

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PATO BRANCO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

FELIPE ELFRANCE BEZ GORIO

SISTEMA DE GERENCIAMENTO PARA CASA DE APOIO

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2013**

FELIPE ELFRANCE BEZ GORIO

SISTEMA DE GERENCIAMENTO PARA CASA DE APOIO

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora:

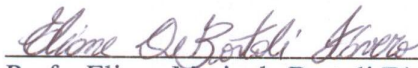
Prof. Eliane Maria De Bortoli Fávero

**PATO BRANCO
2013**

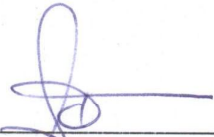
ATA Nº: 217

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO FELIPE ELFRANCE BEZ GORIO.

Às 14:04 hrs do dia 19 de setembro de 2013, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Eliane Maria de Bortoli Fávero (Orientadora), Soelaine Rodrigues Ascari (Convidada) e Rúbia E. O. Schultz Ascari (Convidada), para avaliar o Trabalho de Diplomação do aluno Felipe Elfrance Bez Gorio, matrícula 1066854, sob o título **Sistema para Gestão de Casas de Apoio**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 14:38 hrs foi encerrada a sessão.



Profa. Eliane Maria de Bortoli Fávero, M.Sc.
Orientadora



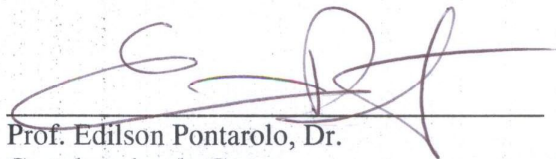
Profa. Soelaine Rodrigues Ascari, M.Sc.
Convidada



Profa. Rúbia E. O. Schultz Ascari, M.Sc.
Convidada



Prof. Eliane Maria De Bortoli Fávero, M. Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus que iluminou o meu caminho durante esta caminhada. À minha mãe que não mediu esforços para que eu chegasse até esta etapa de minha vida.

Ao professor Omero Francisco Bertol e a professora Eliane Maria De Bortoli Fávero, pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia. A todos os professores do curso, que foram tão importantes na minha vida acadêmica.

Aos amigos e colegas, pelo incentivo e pelo apoio constantes.

Pouco conhecimento faz com que as criaturas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o céu, enquanto que as cheias as baixam para a terra, sua mãe.

Leonardo da Vinci

RESUMO

BEZ GORIO, Felipe Elfrance. Sistema de Gerenciamento para Casa de Apoio. 2013. 53f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Câmpus Pato Branco. Pato Branco, 2013.

Considerando a necessidade que casas de apoio – destinadas à hospedagem de pessoas doentes durante o período de tratamento – possuem com relação ao controle dos serviços prestados e a carência desse tipo de sistema de software justificada pela escassez de recursos financeiros para destinar à essa finalidade, o objetivo desse trabalho é realizar o desenvolvimento de um protótipo de sistema que auxilie os administradores desse tipo de entidade a terem um melhor controle para a gestão. Para o desenvolvimento do protótipo foram utilizadas as ferramentas Embarcadero Delphi XE para a programação do sistema *desktop*, e Firebird, para o armazenamento das informações geradas e capturadas pelo sistema, ambas ferramentas bastante difundidas comercialmente. Os objetivos do trabalho foram atingidos e o sistema foi implantado em uma casa de apoio, permitindo a realização de testes de usabilidade e obtenção do *feedback* dos usuários.

Palavras-chave: Gestão de Casas de Apoio, Software de controle, Delphi XE, Aplicação Desktop.

ABSTRACT

BEZ GORIO, Felipe Elfrance. Sistema de Gerenciamento para Casa de Apoio. 2013. 53 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2013.

Considering the need to support homes - designed to hosting people sick during the treatment period - have over the control of the services provided and the lack of this type of system software justified by the scarcity of financial resources intended for this purpose, the goal this work is to develop a prototype system that assists administrators of such entity to have a better control for management. For the development of prototype tools were used Embarcadero Delphi XE for programming system desktop, and Firebird, for the storage of information generated and captured by the system, both tools quite widespread commercially. The objectives were achieved and the system was deployed in a support, allowing conducting usability testing and getting feedback from users.

Keywords: Management Support Homes, control software, Delphi XE, Desktop Application.

LISTA DE FIGURAS

| | |
|----------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 - Logotipo do Firebird, em 05/03/2013. | 19 |
| Figura 2 - Ferramenta Administrativa IBExpert em Execução..... | 21 |
| Figura 3 - Embarcadero Delphi XE em Execução..... | 22 |
| Figura 4 - Formulário Padrão Menu Principal..... | 27 |
| Figura 5 - Formulário Padrão Base Geral..... | 27 |
| Figura 6- Formulário Padrão Base Cadastro..... | 28 |
| Figura 7- Formulário Padrão Pesquisa..... | 28 |
| Figura 8- Formulário Padrão Base Relatórios. | 29 |
| Figura 9- Formulário Padrão Mensagens. | 29 |
| Figura 10 – Diagrama de atividades – Processos de <i>check-in</i> e <i>check-out</i> | 32 |
| Figura 11- Diagrama de entidade e relacionamento do sistema proposto criado com o IBExpert..... | 38 |
| Figura 12- Diagrama de entidade e relacionamento do sistema proposto criado com o software Dezign for Database. | 39 |
| Figura 13- Tela Principal do Sistema Proposto. | 40 |
| Figura 14 - Cadastro de Pessoas preenchido. | 41 |
| Figura 15 - Relatório de <i>Check-In</i> | 42 |
| Quadro 2- Exemplos da programação Delphi utilizada no desenvolvimento do sistema proposto. | 43 |
| Figura 16 – Formulário, untBaseCadastro, utilizado como base de cadastros..... | 45 |

LISTA DE QUADROS

| | |
|--------------------------------------------------------------------------------------------------|----|
| Quadro 1 - Processos <i>versus</i> iterações desenvolvidas..... | 25 |
| Quadro 2 – Requisitos funcionais – Cadastrar Municípios..... | 33 |
| Quadro 3 – Requisitos funcionais – Cadastrar Pessoas..... | 33 |
| Quadro 4 – Requisitos funcionais – Cadastrar <i>Check-In</i> | 33 |
| Quadro 5 – Requisitos funcionais – Efetuar Check-Out..... | 34 |
| Quadro 6 – Requisitos funcionais – Efetuar Aprovação de Estadia..... | 34 |
| Quadro 7 – Requisitos funcionais – Efetuar Alteração..... | 34 |
| Quadro 8 – Requisitos funcionais – Efetuar Exclusão..... | 34 |
| Quadro 9 – Requisitos funcionais – Efetuar Consulta..... | 35 |
| Quadro 10 – Requisitos funcionais – Gerar Relatórios..... | 35 |
| Quadro 11 - Exemplos de programação Delphi utilizada no desenvolvimento do sistema proposto..... | 44 |
| Quadro 12- Exemplos de instruções SQL implementadas..... | 46 |

LISTA DE TABELAS

| | |
|----------------------------------------------------------|----|
| Tabela 1- Tabela de Municípios..... | 35 |
| Tabela 2- Tabela de Pessoas..... | 36 |
| Tabela 3- Tabela de Estadias..... | 36 |
| Tabela 4- Tabela de Acompanhantes. | 37 |
| Tabela 5- Tabela de Procedimentos Médicos. | 37 |
| Tabela 6- Relação das Funcionalidades Implementadas..... | 40 |

LISTA DE SIGLAS

| | |
|------|----------------------------------------------|
| ANSI | <i>American National Standards Institute</i> |
| CASE | <i>Computer-Aided Software Engineering</i> |
| DDL | <i>Data Definition Language</i> |
| DER | Diagrama de Entidade e Relacionamento |
| DML | <i>Data Manipulation Language</i> |
| FK | <i>Foreign Key</i> |
| IDE | <i>Integrated Development Environment</i> |
| ISO | <i>International Standards Organization</i> |
| OS | <i>Operating System</i> |
| PK | <i>Primary Key</i> |
| RAD | <i>Rapid Application Development</i> |
| SGBD | Sistemas de Gerenciamento de Banco de Dados |
| SQL | <i>Structured Query Language</i> |
| SAA | <i>Systems Application Architecture</i> |

SUMÁRIO

| | |
|-------------------------------------------------------------|--------------------------------------|
| 1 INTRODUÇÃO | 12 |
| 1.1 CONSIDERAÇÕES INICIAIS | 12 |
| 1.2 OBJETIVOS | 13 |
| 1.2.1 Objetivo Geral | 13 |
| 1.2.2 Objetivos Específicos | 13 |
| 1.3 JUSTIFICATIVA | 13 |
| 1.4 ESTRUTURA DO TRABALHO | 14 |
| 2 REFERENCIAL TEÓRICO | 14 |
| 2.1 BANCO DE DADOS | 15 |
| 2.2 SISTEMAS DE INFORMAÇÃO | 16 |
| 2.3 DESENVOLVIMENTO DE SOFTWARE PARA AMBIENTE DESKTOP | 17 |
| 2.4 CASA DE APOIO | 18 |
| 3 MATERIAIS E MÉTODO | 19 |
| 3.1 MATERIAIS | 19 |
| 3.1.1 Banco de Dados Firebird..... | 19 |
| 3.1.2 Ferramenta Administrativa IBExpert..... | 20 |
| 3.1.3 Ferramenta de Desenvolvimento Delphi XE..... | 22 |
| 3.2 MÉTODO..... | 24 |
| 3.2.1 Formulários Padrão..... | Erro! Indicador não definido. |
| 3.2.2 Linguagem SQL..... | Erro! Indicador não definido. |
| 4 RESULTADOS E DISCUSSÕES | 30 |
| 4.1 APRESENTAÇÃO DO SISTEMA..... | 30 |
| 4.2 MODELAGEM DO SISTEMA | 30 |
| 4.2.1 Modelagem de negócio para o sistema..... | 35 |
| 4.2.2 Identificação e classificação de requisitos..... | 38 |
| 4.2.3 Descrição das tabelas de dados | 384 |
| 4.2.4 Diagrama de entidade-relacionamento (DER)..... | 387 |
| 4.3 DESCRIÇÃO DAS FUNCIONALIDADES IMPLEMENTADAS | 39 |
| 4.4 ASPECTOS DESTACÁVEIS DA PROGRAMAÇÃO UTILIZADA | 42 |
| 5 CONCLUSÃO | 47 |
| REFERÊNCIAS | 48 |
| APÊNDICES | Erro! Indicador não definido. |
| ANEXOS | Erro! Indicador não definido. |

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos gerais e específicos, a justificativa do trabalho desenvolvido e a organização do texto.

