



FACULDADE LOURENÇO FILHO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
CLÁUDIO PINTO DE ARAGÃO

AMBIENTE COMPUTACIONAL DE GESTÃO EM UMA CLÍNICA MÉDICA

Monografia elaborada pelo acadêmico Cláudio Pinto de Aragão como exigência do curso de graduação em Ciência da Computação da Faculdade Lourenço Filho, sob a orientação do Professor Dr. Gerardo Valdisio Rodrigues Viana.

Fortaleza – Junho / 2010

Cláudio Pinto de Aragão

AMBIENTE COMPUTACIONAL DE GESTÃO EM UMA CLÍNICA MÉDICA

Monografia elaborada pelo acadêmico Cláudio Pinto de Aragão como exigência do curso de graduação em Ciência da Computação da Faculdade Lourenço Filho, sob a orientação do Professor Dr. Gerardo Valdisio Rodrigues Viana.

Fortaleza – Junho / 2010

Cláudio Pinto de Aragão

AMBIENTE COMPUTACIONAL DE GESTÃO EM UMA CLÍNICA MÉDICA

Monografia Apresentada ao curso de Bacharelado em Ciência da Computação da Faculdade Lourenço Filho, como parte dos requisitos necessários à obtenção do grau de Bacharel em Ciência da Computação.

Aprovada em: ____/____/____

Conceito: _____

Composição da Banca Examinadora:

Prof. Dr. Gerardo Valdisio Rodrigues Viana

Faculdade Lourenço Filho – FLF e Universidade Estadual do Ceará - UECE

Orientador

Prof. Esp. Hélio Augusto Sabóia Moura

Faculdade Lourenço Filho – FLF

Prof. Dr. Antônio Clécio Fontelles Thomaz

Universidade Estadual do Ceará - UECE) - Membro Externo

Prof^a Msc Márcia Terezinha Toniato

Coordenador do Curso

AGRADECIMENTOS

À Deus, por está sempre guiando meus passos e me levando para o caminho do bem;

À minha família, pelo exemplo, compreensão e pelo apoio incondicional;

À pessoa do Dr. Gothardo Peixoto Figueiredo Lima, pela grande amizade e oportunidade de me permitir explorar o meu conhecimento prático em sua clínica médica.

Ao professor orientador Dr. Gerardo Valdisio Rodrigues Viana, pela disponibilidade em contribuir e compartilhar para a realização deste trabalho;

Aos professores da banca examinadora, pela disponibilidade e presteza da leitura, em especial o professor Dr. Antônio Clécio Fontelles Thomaz;

A todos os amigos e colegas de turma, pelos momentos de alegrias e parceria.

Resumo

Nesta monografia estudamos o ambiente computacional da gestão em uma clínica médica especializada, buscando esclarecer por meio de uma linguagem simples e de fácil entendimento todo o processo de funcionamento de um sistema voltado para a medicina dos dias atuais. A utilização de um *software* para uma clínica médica na especialidade de oncologia clínica, que tem como principal objetivo, proporcionar um tratamento diferenciado, que garanta qualidade de vida a pacientes de câncer, serviu para aprofundar nosso conhecimento prático sobre a importância de desenvolvermos e aprimorarmos ferramentas computacionais que possam contribuir significativamente para que os pacientes sejam sempre os maiores beneficiados por um atendimento de qualidade, além da contribuição significativa desse sistema para a preservação das informações e gerenciamento das tomadas de decisões. Além da abordagem prática, procuramos nos basear também na revisão de pesquisas bibliográficas em algumas áreas da Ciência da Computação, especificamente, desenvolvimento de sistemas, banco de dados e engenharia de *software*, que nos permitiram um entendimento de como um *software* é projetado desde o seu início até a entrega final ao cliente, onde será oferecido posteriormente treinamento adequado, manutenções preventivas ou corretivas e atualizações de acordo com o avanço tecnológico e necessidades futuras de funcionalidades.

PALAVRAS-CHAVE: *Software* médico, ferramentas computacionais, gestão médica e ambiente computacional.

Sumário

INTRODUÇÃO.....	10
APRESENTAÇÃO DO TEMA.....	10
METODOLOGIA CIENTÍFICA	10
OBJETIVOS.....	11
JUSTIFICATIVA E MOTIVAÇÃO	11
ORGANIZAÇÃO DO TRABALHO.....	12
CAPÍTULO 1 – CARACTERIZAÇÃO DO AMBIENTE ESTUDADO	14
1.1 – CENÁRIO DO AMBIENTE INTERNO E EXTERNO	14
1.2 – SURGIMENTO DO AMBIENTE COMPUTACIONAL	15
1.3 - ORGANOGRAMA FUNCIONAL	16
1.4 – INSTALAÇÕES FÍSICAS.....	17
1.5 - ESTRUTURA COMPUTACIONAL: <i>HARDWARE</i> E <i>SOFTWARE</i>	17
1.5.1 - <i>Hardware</i>	17
1.5.2 - <i>Software</i>	18
CAPÍTULO 2 – O SISTEMA COMPUTACIONAL.....	19
2.1 – INTRODUÇÃO	19
2.2 – HISTÓRICO DO SISTEMA NA GESTÃO DA CLÍNICA.....	20
2.3 – CARACTERÍSTICAS DO SISTEMA.....	20
2.4 – FUNCIONAMENTO DO PROCESSO RESUMIDO DE ROTINAS DO SISTEMA	23
2.5 - FLUXOGRAMAS DE DESCRIÇÃO DAS ATIVIDADES	25
2.5.1 – <i>Marcação de Consultas</i>	26
2.5.2 – <i>Consultas</i>	27
2.5.3 – <i>Cadastro de Pacientes</i>	28
2.5.4 – <i>Procedimento Ambulatorial</i>	29
2.5.5 - <i>Faturamento</i>	30
2.6 – SUPORTE E MANUTENÇÃO.....	31
2.7 – ROTINAS DE BACKUPS E SEGURANÇA	31
2.8 – MODELOS DE RELATÓRIOS (INFORMAÇÕES GERADAS E SAÍDAS)	32
2.9 – INTERFACE VISUAL DO SISTEMA	32
CAPÍTULO 3 – FERRAMENTAS COMPUTACIONAIS PARA DESENVOLVIMENTO DE <i>SOFTWARE</i>	34
3.1 – INTRODUÇÃO E CLASSIFICAÇÃO DAS LINGUAGENS DE COMPUTAÇÃO	34
3.2 – LINGUAGENS DE ALTO NÍVEL	35
3.2.1 – <i>Linguagem Delphi</i>	35
3.2.2 – <i>Linguagem Java</i>	38
3.2.3 – <i>Comparativos – Vantagens e Desvantagens: Delphi x Java</i>	41
3.2.4 - <i>Resultado Comparativo: Java x Delphi</i>	42
CAPÍTULO 4 – SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD).....	44
4.1- DEFINIÇÃO	44
4.2 – BREVE HISTÓRICO	44
4.3 – ESTRUTURA LÓGICA	45
4.4 – CARACTERÍSTICAS E FUNCIONALIDADES	46
4.5 - MODELOS DE BANCO DE DADOS	46
4.5.1 – <i>Modelo Relacional</i>	46
4.6- FERRAMENTAS DE BANCOS DE DADOS	48
4.6.1 - <i>Paradox</i>	49
4.6.2 - <i>SQL Server (Structured Query Language)</i>	50
4.6.3 – <i>Extensible Markup Language (XML)</i>	52

CAPÍTULO 5 – ENGENHARIA DE SOFTWARE	57
5.1- INTRODUÇÃO	57
5.2 – TÉCNICAS DE COMUNICAÇÃO: DESENVOLVEDOR X CLIENTE	58
5.3 - PLANEJAMENTO, IDENTIFICAÇÃO DOS RISCOS, PRAZOS, MONITORAMENTO E CONTROLE	59
5.4 – GERÊNCIA DE PROJETOS DE <i>SOFTWARE</i>	59
5.4.1 - <i>Análise e Definição de Requisitos</i>	60
5.4.2 - <i>Projeto do Sistema</i>	62
5.4.3 - <i>Projeto do Programa ou Implementação</i>	63
5.4.4 - <i>Testes de Unidades, Integração e Sistema</i>	64
5.4.5 - <i>Entrega do Sistema</i>	64
5.4.6 - <i>Manutenção</i>	65
CONCLUSÃO.....	66
CONSIDERAÇÕES FINAIS.....	66
RECOMENDAÇÕES	67
REFERÊNCIAS BIBLIOGRÁFICAS	69

Lista de Figuras

FIGURA 1 - TELA INICIAL DO SISTEMA DA CLÍNICA MÉDICA (VERSÃO ATUAL)	20
FIGURA 2 - MODELO DE RELATÓRIO GERADO PELO SISTEMA	32
FIGURA 3 - PAINEL DE CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO <i>DELPHI</i>	36
FIGURA 4 - COMUNICAÇÃO <i>DELPHI</i> COM BANCO DE DADOS.	37
FIGURA 5 – AMBIENTE DE DESENVOLVIMENTO EM <i>JAVA</i>	40
FIGURA 6 – EXEMPLO LÓGICO DE MODELO ENTIDADE/RELACIONAMENTO	45
FIGURA 7 - DIAGRAMA DE BANCO DE DADOS DO SISTEMA UTILIZADO	49
FIGURA 8 – CARACTERIZAÇÃO DA ENGENHARIA DE <i>SOFTWARE</i>	58
FIGURA 9- ELABORAÇÃO DE DEFINIÇÃO DE REQUISITOS	61

Lista de Quadros

QUADRO 1- MÓDULO "RECEPÇÃO"	21
QUADRO 2 - MÓDULO "ATENDIMENTO"	22
QUADRO 3 - MÓDULO "FATURAMENTO"	22
QUADRO 4 - MÓDULO "CONFIGURAÇÕES"	23
QUADRO 5 - FLUXO PRINCIPAL DO CENÁRIO "MARCAÇÃO DE CONSULTAS"	26
QUADRO 6 - FLUXO ALTERNATIVO DO CENÁRIO "MARCAÇÃO DE CONSULTAS"	26
QUADRO 7 - FLUXO PRINCIPAL DO CENÁRIO "CONSULTAS"	27
QUADRO 8 - FLUXO ALTERNATIVO DO CENÁRIO "CONSULTAS"	28
QUADRO 9 - FLUXO PRINCIPAL DO CENÁRIO "CADASTRO DE PACIENTES"	28
QUADRO 10 - FLUXO ALTERNATIVO DO CENÁRIO "CADASTRO DE PACIENTES"	29
QUADRO 11 - FLUXO PRINCIPAL DO CENÁRIO "PROCEDIMENTO AMBULATORIAL"	29
QUADRO 12 - FLUXO ALTERNATIVO DO CENÁRIO "PROCEDIMENTO AMBULATORIAL"	29
QUADRO 13 - FLUXO PRINCIPAL DO CENÁRIO "FATURAMENTO DO PROCEDIMENTO AMBULATORIAL"	30
QUADRO 14 - FLUXO ALTERNATIVO DO CENÁRIO "FATURAMENTO DO PROCEDIMENTO AMBULATORIAL"	30
QUADRO 15- EXEMPLO DE CÓDIGO-FONTE EM ASSEMBLY	34
QUADRO 16 - ESTRUTURA DE CÓDIGO-FONTE EM <i>DELPHI</i>	37
QUADRO 17 - EXEMPLO DE CÓDIGO-FONTE UTILIZANDO HERANÇA EM <i>JAVA</i>	39
QUADRO 18 - EXEMPLO DE CÓDIGO-FONTE UTILIZANDO POLIMORFISMO EM <i>JAVA</i>	39
QUADRO 19 – DEFINIÇÃO DA REGRA DE ESQUEMA EM <i>SQL</i>	51
QUADRO 20 - EXEMPLO DE ESQUEMA <i>SQL</i> : CRIAÇÃO DA TABELA "MEDICOS"	51
QUADRO 21 - EXEMPLO DE FORMATAÇÃO XML PARA ENVIO DE FATURAMENTO ENTRE A CLÍNICA E CONVÊNIOS.	54
QUADRO 22 - EXEMPLOS DE REQUISITOS FUNCIONAIS	62
QUADRO 23 - EXEMPLOS DE REQUISITOS NÃO-FUNCIONAIS	62

Lista de Tabelas

TABELA 1 - COMPARATIVO DE TEMPO ENTRE AS ATIVIDADES REALIZADAS (VALORES APROXIMADOS).....	33
TABELA 2 - TABELA DE RELACIONAMENTO "PACIENTES"	47
TABELA 3 - TABELA DE RELACIONAMENTO "MÉDICOS"	47

Introdução

O cotidiano de atividades desenvolvidas pelo ser humano em sua jornada de trabalho tem tornado-se cada vez mais complexo e no mercado competitivo que enfrentamos nos dias atuais, exige-se um maior e mais eficaz controle de todo o funcionamento da gerência do negócio de uma simples empresa a uma grande indústria, nos diversos ramos existentes, necessitando para isso de um apoio computacional que seja capaz de facilitar todo esse mecanismo, tornando a tomada de decisões e a produção humana mais otimizada, segura e organizada.

Apresentação do Tema

O surgimento e aprimoramento de *software* comerciais e o estudo de sua engenharia, a partir da década de 80 até hoje, foram capazes de criar sistemas e ferramentas computacionais que geram mecanismos de auxílio para gerenciar os mais diversos tipos de atividades existentes, entre estas, a medicina e as diversas outras áreas envolvidas direta ou indiretamente, no qual será focado o desenvolvimento desse estudo.

Sabe-se que a criação de um *software* consiste em um aprofundamento amplo de importantes áreas da Ciência da Computação, as quais serão abordadas durante o decorrer desse trabalho. Inclusive, com todo o rápido avanço na tecnologia de informação atualmente, podemos dizer que um *software*, ou seu ciclo de vida, percorre um longo caminho desde o projeto do seu desenvolvimento e não terminam simplesmente no momento que são entregues ao cliente, necessitando de um suporte e de uma manutenção, as quais serão inseridas atualizações, solicitadas ou não, analisadas e testadas posteriormente.

Metodologia Científica

Para um maior aprofundamento desse estudo, além de procurarmos nos basear em pesquisas bibliográficas, artigos em jornais e revistas, sites de internet, dentre outros, onde coletamos as diversas fontes da informação que desejamos transmitir nas áreas que

influenciam o bom desenvolvimento de um *software*, tivemos a oportunidade de conhecer e analisar o funcionamento de uma clínica médica que utiliza tecnologias computacionais voltadas para o segmento da saúde, possibilitando assim que exploremos diretamente toda a sistemática de atendimento ao paciente e todo o processo que está associado a estas funcionalidades, que podem ser particularidades da medicina em alguns casos, mas também comum a qualquer ramo de negócios.

Objetivos

O objetivo deste trabalho consiste na análise de gestão de informática em clínicas médicas especializadas, buscando demonstrar como todo o processo desse tipo de atividade é estudado, desenvolvido e principalmente seu funcionamento no âmbito computacional, podendo de alguma maneira verificar novas idéias e ações a serem introduzidas ou aprimoradas dentro da área de saúde através da Ciência da Computação, a fim de garantir atendimento de qualidade aos pacientes, reduzir custos operacionais e adaptar-se as pressões e mudanças de mecanismos reguladores e operadoras de saúde, que tem tornado esse área cada vez mais complexa e dinâmica.

Para isso, abordaremos todas as etapas do processo de administração desse ramo de atividades através das suas ferramentas tecnológicas, que são fatores essenciais para uma segurança e maior racionalidade no tratamento aos pacientes, pois facilitam e reduzem de maneira expressiva o trabalho e tempo empregados, contribuindo assim para uma melhoria considerável na qualidade do funcionamento e atendimento dessa estrutura organizacional, além de garantir a segurança dos dados cadastrais, estatísticos e históricos.

Justificativa e Motivação

A escolha do tema “Ambiente Computacional de Gestão em uma Clínica Médica” para desenvolvimento dessa monografia surgiu pelo motivo de estar contribuindo de alguma maneira com o alinhamento do ramo de atividades profissional na administração de uma clínica na área de saúde, exercido por este autor, com o curso de graduação em Ciência da

Computação, que possibilitou demonstrar através da teoria e da prática das disciplinas estudadas ao longo do curso, como todo o avanço da informática nos últimos anos foi e continua sendo útil e de fundamental importância para facilitar o trabalho humano, proporcionando um atendimento diferenciado no setor da saúde e possibilitando uma organização clínica, administrativa e financeira de todo o processo operacional.

