

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**SISTEMA PARA CLÍNICA HEMODINÂMICA UTILIZANDO
PARA DESENVOLVIMENTO A FERRAMENTA ACTIVE X
“*THE BENNET ALL TEXT TECHNOLOGY*”**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

MAURO HINZ

BLUMENAU, NOVEMBRO/1999.

1999/2-29

**SISTEMA PARA CLÍNICA HEMODINÂMICA UTILIZANDO
PARA DESENVOLVIMENTO A FERRAMENTA ACTIVE X
“THE BENNET ALL TEXT TECHNOLOGY”**

MAURO HINZ

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Oscar Dalfovo — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Oscar Dalfovo — Orientador na FURB

Prof. Ricardo Guilherme Radünz

Prof. Roberto Heinzle

AGRADECIMENTO

Ao cirurgião Dr. Virgílio De Mares pelo contato e auxílio a criação e execução deste projeto.

A Administradora Terezinha Weierz (Cecci) pelas entrevistas e informações relevantes contidas, além de empréstimo de materiais sobre o tema em questão.

A secretária Joice Mara de Abreu, pela atenção e pela utilização que auxiliaram na produção de um sistema simples de utilização.

A enfermeira Claudia Burigo Zanuzzi pelas dicas que auxiliaram no aperfeiçoamento do sistema.

Ao laboratório Hospital Santa Isabel Cardiologia Intervencionista pela utilização do sistema.

Ao professor Oscar Dalfovo, pelo atendimento com visão crítica das informações que lhe foram apresentadas.

OFERECIMENTO

Aos meus pais, pelo exemplo de estímulo que sempre demonstraram ter em todas as circunstâncias da vida.

À minha namorada Roberta Tomasi Pires, pelo incentivo e paciência que tem para comigo.

À minha irmã Gisele Hinz, pelo auxílio prestado.

SUMÁRIO

AGRADECIMENTO	iii
OFERECIMENTO	iv
SUMÁRIO.....	v
LISTA DE FIGURAS	ix
LISTA DE QUADROS	xi
LISTA DE TABELAS	xii
RESUMO	xiii
ABSTRACT	xiv
1 Introdução	2
1.1 Objetivos	3
1.2 Justificativa.....	4
1.3 Sinopse	4
2 Hemodinâmica	6
2.1 O Que é Hemodinâmica?	7
2.1.1 Alguns Exames da Hemodinâmica	7
2.1.1.1 Cateterismo Cardíaco.....	7
2.1.1.2 Coronariografia.....	10
2.1.1.3 Aortografia.....	10
2.1.2 Alguns Procedimentos da Hemodinâmica	10
2.1.2.1 Angioplastia com Implante de “STENT” Coronário ou Periférico	10
2.1.2.2 Aterectomia Rotacional ou Rotablator	10
2.1.2.3 Valvoplastia	10
2.1.2.4 Pacientes	10

2.2 Vantagens em Utilizar um Software na Hemodinâmica	11
3 Ferramentas e Tecnologias Utilizadas	12
3.1 Visual Studio 6	12
3.1.1 Visual C++ 6	12
3.1.2 Visual J++ 6	13
3.1.3 Visual InterDev 6	13
3.1.4 Visual FoxPro 6	13
3.1.5 Visual Basic 6	13
3.2 Banco de Dados	14
3.2.1 Visual Basic e Access	16
3.2.2 Tabelas	18
3.2.3 Data Control	20
3.2.3.1 Funcionamento e Utilização	20
3.2.3.2 Definição de Propriedades	21
3.2.3.3 Vantagens	22
3.2.3.4 Desvantagens	22
3.2.3.5 Eventos	23
3.2.4 Linguagem SQL	24
3.2.4.1 Instrução “Select”	25
3.3 Seagate Crystal Reports.....	26
3.3.1 Projeto em Crystal Reports	27
3.3.1.1 Relatórios com o Crystal Reports	27
3.3.1.2 Criando um Relatório	27
3.3.1.3 Agrupando e ordenando registros.....	30
3.3.1.4 Fórmulas.....	30

3.3.1.5 Imprimindo o relatório a partir da aplicação no Visual Basic	32
4 All-Text Bennet Tecnology	33
4.1.1 Detalhes Técnicos do componente “Bennet ALL Text”	34
4.1.2 Quadro de Características	43
4.2 Tecnologia Active X	43
5 Princípios de Análise de Sistema.....	45
5.1 Análise Estruturada	45
5.1.1 Características da Análise Estruturada.....	46
5.1.2 Objetivos da Análise Estruturada.....	47
5.1.3 Ferramentas da Análise Estruturada.....	47
5.1.3.1 Diagrama de Fluxo de Dados (DFD).....	47
5.1.3.1.1 Fluxo de Dados	48
5.1.3.1.2 Processo	49
5.1.3.1.3 Depósito de Dados	49
5.1.3.1.4 Ponto Terminal (Entidade Externa)	50
5.1.3.2 Dicionário de Dados (DD).....	50
5.1.3.3 MER (MODELO Entidade e Relacionamento).....	50
5.1.4 Ciclo de Vida Estruturado	51
5.1.4.1 Levantamento	53
5.1.4.2 Análise do Sistema	54
5.1.4.3 Projeto do sistema.....	54
5.1.4.4 Implementação.....	54
5.1.4.5 Geração do Teste de Aceitação.....	55
5.1.4.6 Atividades Finais	55
6 Desenvolvimento do Sistema.....	56

6.1 Levantamento doS Dados.....	56
6.1.1 Análise.....	57
6.1.2 Lista de Eventos	57
6.1.3 Diagrama de Contexto.....	58
6.1.4 MER (MODELO Entidade Relacionamento)	59
6.1.5 Dicionário de Dados.....	59
6.1.5.1 Processos.....	60
6.1.5.2 Depósitos de Dados	62
7 Implementação.....	65
7.1 Apresentação das Telas	65
8 Conclusões	81
8.1 Dificuldades.....	81
8.2 Sujestões.....	82
Referências Bibliográficas.....	83

LISTA DE FIGURAS

Figura 1. <i>Tratamento do infarto agudo através da angioplastia e implantação de prótese.</i>	7
Figura 2. <i>Controle de dados que se encontra na caixa de ferramentas.</i>	21
Figura 3. <i>Janela de propriedade do controle de dados.</i>	21
Figura 4. <i>Janela com controle de acesso a banco de dados.</i>	22
Figura 5. <i>Opção New do menu File.</i>	28
Figura 6. <i>Assistente Create Report Expert.</i>	29
Figura 7. <i>Preview Report.</i>	29
Figura 8. <i>Janela Insert Group Section.</i>	30
Figura 9. <i>Editor de Fórmulas.</i>	31
Figura 10. <i>Janela comercial de visualização do componente Active X “AllText HT/Pro”.</i>	42
Figura 11. <i>Notação de fluxo de dados.</i>	49
Figura 12. <i>Notação de processo.</i>	49
Figura 13. <i>Notação de depósito de dados.</i>	49
Figura 14. <i>Notação de entidade externa.</i>	50
Figura 15. <i>Diagrama de Contexto.</i>	58
Figura 16. <i>Modelo Entidade Relacionamento.</i>	59
Figura 17. <i>Tela de entrada do sistema.</i>	67
Figura 18. <i>Tela principal.</i>	68
Figura 19. <i>Tela de cadastro de pacientes.</i>	69
Figura 20. <i>Tela de cadastro de médicos.</i>	70
Figura 21. <i>Tela de cadastro de fornecedores.</i>	70
Figura 22. <i>Tela de cadastro de autoridades.</i>	71
Figura 23. <i>Tela de cadastro de hospitais.</i>	71

Figura 24. <i>Tela de cadastro de orçamento para o paciente.</i>	72
Figura 25. <i>Tela de impressão do cadastro de orçamento para o paciente.</i>	73
Figura 26. <i>Tela de cadastro da conta hospitalar para o paciente.</i>	73
Figura 27. <i>Tela de cadastro de materiais e medicamentos do cadastro da conta hospitalar.</i> .	73
Figura 28. <i>Tela de cadastro de materiais de alto custo do cadastro da conta hospitalar.</i>	74
Figura 29. <i>Tela de cadastro do laudo para o paciente.</i>	75
Figura 30. <i>Tela de configuração de parágrafos.</i>	75
Figura 31. <i>Tela de configuração de bordas.</i>	75
Figura 32. <i>Tela de configuração de sombra e cor.</i>	76
Figura 33. <i>Tela de edição de tipo de fonte.</i>	78
Figura 34. <i>Tela de edição de tabelas.</i>	78
Figura 35. <i>Tela de impressão do cadastro do laudo para o paciente.</i>	78
Figura 36. <i>Tela de cadastro de entidades do paciente para o cardiologista.</i>	79
Figura 37. <i>Tela de seleção do tipo de relatório do paciente.</i>	79
Figura 38. <i>Tela de impressão do tipo de relatório do paciente.</i>	79
Figura 39. <i>Tela de seleção do tipo de gráfico do paciente.</i>	80
Figura 40. <i>Tela de impressão do tipo de gráfico do paciente.</i>	80

LISTA DE QUADROS

Quadro 1. <i>Exemplo de código fonte.</i>	22
Quadro 2. <i>Exemplo de código fonte sem SQL.</i>	25
Quadro 3. <i>Exemplo de código fonte com SQL.</i>	25
Quadro 4. <i>Cláusulas e finalidades.</i>	25
Quadro 5. <i>Código fonte.</i>	32
Quadro 6. <i>Características ALLText HT/Pro.</i>	43

LISTA DE TABELAS

Tabela 1. <i>Tabela de dados cliente.</i>	19
Tabela 2. <i>Propriedades de ALLText.</i>	34
Tabela 3. <i>Funções de ALLText.</i>	40
Tabela 4. <i>Eventos de ALLText.</i>	41
Tabela 5. <i>Depósito de dados.</i>	62

RESUMO

Este TCC visa o estudo e implementação de um sistema para controle de pacientes para clínica cardíaca com o objetivo de auxiliar na geração da conta dos serviços prestados e cadastro do laudo do paciente. Para o desenvolvimento do sistema será utilizada a ferramenta *Active X “The Bennet All-Text Technology”*, juntamente com o compilador *Visual Basic* da *Microsoft* e o gerador de relatórios da *Seagate Crystal Reports*. O sistema utiliza o gerenciador de banco de dados *Microsoft Access* e interface gráfica compatível com o *Microsoft Windows*.

ABSTRACT

This TCC has as objective the study and implementation of a system for patients' control that could be used by hearts clinics. The objective of this system is to be used as an in the hacking of the bill of the services and patients' script's register. For the development of the system the tool that will be used are *Active X the Bennet All-Text Tecnology*, Visual Basic of Microsoft and the report creator of Seagate Crystal Reports. The system uses the database manager Microsoft Access and is compatible with Microsoft Windows graphics interface.

1 INTRODUÇÃO

Com a globalização através de meios de comunicação o profissional clínico se depara com um obstáculo a ser transposto, como manipular adequadamente o conjunto de informações de que necessita para atender suas atividades cotidianas. Este volume de informações tem crescido nos últimos anos em função do desenvolvimento de uma variedade de procedimentos clínicos com novos exames, e uma das grandes preocupações consiste na busca de soluções para armazenagem dos detalhes e posterior recuperação.

Na década de noventa as empresas enfrentam desafios de competitividade até então não perceptíveis. O mercado passa por uma fase de adaptação a fim de atender a demanda por produtos e serviços com novas exigências. A busca da melhor resposta do mercado, redução dos custos e aumento da qualidade e competitividade, depende da informação, de modo que a informática pode fornecer meios para se obter os resultados adequados.

As aplicações na área da informática multiplicam-se e atingem os mais diversos ramos do conhecimento, dentre os quais se destaca a área da saúde. Partindo de sistemas informatizados, a área de informática na saúde está evoluindo e especializando, abrangendo desde o processamento de imagens, sons e sinais até os sistemas de informação, contribuindo para informações confiáveis e relevantes, auxiliando o profissional clínico na tomada de decisões estratégicas.

Inicialmente em 1992, de acordo com [TOM95], um sistema para gerenciamento de uma clínica necessitava apenas de cadastro de pacientes, relatório informando o endereço, telefone, hora de atendimento, agenda para marcar as consultas, entre outros.

À medida que os sistemas foram evoluindo houve necessidade de controlar melhor a clínica e principalmente ao custo dos pacientes, já que as variáveis eram muitas e a criação da conta do paciente tomava muito tempo e tornava-se muito complexa.

Tal complexidade justifica-se pelo fato que o número de planos de saúde estar crescendo periodicamente e que cada plano possui suas próprias regras de cobrança.

O custo do procedimento ou intervenção cirúrgica modifica dependendo:

- a) do plano de saúde do paciente;
- b) materiais de alto custo importados utilizados na intervenção cirúrgicas;
- c) materiais e medicamentos utilizados durante o procedimento clínico;

- d) custo dos profissionais envolvidos na intervenção cirúrgica;
- e) do número de auxiliares utilizados;
- f) número de acompanhantes na sala de internação;
- g) se o plano cobre ou não prótese coronária.

Outra necessidade é o cadastro do laudo do paciente. Este laudo apresenta de forma descritiva os procedimentos realizados no paciente.

Conforme descrito acima, pretende-se com este protótipo que será implantado no Hospital Santa Isabel Intervencionista, os seguintes benefícios:

- a) agilidade no processo de consulta de informações do paciente;
- b) cadastro e localização do laudo do paciente;
- c) emissão de relatórios específicos;
- d) geração de gráficos estatísticos;
- e) geração da conta do paciente.

Para melhor apresentação do laudo do paciente será utilizado neste sistema um controle *OCX* (de *Ole Control eXtension*) da Empresa *Bennet-Tec* [BEN97].

Um controle *OCX* é um trecho de programa que foi escrito para uma especificação muito restrita. Por causa da rigidez de sua especificação, um controle *OCX* pode ser introduzido no interior de outros programas sem a perda da performance ou instabilidade [CHA97] [BEN97].

Para o desenvolvimento do sistema será utilizada a Análise Estruturada como a metodologia de desenvolvimento do sistema.

Será utilizados para implementação o Microsoft Visual Basic 6 que é uma ferramenta para programação, Microsoft Access que é um gerenciador de Banco de Dados, Seagate Crystal Reports 6 que é uma ferramenta Active X com controle OCX para auxiliar na geração de relatórios complexos e *The Bennet All-Text Control* que é uma ferramenta Active X com controle OCX utilizado para auxiliar com funções estendidas de formatação de texto.

1.1 OBJETIVOS

O objetivo do trabalho é a construção de um sistema para controle de informação para uma clínica hemodinâmica. Tal sistema deve possibilitar o agendamento de consultas, a

manutenção e atualização das fichas clínicas dos pacientes, controle administrativo e financeiro da clínica e emissão de relatórios das consultas realizadas.

1.2 JUSTIFICATIVA

A necessidade de automação de clínicas hemodinâmicas surgiu da preocupação natural quanto aos problemas causados pelo manuseio das fichas dos pacientes e os laudos mantidos em arquivo, e a dificuldade de atualização destas fichas.

Em clínicas hemodinâmicas não informatizadas percebe-se fatores que contribuem para que este se torne moroso, dentre os quais pode-se destacar, a dificuldade de manuseio de fichas, ilegibilidade, perda ou duplicidade de informações, falta de padronização, inconsistência. Tabular dado para a elaboração do laudo, por exemplo, exige um esforço relativamente grande, pois se torna necessário o manuseio longo e tedioso dos fichários, que necessita de grande espaço físico para armazenagem sem que ocorra a deterioração dos prontuários arquivados.

O processamento eletrônico de dados oferece uma solução racional para os problemas acima.

1.3 SINOPSE

No capítulo 1 são descritos informações do trabalho, introdução, objetivos, justificativas e a estrutura do trabalho.

No capítulo 2 são conceitualizados, descritos as características da hemodinâmica com as vantagens de utilizar um software nesta área.

No capítulo 3 são conceitualizados, descritos as características das ferramentas utilizadas para desenvolvimento do sistema.

No capítulo 4 é conceitualizado, descrito as características da ferramenta *Active X “The All-Text Bennet Technology”*.

No capítulo 5 aborda a análise estruturada suas características e definições.

No capítulo 6 é apresentada a metodologia para desenvolvimento do sistema utilizado para desenvolvimento do trabalho.

Os resultados obtidos na construção do sistema para controle de clínica hemodinâmica são apresentados no sétimo capítulo. Sendo apresentadas algumas telas geradas pelo sistema quando da execução do sistema.

O último capítulo apresenta as conclusões, propondo melhoramentos futuros.

2 HEMODINÂMICA

O histórico da hemodinâmica teve o início à mais de meio século. A aplicabilidade inicial mostrou-se viável e ganhou confiança nos anos que se seguiram. Nas duas últimas décadas, a contribuição deste método foi decisiva na ampliação dos conhecimentos sobre o comportamento normal e patológico do sistema cardiovascular, permitindo um melhor manejo das afecções cardíacas, com base em informações, atualmente o serviço de hemodinâmica vem se expandindo, beneficiando tanto o exame como a intervenção cirúrgica.

O Hospital Santa Isabel Cardiologia Intervencionista (SICI) é um laboratório de estudos hemodinâmicos. Especializada no diagnóstico e tratamento das doenças cardiovasculares, do infarto agudo do miocárdio e das suas conseqüências, instalado no Hospital Santa Isabel da Sociedade Divina Providência, na cidade de Blumenau, atende toda a população do estado de Santa Catarina. Este serviço está ligado a laboratórios de hemodinâmica do país e instituições da Cardiologia Intervencionista dos Estados Unidos e da Europa. As instalações da SICI incluem uma sala para procedimentos de hemodinâmica equipada com aparelhos eletrônicos. Após o exame, o paciente permanece em recuperação sob orientação médica com o acompanhamento de familiares.

Os exames diagnósticos realizados são:

- a) cateterismo cardíaco;
- b) coronariografia;
- c) aortografia;

Os procedimentos intervencionistas realizados são:

- a) angioplastia com implante de prótese coronário ou periférico;
- b) aterectomia rotacional;
- c) valvoplastia.

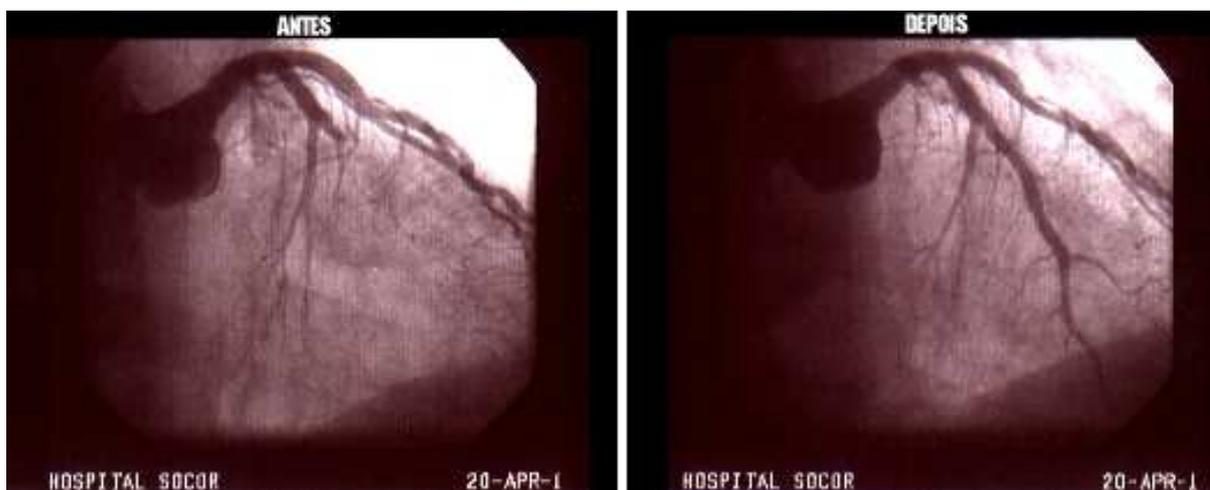
Os atendimentos são a:

- a) pacientes de convênios;
- b) pacientes particulares;
- c) pacientes pelo SUS.

2.1 O QUE É HEMODINÂMICA?

Segundo [TOM95], hemodinâmica é uma técnica de estudo por imagem dos caminhos das artérias e veias, tem como principal característica o tratamento do infarto agudo através da angioplastia e implantação de prótese conforme apresentado na figura 1.

Figura 1. Tratamento do infarto agudo através da angioplastia e implantação de prótese.



Fonte: <http://www.socor.com.br/images/hemod.JPG>

2.1.1 ALGUNS EXAMES DA HEMODINÂMICA

O Capítulo 2.1.1 foi elaborado conforme [TOM95] e entrevistas realizados com funcionários da SICI.