1.1 CONSIDERAÇÕES INICIAIS

As casas de apoio são residências que acolhem pessoas com algum tipo de doença. Funcionam como uma estrutura de suporte de acolhimento temporário durante o período necessário de tratamento dessas pessoas que se deslocam de vários lugares. Geralmente essas casas de apoio não cobram pelos serviços prestados, onde toda a equipe de colaboradores é formada por voluntários.

Possivelmente por se tratar de uma organização sem fins lucrativos e, por sua vez, sem condições para adquirir um sistema de software para a gestão de seus processos, existe uma carência de aplicativos destinados a esse tipo de estabelecimento, voltados especificamente ao auxílio da gestão dessas casas de apoio, gerenciando entrada e saída de pacientes, entre outros. Geralmente o processo de gestão é realizado manualmente, com o auxílio de planilhas eletrônicas. Dessa forma o controle dos processos realizados nessas entidades acaba sendo trabalhoso, demorado e muitas vezes não eficiente.

O objetivo desse trabalho é desenvolver um protótipo que auxilie a gestão de casas de apoio dessa natureza de uma forma mais rápida e eficiente, facilitando o trabalho dos seus responsáveis. Para isso, serão utilizadas tecnologias de desenvolvimento usadas comercialmente para o desenvolvimento de sistemas de software *desktop*, como por exemplo, a IDE (*Integrated Development Environment*) do Delphi XE.

1.2 OBJETIVOS

Este trabalho de conclusão de curso trata de um desenvolvimento de software e a seguir serão descritos seus objetivos gerais e específicos.

1.2.1 Objetivo Geral

- Desenvolver um protótipo de um sistema de software para a gestão de casas de apoio, facilitando o controle dos serviços por ela prestados.

1.2.2 Objetivos Específicos

- Controlar as entradas e saídas de pacientes hóspedes de uma casa de apoio;
- Facilitar o controle dos serviços prestados pela casa de apoio a partir da implementação das seguintes funcionalidades:
 - Cadastro, manutenção e consulta de pessoas, municípios e estadia;
 - Emissão de relatórios de controle: comprovante de entrega de *tickets alimentação*, pessoas, municípios, *check-in* e *check-out*.

1.3 JUSTIFICATIVA

Considerando a atual importância da informatização nos diversos ramos de atividade existentes, percebeu-se uma grande carência de sistemas de software aplicáveis à gestão de casas de apoio destinadas ao atendimento de pacientes em tratamento, conforme citado no item 1.1 deste documento, tendo em vista que não se trata de uma área que visa lucro. Sendo assim, esse trabalho tem como objetivo

contribuir com esse tipo de estabelecimento, oferecendo uma ferramenta fácil e eficiente para o seu uso.

Com relação ao acadêmico, esse trabalho justifica-se pela possibilidade do mesmo realizar todo o processo de desenvolvimento de um software, desde o seu levantamento de requisitos até a sua implantação, a partir da aplicação dos conhecimentos obtidos durante a graduação.

1.4 ESTRUTURA DO TRABALHO

Apresenta-se a seguir como será a estrutura do presente trabalho:

- Capítulo 1 - Apresenta uma breve introdução ao trabalho, contendo as considerações iniciais, os objetivos e as justificativas do estudo;
- Capítulo 2 - Apresenta o referencial teórico do trabalho, com os conceitos teóricos utilizados para o entendimento do estudo desenvolvido;
- Capítulo 3 - Descreve os materiais e métodos utilizados no desenvolvimento do sistema;
- Capítulo 4 – Encontra-se os resultados e discussões do trabalho, contendo a descrição, a modelagem e a implementação do sistema, além dos dados obtidos e consequentes discussões.

2 REFERENCIAL TEÓRICO

Este capítulo contém o referencial teórico do trabalho definindo banco de dados, sistemas de informação, desenvolvimento de software para ambiente *desktop* e apresentando os controles realizados em uma casa de apoio.

2.1 BANCO DE DADOS

Segundo Silberschatz; Korth e Sudarshan (2010), um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico. Sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto tem-se a possibilidade de construção de um banco de dados.

Para Date (2003, pág. 10), um banco de dados é uma coleção de dados persistentes, usados pelos sistemas de aplicação de uma determinada empresa.

Já para O'Brien (2002, pág. 143), um banco de dados é um conjunto integrado de registros ou objetos logicamente afins. Um objeto consiste em valores de dados descrevendo os atributos de uma entidade, além de suas operações que podem ser executadas sobre os dados.

Para que os dados das aplicações computacionais possam ser manipulados em um ambiente eficiente existem os Sistemas de Gerenciamento de Banco de Dados, mais conhecidos como SGBD. Como exemplos de SGBD têm-se: Oracle, SQL Server, DB2, PostgreSQL, MySQL, Firebird, Microsoft Access ou Paradox, entre outros. Um SGBD deve garantir a segurança das informações e impedir que usuários sem permissão e em algumas vezes mal intencionados, acessem e manipulem os dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2010).

2.2 SISTEMAS DE INFORMAÇÃO

Segundo O'Brien (2002, pág. 6), sistema de informação é um conjunto organizado de pessoas, hardware, software, redes de comunicação e recursos de dados que coleta, transforma e dissemina informações em uma organização.

Os sistemas de informação transformam a informação em uma forma utilizável para a coordenação de trabalhos de uma empresa ajudando empregados ou gerentes a tomar decisões, analisar e visualizar assuntos complexos e resolver variados tipos de problemas (LAUDON E LAUDON, 1999, pág. 4).

Um sistema de informação possui, segundo O'Brien (2002, pág. 17), três componentes em interação:

- Entrada: abrange a coleta de elementos, na forma de dados brutos, que serão inseridos no sistema para serem processados.
- Processamento: é o processo de transformação da matéria prima em produto (dados transformados em informações úteis).
- Saída: é o transporte das informações produzidas pelo processamento até o usuário final.

Segundo Rezende (2005, pág. 27), os sistemas de informação têm como objetivo auxiliar os processos de tomada de decisões nas organizações. Ou seja, com os dados processados e disponibilizados na forma de informação concreta e com a possibilidade de tomar decisões mais consistentes diminui as chances de escolhas erradas.

Para que um sistema de informação contemple suas etapas, são necessários alguns recursos, apresentados por O'Brien (2002, pág. 22) como:

- Recursos humanos: são os usuários do sistema, especialistas em Tecnologia da Informação e coordenadores das tecnologias. Todos os envolvidos irão e deverão cooperar para o desempenho do sistema. São importantes, pois operam diretamente e utilizam as informações dos sistemas de informação.
- Recursos de hardware: são todos os equipamentos físicos utilizados para fazer a entrada, processamento e saída dos dados de um sistema de informação. São considerados equipamentos físicos, além dos

computadores, periféricos, mouse, teclado, todas as formas de armazenamento, como CD, DVD, discos rígidos ou pen-drives.

- Recursos de software: são instruções computacionais pré-programadas que controlam e coordenam os hardwares e fazem com que executem todas as instruções feitas pelo sistema de informação. Softwares, além de comandarem o trabalho de hardwares, apresentam uma interface gráfica para que o usuário possa de forma confortável e fácil se utilizar do sistema.
- Recursos de dados: dados são mais que a matéria-prima dos sistemas de informação. Sendo assim, uma das maiores preocupações é a disponibilidade, utilidade, segurança, armazenamento e organização desses dados. Para isso incluem-se os meios físicos como discos rígidos, magnéticos, ópticos e ainda conta com softwares que regem a organização desses dados em meios físicos.
- Recursos de rede: utilizados para transferir dados entre dois pontos distintos e interligar diferentes hardwares através de uma rede. Para que as informações geradas pelos sistemas de informações sejam acessadas em diversos locais e para que o ligamento de diversos módulos do sistema seja possível existem os recursos de redes.

Segundo O'Brien (2002), a união e a correta utilização dos recursos dos sistemas de informação faz com que as empresas se tornem mais organizadas e competitivas no mercado.

Para Laudon e Laudon (1999, pág. 27), sistemas de informações mudam objetivos, produtos e relações internas e externas dentro de uma empresa.

2.3 DESENVOLVIMENTO DE SOFTWARE PARA AMBIENTE *DESKTOP*

Um software desenvolvido para ambiente *desktop* é capaz de ser executado diretamente na máquina em que foi instalado. Geralmente utilizado em organizações onde existe a necessidade de rapidez, segurança e capacidade para utilizar com facilidade todos os recursos da máquina.

O mercado apresenta diversas linguagens de programação para desenvolvimento de software para ambiente *desktop*, entre elas pode-se destacar, Delphi, VB.NET e Java. Muitas dessas linguagens possuem componentes importantes para acelerar o processo de desenvolvimento de software. Um mapa com a história das linguagens de programação até os dias atuais pode ser obtido em Levenez (2013).

2.4 CASA DE APOIO

Fundado há 25 anos, pela Sra. Ivanir, o Lar Nossa Senhora Aparecida é uma casa de apoio para as pessoas de fora da capital do estado do Paraná, que vão em busca de tratamento para diversas doenças sérias em diversos hospitais de Curitiba. Localizado na R. Dr. Ovande do Amaral, 270 - Jardim das Américas Curitiba - PR, 81520-060, o lar acolhe pacientes e seus acompanhantes. Em média são 40 pessoas hospedadas, a maioria com câncer de algum tipo. O lar dá assistência e abriga homens, mulheres, crianças e idosos.

A casa de apoio em questão sobrevive de doações espontâneas, e toda equipe de trabalho é voluntária, não há servidores contratados. Cada paciente fica, em média, hospedado durante 40 dias, buscando, em geral, quimioterapia ou radioterapia.