Organização do Trabalho

Este trabalho está dividido em cinco capítulos práticos e teóricos da gestão de informática em uma clínica médica especializada em oncologia.

No primeiro capítulo “Caracterização do Ambiente Estudado”, será realizada uma abordagem prática acerca do histórico da indispensável necessidade de implantação de um sistema de gestão de clínica médica no ambiente estudado, aspectos funcionais e profissionais, instalações físicas, a aparelhagem tecnológica e estrutura computacional de *hardware* e *software*.

No segundo capítulo “O Sistema Computacional”, será comentado um breve histórico da aquisição do *software* médico pela clínica, descreveremos a características do sistema e o seu processo de execução das atividades decorrentes do tratamento ao paciente. Serão descritos procedimentos computacionais essenciais para que haja conservação na qualidade do sistema, tais como manutenções através de atualizações preventivas e/ou corretivas, suporte técnico, criação de rotinas de *backups* periódicos e mecanismos de segurança das informações.

No terceiro capítulo, “Ferramentas Computacionais para Desenvolvimento de *Software*”, serão caracterizados e analisados dois ambientes de desenvolvimento amplamente conhecidos na área da Ciência da Computação, o *Java* e o *Delphi*, ferramentas de elevado poder computacional e de fundamental importância na história das linguagens de programação. O *Delphi* é o ambiente utilizado atualmente no sistema da clínica analisada e tem suprido as necessidades, porém decidimos também abordar o *Java* devido ao crescente avanço desse ambiente de desenvolvimento em diversos aspectos que atualmente o favorecem

em relação aos demais, além da migração contínua de programadores para a constante evolução do aprendizado e obtenção de certificações dessa linguagem, a fim de competirem de acordo com as necessidades do mercado.

No quarto capítulo “Sistema de Gerenciamento de Banco de Dados”, será abordada a importância da utilização de uma ferramenta de banco de dados capaz de gerenciar e garantir a segurança física e consistência das informações inseridas no *software* médico. Também será descrita algumas ferramentas de bancos de dados, como o Paradox, no qual se deu o início do sistema da clínica, o SQL Server (*Structured Query Language Server*), no qual as informações foram migradas no decorrer desse estudo e o XML (*Extensible Markup Language*), modelagem de dados utilizada atualmente pra troca de informações entre órgãos, operadoras e prestadores de saúde.

No quinto capítulo “Engenharia de *Software*”, será abordada a necessidade constante da melhoria na qualidade do processo de desenvolvimento de *software* designados para o mercado nos diversos setores da economia, com o aprimoramento de técnicas e estágios para focados em um trabalho em equipe, iniciando-se com a documentação do levantamento dos requisitos pelo cliente até a sua entrega final, sempre viabilizando posteriormente possibilidades de manutenções em sua operacionalização.

Por fim, serão feitas as considerações finais acerca do que foi analisado pelo estudo de revisão bibliográfica desse trabalho e da prática em campo. Além disso, iremos sugerir recomendações futuras a serem introduzidas, visando à melhoria da qualidade do atendimento e da administração financeira e gerencial das atividades desenvolvidas na clínica, através de soluções computacionais.

Capítulo 1 – Caracterização do Ambiente Estudado

1.1 – Cenário do Ambiente Interno e Externo

Para melhor entendermos o combate ao câncer, podemos primeiramente defini-lo de maneira simples e clara a uma pessoa não relacionada à área de saúde, segundo o Instituto Nacional do Câncer (INCA), como um conjunto de doenças que têm em comum um crescimento desordenado (maligno) de células que rapidamente e de maneira muito agressiva invadem os tecidos e órgãos, podendo espalhar-se (metástase) para outras regiões do corpo e suas causas são variadas, podendo estar relacionada a hábitos alimentares, físicos, relacionadas ao meio ambiente, de ordem genética, costumes sociais, como o tabagismo, dentre outros aspectos que podem influenciar diretamente ou indiretamente na capacidade de defesa do organismo.

Na década de 30, ainda segundo pesquisa feita no site do Instituto Nacional do Câncer (INCA), observou-se um grande aumento da mortalidade de câncer. O então presidente da república Getúlio Vargas decretou como medida de combate a esse aumento gradativo, a instalação de um Centro de Cancerologia na cidade do Rio de Janeiro, onde poderia ser tratada e pesquisada essa doença que estava em fases iniciais de tratamento no Brasil, enfrentando grandes problemas de instalações. Após três anos, esse Centro de Cancerologia transformou-se no Instituto Câncer, e posteriormente, foi reconhecido oficialmente como Instituto Nacional do Câncer, dessa maneira atribuindo-lhe novas competências assistenciais, científicas e educacionais, além de ser integrá-lo diretamente ao Ministério da Saúde.

Em 1967, foi criada a I Campanha Nacional de Combate ao Câncer, com o objetivo de organizar de maneira administrativa e financeira o controle do câncer no país, e desde então, várias políticas de controle do câncer foram realizadas através de reformas, políticas públicas e convênios técnico-científicos firmados, que em pouco espaço de tempo duplicou a prestação de serviços médicos na especialidade. Logo, foi tornando-se possível a criação e aperfeiçoamento dos primeiros registros de câncer no país, possibilitando direcionar eficientemente a prevenção de cânceres, combate ao tabagismo e inclusão do estudo da Cancerologia nos cursos de graduação em Ciências da Saúde, que passou a receber uma atenção especial na medicina, trazendo melhorias freqüentes nos diagnósticos da doença.

No início da década de 90, foi criado o Sistema Único de Saúde (SUS), que possibilitou um novo impulso no combate ao câncer no país e um crescimento significativo de instalação de novos centros de oncologia pelo país, buscando garantir futuramente uma assistência oncológica de melhor qualidade à população brasileira que não vivia nas regiões metropolitanas, e conseqüentemente a isso, clínicas particulares especializadas foram surgindo com maior frequência, permitindo uma elevada expansão da oferta de serviços de diagnósticos, cirúrgicos, quimioterápicos, radioterápicos e cuidados paliativos.

Nos últimos anos, o combate ao câncer tem se tornado cada vez mais abrangente, com um tratamento praticamente uniforme em todo o mundo, pois novos protocolos de tratamento com medicações mais eficientes e menos agressivas ao paciente são constantemente estudados por especialistas da área, através de pesquisas em várias fases de evolução, incentivados financeiramente pela indústria farmacêutica, que os comercializa posteriormente à aprovação pelos órgãos fiscalizadores de cada país.

Além da existência de diversas sociedades médicas voltadas para a área em todo o mundo, são realizados anualmente grandes congressos de oncologia, que reúnem milhares de profissionais de saúde envolvidos na especialidade, possibilitando uma maior globalização do conhecimento, através da colaboração científica de renomados oncologistas de todo o mundo, assim como através de lançamentos de novas medicações e protocolos mais eficientes.

Dentro desse cenário, surgiu em 1994, a Quimioclinic, clínica que atua há 16 anos na área de saúde e seu atendimento é especializado na área da medicina denominada Oncologia Clínica, ou seja, direcionado principalmente ao atendimento a pacientes portadores de diversos tipos de câncer, através de consultas médicas com especialistas e tratamentos de quimioterapia e hormonioterapia de alta complexidade. Atualmente, com a experiência adquirida na administração de um negócio de tal porte e em constante avanço, entendemos que seria praticamente inviável gerir um serviço de qualidade se não tivéssemos o apoio computacional envolvido nesse processo.

1.2 – Surgimento do Ambiente Computacional

Há aproximadamente oito anos, o atendimento e controle de funcionamento da clínica “passou” pela transformação necessária e indispensável, que foi optar pelo uso de *software* voltados para a área da saúde, ou seja, ferramentas computacionais que passaram a facilitar o

trabalho interno desenvolvido, que se tornava cada vez mais crescente nesse tipo de estrutura organizacional. Obviamente, foi necessária também uma atualização (*upgrade*) na aquisição de melhoria de *hardware* para que isto fosse possível, pois sistemas robustos necessitam de maior poder de processamento, memória, dentre outras características.

Naquela época, todo o controle de atendimento e financeiro era feito em agendas anuais de papel e arquivos de planilhas e textos. A clínica não possuía um controle efetivo dos agendamentos de consultas e procedimentos médicos, havia dificuldade na obtenção e localização de dados do paciente (apenas prontuário físico), problemas de armazenamentos incorretos e comprometimento do sigilo das informações.

Com a evolução da informática e o aparecimento de bons software médicos, desejou-se criar um sistema personalizado para a própria clínica, porém devido às singularidades necessárias para tal desenvolvimento e as empresas orçadas naquele momento estarem começando a aperfeiçoar a maneira de como produzir *software* de qualidades voltados para a área de saúde, os longos prazos e os custos de desenvolvimento e manutenção seriam muito elevados.

Por isso, ficaria inviável naquele momento, optando-se pela utilização de um produto já desenvolvido (*off-the-shelf*), que estava sendo comercializado para outros clientes e que poderiam ser customizados para a clínica.

1.3 - Organograma Funcional

A equipe médica é formada por especialistas que compõem o corpo clínico nas áreas de oncologia clínica, hematologia, mastologia, radioterapia, enfermagem oncológica e psico-oncologia. O quadro de funcionários administrativos é composto de quinze pessoas que englobam também cargos nos setores de administração, contas médicas, faturamento, controle financeiro (contas a pagar e receber), computação, recepção, serviço de copa, limpeza e motorista.

1.4 – Instalações Físicas

A sua estrutura física tem aproximadamente 250m² de área construída, no qual possui recepção climatizada, consultórios médicos, ampla sala para tratamento quimioterápico equipada com bombas de infusão modernas, apartamentos para tratamento individualizado, serviço de UTI móvel, posto de coleta de exames, emergência, capela de fluxo laminar (local de preparo de medicações) com sala de estoque de medicamentos e materiais, setores subdivididos de administração, contas médicas, além da área externa no qual é disponibilizado um estacionamento privativo.

1.5 - Estrutura Computacional: *Hardware* e *Software*

A estrutura computacional em uma clínica médica especializada, que busca uma gestão moderna de atendimento, vem se tornando indispensável e com o surgimento constante de novas tecnologias de *hardware* e *software*, o funcionamento das atividades cotidianas ocorre de maneira cada vez mais organizada, auxiliando os funcionários e proporcionando ao paciente uma maior comodidade e segurança durante todo o período de seu tratamento.

1.5.1 - *Hardware*

Um computador, através de programas, é capaz de fazer cálculos e tomar decisões lógicas que podem se estender por altos níveis de complexidade e velocidades, no qual o ser humano poderia levar elevados períodos de tempo para concretizar.

Os vários dispositivos físicos que compõem um computador são chamados de *hardware*, tais como a memória principal (*RAM*, do inglês *Random Access Memory*), processador, disco rígido (*HD*, do inglês *Hard Disk*), monitor, unidades de CD/DVD/Cartões de Memória, teclado, mouse, dentre outros periféricos que têm surgido a partir da constante evolução da informática.

A estrutura computacional em termos de *hardware* do ambiente estudado possui um servidor de dados, nove computadores com configuração entre 250Gb e 500Gb de *Hard Disk(HD)*, 2Gb de memória *RAM*, Processadores Intel *Pentium Dual CPU E2180 2.0GHz*, interligados por uma rede privada através de um *switch/hub* e internet *wireless*.

1.5.2 - *Software*

Os dispositivos de *hardware*, citados anteriormente, necessitam de programas para execução das rotinas realizadas pelo computador, chamados *software*, que influem de maneira relevante sobre a eficiência e qualidade de um ambiente computacional.

Atualmente, a aquisição de ferramentas de *hardware*, vem tornando-se mais acessível e o computador tornou-se um bem de consumo comum em diversas classes da sociedade, porém os custos com desenvolvimento de *software* são cada vez mais significativos à medida que são criados programas cada vez mais robustos e úteis a simples necessidades pessoais até complexas aplicações industriais e comerciais. Citando o exemplo para qual o nosso trabalho é direcionado, podemos afirmar que as ferramentas computacionais vêm tornando-se grandes aliadas da medicina para desenvolvimento de complexas investigações e pesquisas científicas, cirurgias mais eficientes e menos traumáticas, procedimentos especializados mais eficientes, fabricação de medicamentos focados na melhor qualidade de vida e na própria gestão administrativa nessa área, que é o foco principal desse estudo.

Em termos de *software*, a clínica possui sistemas voltados para área de saúde, que contribuirão de maneira fundamental para retratar a gestão de informática e do ambiente computacional em uma clínica médica e todo o apoio indispensável que é obtido para a realização das tomadas de decisões que esse tipo de empresa necessita para tornar o processo de operacionalização das atividades desenvolvidas mais produtivas, eficientes, organizadas e seguras, como veremos nos capítulos a seguir.

Capítulo 2 – O Sistema Computacional

2.1 – Introdução

Um sistema de gestão na área de saúde tem como principal objetivo a integração de diversos módulos, cujo objetivo maior é proporcionar um aumento da produtividade em todos os setores envolvidos na administração de um complexo hospitalar ou de uma clínica médica especializada. Com a alta competitividade que enfrentamos atualmente para se estabelecer em qualquer atividade profissional, principalmente na área médica, que avança tão rapidamente quanto a Ciência da Computação, esses sistemas de informações podem ser decisivos, aliado a outros fatores técnicos e humanos, sobre o destaque de uma empresa no mercado.

No caso desse estudo, procuramos direcionar nosso foco mais especificamente às clínicas médicas, pois além de termos uma maior experiência, já será suficiente para retratarmos estruturas hospitalares de maior escala. Um sistema, mais conhecido no âmbito computacional como *software*, precisa atender diversos requisitos e funcionalidades para englobar importantes necessidades de todos os seus potenciais usuários. Para isso, sua criação necessita ser muito bem conduzida em diversas áreas da informática, tais como a engenharia de *software*, a gerência do projeto, o sistema de gerenciamento de banco de dados, as ferramentas de desenvolvimento, dentre outras áreas envolvidas. Todos esses temas são de alta relevância para criação de aplicações que possam contribuir com a sociedade médica, por isso, iremos abordá-los através de revisões bibliográficas nos capítulos seguintes.

O controle de uma clínica médica especializada de grande porte e alta complexidade de tratamento, como no nosso caso a quimioterapia, necessita de uma capacidade de gerenciamento no qual é indispensável um apoio computacional capaz de processar informações e gerar resultados, para que todo o controle da clínica seja realizado de maneira correta e precisa, dando maior confiabilidade e previsibilidade de rotinas a serem executadas.

Ao ser iniciado, posteriormente ao *login* de entrada e a conseqüente verificação da validação do usuário de acordo com suas regras pré-determinadas de controle de acesso, liberando as funcionalidades estabelecidas, o menu principal do sistema aparecerá na tela inicial, que na linguagem de desenvolvimento *Delphi*, é comum chamarmos de formulário principal (*main*). Aliás, essa linguagem de desenvolvimento, se utilizada corretamente e com clareza, tem como característica principal uma simples navegação entre todas as demais telas do sistema, proporcionando ao usuário uma fácil interação com o sistema.