2.1.1.1 CATETERISMO CARDÍACO

Conforme [TOM95], o cateterismo cardíaco é um método de diagnóstico no qual se introduz um tubo comprido e fino chamado cateter em um vaso sanguíneo, para se chegar ao coração. O cateter pode ser introduzido por uma artéria ou uma veia, de acordo com a informação que se precisa obter, a partir da perna ou do braço. Para determinar se existem bloqueios ou estreitamentos nas artérias coronárias (as artérias do coração) e também para verificar com precisão como estão funcionando as válvulas cardíacas e o músculo cardíaco, injeta-se contraste através do cateter. Assim, de acordo com as necessidades particulares do paciente, podem ser realizados vários procedimentos especializados.

Este procedimento é necessário porque apesar do exame físico, eletrocardiograma, prova de esforço e do ecocardiograma fornecerem consideráveis informações sobre a função cardíaca, o cateterismo cardíaco ainda é de fundamental importância para a visualização das artérias coronárias e determinar com exatidão o grau de obstrução nestes vasos.

Todos os procedimentos anteriormente mencionados são utilizados em conjunto para se obter a maior quantidade de informação possível, com o propósito de conseguir um diagnóstico exato e, assim, decidir qual o tratamento mais adequado.

O procedimento é realizado da seguinte forma, o paciente é levado ao serviço de hemodinâmica. Ao chegar lá, as enfermeiras revisarão os seus dados e verificarão os seus sinais vitais.

Com a ajuda dos técnicos e das enfermeiras, o paciente será transferido para a mesa de cateterismo. A seguir, rasparão cuidadosamente a sua virilha ou o seu braço. A região raspada será pintada com uma solução anti-séptica. O paciente será envolvido em lençóis estéreis. Enquanto a equipe se prepara, o cardiologista injetará um anestésico local onde será introduzido o cateter.

Se o lugar da introdução for a virilha, se fará um pequeno corte de menos de meio centímetro na pele, com um bisturi muito afiado. Será introduzido um filamento flexível (fio-guia) no vaso sanguíneo, que se faz chegar ao coração com o auxílio da aparelhagem de raio X. O cateter é passado sobre o fio-guia. Se o lugar da introdução for o braço, geralmente se fará uma pequena incisão no vaso, pôr onde o cateter será introduzido.

O cateter é colocado em posição com a ajuda de uma televisão e leituras diretas da pressão arterial. Durante o procedimento, provavelmente se pedirá que o paciente respire fundo ou que tussa várias vezes. Se precisar tossir em outro momento do exame, comunique o médico. Depois, injeta-se um meio de contraste nos vasos sanguíneos e nas câmaras do coração para que possa ser visto pôr fluoroscopia e para que se obtenham imagens radiográficas permanentes. Ao se injetar o contraste nas artérias coronárias, o paciente pode sentir uma sensação dolorosa no peito. Se isso acontecer, avise imediatamente o médico. Quando o contraste for injetado nas câmaras de bombeamento do coração, o paciente pode sentir uma sensação de calor durante 15 a 20 segundos. Atualmente existem contrastes mais

modernos que provocam menor incidência de sensações desagradáveis. As imagens radiográficas são gravadas em filme para que se tenha um registro permanente e dinâmico, e posteriormente se faça um diagnóstico definitivo.

O tempo total necessário para o exame varia entre 30 e 60 minutos, desde o preparo até o final. Hoje em dia, com o uso das técnicas hemodinamicistas, equipamentos e contrastes disponíveis, o cateterismo cardíaco é um procedimento bastante seguro. Entretanto, como se realiza a punção ou dissecação de um vaso sanguíneo importante, o sangramento limita-se a ponto da punção ou dissecação e podem ser produzidas manchas escuras (hematomas) que desaparecem rapidamente. Algumas pessoas têm um sangramento maior, podendo aparecer uma inchação pôr derrame que pode incomodar pôr alguns dias, mas que, em geral, desaparece em três dias.

Uma vez fechado o local, o paciente será levado para o seu quarto, se o procedimento foi realizado através do braço, será colocado um curativo sobre a incisão. Em geral, permitirão que se levante da cama para sentar, e pouco tempo depois de chegar ao seu quarto, o paciente poderá usar o banheiro.

Quando voltar ao seu quarto, a pressão arterial e o pulso de seu braço ou de sua perna serão verificados e registrados com frequência. Recomendarão que o paciente beba líquidos para eliminar o meio de contraste que foi injetado. O curativo será freqüentemente revisado para se assegurar que não houve sangramento. Qualquer dor, sangramento ou incômodo no local da punção ou em qualquer outra parte do braço ou da perna deve ser imediatamente informado à enfermeira.

Os dados gravados no filme devem ser considerados preliminares, pois são principalmente umas orientações para o médico que realiza o cateterismo. A revelação e o processamento das radiografias levam cerca de 30 a 60 minutos. A avaliação final dos resultados requer tempo e pode ser necessário consultar o seu médico particular. Muitas vezes os resultados preliminares são discutidos no final do procedimento, e o diagnóstico definitivo e os planos de tratamento são obtidos de 12 a 24 horas mais tarde. Outros médicos preferem discutir os resultados definitivos no consultório, depois da alta do hospital.

2.1.1.2 CORONARIOGRAFIA

Coronariografia é o exame por cateterismo das artérias coronárias (irrigam o coração).

2.1.1.3 AORTOGRAFIA

Além das cavidades e artérias do coração, estuda-se também a artéria aorta, a qual é a grande artéria que recebe todo sangue que sai do coração, distribuindo para todo o organismo.

2.1.2 ALGUNS PROCEDIMENTOS DA HEMODINÂMICA

2.1.2.1 ANGIOPLASTIA COM IMPLANTE DE “STENT” CORONÁRIO OU PERIFÉRICO

É uma técnica que utiliza um minúsculo balão inflado dentro da artéria obstruída com placas de gordura e sangue, além de uma minitela de aço que, aberta, facilita a passagem do sangue. O procedimento é usado desde 1983 nos EUA e chegou ao Brasil na década atual. Agora, os pacientes também recebem, durante a operação, uma substância que impede o reinfarto.

2.1.2.2 ATERECTOMIA ROTACIONAL OU ROTABLATOR

Trata-se de um procedimento semelhante ao descrito para a angioplastia coronária "convencional", exceto pelo fato de se utilizar um sistema rotativo para desobstruir a artéria coronária.

2.1.2.3 VALVOPLASTIA

Valvoplastia é o diagnóstico ecocardiográfico da estenose mitral.

2.1.2.4 PACIENTES

- a) conveniados, são os pacientes que possuem qualquer tipo de plano de saúde que cobre os procedimentos realizados pela clínica de hemodinâmica;
- b) particulares, são os pacientes que não possuem qualquer tipo de plano de saúde que cobre os procedimentos realizados pela clínica de hemodinâmica;
- c) sus (Sistema único de Saúde), são os pacientes que possuem planos de saúde conveniados ao SUS (Sistema Único de Saúde).

2.2 VANTAGENS EM UTILIZAR UM SOFTWARE NA HEMODINÂMICA

As vantagens em ter um sistema para uma clínica de hemodinâmica são a agilidade e a velocidade com que traz as informações do paciente, emite relatórios, gera gráficos e mostra as informações cadastradas. É importante possuir informações do:

- a) paciente, é a pessoa atendida pelo SICI, é necessário informações para cadastro do paciente;
- b) médico, é pessoa que age com a intervenção cirúrgica ao paciente, este médico pode ser de outra clínica;
- c) hospital, é necessário cadastrar informação de vários hospitais, principalmente a pessoa de contato;
- d) fornecedor, é necessário cadastrar informação de vários fornecedores, principalmente a pessoa de contato, para encontrar bom preço e qualidade;
- e) convênios, é necessário cadastrar informação do convênio, principalmente as características e tabela de preços, para encontrar as limitações de cada plano e gerar a conta;
- f) laudo, após o procedimento é gerado um laudo que informa detalhes médicos do procedimento realizado;
- g) custo, é necessário gerar o custo do paciente de forma clara e detalhada;
- h) autoridade, é necessário cadastrar informação de várias autoridades, pois, sendo um serviço que trabalha com a vida, problemas de ordem social podem surgir.

Com estas informações organizadas o sistema poderá emitir relatórios, estatísticas e etiquetas para mala-direta.

3 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Para implementação deste trabalho será utilizada a ferramenta *Microsoft® Visual Basic 6*, que é uma ferramenta de programação e compilação integrante do produto *Microsoft® Visual Studio 6* criado para desenvolvedores de sistemas informatizados.

Para cadastro dos dados do sistema será utilizado o gerenciador de banco de dados *Microsoft® ACCESS 97*, que auxiliará no gerenciamento das tabelas do sistema.

Para geração dos relatórios do sistema será utilizada a ferramenta de programação *ACTIVE X, Seagate Crystal Reports 6*, que auxilia o programador a desenvolver relatórios para o sistema sem a necessidade de recompilação do código fonte.

Para geração do laudo do paciente no sistema será utilizada a ferramenta de programação *ACTIVE X, "THE BENNET ALL TEXT TECHNOLOGY"*, que auxilia o programador a desenvolver editor de hipertexto para um sistema de informações, este hipertexto será gravado em um campo tipo *MEMO* em uma tabela do gerenciador de banco de dados *Microsoft® ACCESS 97*.

3.1 VISUAL STUDIO 6

Segundo [MIC99b] o *Microsoft® Visual Studio 6* é um conjunto de ferramentas da *Microsoft®* que auxilia o programador no desenvolvimento de sistemas. Segue as linguagens de programação e seu foco:

3.1.1 VISUAL C++ 6

A linguagem C++ é conhecida por auxiliar outras linguagens de programação. Na versão 6, o foco da equipe de desenvolvimento do produto foi a codificação da programação. Recursos como *IntelliSense*, que completa o código do desenvolvedor, e o *Edit and Continue*, que permite alterar valores de variáveis durante a depuração e continuar executando o aplicativo sem a necessidade de recompilação.

3.1.2 VISUAL J++ 6

A proposta do *Microsoft® Visual J++* é oferecer um ambiente de programação para aqueles que desejam trabalhar com a linguagem *Java*. A *WFC (Microsoft Windows® Foundations Classes)* é uma biblioteca de classes que oferece acesso e facilidades de programação aos recursos do ambiente *Microsoft Windows®*.

3.1.3 VISUAL INTERDEV 6

O *Microsoft® Visual InterDev*, oferece um conjunto de ferramentas visuais para os desenvolvedores que precisam criar rapidamente páginas *web*, com acesso a bases de dados diversas. O *InterDev* oferece ainda a possibilidade de utilizar componentes *COM* (Objeto de comunicação a Internet) criados com outras linguagens, como *Microsoft® Visual Basic* ou *Microsoft® Visual J++*, dentro das páginas *web*.

3.1.4 VISUAL FOXPRO 6

O *Microsoft® Visual FoxPro* tem por objetivo auxiliar programadores em *Clipper* e *dBASE* para o ambiente *Microsoft Windows® 32 bits*, oferecendo todos os recursos disponíveis neste ambiente, numa linguagem familiar a esses programadores. Problemas sérios como os erros do ano 2000 foram abordados pela equipe de desenvolvimento e o *Microsoft® Visual FoxPro 6* oferece alguns assistentes que ajudam na migração de aplicativos antigos, que terão problemas no ano 2000.

3.1.5 VISUAL BASIC 6

O *Microsoft® Visual Basic* é uma linguagem de desenvolvimento de *softwares*. Os novos drivers *OLEDB* permitem que o *Microsoft® Visual Basic* tenha acesso a dados. As ferramentas de bancos de dados lhe garantem ferramentas de programação de bancos de dados, também é muito utilizado para tarefas de programação baseadas em *COM* (Objeto de comunicação a Internet). A criação de controles *ActiveX* (Controles especiais para auxiliar o programador) e de componentes do *Microsoft® Transaction* é simplificada pelo suporte do depurador e pelo modelo de objeto nativo *COM* (Objeto de comunicação a Internet).

Segundo [MIC99a], tanto o *Delphi* quanto o *Microsoft® Visual Basic* têm compiladores de modo nativo que geram códigos comparáveis. O código semelhante compilado com o *Delphi* é executado um pouco mais rápido do que o código compilado com o *Microsoft® Visual Basic*, mas o componente entregue com o *Microsoft® Visual Basic* (vários codificados em *assembler e C++*) são freqüentemente mais rápidos do que o componente entregue com o *Delphi*. Os componentes de acesso a banco de dados *OLEDB* baseados no *Microsoft® Visual Basic* são mais rápidos do que os componentes de acesso a banco de dados do *Delphi* baseados em seu mecanismo de banco de dados da *Borland*.

3.2 BANCO DE DADOS

Segundo [HIN97], descreve banco de dados como qualquer sistema que reúna e mantenha organizada uma série de informações relacionadas a um determinado assunto em uma determinada ordem. A lista telefônica é um exemplo, nela percebo que todos os dados referentes a uma pessoa estão na mesma linha, a isso é chamado registro. O tipo ou categoria da informação (nome, telefone, etc.) sobre uma pessoa está separada em colunas, o qual é chamado de campos.

Segundo [DAT91], banco de dados consiste basicamente em um sistema de manutenção de informações por computador, ou seja, um sistema cujo objetivo principal é manter as informações e torna-las disponíveis aos seus usuários quando solicitadas. Trata-se de qualquer informação considerada como significativa ao usuário ou a organização servida pelo sistema. Em outras palavras seria toda informação necessária ao processo de tomada de decisão do usuário ou organização.

Um sistema de banco de dados é formado basicamente pelos componentes, [DAT91]:

- a) dados: é a totalidade dos dados armazenados em um banco de dados. Uma série de arquivos agrupados que parecem para o usuário como se fosse um único arquivo de dados. Estes dados são integrados e podem ser compartilhados entre diversos usuários de várias formas distintas ou de uso exclusivo de um único usuário;
- b) hardware: consiste na máquina ou plataforma onde reside o banco de dados. É considerado um fator determinante em questões que envolvam o compartilhamento

do banco de dados entre vários usuários ou se apenas será utilizado por um único usuário

- c) software: é o elo de ligação entre o banco de dados físico e os usuários do sistema. Todas as solicitações de acesso ao banco de dados feitos pelos usuários são manipuladas por algum software, que no caso é comumente conhecido como Sistema Gerenciador de Banco de Dados (SGBD), nome que provem do inglês *DataBase Management System (DBMS)*. Sua função é isolar os usuários do banco de dados dos detalhes ao nível de hardware, ou seja, fazer com que os usuários tenham uma visão do banco de dados acima do nível de hardware, suportando suas operações de manipulação dos dados como inserções, exclusões, alterações e consultas a um nível mais elevado. Segundo [SAL93], é comum o Sistema Gerenciador de Banco de Dados e o aplicativo de banco de dados residirem e serem executados no mesmo computador. Entretanto, muita atenção tem sido dispensada atualmente para o estágio mais recente da evolução tecnológica do Sistema Gerenciador de Banco de Dados, que é a tecnologia de banco de dados Cliente/Servidor;
- d) usuários: considera-se usuários o conjunto de pessoas que poderão acessar as informações armazenadas em um banco de dados. Segundo [DAT91], existem três grandes classes de usuários:
- os programadores de aplicações, que seriam responsáveis pela definição dos programas de aplicação que utilizam alguma linguagem de programação de aplicativos de banco de dados;
 - os usuários finais, que interagem com o sistema a partir de um terminal *on-line*, utilizando alguma aplicação ou linguagem de consulta;
 - e o administrador de banco de dados (DBA), que seria uma pessoa ou um grupo de pessoas identificáveis que detenham a responsabilidade central sobre os dados operacionais armazenados no banco. Estes utilizam ferramentas de administração e linguagens de consulta (SQL) para consultar o dicionário de dados.

Outras características relevantes a sistemas de banco de dados, segundo [DAT91], são:

- a) a permissão da redução considerável da redundância de informações, pois uma certa informação que é utilizada em dois lugares diferentes pode ser apenas armazenada em um único lugar e acessada por ambos, eliminando a redundância;
- b) a inconsistência dos dados pode ser evitada (até certo ponto) levando-se em consideração que os dispositivos de consistência sejam conhecidos pelo SGBD. Este item é uma consequência natural do item anterior;
- a) o compartilhamento dos dados pode ser feito entre aplicações já existentes, como também novas aplicações podem ser desenvolvidas para operar dados já armazenados;
- b) a possibilidade de aplicação de restrições de segurança sobre o acesso ao banco de dados, onde o acesso as informações e feito através de certos canais definidos pelo administrador do banco de dados, o qual indica quais controles de segurança devem ser adotados em determinados casos;
- c) manter a integridade dos dados, assegurando que os mesmos sejam corretos, sem inconsistência e redundância no armazenamento;
- d) a independência dos dados entre aplicações, ou seja, as aplicações não dependem de qualquer estrutura de armazenamento (como os dados são fisicamente armazenados) ou de estratégia de acesso (como são lidos e gravados).

3.2.1 VISUAL BASIC E ACCESS

Um Sistema Gerenciador de Banco de Dados relacionais (SGBDR) é usado para armazenar as informações de uma forma que permita às pessoas examiná-las de diversas maneiras. O Gerenciador relacional de bancos de dados do *Microsoft® Visual Basic* e do *Microsoft® Access* é o *Microsoft® Jet*, ele pertence a uma categoria diferente dos Gerenciadores tradicionais, como o *Dbase* e o *Paradox*, pois possui características em comum com os banco de dados cliente/servidor. Tais características comuns são:

- a) todas as tabelas, índices, consultas, relatórios e código são armazenados num único arquivo *.MDB*;
- b) campos de data suportam informação de hora;

- c) armazenamento permanente de comandos *SQL*;
- d) é possível forçar a integridade referencial entre as tabelas;
- e) campos suportam valores nulos (*Null*).

No *Dbase/Clipper* banco de dados significa um arquivo que contém a estrutura de dados (campos) e os dados (Arquivo padrão *DBF*). Para o padrão *MDB* este conjunto de dados e sua estrutura denominam-se Tabela. Portanto aquilo que o *Dbase/Clipper* considera um banco de dados o *Microsoft® Access* e o *Microsoft® Visual Basic* considera como uma Tabela.

Para o *Microsoft® Access* e o *Microsoft® Visual Basic* todos os componentes do sistema estão em um único arquivo com extensão *MDB*, a este “pacote” considero de banco de dados. Logo quando abrimos um arquivo *MDB* temos acesso a todos os componentes do sistema, como tabelas, consultas, macros, relatórios, etc. A esses componentes chamo objetos do sistema e descrevo a seguir:

- a) tabelas onde se armazena as informações que queremos tratar;
- b) consultas que filtram as informações das tabelas e permitem sua visualização;
- c) formulários que são janelas destinadas à edição e visualização dos dados;
- d) relatórios que organizam os dados de tabelas e consultas de uma maneira que possam ser impressos;
- e) macros que são rotinas que automatizam determinadas tarefas sem necessidade de programação;
- f) módulos que armazenam instruções e comandos da linguagem *Access Basic/VBA* e permitem melhorar e expandir os recursos do sistema.

Embora o *Microsoft® Visual Basic* utilize arquivos padrão *MDB*, formulários, relatórios e módulos são tratados de forma diferente pelo próprio *Microsoft® Visual Basic* e, nativamente, o *Microsoft® Visual Basic* não utiliza Macros. Além disso, no *Microsoft® Access* e *Microsoft® Visual Basic* é possível utilizar outros arquivos além dos arquivos *MDB*; como arquivos *DBF* do *Dbase/Clipper*, arquivos do *Paradox*, do *Btrieve*, etc.

Os recursos de definição de dados do mecanismo *Jet* permitem a criação, a modificação e a exclusão de tabelas, índices e consultas. O *Jet* também aceita a validação de

dados em nível de campo e registro. A integridade de dados tem suporte sob a forma de chaves primárias e integridade referencial entre tabelas.

Para manipulação de dados, o *Jet* admite o uso da *SQL* e de objetos de acesso aos dados. Esses objetos permitem ao programador manipular informações contidas no banco de dados, através da definição das propriedades dos objetos e pela execução dos métodos associados aos objetos. A tabela abaixo relaciona esses objetos e descreve resumidamente suas funções:

- a) *dbengine* é um objeto que referencia o mecanismo de bancos de dados do *Microsoft® Jet* ;
- b) *workspace* é uma área na qual o usuário pode trabalhar com os bancos de dados;
- c) *database* é uma coleção de informações organizadas em tabelas, juntamente com informações a respeito de índices e relações sobre as tabelas;
- d) *tabledef* é uma definição da estrutura física de uma tabela de dados;
- e) *querydef* é uma consulta armazenada de *SQL* das informações contidas no banco de dados;
- f) *recordset* é uma coleção de registros de informações sobre um único tópico;
- g) *field* é uma única unidade de informações em um banco de dados;
- h) *index* é uma lista ordenada de registros em um *recordset*, baseada em um campo chave definido;
- i) *relation* é a informações armazenadas a respeito do relacionamento entre duas tabelas.

3.2.2 TABELAS

As tabelas são o coração dos bancos de dados, e, uma das tarefas fundamentais que o programador deverá fazer será organizar os dados em tabelas. Para criar uma tabela no *Microsoft® Visual Basic* o programador pode usar o *Data Manager*, mas, uma maneira mais fácil, se o programador tiver o *Microsoft® Access*, será utilizar os seus recursos para este fim.