O controle de pessoas acolhidas pela casa de apoio anteriormente era realizado em planilhas eletrônicas, a qual demandava tempo para a realização da inclusão de novas pessoas e acompanhantes.

3 MATERIAIS E MÉTODO

Este capítulo foi dividido em duas seções, sendo uma para descrever os materiais (ou tecnologias) e outra para os métodos utilizados para alcançar os objetivos do trabalho de diplomação.

3.1 MATERIAIS

Nesta seção serão descritos o banco de dados e as ferramentas utilizadas na modelagem do sistema, criação e manutenção do banco de dados e também o ambiente de desenvolvimento, abordando conceitos, funcionalidades e características das tecnologias empregadas.

3.1.1 Banco de Dados Firebird

O Firebird, identificado com o logotipo apresentado na Figura 1, é um banco de dados cliente/servidor *free* (software de utilização livre) e *open source* (código aberto), de alta performance, compatível com o padrão SQL ANSI-92 e que foi desenvolvido a partir do código fonte aberto do InterBase 6. Segundo Cantu (2008), ele roda em mais de 10 (dez) sistemas operacionais e vem conquistando espaço no disputado mundo dos servidores SQL.



Figura 1 - Logotipo do Firebird
Fonte: Firebird (2013)

Ainda segundo Cantu (2008), em Julho de 2000 a Borland abriu o código fonte do InterBase 6, tornando-o *free* e *open source*. Nesse momento, um grupo de pessoas resolveu criar um novo banco de dados baseado neste código. Entre essas pessoas estavam nomes influentes da comunidade InterBase, como Ann Harrison, conhecida como “mãe do InterBase” e indicada como uma das pessoas que mais entendem da sua arquitetura interna. Coincidência ou não, ela é casada com Jim Starkey, criador do sistema de *versioning* utilizado no InterBase e no Firebird.

Destacado por Cantu (2008), o Firebird é um servidor que pode ser instalado, mantido e configurado facilmente e não exige qualquer tipo de conhecimento específico. Para baixar o instalador basta entrar no site do projeto do Firebird¹ e localizar para *download* a versão para o sistema operacional desejado.

3.1.2 Ferramenta Administrativa IBExpert

Uma das principais necessidades dos administradores ou desenvolvedores de aplicações que utilizam o Firebird é uma ferramenta que possibilite administrar seu banco de dados com facilidade e rapidez. Infelizmente não existe uma ferramenta gráfica “padrão” para o Firebird.

Existem inúmeras ferramentas disponíveis e que podem ser baixadas da Internet, muitas delas gratuitas. Para obter uma lista com inúmeras opções basta entrar na área de *downloads*, categoria “Ferramentas administrativas”, do site da FireBase².

Segundo Cantu (2008), o IBExpert (**Figura 2**) é uma das principais ferramentas disponíveis para administração, através de interfaces gráficas, dos bancos de dados Interbase e Firebird. Ela é uma ferramenta CASE (*Computer-Aided Software Engineering*, ou Engenharia de Software Apoiada por Computador) desenvolvida e distribuída pela empresa HK Software³.

¹ <http://www.firebirdsql.com/>, visitado em 05/03/2013.

² <http://www.firebase.com.br/fb/downloads.php?categ=2>, visitado em 05/03/2013.

³ <http://ibexpert.net/ibe/>, visitado em 05/03/2013.

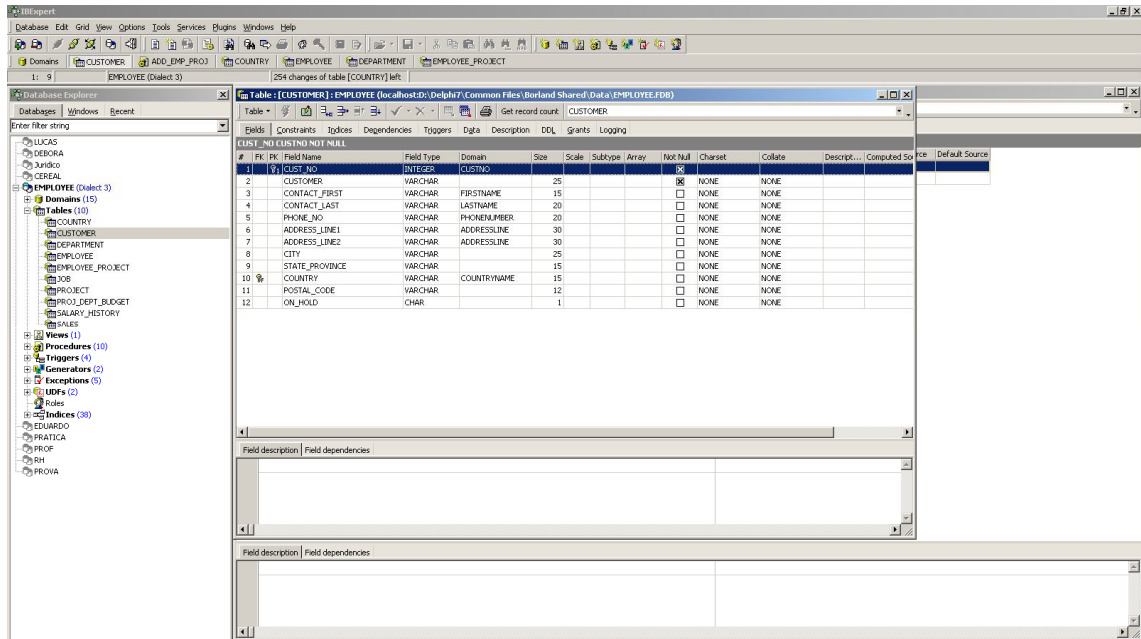


Figura 2 - Ferramenta Administrativa IBEExpert em Execução

Para Cantu (2008), o IBEExpert oferece muitos recursos para administração de bancos de dados. Os mais importantes estão relacionados a seguir:

- Suporte aos bancos de dados Interbase e FireBird;
- Criação e registro de banco de dados Interbase e Firebird;
- Criação de tabelas, chaves e índices;
- Modelagem de dados (*Database Designer*);
- Editores para todos os objetos do banco de dados: tabelas, *triggers* (gatilhos), *stored procedures* (procedimentos armazenados), *domains* (domínios), entre outros;
- Depurador para *triggers* e *stored procedures*;
- Recursos avançados de edição, tais como: *Code insight*, *Code Completion* e *Keyboard Templates*;
- Construtor visual de consultas SQL (*Visual Query Builder*);
- Documentação do banco de dados em formato HTML.

3.1.3 Ferramenta de Desenvolvimento Delphi XE

O Embarcadero Delphi XE⁴, apresentado na **Figura 3**, anteriormente conhecido como CodeGear Delphi, Inprise Delphi, Borland Delphi, ou ainda, de forma simplificada também denominado de Delphi, é um ambiente integrado de desenvolvimento (ou IDE- *Integrated Development Environment*) e uma linguagem de desenvolvimento rápido de aplicações (ou RAD- *Rapid Application Development*).

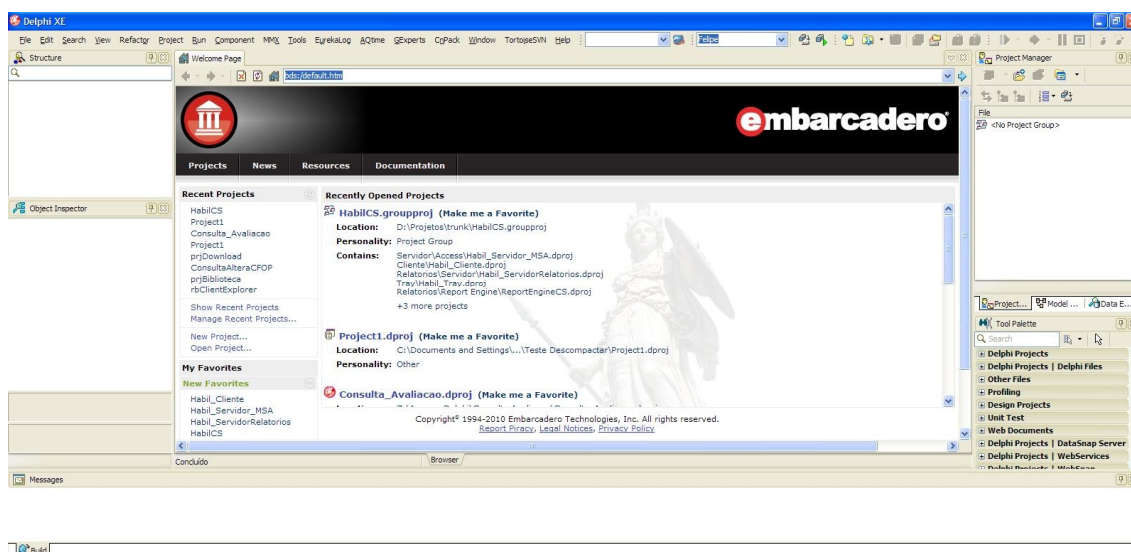


Figura 3 - Embarcadero Delphi XE em Execução.

A linguagem de programação Delphi, produzida e distribuída antigamente pela Borland Software Corporation⁵ e atualmente produzida pela Embarcadero Technologies⁶, foi criada para a plataforma Windows e chegou a ser usada para desenvolvimento de aplicações nativas para Linux e Mac OS, por meio do Kylix⁷, e para o *framework* Microsoft .NET em suas versões mais recentes.

O Delphi é um ambiente de desenvolvimento de aplicações orientado a objetos que permite a criação de aplicações para sistemas operacionais com *templates* e *experts* de aplicações e formulários (*wizards*) que aumentam muito a produtividade e facilitam a programação. Além disso, ele utiliza um dialeto da linguagem *Object Pascal* que tem seus componentes definidos permitindo heranças

⁴ <http://www.embarcadero.com/products/delphi>, visitado em 05/03/2013.

⁵ <http://www.borland.com/>, visitado em 05/03/2013.

⁶ <http://www.embarcadero.com/br>, visitado em 05/03/2013.

⁷ O Kylix é uma IDE para as linguagens C++ e *Object Pascal*.

e criação de novos componentes na própria linguagem. Ele também possibilita o acesso a banco de dados de forma fácil e possui um ambiente de depuração integrado facilitando a solução dos erros de programação que poderão surgir durante o desenvolvimento.

