



FUNDAÇÃO FACULDADES LUIZ MENEGHEL
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

KAÍSA ANDRÉA CAPPI

**USO DA FERRAMENTA TOGETHER INTEGRADA COM
UML NO DESENVOLVIMENTO DE UM PROTÓTIPO
PARA CONSULTÓRIO DE PSICOLOGIA**

Bandeirantes - PR
2006

FUNDAÇÃO FACULDADES LUIZ MENEGHEL

Instituição de ensino superior vinculada à Universidade Estadual do Norte do Paraná, em acordo com a Lei Estadual nº 15.300, de 28 de setembro de 2006.

KAÍSA ANDRÉA CAPPI

USO DA FERRAMENTA TOGETHER INTEGRADA COM UML NO DESENVOLVIMENTO DE UM PROTÓTIPO PARA CONSULTÓRIO DE PSICOLOGIA

Trabalho de conclusão de curso submetido à Fundação Faculdades Luiz Meneghel como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Professor Mestre Ricardo Gonçalves Coelho

KAÍSA ANDRÉA CAPPI

**USO DA FERRAMENTA TOGETHER INTEGRADA COM
UML NO DESENVOLVIMENTO DE UM PROTÓTIPO
PARA CONSULTÓRIO DE PSICOLOGIA**

COMISSÃO EXAMINADORA

Professor Mestre Ricardo Gonçalves Coelho

Professor Mestre André Luis Andrade Menolli

Professora Mestre Katia Romero Felizardo

Bandeirantes, 29 de novembro de 2006.

DEDICATÓRIA

*“A mim e minha família:
Mãe, irmãos, sobrinhos
e cunhados”.*

AGRADECIMENTOS

Agradeço a Deus por colocar esta Universidade em meu caminho, pois assim tenho a chance de me formar profissionalmente.

A minha mãe que me deu a vida, o incentivo para eu prestar o vestibular e a tudo que ela me ajudou tanto financeiramente como em suas orações.

A meu pai que mesmo não estando presente na terra sei que lá do céu está implorando para Deus me dar mais uma chance e eu poder vencer na vida.

Ao meu orientador por tantas vezes tira-lo do sério, por sua ajuda, seu incentivo, pois sem ele juro não iria conseguir obrigada por todas as broncas, sem você hoje não estaria aqui.

Aos meus professores, companheiros de sala, todos vocês são mais que amigos.

Aos meus amigos, irmãos e conselheiros, enfim a todos que torceram por mim.

A grande amiga Silvia Paschoa, pela leitura de minha monografia, pelo incentivo e por toda ajuda.

Aos amigos que de longe estiveram torcendo por mim! Obrigada família Ferdin.

EPÍGRAFE

Uma educação superior deveria preparar-nos para produzir três coisas;

Um amigo, uma idéia e uma personalidade.

THOMAS EHRLICH.

LISTA DE TABELAS

TABELA 1 – Documentação do caso de uso cadastrar paciente	24
TABELA 2 – Documentação do caso de uso agendar consulta	25
TABELA 3 – Documentação do caso de uso extend agendar consulta externa	26
TABELA 4 – Documentação do caso de uso avaliar sessão	27
TABELA 5 – Documentação do caso de uso visualizar relatório	28

LISTA DE FIGURAS

FIGURA 1 – Diagrama de caso de uso	29
FIGURA 2 – Diagrama de classe	31
FIGURA 3 – Diagrama cadastro de paciente	32
FIGURA 4 – Diagrama agendar consulta paciente	33
FIGURA 5 – Diagrama agendar consulta pessoas externas	34
FIGURA 6 – Diagrama sessão	35
FIGURA 7 – Diagrama relatório	36
FIGURA 8 – Modelagem banco de dados	38
FIGURA 9 – Tela inicial do sistema	39
FIGURA 10 – Tela cadastro	40
FIGURA 11 – Classe paciente	43
FIGURA 12 – Tela consultas	44
FIGURA 13 – Tela sessões	45

LISTA DE ABREVIATURAS E SIGLAS

ECO III	Enterprise Core Objects III
IDE	Ambiente de Desenvolvimento Integrado
OMT	Object Modeling Technique
OOSE	Object-Oriented Software Engineering
T I	Tecnologia da Informação
UML	Unified Modeling Language

RESUMO

Este trabalho teve como objetivo, desenvolver um sistema de Consultório de Psicologia, para a utilização em uma clínica na cidade de Bandeirantes – Paraná. Este estudo propõe o desenvolvimento da ferramenta *Together integrada com UML* destinada às funcionalidades básicas para possibilitar o cadastro dos clientes, suas consultas, as sessões, a verificação da melhora ou piora do paciente em relação ao tratamento. A utilização da Linguagem de Programação *Borland Delphi 2006*, proveu suporte para a utilização da Ferramenta *Together integrada com UML 2 (Unified Modeling Language* ou Linguagem de Modelagem Unificada). As etapas do desenvolvimento de um sistema foram realizadas na própria ferramenta como o levantamento de requisitos, diagrama de casos de uso, de classes e de seqüência.

ABSTRACT

This work had as objective, to develop a system of Doctor's office of Psychology, for the use of a clinic in the city of Bandeirantes - Paraná. This study it considers the development of the Together modeling integrated with UML destined to the basic functionalities to make possible I register in cadastre it of the customers, its consultations, the sessions, the improvement or worsening of the patient. The use of the tool Borland Delphi 2006, to provide support for the use with the Together Modeling integrated with UML 2 (Unified Modeling Language or Linguagem de Unified Modelagem). The stages of the development of a system had been carried through in the proper tool as the survey of requirements, diagram of use cases, of classrooms and sequence.

SUMÁRIO

1 Introdução	12
1.1 Justificativa	14
1.2 Objetivos	14
1.3 Organização do Texto	15
2 Revisão de Literatura	16
2.1 Análise de Requisitos	16
2.2 Linguagem de Programação Delphi e Ferramenta Together	18
2.3 UML	20
3 Metodologia	22
3.1 Descrição dos Métodos	22
4 Modelagem do Protótipo Proposto	24
4.1 Diagrama de Uso de Caso	24
4.2 Diagrama de Classe	30
4.3 Diagramas de Seqüência	32
4.4 Modelagem Banco	37
4.5 Interface do Protótipo	39
4.5.1 Tela Inicial do Sistema	39
4.5.2 Tela Cadastro de Pacientes	40
4.5.3 Tela Consultas	44
4.5.4 Tela Sessões	45
5 Conclusão e Trabalhos Futuros	46
6 Referências Bibliográfica	47

1 INTRODUÇÃO

Nos últimos tempos o computador está invadindo todas as atividades profissionais da vida humana. Tudo o que se faz, pensa-se ou o que está ao redor do indivíduo, está envolvido com a informatização. Segundo Monteiro [2006], *“a informática é um alicerce importantíssimo no processo de melhoria. A sua evolução permite que as informações alcancem o seu destino com maior rapidez, onde quer que os clientes internos e externos estejam”*.

Lugares, pessoas e todo o universo da atividade empresarial, do comércio, do mercado de trabalho e prestação de serviços precisam da Informática. A cada minuto a informatização assume mais um lugar nesse universo. Pode-se dizer que hoje é difícil viver sem a Informática. As pessoas necessitam de mais rapidez, agilidade e praticidade. Telegramas, cartas e outros meio de comunicação mais antigos parecem estar fora da realidade, pois hoje utiliza-se a Internet, seja em casa, nos estudos e no competitivo mercado de trabalho.

Quais as áreas que ainda não se utilizam da informatização? Deve-se estar preparado para ela, senão fica-se perdido em um mundo cheio de novidades, benefícios e conforto do mundo digital.

Os documentos em papéis que precisam ser arquivados apresentam um volume muito grande, geralmente de difícil acesso imediato. Na maioria das vezes esses documentos não estão organizados e nem catalogados, podendo provocar grande demora em sua procura, causando ao usuário uma enorme frustração, perda de tempo por muitas vezes, não se encontrando os documentos.

As empresas que já estão no mundo da informatização usam CDs e outras mídias para diminuir a quantidade de documentos em papel usada diariamente. Considerando-se esta preocupação com a consulta de cada paciente, torna-se necessário a implantação de um sistema diferente e original, principalmente em

empresas com grandes quantidades de informação sigilosas de pacientes, como é o caso de um Consultório de Psicologia.

Há alguns anos atrás os psicólogos achavam que o computador era apenas uma máquina de escrever sofisticada. Hoje, grande parte deles já mudou e opinião, porém ainda não sabem como usar o computador para outra coisa além de ler e escrever *e-mail*. A partir daí surge à idéia da implantação de um protótipo de um sistema que possibilite o cadastro dos clientes, o agendamento de suas consultas, suas sessões, a melhora ou piora do paciente em relação ao tratamento, outro facilitador é a existência de um critério de organização para que qualquer documento de consulta dos clientes possa ser localizado com rapidez. Para isso, torna-se essencial o uso de um sistema com essas características com a utilização do computador.