A tela inicial possui quatro módulos distintos (e atalhos visuais), conforme Figura 1, que conduzirão o usuário na prática das funcionalidades existentes do sistema, nas quais destacaremos as principais:

Quadro 1- Módulo "RECEPÇÃO"

1. Cadastros
 - 1.1. Prestadores de Serviços
 - 1.2. Pacientes
 - 1.3. Formas de Indicação do Paciente
 - 1.4. Tipos de Agendamento do Médico
 - 1.5. Tipos de Atendimento
 - 1.6. Impressos de Atendimento
 - 1.7. Feriados
 - 1.8. Ocupações
 - 1.9. Motivos de Atendimento
 - 1.10. Modelos de Impressos
 - 1.11. Recursos de Agenda
 - 1.12. Textos para Correspondência
 - 1.13. Modelos de Etiqueta
 - 1.14. Convênios
2. Rotinas
 - 2.1. Agenda do Médico
3. Consultas
 - 3.1. Seleção de Pacientes
 - 3.2. Seleção de Prestadores de Serviços
4. Relatórios
 - 4.1. Agenda do Médico
 - 4.2. Relação de Agendamentos do Médico
 - 4.3. Agendamentos por Indicação
 - 4.4. Agendamentos por Tipo
 - 4.5. Pacientes com Observação
 - 4.6. Pacientes Candidatos a Duplicidade

Quadro 2 - Módulo "ATENDIMENTO"

1. Cadastros
 - 1.1 Tipos de Consultas
 - 1.2 Modelos de Fichas de Pacientes
 - 1.3 Medicamentos
2. Rotinas
 - 2.1 Atendimento Clínico
3. Gráficos
 - 3.1 Estatística de Atendimento Clínico
4. Relatórios
 - 4.1. Relação de atendimentos Clínicos

Quadro 3 - Módulo "FATURAMENTO"

1. Cadastros
 - 1.1. Convênios
 - 1.2. Grupos de Faturamento
 - 1.3. Tipo de Conta do Atendimento
 - 1.4. Grupos de Produto
 - 1.5. Produtos
 - 1.6. Regras para Cobrança
 - 1.7. Tabela de Cobrança de Produtos
2. Rotinas
 - 2.1. Manutenção dos Lotes de Medicamentos
 - 2.2. Faturamento Simplificado
 - 2.3. Importação de Tabelas Brasíndice
 - 2.4. Atualização de Preço Brasíndice
3. Consultas
 - 3.1. Consulta de Produtos
 - 3.2. Consulta Tabela Brasíndice
 - 3.3. Consulta Contas de Atendimento
 - 3.4. Consulta Itens de Conta
4. Relatórios
 - 4.1. Preços de Tabela de Cobrança
 - 4.2. Preços do Convênio/Plano
 - 4.3. Contas Faturadas Por Paciente / Grupo de Pacientes
 - 4.4. Evolução Mensal de Faturamento
 - 4.5. Faturamento por Período de Competência
5. Gráficos
 - 5.1. Faturamento por Convênio
 - 5.2. Comparativo Anual de Faturamento

Quadro 4 - Módulo "CONFIGURAÇÕES"

1. Cadastros
 - 1.1. Estrutura Organizacional
 - 1.2. Perfis de Usuário
 - 1.3. Usuários
 - 1.4. Estações de Trabalho
 - 1.5. Históricos
 - 1.6. Países
 - 1.7. Unidades Federativas
 - 1.8. Cidades
 - 1.9. Caixas de Saída
2. Rotinas
 - 2.1. Unificar Prontuário
 - 2.2. Unificar Prestadores de Serviços
 - 2.3 Selecionar Prestadores de Serviços Candidatos a Duplicidade
3. Controles
 - 3.1. Configurações de Sistema
 - 3.2. Configurações de Sistema por Empresa/Filial
 - 3.3. Controle de Acesso
 - 3.4. Auditoria do Sistema
4. Ferramentas
 - 4.1. Mensagens e Alertas

2.4 – Funcionamento do Processo Resumido de Rotinas do Sistema

Todo o processo de integração começa no momento da marcação de consulta do paciente, no qual os dados iniciais são inseridos no sistema. No momento da consulta, são coletadas as demais informações para que seja efetivada com agilidade e segurança a ficha eletrônica do paciente para que o médico, ao atendê-lo, possa dar andamento ao seu tratamento. Após o término da consulta, e caso haja necessidade de um tratamento oncológico, na qual a clínica é especializada, o médico deverá encaminhar o paciente para que a equipe de enfermagem programe a disponibilidade dos apartamentos ou sala de infusão ambulatorial, local onde o paciente receberá o protocolo de medicações que fora designado pela prescrição médica inserida no prontuário eletrônico do sistema.

Nesse mesmo momento, o setor de faturamento já terá recebido a informação se o paciente terá tratamento através de convênios (operadoras de saúde) ou pagamento particular. A cobrança da consulta é gerada, e caso seja via convênio, será necessária uma autorização prévia para inicialização do tratamento quimioterápico, que independente disto, poderá ser

programada até para o mesmo dia ou data futura, conforme quadro clínico e conduta médica.

Caso seja particular, a fatura é gerada pelo sistema, e conseqüentemente, a nota fiscal e recibo são gerados para que seja realizado o crédito e inserção deste no faturamento, que enviará posteriormente ao setor contábil, para que seja feito os lançamentos fiscais e geração de impostos e relatórios.

Para isso, o setor financeiro, através do sistema, deverá verificar a disponibilidade de medicações e materiais, além de espaço físico e das bombas de infusões para aplicações quimioterápicas, para todos os tratamentos previstos em um período de tempo e reservá-los, de acordo com o recebimento das autorizações. Caso haja falta no estoque, será preciso gerar uma ficha de pedido de medicação aos fornecedores, geralmente um laboratório da indústria farmacêutica ou uma distribuidora local, nos quais a clínica possui cadastro, com todas as regras da compra, tais como, preços, descontos, parcelamentos, forma de pagamento, prazos de entrega etc.

No ato da chegada, deverá ser conferido no sistema, se o pedido de medicações enviadas corresponde em quantidade e dosagem, além da verificação da nota fiscal para constatar que foram obedecidas também todas as regras de compras. Após isso, a nota fiscal é inserida no sistema com todas as suas características, tal como nome do fornecedor, produto, número da nota fiscal, data da chegada, discriminação das parcelas nos seus referidos prazos e valores. Posteriormente, como forma mais comum de pagamento, são enviados via correio os boletos bancários correspondentes e nesse momento o usuário deverá confirmar no sistema a chegada de cada parcela destes e aguardar pelo vencimento do pagamento e sua quitação.

Após ter sido realizado o processo de autorização pelo convênio e a verificação da disponibilidade de recursos para o início do tratamento oncológico, o paciente deverá ser conduzido para a sala de infusão para procedimento ambulatorial. Nesse momento, o estoque, obedecendo à prescrição médica, será decrescido de todos os materiais e medicamentos utilizados para efetuar a elaboração da fatura do paciente de acordo com as regras de cada convênio.

A fatura gerada é inserida em outro sistema, que padroniza as guias de acordo com o padrão TISS (Troca de Informação em Saúde Suplementar), gerando o arquivo no formato de dados XML, que será estudado posteriormente, através de uma interface de aplicação para interagir com o usuário. Ao final de cada mês, as faturas dos pacientes de cada convênio são reunidas em cartas-remessas ou lotes, onde o arquivo gerado em XML será enviado através da

interface de comunicação do sistema da clínica com a do convênio ou operadora de saúde, como são comumente chamados. Por se tratar de medicação de alto custo, ainda assim, as guias em papel com autorizações assinadas pela auditoria dos convênios e pelo paciente são enviadas fisicamente.

2.5 - Fluxogramas de Descrição das Atividades

Um *software* é documentado de várias maneiras possíveis, como linguagens formais, máquinas de estado, dicionário de dados, orientação a objetos, fluxogramas de dados e rotinas, além de diversas outras técnicas. Portanto, iremos demonstrar como é traçado um fluxograma das atividades principais do ambiente estudado para se conhecer os passos procedimentais para orientação ao projeto de desenvolvimento de *software*, ou seja, é necessário definir quais as atividades a serem executadas ao longo do processo, assim como em qual momento quem e como as executará.

Para isso, é de fundamental importância que seja realizado um levantamento e análise dos requisitos da clínica ora estudada, com a participação de um potencial usuário, para que seja simulado um ciclo de vida do *software* a ser implementado ou para constatar se um sistema já desenvolvido, como no caso especificamente desse estudo, satisfaz as necessidades reais do cliente, possibilitando assim adaptações não previstas no momento da aquisição. A seguir, veremos alguns fluxogramas com possíveis cenários existentes na rotina da clínica.

2.5.1 – Marcação de Consultas

Quadro 5 - Fluxo Principal do Cenário “Marcação de Consultas”

1. O usuário recebe a ligação do paciente e informa a próxima data e horário de vaga disponível de atendimento no sistema.
2. O usuário solicita nome do paciente e verifica se o paciente já possui cadastro no sistema.
3. O usuário localiza e confere os dados cadastrais do paciente.
4. O paciente confirma os dados cadastrais e o usuário registra o paciente na agenda de consultas do médico responsável.

Quadro 6 - Fluxo Alternativo do Cenário “Marcação de Consultas”

1a. O paciente solicita outras datas e horários disponíveis e o usuário informa demais datas e horários disponíveis no sistema
1b. O paciente concorda com o horário.
1c. Voltar para item 2 do Fluxo Principal.
2a. O usuário informa que paciente não possui cadastro e solicita alguns dados para marcação da consulta.
2b. O paciente fornece alguns dados iniciais para marcação da consulta.
2c. Voltar para item 3.

2.5.2 – Consultas

Quadro 7 - Fluxo Principal do Cenário "Consultas"

1. O usuário registra chegada do paciente.
2. O usuário recebe informações iniciais do paciente e confere na agenda de consultas.
3. O usuário preenche dados cadastrais ausentes.
4. O usuário registra que o pagamento será através de convênio no sistema.
5. O usuário checa regularidade do paciente no convênio.
6. O sistema contabiliza tempo de espera.
7. O médico observa chegada do paciente no sistema e autoriza entrada do paciente ao usuário (recepção).
8. O paciente entra para a consulta e o usuário registra atendimento da consulta no sistema.
9. O médico atende e registra informações da consulta do paciente (sintomas, resultado de exames físicos e laboratoriais, diagnóstico, solicitação de novos exames).
10. O médico realiza indicação da conduta ou protocolo (na linguagem médica, refere-se a prescrição da medicação ou combinação de medicações quimioterápicas a serem realizadas no tratamento oncológico) de tratamento do paciente.
11. O médico agenda no sistema a data do retorno para realização do procedimento ambulatorial a ser executado.
12. O paciente retorna para a recepção e assina a guia de autorização do pagamento da consulta pelo convênio.
13. O usuário registra no sistema a inclusão do pagamento.
14. O sistema registra pagamento no módulo faturamento.

Quadro 8 - Fluxo Alternativo do Cenário “Consultas”

4a. O paciente não possui convênio
4b. O usuário registra que o pagamento será particular, ou seja, pago diretamente pelo paciente.
4c. Voltar para o item 5.
5a. O paciente contém alguma irregularidade no convênio.
5b. O usuário verifica com o paciente do que se trata a irregularidade. (Ex.: Carteira de beneficiário vencida, falta de pagamento, dentre outros)
5c. A situação é regularizada.
5d. Voltar para o item 7
8a. O médico verifica que o paciente não precisará de tratamento médico na clínica
8b. O médico encaminha o paciente para outro especialista.
8c. Voltar para 10.

2.5.3 – Cadastro de Pacientes

Quadro 9 - Fluxo Principal do Cenário “Cadastro de Pacientes”

1. O usuário acessa a lista de cadastro de pacientes.
2. O usuário verifica se o paciente é cadastrado.
3. O usuário preenche o formulário de cadastramentos com os dados pessoais fornecidos pelo paciente.
4. O usuário salva a informação.
5. O sistema valida e registra os dados
6. O usuário encerra a operação.

Quadro 10 - Fluxo Alternativo do Cenário "Cadastro de Pacientes"

1a. Caso o paciente seja cadastrado, voltar para item 6.
3a. O sistema acusa que CPF não é válido.
3b. Usuário reconfirma CPF do paciente e reinsere no sistema.
3c. Voltar para item 4.

2.5.4 – Procedimento Ambulatorial

Quadro 11 - Fluxo Principal do Cenário "Procedimento Ambulatorial"

1. O médico analisa exames médicos e os anota no sistema.
2. O médico prescreve protocolos de medicamentos.
3. O paciente entra na sala de infusão ou apartamentos para realização do tratamento e a equipe de enfermagem confirma no sistema a execução da prescrição médica.
4. O médico anota reações do paciente.
5. O médico anota as respostas do paciente a medicação ou protocolo de medicações
6. O médico receita medicações e marca retorno do paciente

Quadro 12 - Fluxo Alternativo do Cenário "Procedimento Ambulatorial"

1a. O médico verifica e descreve que exame do paciente o impossibilita de realizar o tratamento.
2a. O médico receita medicações de suporte e marca novos exames.
3a. Volta para item 2.

2.5.5 - Faturamento

Quadro 13 - Fluxo Principal do Cenário “Faturamento do Procedimento Ambulatorial”

1. O médico prescreve o protocolo de medicações no sistema.
2. O setor de faturamento lança dados das medicações e honorários do tratamento
3. O setor de contas médicas solicita autorização ao convênio.
4. O convênio autoriza medicação e o procedimento ambulatorial é realizado, deduzindo as medicações necessárias do estoque.
5. O paciente assina a guia médica e a recepção informa confirmação do tratamento pelo paciente no sistema
6. Ao fim do mês, o arquivo XML é gerado por paciente/convênio e enviado via online.
7. O convênio recebe arquivo XML e efetua pagamento dos tratamentos realizados pelos pacientes no prazo acordado.
8. O financeiro realiza a baixa do pagamento e faturamento é realizado com sucesso.

Quadro 14 - Fluxo Alternativo do Cenário “Faturamento do Procedimento Ambulatorial”

4a. O convênio não autoriza o procedimento
4b. O processo é interrompido até o paciente resolver situação com o convênio.
4c. Voltar para item 4.

2.6 – Suporte e Manutenção

A manutenção de um *software* de grande porte está diretamente relacionada à sua qualidade, pois com a crescente e rápida evolução de novas tecnologias de informação, constantemente são necessárias a realização de correções no sistema, para que a clínica, como nosso exemplo, possa cumprir os processos de gestão da área de saúde competitivamente com o que está sendo oferecido na atualidade. No capítulo 5, mais informações sobre este tópico serão abordadas.

2.7 – Rotinas de Backups e Segurança

A segurança de um *software* relaciona-se também de maneira direta com a qualidade deste, principalmente no momento que vivenciamos atualmente, no qual a prática da informática aliada à *Internet* amplamente inserida ao acesso de grande parte da população mundial, muitas vezes somos alvos de indivíduos com intenções prejudiciais gratuitas ou fraudulentas, gerando diariamente grandes ameaças à ambientes computacionais, como por exemplo, invasões de privacidade, destruição de dados, dentre outras ações que comprometem a integridade de um *software*.

Para isso, um *software* na área médica que lida com informações pessoais referentes à saúde de pacientes, históricos de tratamentos e exames realizados, dentre outros aspectos administrativos e financeiros da gestão de uma clínica que podem ser afetados, é fundamental que a capacidade de sua proteção e segurança seja testada exaustivamente antes de ser entregue ao cliente e também resguardada no tempo que o cliente estiver utilizando o *software*, pois as ameaças são diariamente atualizadas e aperfeiçoadas por esses intrusos mal intencionados. Além disso, uma ferramenta de *hardware* ou *software* está frequentemente vulnerável a problemas ou falhas diversas que podem ocorrer inesperadamente, ocasionando perdas de dados, que poderiam prejudicar de maneira drástica o funcionamento e integridade da clínica perante seus clientes, funcionários e gestores.

No caso da clínica, são realizados como medidas preventivas *backups* semanais, programados automaticamente pelo sistema, para outras estações da rede privada e manualmente uma vez por quinzena para um disco rígido externo. Além disso, os

computadores pessoais dos funcionários estão em um nível adequado de segurança contra ameaças existentes, procurando manter sempre atualizados as ferramentas de anti-vírus, pelo setor responsável de manutenção geral de informática da clínica.

2.8 – Modelos de Relatórios (Informações geradas e saídas)

Os relatórios são ferramentas que geram saídas impressas, na tela ou incorporadas a um documento, de um resumo legível ao usuário através de textos formatados, gráficos ou tabelas personalizadas às preferências do cliente. O sistema médico analisado permite a criação de importantes funcionalidades para as tomadas de decisões na clínica estudada.