3.1.4 Linguagem SQL

Segundo Date (2003), a SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada) tem representado o padrão para linguagens de banco de dados relacionais. Existem diversas versões de SQL. A versão original foi desenvolvida pela IBM no início dos anos 70.

Em 1986, o *American National Standards Institute* (ANSI) e a *International Standards Organization* (ISO) publicaram os padrões para a SQL, chamada de SQL-86. A IBM publicou seus próprios padrões para a SQL, a *Systems Application Architecture Database Interface* (SAA-SQL) em 1987. Uma extensão para o padrão SQL, a SQL-89, foi publicada em 1989. A versão em uso do padrão ANSI/ISO SQL é o padrão SQL-92 (DATE, 2003, pág. 71 e 72).

Embora seja definida como uma “linguagem de consulta” a linguagem SQL possui muitos outros recursos além de consulta ao banco de dados, como meios para a definição da estrutura de dados (esquema), para modificação de dados no banco de dados e para a especificação de restrições de segurança.

Os recursos da Linguagem SQL podem ser divididos, principalmente, em duas partes:

- a) Linguagem de definição de dados (DDL - *Data Definition Language*): comandos para a definição de esquemas de relações⁸, exclusão de relações, modificação nos esquemas de relações e criação de domínios⁹. Por exemplo: *create table*, *alter table*, *drop table*, *create domain*, *create view*, *drop view*, entre outros. Sendo que: *create* é usado para indicar a criação de um objeto (uma tabela ou um domínio); *drop* é utilizado para apagar um objeto do banco de dados; *alter*

⁸ Esquema de uma relação: especifica o nome da relação (ou tabela), mais o nome e o tipo de cada atributo (campo, ou coluna).

⁹ Conjunto de valores permitidos a um atributo.

permite ao usuário alterar um objeto; e, *view* é uma visão, que no contexto dos bancos de dados é uma relação que não armazenam dados.

b) Linguagem interativa de manipulação de dados (DML - *Data Manipulation Language*): abrange uma linguagem de consulta baseada tanto na álgebra relacional quanto no cálculo relacional de tuplas. Engloba também comandos para inserção, exclusão e modificação de tuplas no banco de dados. Por exemplo:

- *select*: comando usado em SQL para realizar consultas a dados pertencentes a uma tabela.
- *insert*: instrução usada para inserir um registro em uma tabela existente.
- *update*: instrução usada para mudar os valores de dados em uma ou mais linhas de uma tabela existente.
- *delete*: permite ao usuário remover linhas existentes de uma tabela.

3.2 MÉTODO

Este tópico apresenta o modelo de ciclo de vida adotado para o desenvolvimento do projeto a que se propõe este trabalho.

O processo de modelagem aplicado no sistema para auxiliar o gerenciamento de uma casa de apoio teve como base o processo iterativo e incremental. Inicialmente foram definidos os requisitos principais, obtendo-se uma visão geral do sistema. Esses requisitos foram complementados à medida que o sistema era modelado e implementado.

Foi tomado como base o Processo Unificado (JACOBSON; BOOCH; RUMBAUGH, 2006; WAZLAVICK, 2011), que caracteriza-se por ser iterativo e incremental, adequando-se às características deste trabalho, que consiste em desenvolver gradativamente e em paralelo os requisitos do objeto e sua implementação, sendo que a cada discussão com o orientador e interessados, novos requisitos podem ser implementados e modificações podem ser sugeridas, gerando um incremento ou uma nova versão do protótipo. Desta forma, uma iteração pode ser definida como o aperfeiçoamento sucessivo de um determinado processo. Ela

pode ocorrer de dois modos: cada fase pode ser realizada de forma iterativa, com os resultados desenvolvidos incrementalmente e, além disso, o conjunto total das fases pode também ser realizado de forma incremental (SOMMERVILLE, 2007).

Neste trabalho, uma iteração consiste na repetição das fases do processo de desenvolvimento de software padrão (e de realização sequencial) conforme propostas por Pressman (2005). São elas: levantamento de requisitos e análise, projeto, implementação e testes.

O Quadro 1 apresenta os processos (fluxos de trabalho) e as iterações desenvolvidas durante o ciclo de vida do sistema proposto.

| Processo | 1ª iteração | 2ª iteração |
|---------------------------------------------------|--------------------------------------------------------------|-----------------------------------------------------------------|
| Levantamento dos Requisitos | Primeira versão | Refinamento dos requisitos |
| Análise de Requisitos Funcionais e não-funcionais | Primeira versão | Refinamento dos requisitos funcionais e não-funcionais |
| Banco de Dados | Definição do modelo conceitual do banco de dados | Implementação do modelo conceitual do banco de dados |
| Implementação | Formulários padrão | Implementação das funcionalidades do sistema |
| Testes | De modelo de interface e de código pelo autor deste trabalho | De usabilidade pela responsável administrativa da Casa de Apoio |

Quadro 1 - Processos versus iterações desenvolvidas.

A seguir estão descritos os processos realizados no desenvolvimento do sistema e as principais atividades de cada uma dessas etapas:

- a. Levantamento dos requisitos: iniciou com entrevistas mediadas por computador, envolvendo troca de e-mails com a responsável administrativa da casa de apoio. Esses requisitos definiram a ideia e visão geral do sistema em termos dos dados manipulados nas tarefas administrativas realizadas durante o registro e acompanhamento de uma estadia na casa de apoio.
- b. Análise de requisitos funcionais: iniciou com entrevistas mediadas por computador, através de troca de e-mails, com a responsável administrativa da Casa de Apoio. Esses requisitos definiam a ideia e visão geral do sistema em termos das funcionalidades (processos) administrativas realizadas durante o registro e acompanhamento de uma estadia na casa de apoio.
- c. Banco de dados: definição do modelo conceitual do banco de dados a partir dos requisitos de dados levantados, usando o Firebird por meio da ferramenta administrativa IBExpert.

- d. Implementação: realizada utilizando o ambiente de desenvolvimento Delphi XE e banco de dados Firebird para a implementação das tarefas administrativas da casa de apoio.

Para o desenvolvimento do projeto proposto, foram utilizados formulários padrão, que segundo Bertol (2013) são formulários pré-fabricados, que ajudam a desenvolver com maior rapidez uma aplicação. Sua principal vantagem é trazer elementos comuns encontrados na maioria dos formulários de cadastro, por exemplo, tornando estes padronizados e funcionando como modelos para o usuário desenvolvedor.

Foi utilizado um conjunto de 6 (seis) formulários padrão, com as seguintes finalidades:

- 1) Menu principal: apresentado na Figura 4, será utilizado na construção de formulários com o menu de opções de uma aplicação.
- 2) Base geral: apresentado na Figura 5, será utilizado no desenvolvimento dos formulários base de cadastro e relatórios.
- 3) Base cadastro: apresentado na Figura 6, será utilizado no desenvolvimento dos formulários com as opções de cadastros, como por exemplo, cadastro de cidades, cadastro de pessoas e cadastro de estadias.
- 4) Pesquisa: apresentado na Figura 7, será utilizado no desenvolvimento dos formulários que realizarão a pesquisa de pessoas, municípios e estadias;
- 5) Base relatórios: apresentado na Figura 8, será utilizado no desenvolvimento dos formulários para realizar filtros de relatórios;
- 6) Mensagem: apresentado na Figura 9, será utilizado no desenvolvimento dos formulários para exibir mensagens de texto na tela.

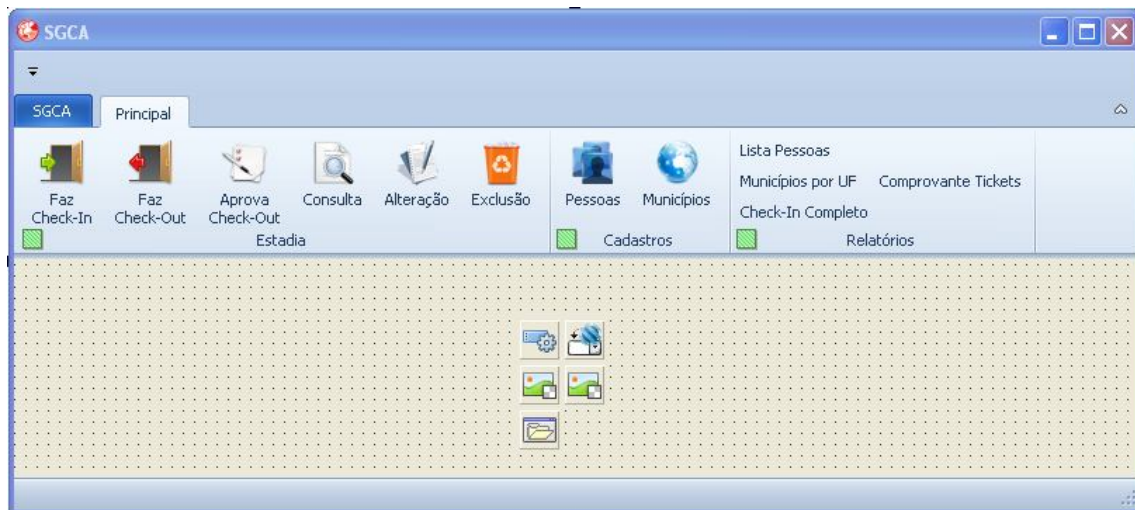


Figura 4 - Formulário Padrão | Menu Principal.