Como suíte de modelagem de códigos da *Borland*, o *Together* ajuda a minimizar as diferenças entre os negócios e as organizações de TI através de um conjunto de linguagens de modelagem visual, fornecendo um completo suporte de aplicações de modelagem para analistas de mercado, *designers*, arquitetos e desenvolvedores de *software*. Com suporte para modelagem de processos de negócios, de dados, de aplicações e visualizações, e completas auditorias e métricas para modelos e códigos, a solução da *Borland* ajuda a acelerar a entrega de aplicações de *softwares* flexíveis e de alta qualidade. Segundo Hauck (2006) “a facilidade de manuseio do *Borland Together*® capacita organizações a implementarem modelagens e processos de design visual em seus esforços de software, o que pode significar melhorias em suas habilidades de entregar software de alta qualidade”.

O *Together* é muito mais que uma ferramenta CASE. É uma ferramenta de projeto de *software*. A avaliação foi efetuada na edição *Enterprise Architect*. O casamento *Together* e *Delphi* 2006 levou um tempo para acontecer, mas sem dúvida, tornou o *Delphi* um ambiente completo para toda a fase de análise e construção de uma aplicação Win32 ou .Net. O resultado deste casamento é o aumento da qualidade de suas aplicações sem impactar diretamente na sua produtividade, se for somadas então

as capacidades do ECO III, terá não só maior qualidade como um aumento de produtividade no desenvolvimento para .Net seja em WinForms ou ASP.Net.

Com isso pode-se dizer que o protótipo do sistema de um Consultório de Psicologia pode ao mesmo tempo, além de ser implementado, modelado ajudando assim na diminuição de tempo para o sistema ficar pronto.

1.1 JUSTIFICATIVA

Este estudo foi realizado para se verificar quais são as melhorias em se fazer a modelagem junto com a codificação, ao invés de fazer a modelagem antes e depois implementar. Com a Ferramenta *TOGHETER* terá ganho de tempo melhora na qualidade das aplicações com uma produtividade não encontrada em nenhuma outra ferramenta. A escolha deste sistema foi por achar interessante poder informatizar um consultório de psicologia, podendo minimizar o tempo da psicóloga, melhorar o rendimento de trabalho.

1.2 Objetivos

Este trabalho teve como objetivo geral desenvolver e implantar um protótipo utilizando Ferramenta *Together* Integrada com *UML* para um Consultório de Psicologia com a finalidade de cadastrar os pacientes, armazenar as informações relativas às sessões, apresentação de gráficos sobre as sessões dos pacientes, para facilitar a rotina de trabalho, de modo que a consulta possa ser feita de forma automatizada.

Os objetivos específicos deste trabalho foram:

- Realizar um estudo da viabilidade de usar a Ferramenta *Together* Integrada com *UML* para o desenvolvimento deste protótipo.

- Realizar um estudo sobre o *UML* e o que a ferramenta disponibiliza.

1.3 Organização do Texto

Esse trabalho está organizado em 6 (seis) capítulos.

No Capítulo 1 é apresentada a introdução, justificativa e objetivos.

No Capítulo 2 é apresentada uma revisão de literatura com explicação sobre análise dos requisitos, linguagem de programação *delphi* e ferramenta *together* e *uml*.

No Capítulo 3 é feita a descrição dos métodos.

No Capítulo 4 é mostrado a modelagem do protótipo proposto.

No Capítulo 5 é feita a conclusão de todos o trabalho e possível trabalho futuro.

No Capítulo 6 é mostrada as referencias bibliográfica.

2 REVISÃO DA LITERATURA

2.1 Análise de Requisitos

De base para a qualificação do conhecimento humano a informática passa a ser um estereótipo de prioridade, o instrumento indispensável para qualquer simples tarefa do cotidiano. É fácil enxergá-la no dia-a-dia, quando os olhares se voltam à mídia; aos brinquedos eletrônicos, meios de locomoção, bancos, supermercados, shoppings, magazines, indústrias (STEFANI, 2003).

O estabelecimento de uma relação direta entre o avanço tecnológico e o aperfeiçoamento profissional dos trabalhadores em geral, parece constituir um consenso. É uma idéia da qual ninguém parece duvidar (MONTE CLARO, 2005).

Muito freqüentemente clientes e engenheiros de *software* tem inconscientemente um estado de espírito de “nós e eles”. Em vez de trabalhar como equipe para identificar e refinar requisitos, cada parte define seu próprio “território” e se comunica por intermédio de uma série de memorando, declarações formais de posicionamento, documentos e sessões de perguntas e respostas. A história tem mostrado que essa abordagem não funciona muito bem (PRESSMAN, 2003).

Em reuniões com o cliente, o desdobramento de funções é usado para determinar o valor de cada função necessária ao sistema. O desdobramento da informação identifica tanto os objetivos de dados quanto os eventos que o sistema deve consumir e produzir (PRESSMAN, 2003).

Para o autor citado análise de requisitos é o primeiro passo técnico do processo de *software*. É nesse ponto que uma declaração geral do escopo do *software* é refinada para construir uma especificação completa que se torna a base de todas as atividades de engenharia de software que se seguem.

Os requisitos para um sistema de *software* estabelecem o que o sistema deve fazer e definem restrições sobre sua operação e implementação (SOMMERVILLE, 2003).

Os requisitos são obtidos no início do desenvolvimento, e o objetivo é determinar a natureza do problema do cliente (PFLEEGER, 2004).

Escrita de maneira que o cliente possa entender, a definição dos requisitos é uma listagem completa de tudo que o cliente espera que o sistema proposto faça. Ela representa um consenso entre o cliente e o desenvolvedor sobre o que o cliente precisa ou quer, e é geralmente escrita em conjunto pelo desenvolvedor e o cliente. Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar, para atingir os seus objetivos. A identificação dos requisitos é uma parte especialmente importante do processo. Deve-se utilizar uma variedade de técnicas para determinar o que os usuários e os clientes realmente querem. Não importa que métodos são escolhidos para definir os requisitos, deve-se manter um conjunto de documentos que registrem o resultado (PFLEEGER, 2004).

O objetivo da análise é coletar informações sobre os requisitos de um novo sistema, e para isso, uma das técnicas empregadas consiste em fazer uma entrevista com o usuário para adquirir essas informações.

A técnica sugere que deve-se fazer não só uma, mas várias entrevistas durante o processo de análise de requisitos ou elicitação dos requisitos do sistema para confirmar do bom entendimento por parte do analista. As entrevistas deverão ser planejadas com as perguntas certas, já prevendo algumas metas que o usuário deseja para seu sistema.

2.2 Linguagem de programação Delphi e Ferramenta Together

O *Delphi* permite exportar as configurações de uma linguagem para um arquivo e importá-las, facilitando assim para os desenvolvedores a troca de seus próprios modelos personalizados (MARCO CANTÚ, 2003)

A linguagem *Delphi* é uma extensão orientada a objetos da linguagem *PASCAL* clássica, a qual a *BORLAND* promoveu por muitos anos em seus compiladores Turbo Pascal (MARCO CANTÚ, 2003).

Os aplicativos *Windows* modernos normalmente têm diversas maneiras de fornecer um comando, incluindo itens de menu, botões de barra de ferramenta, menu de atalho. Para separar os comandos reais que um usuário pode fornecer das suas diversas representações na interface com o usuário, o *Delphi* usa a noção de ações (MARCO CANTÚ, 2003).

Ainda, conforme o autor citado, graças ao suporte automático do *Delphi*, o mesmo aplicativo executado em um sistema com um tamanho de fonte diferente muda de escala sozinho, automaticamente, sem qualquer código específico.

O *Delphi* é um ambiente de desenvolvimento de aplicações, com pouca codificação, permitindo que se construa aplicações orientadas a objetos, assim possui um ambiente de desenvolvimento fácil de usar, com as vantagens em facilidade, em alterações e implementações melhor estruturação do código e velocidade.

O *Borland Delphi Enterprise* foi desenhada para organizações de desenvolvimento de pequeno a médio porte, e para desenvolvedores individuais avançados que desenvolvam *softwares* críticos à missão, com necessidade de servidor de base de dados de alta performance (BORLAND, 2006).

Borland Delphi 2006 é uma plataforma de desenvolvimento integrado desenhada para acelerar a análise, o *design* e o desenvolvimento de aplicações corporativas complexas. Tirando máximo proveito de múltiplas plataformas com suporte aperfeiçoado a *UML™*, o *Together Developer* ajuda as equipes a acelerarem o desenvolvimento de sistemas de *software* adaptáveis e de qualidade e consiste no uso de diagramas de classe *UML* e de seqüência, capacitando as equipes de desenvolvimento a visualizar o código de modo facilitado, aumentando seu raio de visão e ampliando também a compreensão sobre a aplicação. A exclusiva tecnologia *Together LiveSource™* representa o código da aplicação através de modelos que se mantêm sempre atualizados, o que auxilia os desenvolvedores a acompanharem a geração de novo código, e durante a realização de engenharia-reversa de código de legado (BORLAND, 2006)

O *Borland Together Developer* se traduz em uma inigualável solução de modelagem centrada no código para desenvolvedores de *software*, e reduz de modo significativo a complexidade da aplicação, através do uso de diagramas *UML* de classe e de seqüência.

A tecnologia *Together* também desempenha um papel fundamental como parte da suíte de produtos para Gerenciamento e Definição de Requisitos da *Borland*, o *Caliber Analyst*, fornecendo análises comerciais com visão para requisitos técnicos e comerciais.