Data	Hora	Prontuário	Nome do paciente	Fone	Tipo agend	Convênio	Status
Médico:							
01/04/2010	08:30	00002944	ANA MARIA	(85) 3257	QUIMIO SUB	GEAP	Atendido
01/04/2010	08:30	00002181	MARIA SELMA	(85) 3290	FORMONIO I	IPM	Atendido
01/04/2010	09:00	00002782	MARGARIDA	(85) 3261	F.CRESC HOM	FUNCEF	Atendido
01/04/2010	10:00	00002782	MARGARIDA	(85) 3261	FLUIDOTERAP	FUNCEF	Atendido
05/04/2010	09:00	00002734	BENEDITA	(85) 3267	QUIMIO SUB	CASSI	Atendido
05/04/2010	09:30	00001918	FRANCISCO	(85) 3248	MOstrar EX	UNINED CE	Atendido
05/04/2010	10:00	00003542	JOSE	(05) 3260	FORMONIO +	PETRODRÁS	Atendido
05/04/2010	10:00	00003752	EDVIGES	(85) 3289	QUIMIO SUB	GEAP	Atendido
05/04/2010	15:02	00002145	MARIA	(85) 3267	MANUTENCAO	CAMED NATURAL	Atendido
06/04/2010	11:00	00002061	DIANA	(88) 5969	QUIMIO SUB	FUNCEF	Atendido
06/04/2010	14:30	00039004	ELIEZER	(85) 5946	CONSULTA IN	CAMED NATURAL	Atendido
06/04/2010	15:00	00001911	MARIA	(85) 3454	MOstrar EX	GEAP	Atendido
06/04/2010	15:30	00000162	ALBA	(85) 3257	MOstrar EX	ASSEF	Atendido
06/04/2010	16:00	00039012	JOAO	(85) 3256	CONSULTA IN	GEAP	Atendido
06/04/2010	16:02	00002335	RAIMUNDO	(85) 3243	FORMONIOTE	GEAP	Atendido
07/04/2010	09:05	00001921	MARIA	(85) 3247	QUIMIO SUB	IPM	Atendido
07/04/2010	09:30	00002196	JOSE	(85) 3246	CONSULTA	GEAP	Atendido
07/04/2010	10:00	00002893	RITA	(85) 3265	ZOMETA	ASSEF	Atendido

Figura 2 - Modelo de Relatório Gerado pelo Sistema

2.9 – Interface Visual do Sistema

A interface do sistema analisado que foi adquirido pela clínica era muito amigável ao usuário na época de seu desenvolvimento, uma característica marcante da linguagem *Delphi*, ou seja, *Object Pascal* (orientada a objetos).

A facilidade de criação de elementos visuais dessa linguagem ocasionou em uma grande tendência de vários programadores optarem por essa linguagem, pois eram capazes de desenvolverem sistemas de maneira bastante produtiva, utilizando componentes gráficos facilmente inseridos ao invés de longas linhas de código. Descreveremos no capítulo seguinte um maior aprofundamento sobre o conhecimento do *Delphi*.

No decorrer desse estudo, ainda nesse ano de 2010, essa interface foi atualizada para um padrão voltado para a idéia de navegação na internet, conforme Figura 1 (pág. 20), melhorando a sua usabilidade, onde as opções da tela principal possuem uma fácil e prática visualização, de acordo com sistemas atuais.

Para se determinar a usabilidade de um *software*, ou seja, a sua interface, será necessário identificar se o usuário facilmente terá habilidades para operá-lo ou se será necessário desperdiçar longos períodos de tempo com treinamentos em cada departamento funcional da clínica, como também questionar ao cliente, se após sua instalação, a clínica teve aumento de produtividade sobre o processo utilizado anteriormente e em quais fatores ou setores foram identificados e obtidos esses ganhos.

Anteriormente a aquisição do *software*, todas as marcações eram realizadas em uma agenda médica (papel) onde a recepcionista anotava dados da consulta ou do tratamento e lançavam em uma planilha os dados financeiros, ou seja, esse trabalho manual era um processo rigorosamente lento e desorganizado, além da falta de integridade e segurança das informações. A Tabela 1 foi gerada através de simulação de ambos os tipos de realização de atividades desenvolvidas, pois dependem de outros fatores de *hardware*, *software* e sistema operacional, que podem influir na conclusão das atividades, mas podemos ver claramente que em relação à produtividade, os ganhos obtidos foram em larga escala.

Tabela 1 - Comparativo de tempo entre as atividades realizadas (valores aproximados)

<u>ATIVIDADE / REALIZAÇÃO</u>	<u>SISTEMA</u>	<u>MANUAL</u>
CADASTRO DE PACIENTE	1 minuto	4 minutos
AGENDAMENTO DE CONSULTA	45 segundos	3 minutos
GERAÇÃO DE RELATÓRIOS (Pacientes x Médicos)	30 segundos	15 minutos

Capítulo 3 – Ferramentas Computacionais para Desenvolvimento de *Software*

3.1 – Introdução e Classificação das Linguagens de Computação

Com o forte crescimento da área de desenvolvimento de *software*, atualmente existem diversas linguagens de computador sendo utilizadas com a finalidade de desempenhar as mais diversas funções.

Segundo Deitel (2005), uma linguagem de máquina é a linguagem natural de um computador na qual ele pode entender e esta consiste em *strings* (cadeias) de números 0 e 1, que instruem os computadores a realizar suas operações mais elementares, mas ao mesmo tempo bastante complexas ao ser humano. Buscando tornar a linguagem mais didática ao nosso entendimento, os programadores começaram a utilizar abreviações em inglês para representar essas operações, dando surgimento à base da linguagem *Assembly*, e desse modo o uso de computadores tornou-se mais comum com novas possibilidades de criação de ferramentas legíveis, embora ainda fossem necessárias muitas instruções para realização de simples tarefas.

Quadro 15- Exemplo de Código-Fonte em Assembly

```
POS    DS.W    1
; Busca 0 na sequencia de inteiros
SRCH0  MOVEA.L #DATUM,A0 ; (DATUM) definido alhures
        MOVE.L #DATUM,D0 ; guarda inicio
        CLR.W  D1
LOOP   CMP.W  (A0)+,D1
        BNE    LOOP
        SUB.L  A0,D0
        MOVE.W D0,POS
        RTS
        END
```

3.2 – Linguagens de Alto Nível

Com o intuito de reduzir essa quantidade de instruções de um simples código até os mais extensos, foram desenvolvidas a partir daí, as linguagens de alto nível, em que uma simples instrução seria capaz de realizar tarefas substanciais, tais como as linguagens *Delphi*, que é utilizada atualmente no sistema de gestão da clínica analisada, na qual tivemos a oportunidade de agregar ao nosso estudo, e a linguagem *Java*, que fará parte de nosso estudo, pois atualmente é mais amplamente utilizada por empresas de tecnologia da informação (TI) e programadores, fazendo-se seu uso necessário para que o sistema de gestão médica possa acompanhar as tendências globais de eficiência e qualidade. As diretrizes gerais que compõem uma linguagem são muitas vezes comuns a todas as outras, portanto, aplicáveis a qualquer outra implementação.

3.2.1 – Linguagem *Delphi*

O *Delphi*, como é mais conhecido, é um compilador para desenvolvimento de *software*, comercializado pela *Borland Software Corporation* a partir de 1995, originado a partir da linguagem Pascal, porém sua programação orientada a objetos tornou-se uma evolução desta primeira, na qual passou a denominar-se *Object Pascal*.

Seu ambiente de desenvolvimento integrado (IDE, do inglês *Integrated Development Environment*) reúne ferramentas de apoio ao desenvolvimento de *software* com o objetivo de tornar a programação mais interativa e ágil para os programadores, pois permite que códigos sejam criados e testados com muita facilidade, transformando, a partir de seus componentes e início de códigos gerados, protótipos em aplicações robustas que satisfaçam as necessidades dos clientes. Sua utilização é largamente utilizada em aplicações do tipo *desktop* e cliente/servidor, dependendo do tipo de edição que se deseje realizar.

O *Delphi* possui em seu ambiente de desenvolvimento, diversos elementos essenciais para tornar a programação mais produtiva. Logo em sua aparência inicial, são apresentados os principais, como a função *Project Manager*, que oferece uma visualização geral de todos os formulários, apresentando um conveniente mecanismo para gerenciamento do projeto de desenvolvimento, como veremos na Figura 3.

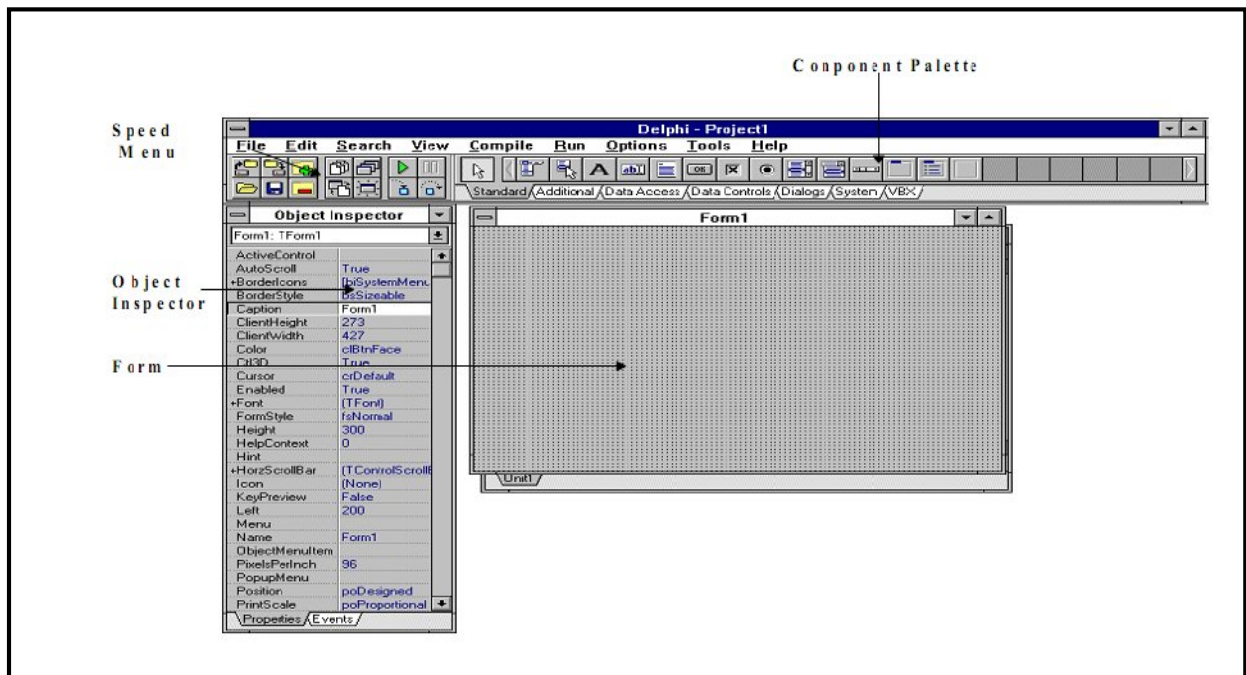


Figura 3 - Pannel de Configuração do Ambiente de Desenvolvimento *Delphi*

A linguagem de programação *Delphi* tornou-se muito aceita principalmente devido ao fato de estabelecer um ambiente de desenvolvimento de fácil uso para aplicações do tipo Cliente/Servidor, possuir uma grande biblioteca de componentes visuais (*VCL – Visual Component Library*) disponíveis que consiste na reutilização de objetos e boa interface com o usuário, além da alta performance no gerenciamento de dados e gráficos. A sua maneira de estruturação de códigos deu início a um novo conceito de programação, que atualmente é muito utilizada, chamada de orientação a Objetos (OO), onde os vários atributos e métodos estão contidos em classes, Segundo Pflieger (2004), podemos definir orientação a objetos como uma abordagem de desenvolvimento de *software* que organiza os problemas e suas soluções em um conjunto de objetos distintos, onde as estruturas e comportamento dos dados estão incluídos na representação de características particulares. Todas essas, reunidas ou isoladamente, possibilitam uma maior velocidade, além de proporcionar uma maior organização e facilidades no desenvolvimento de aplicativos.

Quadro 16 - Estrutura de Código-Fonte em *Delphi*

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

type
  TForm1 = class(TForm) |
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

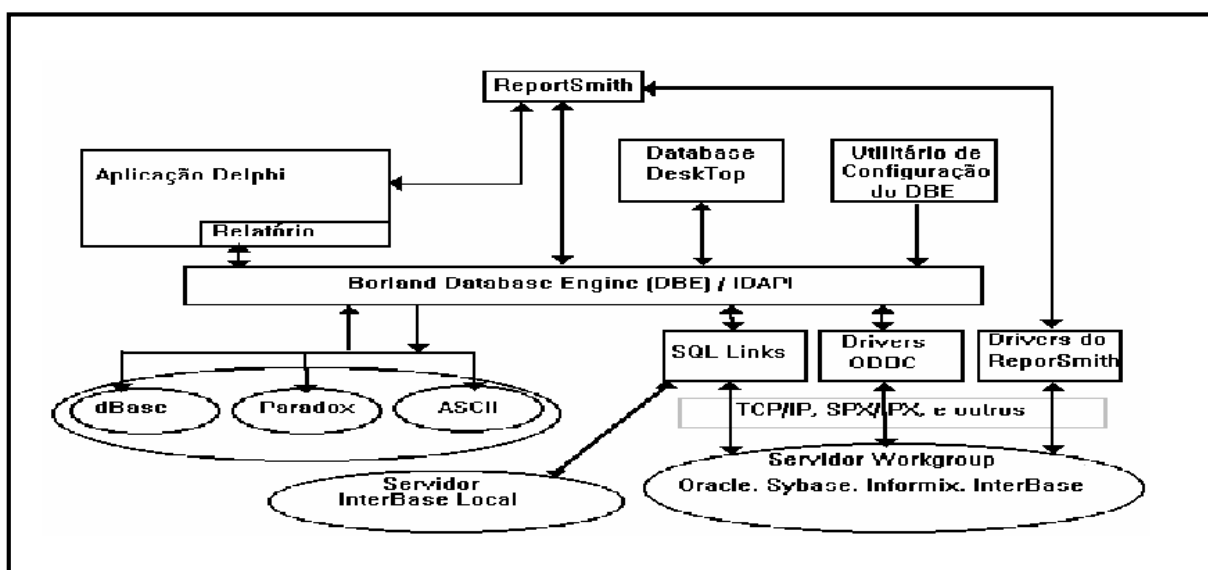
var
  Form1: TForm1;

implementation

{$R *.DFM}

```

Em uma aplicação de banco de dados em *Delphi*, os componentes se conectam com o *Borland Database Engine (BDE)*, que por sua vez, comunica-se e é responsável pelo controle com outras fontes de dados, tais como o Paradox, Interbase, Oracle, dentre outros.

Figura 4 - Comunicação *Delphi* com Banco de Dados.

3.2.2 – Linguagem *Java*

A Linguagem *Java* ou simplesmente o *Java*, como é mais conhecido, originou-se da linguagem C++ (Linguagem C orientada a objetos), no início da década de 90, através de um financiamento feito pela *Sun Microsystems* para um projeto de pesquisa corporativa interna, com o codinome Green, que após passar por grandes dificuldades até a crescente popularidade da *World Wide Web (www)* em 1993, onde o *Java* foi ganhando seu espaço. Em 2005, o *Java* foi anunciado formalmente pela primeira vez, já atraindo diversas oportunidades de negócios e obtendo seu devido e merecido sucesso, pois sua principal característica no momento proporcionou um conteúdo dinâmico as páginas da *WEB* com maior interatividade com o usuário.

Um programa *Java* é composto por classes e objetos, que contém atributos (características) e métodos (tarefas) bem definidos que interagem entre si com a finalidade de gerar aplicações de simples uso ou grandes e complexas aplicações comerciais, como no contexto desse estudo, um *software* voltado para a gestão na área de saúde.

Com o passar do tempo, foram agregadas e aprimoradas características e qualidades particulares do *Java*, que aumentaram ainda mais o seu valor em relação ao uso constante de orientação a objetos, tais como a identidade, abstração, classificação, encapsulamento, herança, polimorfismo e persistência (Deitel, 2005). Com estas e outras vantagens, ela se torna um linguagem de programação diferenciada capaz de possibilitar o desenvolvimento dos mais variados programas para corporações de diversos ramos da sociedade, dentre elas, a medicina e suas clínicas médicas especializadas, que tiveram seu crescimento cada vez mais alavancado com sistemas desenvolvidos por essa tecnologia que produziu, e promete produzir ainda mais, aplicações de grande porte para contribuir com o futuro da saúde e da administração de seus recursos.

Destacaremos dentre estes benefícios da orientação a objetos, a Herança e o Polimorfismo muito utilizados pela estrutura dos códigos-fonte nessa linguagem.

- **Herança** é um mecanismo que permite que uma classe seja construída a partir de uma outra já existente, através de adaptações realizadas na primeira (superclasse) para especificação da segunda (classe).

Quadro 17 - Exemplo de Código-Fonte utilizando Herança em Java