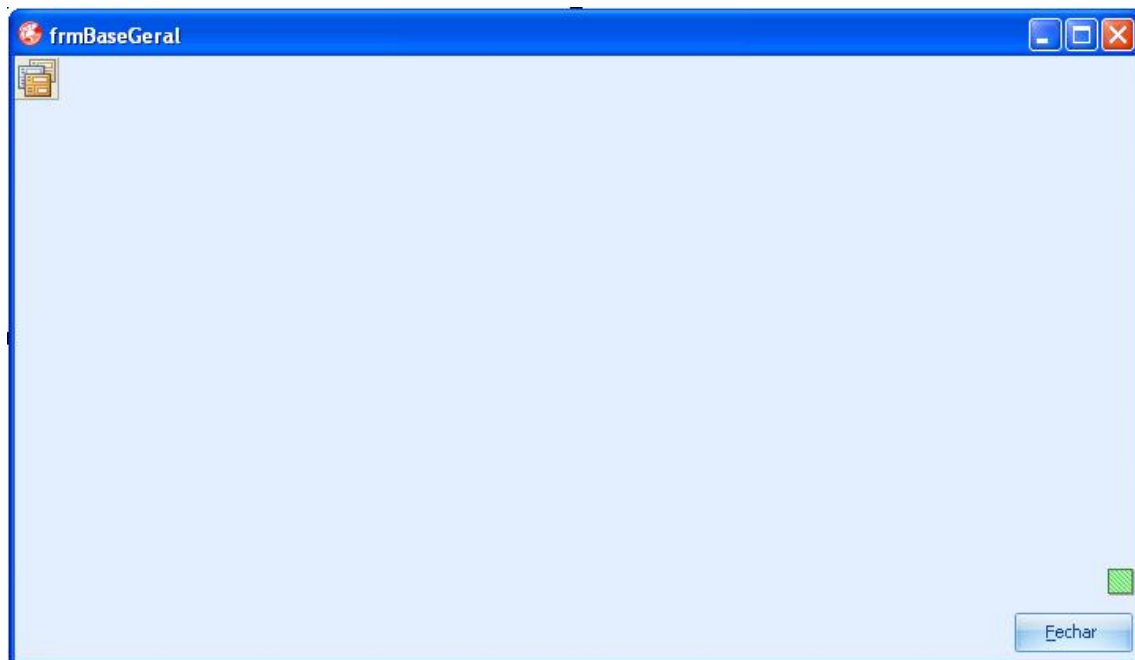


Figura 5 - Formulário Padrão | Base Geral.

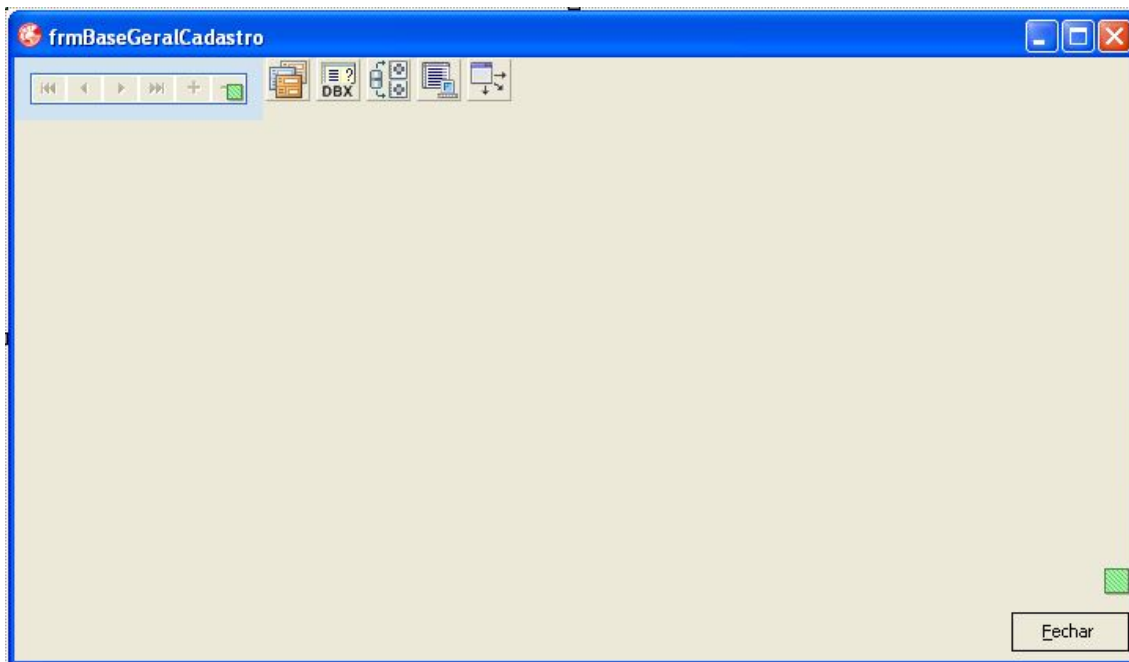


Figura 6- Formulário Padrão | Base Cadastro.

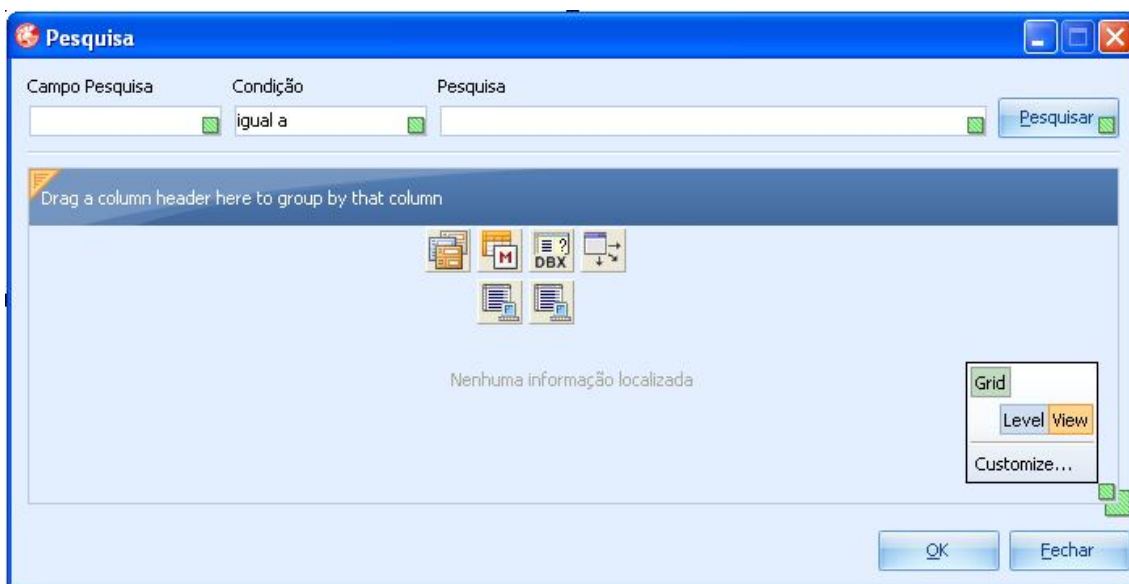
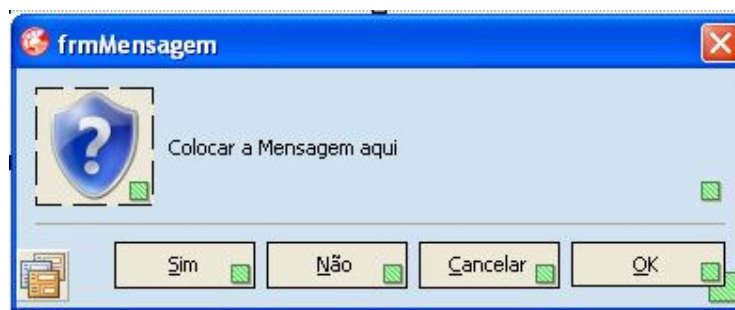


Figura 7- Formulário Padrão | Pesquisa.



The screenshot shows a window titled "Relatório" with a blue header bar. Inside the window, there is a label "Títul" followed by a text input field, and below it, a label "Descriç" followed by another text input field. At the bottom of the window, there are two buttons: "Gerar" and "Fechar".

Figura 8- Formulário Padrão | Base Relatórios.



The screenshot shows a window titled "frmMensagem" with a blue header bar. Inside the window, there is a blue shield icon with a question mark and the text "Colocar a Mensagem aqui". At the bottom of the window, there are four buttons: "Sim", "Não", "Cancelar", and "OK".

Figura 9- Formulário Padrão | Mensagens.

- e. Testes: na fase desenvolvimento foram realizados os testes de interface e codificação pelo autor do trabalho. Na etapa de implantação foram realizados testes de usabilidade pela responsável administrativa da casa de apoio.

4 RESULTADOS E DISCUSSÕES

Este capítulo apresenta o que foi alcançado como resultado deste trabalho de conclusão de curso fazendo a apresentação do sistema proposto para gerenciamento de uma casa de apoio, descrevendo a modelagem de dados e as funcionalidades implementadas.

4.1 APRESENTAÇÃO DO SISTEMA

O projeto desenvolvido neste trabalho de conclusão de curso corresponde a um sistema de informação, definido na seção 2.2, para ser executado em ambiente *desktop* (apresentado na seção 2.3). É um software para realizar o gerenciamento de uma casa de apoio, descrita na seção 2.4, incluindo os recursos de cadastro de município, cadastro de pessoas, *check-in* e *check-out* de pacientes, alteração de estadia, consulta de estadia, aprovação de *check-out*, relatórios de listagem de pessoas, de municípios, comprovante de *tickets* entregues, impressão do *check-in*.

A linguagem de programação escolhida foi Delphi por meio da ferramenta de desenvolvimento Delphi XE, descrita na subseção 3.1.3, com o apoio de formulários padrão, relacionados na subseção 3.2.1, e recursos da linguagem SQL, apresentada na subseção 3.1.4. Também foram utilizados o banco de dados Firebird (subseção 3.1.1) e a ferramenta administrativa de banco de dados IBExpert (subseção 3.1.2).

4.2 MODELAGEM DO SISTEMA

O levantamento de dados foi realizado por meio de entrevistas mediadas por computador, envolvendo também a troca de e-mails com a responsável administrativa da casa de apoio e a colaboradora da casa de apoio Juliana Schuster, a qual possui um contato próximo aos responsáveis pela casa de apoio, e explicou

como é realizado o processo desde que o paciente entra na casa, até sua saída. Como resultado dessas reuniões e análise de documentos gerados pela colaboradora, conforme exemplo disponível no Anexo 1, foram definidos os requisitos funcionais e não-funcionais, bem como o modelo de dados para o sistema proposto.

4.2 Levantamento de Requisitos e Análise do Sistema

A seguir serão apresentados os artefatos relativos ao levantamento de requisitos e análise do sistema proposto.