Basicamente o programador irá dar nome aos campos, e definir o tipo de dados para estes campos. Isto depende da informação que deseja armazenar no campo, se for armazenar um dado que não fará parte de cálculos, um nome por exemplo, o tipo pode ser texto, se for

armazenar valores numéricos que farão parte de cálculos o tipo será numérico, se for armazenar datas o tipo será data e assim por diante.

O tamanho do campo define a quantidade de informação relacionada ao item que o programador pode armazenar, assim para um campo do tipo texto com tamanho 30 o programador poderá armazenar textos com no máximo 30 caracteres, um campo numérico de tamanho 2 armazena números com até dois dígitos.

Veja na tabela 1 a estrutura de uma tabela cliente.

Tabela 1. Tabela de dados cliente.

Nome do Campo	Tipo do Dado	Tamanho
Código	Numérico	3
Nome	Texto	40
Endereço	Texto	40
Cep	Texto	8
Idade	Numérico	3

Fonte. [HIN97].

Embora não exista nenhuma regra absoluta para os dados que devem ser colocados em cada tabela, abaixo segue as diretrizes gerais para um projeto de banco de dados eficiente:

- a) determine um tópico para cada tabela e certifique-se de que todos os dados contidos na tabela estão relacionados com o tópico.
- b) se uma série de registros em uma tabela apresenta campos intencionalmente deixados em branco, divida a tabela em duas tabelas similares.
- c) se as informações se repetem em vários registros, desloque essas informações para outra tabela e defina um relacionamento entre elas.
- d) campos repetidos indicam a necessidade de uma tabela secundária.
- e) use tabelas de pesquisa para reduzir o volume de dados e aumentar a precisão da entrada de dados.
- f) não armazene informações em uma tabela se elas puderem ser calculadas a partir dos dados contidos em outras tabelas.
- g) para cada linha que o programador adiciona a sua tabela o programador tem um registro. Feito isto é importante o programador definir os relacionamentos entre as tabelas de seu banco de dados. Por exemplo, se o programador criou duas tabelas, uma para o cliente e outra para o pedido feito por esse cliente, o programador

poderá relacionar as duas tabelas por meio de um campo código do cliente comum às duas tabelas.

3.2.3 DATA CONTROL

O *Data Control* foi criado para proporcionar um meio simples de acesso a um banco de dados. Para usar o controle de dados, siga os passos a seguir:

- a) selecione o controle na caixa de ferramentas;
- b) desenhe o controle em seu formulário;
- c) defina a propriedade *DatabaseName*;
- d) defina a propriedade *Recordsource*;
- e) defina a propriedade *RecordsetType*;

A essa altura o programador criou os vínculos para o banco de dados e o *Recordset* com o qual deseja trabalhar. O controle assume para si as tarefas de:

- a) abrir o banco de dados e as tabelas ou consultas associadas;
- b) navegar pelos registros. (Utiliza métodos *MoveFirst*, *Movelast*, *Movenext* e *Moveprevious*) ;
- c) carregar os controles de dados, exibir e editar os dados;
- d) gravar os dados editados dos controles no *Recordset*.

3.2.3.1 FUNCIONAMENTO E UTILIZAÇÃO

Com base nas propriedades *Name* e *RecordSource* o controle de dados cria um *Recordset* que fornece acesso aos seus dados.

De início selecione o objeto controle de dados na *Toolbox* do *Microsoft® Visual Basic* e acrescente-o ao seu formulário, conforme figura 2.

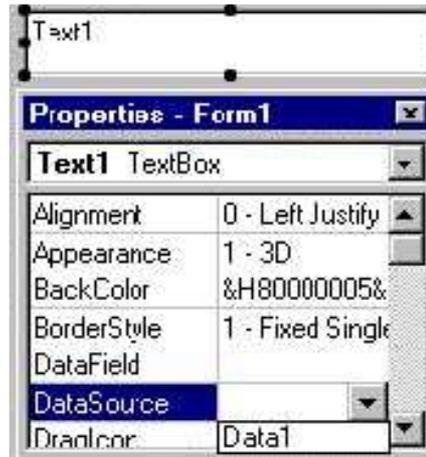
Figura 2. Controle de dados que se encontra na caixa de ferramentas.



Fonte: [MIC99c]

Defina a seguir as propriedades *Name* e *Caption* - Especifica o nome que aparece no controle de dados e dimensione o controle de dados, conforme figura 3.

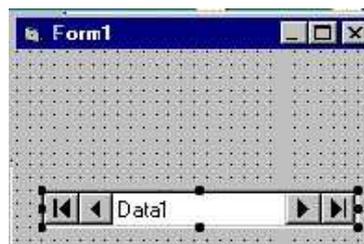
Figura 3. Janela de propriedade do controle de dados.



Fonte: [MIC99c].

A seguir acrescente ao seu formulário um quadro de texto (*TextBox*) para cada campo que desejar acessar a partir do *Recordsource*, anexe cada quadro de texto ao objeto de controle de dados (*DataSource*) e especifique o campo que cada quadro de texto deverá exibir(*DataField*) conforme figura 4.

Figura 4. Janela com controle de acesso a banco de dados.



Fonte: [MIC99c].

3.2.3.2 DEFINIÇÃO DE PROPRIEDADES

O programador deve associar um banco de dados e um *recordset* a um controle de dados definindo as propriedades do controle. As propriedades necessárias são:

- a) *datasasename* - Para bancos de dados do *Microsoft® Access*, representa o nome do arquivo do banco de dados, inclusive o nome completo do caminho. (Para banco de

dados diferentes do *Access* é necessário informar o nome do caminho até o subdiretório dos dados. *Ex: Dbase : DatabaseName=c:\dir*;

- b) *recordsource* - Especifica as informações que deseja obter no banco de dados - uma tabela, uma instrução *SQL*;
- c) *recordsettype* - Se deseja criar um *2-snapshot* ou um acessar uma *0-table* altere esta propriedade, pois o padrão é *1-dynaset*;
- d) *connect* - Necessária quando o banco de dados não for do *Access*. *Ex: dbase - connect=dBASEIII*.

É possível optar por ativar essas propriedades durante a execução do seu aplicativo. O programador pode acrescentar um controle dados a seu formulário e definir a propriedade *Name* como *Data1*, e quer ter acesso no arquivo *TESTE.MDB* (arquivo *Access*) aos dados da tabela clientes, veja quadro 1:

Quadro 1. Exemplo de código fonte.

<code>data1.DatabaseName="C:\TESTE\TESTE.MDB"</code>	'path/nome do arquivo
<code>data1.RecordSource="Clientes"</code>	'fonte dos dados(tabela clientes)
<code>data1.recordsetType=0</code>	'tipo de recordset(0-tabela)
<code>data1.Refresh</code>	'implementa as alterações

Fonte: [MIC99c].

3.2.3.3 VANTAGENS

A principal vantagem do controle de dados é a necessidade de menor trabalho de programação para desenvolver um aplicativo de acesso de dados. O programador não precisa fornecer código de programa para abrir ou criar um banco de dados ou um *recordset*, para se movimentar pelos registros ou para editar registros. Outra vantagem é o vínculo direto com os dados. O programador não precisa invocar os métodos *Edit* e *Update* para modificar os dados contidos no banco de dados. As alterações aparecem no banco de dados tão logo são digitadas. Os vários controles vinculados (*DbList*, *DbCombo*, *DbGrid*, ...) oferecem um meio fácil de realizar tarefas que via código seriam complexas.

3.2.3.4 DESVANTAGENS

As limitações do controle de dados são:

- a) não apresentam função para alterar ou excluir registros;
- b) é mais difícil a implementação do processamento de transações;

- c) um número excessivo de controles vinculados em um formulário torna a sua carga mais lenta.

3.2.3.5 EVENTOS

Utilizar como ferramenta de programação uma linguagem procedural como o *Clipper* em que o código do programa era um fluxo contínuo onde havia chamadas a outros programas e onde o usuário deveria obedecer a uma ordem pré-determinada pela aplicação, como exemplo temos que em uma tela de entrada de dados para Nome do Cliente, Endereço, CEP e idade se o usuário quisesse entrar o número do Cep, deveria passar obrigatoriamente pelos campos anteriores.

Os programas que rodam no *Microsoft® Windows* não se comportam dessa maneira. O programador constrói o seu código em torno de eventos e os eventos são determinados pela ação do usuário que podem ser:

- a) *click*, ocorre quando o usuário pressiona e libera um botão do mouse sobre um objeto. O evento *click* também pode ocorrer quando a definição da propriedade *value* de um controle é alterada. Normalmente, o programador anexa uma macro ou procedimento de evento *click* a um botão de comando para executar comandos ou ações compatíveis;
- b) *change*, ocorre quando o conteúdo de uma caixa de texto ou porção de texto de uma caixa de combinação é alterado. Exemplos deste evento incluem o fornecimento de caracteres diretamente na caixa de texto ou caixa de combinação, ou a alteração da definição da propriedade *text* de um controle usando-se uma macro ou o *Microsoft® Access Basic*. Durante a execução de uma macro ou um evento de procedimento, quando ocorrer um evento *change*, o programador pode coordenar a exibição de dados entre os controles. Poderá também exibir os dados em um formulário de um controle, e os resultados em um outro controle;
- c) *load*, ocorre quando um formulário é aberto e os registros são exibidos. O evento *load* pode ocorrer quando uma aplicação se inicia;
- d) *open*, o evento *open* ocorre quando um formulário é aberto, mas antes que o primeiro registro seja apresentado. Para os relatórios, o evento ocorre antes que um

relatório seja visualizado ou impresso. O evento *open* ocorre antes do evento *load*, que por sua vez é disparado quando um formulário é aberto.

- e) *close*, ocorre quando um formulário ou relatório é fechado e removido da tela. O evento *close* ocorre depois do evento *unload*, que é disparado depois que o formulário é fechado, mas antes que ele seja removido da tela.

Os formulários, relatórios, botões de comando, caixa de listagem, caixa de verificação, botão de opção, grupo de opção e demais objetos possuem propriedades, métodos e eventos associados que utilizo para executar rotinas apropriadas ao contexto. Além disso, eventos podem disparar outros eventos.

3.2.4 LINGUAGEM SQL

A linguagem *SQL* (*Structured Query Language*) é uma linguagem de alto nível para manipulação de dados dentro do modelo relacional. Seu objetivo é fornecer uma interface de alto nível ao usuário.

Uma instrução SQL consiste em três partes:

- a) declarações de parâmetros;
- b) instrução manipulativa;
- c) declarações de opções.

Para uma comparação, atualizando o campo valor em 10% de uma tabela com diversos registros. Na abordagem procedural, tem-se os seguintes passos a seguir, ver código fonte no quadro 2 e 3:

- a) abrir a tabela;
- b) posicionar o ponteiro no início da tabela;
- c) atualizar o campo valor em 10%;
- d) mover o ponteiro para o próximo registro.

Quadro 2. Exemplo de código fonte sem SQL.

```
Dim db as database
Dim tabela as recordset
set db=workspaces(0).Opendatabase("c:\base.mdb")
set tabela=db.Openrecordset("tabela")
While! not tabela.eof
tabela.edit
tabela.valor=tabela.valor*1.10
tabela.update
tabela.movenext
tabela.close
```

Fonte: [MIC99c].

Quadro 3. Exemplo de código fonte com SQL.

```
Dim db as Database
Set db=Workspaces(0).Opendatabase("c:\base.mdb")
db.execute "UPDATE tabela SET valor=valor*1.10"
db.close
```

Fonte: [MIC99c].

Veja em quadro 4 um resumo das cláusulas manipulativas e suas finalidades:

Quadro 4. Cláusulas e finalidades.

Instrução	Função
SELECT	Obtém um grupo de registros e insere os registros em um dynaset ou em uma tabela.
UPDATE	Define os valores dos campos de uma tabela em uma atualização.
TRANSFORM	Cria uma tabela de resumo, utilizando o conteúdo de um campo como cabeçalho de cada coluna.
DELETE FROM	Remove registros de uma tabela.
INSERT INTO	Acrescenta um grupo de registros a uma tabela.

Fonte: [HIN97].

3.2.4.1 INSTRUÇÃO “SELECT”

- seleciona os campos “Primeiro nome” e “Sobrenome” de todos os registros da tabela empregados. Ex: *SELECT [Primeiro nome], [Sobrenome] FROM Empregados;*
- seleciona todos os campos da tabela empregados. O uso parâmetro (*) indicando todos os campos da tabela indicada. Ex: *SELECT Empregados.* FROM Empregados;*
- conta o número de registros que têm uma entrada no campo “Código postal” e coloca o título contagem no topo da coluna. Ex: *SELECT Count ([Código postal]) AS Contagem FROM Clientes;*

- d) seleciona os campos “Primeiro nome” e “Sobrenome” de cada registro cujo sobrenome seja “pereira”. Ex: *SELECT [Primeiro nome], [Sobrenome] FROM Empregados WHERE [Sobrenome] = 'Pereira'*;
- e) seleciona os campos “primeiro nome” e “Sobrenome” para empregados cujos sobrenomes começam pela letra “S”. Ex: *SELECT [Primeiro nome], [Sobrenome] FROM Empregados WHERE [Sobrenome] Like 'S**.

3.3 SEAGATE CRYSTAL REPORTS

Segundo [SEA99],[SEA99a], o *Seagate Crystal Reports* é um software que auxilia o programador a desenvolver relatórios. Embora programas de bases de dados muitas vezes incluam seus próprios geradores de relatórios, estes exigem grande conhecimento da base de dados a qual estão acoplados.

Segundo [HIN97], o *Seagate Crystal Reports* é um conjunto de ferramentas que permite tipos ilimitados de análise de dados e geração de gráficos no relatório. O *Seagate Crystal Reports* possui assistentes de relatórios que auxiliam produção de relatórios informativos, com interface '*drag-and-drop*' possui a opção de editar relatórios enquanto visualiza dados, quando o relatório estiver completo o *Seagate Crystal Reports* auxilia o envio para quase qualquer destino.

O processamento no servidor lhe permite projetar relatórios que efetuam agrupamento, ordenação e expressões *SQL* no servidor de base de dados.

Características do *Seagate Crystal Reports*:

- a) controle ActiveX (OCX) (16- e 32-bit);
- b) controle Visual Basic Custom (VBX);
- c) Microsoft® Foundation Class Library com AppWizard for Visual C++;
- d) Delphi Visual Component Library (VCL);
- e) Informix-new era class library;
- f) automation server e object library;
- g) report designer component para Visual Basic 5/6.

3.3.1 PROJETO EM CRYSTAL REPORTS

O *Seagate Crystal Reports* é utilizado para gerar seus relatórios e incorporá-los a aplicação. A seguir descrevo como gerar relatórios utilizando os recursos do *Seagate Crystal Reports*, como controlar a ordem de impressão dos registros, trabalhar com variáveis fornecidas pelo usuário.

3.3.1.1 RELATÓRIOS COM O CRYSTAL REPORTS

O *Seagate Crystal Reports* é o gerador de relatórios do *Visual Basic* e utilizado para gerar relatórios no aplicativo. Embora possua um objeto chamado *Printer* para imprimir dados, sua utilização além de complexa é trabalhosa, pois tudo deve ser codificado. O *Seagate Crystal Reports* utiliza uma interface gráfica a partir de onde é possível construir relatórios.

É possível iniciar o *Seagate Crystal Reports* através da opção *Report Designer* do menu *Add-Ins* ou pelo ícone correspondente na pasta de trabalho do *Visual Basic no Microsoft Windows®*.

3.3.1.2 CRIANDO UM RELATÓRIO

Para gerar um relatório baseado na tabela agenda que se encontra no banco de dados controle você deve seguir os seguintes passos:

- a) campos a serem impressos:
 - sobrenome;
 - endereço;
 - data de nascimento.
- b) relatório ordenado pelo campo sobrenome;
- c) permitir inicialmente a visualização do relatório para posterior impressão;
- e) nome do relatório será *agenda.rpt*.

Selecionando a opção *New* do menu *File* temos a figura 5:

Figura 5. Opção New do menu File.

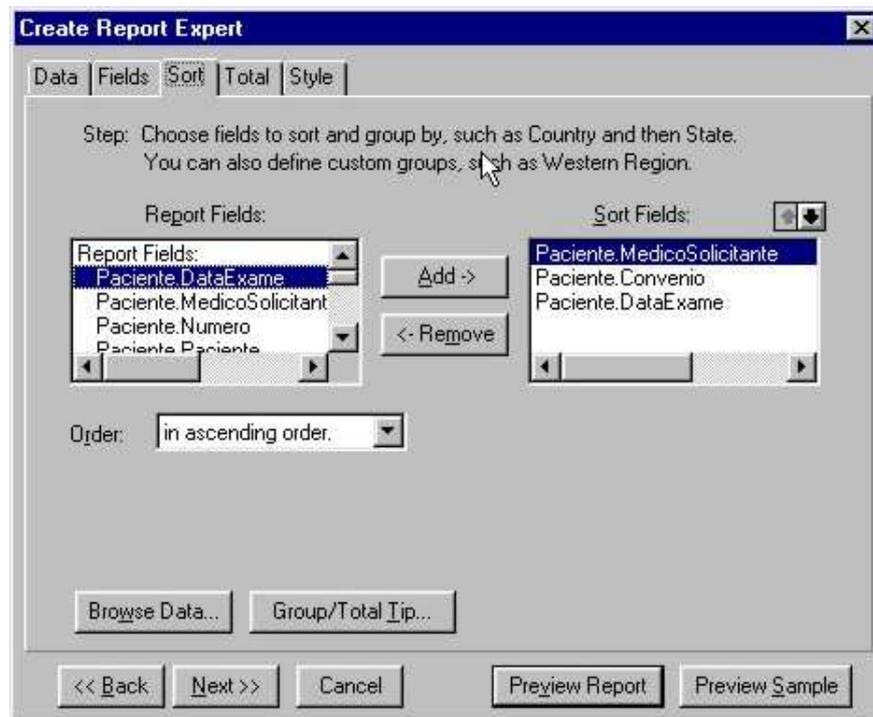


Fonte: [SEA99].

Após selecionar o botão *Standard*, selecione a base de dados na opção *Data File*. A seguir temos uma lista de todas as tabelas e consultas gravadas na base de dados e clique no botão *Next*.

Como o relatório está baseado somente na tabela *Agenda*, o próximo passo *Links*, pode ser dispensado, clique no botão *Next*. Selecione os campos da tabela que serão impressos no relatório, selecione cada campo e clique no botão *Add*, veja a figura 6:

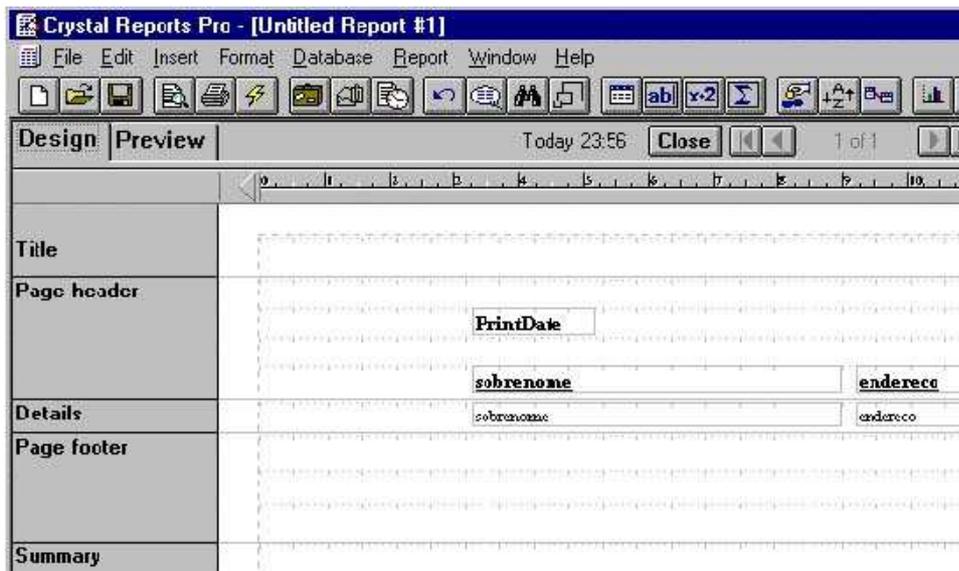
Figura 6. Assistente Create Report Expert.



Fonte: [SEA99].

Agora o relatório está praticamente terminado, para visualizá-lo clique no botão *Preview Report*.

Figura 7. Preview Report



Fonte: [SEA99].