```

class SuperClasse{
public void nome(){
System.out.println("SuperClasse"); }
}
class SubClasse extends SuperClasse{
public int novo;
public void nome(){
System.out.println("SubClasse");
}
public void novometodo(){
System.out.println("Novo metodo"+novo); }
}
class Principal{
public static void main(String args[]){
SubClasse esub1=new SubClasse();
SuperClasse esuper1=new SuperClasse();
esub1.nome();
esub1.novo=10;
esuper1.nome();
esuper1=esub1;
esuper1.nome();
}}

```

- **Polimorfismo** é capacidade de se criar métodos com o mesmo nome dentro de variadas classes, através de um parâmetro na chamada do método.

Quadro 18 - Exemplo de Código-Fonte utilizando Polimorfismo em Java

```

class Polimo { //Inicio da classe
public int novo;
public void nome(){ //inicio método nome
System.out.println("SubClasse");
} //fim método nome
public void nome(String n){ //inicio método nome(n)
System.out.println("Novo metodo"+n);
} //fim método nome
} //fim classe

```


O *Java* tornou-se a linguagem preferida para implementação de aplicativos de comunicação em rede de computadores e tornou a comunicação entre usuário e internet mais dinâmica e simples. O seu ambiente de desenvolvimento passa por cinco fases: edição, compilação, carga, verificação e execução. Veremos cada uma a seguir:

- A fase de **edição** consiste na elaboração de um programa *Java* denominado código-fonte com um arquivo de extensão *.Java*.
- A fase de **compilação** consiste na conversão do código-fonte criado em um código intermediário denominado *bytecodes*, que pode ser interpretado pela *Java Virtual Machine (JVM)*, um aplicativo que simula um computador internamente dentro de um real, fazendo com que programas *Java* sejam independentes de plataformas de sistemas operacionais, uma característica marcante e muito importante para que o *Java* ganhasse mais uma vantagem em relação a outras linguagens de programação e pudesse dominar esse mercado. Para melhor exemplificar esse conceito, podemos citar o exemplo da clínica. Caso desejássemos migrar da linguagem de programação de *Delphi* para *Java*, ao contrário do que poderia ser (caso necessitássemos fazer uma migração de *Java* para *Delphi*), não precisaríamos fazer modificações no sistema operacional e outras severas atualizações aos computadores da clínica, podendo desta maneira ocasionar outros conflitos diversos, tal como problemas com outros programas já utilizados anteriormente a essa alteração.
- A fase de **carregamento** é exatamente quando o programa é alocado na memória para ser executado posteriormente. O arquivos *.class* são transferidos para a memória principal contendo seus *bytecodes* das fases anteriores.
- A fase de **verificação** valida os *bytecodes* dos arquivos carregados, examinando as restrições de segurança do *Java*.
- A fase de **execução** acontece finalmente quando a JVM¹ executa os *bytecodes* que foram aceitos através de um compilador *HotSpot*, que tem a função de traduzi-los para a linguagem de máquina.

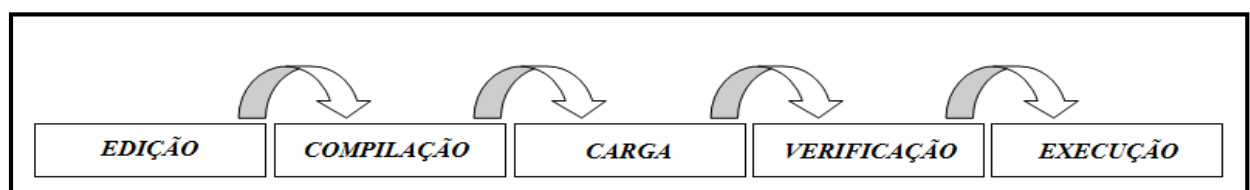


Figura 5 – Ambiente de Desenvolvimento em *Java*

3.2.3 – Comparativos – Vantagens e Desvantagens: *Delphi* x *Java*

- Vantagens do *Delphi*:

- A principal vantagem do ambiente de desenvolvimento *Delphi* é a facilidade que o programador tem em aprender e criar programas, pois seu ambiente de desenvolvimento integrado (*IDE*, do inglês *Integrated Development Environment*) possui variados recursos prontos para que sejam feitos os ajustes de acordo com as necessidades no momento, tornando o desenvolvimento mais rápido.

- A interface de programação possui diversas ferramentas visuais que possibilitam que um painel seja inserido ou atualizado na aplicação em poucos cliques do botão do mouse.

- Desvantagens do *Delphi*:

- A dificuldade de conexão com importantes e atuais bancos de dados como o *MySQL* em alguns casos, sendo necessário a instalação de um outro aplicativo para que isso seja realizado.

- Uso mais restrito a aplicações do tipo *desktop* pelo usuário, limitando o uso do sistema em um ambiente externo ao da rede privada da clínica.

- Maior custo com licenças pagas, pois dificilmente são encontrados ambientes gratuitos para desenvolvimento de programação.

- Maior direcionamento para a plataforma Windows, embora posteriormente fosse possível o desenvolvimento de aplicações em Linux, através do Kylix, que passou a ser a denominação do *Delphi* direcionado para esse sistema operacional, pela mesma fabricante do *Delphi*. Porém, o Kylix não se fortificou de maneira sólida no mercado como ferramenta de desenvolvimento suficiente para a plataforma Linux, pois possuía algumas diferenças em relação ao *Delphi* para Windows, tais como desvantagens na sua programação visual, conectividade com banco de dados e orientação a eventos.

- Vantagens do *Java*:

- Um *software* implementado em *Java* permite que suas aplicações sejam executadas em qualquer plataforma de desenvolvimento por tratar-se de uma linguagem interpretada, ou seja, independe de qualquer configuração interna do computador, como por exemplo, que o

sistema operacional utilizado seja Linux ou Windows é indiferente, fornecendo uma aceitação maior do mercado devido essa característica que pode ser denominada de portabilidade.

- Existência de *IDEs* gratuitas, tais como o JCreator, NetBeans e Eclipse, reduzindo custos de desenvolvimento em alguns tipos de aplicações.

- Além da criação de programas do tipo *desktop*, permite como característica marcante a implementação de páginas da web dinâmicas, possibilitando dessa maneira uma maior facilidade de atualizações pelo motivo de serem realizadas unicamente no servidor, podendo ser geradas novas versões de forma automática e com maior mobilidade de utilização, precisando apenas de uma conexão com a internet para acessar o sistema, transmitindo uma maior segurança a ataques de vírus e outras ameaças existentes.

- Totalmente orientada a objetos, que consolida o seu conceito de herança, ou seja, permite a reutilização de códigos já escritos anteriormente.

- Sua vantagem em relação ao C e C++, no qual o *Java* se baseou para sua criação, foi a retirada da responsabilidade do programador em gerenciar ponteiros e memória.

- O uso de *applets* que executa instruções automáticas em redes públicas (internet) com segurança garantida pela linguagem, impedindo ao máximo que uma aplicação considerada insegura seja executada em uma máquina que a acesse.

- Desvantagens do *Java*:

- Sua portabilidade, capaz de suportar qualquer plataforma para rodar um mesmo sistema, possui um custo de tornar o sistema mais lento em algumas aplicações. No entanto, essa perda se justifica pela quantidade de recursos de alto desempenho que o *Java* é capaz de executar. Vários desenvolvedores estão buscando, inclusive otimizar esses esforços aplicados para que brevemente sua performance possa ser comparada a uma linguagem compilada diretamente para o processador.

3.2.4 - Resultado Comparativo: *Java* x *Delphi*

Após analisarmos todas as vantagens e desvantagens entre *Delphi* e *Java*, podemos concluir que ambas possuem pontos particulares que contribuem para uma divisão de opiniões entre programadores para a melhor escolha de desenvolvimento, porém nesse estudo voltado para uma aplicação de gestão de clínicas médicas, que necessita de um desenvolvimento

robusto e de longo prazo, iremos demonstrar argumentos baseados em pesquisa de mercado e bibliográficas que apontam a linguagem *Java* como sendo a ferramenta de desenvolvimento de *software* que mais está em ascensão para aplicações de grande porte em empresas no ramo de TI e acompanhar essa tendência é uma atitude necessária para sistemas que precisam de um longo ciclo de vida, como no caso da área de saúde.

Capítulo 4 – Sistema de Gerenciamento de Banco de Dados (SGBD)

4.1- Definição

Os sistemas de banco de dados se tornaram ferramentas cotidianas na organização de informações de empresas em todos os setores da sociedade. Seria praticamente impossível competir com o mercado atualmente sem utilizar ferramentas dessa área da Ciência da Computação.

Segundo Silberschatz (2006), podemos dizer que um sistema de gerenciamento de banco de dados é uma coleção de dados, onde são armazenados os diversos tipos de informações de uma empresa, relacionados entre si e um conjunto de programas que os acessa. Sistemas de banco de dados são projetados para gerenciar grandes blocos de informações, fornecendo mecanismos para a manipulação e segurança destas, descrevendo as suas estruturas, tipos de dados, relacionamentos e restrições, tornando-se parte essencial para desenvolvimento de qualquer *software* nas mais variadas empresas de tecnologia da informação.

4.2 – Breve Histórico

O crescimento do uso de banco de dados surgiu a partir da década de 1970, mas somente com a revolução da internet no final da década de 1990 os usuários passaram a ter um acesso mais direto a essa ferramenta computacional. Atualmente, em quase todo tipo de transação bancária, acesso a internet, dentre outros, estamos há um freqüente acesso aos bancos de dados das respectivas empresas responsáveis por estes.

4.3 – Estrutura Lógica

A estrutura lógica de um banco de dados pode ser expressa por um diagrama, constituído dos seguintes componentes, conforme veremos na Figura 6:

- **Retângulos:** Representam uma entidade ou conjuntos de entidade, ou seja, um objeto com atributos particulares. Ex.: paciente e médico.
- **Elipses:** Representam atributos de uma entidade, ou seja, as informações pessoais de um paciente ou médico. Ex.: *idpaciente*, *nome_paciente* e *crm_medico*.
- **Losangos:** Representam as relações entre uma entidade ou conjunto de entidades. Ex.: médico atende paciente
- **Linhas:** Indicam as ligações de entidades ou conjunto de entidades aos seus atributos e relações.

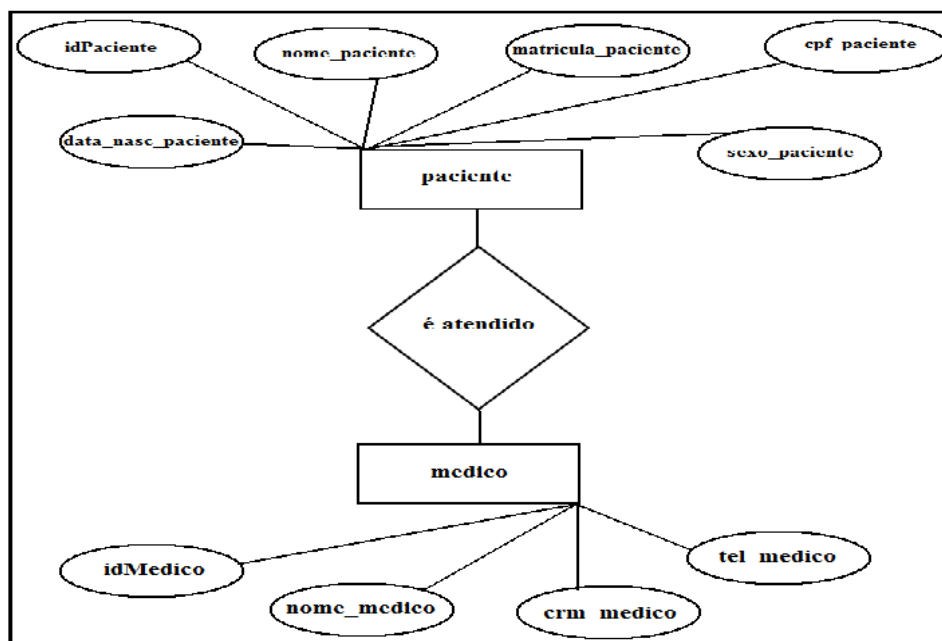


Figura 6 – Exemplo Lógico de Modelo Entidade/Relacionamento

4.4 – Características e Funcionalidades

Um sistema de gerenciamento de banco de dados (SGBD) contém entre suas características, a forma eficiente e correta de sua utilização por um sistema de gestão de saúde, transmitindo a confiabilidade necessária ao cliente.

- **Controle de Redundâncias:** Permite que informações semelhantes não sejam armazenadas em locais diferentes, ocasionando informações cadastrais e históricas da duplicadas da clínica.
- **Compartilhamento de Dados:** Permite controlar a concorrência ao acesso de dados, garantindo que operações de escrita e leitura de dados funcionem sem erros.
- **Controle de Acesso:** Permite dispor recursos que selecionam a autoridade de cada usuário. Desta maneira, alguns usuários somente são capazes de visualizar ou alterar os dados que foram selecionados.
- **Interface:** Permite disponibilizar ferramentas gráficas para melhor interação do sistema com o usuário através de gráficos, consultas SQL e outros acesso no menu.

4.5 - Modelos de Banco de Dados

O modelo de dados é o conjunto de conceitos utilizados para descrever um banco de dados, no qual existem diversas formas de representação, que permitem a compreensão simplificada de sua estrutura física, traduzindo todo o processo que realmente acontece no armazenamento de informações inseridas pelo usuário do sistema.

4.5.1 – Modelo Relacional

O modelo relacional é atualmente mais utilizado em aplicações comerciais de grande porte para processamento de dados, pois suas importantes funcionalidades não implicam em maiores dificuldades e tornam o trabalho do programador mais fácil.

Segundo Silberschatz (2006), a definição para banco de dados relacional é que ele consiste em uma coleção de tabelas, cada uma com um nome único atribuído. As linhas (entidade) da tabela (conjunto de entidades) representam uma relação entre um conjunto de valores.

No modelo relacional, as informações de uma entidade (tabelas) são representadas por tuplas (linhas ou registros) e os atributos (colunas ou campos). Os nomes das tabelas e dos campos são fundamentais para compreensão do que está sendo armazenado, o local e a relação entre os dados armazenados.

Tabela 2 - Tabela de Relacionamento "Pacientes"

Id_Paciente	nome_Paciente	cpf_Paciente	cidade_Paciente	Medico_idMedico
125	Carlos Maia	xxx.xxx.xxx-xx	Fortaleza	125
578	Igor Araújo	xxx.xxx.xxx-xx	Rio de Janeiro	483
483	Renata Alencar	xxx.xxx.xxx-xx	São Paulo	578

Tabela 3 - Tabela de Relacionamento "Médicos"

id_Medico	nome_Medico	crm_Medico
125	Fernando Sales	1234
578	Celso Ponte	5678
483	Eduardo Maia	9123

As tuplas em uma relação de banco de dados precisam da utilização de chaves, que são utilizadas para a identificação única de um registro. Por exemplo, um paciente com o mesmo nome de outro deve ter uma característica que seja exclusiva dele, e dessa maneira, ser possível diferenciá-los no caso de um sistema médico. As principais chaves são:

- **Superchave:** Conjunto de um ou mais atributos de uma tabela que permitem identificar uma tupla unicamente. Ex.: Um paciente pode ser identificado pelo seu código (id) na clínica agregado ao valor do seu Cadastro de Pessoas Física (CPF). (Vide Tabela 2)

- **Candidatas:** Conjunto de uma ou mais atributos de uma tabela que deverão ser escolhidos a ser uma superchave ou chave primária. Ex.: A matrícula do paciente no convênio poderia se candidatar a ser uma dessas duas chaves, mas apesar dessa matrícula ser única para um determinado convênio, ela pode ser semelhante à de outro paciente em outra operadora.
- **Primária** (Representada por *PK*, do inglês *Primary Key*): Trata-se da chave candidata escolhida para identificação de registros em uma tabela e deve ser escolhida com muito cuidado para se evitar inconsistências ou duplicidades no banco de dados. Ex.: O médico poderia ser identificado unicamente com seu código (*id*) na clínica, sua inscrição no Conselho Regional de Medicina (CRM) ou ambos. (Vide Tabela 2)
- **Estrangeira** (Representada por *FK*, do inglês *Foreign Key*): Trata-se da chave primária de uma tabela que é utilizada em uma segunda tabela para referenciar uma a outra. Ex.: Se colocarmos o código (*id*) de um médico, no caso de ser a chave primária, referenciado na tabela que contém os atributos da tabela de pacientes, estamos informando implicitamente que o paciente é consultado por esse médico. (Vide Tabelas 2 e 3)

4.6- Ferramentas de Bancos de Dados

O *software* médico analisado nesse estudo teve inicialmente o uso do banco de dados Paradox, mas devido às necessidades atuais de migração de aplicações antigas para as novas tendências de desenvolvimento de sistemas de informação, recentemente está sendo utilizado o banco SQL Server, no qual mostraremos suas características nos tópicos a seguir.

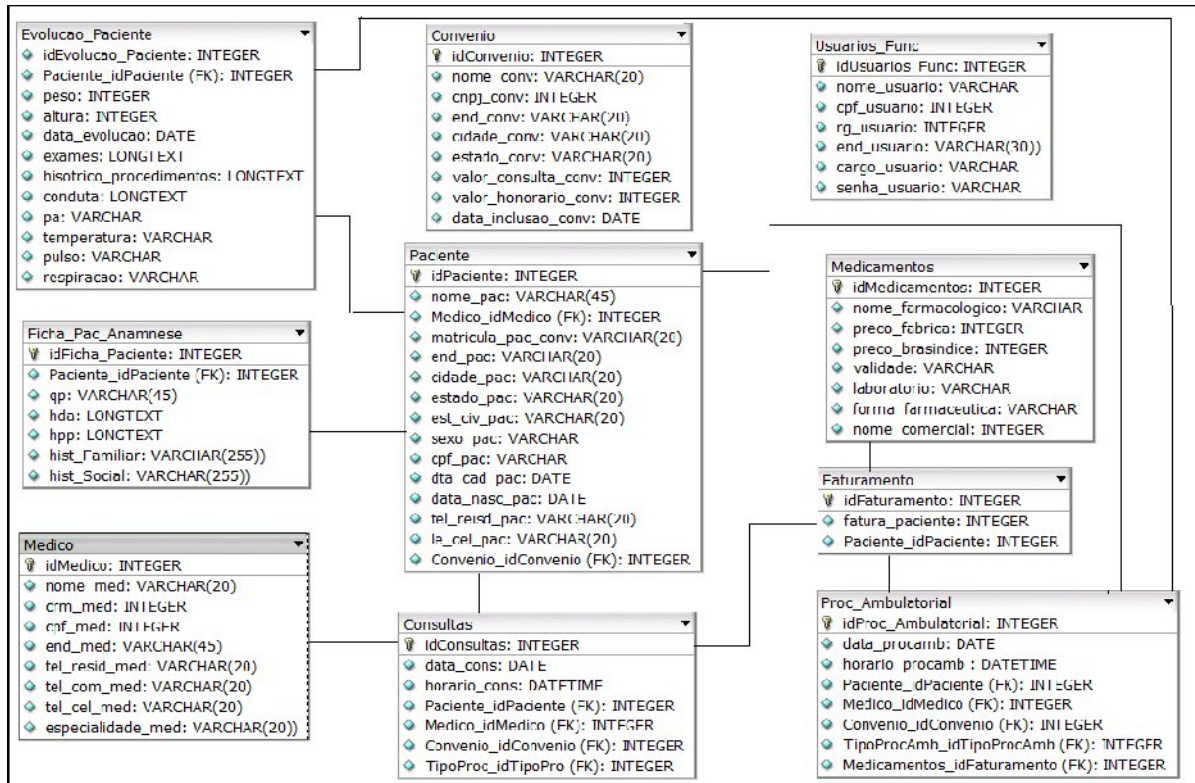


Figura 7 - Diagrama de Banco de Dados do Sistema Utilizado

4.6.1 - Paradox

O Paradox é um sistema de banco de dados baseado em arquivos que contém registros de dados de ordem fixa que somente poderão ser alterados fisicamente dentro do próprio arquivo, permitindo uma fácil navegação entre os registros, já que é possível localizar um registro em uma tabela através de sua posição sem necessidade de referenciar seu conteúdo.

Segundo adaptações realizadas a partir do artigo “Interbase vs Paradox” da *Delphi Informant Magazine*, podemos concluir que apesar de ser bastante veloz em um disco rígido local, sua performance diminui significativamente em uma rede devido as diversas rotinas que são necessárias para ler ou manipular dados, pois o servidor de arquivos apenas tem a função de enviar os dados requeridos sem que nenhum processamento de dados seja realizado, tornando a aplicação mais lenta a medida que novos usuários conectam-se no sistema.

Atualmente, essa linguagem não é muito utilizada devido a diversas desvantagens computacionais que apresenta em relação aos novos bancos de dados existentes no mercado, não atendendo satisfatoriamente as necessidades de um sistema de porte mais avançado.

4.6.2 - SQL Server (*Structured Query Language*)

Com a crescente necessidade de novas ferramentas de banco de dados comerciais na área da Ciência da Computação, pois com apenas as práticas de organização das informações através de conceitos extraídos da álgebra relacional não estavam sendo suficientes para atender sistemas de informação comerciais cada vez mais específicos e com poder computacional para coletar, processar e transmitir um grande volume de dados, o sistema de gerenciamento de banco de dados SQL surgiu baseada no aprimoramento desses conceitos, a partir de um projeto da IBM, como uma ferramenta para auxílio de consultas a banco de dados e atualmente se estabelece como a linguagem padrão de banco de dados relacional. A SQL é dividida em várias partes, tais como:

- **Linguagem de Definição de Dados (DDL**, do inglês: *Data Definition Language*): Fornece definições de esquemas de relação.
- **Linguagem de Manipulação de Dados (DML**, do inglês: *Data Manipulation Language*): Fornece uma linguagem de consulta baseada na álgebra relacional.
- **Integridade**: Fornece restrições de integridade em suas *DDLs*, restringindo as atualizações que as violam.
- **Definição de View**: Fornece comandos para definições de visões.
- **Controle de Transação**: Fornece comandos para especificar o início e fim das transações.
- **Autorização**: Fornece comandos para especificar direitos em visões e relações.
- **SQL Embutida / SQL Dinâmica**: Fornece instruções de como incorporar a SQL às linguagens de programação.

A linguagem SQL aceita diversos tipos de dados, onde destacamos:

- ***char(n)* ou *varchar(n)***: permite a inclusão de letras e números com um determinado tamanho de caracteres (n).
- ***int***: permite apenas a inclusão de números inteiros.

Em regra geral, a SQL pode ser definida parcialmente como r sendo o esquema da relação ou nome da tabela, A_i é nome dos atributos e D_i é o tipo dos valores dos atributos, conforme Quadro 19.

Quadro 19 – Definição da Regra de Esquema em SQL