O *Together* disponibiliza todos os diagramas *UML* e também a confecção de DER além de alguns diagramas específicos. É possível gerar diagramas novos, ou a partir de código já existente nas linguagens suportadas (C++, C#, Java, VB6, VB.Net e XML).

2.3 UML

A *UML* (*Unified Modeling Language* - Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para modelar sistemas computacionais orientados a objeto. Nos últimos anos, a UML consagrou-se como a linguagem-padrão de modelagem adotada pela indústria de engenharia de software.

A *UML* surgiu da união de três metodologias de modelagem: o método de *Booch*, o método *OMT* (*Object Modeling Technique*) de Jacobson e o método *OOSE* (*Object-Oriented Software Engineering*) de *Rumbaugh*. Estas eram, até meados da década de 90, as três metodologias de modelagem orientada a objeto mais populares entre os profissionais da área de engenharia de *software*. A união dessas metodologias contou com o amplo apoio da *Rational Software*, que incentivou e financiou a união das três metodologias (GUEDES).

Para o autor citado cada diagrama da *UML* analisa o sistema, ou parte dele, sob uma determinada ótica, é como se sistema fosse modelado em camadas, sendo que alguns diagramas enfocam o sistema como é o objetivo do Diagrama de Casos de Uso, enquanto outros oferecem uma visão de uma camada mais profunda do software, apresentando um enfoque mais técnico ou ainda visualizando apenas uma característica específica do sistema ou um determinado processo.

Os diagramas da *UML* 2.0 dividem-se em Diagramas Estruturais e Diagramas Comportamentais, sendo que estes últimos possuem ainda uma sub-divisão representada pelos Diagramas de Interação. Os Diagramas Estruturais abrangem os diagramas de Classes, de Estrutura Composta, de Objetos, de Componentes, de Implementação e de Pacotes, enquanto os Diagramas Comportamentais englobam os diagramas de Casos de Uso, atividade, Máquina de Estados, Seqüência, Comunicação, de Interação Geral e de Tempo, sendo que estes quatro últimos correspondem aos diagramas da sub-divisão de Diagramas de Interação.

Diagrama de Casos de Uso é o diagrama mais geral e informal da UML, sendo utilizado normalmente nas fases de Levantamento e Análise de Requisitos do sistema, onde são determinadas as necessidades do usuário, embora venha a ser consultado durante todo o processo de modelagem e sirva de base para todos os outros diagramas.

Diagrama de Classes é o diagrama mais utilizado e o mais importante da UML, servindo de apoio para a maioria dos outros diagramas. Como o próprio nome diz, esse diagrama define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos possuídos por cada classe, além de estabelecer como as classes se relacionam e trocam informações entre si.

Diagrama de Seqüência preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um Caso de Uso definido pelo diagrama de mesmo nome e apóia-se no Diagrama de Classes para determinar os objetos das classes envolvidas em um processo, bem como, os métodos disparados entre os mesmos.

3 METODOLOGIA

3.1 Descrição do Método

O levantamento dos dados foi realizado em um consultório de psicologia do município de Bandeirantes, Estado do Paraná.

A implementação e modelagem do sistema foi feita utilizando a IDE Delphi 2006. O ambiente de programação utilizado foi escolhido pelo fato de apresentar uma interface gráfica amigável, de fácil manuseio e permitir o desenvolvimento através de suporte à engenharia reversa, criação rápida de protótipos e integração avançada do IDE com modelagem UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada), gerenciamento de requisitos, controle de versão, rastreamento de defeitos e colaboração em equipe. O Banco de dados utilizado foi o *Paradox*, que proporciona o armazenamento das informações.

Foi realizado um levantamento de requisitos prévios e constatado que as sessões tem duração de uma hora. Quando o paciente for adolescente ou criança, primeiramente a psicóloga entrevista a mãe ou o pai. Paciente adulto a consulta já começa com ele mesmo. Poderá ser chamada para uma entrevista alguma pessoa em específico que tenha alguma relação significativa com problemas com o paciente.

As sessões são feitas uma vez por semana, com tempo indeterminado de tratamento. Se o paciente apresentar melhora, as sessões serão de 15 em 15 dias, depois uma vez por mês e com a melhora, uma vez a cada três meses, até a cura do paciente.

A ficha do paciente consta dos seguintes dados: nome, data de nascimento, idade, sexo, ocupação, escolaridade, profissão, endereço, cidade, telefone, fax, com quem está vivendo no momento, estado civil, nome do cônjuge, religião na

infância e como adulto, indicação aos serviços da psicóloga, uso de algum medicamento, ficha médica e no final da ficha é registrada a data da primeira consulta.

A ficha é passada para a psicóloga que deverá fazer algumas perguntas do dia a dia do paciente. Inicia-se assim a primeira sessão, que indicará a forma de tratamento.

4 MODELAGEM DO SISTEMA PROPOSTO

4.1 DIAGRAMA DE USO DE CASO

O Diagrama de Casos de Uso consta somente de um ator que é a psicóloga, os casos de uso são: Cadastrar Paciente, Agendar Consulta um *extend* para Agendar Consulta Externa, Avaliar Sessão e Visualizar Relatório.

O Caso de Uso Cadastrar Paciente a Psicóloga digita todos os dados do Paciente como: nome, endereço, cidade, estado, telefone, data de nascimento, idade, sexo, ocupação, escolaridade, profissão, estado civil, nome do cônjuge, religião na infância e religião adulto.

TABELA 1 – Documentação do Caso de Uso Cadastrar Paciente

Nome do Caso de Uso	Cadastrar Paciente
Ator Principal	Psicóloga
Resumo	Este caso de uso descreve as etapas percorridas por uma psicóloga para cadastrar um paciente em seu consultório.
Pré-Condições	A Psicóloga precisa estar logada
Ações do Ator	Ações do Sistema
1 Selecionar Cadastrar Paciente	
	2 Selecionar Paciente existente ou cadastrar novo Paciente
3 Selecionar Paciente	
	4 Consultar Paciente
5 Informar dados do Paciente	
	6 Incluir dados do Paciente
7 Confirmar	
	8 Gerar Dados fornecidos e gravar
Restrições/ Validações	Todos os campos são obrigatórios

O Caso de Uso Agendar Consulta a Psicóloga agenda a consulta para o paciente: data consulta, horário, número da consulta.

TABELA 2 – Documentação do Caso de Uso Agendar Consulta

Nome do Caso de Uso	Agendar Consulta
Ator Principal	Psicóloga
Resumo	Este caso de uso descreve as etapas percorridas por uma psicóloga para agendar a consulta de seu paciente.
Pré-Condições	A Psicóloga precisa estar logada
Ações do Ator	Ações do Sistema
1 Consultar Agenda	
	2 Selecionar Dias Vagos
3 Selecionar Paciente	
	4 Informar Paciente dia Vago
5 Confirmar data e horário Consulta	
	6 Incluir Paciente
7 Confirmar	
	8 Gerar Dados fornecidos e gravar
Restrições/ Validações	Todos os campos são obrigatórios

O Caso de Uso *Extend Agendar Consulta Externa* a psicóloga agenda uma consulta com pessoas que tem algum tipo de relacionamento com o paciente em questão.

TABELA 3 – Documentação do Caso de Uso *Extend Agendar Consulta Externa*

Nome do Caso de Uso	Cadastrar Paciente
Ator Principal	Psicóloga
Resumo	Este caso de uso descreve as etapas percorridas por uma psicóloga para agendar a consulta de uma pessoa externa.
Pré-Condições	A Psicóloga precisa estar logada
Ações do Ator	Ações do Sistema
1 Consultar Agenda	
	2 Selecionar Dias Vagos
3 Selecionar Pessoa Externa	
	4 Informar Pessoa Externa dia Vago
5 Confirmar data e horário Consulta	
	6 Incluir Pessoa Externa
7 Confirmar	
	8 Gerar Dados fornecidos e gravar
Restrições/ Validações	Todos os campos são obrigatórios

Caso de Uso Avaliar Sessão a Psicóloga avalia como está o andamento das Sessões.

TABELA 4 – Documentação do Caso de Uso Avaliar Sessão

Nome do Caso de Uso	Cadastrar Paciente
Ator Principal	Psicóloga
Resumo	Este caso de uso descreve as etapas percorridas por uma psicóloga avaliar o paciente no momento da sessão.
Pré-Condições	A Psicóloga precisa estar logada
Ações do Ator	Ações do Sistema
1 Consultar Sessão	
	2 Selecionar Sessão
3 Selecionar Paciente	
	4 Informar Andamento da Sessão
5 Confirmar Dados Sessão	
	6 Incluir Dados
7 Confirmar	
	8 Gerar Dados fornecidos e gravar
Restrições/ Validações	Todos os campos são obrigatórios

Caso de Uso Visualizar Relatório a Psicóloga pode ver os gráficos das consultas se o paciente obteve alguma melhora no tratamento.