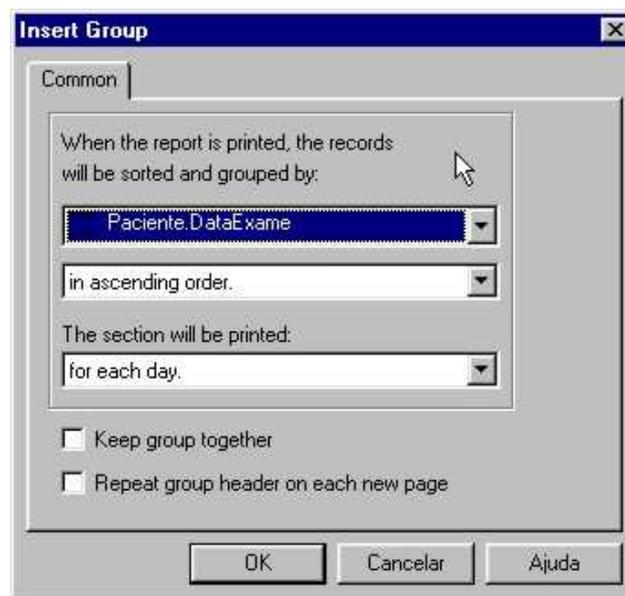
É mostrada a tela da figura 7 onde após clicar na aba *Design* há cinco seções:

- a) *title*, para o título da aplicação;
- b) *page header*, contém os elementos do cabeçalho da página;
- c) *details*, contém os campos de dados a serem impressos;
- d) *page footer*, refere-se ao rodapé da página;
- e) *summary*, impressão de resumos.

3.3.1.3 AGRUPANDO E ORDENANDO REGISTROS.

Para agrupar registros, ordenando-os por uma determinada coluna selecione a opção *Group Section* do menu *Insert*. Agrupando pelo campo *sobrenome* em ordem ascendente, veja a figura 8:

Figura 8. Janela *Insert Group Section*



Fonte: [SEA99].

3.3.1.4 FÓRMULAS.

Ao montar uma fórmula para imprimir o número da página no rodapé do relatório. Para isso uso o editor de fórmulas do *Seagate Crystal Reports* que pode ser disparado através do ícone  ou da opção *Formula Field* do menu *Insert*. Após isso deve informar o nome da fórmula no campo *Formula Name*, informe a “página” e clique no botão OK. O editor é mostrado na figura 9:

Figura 9. Editor de Fórmulas.



Fonte: [SEA99].

Agora basta digitar a fórmula na caixa *Formula Text*, digite: "Página: " + e, a seguir selecione a função *TrimLeft* na lista *Functions*, ainda na lista *Functions*, selecione a função *ToText* e finalmente selecione o item *PageNumber* no final da lista *Functions*. Ao final deveremos ter o seguinte na caixa *Formula Text*: "Página: " + *TrimLeft (ToText (PageNumber, 0))*.

A função *PageNumber* retorna um valor numérico do número da página por isso uso a função *ToText* para convertê-la em uma *string*, e a seguir uso a função *TrimLeft* para removermos os espaços a direita.

Ao verificar a fórmula clicando no botão *Check*, se tudo estiver correto o *Seagate Crystal Reports®* informa com a mensagem "No errors found" indicando que a sintaxe está correta. Agora basta clicar no botão *Accept* e posicionar a fórmula no canto esquerdo da seção *Page Footer*.

Encerrado o relatório basta salvá-lo através da opção *Save* do menu *File* e informar o nome para o relatório. Devo ressaltar que a linguagem de fórmulas do *Seagate Crystal Reports®* é diferente do *Visual Basic*, assim, por exemplo, se uso a propriedade *SelectionFormula* do *Seagate Crystal Reports®* que permite definir condições para a impressão no relatório de forma a imprimir somente os nomes iniciados pela letra "J" teros algo como: `CrystalReport1.SelectionFormula = "{AGENDA.NOME} >= " & "" & "J" & ""`.

3.3.1.5 IMPRIMINDO O RELATÓRIO A PARTIR DA APLICAÇÃO NO VISUAL BASIC.

Quando o relatório estiver pronto basta associá-lo a aplicação de forma a poder imprimir a partir do Visual Basic.

Para isso deve-se ativar o componente do *Crystal Reports* para a aplicação na opção *Components* do menu *Project* e a seguir selecionar o controle *Seagate Crystal Reports* e copiá-lo para o formulário.

É necessário definir algumas propriedades para o controle *CrystalReport1*

a) *copiestoprinter*, determina o número de cópias do relatório.

b) *destination*, direciona a impressão:

- 0-na tela;
- 1-na impressora;
- 2-em arquivo.

c) *reportfilename*, indica a localização do relatório (arquivo RPT) a ser impresso;

d) *windowtitle*, é o título da janela *Preview*;

e) *sortfields*, configura a ordem de classificação.

Crie um botão de comando no formulário que irá disparar a impressão do relatório com o nome de “Imprime” e a seguir associe o seguinte código do quadro 5 ao botão de comando:

Quadro 5. Código fonte.

```
Private Sub imprime_Click()
CrystalReport1.Destination = 0
CrystalReport1.ReportFileName = "c:\Controle\agenda.rpt"
CrystalReport1.SortFields(0) = "+{Agenda.Sobrenome}"
CrystalReport1.Action = 1
End Sub
```

Fonte: [MIC99c].

A propriedade *Action* definida para 1 dispara a impressão do relatório. Outra forma de ordenar os registros via código é utilizar a propriedade *Sortfields*. Na propriedade *Sortfields*, "+{Agenda.Sobrenome}", indica que a ordem de impressão será por campo sobrenome (Agenda.sobrenome) e em ordem ascendente (+).

4 ALL-TEXT BENNET TECHNOLOGY

Segundo [BEN97], *ALLText* é um OCX (de *Ole Control eXtension*) que substitui o *textbox* com algumas características importantes, apóia na produção de parágrafos formatados, fontes múltiplas, características de fontes, cores, corte formatado & pasta (até mesmo entre aplicações), fundos transparentes, I/O automatizado principalmente quebra a barreira dos 32K de limitação de texto.

ALLText HT/Pro é um *Rich Text Box* com mais recursos, possui um controle de caixa de texto, inclui todas as características do *ALLText* básico, controla melhor o *Hypertext*, possui marcador de páginas etiqueta, arquivo de apoio *RTF(Rich Text Format)*, Quadros Embutidos (WMF, BMP, ICO, JPEG) e Objetos de *OLE*.

Neste trabalho irei utilizar o *ALLText HT/Pro* que possui as seguintes características:

- a) edição rápida do VBX para 16 aplicações do bit;
- b) tipo de fonte:
 - bold (realce);
 - underline;
 - itálico;
 - strikethrough;
 - super;
 - subscrições.
- c) cores (sustentação para o padrão 16 QBColors) e sombrear ;
- d) formato do parágrafo (para cada parágrafo):
- e) alinhamento:
 - centrado;
 - justificado;
 - direito;
 - esquerdo/direito justificado ;
- f) margens
 - esquerdo;
 - direito.
- g) afastamento vertical: espaço antes, em seguida, e afastamento de linha;
- h) cortar e colar incluindo a preservação do formato entre aplicações;

- i) criação e uso de uma tabela facilitando a mudança lisa e rápida das tabelas;
- j) fundo bitmapped pode ser movido/posicionado em tempo de execução, grande para efeitos;

4.1.1 DETALHES TÉCNICOS DO COMPONENTE “BENNET ALL TEXT”

A versão atual do controle é 4.0, as linguagens suportadas são:

- a) Visual Basic;
- b) Visual C++;
- c) Borland C++;
- d) Delphi.

Esta é uma lista das propriedades, funções, e eventos do controle de ALLText. Características para a edição de HT/Pro são denotadas com (HT/Pro).

Tabela 2. Propriedades de ALLText.

Propriedades	Características
• Alignment:	Especifica Alinhamento de Parágrafo;
• BackColor:	Especifica a fundo cor;
• BackPicture:	Especifica um fundo atrás do texto;
• BackPictureRefresh:	Ao fixar, move o quadro de fundo para as coordenadas de X,Y;
• BackPictureX:	Especifica horizontal do quadro de fundo;
• BackPictureY:	Especifica vertical do quadro de fundo;
• BackStyle:	Especifica se ALLText é transparente ou opaco;
• Border • (HT/Pro)	Especifica o estilo de borda para um parágrafo;
• BorderStyle	Especifica a borda para o controle de ALLText;
• BottomIndent	Especifica linha que espaça depois de um parágrafo;
• CaretWidth	ALLText marcador de posição atual;
• ChangeEventMask	Especifica eventos que ativarão o Evento de Mudança;

• ClearAll	Esvazia a caixa de texto;
• ClipboardAction	Apoios para Cortar, Cópiar e operações de Pasta;
• CurChar	Especifica posição de cursor relativo começar de parágrafo atual;
• CurPar	Especifica parágrafo atual no qual cursor é localizado;
• DataChanged	Indica se conteúdo do controle ALLText mudou;
• DataField • (HT/Pro)	Campo de Banco de dados para o qual ALLText é ligado. (Para hipertexto o usuário deve ligá-lo a um campo MEMO);
• DataSource • (HT/P)	Nome do Banco de Dados;
• DataType	Especifica formato usado para arquivo ou banco de dados I/O;
• DirectScreenOut	Especifica método no qual ALLText prepara texto para tela;
• DocHeight	Especifica a altura de documento dentro do controle;
• DocWidth	Especifica a largura de envoltura de palavra de documento;
• DropFileMode	Especifica resultado de um arquivo dropping;
• Enabled	Especifica se o controle é habilitado;
• ExtDataType • (HT/Pro)	Controles que formatando códigos são usados quando administrando arquivo ou banco de dados I/O com DataType;
• ExtObjParam • (HT/Pro)	Especifica o um atributo de objeto externo a ter acesso com a propriedade de ExtObjValue;
• ExtObjPicture • (HT/Pro)	Especifica um valor para o quadro de um objeto embutido;
• ExtObjValue • (HT/Pro)	Especifica um atributo de um objeto externo embutido;
• F2ON	determina se o controle para escolher a fonte responde à F2

• F3ON	determina se o controle responde à F3 enviando o conteúdo ao controle de impressão;
• FileLoad	Inicia Arquivo I/O;
• FileName	Especifica nome de arquivo para Arquivo I/O;
• FileSave	Inicia Arquivo I/O;
• FirstLineIndent	Especifica indentation pendente para um parágrafo;
• FontBold	Especifica tipo em negrito;
• FontColor	Especifica cor da fonte;
• FontFamily	Especifica a família de fontes selecionadas;
• FontHidden • (HT/Pro)	Especifica se selecionou fonte do tipo escondida;
• FontIndex	Especifica a localização na lista indexada das fontes;
• FontItalic	Especifica tipo em Itálico;
• FontName	Especifica o nome da fonte;
• FontShadow	Especifica o estilo de sombreando para os caracteres selecionados;
• FontSize	Especifica o tamanho do caracteres selecionados;
• FontStrike	Especifica se selecionou caráter riscado;
• FontSubSup	Fixa atributos de super/subscript para texto à localização de cursor atual;
• FontTableSize	Especifica número de entradas no estilo da fonte;
• FontUnder	Especifica estilo sublinhado para caráter selecionado;
• FontWidth	Especifica largura da fonte;
• FooterFText • (HT/Pro)	Contém texto de footer formatado para o documento;
• FormatPaste	Determina se a cópia e funções de pasta incluem fontes de interpretação que auxilia na formatação das fontes;
• FText	Contém os primeiros 32k de conteúdo formatado;
• HeaderFText • (HT/Pro)	Contém texto de cabeçalho formatado para o documento;

• Height	Especifica altura do controle;
• HelpContextID	Especifica o contexto ID de vínculo para um arquivo de ajuda externo;
• HTag • (HT/Pro)	Especifica uma Hypertext HotSpot Etiqueta para texto selecionado;
• HTagL • (HT/Pro)	Especifica uma Hypertext HotSpot Etiqueta para texto para partiu de cursor;
• Hwnd	Retorna o comando para um controle;
• Index	Especifica um indexador para os controles;
• Left	Especifica a extremidade esquerda do controle ALLText;
• LeftMarg	Especifica uma margem esquerda para parágrafo selecionado;
• LineNumber	Especifica número de linha compensado desde o começo de documento;
• LineSpacing	Especifica o espaçamento de linha dentro de parágrafo selecionado;
• Mousepointer	Especifica o Ponteiro do mouse para uso em cima do controle de ALLText;
• MouseHPointer • (HT/Pro)	Especifica o MousePointer para usar em cima de hotspots;
• Name	Contém o nome do controle de ALLText;
• NTagL • (HT/Pro)	Especifica uma Etiqueta de marcador de páginas para texto para partiu de cursor;
• NTag • (HT/Pro)	Especifica uma Etiqueta de marcador de páginas para texto selecionado;
• NumParagraphs	Número de parágrafo dentro do documento;
• OLECode • (HT/Pro)	Especifica um código de caráter a ser devolvido dentro de SelText amarre como um placeholder para objetos embutidos;
• OleObject • (HT/Pro)	Controla embutindo de Objetos externos;

• OleVerb • (HT/Pro)	Controla ação de um clique duas vezes em objetos de OLE embutidos;
• Overtime	Especifica se digitando inserções caráter novos ou substitui caráter existentes;
• PageHeight	Especifica altura de uma página de texto;
• PageLineNumber	Especifica número de linha compensado de começo de página atual;
• PageNumber	Especifica número de páginas atual;
• Parent	Retorna o pai do objeto chamado;
• PrinterDC	Controla se ALLText usa Impressora DC;
• Ptag • (HT/Pro)	Especifica um valor de etiqueta de Inteiro Longo por parágrafo selecionado;
• RightMarg	Especifica uma margem direita para parágrafo selecionado;
• ScrollBarH	Apresentação da barra de rolagem horizontal ou Em, Fora ou Auto;
• ScrollBarV	Apresentação da barra de rolagem Vertical ou Em, Fora ou Auto;
• ScrollHorz	Coordenada X da janela em relação a começo de texto;
• ScrollVert	Coordenada Y da janela em relação a começo de texto;
• Select	Determina se a região selecionada esta dentro ou fora;
• SelfText	Especifica o texto selecionado que inclui códigos de formato embutidos;
• SelfType	Especifica como formatando códigos são interpretados para texto selecionado;
• SelLength	Especifica a duração de uma região seleta;
• SelStart	Especifica começando localização para uma região seleta;

• SelText	Especifica o texto selecionado sem códigos embutidos;
• SelToChar	Especifica carácter compensado dentro de parágrafo para o qual região de seleção estende;
• SelToPar	Especifica parágrafo para o qual região de seleção estende
• ShowHidden • (HT/Pro)	Determina se ou não mostrar texto escondido
• TabAdd • (HT/Pro)	Soma uma parada de aba definida para parágrafo especificado;
• TabAlignment • (HT/Pro)	Uma ordem que segura os tipos de alinhamentos de abas definiu para parágrafo selecionado;
• TabCount • (HT/Pro)	Lucros que o número de posições de aba definiu para parágrafo selecionado;
• TabDel • (HT/Pro)	Remove uma localização de parada de aba do parágrafo selecionado ;
• TabEnabled • (HT/Pro)	Especifica a função da chave de Aba ;
• TabIndex	Especifica a posição dentro da ordem de mudança de enfoque de controles na forma;
• TabLocations	Uma ordem que segura localizações de abas alinhadas para parágrafo selecionado;
• TabStep	Especifica falta aba passo tamanho para parágrafo selecionado;
• TabStop	Indica se o controle pode receber enfoque por uso da chave de Aba;
• Tag	Especifica um usuário definiu fio associado com o controle;
• TextFormatted	Indica quanto de um documento foi formatado completamente depois de I/O;
• TextLength	Lucros o número de carácter no documento;
• Text	Contém 32k de conteúdo primeiro • unformatted, sem códigos embutidos;
• Top	Especifica compensado de extremidade de topo do

	recipiente de ALLText;
• TopIndent	Especifica espaçando antes de um parágrafo selecionado;
• UndoAction	Gatilhos um Desfaça Ação;
• Visible	Especifica se ALLText é Visível;
• Width	Especifica a largura do controle de ALLText;
• WriteProtect	Especifica o modo de protegido contra digitação, o usuário apenas poderá ler o texto e não altera-lo.

Fonte: [BEN97].

Tabela 3. Funções de ALLText.

Propriedades	Características
• ATX_CurToXY	traduz de uma posição de cursor a X, Y para coordenadas;
• ATX_Print	impressões especificaram texto, e inicia eventos de impressora;
• ATX_PrintCancel	aborta imprimindo operação;
• ATX_Print_Finish	termina a conexão de ALLText para a impressora;
• ATX_Print_Region	imprime a região especificada de uma página;
• Atx_Print_Start	inicializa ALLText unem a dispositivo de impressora;
• ATX_Print_Title	provê controle em cima de texto de diálogo de impressão;
• ATX_ToLower	estes convertidos de função selecionaram texto para letra minúscula;
• ATX_ToUpper	estes convertidos de função selecionaram texto para Caso Superior;
• ATX_XYToCur	traduz coordenadas de X/Y a parágrafo e localização de caráter;
• ATX_RegNewExtern(HT/Pro)	registra objeto externo DLL;
• Find_Htag • (HT/Pro)	procuras para a ocorrência de HotSpot especificado;
• Find_Ntag • (HT/Pro)	procuras para a ocorrência de nth de Marcador de

	páginas especificado;
• Find_Phrase	procuras para um fio de texto especificado dentro do conteúdo do controle;
• Find_PhraseEX	procuras para um fio de texto especificado de formato especificado;
• FontTableGet	lê uma entrada;
• FontTablePut	jogos uma entrada;
• Get_Border • (HT/Pro)	traduz um valor de configuração de borda em características de componente;
• Get_LineTopHeight	adquire linha que formata detalhes para uma linha especificada;
• Get_ParDet	adquire todo o Parágrafo que formata detalhes para um determinado parágrafo;
• Get_Shadow	traduz um valor de Sombra em características de componente executáveis;
• Get_Underline	traduz um Sublinhe valor em características de componente executáveis;
• Make_Border • (HT/Pro)	cria valor de borda de componentes
• Make_Shadow	cria valor de sombra de componentes
• Make_UnderLine	cria sublinhe valor de componentes
• Print_AText	impressões especificaram alcance de parágrafo
• Print_ATextPages	impressões especificaram páginas

Fonte: [BEN97].

Tabela 4. Eventos de ALLText.

Propriedades	Características
• ATXChange	ativou por uma mudança de estado
• ATXGet	ativou pela colocação da propriedade de FileLoad; apoios Baixo I/O Nivelado;
• ATXPut	ativou pela colocação da propriedade de FileSave; apoios Baixo I/O nivelado;
• ATXHScrollClick	ativou através de trinco em barra de rolagem

	Horizontal;
• ATXVScrollClick	ativou através de trinco em barra de rolagem Vertical;
• DragDrop	padrão Evento Básico Visual;
• DragOver	padrão Evento Básico Visual;
• DropFileStart • (HT/Pro)	ativou o derrubando de um arquivo no controle;
• DropFile • (HT/Pro)	ativou o derrubando de um arquivo no controle;
• ExternOLEAction • (HT/Pro)	provê dados trocam com objetos embutidos;
• PrintThisPage	ativou o prior para imprimindo cada página;
• PrintStartPage	ativou o prior para o começo de cada cada página impressa;
• PrintEndPage	ativou em conclusão da impressão de cada página de texto.

Fonte: [BEN97].

Na figura 10, temos os principais efeitos do componente Active X “AllText HT/Pro”.

Figura 10. Tela comercial de visualização do componente Active X “AllText HT/Pro”.

ALLText

HYPERTEXT **Mail Merge**

Word Processing, Report Generation

Shadowed Text
bmp, ico, jpeg, wmf

Hot Spot

Twinkle, Twinkle Silly Star , How I wonder what you are.
Up above the world so high, Like a diamond in the sky.
Insert here a spreadsheet now, see what's better -
diamond or cow.

	A	B	C	D	E
1	Possible Material of Star	Diamond		Milk	
2	Mass of Star	3.82E+33 oz		3.82E+31 oz	
3	Mass of \$1 of Material	0.0025 oz		90 oz	
4	Dollar Value of Star	1.53E+36 \$		4.78E+29 \$	

Poor Miss Muffett sat on her tuffet. If she spent her time with **ALLText HT/Pro** no spider would have dared to mess with her.

Tables & OLE Objects

B I U S [Color] [Align] [List] [List] [List] [List]

Fonte: [BEN97].

4.1.2 QUADRO DE CARACTERÍSTICAS

No quadro 6 são apresentadas as principais diferenças entre as versões lançadas no mercado.

Quadro 6. Características ALLText HT/Pro.

	AllText 4.0 Standard VBX	AllText 4.0 HT/Pro VBX	AllText 4.0 Pen VBX	AllText 4.0 HT/Pro 16 Bit OCX	AllText 4.0 Pen 16 Bit OCX	AllText 4.0 HT/Pro 32 Bit OCX
Lançado	Sim	Sim	Sim	Sim!	Não	Sim!
Dados	Não	Sim	Sim	32K limite	32K limite	Sim
Suporte a Objetos OLE e figuras	Não	Sim	Sim	Sim	Sim	Sim
Suporta fonte escondida	Não	Sim	Sim	Sim	Sim	Sim
Suporta cabeçalho e paginação	Não	Sim	Sim	Sim	Sim	Sim
Hypertext e estilos	Não	Sim	Sim	Sim	Sim	Sim
Suporta tabela do MSWord	Não	Sim	Sim	Sim	Sim	Sim
Detalhes especiais			Habilidade para fazer esboços e texto destacando. Habilidade para exibir Objetos em texto ou debaixo de texto.	Suporte Ole 2.0 Drag and Drop	Suporte Ole 2.0 Drag and Drop	Suporte Ole 2.0 Drag and Drop

Fonte: [BEN97].