```
create table  $r_1(a_1d_1, a_2d_2, \dots, a_nd_n)$ 
  <restrição-de-integridade1>,
  ...
  <restrição-de-integridadek>
```

Para uma melhor compreensão dentro do contexto na área médica, citaremos também um exemplo da criação de uma tabela de médicos, na qual descrevemos no modelo relacional anteriormente, na linguagem SQL:

Quadro 20 - Exemplo de Esquema SQL: Criação da Tabela "MEDICOS"

```
create table medico
  (idMedico int,
   nome_Medico char(30), crm_Medico int,
   primary key (idMedico, crm_Medico))
```

Após a criação de uma tabela na SQL, vários outros importantes comandos podem ser utilizados para realizarmos as operações que a compõem, tais como inserir e remover registros, remover a tabela criada, inserir e remover atributos, dentre outros.

A SQL possui também um conjunto de cláusulas que permite a realização de variados tipos de consultas ao banco de dados, nas quais podemos selecionar os atributos desejados de uma tabela estabelecendo restrições para que isto aconteça, dentre outros.

Geralmente, os usuários de um determinado sistema estão conectados ao banco de dados através de uma rede. Desse modo, veio a denominação Cliente/Servidor, onde a máquina servidora é o local de execução real do banco de dados, enquanto a máquina cliente trabalha de modo remoto com o primeiro.

Por isto, a Microsoft criou a *SQL Server*, um sistema de gerenciamento de banco de dados relacional utilizado em servidores de empresas, tais como hospitais e clínicas de grande porte operacional. Sua aplicação permite um fornecimento de variadas ferramentas gráficas e assistentes que orientam os administradores a realizar operações que garantam uma maior segurança física dos dados, tais como agendamento de *backups* regulares, duplicação de dados, analisar desempenho, uma melhoria em relação ao SQL nas consultas. Através da ferramenta interna *SQL Server Management Studio*, o administrador de banco de dados se dispõe de importantes mecanismos que o auxiliam no projeto.

4.6.3 – *Extensible Markup Language* (XML)

Com o surgimento do TISS, através das Resoluções Normativas, entre 2005 e 2007, após pesquisas de sua elaboração desde maio de 2003, foi estabelecido como padrão obrigatório um modelo unificado para troca de informações entre prestadores de serviço e operadoras de saúde, lançado pela Agência Nacional de Saúde Suplementar (ANS). Esse padrão buscava aprimorar a comunicação entre o setor da saúde, informatizando o processo de envio de dados de maneira a facilitar o acesso do beneficiário aos serviços oferecidos, as informações epidemiológicas para definições de políticas de saúde, favorecimento da análise de custos e investimentos, redução de custos administrativos e excesso de uso de papéis; isto tem colaborado para reduzir de maneira significativa, a ocorrência de glosas dos procedimentos realizados, além de outras diversas melhorias que eram pretendidas com o TISS.

A partir daí, foi definido pela ANS que essa comunicação entre prestadores e operadoras de saúde seria através da linguagem de banco de dados XML. Com isso, os sistemas voltados para a área de saúde precisaram criar módulos que possibilitassem a geração dos arquivos das guias médicas e outras informações para este formato.

A XML tem como característica principal a estruturação de um grande volume de informações. Inicialmente era conhecida como uma linguagem de gerenciamento de documentos, oferecendo um padrão de marcação de dados. Em comparação com o armazenamento de dados em um banco no modelo relacional, a representação XML pode ser

ineficaz, pois os nomes de *tag* (cabeçalhos ou nome dos atributos com seus respectivos registros) são repetidos em todo o documento. Como veremos a seguir, sua estrutura física poderá ser representada como estruturas de árvores do tipo pai-filho, com os nós correspondendo aos elementos e atributos.

Apesar da sobrecarga de informações ocasionada pelo uso repetido de *tags*, esse tipo de representação de banco de dados a torna auto-documentável e sem muita rigidez no envio de dados, permitindo ao remetente acrescentar informações complementares e cabendo ao destinatário aceitá-las ou descartá-las. Por esse e outros motivos, essa linguagem foi escolhida pela ANS como padrão para comunicação entre prestadores e operadoras de saúde (convênios).

Os dados na XML podem ser armazenados em sistemas de arquivos ou em através de sua representação interna e sua estrutura de consulta de dados chamada *XQuery* que parece bastante com a SQL. O padrão SQL/XML estabelece uma extensão padrão da SQL, permitindo uma saída de dados no formato XML, devido à sua grande utilização de armazenamento de dados complexos e na troca de informações padronizadas entre organizações comerciais e industriais. Veremos logo a seguir (Quadro 21), um exemplo de um arquivo gerado nesse formato, demonstrando um lote de entrega de faturamento da clínica.

Quadro 21 - Exemplo de Formatação XML para Envio de Faturamento Entre a Clínica e Convênios.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<ns:mensagemTISS xmlns:ans="http://www.ans.gov.br/padroes/tiss/schemas"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ans.gov.br/padroes/tiss/schemas
    http://www.ans.gov.br/padroes/tiss/schemas/tissV2_02_01.xsd">
  <ns:cabecalho>
    <ns:identificacaoTransacao>
      <ns:tipoTransacao>ENVIO_LOTE_GUIAS</ns:tipoTransacao>
      <ns:sequencialTransacao>739</ns:sequencialTransacao>
      <ns:dataRegistroTransacao>2010-06-01</ns:dataRegistroTransacao>
      <ns:horaRegistroTransacao>09:44:49.0Z</ns:horaRegistroTransacao>
    </ns:identificacaoTransacao>
    <ns:origem>
      <ns:codigoPrestadorNaOperadora>
        <ns:CNPJ>000000000000000</ns:CNPJ>
      </ns:codigoPrestadorNaOperadora>
    </ns:origem>
    <ns:destino>
      <ns:registroANS>000000</ns:registroANS>
    </ns:destino>
    <ns:versaoPadrao>2.02.01</ns:versaoPadrao>
  </ns:cabecalho>
  <ns:prestadorParaOperadora>
    <ns:loteGuias>
      <ns:numeroLote>739</ns:numeroLote>
    <ns:guias>
      <ns:guiaFaturamento>
        <ns:guiaSP_SADT>
          <ns:identificacaoGuiaSADTSP>
            <ns:identificacaoFontePagadora>
              <ns:registroANS>403911</ns:registroANS>
            </ns:identificacaoFontePagadora>
            <ns:dataEmissaoGuia>2010-05-20</ns:dataEmissaoGuia>
            <ns:numeroGuiaPrestador>110741409221</ns:numeroGuiaPrestador>
            <ns:numeroGuiaOperadora>110741409221</ns:numeroGuiaOperadora>
          </ns:identificacaoGuiaSADTSP>
          <ns:numeroGuiaPrincipal>110741409221</ns:numeroGuiaPrincipal>
          <ns:dadosAutorizacao>
            <ns:dataAutorizacao>2010-05-19</ns:dataAutorizacao>
            <ns:senhaAutorizacao>0000000</ns:senhaAutorizacao>
            <ns:validadeSenha>2010-06-17</ns:validadeSenha>
          </ns:dadosAutorizacao>
          <ns:dadosBeneficiario>
            <ns:numeroCarteira>0000000000000</ns:numeroCarteira>
            <ns:nomeBeneficiario>José Sousa </ns:nomeBeneficiario>
            <ns:nomePlano>RE</ns:nomePlano>
            <ns:validadeCarteira>2111-10-31</ns:validadeCarteira>
          </ns:dadosBeneficiario>
          <ns:dadosSolicitante>
            <ns:contratado>
              <ns:identificacao>
                <ns:CNPJ>000000000000000</ns:CNPJ>
              </ns:identificacao>
              <ns:nomeContratado>Quimioclinic Ltda </ns:nomeContratado>
              <ns:enderecoContratado>
                <ns:tipoLogradouro>081</ns:tipoLogradouro>
                <ns:logradouro>Osvaldo Cruz</ns:logradouro>
              </ns:enderecoContratado>
            </ns:contratado>
          </ns:dadosSolicitante>
        </ns:guiaFaturamento>
      </ns:loteGuias>
    </ns:prestadorParaOperadora>
  </ns:mensagemTISS>