TABELA 5 – Documentação do Caso de Uso Visualizar Relatório

Nome do Caso de Uso	Cadastrar Paciente
Ator Principal	Psicóloga
Resumo	Este caso de uso descreve as etapas percorridas por uma psicóloga para Gerar um Relatório de seus pacientes.
Pré-Condições	A Psicóloga precisa estar logada
Ações do Ator	Ações do Sistema
1 Consultar Relatório	
	2 Selecionar Paciente
3 Selecionar Dados	
	4 Confirmar dados
5 Gerar Relatório	
	6 Imprimir ou visualizar Relatório
7 Confirmar	
	8 Gerar Dados fornecidos e gravar
Restrições/ Validações	Todos os campos são obrigatórios

Pode-se observar na Figura 1 o Use Case Diagram ou Diagrama de Caso de Uso.

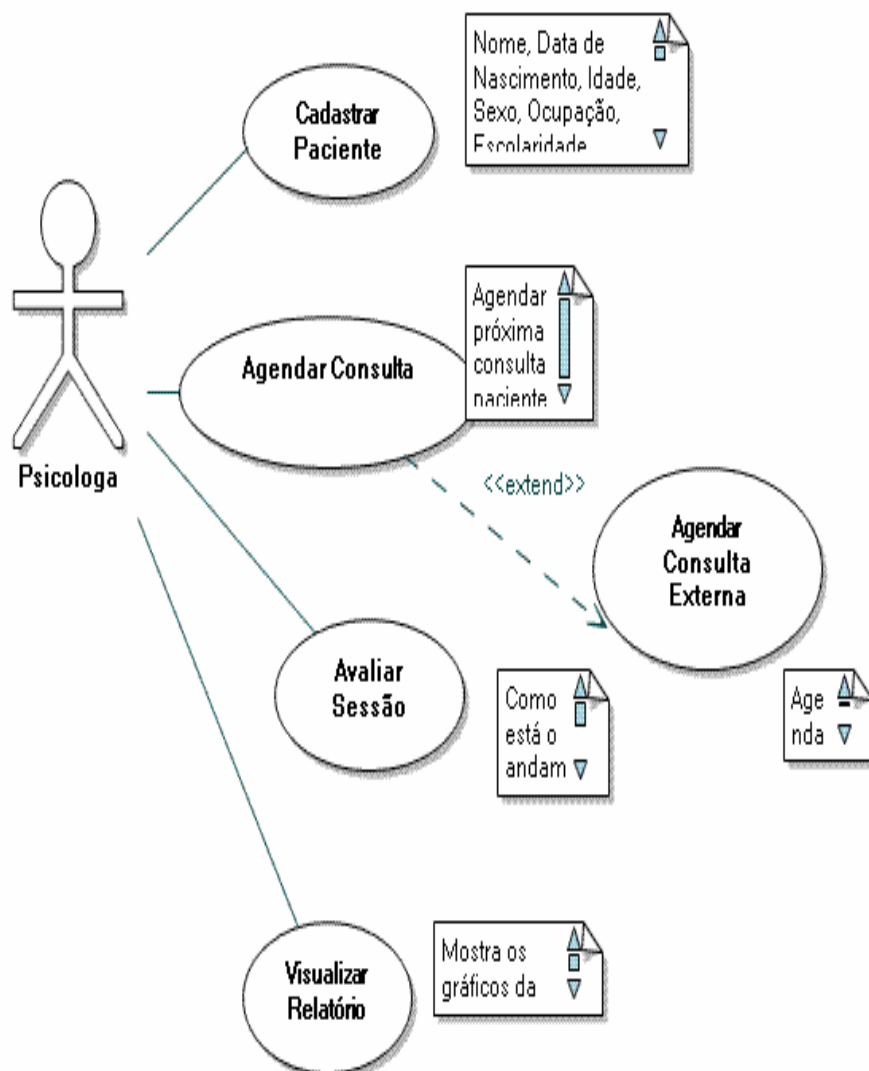


Figura 1 – Diagrama de Caso de Uso

4.2 DIAGRAMA DE CLASSE

O Diagrama de Classe está composto pelos seguintes componentes:

Classe: Paciente

Atributos: Nome, Endereço, Cidade, Estado, Telefone, Data De Nascimento, Idade, Sexo, Ocupação, Escolaridade, Profissão, Estado Civil, Nome Do Cônjuge, Religião Na Infância E Religião Adulto.

Métodos: Primeiro, Próximo, Anterior, Último, Incluir, Alterar, Excluir, Gravar e Cancelar.

Classe: Consulta.

Atributos: Data Consulta Horário, Número da Consulta.

Métodos: Primeiro, Próximo, Anterior, Último, Incluir, Alterar, Excluir, Gravar e Cancelar.

Classe: Sessões

Atributos: Nomes Citados, Nota, Entrega Paciente, Duração, Observações;

Métodos: Primeiro, Próximo, Anterior, Último Incluir, Alterar, Excluir, Gravar e Cancelar.

Classe: Encaminhamento.

Atributos: Paciente e Diagnóstico.

Métodos: Primeiro, Próximo, Anterior, Último Incluir, Alterar, Excluir, Gravar e Cancelar.

Classe: Liberar Paciente.

Atributos: Data Liberação, Diagnóstico e Paciente.

Métodos: Primeiro, Próximo, Anterior, Último, Incluir, Alterar, Excluir, Gravar e Cancelar.

Classe: Gerar Relatório.

Atributos: Estado Emocional do Paciente, Necessita de Nova Avaliação e Gráficos.

Métodos: Primeiro, Próximo, Anterior, Último, Incluir, Alterar, Excluir, Gravar e Cancelar.

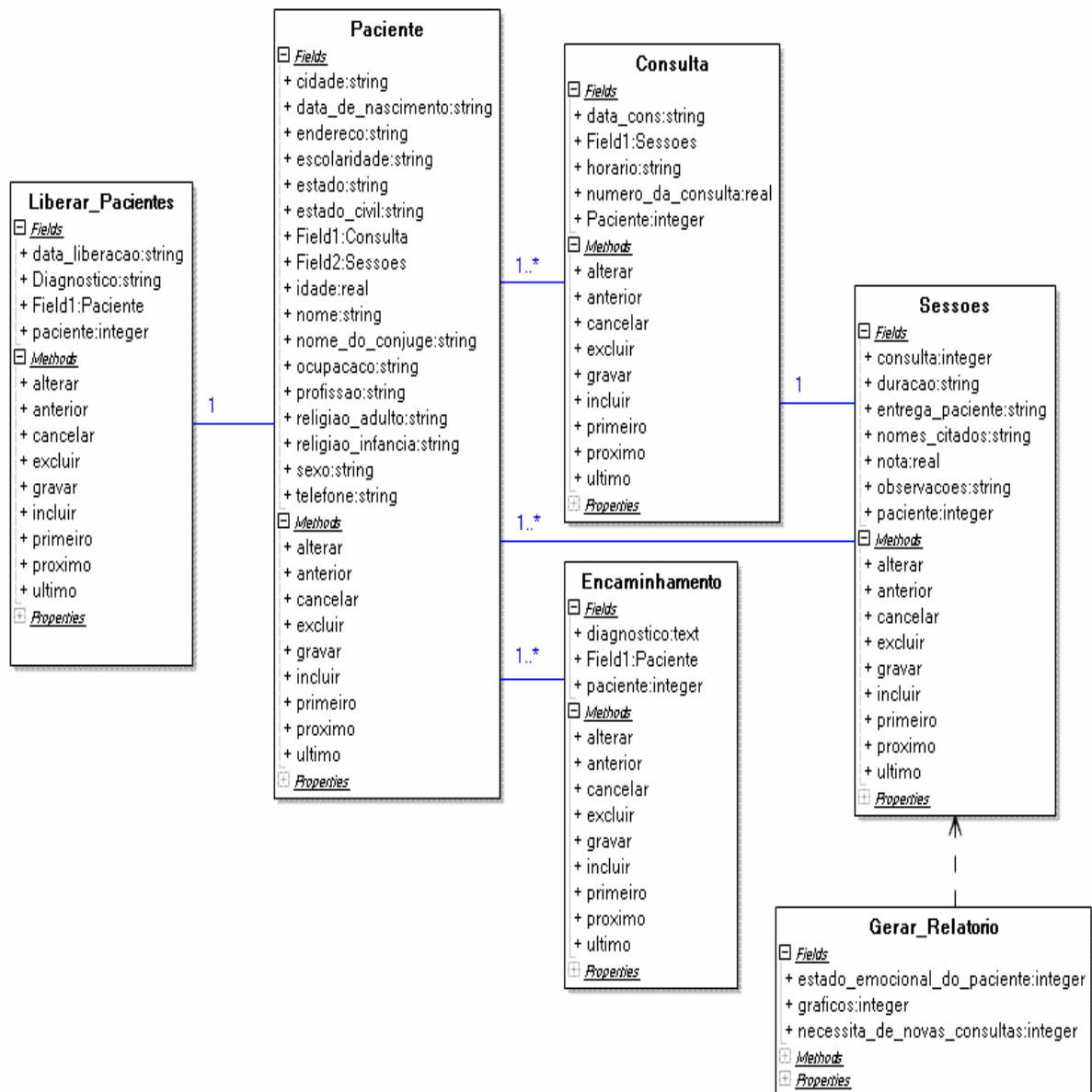


Figura 2 – Diagrama de Classe

4.3 DIAGRAMAS DE SEQÜÊNCIA

Para usar o Diagrama Cadastro de Pacientes, (Figura 3) a Psicóloga loga no Sistema para fazer uma consulta e faz o cadastro do Paciente. Se esse já for cadastrado o sistema abre a página de cadastro do Paciente, caso o contrário a Psicóloga providencia e faz o cadastro do paciente.