4.2 TECNOLOGIA ACTIVE X

Segundo [BOR99] a tecnologia Active X tem muita utilidade no desenvolvimento de aplicações pequenas e sem necessidade de configuração para ambiente distribuído na *web* com ou sem arquitetura de banco de dados Cliente/Servidor multicamadas.

O *Activex* é um novo nome para o novo enfoque (desenvolvimento para INTERNET) visando empacotar as tecnologias chave da *Microsoft*® de reutilização de funções

programadas e fornecimento de serviços entre aplicativos: *COM e OLE (Object Linking and Embedding)*.

A tecnologia *OLE* é usada para a criação de controles *OCX*, que pode ser 32 bits. Devido a um imenso apoio da Microsoft® às empresas deste ramo, logo todas elas atualizaram seus produtos para a nova tecnologia, hoje os *VBX* estão ficando mais que desaparecendo, enquanto o *OCX* cresce a cada dia.

A Microsoft® examinou sua tecnologia de controle do *OLE* - que é o componente central de seus onipresentes sistemas operacionais *Microsoft Windows®* do seu grupo de aplicativos, *Microsoft® Office* e descobriu um dinossauro. O *OLE* tinha inchado, arrastando consigo tantas informações e características de interface que os componentes *OLE* não podiam viajar com rapidez por uma conexão típica da Internet.

Por exemplo, um controle *OLE* precisa carregar consigo informações que definem uma janela na tela, mesmo que este controle jamais seja visível. Assim foi construída uma variedade mais ágil e mais adaptável. Do mesmo *Kernel* do *COM*, com o *OLE*, mas com um conjunto menor de exigências centrais e novas opções para a construção de refinadas e dinâmicas páginas Web.

Agora os controles criados com essa nova tecnologia são chamados *Controles Active X*. Isto quer dizer que toda a indústria de controles *Activex/OCX* esta automaticamente disponível também para a Internet. Do mesmo jeito que um Controle *Activex/OCX* é usado num programa, ele poderá ser usado numa página *HTML*, sozinho, como um *applet* da linguagem *JAVA*, ou dentro de um programa. Dependendo da aplicação, esses controles vão poder ser usados por leigos ou só por programadores. Pode existir um controle que o usuário só precise escrever uma frase para esta ficar pulando pela tela, enquanto outro que trate com banco de dados precise de programação.

Esses arquivos serão executados junto com a página que os contém e colocados num cache permanente no computador do usuário, que será notificado instalação na máquina. Devido à questão da segurança, existe um tipo de diploma dado pela Microsoft® ou por uma representante sua que certifica que este controle não pode causar danos ao usuário.

5 PRINCÍPIOS DE ANÁLISE DE SISTEMA

Para a implementação deste trabalho, fez-se uso da metodologia de análise estruturada, apresentadas como a seguir.

Um aspecto primordial a se considerar é o significado da palavra análise. Segundo[DEM89] análise é o estudo de um problema, que antecede à tomada de uma decisão. Realizando-se uma adaptação ao contexto de desenvolvimento de sistemas computacionais, análise refere-se ao estudo de alguma área de trabalho ou de uma aplicação, levando quase sempre à especificação de um novo sistema. A ação que será tomada é a implementação deste sistema.

O uso de uma metodologia de desenvolvimento de sistemas é um roteiro de trabalho sobre o qual pode-se visualizar todas as etapas de construção de um software, suas interdependências, bem como o progresso do mesmo. A especificação de uma Metodologia de Desenvolvimento de Sistemas tem como pré-requisito indispensável à definição do ciclo de vida do software a ser adotado no processo de desenvolvimento.

Ao se adotar uma metodologia, na maioria das vezes adota-se também um software para o apoio ao processo de desenvolvimento de sistemas, um ambiente de desenvolvimento automatizado: CASE (*Computer Aided Software Engineering*).

5.1 ANÁLISE ESTRUTURADA

É um conjunto de técnicas e ferramentas cujo objetivo é auxiliar na análise e definição de sistemas. O conceito fundamental é a construção de um modelo do sistema utilizando técnicas gráficas. A metodologia envolve a construção *top-down* do sistema por refinamentos sucessivos; isto é, *top-down* consiste na estratégia de se projetar um sistema, dividindo-o em funções principais, fracionando-as sucessivamente, em partes menores, até que a implementação possa ser expressa em termos de comando de programa.

Na visão de [YOU89], análise estruturada, refere-se ao “extremo inicial” de um projeto de desenvolvimento de sistemas, durante o tempo em que os requisitos do usuário são definidos e documentados.

De acordo com [KEL90] o desenvolvimento estruturado de sistemas e a produção de uma especificação de sistemas concisa, não-ambigua, não-redundante e rigorosa. A análise estruturada tem sua especificação baseada em diagramas de fluxo de dados e é apoiada por uma breve narrativa ainda que estruturada em português, por uma descrição do banco de dados lógico e por um dicionário de dados completo para o projeto. Esta especificação estruturada das necessidades do usuário e convertida em diagramas de estrutura de módulos durante o projeto que por sua vez são transformados em programas estruturados durante a implementação.

Pode-se citar como uma importante característica da Análise Estruturada o encorajamento da uniformidade e coerência nas atividades da análise, projeto e codificação,

5.1.1 CARACTERÍSTICAS DA ANÁLISE ESTRUTURADA

Os projetos estruturados são caracterizados por melhores ferramentas para expressarem os requisitos dos usuários, ênfase no projeto de qualidade e bom código, e desenvolvimento de sistemas top-down [YOU92].

A especificação de sistema que utilize a análise estruturada deve conter as seguintes características, segundo [MEL90]:

a) métodos de representação gráfica capaz de sugerir ao usuário e ao analista a dinâmica dos componentes sistêmicos, quais sejam: os processos, os dados e seus fluxos, os arquivos e os órgãos que participam da execução dos processos;

b) particionamento dos componentes, possibilitando a análise detalhada de cada uma das partes que compõe o “todo sistêmico”. A idéia é a de que para se compreender como funciona um determinado processo, não se tenha que estudar os demais processos;

c) hierarquia na especificação dos componentes, facilitando a percepção da integração entre estes processos;

d) elaboração de um modelo que permita uma avaliação previa que servira de base para o desenvolvimento das fases subsequentes ao levantamento e análise;

e) facilidade de manutenção, tendo em vista as constantes e naturais modificações durante a vida do software.

5.1.2 OBJETIVOS DA ANÁLISE ESTRUTURADA

Os principais objetivos das técnicas estruturadas [MAC91] são:

- a) construir programas de alta qualidade que tenham comportamento previsível;
- b) construir programas que sejam facilmente modificáveis (manuteníveis);
- c) simplificar os programas e seu processo de desenvolvimento;
- d) conseguir maior previsibilidade e controle no processo de desenvolvimento;
- e) acelerar o desenvolvimento de sistemas;
- f) diminuir o custo de desenvolvimento de sistemas.

A Análise Estruturada se propõe a fornecer um meio de comunicação comum entre usuários e analistas, com o intuito de total acordo e apoio entre ambos. Além de descrever os requisitos lógicos do sistema sem, entretanto, ditar a forma de implementação física. Ou seja, a Análise Estruturada, estabelece uma separação formal entre o lógico e o físico.

5.1.3 FERRAMENTAS DA ANÁLISE ESTRUTURADA

Para[YOU89) as ferramentas de documentação da Análise Estruturada consistem no seguinte:

- a) diagramas de Fluxo de Dados (DFDs);
- b) dicionário de Dados (DD);
- c) diagrama de entidades relacionadas (MERs).

5.1.3.1 DIAGRAMA DE FLUXO DE DADOS (DFD)

Pode-se definir o diagrama de fluxo de dados como a principal ferramenta da análise estruturada para mostrar graficamente o processamento de dados; a fim de desenvolver um sistema que atenda as necessidades do usuário, tornando-se assim, uma linguagem comum para discussão entre usuários e analistas.

Definição de um DFD segundo [DEM89]:

“Um Diagrama de Fluxo de Dados é uma representação em rede de um sistema. O sistema pode ser automatizado. O DFD retrata o sistema em termos de suas partes componentes, com todas as interfaces entre os componentes indicadas”.

Contudo, algumas vezes é comum confundir os termos DFD e fluxogramas. A diferença existente entre ambos é que o DFD mostra o fluxo de dados, não o de controle. O DFD retrata uma situação sob o ponto de vista dos dados, enquanto o fluxograma representa uma situação sob o ponto de vista do que age sobre os dados [DEM89].

O primeiro diagrama do conjunto diagrama de fluxo de dados corrente é o diagrama de contexto. O diagrama de contexto define o escopo do projeto estruturado quando mostra quais dados físicos o sistema recebe e envia. Um sistema típico requer vários níveis de diagramas de fluxo de dados.

De acordo com [FOU94] a finalidade do diagrama de contexto é situar o sistema dentro do ambiente de negócios da empresa, para posteriormente identificar e formalizar o escopo do projeto.

Os diagramas de fluxo de dados são compostos basicamente por quatro elementos:

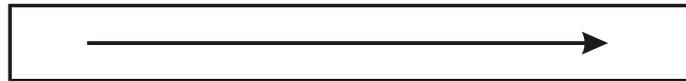
- a) fluxo de dados, representados por uma seta;
- b) processos, representados por círculos ou “bolhas” ou ainda por um círculo ou retângulo.
- c) arquivos, representados por um par de linhas paralelas;
- d) fontes e destinos de dados, representados por um retângulo.

Consta a seguir uma breve explanação de cada elemento, segundo [MCC91].

5.1.3.1.1 FLUXO DE DADOS

O fluxo de dados conduz o fluxo de informações através dos processos de um sistema graficamente. O sentido do fluxo de dados é indicado por uma seta. Os dados são identificados por nomes escritos ao lado de sua flecha correspondente.

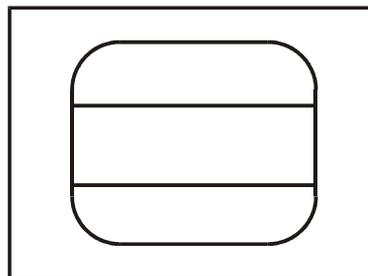
Figura 11. Notação de fluxo de dados.



5.1.3.1.2 PROCESSO

Um processo é uma transformação do(s) fluxo(s) de dados de entrada em fluxo(s) de saída. O processo é um componente procedural do sistema. Opera sobre (ou transforma) os dados. Por exemplo, pode executar operações aritméticas ou lógicas com os dados, para produzir algum(ns) resultado(s). Cada processo é representado no DFD por um círculo ou um retângulo com os vértices arredondados. O nome do processo é escrito dentro do círculo. Deve ser usado um nome significativo, para definir a operação executada pelo processo.

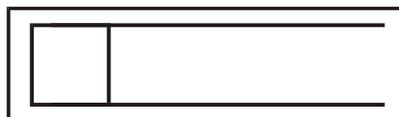
Figura 12. Notação de processo.



5.1.3.1.3 DEPÓSITO DE DADOS

Um depósito de dados representa um arquivo lógico (um repositório de informações). É desenhado no DFD como um par de linhas paralelas (às vezes, fechadas em um dos lados). O nome do depósito de dados é escrito entre as linhas.

Figura 13. Notação de depósito de dados.

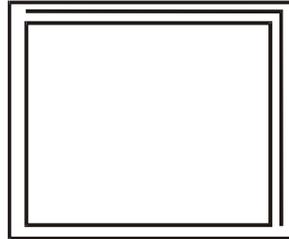


Cada depósito de dados é ligado a um "retângulo" de processo por meio de um fluxo de dados. O sentido da seta do fluxo de dados mostra se os dados estão sendo lidos do depósito de dados para o processo ou produzidos pelo processo e então enviados para o depósito de dados.

5.1.3.1.4 PONTO TERMINAL (ENTIDADE EXTERNA)

Um ponto terminal mostra a origem dos dados usada pelo sistema e o ultimo receptor de dados produzidos pelo sistema. A origem dos dados e chamada de fonte e o receptor dos dados e chamado de destino. Para representar um ponto terminal em um DFD, e usado um retângulo ou um quadrado duplo.

Figura 14. Notação de entidade externa.



5.1.3.2 DICIONÁRIO DE DADOS (DD)

O dicionário de dados fornece a informação de texto de suporte para complementar a informação gráfica mostrada no DFD. Um Dicionário de Dados é simplesmente um grupo organizado de definições de todos os elementos de dados no sistema sendo modelado. [DEM89]

O dicionário de dados é um repositório sobre os dados do sistema. De maneira figurativa, pode-se dizer que se trata de um banco de dados sobre os dados do sistema. Deve conter definições dos elementos que tornam o DFD preciso, os quais são:

- a) processo (Código, Nome, Objetivo, Mini-especificação);
- b) fluxo de dados (Nome, Estrutura de Dados);
- c) entidade externa (Nome, Descrição);
- d) elementos de dados (Nome, Tipo (formato), Tamanho, Domínio).

5.1.3.3 MER (MODELO ENTIDADE E RELACIONAMENTO)

Segundo [VAZ99], as entidades são abstrações de dados do mundo real, possuindo as seguintes características:

- a) possuem existência própria no banco de dados sendo tratados como unidades lógicas de manipulação;

- b) podem mudar de tipo ao longo de sua existência;
- c) são individualizados, isto é, os objetos são distintos entre si.

O relacionamento pode aparecer num modelo como um atributo, entidade, elemento independente ou função. Relacionamento como atributo é aquele que o atributo de um objeto está conectado a, apontando para ou está derivando de outros objetos.

Relacionamento como entidade é apresentado se o relacionamento de dois ou mais objetos descrevem um modelo de objetos distintos.

Relacionamento pode também ser visto como objetos independentes distintos de entidades. Isso não implica que o banco de dados físico representa relacionamentos e entidades com diferentes estruturas, mas ao nível de conceito, entidades são separadas de relacionamentos.

Representação funcional é obtida por permitir especificação de relacionamentos de objetos através de definições funcionais nas linguagens de definição de dados.

5.1.4 CICLO DE VIDA ESTRUTURADO

Para o desenvolvimento de um projeto de software faz-se necessário o uso de uma metodologia, a fim de disciplinar os métodos de trabalho e facilitar os controles. Desta maneira organização de pequeno a grande portes tem adotado o ciclo de vida, conhecido como metodologia de desenvolvimento de sistemas. Um ciclo de vida do projeto documentado oferece um modo simples para que qualquer pessoa da organização de desenvolvimento de sistemas possa entender as atividades de desenvolvimento de um sistema de processamento.

O ciclo de vida do software deve, em um processo de detalhamento, especificar os elementos referentes aos aspectos gerenciais (pessoas, tarefas, etapas, documentos, pontos de controle, formas de controle, etc.) e tecnológicos (ferramentas e métodos) do processo de desenvolvimento.

Segundo [YOU92] apresenta propósitos pelo qual deve ter um ciclo de vida de projeto:

- a) para definir as atividades a serem executadas em um projeto de desenvolvimento de sistemas;

- b) para introduzir consistência entre muitos projetos de desenvolvimento de sistemas da mesma organização;
- c) para introduzir pontos de verificação para o controle gerencial de decisões.

O primeiro objetivo é muito importante para casos onde ocorre troca de controle gerencial no decorrer de um projeto. O novo gerente necessita ficar bem entrosado com o projeto em desenvolvimento, a fim de não subestimar a importância de alguma das fases.

Para organizações maiores o segundo objetivo tem uma importância mais acentuada, devido aos vários projetos que são desenvolvidos em uma empresa, pois se tornaria um caos se cada projeto seguisse uma metodologia diferente.

O terceiro objetivo do ciclo de vida de um projeto comum relaciona-se com a necessidade da gerência de controlar um projeto. Para projetos menores geralmente os pontos de verificação são saber se o projeto foi acabado no tempo e orçamento previstos e se se observou os requisitos do usuário; contudo, para projetos maiores e relevantes estes pontos de controle intermediários, através dos quais tem oportunidades de determinar se o projeto está fora das previsões e se será. Preciso obter recursos adicionais; em alguns casos até mesmo o usuário pode requisitar que durante o projeto haja pontos de controle.

Em suma, um ciclo de vida do projeto implica em organização para as atividades gerenciais, tornando possível que sejam detectados e solucionados problemas em momentos antecedentes ao término do projeto.

Consta a seguir cada uma das atividades de um ciclo de vida segundo os autores [YOU92] e [FOU94].

- a) levantamento;
- b) análise do sistema;
- c) projeto;
- d) implementação;
- e) geração do teste de aceitação;
- f) atividades finais.

5.1.4.1 LEVANTAMENTO

Essa atividade e também conhecida como estudo da viabilidade ou estudo inicial das atividades, segundo [YOU92]. Contudo [FOU94] define como Fase da Pesquisa.

Os requisitos nos quais esta fase deve ater-se são:

- a) problemas atuais que necessitam ser eliminados no ambiente do usuário;
- b) definição do escopo do sistema;
- c) pesquisa inicial para soluções de implementação do sistema;
- d) elaboração de uma análise preliminar de custo e benefício.

A análise da situação atual tem como objetivo a reunião com os principais grupos de usuários, que provavelmente serão afetados pelo projeto, e descrever a justificativa de negocio para o novo sistema.

E importante uma adequada análise do sistema em uso, se houver, para uma melhoria nas deficiências do sistema atual, devidamente documentadas. Ocorre a identificação das funções que podem ser reaproveitadas ou aprimoradas e funções que necessitam ser elaboradas.

Definir o escopo para o novo sistema. Documentar os objetivos que devem ser alcançados com a instalação do sistema, devidamente esquematizados em grau de prioridade e importância e uma sugestão de [FOU94].

A criação de um diagrama de contexto nesta fase tem como objetivo situar o sistema proposto dentro do ambiente de negocio existente, realizando desta forma uma verificação se o sistema proposto não repete outro já existente. O diagrama de contexto deve documentar todos os elementos gráficos representados.

Enfim esta atividade e considerada critica, pois, e a partir do levantamento que a gerência decide a execução ou cancelamento do projeto, caso este não for satisfatório sob o ponto de vista custo/benefício.

Técnicas de obtenção de fatos:

- a) entrevistas;
- b) questionários de pesquisa;

- c) sessões informais de *brainstorming* em grupo;
- d) análise de fluxo de processo;
- e) análise de observações;
- f) sessões tipo JAD;
- g) revisão da documentação.

5.1.4.2 ANÁLISE DO SISTEMA

A finalidade desta atividade é investigar as soluções de implementação de sistemas que forma identificada durante a atividade de pesquisa e eleger a solução que melhor atenda as necessidades dos usuários. Durante a análise são examinadas as necessidades dos usuários e definidas as propriedades que o sistema deve possuir para satisfazer a essas necessidades. São identificados também às restrições e os requisitos do sistema e os requisitos de desempenho. As funções a serem calculadas são definidas precisamente, mas não se considera como estas funções irão operar. O resultado da atividade análise é a especificação funcional.

5.1.4.3 PROJETO DO SISTEMA

A atividade de projeto tem como objetivo definir a arquitetura interna do sistema. As especificações detalhadas para codificar e testar os programas serão desenvolvidos gradualmente em paralelo com o projeto dos arquivos e bancos de dados físicos do sistema.

Alem disso, a atividade de projeto ocupa-se com a transformação de modelos de dados de entidades e relacionamentos em um projeto de banco de dados, [YOU92].

5.1.4.4 IMPLEMENTAÇÃO

A finalidade primária da fase de implementação é liberar um sistema completamente operacional para os usuários. Com base nas especificações de projeto do sistema realizada nas atividades anteriores, os programas são codificados, testados e gradualmente integrados em um sistema completo. Os procedimentos manuais e administrativos dos sistemas são finalizados e também testados em conjunto com a parte automatizada. É realizada a instalação dos equipamentos e recursos de hardware/software/rede apropriados no local.

5.1.4.5 GERAÇÃO DO TESTE DE ACEITAÇÃO

Esta atividade pode iniciar-se paralelamente as atividades de projeto e implementação. Os testes de aceitação são realizados a partir da especificação estruturada.

5.1.4.6 ATIVIDADES FINAIS

Esta atividade engloba a garantia da qualidade, descrição dos procedimentos e instalação.

6 DESENVOLVIMENTO DO SISTEMA

Este capítulo destina-se a explicar desde a metodologia de sistema utilizada para desenvolvimento do trabalho, quanto os aspectos relacionados à implementação do sistema.

Seguiram-se os seguintes passos do ciclo de vida:

- a) levantamento de Dados;
- b) análise do sistema;
- c) projeto do sistema;
- d) implementação;
- e) geração do teste de aceitação;
- f) atividades finais.

6.1 LEVANTAMENTO DOS DADOS

Para este trabalho utilizou-se como técnica de levantamento de dados a entrevista realizada com a Sra. Cecci, Administradora da Hemodinâmica.