4.2.1 Modelagem de negócio para o sistema

O diagrama de atividades da Figura 10 apresenta os processos de negócio relativos ao *check-in* e ao *check-out* dos hóspedes, da forma como devem ocorrer na casa de apoio, elaborado conforme conversa com a colaboradora do estabelecimento. Esse diagrama auxiliou na descoberta dos requisitos funcionais e não funcionais que o sistema deveria apresentar e dos dados a serem mantidos pelo sistema.

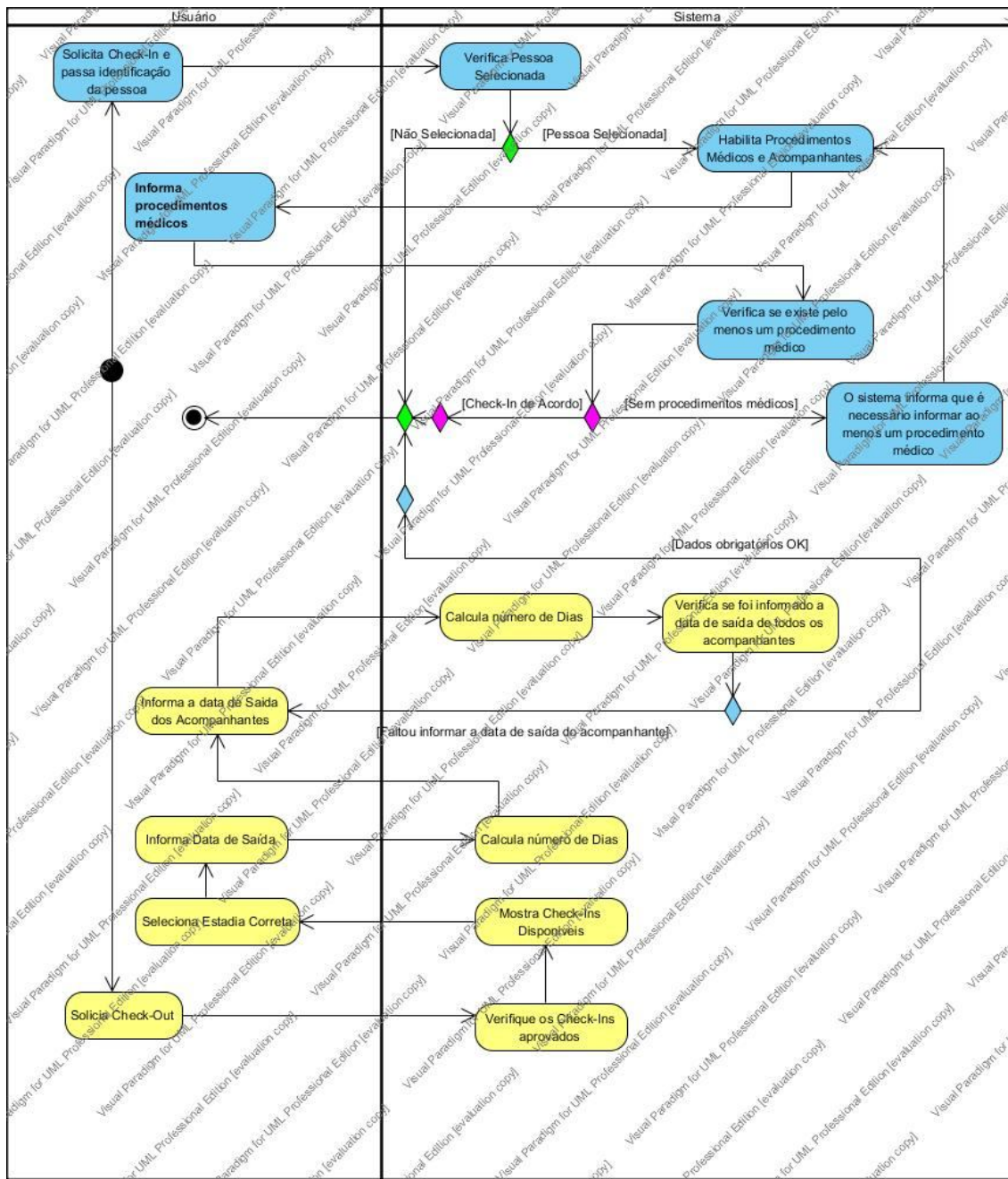


Figura 10 – Diagrama de atividades – Processos de check-in e check-out

4.2.2 Identificação e classificação de requisitos

A seguir serão apresentados os Quadros 2 a 10, o quais apresentam a especificação dos requisitos funcionais e não-funcionais levantados para o sistema desenvolvido.

| F1 Cadastrar Municípios | | Oculto () | | |
|-------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deve permitir registrar o município, contendo em seu cadastro o código, o município e a UF. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF1.1 Identificação do Município | O município deve ser identificado por um código único gerado automaticamente. | Regra de negócio | () | (X) |
| NF1.2 Janela Única | Todas as funções relacionadas ao cadastro de municípios devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 2 – Requisitos funcionais – Cadastrar Municípios

| F2 Cadastrar Pessoas | | Oculto () | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deve permitir registrar pessoas, contendo em seu cadastro pelo menos um código, nome, RG, data de nascimento, telefone residencial, telefone celular, endereço, número, complemento, bairro, cep e código do município. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF2.1 Identificação da Pessoa | A pessoa deve ser identificada por um código único gerado automaticamente. | Regra de negócio | () | (X) |
| NF2.2 Janela Única | Todas as funções relacionadas ao cadastro de pessoas devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 3 – Requisitos funcionais – Cadastrar Pessoas

| F3 Cadastrar <i>Check-In</i> | | Oculto () | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deve permitir registrar o check-in, contendo em seu cadastro pelo menos uma pessoa informada e um procedimento médico cadastrado. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF3.1 Identificação do Check-In | O Check-In deve ser identificado por um código único gerado automaticamente. | Regra de negócio | (X) | (X) |
| NF3.2 Habilitar abas de Procedimentos Médicos e Acompanhantes | Dever ser selecionada uma pessoa para que as opções de cadastro para o <i>check-in</i> sejam carregadas | Regra de negócio | (X) | (X) |
| NF3.3 Janela Única | Todas as funções relacionadas ao cadastro de usuários devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 4 – Requisitos funcionais – Cadastrar *Check-In*

| F4 Efetuar <i>Check-Out</i> | | Oculto () | | |
|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O usuário efetua o <i>check-out</i> do paciente. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF4.1 Seleciona Estadia | O usuário deve selecionar a estadia, identificada por meio de um código único gerado <i>no check-in</i> . | Regra de Negócio | (X) | (X) |
| NF4.2 Janela Única | Todas as funções relacionadas ao <i>check-out</i> de pacientes devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 5 – Requisitos funcionais – Efetuar Check-Out

| F5 Efetuar Aprovação de Estadia | | Oculto () | | |
|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O usuário deve efetuar a aprovação da atual situação da estadia, sendo ela <i>check-in</i> ou <i>check-out</i> . | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF5.1 Seleciona Estadia | O usuário seleciona a estadia, identificada por meio de um código único gerado <i>no check-in</i> . | Regra de Negócio | (X) | (X) |
| NF5.2 Janela Única | Todas as funções relacionadas à aprovação de estadias devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 6 – Requisitos funcionais – Efetuar Aprovação de Estadia

| F6 Efetuar Alteração | | Oculto () | | |
|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deve permitir ao usuário efetuar alterações e correções nas estadias cadastradas. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF6.1 Seleciona Estadia | O usuário seleciona a estadia, identificada por meio de um código único gerado <i>no check-in</i> . | Regra de Negócio | (X) | (X) |
| NF6.2 Janela Única | Todas as funções relacionadas à alteração de estadias devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 7 – Requisitos funcionais – Efetuar Alteração

| F7 Efetuar Exclusão | | Oculto () | | |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deve permitir ao usuário efetuar a exclusão de uma estadia. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF7.1 Seleciona Estadia | O usuário seleciona a estadia, identificada por meio de um código único gerado <i>no check-in</i> . | Regra de Negócio | (X) | (X) |
| NF7.2 Janela Única | Todas as funções relacionadas à exclusão de estadias devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 8 – Requisitos funcionais – Efetuar Exclusão

| F8 Efetuar Consulta | | Oculto () | | |
|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deve permitir ao usuário efetuar consultas de estadias dos pacientes da casa de apoio. | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF8.1 Seleciona Estadia | O usuário seleciona a estadia, identificada por meio de um código único gerado <i>no check-in</i> . | Regra de Negócio | (X) | (X) |
| NF8.2 Janela Única | Todas as funções relacionadas às consultas de estadias devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 9 – Requisitos funcionais – Efetuar Consulta

| F9 Gerar Relatórios | | Oculto () | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------|-----------|------------|
| Descrição: O sistema deverá permitir que o usuário gere relatórios de lista de municípios, lista de pessoas, <i>check-in</i> completo e comprovante de entrega de <i>tickets</i> . | | | | |
| Requisitos Não Funcionais | | | | |
| Nome | Restrição | Categoria | Desejável | Permanente |
| NF9.1 Seleciona relatório | Verifica o relatório informado pelo usuário. | Regra de Negócio | (X) | (X) |
| NF9.2 Monta os Parâmetros de Pesquisa. | Os parâmetros de pesquisa do relatório serão montados em tela conforme o relatório selecionado | Interface | (X) | (X) |
| NF9.4 Janela Única | Todas as funções relacionadas a relatórios devem ser efetuadas em uma única janela. | Interface | (X) | (X) |

Quadro 10 – Requisitos funcionais – Gerar Relatórios

4.2.3 Descrição das Tabelas de Dados

A seguir são apresentadas as tabelas do banco de dados, sendo possível verificar a finalidade de cada uma, bem como o seu nome, a relação de campos, a definição das restrições de chave primária (*Primary Key*, ou PK) e chave estrangeira (*Foreign Key*, ou FK) e a descrição dos campos.

A Tabela 1 descreve os campos do cadastro destinado a armazenar os municípios de origem das pessoas atendidas pela casa de apoio.