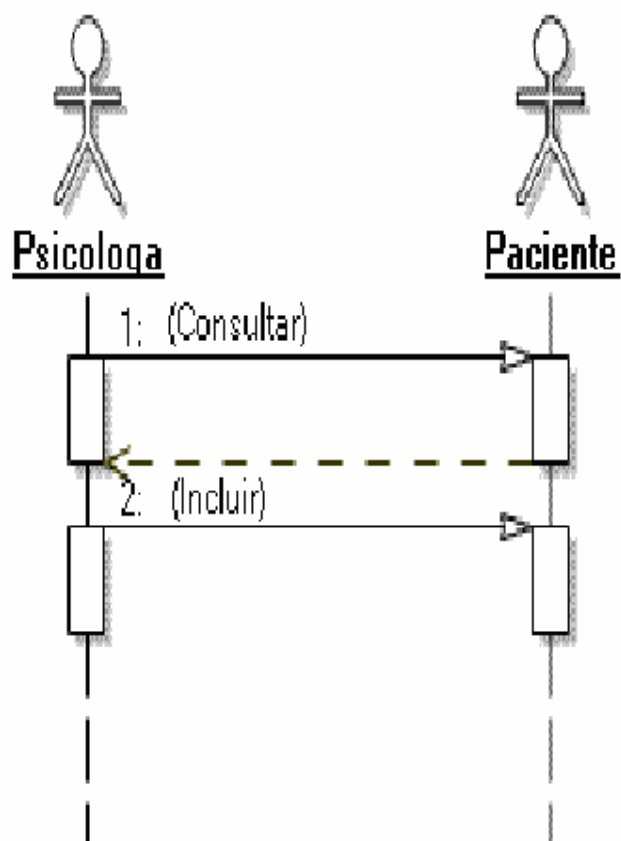


Figura 3 – Diagrama Cadastro de Paciente

Diagrama Agendar Consulta Paciente, a Psicóloga abre sua agenda e pode verificar quais as datas disponíveis para agendamento de uma consulta com o Paciente. O paciente escolhe uma data o sistema acusa se essa está disponível para agendamento ou não. Caso esteja disponível a consulta é marcada, caso a data esteja ocupada, a Psicóloga passa as datas disponíveis e a data da consulta então é confirmada. Como se pode observar na Figura 4.

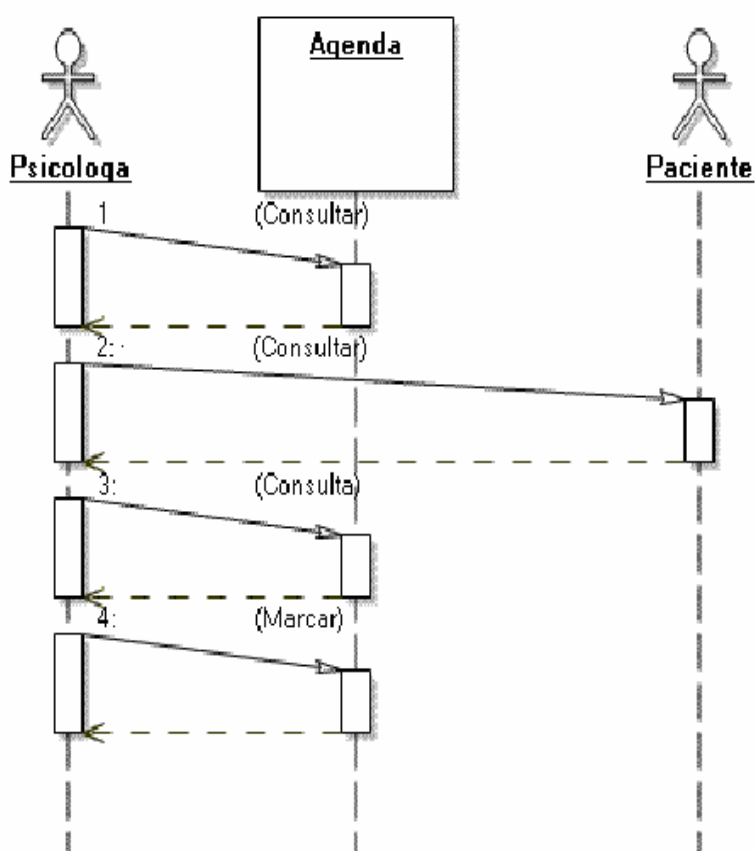


Figura 4 – Diagrama Agendar Consulta Paciente

Diagrama Agenda Consulta Pessoas Externas: durante a consulta o Paciente pode citar alguns nomes de pessoas com quem tem algum tipo de relacionamento. Caso isso ocorra, a Psicóloga agenda um horário com essa pessoa, e é verificado se esse horário pode ser confirmado, caso não, outro horário será marcado e, se estiver de acordo o horário é confirmado. Isto é demonstrado na Figura 5.

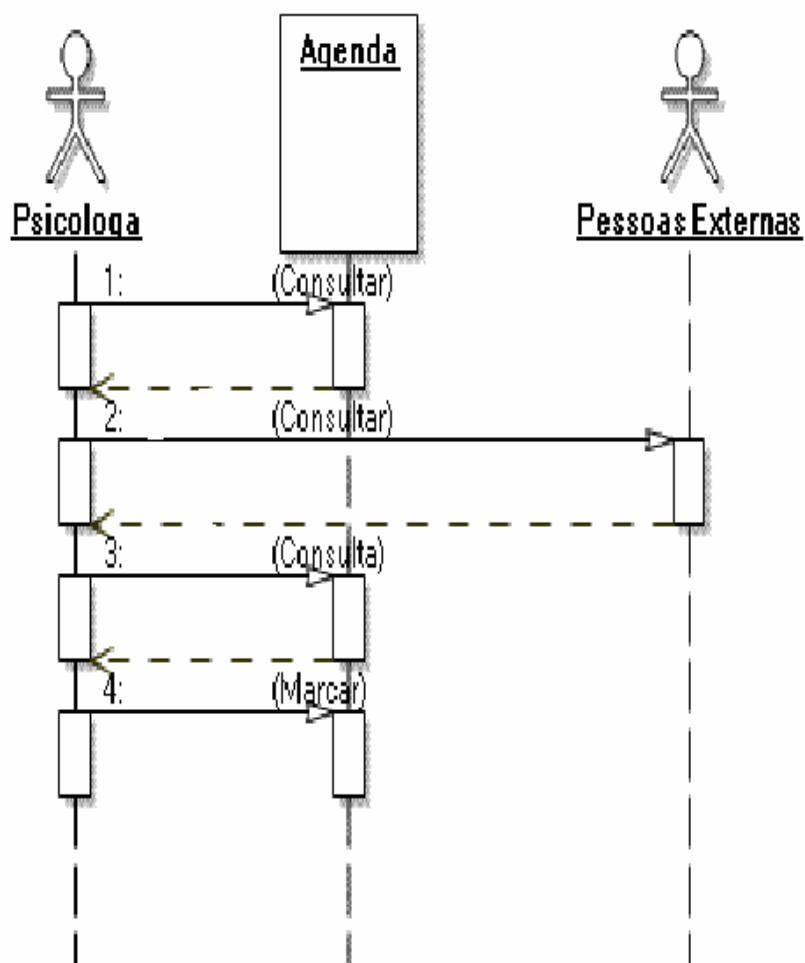


Figura 5 – Diagrama Agendar Consulta Pessoas Externas

Diagrama Sessões (Figura 6): neste diagrama a Psicóloga grava todos os dados da consulta e se estiver tudo certo os dados são armazenados.

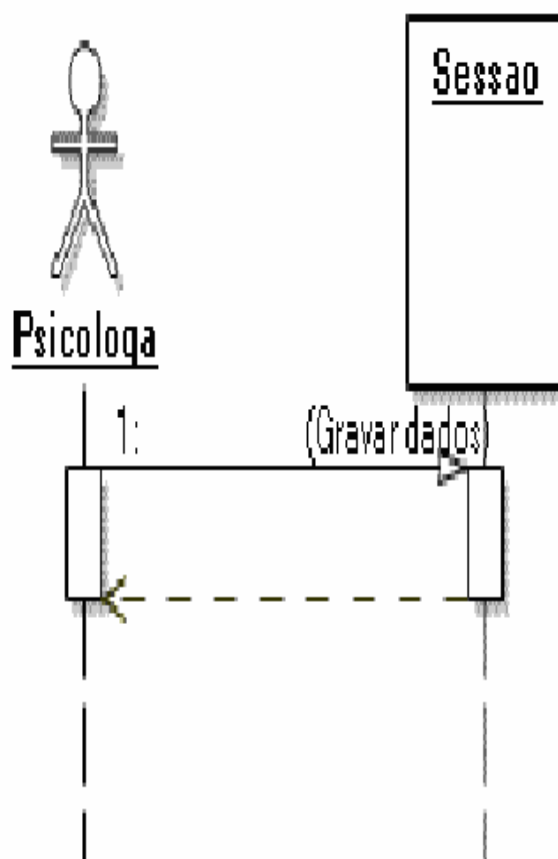


Figura 6 – Diagrama Sessões

Diagrama Relatório (Figura 7): este diagrama gera todos os dados do pacientes, gráficos, o andamento da sessão. A Psicóloga pede para esses dados sejam gerados. A geração dos dados será possível se eles forem existentes.