A Sra. Cecci inicialmente descreveu os seguintes processos que necessitavam de informatização:

- a) arquivo de pacientes;
 - informações do laudo;
 - informações do paciente;
 - informações cadastradas em forma de texto compatível com o Microsoft® Word;
 - capacidade de texto longos, até 30 páginas.
 - informações sobre os orçamentos;
 - informações sobre os custos do paciente ou conta;
- b) arquivo de médicos;
- c) arquivo de fornecedores;
- d) arquivo de autoridades;
- e) arquivo de hospitais;
- f) arquivo de convênios;
- g) arquivo de procedimentos;

- h) arquivo de medicamentos;
- i) arquivo de materiais de alto custo;

Para que pudessem gerar os relatórios e gráficos estatísticos dos mesmos.

6.1.1 ANÁLISE

Para a análise, que é um misto das análises estruturadas e essenciais, fazendo uso das facilidades da lista de eventos fornecidos pela análise essencial, e diagramas de fluxo de dados da análise estruturada, para fins de documentação do sistema teve-se como auxílio parcial a ferramenta ACCESS. A seguir será apresentada a lista de eventos, diagrama de contexto, diagrama de fluxo de dados nível 0, modelo de entidade relacionamento, seguido do dicionário de dados.

6.1.2 LISTA DE EVENTOS

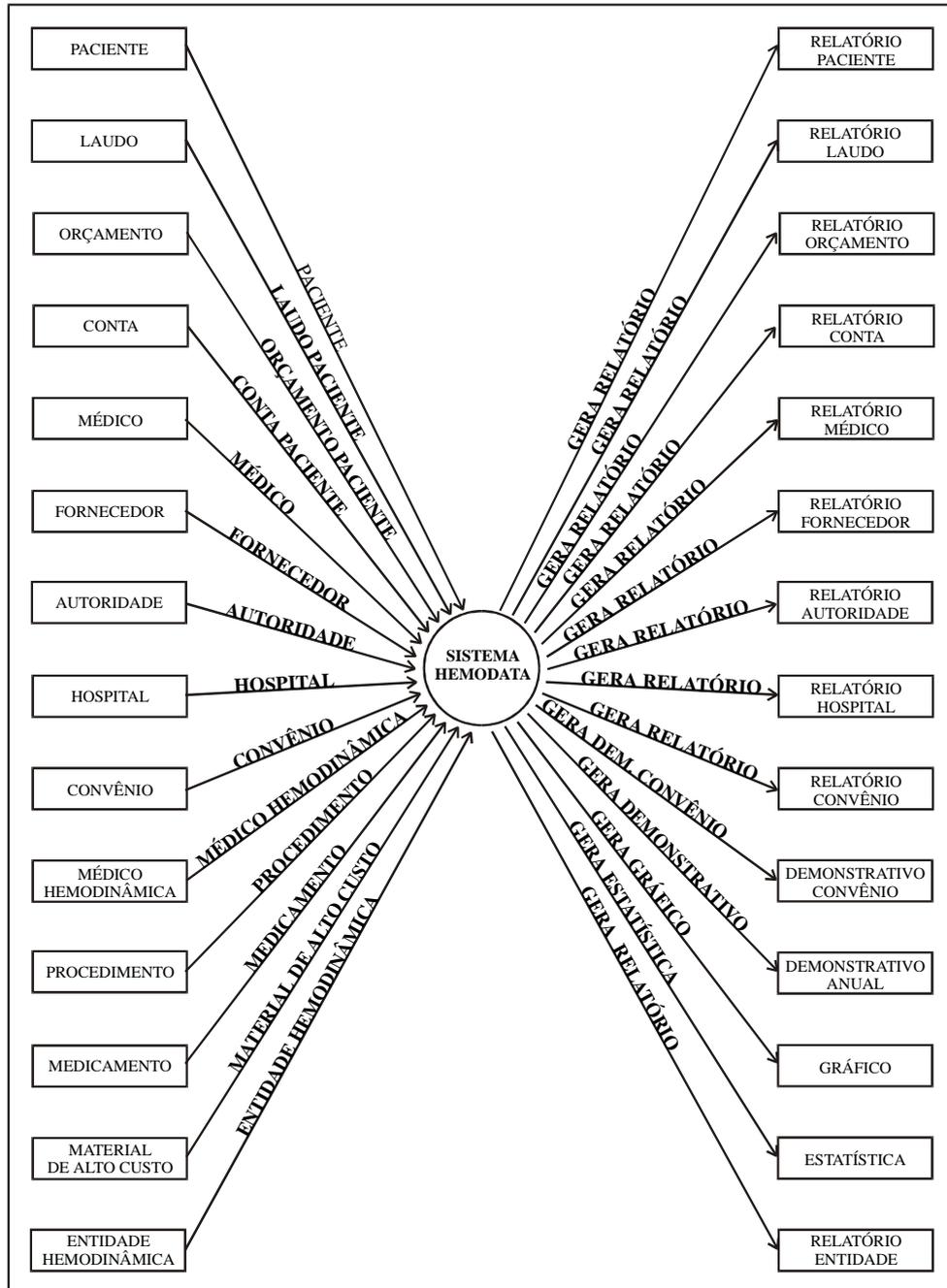
Relacionados abaixo se encontra a lista de eventos utilizada pelo Sistema Hemodata para controle de informações para a Clínica de Hemodinâmica.

- a) paciente efetuam cadastro;
- b) médicos efetuam cadastro;
- c) fornecedores efetuam cadastro;
- d) autoridades efetuam cadastro;
- e) hospitais efetuam cadastro;
- f) convênios efetuam cadastro;
- g) paciente é examinado;
- h) laudo é gerado emitido pelo médico;
- i) conta é confeccionada;
- j) paciente recebe relação de procedimentos;
- k) periodicamente médico recebe relatórios de consultas;
- l) periodicamente médico recebe relatórios de procedimentos realizados;
- m) mensalmente é emitida relação de guias de consultas para os respectivos convênios;

6.1.3 DIAGRAMA DE CONTEXTO

Desenvolveu-se um modelo formal dos processos manuais e automatizados do sistema, usando a técnica de diagrama de fluxo de dados. Para a visualização do diagrama de contexto do sistema veja figura 15:

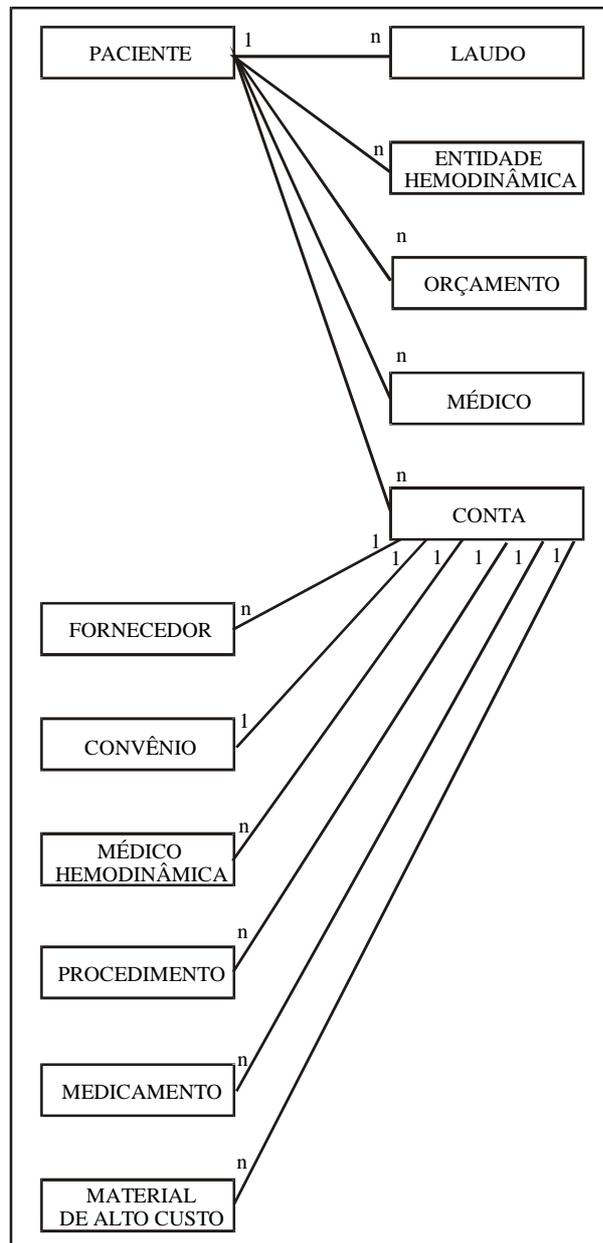
Figura 15. Diagrama de Contexto.



6.1.4 MER (MODELO ENTIDADE RELACIONAMENTO)

A seguir na figura 16 está ilustrado o modelo de entidade relacionamento utilizado neste sistema.

Figura 16. Modelo Entidade Relacionamento.



6.1.5 DICIONÁRIO DE DADOS

A seguir descreve-se os processos, depósitos de dados, entidades externas. Os elementos de dados.

6.1.5.1 PROCESSOS

CADASTRAR PACIENTE	Cadastrar informações relevantes ao paciente, após o cadastro dos dados é que é possível cadastrar o laudo, orçamento e gerar a conta dos procedimentos realizados no paciente.
LAUDO	Cadastrar o laudo do(s) procedimento(s) efetuados no paciente, bem como o resultado da intervenção cirúrgica, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa <i>“THE BENNET ALL TEXT TECHNOLOGY INC.”</i>
ORÇAMENTO	Cadastrar informações do orçamento entregue ao paciente para confrontá-lo posteriormente com a conta gerada pelo mesmo.
ENTIDADE HEMODINÂMICA	Cadastrar informações do histórico hemodinâmico do paciente, este histórico é preenchido em uma ficha pelo cardiologista na consulta.
CONTA	Cadastrar informações dos materiais, materiais de alto custo, fornecedor, médicos, procedimentos, convênio utilizados na intervenção cirúrgica para tratamento do paciente.
MÉDICO	Cadastrar informações relevantes sobre os médicos.
FORNECEDOR	Cadastrar informações relevantes sobre fornecedores.
AUTORIDADE	Cadastrar informações relevantes sobre autoridades
HOSPITAL	Cadastrar informações relevantes sobre hospitais
CONVÊNIO	Cadastrar informações relevantes sobre convênios de saúde.
MÉDICO HEMODINÂMICA	Cadastrar informações relevantes sobre os médicos que fazem parte da clínica de hemodinâmica.
PROCEDIMENTO	Cadastrar informações relevantes sobre os procedimentos realizados pela clínica de hemodinâmica, neste processo se cadastra um fator para gerar conta.

MEDICAMENTO	Cadastrar informações relevantes sobre os medicamentos utilizados pela clínica de hemodinâmica neste processo se cadastra um fator para gerar conta.
MATERIAL DE ALTO CUSTO	Cadastrar informações relevantes sobre os materiais de alto custo importados, controlados pelo governo que são utilizados pela clínica de hemodinâmica neste processo se cadastra um valor para gerar conta.
RELATÓRIO PACIENTE	Neste processo o objetivo é gerar relatório dos dados dos pacientes, organizado pelo <i>n</i> , Agrupado pelo <i>n</i> No período de <i>n</i> Até <i>n</i> , Convênios, Encaminhados diversos, procedimentos e por último gerar etiquetas para mala-
RELATÓRIO LAUDO	Neste processo o objetivo é gerar relatório dos dados do laudo do paciente, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>THE BENNET ALL TEXT TECHNOLOGY INC.</i> ".
RELATÓRIO ORÇAMENTO	Neste processo o objetivo é gerar relatório dos dados do orçamento entregue ao paciente, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO CONTA	Neste processo o objetivo é gerar relatório dos dados da conta do paciente, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO MÉDICO	Neste processo o objetivo é gerar relatório dos dados dos médicos, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO FORNECEDOR	Neste processo o objetivo é gerar relatório dos dados dos fornecedores, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO AUTORIDADE	Neste processo o objetivo é gerar relatório dos dados das autoridades, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO HOSPITAL	Neste processo o objetivo é gerar relatório dos dados dos hospitais, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO CONVÊNIO	Neste processo o objetivo é gerar relatório dos dados dos convênios, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".
RELATÓRIO ENTIDADE	Neste processo o objetivo é gerar relatório dos dados do histórico hemodinâmico do paciente, neste processo foi utilizado a tecnologia <i>ACTIVE X</i> da empresa " <i>SEAGATE CRYSTAL REPORTS.</i> ".

DEMONSTRATIVO CONVÊNIO	Neste processo o objetivo é cadastrar o demonstrativo mensal dos provenientes dos convênios ou planos de
DEMONSTRATIVO ANUAL	Neste processo o objetivo é cadastrar o demonstrativo anual dos planos de saúde.
GRÁFICO	Neste processo o objetivo é gerar gráficos estatísticos dos realizados, via de acesso, idade, médico solicitante, plano de saúde, sexo emergência.
ESTATÍSTICA	Neste processo o objetivo é gerar estatísticas dos procedimentos realizados, de acesso, idade, médico solicitante, plano de saúde, sexo e emergência outros.

6.1.5.2 DEPÓSITOS DE DADOS

A tabela 5 mostra os elementos de dados dos depósitos

Tabela 5. Depósito de dados.

TABELA	ELEMENTOS DE DADOS
Agenda	Código, Data, Informação.
AMB_90	Código, Tabela, Total_CRRCH, Número de Auxiliares
AMB_92	Código, Tabela, Descrição, Total_CRRCH, Número de Auxiliares.
AMB_96	Código, Tabela, Descrição, Total, Número de Auxiliares
Autoridade	Código, Título, Autoridade, Função, Contato, Rua, Número da Rua, Adicional, Estado, Bairro, Telefone, Cidade, CEP, Observações.
CIEFAS	Código, Tabela, Descrição, Total, Número de Auxiliares.
Conta	Código Conta, Código, Código_Tabela, Procedimento, Procedimento4, Procedimento5, Valor_Convertido, Hemodinamicista, Observações, Auxiliar1, Auxiliar2, Código_Tabela1, Exames Complementares1, Valor_Convertido1, Código_Tabela2, Exames Complementares2, Valor_Convertido2, Código_Tabela5, Código_Tabela4, Código_Tabela3, Exames Complementares3, Valor_Convertido3, Tabela Cobrança, Taxa Mat Altocusto, Taxa Administrativa, Taxa Auxiliar, Taxa Auxiliar2, Taxa Desconto, Fator Multiplicativo, Fator Multiplicativo1, Fator Multiplicativo2, Fator Multiplicativo3, Desconto Reais, Desconto sobre Honorários, Fator Multiplicativo4, Fator Multiplicativo5, Valor_Convertido4, Valor_Convertido5, Fator Conversão Tabela, Auxiliar_new1, Auxiliar_new2, Auxiliar_new3, Auxiliar_new4, Auxiliar_new5, Auxiliar_new6.
Convênio	Código, Título, Diretor, Convênio, Rua, Número da Rua, Adicional, Estado, Bairro, Telefone, Cidade, CEP, Observações.
Demonstrativo	Código Demonstrativo, Ano, Convênio, Total Fatura Janeiro, IRRF Janeiro, Taxas Janeiro, Valor Líquido Janeiro, Total Fatura Fevereiro, IRRF Fevereiro, Taxas Fevereiro, Valor Líquido Fevereiro,

	<p>TotalFaturaMarço, IRRFMarço, TaxasMarço, ValorLíquidoMarço, TotalFaturaAbril, IRRFAbril, TaxasAbril, ValorLíquidoAbril, TotalFaturaMaio, IRRF Maio, TaxasMaio, ValorLíquidoMaio, TotalFaturaJunho, IRRFJunho, TaxasJunho, ValorLíquidoJunho, TotalFaturaJulho, IRRFJulho, TaxasJulho, ValorLíquidoJulho, TotalFaturaAgosto, IRRF Agosto, TaxasAgosto, ValorLíquidoAgosto, TotalFaturaSetembro, IRRFSetembro, TaxasSetembro, ValorLíquidoSetembro, TotalFaturaOutubro, IRRFOutubro, TaxasOutubro, ValorLíquidoOutubro, TotalFaturaNovembro, IRRFNovembro, TaxasNovembro, ValorLíquidoNovembro, TotalFaturaDezembro, IRRFDezembro, TaxasDezembro, ValorLíquidoDezembro, Resultado1, Resultado2, Resultado3, Resultado4, DataPagJaneiro, DataPagFevereiro, DataPagMarço, DataPagAbril, DataPagMaio, DataPagJunho, DataPagJulho, DataPagAgosto, DataPagSetembro, DataPagOutubro, DataPagNovembro, DataPagDezembro, ObsJaneiro, ObsFevereiro, ObsMarço, ObsAbril, ObsMaio, ObsJunho, ObsJulho, ObsAgosto, ObsSetembro, ObsOutubro, ObsNovembro, ObsDezembro, Data2PagJaneiro, Data2PagFevereiro, Data2PagMarço, Data2PagAbril, Data2PagMaio, Data2PagJunho, Data2PagJulho, Data2PagAgosto, Data2PagSetembro, Data2PagOutubro, Data2PagNovembro, Data2PagDezembro.</p>
Entidade	<p>CódigoEntidade, Codigo, QuadroClínico, DataInfarto, Sk, Idade, Diabético, Hipertenso, Tabagismo, HistóriaFamiliar, Dislipidemia, Obesidade, AoInicio, AoFim, VeInicio, VeMeio, VeFim, Cp, Ap, VdInicio, VdMeio, VdFim, Ad, GradienteValvar, GradienteMitrál, FraçãoEjeção, Aortografia, CoronáriaDireita, Ostium, Tronco, DescendenteAnterior, Circunflexa, Diagonalis, Arteriografia, By-Pass, Observações, Material1, Material2, Material3, Modelo1, Modelo2, Modelo3, Quantidade1, Quantidade2, Quantidade3, VasoLocal1, VasoLocal2, VasoLocal3, VasoLocal4, Dilatado1, Dilatado2, Dilatado3, Dilatado4, AB1B2C1, AB1B2C2, AB1B2C3, AB1B2C4, Segmentar1, Segmentar2, Segmentar3, Segmentar4, AnguloMaior90Graus1, AnguloMaior90Graus2, AnguloMaior90Graus3, AnguloMaior90Graus4, EnvRamos1, EnvRamos2, EnvRamos3, EnvRamos4, Cálcio1, Cálcio2, Cálcio3, Cálcio4, Úlcera1, Úlcera2, Úlcera3, Úlcera4, Excêntrica1, Excêntrica2, Excêntrica3, Excêntrica4, Trombo1, Trombo2, Trombo3, Trombo4, Via de Acesso, Introdutor, CateterGuia, CateterBalão, Guia, Outros, KissingBallon, KissingWire, Via de AcessoRot, IntrodutorRot, CateterGuiaRot, CateterBalãoRot, GuiaRot, OutrosRot, Eletivo, Emergência, Via de AcessoStent, IntrodutoStent, CateterGuiaStent, CateterBalãoStent, GuiaStent, OutrosStent, TipodeStent, Comprimento, Diâmetro, PressãoFinal, Ultrason, Resultado.</p>
FatorConversão	<p>AMB_90, AMB_92, AMB_96, CIEFAS, SAUDEBAMERINDUS, TELESC, SAUDEBRADESCO.</p>
Fornecedor	<p>Código, Fornecedor, Contato, Rua, NumerodaRua, Adicional, Estado, Bairro, Telefone, Cidade, CEP, Observações, CGC, InscriçãoEstadual.</p>

Hospital	Código, Diretor, Hospital, Rua, NumerodaRua, Adicional, Estado, Bairro, Telefone, Cidade, CEP, Observações.
Laudo	CódigoLaudo, NúmeroLaudo, Observações, Codigo.
MateriaiseMedicamentos deSalaUsados	CódigoMateriaiseMedicamentosdeSalaUsados, Codigo, Código_Tabela_n, Descrição_n, Apresentação_n, Quantidade_n, Valor_Unitário_n.
Material	Código, Código_Tabela, Descrição.
MaterialUtilizado	CódigoMaterialUsado, Codigo, Código_Tabela_n, Descrição_n, Valor_Unitário_n, Fornecedor_n.
Medicamento	Código, Código_Tabela, Descrição, Apresentação, Valor_Unitário.
MedicamentoUsado	CódigoMedicamentoUsado, Codigo, Código_Tabela_n, Descrição_n, Apresentação_n, Quantidade_n, Valor_Unitário_n.
Medico	Código, CRM, Título, Especialidade, Nome, Clinica, Rua, NumerodaRua, Adicional, Estado, Bairro, Telefone, Cidade, CEP, Observações.
MédicosHemodinâmica	Código, CRM, Nome.
Orcamento	CódigoOrçamento, Codigo, NúmeroOrçamento, MateriaisMedicamentos, MateriaisAltoCusto, TaxadeSala, ServiçosProfissionais, Prótese, Desconto%, DescontoR\$, Observações, DataCadastro.
Paciente	Codigo, Titulo, Paciente, Numero, DataExame, Condiçao, DataNascimento, Sexo, Idade, Convenio, Rua, NumerodaRua, Adicional, Estado, Bairro, Telefone, TelefoneEmpresa, Empresa, Profissão, Cidade, CEP, ViadeAcesso, Procedimento, MedicoSolicitante, Observações, Gráfico, Emergência.
SAUDEBAMERINDUS	Código, Tabela, Descrição, Total, NúmerodeAuxiliares
SAUDEBRADESCO	Código, Tabela, Descrição, Total_CRRCH, NúmerodeAuxiliares.
TaxadeSala	Código, Tabela, Descrição, Total.
TELESC	Name, Código, Tabela, Descrição, Total, NúmerodeAuxiliares.