Tabela 1- Tabela de Municípios.

| Nome do Campo | Tipo (tamanho) | PK ¹⁰ | FK ¹¹ | Descrição |
|---------------|----------------|------------------|------------------|-------------------------------------------------|
| ID | Integer | * | | Número sequencial para identificar o município. |
| MUNICIPIO | VarChar(50) | | | Nome do município. |
| UF | Char(2) | | | Unidade Federativa (ou estado) do município. |

¹⁰ PK- *Primary Key* (ou chave primária). Campo que apresenta um valor diferente para cada registro da tabela.

¹¹ FK- *Foreign Key*. Campo de relacionamento. Representa a chave primária da tabela relacionada.

A Tabela 2 descreve os campos do cadastro destinado a armazenar os dados das pessoas atendidas na casa de apoio e acompanhantes. Esta tabela se relaciona com a tabela de municípios (Tabela 1) por meio da chave estrangeira “ID_MUNICIPIO”.

Tabela 2- Tabela de Pessoas.

| Nome do Campo | Tipo (tamanho) | PK | FK | Descrição |
|----------------------|----------------|----|----|----------------------------------------------------------------------------------------------------------------------|
| ID | Integer | * | | Número sequencial para identificar a pessoa atendida na casa de apoio. |
| NOME | VarChar(100) | | | Nome da pessoa. |
| RG | Char(9) | | | RG da pessoa. |
| DATA_NASCIMENTO | Date | | | Data de nascimento da pessoa. |
| TELEFONE_RESIDENCIAL | VarChar(13) | | | Telefone residencial da pessoa. |
| TELEFONE_CELULAR | VarChar(13) | | | Telefone celular da pessoa. |
| ENDERECO | VarChar(100) | | | Endereço da pessoa. |
| NUMERO | Numeric(10,0) | | | Número do endereço da pessoa. |
| COMPLEMENTO | VarChar(100) | | | Informações que complementam o endereço da pessoa. |
| ID_MUNICIPIO | Integer | | * | Número que identifica o código do município de origem da pessoa. Campo de relacionamento com a tabela de municípios. |
| BAIRRO | VarChar(50) | | | Bairro do endereço da pessoa. |
| CEP | VarChar(8) | | | CEP do endereço da pessoa. |

A Tabela 3 descreve os campos do cadastro principal (mestre) destinado a armazenar as estadias das pessoas na casa de apoio. Esta tabela se relaciona com a tabela de pessoas (Tabela 2) por meio da chave estrangeira “ID_PESSOA”.

Tabela 3- Tabela de Estadias.

| Nome do Campo | Tipo (tamanho) | PK | FK | Descrição |
|--------------------|----------------|----|----|--------------------------------------------------------------------------------------------|
| ID | Integer | * | | Número sequencial para identificar a estadia. |
| ID_PESSOA | Integer | | * | Número que identifica o código da pessoa. Campo de relacionamento com a tabela de pessoas. |
| CONVENIADO | SmallInt | | | Indica se a pessoa é conveniada (-1) ou não (0). |
| ENTREGA_TICKETS | SmallInt | | | Indica se a pessoa entregou <i>tickets</i> alimentação (-1) ou não (0). |
| QUANTIDADE_TICKETS | Integer | | | Número informando a quantidade de <i>tickets</i> entregue pela pessoa. |
| SITUACAO | VarChar(9) | | | Descreve a situação em que se encontra a estadia, <i>check-in</i> ou <i>check-out</i> . |
| DATA_ENTRADA | TimeStamp | | | Data e hora que informam a entrada da pessoa na casa de apoio. |
| DATA_SAIDA | TimeStamp | | | Data e Hora que informam a saída da pessoa da casa de apoio. |

| | | | | |
|--------------------|----------|--|--|-----------------------------------------------------------------------------------------------------|
| PERIODO_INTERNACAO | SmallInt | | | Indica se o tempo que a pessoa irá passar na casa de apoio é período de internação (-1) ou não (0). |
| QUANTIDADE_DIARIA | Integer | | | Número informando a quantidade de dias que a pessoa passou na casa de apoio. |
| ACOMPANHANTE | SmallInt | | | Indica se a pessoa tem (-1) acompanhante com ela na casa de apoio ou não (0). |
| APROVADO | SmallInt | | | Indica se o <i>check-in</i> ou <i>check-out</i> do paciente está aprovado (-1) ou não (0). |

A Tabela 4 descreve os campos do cadastro secundário (detalhe) destinado a armazenar os acompanhantes dos pacientes com estadias em aberto. Esta tabela se relaciona com a tabela de estadias (Tabela 3) por meio da chave estrangeira “ID_ESTADIA” e com a tabela pessoas (Tabela 2) por meio da chave estrangeira “ID_PESSOA”.

Tabela 4- Tabela de Acompanhantes.

| Nome do Campo | Tipo (tamanho) | PK | FK | Descrição |
|-------------------|----------------|----|----|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| ID | Integer | * | | Número sequencial para identificar os acompanhantes. |
| ID_ESTADIA | Integer | | * | Número que identifica a qual estadia pertence o acompanhante. Campo de relacionamento com a tabela de estadias. |
| ID_PESSOA | Integer | | * | Número que identifica na tabela pessoas qual é o acompanhante da pessoa atendida na casa de apoio. Campo de relacionamento com a tabela de pessoas. |
| DATA_ENTRADA | TimeStamp | | | Data de entrada do acompanhante. |
| DATA_SAIDA | TimeStamp | | | Data de saída do acompanhante. |
| QUANTIDADE_DIARIA | Integer | | | Número de dias que o acompanhante ficou na casa de apoio. |

A Tabela 5 descreve os campos do cadastro destinado a armazenar os procedimentos médicos dos pacientes, realizados durante a sua estadia na casa de apoio. Esta tabela se relaciona com a tabela de estadias (Tabela 3) por meio da chave estrangeira “ID_ESTADIA”.

Tabela 5- Tabela de Procedimentos Médicos.

| Nome do Campo | Tipo (tamanho) | PK | FK | Descrição |
|---------------|----------------|----|----|--------------------------------------------------------------------------------------------------------------------|
| ID | Integer | * | | Número sequencial para identificar o procedimento médico. |
| ID_ESTADIA | Integer | | * | Número para identificar de qual estadia é o procedimento médico. Campo de relacionamento com a tabela de estadias. |

| | | | |
|--------------|--------------|--|----------------------------------------------------------------|
| LOCAL | VarChar(100) | | Descreve o local onde será realizado esse procedimento médico. |
| DATA_HORA | TimeStamp | | Data e hora em que será realizado o procedimento médico. |
| PROCEDIMENTO | VarChar(50) | | Procedimento médico que será realizado. |

4.2.4 Diagrama de Entidade e Relacionamento (DER)

As Figuras 11 e 12 mostram os DERs do sistema, desenvolvidos por meio da opção *Tools* ⇒ *Database designer* disponível na barra de menu da ferramenta administrativa IExpert (Figura 11). A Figura 12 apresenta um DER correspondente, gerada por meio da opção *Database* ⇒ *Compare to database and update model* disponível na barra de ferramentas do software *Dezign for Database* descrito na subseção 3.1.2, com o propósito de apresentar as tabelas de dados e o relacionamento existente entre elas.

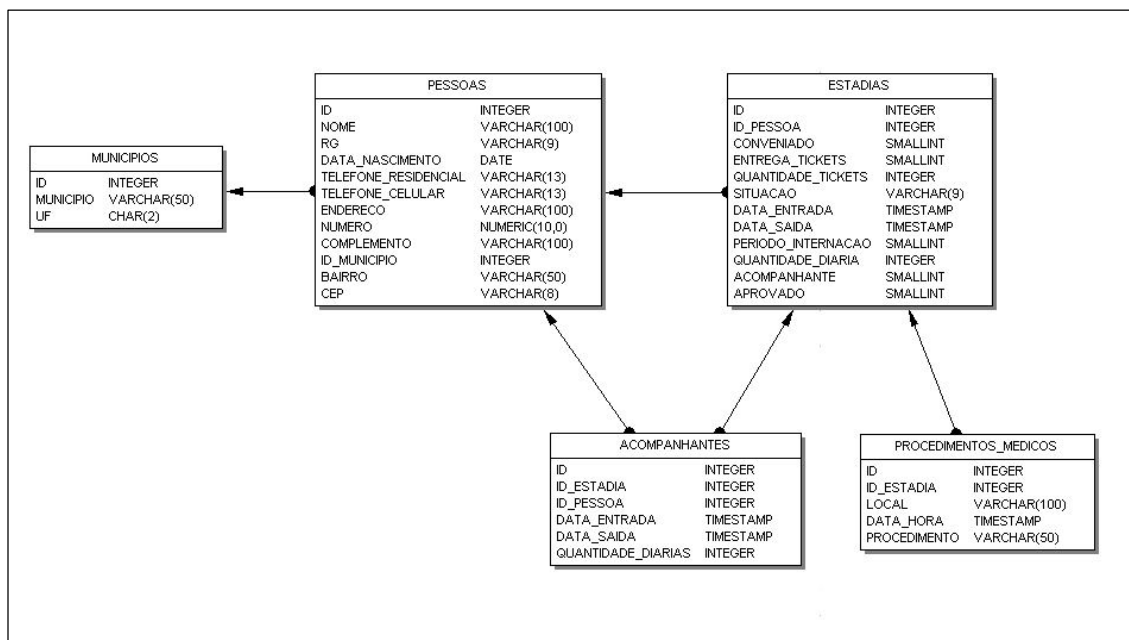


Figura 11- Diagrama de entidade e relacionamento do sistema proposto criado com o IExpert.

