionário, no qual foi resolvido por 20 profissionais da área de TI que trabalham com desenvolvimento de sistemas.

4.4.1 Manutenções mais realizadas

O gráfico de rosca destaca os tipos de manutenções, na qual a de maior ocorrência na pesquisa foi a manutenção Adaptativa, com 40%, seguido ela manutenção Corretiva com 30%, a Perfectiva com 20% e por ultimo a Preventiva com 10%.

Figura 6 - Manutenções mais realizadas

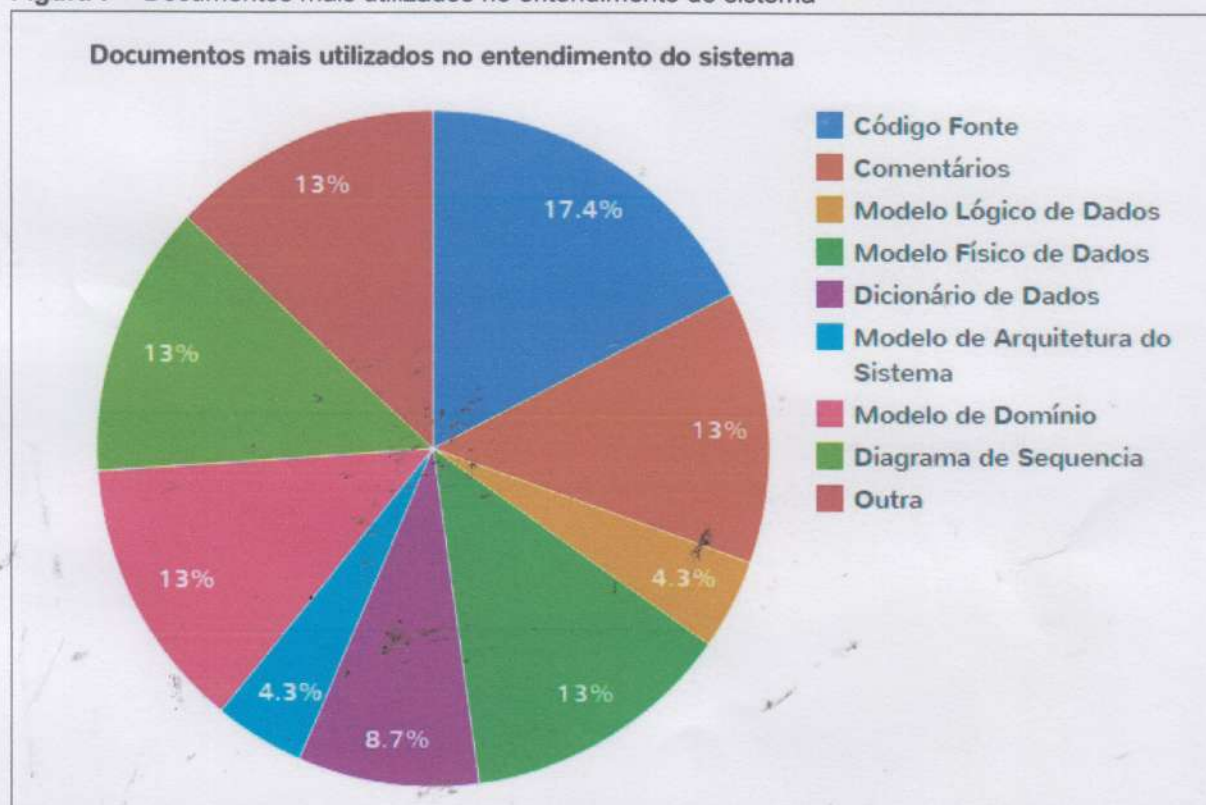


Fonte: Resultado da pesquisa do próprio autor.

4.4.2 Documentos mais utilizados no entendimento do sistema

O gráfico de pizza informa o documento mais utilizado pelos entrevistados no momento de entendimento do sistema, sendo o código fonte o de maior preferência dos entrevistados 17.4%, seguido dos Comentários, Modelo Físico de Dados, Diagrama de Sequencia, Modelo de Domínio e Outra (ferramentas não incluídas na pesquisa), esses documentos ficando 13% da preferência, seguido por Dicionário de Dados com 8.7% e Modelo de Arquitetura do Sistema e Modelo Lógico de Dados com 4.3%.

Figura 7 – Documentos mais utilizados no entendimento do sistema



Fonte: Resultado da pesquisa do próprio autor.

4.4.3 Outros resultados

Dos desenvolvedores entrevistados, 100% consideraram a documentação de software necessária, 100% criam algum tipo de modelagem para iniciar o desenvolvimento de sistemas e as consideram importantes para futuras

manutenções e adições de funcionalidades, e 100% dos desenvolvedores trabalham com sistema de abordagem orientada a objetos.

5 CONCLUSÃO

Na confecção da documentação do UNI-SGCE módulo recepção existiram algumas lacunas, pois se tratava de um sistema já implantado e alguns documentos de software não foram criados antes de sua implantação, a etapa de desenvolvimento, e não foi possível a criação no período de confecção da documentação, documentos como, o plano de testes, que aborda de forma detalhada o fluxo de trabalho durante o processo e alguns diagramas tratando as funções do sistema para compor a documentação.

Mesmo com as ausências foi possível criar uma documentação mínima capaz de orientar no entendimento do software, desenvolvendo alguns materiais referentes ao banco de dados, através de engenharia reversa, juntamente com outros artefatos de software já existentes. Dessa forma, a documentação do módulo recepção do UNI-SGCE será uma ferramenta facilitadora para os próximos desenvolvedores a entrarem no projeto entenderem o sistema de forma mais simples, baseando-se em documentos do software, facilitando nas atividades de manutenção e adições de módulos posteriores. Portanto, para trabalhos futuros ainda á necessidade da criação de uma documentação de uso, ou seja, o manual do usuário, pois a criada para o módulo recepção do UNI-SGCE foi uma documentação técnica, voltada para o desenvolvedor.

Com a pesquisa realizada, foi possível descobrir alguns aspectos sobre a documentação e manutenção de sistemas, se a documentação é importante para os desenvolvedores, a ferramenta que realmente facilita no entendimento do sistema e descobrir se os documentos de software costuma ser desenvolvidos na etapa de projeção do sistema.

REFERÊNCIAS

AMBLER, Scott W. **Agile Documentation**. 2001-2004, The Official Agile Modeling (AM). 2014. Disponível em: <<http://www.agilemodeling.com/essays/agileDocumentation.htm>>. Acesso em: 24 nov. 2014.

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

FORWARD, Andrew. **Software Documentation – Building and Maintaining Artefacts of Communication**. 2002. Teste de Mestrado - Universidade de Ottawa, Canadá, Ottawa, Toronto, 2002.

FREEMAN, Robert M.; MUNRO, Malcolm. **Redocumentation for the Maintenance of Software**, Proceedings of the 30th annual Southeast regional conference, ACM, 1992, p.413-416.

GORDON, Steven R.; Judith R. **Sistemas de Informação uma Abordagem Gerencial**. São Paulo: LTC, 2006.

MICHELLAZZO, P. A **Documentação para documentação de software**. 2006. Disponível em: <<http://www.michelazzo.com.br/textos/dicas-para-documentacao-de-sofware>> Acesso em: 24 nov. 2014.