7 IMPLEMENTAÇÃO

Com o término da fase de análise de sistemas passou-se a implementação. Levando-se em consideração que este sistema precisa ser utilizado por pessoas que utilizam ambientes gráficos em que a performance é o fator prioritário, optou-se por desenvolver o Sistema em *Visual Basic* com o *Microsoft Access* como gerenciador de banco de dados.

7.1 APRESENTAÇÃO DAS TELAS

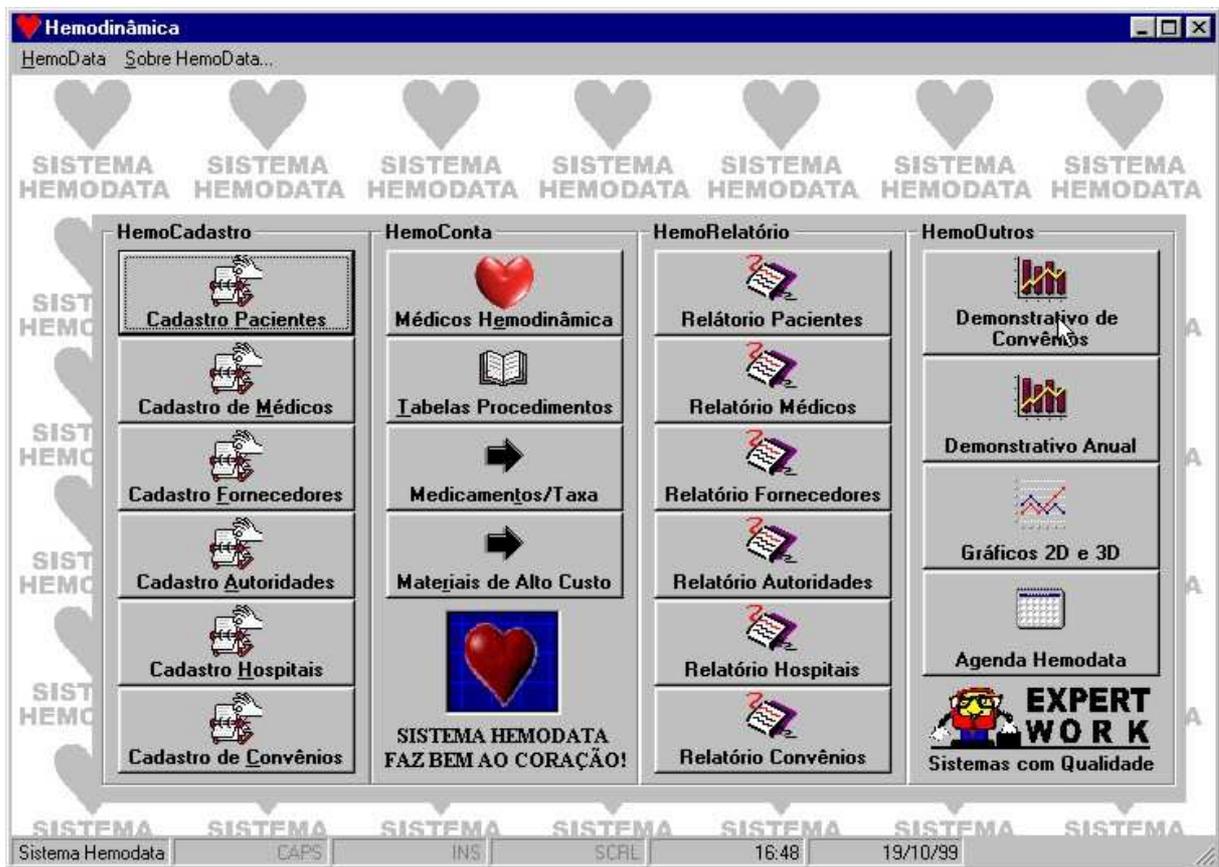
Na figura 17 é solicitada a senha para utilização do sistema, impedindo usuários não autorizados.

Figura 17. Tela de entrada do sistema.



A figura 18 apresenta a tela de opções para o usuário, o sistema é dividido em quatro seções HemoCadastro, HemoConta, HemoRelatório e HemoOutros, estas seções tem por objetivo facilitar a compreensão do usuário pois ela transmite a realidade dos cadastros da clínica hemodinâmica, organizados por prioridade de utilização.

Figura 18. Tela principal.



A figura 19 apresenta a tela de cadastro de pacientes, nesta janela é cadastrado o dado relevante à clínica do paciente, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL*.

Figura 19. Tela de cadastro de pacientes.

Cadastro de Pacientes

PACIENTES | Orçamento | Conta | Laudo | Entidade | Sair

Entrar na Estatística
 Emergência

Pacientes

- ♥ MARLI TERESA LOTH | 18/09/97
- ♥ MARTA DA ROCHA | 02/10/97 | 5
- ♥ MARTA JURACI CIPRIANI SLOMS
- ♥ MARTA RUTH ANTON | 29/03/99
- ♥ MARTINHA DA SILVA | 26/03/98
- ♥ MATIAS LESCOVITZ | 13/04/99 |
- ♥ MATIAS MENDES | 12/04/99 | 13
- ♥ MATILDE ZICK | 23/03/99 | 1298
- ♥ MAURÍCIO MASSANEIRO | 21/07
- ♥ MAURÍCIO MASSANEIRO | 28/10
- ♥ MAURINA BELLI | 05/12/97 | 634
- ♥ MAURINA BELLI | 05/12/97 | 633
- ♥ MAURINA CORREA CUSTÓDIO |
- ♥ MAURINO OSCAR RAINERT | 08.
- ♥ Mauro Hinz | 22/09/98 | 1005
- ♥ MAURO MOSER | 13/08/97 | 502
- ♥ MAURO MOSER | 13/08/97 | 503
- ♥ MAX HEINIG | 12/03/97 | 141
- ♥ MAX HEINIG | 18/06/97 | 416
- ♥ MAX HEINIG | 20/03/97 | 165
- ♥ MAZILDA VICENTE | 09/04/97 | 2
- ♥ MELITO JOÃO VIEIRA | 13/07/99

Título Sr. Imprimir Janela

Nome do Paciente Mauro Hinz

Número do Paciente 1005 **Data Exame** 22/09/98

Condição Ambulatorial **Anos** 21

Data de Nascimento 02/07/78 **Sexo** Masculino

Convênio PARTICULAR

Procedimento AORTOGRAFIA

Via de Acesso PUNÇÃO

Médico Solicitante Dr. Humberto Rebello

Endereço do Paciente

Rua DUQUE DE CAXIAS **818** **CEP** 89140-000

Adicional - **Bairro** CENTRO **Estado** SC

Telefone (047)357-3476 **Cidade** Ibirama

Empresa - **Telefone** () -

Profissão -

Observações do Paciente

357-2147
AORTOGRAFIA PULMONAR + ARTERIOGRAFIA RENAL BILATERAL

Adiciona | Retira | Atualiza | Grava | Planilha | Etiqueta

Record: 1

A figura 20 apresenta a tela de cadastro de médicos, nesta janela é cadastrado o dado relevante dos médicos conhecidos e consultados pela clínica, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL*.

Figura 20. Tela de cadastro de médicos.

A figura 21 apresenta a tela de cadastro de fornecedores, nesta janela é cadastrado o dado relevante dos fornecedores conhecidos e consultados pela clínica, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL*.

Figura 21. Tela de cadastro de fornecedores.

A figura 22 apresenta a tela de cadastro de autoridades, nesta janela é cadastrado o dado relevante das autoridades conhecidas e consultados pela clínica em caso de necessidade, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL*.

Figura 22. Tela de cadastro de autoridades.

A figura 23 apresenta a tela de cadastro de fornecedores, nesta janela é cadastrado o dado relevante dos fornecedores conhecidos e consultados pela clínica, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL*.

Figura 23. Tela de cadastro de hospitais.

A figura 24 apresenta a tela de cadastro de orçamento de paciente, nesta janela é cadastrado o orçamento para paciente, estes dados serão confrontados futuramente com a conta gerada pelo sistema, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL* e um componente *ACTIVE X* da *Seagate Crystal Report* para geração do relatório do orçamento (figura 25).

Figura 24. Tela de cadastro de orçamento para o paciente.

Cadastro de Orçamentos

 **CADASTRO DE ORÇAMENTOS**

Orç. Núm.	1	Cód. Pac.	2360
Paciente	Mauro Hinz		
Procedimento	AORTOGRAFIA		
Médico	Dr. Humberto Rebello		
Via de Acesso	PUNÇÃO		

Orçamento A		Orçamento B	
Data Cadastro	<input type="button" value="Hoje"/> 19/10/1999	Prótese	1000
Materiais e Medicamentos	2000	Desconto R\$	2000
Materiais de Alto Custo	100	Desconto %	5
Taxa de Sala	100	Valor Total	1.135,00
Serviços Profissionais	100		

Observações do Orçamento

Neste orçamento houve desconto, pois, o mesmo será repassado como serviço de terceiro.

Figura 25. Tela de impressão do cadastro de orçamento para o paciente.

Orçamentos

1 of 1+

50%

Preview

Data de Impressão: Terça-feira 19/10/1999

BEMODATA - ORÇAMENTO
 Santa Isabel Intercapital - Serviço de Radiodiagnóstico e Intercapital

Nº /

Paciente: **MATILDO BINZ**

Procedimento: **AORTOCCLAFIA**

Fio de Aço: **PINÇAO**

Médico Solicitante: **DR. HUMBERTO REBELLO**

Data Orçamento: **22/10/99**

Orçamento válido por 30 dias

Medicamentos e Materiais	R\$	2 000,00
Medicamentos de Alta Custo	R\$	100,00
Taxa de Sala	R\$	100,00
Serviços Profissionais	R\$	100,00
Análise	R\$	1 000,00
Desconto em %		5%
Desconto em R\$	R\$	2 000,00
Total	R\$	1.135,00

Clas T. Moura
 Superadora Administrativa

Página 1

HOSPITAL SANTA ISABEL - Rua Marechal Floriano Peixoto, 300 - 4º Andar - CEP 89010-906
 Fone/Fax: (047) 326-9690 - 326-5455 - Marçal 286 - Blumenau - SC

A figura 26 apresenta a janela de conta hospitalar do paciente, nesta janela é cadastrado os procedimentos utilizados, materiais e medicamentos(ver figura 27), materiais de alto custo(ver figura 28), médicos e taxas diversas, estes dados em conjunto serão utilizados para a geração da conta hospitalar, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL* e a um componente *ACTIVE X* da *Seagate Crystal Report* para geração do relatório da conta hospital para o paciente.

Figura 26. Tela de cadastro da conta hospitalar do paciente.

Cadastro de Conta Hemodinamissista

CONTA HOSPITALAR Imprimir Janela Sair

Paciente: **Mauro Hinz**

Condição: **Ambulatorial** Data Exame: **22/09/98**

Procedimento: **AORTOGRAFIA**

Solicitante: **Dr. Humberto Rebello** Via: **PUNÇÃO**

Tabela de Conta: **AMB-92** Fator de conversão: **0,27** Recalcular

Código	Descrição do Procedimento	Quantidade	Valor	Ação
33.33.330-2	Angiografias por cateter	x 2	R\$ 162,00	Limpa
40.08.035-8	Angiografia por subtração digital com injeção arterial	x 50 %	R\$ 162,00	Limpa
32.10.001-9	Angiografia carotídea unilateral (punção direita)	x 50 %	R\$ 135,54	Limpa

CRM - Hemodinamissista: 7340 - Dr. Newton Stadler de Souza Filho

CRM - Aux. 1: 20 % 4923 - Dr. Constantino Constantini Limpa

CRM - Aux. 2: 30 % 7466 - Dr. Virgílio DeMares Limpa

Ex. Complementares: **Medicamentos e Materiais de Sala** Cadastrar Materiais de Alto Custo

40.09.004-3	Angioplastia transluminal coronária com suporte por circulação	x 100 %	R\$ 405,00	Limpa
40.09.027-2	Atrioseptostomia por cateter balão ou lâmina	x 100 %	R\$ 216,00	Limpa

Taxa de Sala

02.01.006	AHESC	x 100 %	R\$ 203,60	Limpa
-----------	-------	---------	------------	-------

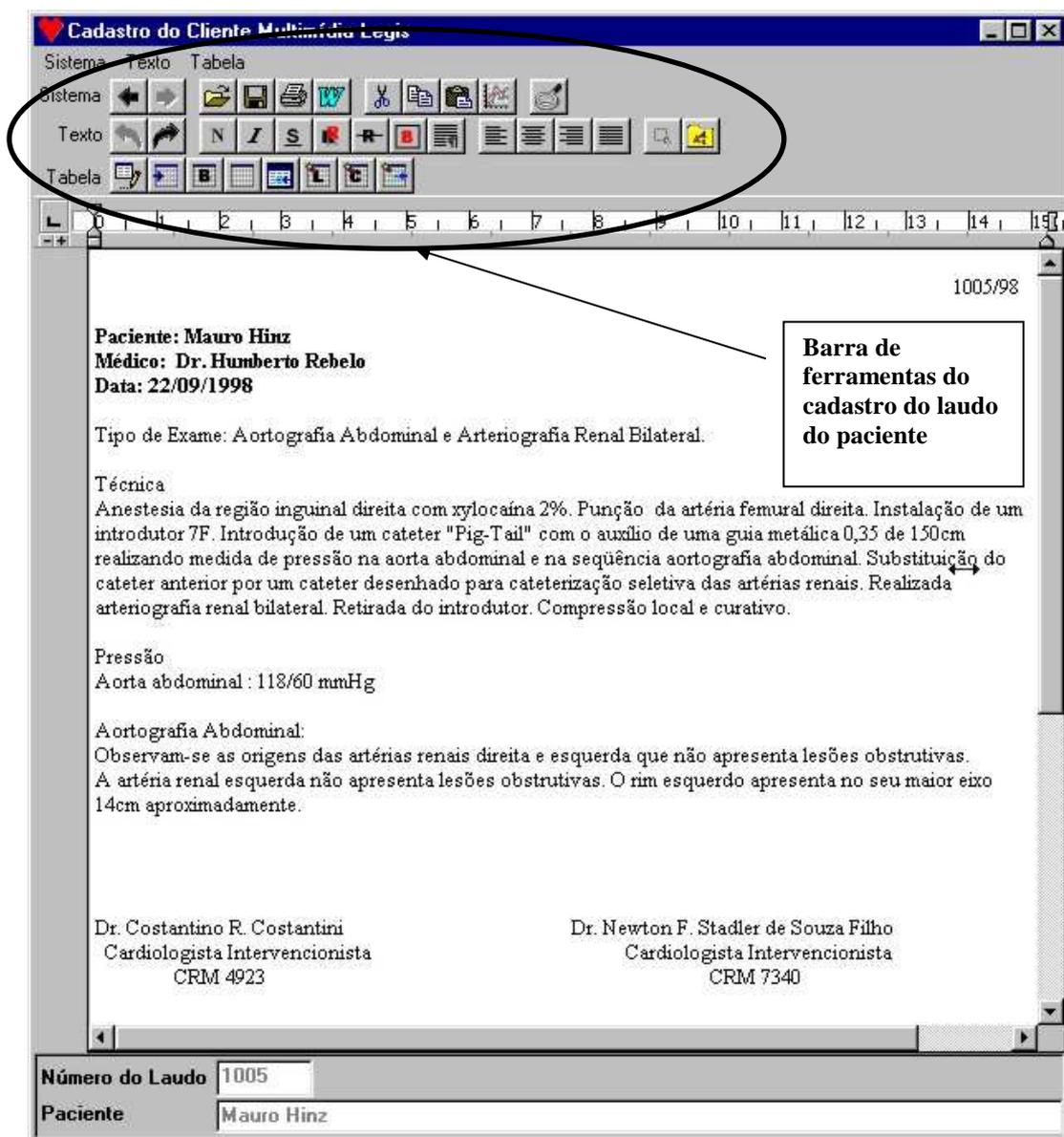
MAC 20 % Taxa Adm. 10 % Desc. Honorário 5 % Desc. Geral 5 % Desc. R\$ R\$ 137,50

Exames pós operatório serem obrigatórios!

Atualiza Grava Gerar Conta

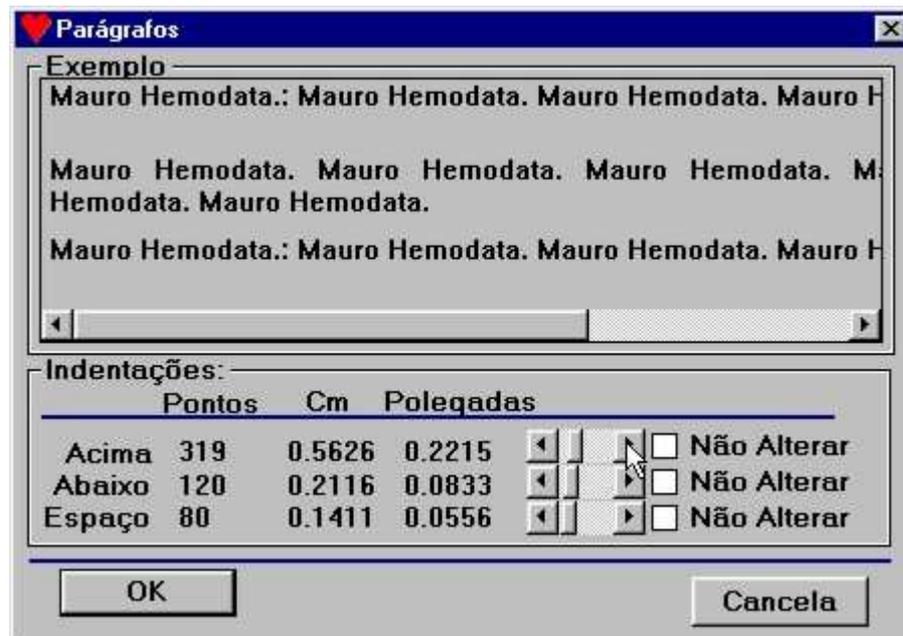
A figura 29 apresenta a tela de laudo do paciente, nesta janela é cadastrado o laudo do paciente, é possível fazer diversas formatações de texto, visualizar a impressão(figura 30), transportar para o *Microsoft WORD* o laudo e vice-versa, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL* e a um componente *ACTIVE X* da “*The Bennet-Tec All-Text Technology*” para gravação do laudo em um campo memorando no banco de dados.

Figura 29. Tela de cadastro do laudo para o paciente.



A figura 30 apresenta a tela de edição de parágrafos existente na barra de ferramentas do cadastro do laudo do paciente esta tela é útil para configurações de editoração eletrônica, nesta tela foi utilizado tecnologia básicas de programação estruturada sobre o componente Active X da “The Bennet-Tec All-Text Technology”.

Figura 30. Tela de configuração de parágrafos.



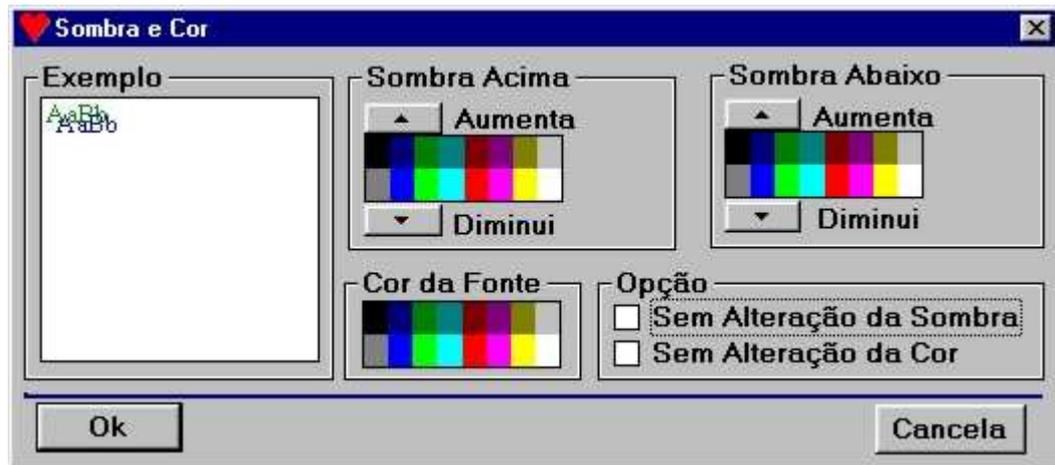
A figura 31 apresenta a tela de edição de bordas existente na barra de ferramentas do cadastro do laudo do paciente esta tela é útil para configurações de editoração eletrônica, nesta tela foi utilizado tecnologia básicas de programação estruturada sobre o componente Active X da “The Bennet-Tec All-Text Technology”.