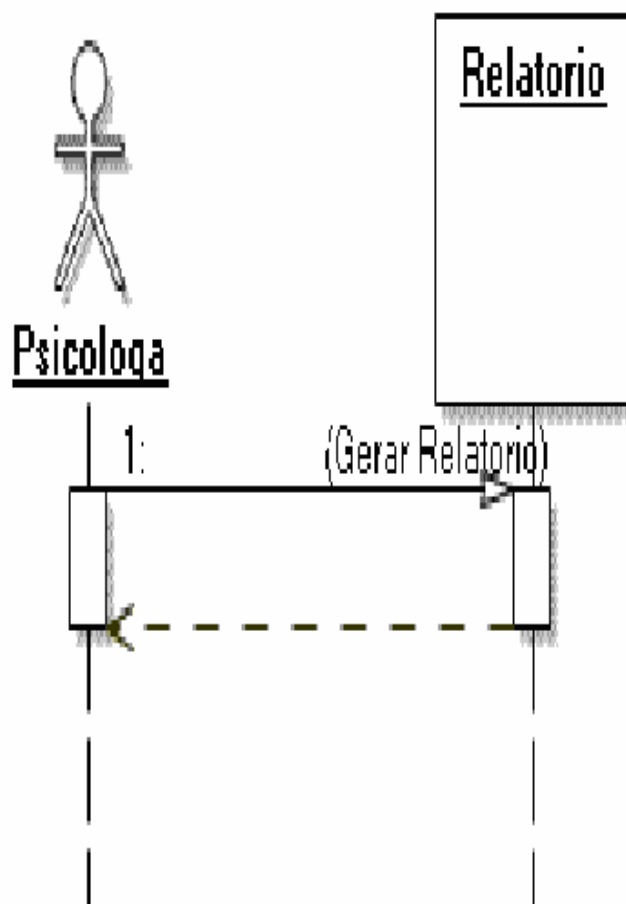


Figura 7 – Diagrama de Relatório

4.4 MODELAGEM BANCO

Foi feita a Modelagem do Banco de Dados utilizando o Diagrama de Entidade Relacionamento no *DBDesigner 4*, com as seguintes funcionalidades:

- Paciente: chave primária *idPaciente* com os seguintes atributos: Nome, Endereço, Cidade, Estado, Telefone, Data De Nascimento, Idade, Sexo, Ocupação, Escolaridade, Profissão, Estado Civil, Nome Do Cônjuge, Religião Na Infância E Religião Adulto;

- Consulta: chave primária *idConsulta*, chave estrangeira *Paciente_idPaciente* com os seguintes atributos: data cons., horário, número da consulta;

- Sessões: chave primária *id Sessões*, chaves estrangeira *Paciente_idPaciente*, *Consulta_idConsulta* com os seguintes atributos: Nomes Citados, Nota, Entrega Paciente, Duração, Observações;

- Encaminhamento: chave primária *idEncaminhamento*, chave estrangeira *Paciente_idPaciente* com o seguinte atributo: Diagnostico.

- Liberar Paciente: chave primária *idLiberar Paciente*, chave estrangeira *Paciente_idPaciente* com os seguintes atributos: Data Liberação, Diagnostico.

A modelagem no programa *DBDesigner 4* é mostrada na Figura 8.