```

Quadro 21 - Continuação

```

ns:numero>0000</ans:numero>
ns:complemento>Dionísio Torres</ans:complemento>
ns:codigoIBGEMunicipio>00000000</ans:codigoIBGEMunicipio>
ns:municipio>Fortaleza</ans:municipio>
ns:codigoUF>CE</ans:codigoUF>
ns:cep>00000000</ans:cep>
</ans:enderecoContratado>
ns:numeroCNES>000000</ans:numeroCNES>
</ans:contratado>
ns:profissional>
ns:nomeProfissional>Dr. XXXXXXXXX</ans:nomeProfissional>
ns:conselhoProfissional>
ns:siglaConselho>CRM</ans:siglaConselho>
ns:numeroConselho>000000</ans:numeroConselho>
ns:ufConselho>CE</ans:ufConselho>
</ans:conselhoProfissional>
</ans:profissional>
</ans:dadosSolicitante>
ns:prestadorExecutante>
ns:identificacao>
ns:CNPJ>0000000000000000</ans:CNPJ>
</ans:identificacao>
ns:nomeContratado>Quimioclinic Ltda</ans:nomeContratado>
ns:enderecoContratado>
ns:tipoLogradouro>081</ans:tipoLogradouro>
ns:logradouro>Osvaldo Cruz</ans:logradouro>
ns:numero>0000</ans:numero>
ns:complemento>Dionísio Torres</ans:complemento>
ns:codigoIBGEMunicipio>2304400</ans:codigoIBGEMunicipio>
ns:municipio>Fortaleza</ans:municipio>
ns:codigoUF>CE</ans:codigoUF>
ns:cep>00000000</ans:cep>
</ans:enderecoContratado>
ns:numeroCNES>000000000</ans:numeroCNES>
ns:profissionalExecutanteCompl>
ns:nomeExecutante>Dr. XXXXXXXXX</ans:nomeExecutante>
ns:conselhoProfissional>
ns:siglaConselho>CRM</ans:siglaConselho>
ns:numeroConselho>0000</ans:numeroConselho>
ns:ufConselho>CE</ans:ufConselho>
</ans:conselhoProfissional>
ns:codigoProfissionalCompl>
ns:codigoPrestadorNaOperadora>90010752</ans:codigoPrestadorNaOperadora>
</ans:codigoProfissionalCompl>
</ans:profissionalExecutanteCompl>
</ans:prestadorExecutante>
ns:indicacaoClinica>Paciente portador de Carcinoma pulmonar, T4 N1 M0 - IIIB, em tratamento
quimioterapico em carater neoadjuvante com 06 ciclos do protocolo CBDP AUC-6 + Pemetrexede
Dissodico 500mg/m2, a cada 21 dias. Realizando o 2º ciclo</ans:indicacaoClinica>
ns:caraterAtendimento>E</ans:caraterAtendimento>
ns:dataHoraAtendimento>2010-05-20T14:00:57.0Z</ans:dataHoraAtendimento>
ns:diagnosticoAtendimento>
ns:CID>
ns:nomeTabela>CID-10</ans:nomeTabela>
ns:codigoDiagnostico>C34</ans:codigoDiagnostico>
ns:descricaoDiagnostico>Carcinoma pulmonar</ans:descricaoDiagnostico>
</ans:CID>

```


Quadro 21 - Continuação

```

ns:tipoDoenca>C</ans:tipoDoenca>
ns:tempoReferidoEvolucaoDoenca>
ns:valor>1</ans:valor>
ns:unidadeTempo>M</ans:unidadeTempo>
</ans:tempoReferidoEvolucaoDoenca>
</ans:diagnosticoAtendimento>
ns:tipoSaida>2</ans:tipoSaida>
ns:tipoAtendimento>08</ans:tipoAtendimento>
ns:procedimentosRealizados>
ns:procedimentos>
ns:procedimento>
ns:codigo>0000000000</ans:codigo>
ns:tipoTabela>06</ans:tipoTabela>
ns:descricao>Terapia oncológica - planejamento e 1º dia de tratamento</ans:descricao>
</ans:procedimento>
ns:data>2010-05-20</ans:data>
ns:quantidadeRealizada>1</ans:quantidadeRealizada>
ns:viaAcesso>U</ans:viaAcesso>
ns:tecnicaUtilizada>C</ans:tecnicaUtilizada>
ns:valor>102.00</ans:valor>
ns:valorTotal>102.00</ans:valorTotal>
</ans:procedimentos>
</ans:procedimentosRealizados>
ns:valorTotal>
ns:materiais>.13</ans:materiais>
ns:medicamentos>1133.39</ans:medicamentos>
ns:totalGeral>1133.52</ans:totalGeral>
</ans:valorTotal>
</ans:guiaSP_SADT>
</ans:guiaFaturamento>
</ans:guias>
</ans:loteGuias>
</ans:prestadorParaOperadora>
ns:epilogo>
ns:hash>E6815F52E97E27C4FD09CFFF3E7C0198</ans:hash>
</ans:epilogo>
</ans:mensagemTISS>

```

Capítulo 5 – Engenharia de *Software*

5.1- Introdução

Segundo Pressman (1992), uma definição adequada para o termo *software*, seria um sistema lógico que contém uma seqüência de instruções (programas de computadores) que ao serem executadas produzem uma função desejada, sendo que, manipulando adequadamente as informações e operações, permite reduzir consideravelmente o tempo de execução de tarefas.

O avanço de tecnologias nas áreas de análise, projeto, desenvolvimento e qualidade de *software* permitem a criação de ferramentas mais poderosas, buscando sempre uma integração entre todas essas etapas, de modo a garantir que os problemas atuais sejam solucionados e evitados no futuro de aplicações, gerando maior satisfação e enriquecimento de informações, principalmente para os potenciais clientes comerciais na área da saúde, que tem buscado controlar suas ações através de robustos sistemas de informações. O intuito é garantir uma melhoria na qualidade dos seus serviços, aliado a um aumento na produtividade do corpo clínico, da gerência e da administração, trazendo grandes retornos de redução de custos e maior satisfação de atendimento aos pacientes.

Para iniciantes na área da Ciência da Computação, geralmente, desenvolver software pode ser facilmente confundido com programação, e apesar de não ser um raciocínio totalmente equivocado, quando se deseja resolver problemas mais complexos como um grande sistema médico, abrangendo nele diversas outras áreas e suas especificações técnicas, essa abordagem mais individual não é muito indicada, precisando-se de uma engenharia necessária para que esse desenvolvimento tenha um longo ciclo de vida e aproveitamento esperado.

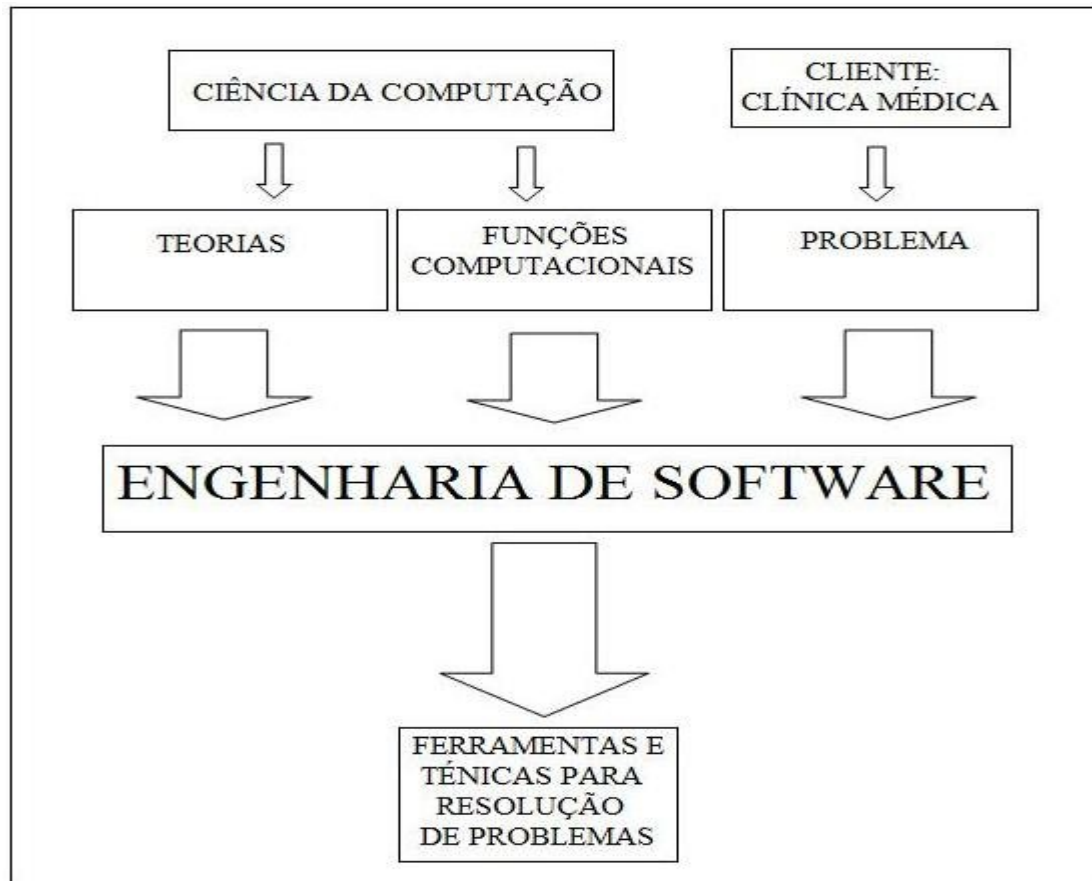


Figura 8 – Caracterização da Engenharia de *Software*

5.2 – Técnicas de Comunicação: Desenvolvedor x Cliente

Apesar da clínica analisada nesse estudo já ter adquirido desde o início o *software* médico quase todo pronto para ser comercializado, algumas modificações foram feitas com o intuito de atender as necessidades internas solicitadas. Para isso, uma reunião entre o desenvolvedor do *software* e o cliente (clínica) foi marcada para definição e compreensão dos objetivos ou funções que o sistema iria desempenhar para conseqüente aprovação deste.

Em casos que seja solicitado uma solução completamente nova a ser implementada, é discutido mais amplamente a questão de interfaces, configurações, bancos de dados utilizados, marcos de prazos de entrega, previsão de orçamentos iniciais, dentre outros. Apresentaremos nos tópicos seguintes as atividades necessárias e fundamentais para um projeto de gerenciamento de *software*.

5.3 - Planejamento, Identificação dos Riscos, Prazos, Monitoramento e Controle

O planejamento do processo de gerenciamento de *software* precisa ser muito bem detalhado em todas as suas fases para que não aconteça, como freqüentemente ocorre, prorrogações de prazos e aumentos orçamentários, que por sua vez, podem implicar por diversas vezes na suspensão ou cancelamento de todo o projeto. Por isso, tão breve seja iniciada a elaboração do projeto, todos os riscos devem ser identificados, colocados em ordem prioritária e solucionados anteriormente ao momento que de fato poderiam acontecer.

O processo de monitoramento determina que todas as atividades desenvolvidas no programa sejam documentadas, permitindo que o gerente de projetos faça o controle através de ferramentas de planejamento automatizadas para que sejam relatadas quais as atividades que estão atrasadas e desta maneira redirecionar recursos através de novas contratações e outros métodos que podem ser empregados para essa finalidade.

5.4 – Gerência de Projetos de *Software*

Os *software* realizaram grandes avanços na maioria dos setores da economia mundial, inclusive na medicina, mas não estão isentos de eventuais falhas que geralmente ocorrem, tal como erros de codificação e outros problemas, no qual embora um sistema funcione, ele não age da maneira esperada. Portanto, a Engenharia de *Software* foi tendo cada vez mais aceitabilidade para que sejam reduzidas ao máximo as freqüentes insatisfações de clientes não atendidos pelas suas necessidades reais, objetivados com a aquisição ou contratação de um *software*, causando perda de tempo, capital investido, entre outros transtornos.

A gerência de projetos é o alicerce fundamental para o processo de criação de um *software*, pois precisa acompanhar o desenvolvimento em todas as etapas. Para que realmente seja criado um projeto de *software* bem sucedido e que atenda plenamente as necessidades básicas dos usuários que lidam com a gestão de uma clínica médica especializada e todas as peculiaridades envolvidas no processo de administração desta, deve-se compreender inicialmente todo o escopo do trabalho a ser feito, levantar riscos e recursos exigidos, planejar as tarefas a serem executadas, traçar cronogramas com marcos de referência e definir estimativas de custos precisas a fim de evitar-se problemas futuros com

clientes, tendo que se realinhar novos recursos financeiros não programados e prejudicando o andamento, e em alguns casos, até mesmo a suspensão ou cancelamento de todo o projeto.

A qualidade de um *software* está relacionada com a visão de quem está o analisando, e muitas vezes, pode ser mal interpretada, pois enquanto um usuário julga um *software* pela fácil usabilidade dele para execução das tarefas, pode acontecer que um segundo *software* tenha um entendimento de aprendizado mais difícil, mas a sua funcionalidade poderá torná-lo de maior qualidade ao primeiro. No caso da visão do desenvolvedor, a sua qualidade final está relacionada com a quantidade de defeitos nas inspeções de requisitos, projetos e códigos.

Para se evitar tantos defeitos e insatisfações de clientes, um projeto de desenvolvimento de *software* necessita de um trabalho em equipe e de importantes estágios de planejamento, que englobam processos conjuntos, nos quais detalharemos nos tópicos a seguir.

5.4.1 - Análise e Definição de Requisitos

Esse estágio do processo de desenvolvimento tem como objetivo o levantamento fornecido pelo cliente, através de entrevistas e acompanhamentos da rotina, de todas as características e funcionalidade que deve conter no sistema e os responsáveis por essa atividade, devem identificar todos esses requisitos, definindo-os e classificando-os em funcionais ou não-funcionais e também os documentando em um banco de dados até que se possa ser criado um protótipo com partes do desenvolvimento para análise do usuário, e dessa maneira, um acordo da utilidade do sistema poderá ser definido e validado. Após isso, esses requisitos são reescritos para uma representação técnica ao projetista, na qual se dará início a um próximo estágio, comentado a seguir.

A ausência ou falha na identificação de requisitos são causas freqüentes de problemas existentes em grandes projetos de *software*. Portanto, durante a ocasião das entrevistas, geralmente um mesmo questionamento pode ser realizado de diversas maneiras, para se ter certeza que foi entendido todos os requisitos. Uma triagem é feita pela equipe e será indicado o que pode ser satisfeito e o que pode ser dispensável, com o intuito de auxiliar na otimização das reais necessidades do sistema, pois o cliente facilmente conhece sobre o seu negócio, mas

nem sempre é capaz de descrever com exatidão o que desejam e dificilmente conseguem relatar seus problemas para pessoas externas a sua área. Da mesma maneira, os desenvolvedores são capazes de criar soluções computacionais, mas não é capaz o suficiente de perceber como essas possíveis soluções irá afetar o negócio de um determinado cliente.

Geralmente, como aconteceu no caso acompanhado na obtenção do *software* médico pela clínica analisada nesse estudo, nessa etapa não há menção a linguagem de programação, banco de dados, especificações de *hardware* a serem utilizados e outras descrições específicas da implementação. Portanto, essas escolhas independem do propósito da funcionalidade do sistema e somente será relacionada se houver uma exigência por parte do cliente.

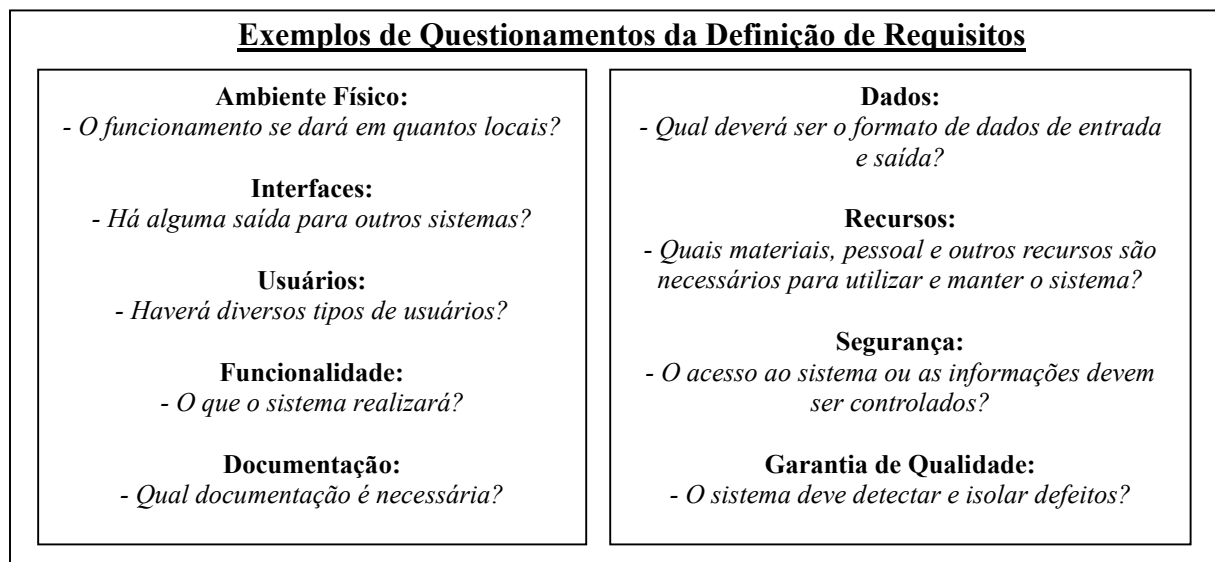


Figura 9- Elaboração de Definição de Requisitos

Quanto à classificação dos requisitos, pode-se concluir que os **funcionais** correspondem ao levantamento das funcionalidades gerais do sistema, enquanto os **não-funcionais** correspondem a uma restrição, comportamento ou propriedade ou preferência de uma especificação deste.