Figura 31. Tela de configuração de bordas.



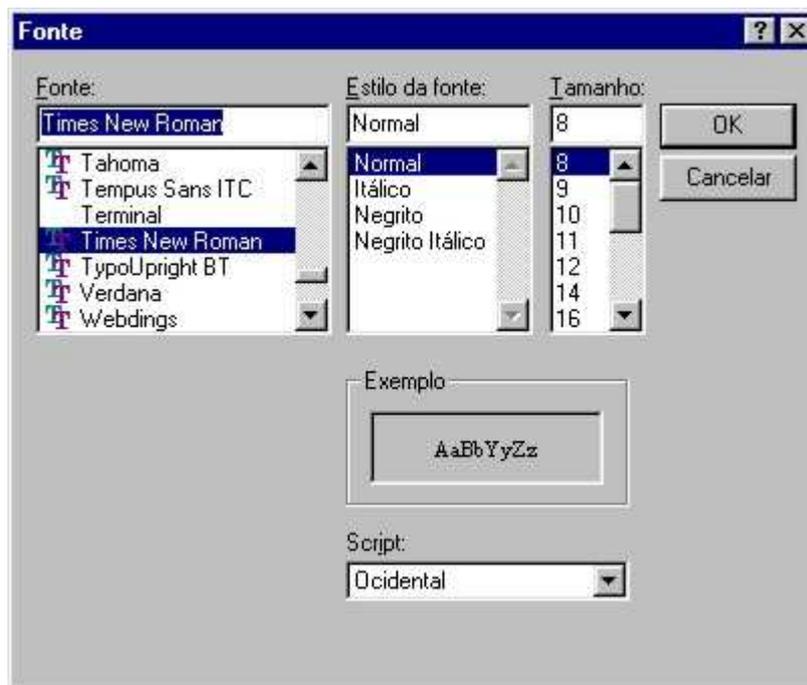
A figura 32 apresenta a tela de edição de sombra e cor existente na barra de ferramentas do cadastro do laudo do paciente esta tela é útil para configurações de editoração eletrônica, nesta tela foi utilizado tecnologia básicas de programação estruturada sobre o componente Active X da “*The Bennet-Tec All-Text Tecnologia*”.

Figura 32. Tela de configuração de sombra e cor.



A figura 33 apresenta a tela de edição de fonte existente na barra de ferramentas do cadastro do laudo do paciente esta tela é útil para configurações de editoração eletrônica, nesta tela foi utilizado tecnologia básicas de programação estruturada sobre o componente Active X da “*The Bennet-Tec All-Text Tecnologia*”.

Figura 33. Tela de edição de tipo de fonte.

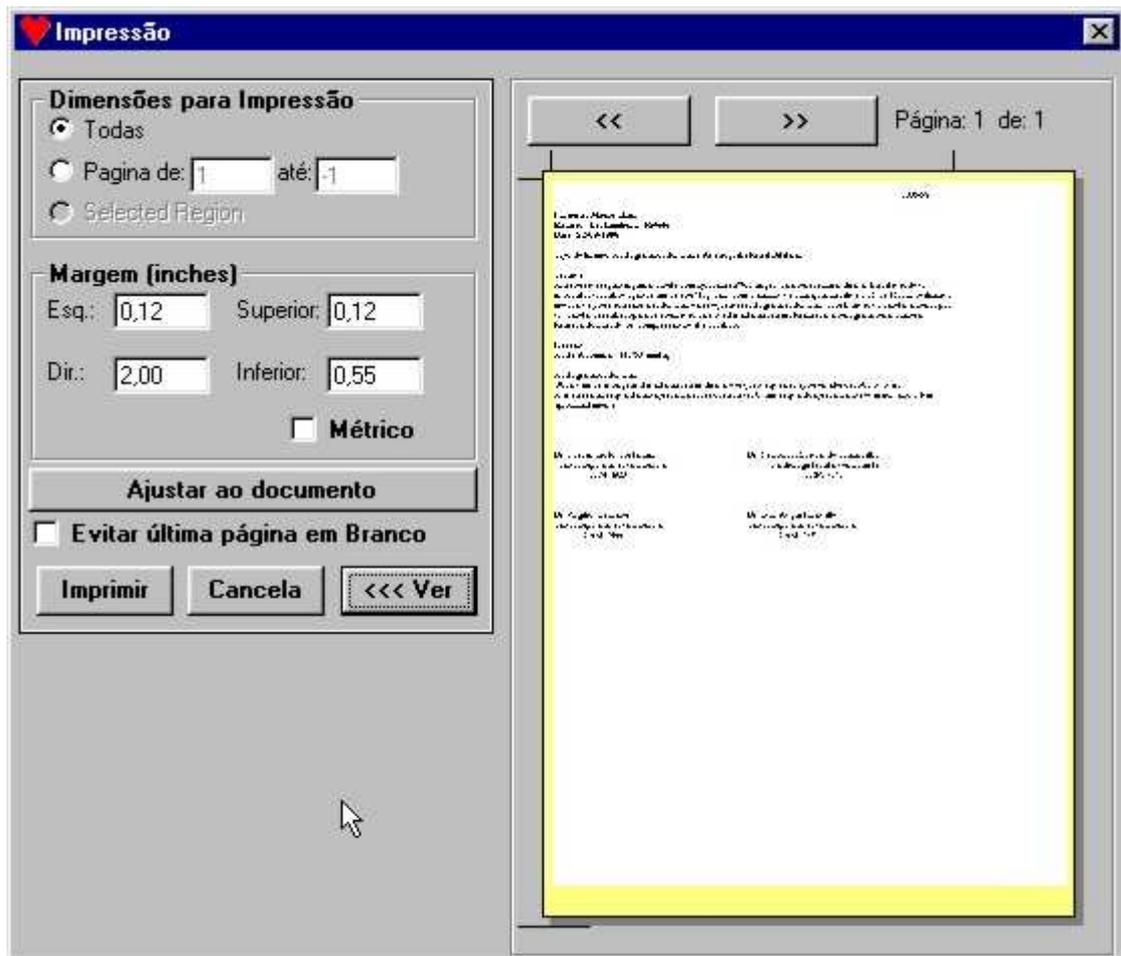


A figura 34 apresenta a tela de edição de tabelas existente na barra de ferramentas do cadastro do laudo do paciente esta tela é útil para configurações de editoração eletrônica, nesta tela foi utilizado tecnologia básicas de programação estruturada sobre o componente Active X da “The Bennet-Tec All-Text Technology”.

Figura 34. Tela de edição de tabelas.



Figura 35. Tela de impressão do cadastro do laudo para o paciente.



A figura 36 apresenta a janela de cadastro de entidades hemodinâmicas do paciente, são informações relevantes ao cardiologista, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL* e a um componente *ACTIVE X* da empresa *Seagate Crystal Report* para geração do relatório para o cardiologista.

Figura 36. Tela de cadastro de entidades do paciente para o cardiologista.

Cadastro de Entidades

 **CADASTRO DE ENTIDADES** Imprimir Janela Sair

Cód. Paciente: 2360 Exame: 22/09/98

Paciente: Mauro Hinz

Procedimento: AORTOGRAFIA

Médico Solicitante: Dr. Humberto Rebello

Via de Acesso: PUNÇÃO

Quadro Clínico | Fatores de Risco | Cat Cardíaco | **Cateterismo Cardíaco** | Alto Custo | ATC | Rotablator | Stent Coronário

- Cateterismo Cardíaco -

Pressões	AO	3	/	23	mmHg	VD	43	/	25	/	2	mmHg
	VE	4	/	5	mmHg	CP	62					mmHg
	Grad. Valvar	67			mmHg	AP	65					mmHg
	Grad. Mitral	68			mmHg	AD	46					mmHg

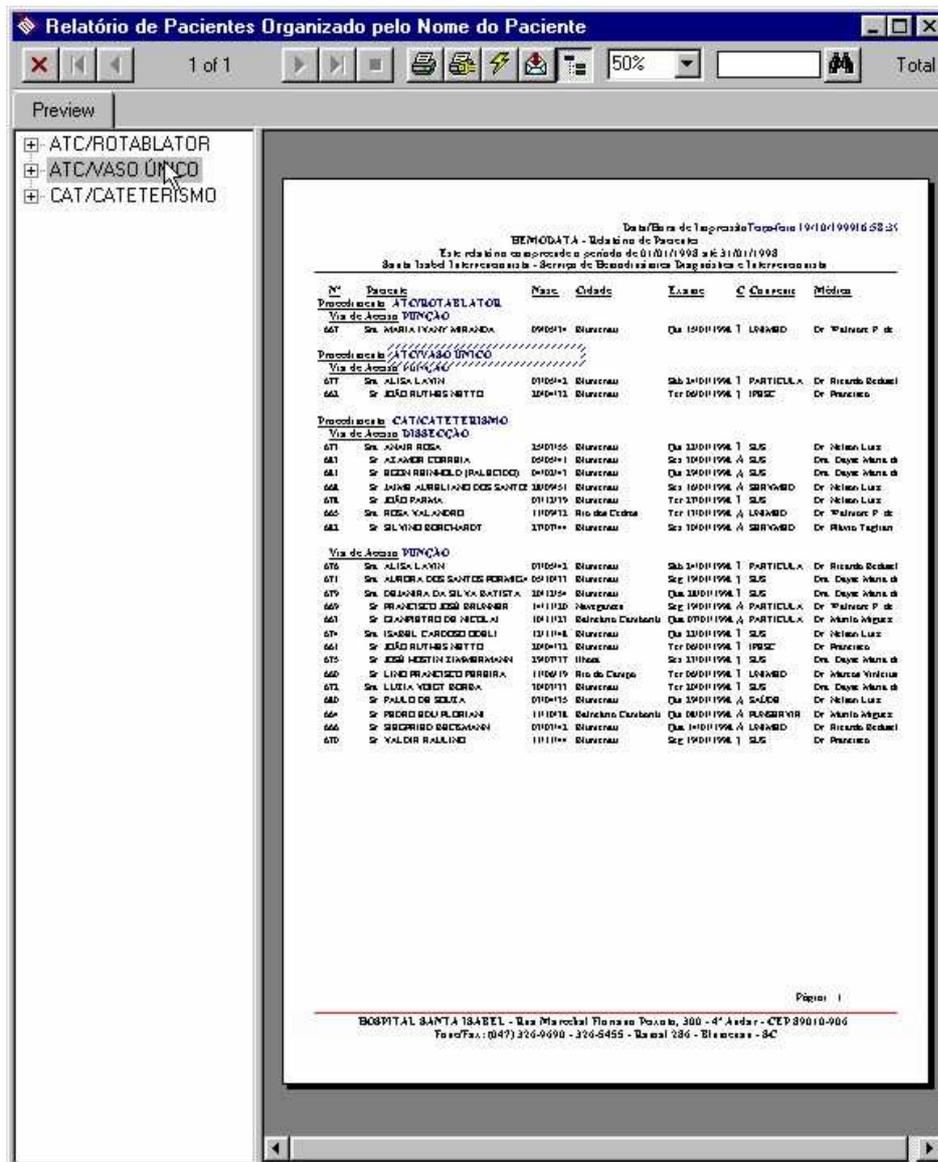
IMPRIMIR **Açualiza** **Grava**

A figura 37 apresenta a janela de opções de relatórios, nesta janela foi utilizada a tecnologia de programação SQL e a um componente *ACTIVE X* da empresa *Seagate Crystal Report* para geração do relatório (figura 38) para a hemodinâmica.

Figura 37. Tela de seleção do tipo de relatório do paciente.



Figura 38. Tela visualização da impressão do tipo de relatório do paciente.

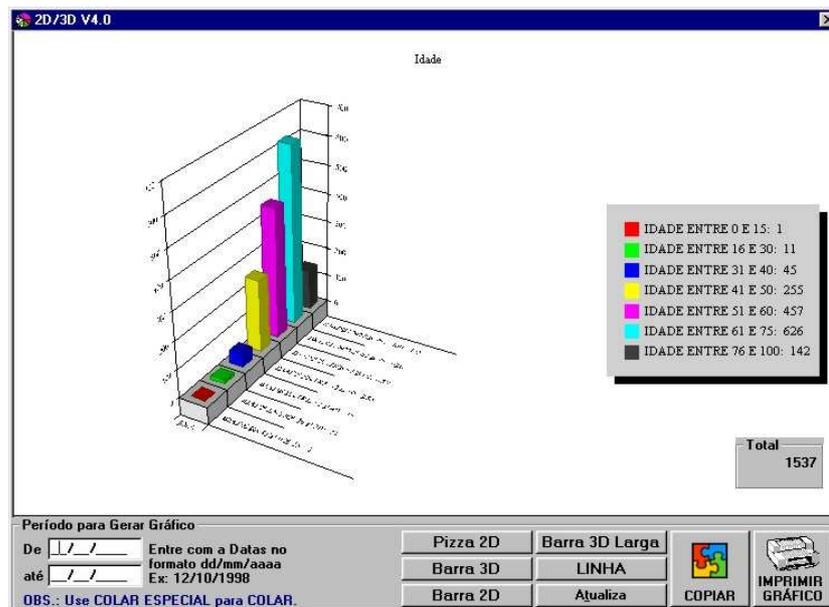


A figura 39 apresenta a tela de opções de gráficos do cadastro de paciente, nesta janela foi utilizado a tecnologia de programação *SQL* ligado a um componente *DATACONTROL* e a um componente gráfico *ACTIVE X* da empresa *Microsoft* para geração do gráfico(figura 40) da base de dados do paciente.

Figura 39. Tela de seleção do tipo de gráfico do paciente.



Figura 40. Tela de impressão do tipo de gráfico do paciente.



8 CONCLUSÕES

Após o término do desenvolvimento do sistema deste trabalho o sistema tornou-se viável devido às facilidades de operação e uniformidade, proporcionando ao usuário destreza no uso do aplicativo, apresentando um ambiente gráfico de qualidade o qual disponibiliza de ícones e botões de fácil aprendizado.

O *Bennet All Text Technology* como um componente *ACTIVE X* de gerenciador de textos possibilitou a implementação de hipertexto com utilização do campo memorando da base de dados. Possui exemplo e funções pré-programadas que facilitaram a construção do editor de laudo do paciente. A programação deste componente foi complexa, por isto, considerou-se como ponto relevante deste trabalho e vale ressaltar que a utilização deste componente exige conhecimento em programação orientada a objetos ligada a programação estruturada.

O *Seagate Crystal Report* como um componente *ACTIVE X* de gerenciador de relatórios possibilitou a implementação de relatórios profissionais, pois, possui um editor de relatório e linguagem de programação nativos com funções e princípios básicos de edição de relatórios simples ou agrupados a várias tabelas do sistema.

O *Microsoft Visual Basic 6* como compilador do sistema possibilitou a implementação das funções e princípios básicos de programação estruturada mesmo acessando objetos orientado a objetos, propriedades, funções dos componentes nativos e dos componentes *ACTIVE X*.

O banco de dados *Access* como um gerenciador de banco de dados relacional possibilitou a implementação das funções e princípios básicos que devem existir em um gerenciador de banco de dados, o que tornou o trabalho de criação, edição de tabelas intuitivo e relativamente rápido.

8.1 DIFICULDADES

- a) material, conseguir material na área de medicina cardiológica é complexo, pois, muitas informações são de extrema relevância a clínica.

- b) profissional de saúde, reunião com o profissional requer cautela, pois, nem sempre está disponível a tratar de assuntos fora da emergência clínica.
- c) profissional de administração da clínica, reunião com o profissional requer cautela e informalidade, pois, nem sempre está disponível a tratar de assuntos fora os administrativos sendo necessário levar com atenção a reunião, pois, muitos detalhes e aspirações do sistema aparecem em uma conversa.

8.2 SUJESTÕES

Os seguintes aspectos do sistema merecem considerações para versões futuras:

- a) migrando da DAO para a ADO, o acesso a base de dados usando a ferramenta *Visual Basic*, a partir da versão 6.0 do *Visual Basic* utiliza a ADO, que oferece objetos para acessar e manipular dados em um servidor ou qualquer outro tipo de informação usando a para isto a interface *OLE DB*. É através da ADO que se acessa a *OLE DB*;
- b) ASP e ADO - Colocar o Banco de Dados na *Internet*, colocar algumas informações na internet para o acesso dos cardiologistas da hemodinâmica através de seus computadores pessoais;
- c) utilizar a tecnologia *Active X* em outras áreas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BEN97] BENNET-TEC, INC. **Bennet-TEC All-TEXT Component 32 bits**. Canadá : Bennet-Tec, 1997.
- [BRO99] BROW, Steve. **Visual Basic 6 – Bíblia do Programador**. São Paulo. Berkeley, 1999.
- [BOR99] BORLAND, CORPORATION, INC. **Tecnologia ActiveX**. 1999, Endereço eletrônico : <http://www.delphibrasil.com/activex.html>.
- [CHA97] CULLENS, Chane. **Usando Visual C++ 4** : tradução de Haroldo R.J. de Macedo. Rio de Janeiro : Campus, 1997.
- [DAL97] DALFOVO, Oscar. **Benefícios para a empresa**. 1997, Endereço eletrônico : <http://www.inf.furb.rct-sc.br/~dalfovo/bdintrl.html>.
- [DAT91] DATE, C. J. **Introdução a Sistemas de Banco de Dados**. Rio de Janeiro : Campus, 1991.
- [DEM89] DeMARCO, Tom. **Análise estruturada e especificação de sistema**. Rio de Janeiro : Campus, 1989
- [FOU94] FOURNIER, Roger. **Guia prático para desenvolvimento e manutenção de sistema estruturados**. São Paulo : Makron Books, 1994.
- [GHE91] GHEZYY, Carlo. **Conceitos de linguagens de Programação** / Carlo Ghezzi e Mehdi Jazayeri; tradução de Paulo S. Veloso. Rio de Janeiro : Campus, 1991.
- [HIN97] HINZ, Mauro. **Visual Basic – Curso Básico**. Joinville : Não publicado, 1997.

- [HOL92] HOLZER, Steven. **Visual Basic** / Steven Holzer e The Peter Norton Computing Group; tradução Edidata Publicações de Informática, Daniel Vieira. Rio de Janeiro : Campus, 1992.
- [KEL90] KELLER, Robert. **Análise Estruturada na Prática: Desmistificando Mitos**. São Paulo : McGraw-Hill, 1990.
- [MAC91] MARTIN, James; McCLURE, Carma. **Técnicas Estruturadas e CASE**. São Paulo : Makron, McGraw-Hill, 1991.
- [MCM91] McMENAMIN, Stefen M.; PALMER, John. **Análise Essencial de Sistemas**. São Paulo : McGraw-hill, 1991.
- [MEL90] MELENDEZ FILHO, Rubens. **Prototipação de Sistemas de Informação**. Rio de Janeiro : LTC, 1990.
- [MIC99] MICROSOFT, CORPORATION, INC. **Visual Basic nas Empresas Brasileiras**. 1999, Endereço eletrônico : <http://www.microsoft.com/BRASIL/vbasic/empresas.html>.
- [MIC99a] MICROSOFT, CORPORATION, INC. **Benchmarks**. 1999, Endereço eletrônico : <http://www.microsoft.com/BRASIL/vbasic/bench1.html>.
- [MIC99b] MICROSOFT, CORPORATION, INC. **Desenvolvedores**. 1999, Endereço eletrônico : <http://www.microsoft.com/brasil/desenvolvedores.html>.
- [MIC99c] MICROSOFT, CORPORATION, INC. **Microsoft Visual Basic programming system for Windows**. USA : Microsoft Corporation, 1999.
- [OLI79] OLIVEIRA, João Carlos de Assis Ribeiro de. **Desenvolvimento de Software de Banco de Dados**. São Paulo : Edgard Blücher, 1979.
- [PER97] PERRY, Greg M. **Usando Visual J++**. Rio de Janeiro : Campus, 1997.
- [SAL93] SALEMI, Joe. **Guia PC Magazine para Banco de Dados Cliente/Servidor**. Rio de Janeiro : Infobook, 1993.

- [SEA99] SEAGATE TECHNOLOGY, INC. **Seagate Crystal Reports 6**. Canadá : SEAGATE, 1999.
- [SEA99a] SEAGATE TECHNOLOGY, INC. **Seagate Tecnology**. 1999, Endereço eletrônico : <http://www.seagate.com>.
- [TOM95] TOMAZINNI, Carlos. **Manual de Procedimentos Hemodinamicistas**. São Paulo : Procor, 1995.
- [VAZ99] VAZ, Maria Salete Marcon Gomes. **Estudo de Aspectos Semânticos nos Modelos de Dados**. 1999, Endereço eletrônico : <http://www.di.ufpe.br/~msm/smt/smt.html>.
- [YOU89] YOURDON, Edward. **Administrando o Ciclo de Vida do Sistema**. Rio de Janeiro : Campus, 1989.