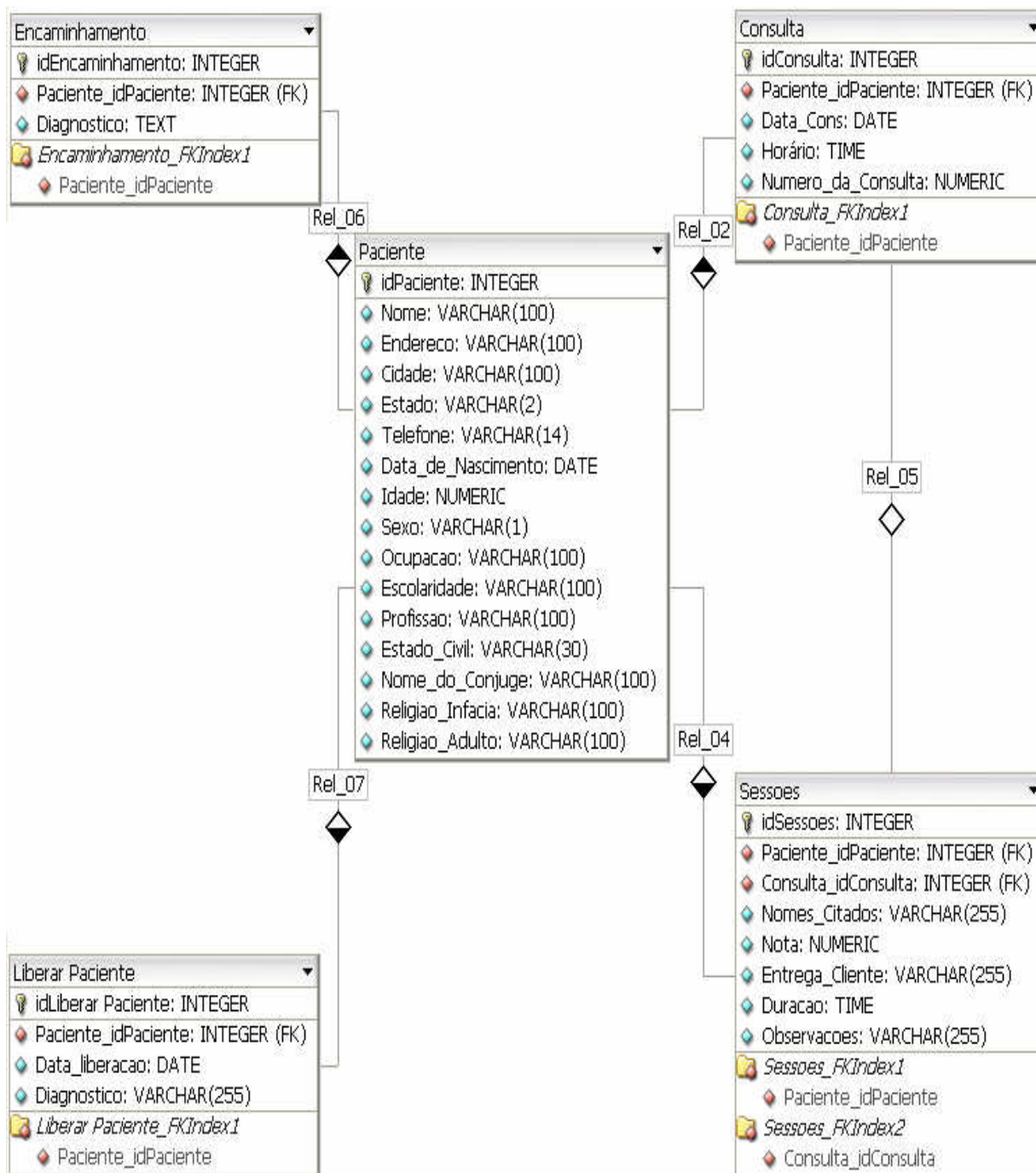


Figura 8 – Modelagem Banco de Dados

4.5 INTERFACE DO SISTEMA

4.5.1 TELA INICIAL DO SISTEMA

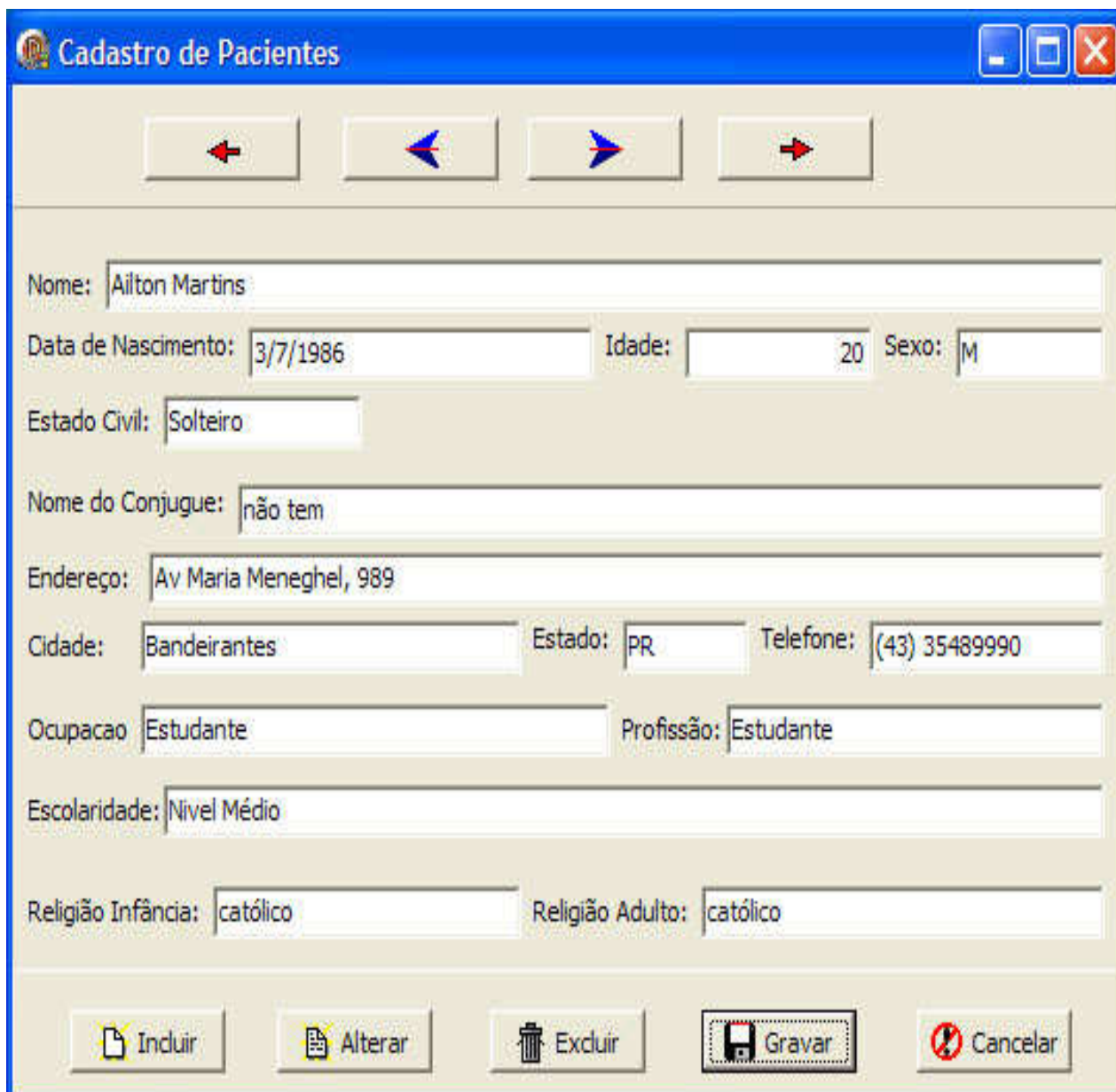
A Psicóloga terá acesso ao sistema abrindo a tela principal conforme mostra a Figura 9:



Figura 9 – Tela Inicial Sistema

4.5.2 TELA CADASTRO DE PACIENTES

Caso a Psicóloga terá acesso à tela Cadastro de Clientes, conforme Figura 10.



Cadastro de Pacientes

Nome: Ailton Martins

Data de Nascimento: 3/7/1986 Idade: 20 Sexo: M

Estado Civil: Solteiro

Nome do Conjuge: não tem

Endereço: Av Maria Meneghel, 989

Cidade: Bandeirantes Estado: PR Telefone: (43) 35489990

Ocupação: Estudante Profissão: Estudante

Escolaridade: Nível Médio

Religião Infância: católico Religião Adulto: católico

Incluir Alterar Excluir Gravar Cancelar

Figura 10 – Tela de Cadastro

O botão Incluir é usado para chamar o método incluir do objeto paciente1

Paciente1.incluir;

O botão Alterar é usado para chamar o método alterar do objeto paciente1
Paciente1.alterar;

O botão Excluir chama o método excluir do objeto paciente1
Paciente1.excluir;

O botão Gravar chama o método gravar do objeto paciente1
Paciente1.gravar;

O botão Cancelar chama o método cancelar do objeto paciente1
Paciente1.cancelar.

Código gerado da Classe Paciente, seus métodos e atributos:

```
Paciente = class
public
procedure gravar;
procedure incluir;
procedure alterar;
procedure cancelar;
procedure excluir;
procedure primeiro;
procedure anterior;
procedure proximo;
procedure Último;

var
nome:string;
endereco:string;
```

```
cidade:string;
estado:string;
telefone:string;
data_de_nascimento:string;
idade:real;
sexo:string;
ocupacao:string;
escolaridade:string;
profissao:string;
estado_civil:string;
nome_do_conjuge:string;
religiao_infancia:string;
religiao_adulto:string;
///
```

Qualquer alteração feita no código altera também a modelagem e se fizer alguma alteração na modelagem conseqüentemente o código também é alterado, sem precisar sair do sistema conforme a Figura 11:

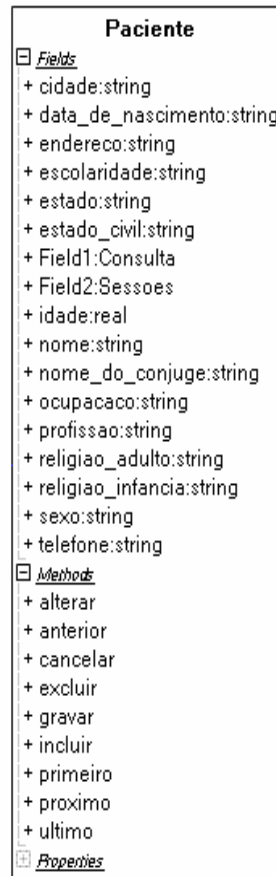
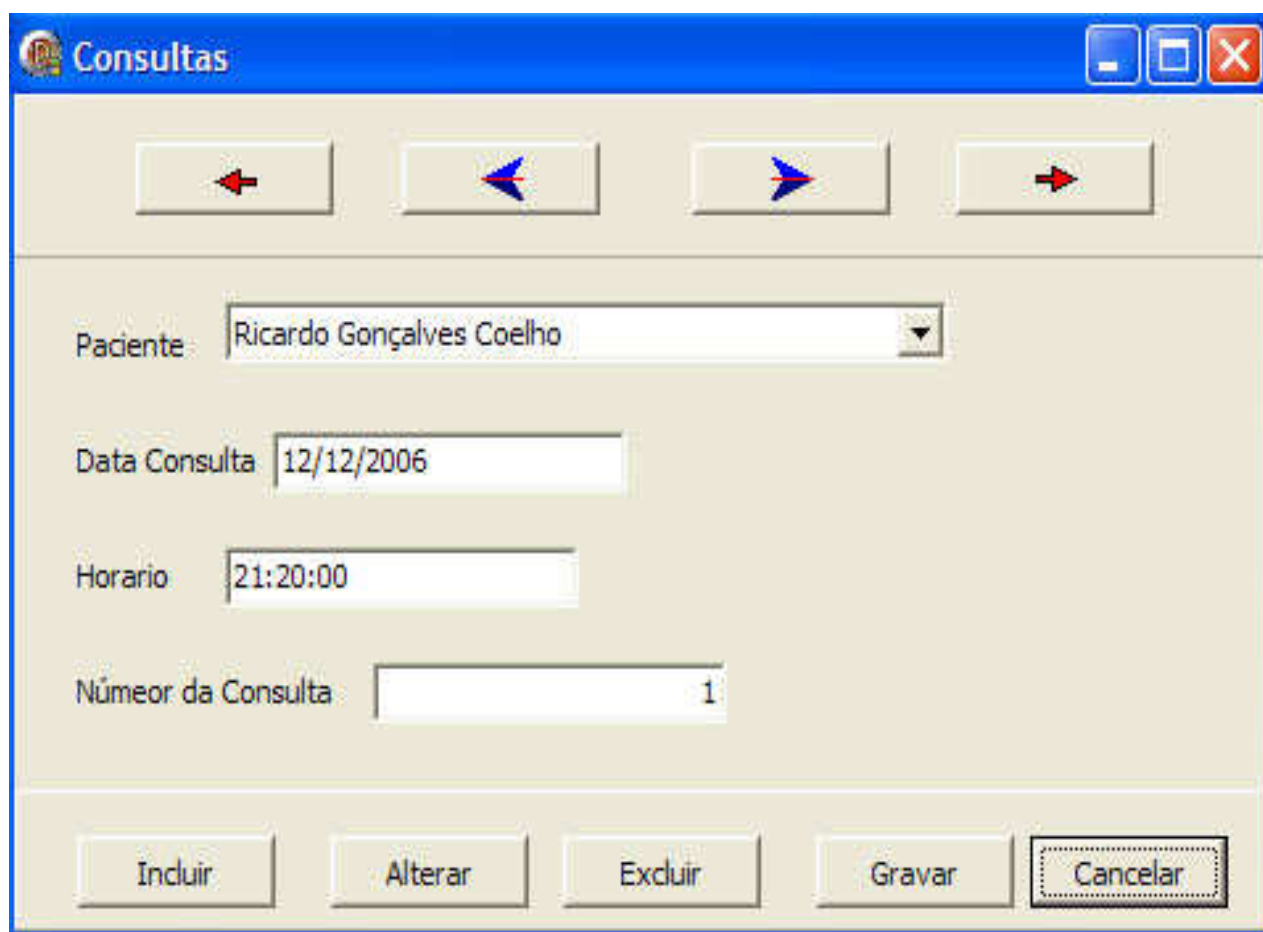


Figura 11 – Classe Paciente

4.5.3 TELA CONSUTLAS

A Psicóloga poderá agendar as consultas de seus pacientes, conforme mostra Figura 11.