Quadro 22 - Exemplos de Requisitos Funcionais

1. O sistema deverá permitir o cadastro e agendamento de pacientes pelos médicos e funcionários da recepção.
2. O sistema deverá permitir a prescrição de medicações, laudos, solicitações de exames somente aos médicos.
3. O sistema deverá permitir o faturamento aos usuários administradores.

Quadro 23 - Exemplos de Requisitos Não-Funcionais

1. A linguagem de desenvolvimento será o <i>Delphi</i> (ou <i>Java</i>)
2. O banco de dados será o Paradox (SQL Server)
3. O sistema deverá controlar e auditar o acesso e modificações por utilização de uma senha privativa ao usuário.

5.4.2 - Projeto do Sistema

Após o levantamento dos requisitos, a próxima etapa será traduzir as necessidades dos clientes para a elaboração de um projeto de desenvolvimento do sistema. Para iniciarmos o esse estágio, precisamos entender o significado de projeto, e Pfleeger (2004) foi capaz de definir como sendo o processo criativo de transformar o problema em uma solução.

Um determinado projeto de *software* possui uma solução adequada quando esta satisfaz os requisitos especificados pelo cliente. Dessa maneira, é nessa etapa que são criados projetos

de sistema (conceituais), que terão de ser capazes de mostrar, na linguagem do cliente, as funcionalidades do sistema, para que este possa aprovar ou solicitar alterações não observadas no processo. Quando aprovados, um projeto técnico será originado, a partir da tradução rigorosa do conceitual, possibilitando a equipe de desenvolvimento analisar as necessidades reais de *hardware* e *software* para resolução do problema do cliente.

Os projetos devem atender características fundamentais para que possuam alta qualidade, tais como facilidades no seu entendimento, implementação, realização de testes e na tradução adequada de requisitos, pois as mudanças necessárias com atualização dos requisitos ou correção de falhas, podem resultar em modificação deste. Também, sabemos que dificilmente será evitada, logo de início, a totalidade de problemas existentes na elaboração de um projeto, mas será preciso disponibilizar um período de tempo com a identificação e tratamentos das exceções e tentar antecipar ou minimizar possíveis conflitos e comprometimento da segurança das informações.

5.4.3 - Projeto do Programa ou Implementação

Essa fase no desenvolvimento do projeto de *software* é um importante passo a ser concretizado, pois incorrerá que diversas dificuldades procedimentais terão de ser solucionadas na estratégia computacional para o início da escrita do código de programação, não impedindo que o propósito principal de um sistema, de garantir sua durabilidade e adaptação a capacidade de permitir modificações, quando requisitadas pelo cliente, seja afetado.

A equipe responsável por esse estágio terá de possuir membros que estejam dispostos a fornecer seus conhecimentos, envolvendo um pensamento flexível, generalizado e criativo, pois estas habilidades são de grande utilidade na especificação da classe de algoritmos a serem utilizados na codificação de cada componente escrito, estimativas de tempo de execução associado à qualidade e para a transformação dos requisitos do cliente para um código simples e adequado para a permanência do ciclo de vida deste.

5.4.4 - Testes de Unidades, Integração e Sistema

Esse estágio tem como objetivo a solução de problemas gerados na implementação ou no levantamento dos requisitos, através de meios eficientes, que garantem uma maior qualidade do sistema que será entregue ao cliente.

Como dito antes, é muito improvável que um sistema de qualidade seja desenvolvido, em sua parcialidade ou totalidade, sem o aparecimento de falhas na sua execução ou funcionamento durante o processo, pois os clientes nem sempre conseguem descrever o que desejam ou o responsável pelo levantamento nem sempre consegue traduzir tecnicamente o que o cliente havia solicitado, além da alta complexidade dos algoritmos, que produzem fórmulas e procedimentos extensos com uma grande variedade de estados. Portanto, podemos afirmar que os testes tem por objetivo minimizar as falhas do sistema.

5.4.5 - Entrega do Sistema

Após a conclusão da realização da elaboração do projeto, implementação e finalização dos testes, o *software* deverá ser entregue ao cliente a partir de um processo formal. Essa fase, embora não pareça, é tão importante quanto as outras já realizadas, pois será preciso mostrar ao cliente e a todos os usuários todas as funcionalidades do sistema, para que estes possam familiarizar-se aos poucos com o ambiente computacional, ao ponto de entenderem como realizar suas atividades, explorando a produtividade total do que fora projetado e oferecido.

Para que essa transferência de tecnologia seja de fato bem-sucedida, é preciso que a equipe tenha criado importantes técnicas de treinamento e uma documentação bem detalhada do desenvolvimento deverá ter sido feita, pois essa tarefa envolve uma grande dificuldade, que seria convencer o usuário de que todos os modos de realizar uma atividade, que ele adquiriu um hábito comum durante um longo período de tempo, seriam apenas aprimorados para a forma computacional. Para isso, os usuários deverão receber cursos teóricos e práticos para o bom funcionamento do sistema, onde serão designados os procedimentos que deverão realizar, quais passos devem seguir e conhecer e as particularidades de cada um. Neste momento, um ou mais funcionários poderão ser eleitos administradores, ou seja, responsáveis pela definição no sistema de quais funções e consultas estarão disponíveis para cada usuário.

Um *software* bem conduzido, além do treinamento intensivo, deverá ter acompanhado a entrega do produto, uma ampla documentação, contendo todas as informações necessárias para sua utilização, através de manuais físicos, agregados como um item de ajuda no próprio sistema ou através de consultas *on-line* disponibilizadas pelo fabricante.

5.4.6 - Manutenção

Como dito antes, um ciclo de vida de um *software*, ao ser entregue ao cliente, ainda está longe de estar concluído, ou seja, é preciso que esse sistema seja sempre revisado com manutenções preventivas a serem incluídas, como também as corretivas apontadas pelos usuários. Esse trabalho, em termos de *software*, é completamente diferente da manutenção de um *hardware*, pois o segundo muitas vezes é resolvido simplesmente na troca de peças danificadas por novas, enquanto um produto de *software* não deve sofrer desgastes do tempo e outros fatores que os diferenciam.

A definição de Pressman (1992), descreve de maneira didática e fácil entendimento o significado de manutenibilidade de um *software* como sendo a facilidade com que um programa possa ser corrigido se um erro for encontrado, adaptado se o seu ambiente se modificar ou ampliado se o cliente desejar inclusões ou alterações nos requisitos funcionais.

Portanto, mais uma vez ressaltamos a importância também de uma documentação mostrando toda a condução do projeto, que venha possibilitar que a equipe de desenvolvimento solucione problemas posteriores, não detectados na fase de teste, ou modificações não previstas inicialmente pelo cliente nos requisitos funcionais, pois uma alteração em qualquer parte do projeto poderá implicar diretamente ou indiretamente em outra parte e isso deverá ser definido nesse documento para que um processo alterado não sobrecarregue ou inviabilize um segundo.

Conclusão

Considerações Finais

Ao final deste estudo, podemos notar claramente que a tecnologia da informação está se mostrando cada vez mais presente em nossas vidas em praticamente todas as atividades que exercemos diariamente. Em geral, as pessoas tem tido uma inclusão digital mais ampla e os diferentes mercados nos diversos setores da sociedade tem demonstrado o surgimento de clientes mais exigentes em relação aos profissionais envolvidos em todas as áreas da Ciência da Computação.

No caso deste estudo, particularmente, podemos concluir que o mercado designado para a medicina tem mostrado grandes avanços em todas as especialidades médicas e a tecnologia da informação tem auxiliado de maneira fundamental, buscando sempre novas soluções para recentes problemas que surgem constantemente. Atualmente, um médico possui uma precisão de diagnósticos que há pouco tempo atrás seria quase raro uma detecção, ocasionando em um tratamento tardio de doenças, podendo muitas vezes prejudicar de maneira tal um determinado paciente. Com esse grande apoio tecnológico, aliada a longos estudos de especialistas, surgem freqüentemente novas medicações que proporcionam aos pacientes de doenças mais graves, tal como o câncer, uma maior eficácia e também uma maior qualidade de vida ao paciente. Os cirurgiões estão cada vez mais munidos de equipamentos e aparelhos que tornam o ato cirúrgico menos complexo e menos traumático.

Foi verificado que a utilização de sistemas computacionais no atendimento a pacientes em clínicas médicas tem auxiliado os profissionais da área, na coleta, armazenamento e gerenciamento de dados técnicos, surgimento e crescimento de novas pesquisas científicas e também nas tomadas de decisões clínicas e gerenciais.

Além de todos esses aspectos, podemos concluir que os sistemas de informação estão cada vez mais especializados e todas as atividades diárias dentro de um complexo médico necessitam e estão controladas por um ambiente computacional eficiente, que proporcione segurança aos profissionais de saúde e funcionários da recepção até o faturamento. Esse estudo procurou mostrar os passos fundamentais, muitas vezes desconhecidos dos profissionais não ligados a área de Computação, para que um *software* médico de qualidade, capaz de desenvolver, auxiliar e controlar rotinas diárias, seja idealizado e projetado até o

momento de sua aquisição pelo cliente, agregando-se a ele um bem de grande valor dentro do seu negócio.

Podemos afirmar que os ganhos obtidos após a instalação do sistema afetou em todos os aspectos, tais como na organização, controle de finanças, agilidade no processo de buscas, análise de relatórios técnicos, gestão e perspectivas futuras.

Recomendações

Após a análise através do conhecimento técnico e prático desenvolvido nesse estudo em relação ao sistema médico da clínica analisada, verificamos que este tipo de *software* comporta, com elevada qualidade de desempenho e integridade das informações, todos os requisitos funcionais contratados pelo cliente. Inclusive, isso nos leva a aceitar que o projeto de desenvolvimento deve ter seguido as diretrizes corretas e apropriadas das áreas da Ciência da Computação citadas nos capítulos deste trabalho.

Alguns problemas como falhas no acesso no *login* ou em consultas, estavam ocorrendo ultimamente com elevada frequência, precisando executar um procedimento de regeneração de índices para resolução do problema. No entanto, no decorrer desse estudo, a clínica observou que alguns dados recentemente inseridos estavam sendo perdidos no sistema, tais como cadastro de pacientes e agendamento de consultas. Então, a empresa de TI responsável pelo desenvolvimento do *software* verificou que o banco de dados Paradox estava sobrecarregado e propôs uma migração das informações para o banco de dados SQL Server e uma atualização da versão do sistema que estava sendo comercializado na atualidade.

Portanto, uma recomendação nos requisitos não-funcionais foi realizada com êxito, que seria a migração do sistema de gerenciamento do banco de dados para uma linguagem mais atual e com maior consistência, conforme no capítulo que abordamos deste assunto.

Apesar da linguagem *Delphi*, atualmente utilizada no desenvolvimento do sistema da clínica, ainda possuir uma grande quantidade de programadores adeptos devido às facilidades oferecidas por seus recursos gráficos, podemos verificar que sua utilidade na implementação de grandes *software* comerciais está sendo cada vez mais restringida pelas vantagens e variedades de funcionalidades da linguagem *Java*. Este sistema pode sofrer futuramente o

risco de ficar sem mão-de-obra adequada para realização de manutenções e atualizações adequadas. Portanto, recomendamos à médio prazo, a viabilização de uma alteração da linguagem de desenvolvimento para o *Java*, tendência mundial em termos de programação, pelos motivos já explicados tecnicamente no capítulo que descrevemos e comparamos cada uma destas.

Além disso, o *Java* tem uma característica que vem trazendo grandes avanços tecnológicos, que vem possibilitando aos sistemas de informação atuais o constante direcionamento para a internet, ou seja, as informações e rotinas desenvolvidas serão disponibilizadas e realizadas pelos usuários, sejam eles clientes, médicos ou funcionários, através de um acesso identificado e seguro, em qualquer lugar do mundo onde haja um computador com acesso a internet. Como exemplo, um paciente da clínica poderá realizar o seu cadastro, marcar sua consulta (aguardando a validação posterior por email), enviar exames e consultar faturas sem a necessidade de ocupar recepcionistas no momento de grande fluxo de atendimento da clínica.

REFERÊNCIAS BIBLIOGRÁFICAS

ANS - Agência Nacional de Saúde Suplementar. Padrão TISS – Troca de Informações em Saúde Suplementar. Disponível em:

http://www.ans.gov.br/portal/site/_hotsite_tiss/pdf/texto_completo.pdf. Acesso em: 22 abr. 2010.

DEITEL H. M. **Java Como programar - 6º Edição**. Tradução de Edson Furmankiewicz. São Paulo: Pearson Education do Brasil, *Java How to Program, 6th Ed*, 2005. ISBN: 85-7605-019-6.

ELMASRI, Ramez, NAVATHE B Shamkant. **Sistemas de Banco de Dados**. Tradução de Marília Guimarães Pinheiro. São Paulo: Pearson Addison Wesley, 2005. Título original: *Fundamentals of Database Systems*. 4ª ed., 2004.

ENGO, Frank. Como Programar em *Delphi 3*. Tradução de Álvaro Antunes. São Paulo: Makron Books do Brasil Editora Ltda. Título Original: *How to Program Delphi 3*, 1997.

GANE, Chris, SARSON, Trish. **Análise Estruturada de Sistemas**. Tradução de Gerry Edward Tompkins. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1983 Título original: *Structured Systems Analysis Tools e Techniques*. 1ª ed., 1979.

HORSTMANN, Cay. **Conceitos de Computação com o Essencial de Java**. Tradução de Werner Loeffler. Porto Alegre: ArtMed Editora SA. Título original: *Computing Concepts with Java Essentials*. 3ª ed., 2004.

INCA - Instituto Nacional do Câncer (Editorial Site Eletrônico). O que é o câncer? O que causa o câncer?

Disponível em: http://www1.inca.gov.br/conteudo_view.asp?id=322. Acesso em: 16 abr. 2010

INCA - Instituto Nacional do Câncer (Editorial Site Eletrônico). História do INCA.

Disponível em: http://www.inca.gov.br/conteudo_view.asp?id=235. Acesso em: 16 abr. 2010

LIMA, C Manolita. **Monografia A Engenharia da Produção Acadêmica**. São Paulo: Editora Saraiva. 2ª ed., 2006.

PFLEEGER, S. Lawrence. **Engenharia de Software - Teoria e Prática**. Tradução de Dino Franklin. São Paulo: Prentice Hall, 2004. Título Original: *Software Engineering - Theory and Practice, Second Edition*, 2004.

PRESSMAN, S. Roger. **Engenharia de Software**. Tradução de João Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995. Título original: *Software Engineering: A Practitioner's Approach*. 3^a ed., 1992.

SILBERSCHATZ, Abraham, KORTH, F. Henry, SUDARSHAN, S. **Sistema de Banco de Dados**. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier Editora Ltda. Título original: *Database System Concepts*. 5^a ed., 2006.

SOMERA, Guilherme.(2007); **Treinamento Profissional em Delphi**. Digerati Books, São Paulo, 2007.

SOMMERVILLE Ian. **Engenharia de Software - 8º Edição**. Tradução de Selma Shin Shimizu Melnikoff. São Paulo: Pearson Addison-Wesley. Título Original: *Software Engineering, Eight Edition*, 2007. ISBN 978-85-88639-28-7.

WILSON FILHO, P. P. **Engenharia de Software – Fundamentos, Métodos e Padrões**. Rio de Janeiro:LTC – Livros Técnicos e Científicos Editora S.A. 3^a ed., 2004.