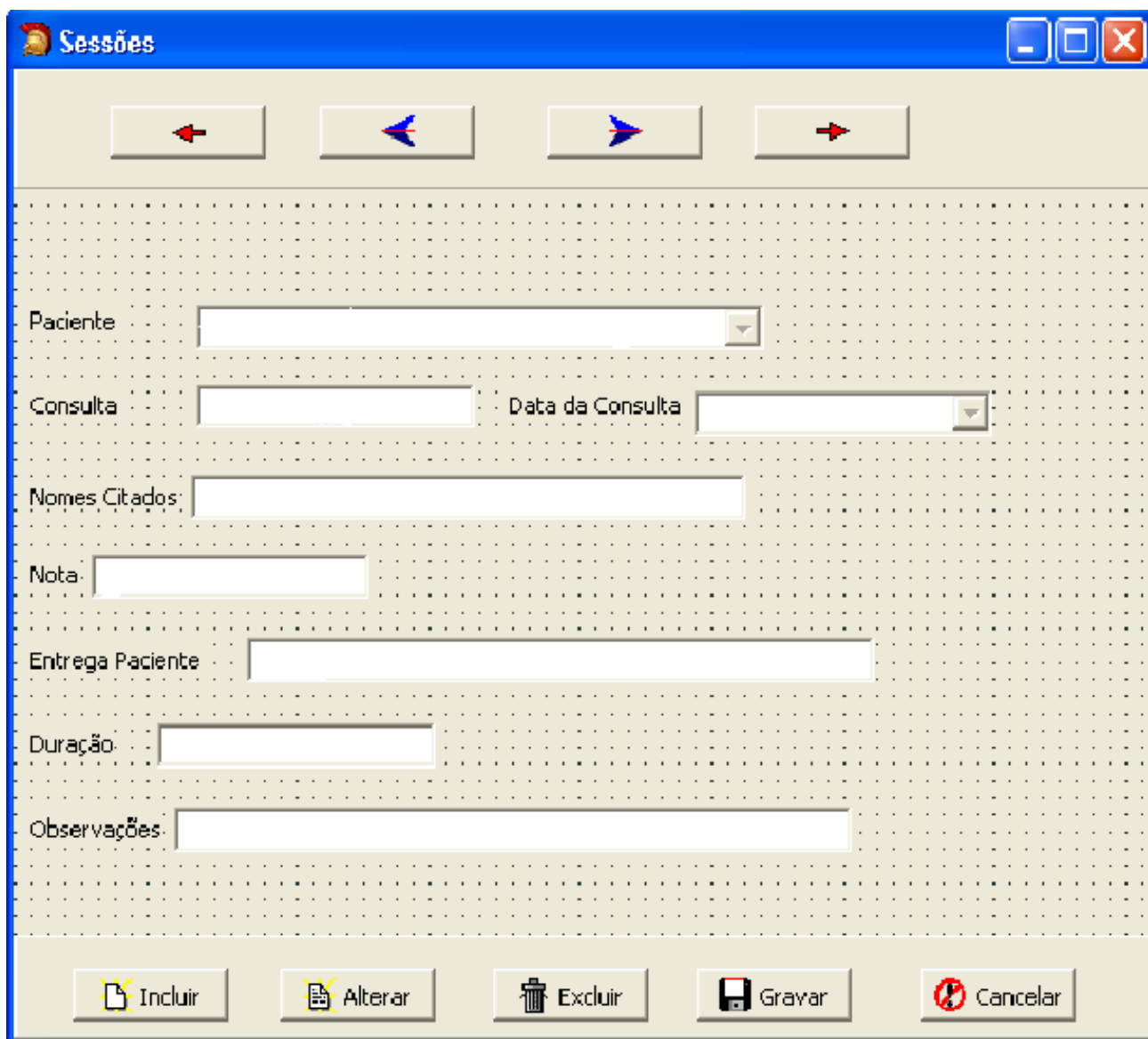
The screenshot shows a software window titled "Consultas" with a blue title bar. The window contains a form with the following fields and controls:

- Four navigation buttons at the top: a red left arrow, a blue double left arrow, a blue double right arrow, and a red right arrow.
- A "Paciente" dropdown menu with the text "Ricardo Gonçalves Coelho".
- A "Data Consulta" text box containing "12/12/2006".
- A "Horario" text box containing "21:20:00".
- A "Número da Consulta" text box containing "1".
- Five action buttons at the bottom: "Incluir", "Alterar", "Excluir", "Gravar", and "Cancelar". The "Cancelar" button is highlighted with a dashed border.

Figura 11 – Tela Consultas

4.5.4 TELA SESSÕES

A Psicóloga terá acesso as sessões, fazendo todas suas anotações, conforme mostra Figura 12.



The screenshot shows a software window titled "Sessões" with a blue title bar. Below the title bar is a navigation bar containing four buttons with arrows: a red arrow pointing left, a blue arrow pointing left, a blue arrow pointing right, and a red arrow pointing right. The main area of the window is a light gray grid with several input fields:

- Paciente:** A dropdown menu.
- Consulta:** A text input field.
- Data da Consulta:** A dropdown menu.
- Nomes Citados:** A text input field.
- Nota:** A text input field.
- Entrega Paciente:** A text input field.
- Duração:** A text input field.
- Observações:** A text input field.

At the bottom of the window is a toolbar with five buttons:

- Incluir:** Button with a plus sign icon.
- Alterar:** Button with a pencil icon.
- Excluir:** Button with a trash can icon.
- Gravar:** Button with a floppy disk icon.
- Cancelar:** Button with a red X icon.

Figura 12 – Tela Sessões

5 CONCLUSÃO E TRABALHOS FUTUROS

A utilização da Modelagem *Together* com *UML* desenvolvida neste estudo serviu para uma ampliação do uso da Informática em campos bem específicos. Como ferramenta, gerou abertura para novas aplicações.

Se, de alguma forma foi útil para aplicações da Informática na prestação de serviços, saúde, indústrias e outras áreas, por outro lado o estudo foi útil para abertura de novos domínios de conhecimento de quem se propôs levar a efeito este estudo.

O uso da modelagem *Together* proporcionou uma grande capacidade de gerenciamento de mudança integrado, modelagem e implementação rápida de aplicações.

Como o modelo de classes sofreu muitas alterações na fase de implementação o uso da ferramenta tornou o processo do gerenciamento da documentação extremamente simples, porém um ponto a destacar é que o modelo fica dependente a tecnologia.

Foi obtido ótimos resultados, através da análise de uma pessoa qualificada sobre o assunto e também pela análise da psicóloga, um protótipo simples mas com fácil entendimento. Foi observado que se pode modelar e ao mesmo tempo fazer a implementação. Se os códigos forem alterados a modelagem automaticamente também é alterada, não é preciso se preocupar em mexer na modelagem fora da implementação, sendo assim também alterando a modelagem os códigos também são alterados.

Para trabalhos futuros, recomenda-se a implementação do sistema com um banco de dados orientado a objeto.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- BORLAND. **Borland Delphi 2006**. Consultado na INTERNET em 08 de junho de 2006. <<http://www.borland.com/br/products/delphi/index.html>>.
- BORLAND. **Together Developer**. Consultado na INTERNET em 08 de junho de 2006. <<http://www.borland.com/br/products/together/index.html#developer>>.
- BORLAND - **Ferramenta de Modelagem de Códigos e Arquitetura de Software da Borland**. Consultado na INTERNET em 09 de novembro de 2006. <http://www.borland.com/br/company/news/press_releases/2006/07_06_06_borland_together_positioned_in_leaders_quadrant_object_oriented_analysis_and_design_tools_magic_quadrant.html>.
- CANTÚ MARCO. **Dominando o Delphi 7: a Bíblia**. 7. Ed. São Paulo: Pearson Education do Brasil, 2003.
- GUEDES, GILLEANES T. A., **Guia De Consulta Rápida UML 2**. 2.^a Ed. São Paulo: Novatec Editora Ltda. Novembro/2.005.
- LINHA DE CÓDIGO. **UML 2**. Consultado na INTERNET em 09 de junho de 2006. <http://www.linhadecodigo.com.br/livros.asp?id=524>.
- MONTE CLARO, L. A. **O mito do aperfeiçoamento profissional na Era da Informação**. Revista Espaço Acadêmico nº 45. São Paulo, 2005.
- MONTEIRO SÉRGIO. **Informatização e Produtividade**. APP – Advanced Permormace Project. Consultado na INTERNET em 07 de maio de 2006. <<http://www.geocites.com/eureka/2471/infotprodut.html>>.

- PFLEEGER, SHARI LAWRENCE. **Engenharia de Software Teoria e Prática**. 2. Ed. São Paulo: Editora Prentice Hall, 2004.

- PRESSMAN, ROGER S. **Engenharia de Software**. 5. Ed. Rio de Janeiro: MaxGraw_Hill, 2002.
- SOMMERVILLE, IAN. **Engenharia de Software**. 6. Ed. São Paulo: Addison Wesley, 2003.

- STEFANI, T. DE. **Trabalho X Informática**. Revista Diversa, Ed. 28. [on line], 2003. Consultado na INTERNET em 02 de abril de 2006. <<http://www.revistadiversa.com.br/InfoDiv%20003.html>>.