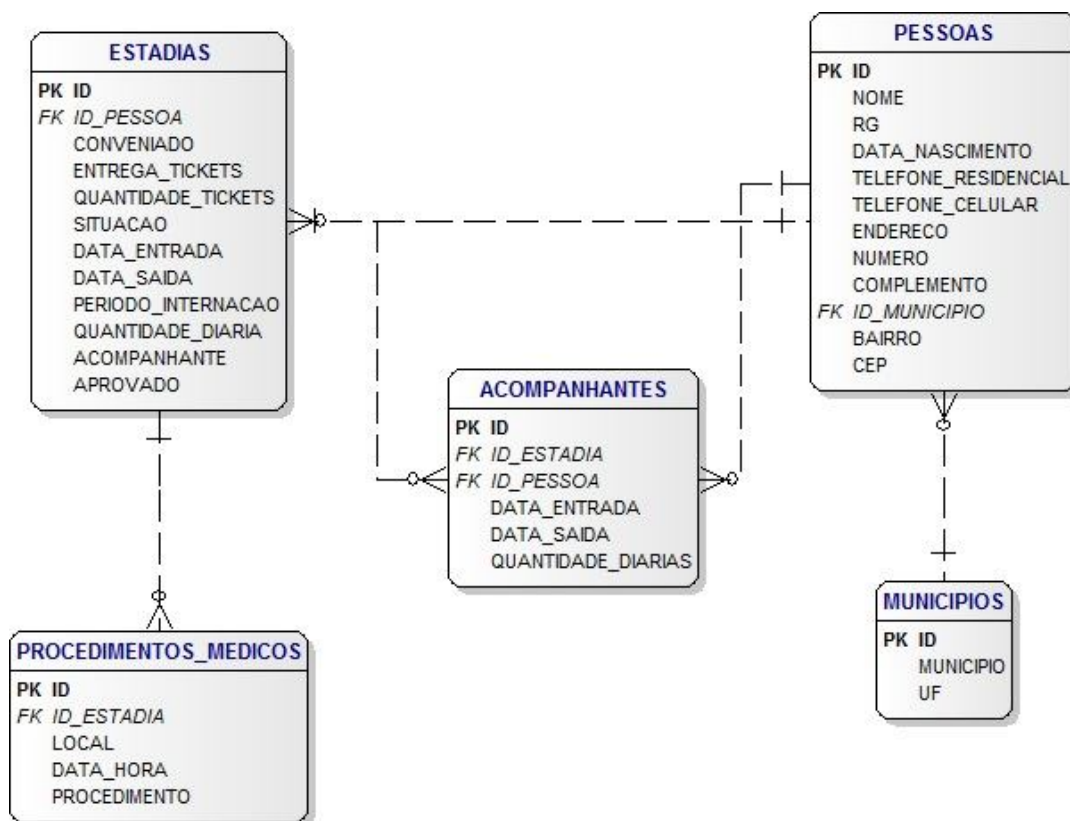


Figura 12- Diagrama de entidade e relacionamento do sistema proposto criado com o software Design for Database.

O projeto do banco de dados com o conjunto de 5 (cinco) tabelas descritas na subseção 4.2.1, foi implementado no banco de dados Firebird, descrito na subseção 3.1.1.

4.3 DESCRIÇÃO DAS FUNCIONALIDADES IMPLEMENTADAS

A linguagem de programação Delphi na versão XE, descrita na subseção 3.1.3, foi utilizada no desenvolvimento do sistema proposto para ambiente *desktop* descrito na seção 2.3 .

Para uma melhor apresentação dos requisitos funcionais do sistema, a tela principal, mostrada na Figura 13, está dividida em duas abas: “SGCA” e “Principal”. Na aba “Principal” as opções foram separadas em 3 (três) grupos:

1. **Grupo Estadia:** “Faz Check-In”, “Faz Check-Out”, “Aprova Check-Out”, “Consulta”, “Alteração” e “Exclusão”.
2. **Grupo Cadastro:** “Pessoas” e “Municípios”.
3. **Grupo Relatórios:** “Lista Pessoas”, “Municípios por UF”, “Check-In Completo” e “Comprovante Tickets”.

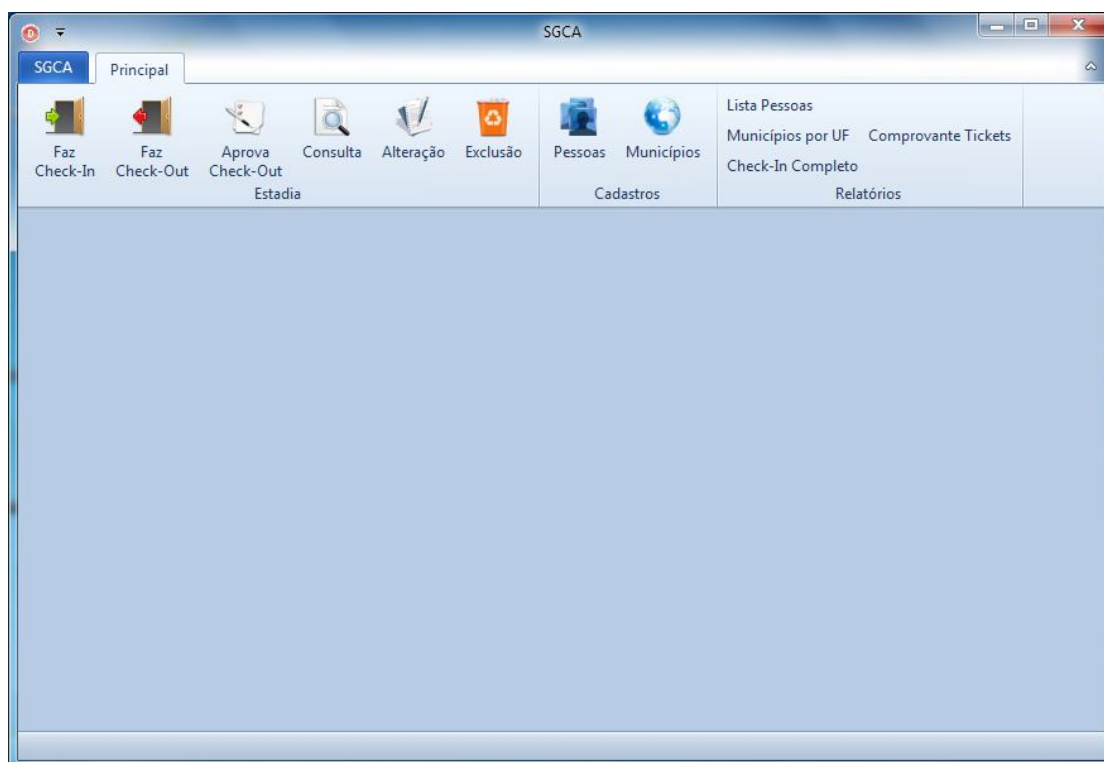


Figura 13- Tela Principal do Sistema Proposto.

Na aba “Principal”, grupo “Estadia”, encontram-se as opções para realizar a manutenção do *Check-In* e *Check-Out*, inclusão, alteração e exclusão, nas tabelas de dados do sistema. Já no grupo “Cadastro” encontram-se as opções para realizar os cadastros de pessoas e municípios do sistema. No grupo “Relatórios” encontram-se as opções para extrair as principais informações mantidas e geradas pelo sistema.

A relação com todas as funcionalidades implementadas pode ser observada na Tabela 6.

Tabela 6- Relação das Funcionalidades Implementadas.

| Funcionalidade (opção da aba Principal) | Descrição |
|--------------------------------------------|-----------|
| Grupo Estadia | |

| | |
|------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 1. Faz Check-In | Formulário ¹² para realizar o cadastro dos <i>check-ins</i> conforme campos descritos nas Tabelas 3, 4 e 5. |
| 2. Faz Check-Out | Formulário para realizar a atualização de <i>check-outs</i> conforme campos descritos nas Tabelas 3, 4 e 5. |
| 3. Aprova Check-Out | Formulário para realizar a aprovação dos <i>check-outs</i> dos pacientes conforme campos descritos nas Tabelas 3, 4 e 5. |
| 4. Consulta | Formulário para realizar consultas de <i>check-ins</i> |
| 5. Alteração | Formulário para realizar alterações cadastrais nas estadias dos pacientes conforme campos descritos nas Tabelas 3, 4 e 5. |
| 6. Exclusão | Formulário para realizar a exclusão das estadias. |
| Grupo Cadastro | |
| 1. Pessoas | Formulário para cadastrar pacientes e acompanhantes da hospedados na casa de apoio, conforme campos da Tabela 2. |
| 2. Municípios | Formulário para cadastro de municípios conforme campos descritos na Tabela 1. |
| Relatórios | |
| 1. Lista Pessoas | Relação cadastral das pessoas cadastradas conforme dados gravados na Tabela 2. |
| 2. Municípios por UF | Relação cadastral dos municípios agrupados por UF conforme dados gravados na Tabela 1. |
| 3. Check-In Completo | Relação com os dados dos <i>check-ins</i> , de forma individual, conforme dados gravados nas Tabelas 3, 4 e 5.. |
| 4. Comprovante Tickets | Relação dos <i>tickets</i> entregues para auxílio no momento do <i>check-in</i> conforme informado na Tabela 3. |

A seguir é apresentado um formulário feito com base no cadastro padrão (Figura 14), que foi base para todos os demais cadastros. Da mesma forma, a Figura 15 mostra um relatório desenvolvido para o sistema, que segue o mesmo modelo dos demais relatórios gerados.

Figura 14 - Cadastro de Pessoas preenchido.

¹² Um formulário no ambiente de desenvolvimento Delphi corresponde a uma janela, uma interface disponibilizada ao usuário para interagir com o sistema desenvolvido.

Check-In

Código: 1 Data de Entrada: 02/04/2013 02:06 Conveniado: Sim Entregou Ticks: Sim, 10

Informações do Paciente

Código: 1
Nome: Felipe Efrance Bez Gorio
R.G.: 99999999
Data de Nascimento: 24/3/1991
Telefone Residencial: 4632244820
Telefone Celular: 4691248492
Endereço: Rua Xingu
Número: 465
Complemento: Casa
Município: Pato Branco
UF: PR
CEP: 85501230

Procedimentos Médicos

| Local | Procedimento | Data e Hora |
|--------------------|-----------------------------------------|------------------|
| Consultório Médico | Consulta | 02/04/2013 02:06 |
| Laboratório | Realizar exames solicitados na consulta | 02/04/2013 02:07 |

Page 1 of 1

Figura 15 - Relatório de *Check-In*

4.4 ASPECTOS DESTACÁVEIS DA PROGRAMAÇÃO UTILIZADA

No desenvolvimento do projeto proposto pode-se destacar alguns aspectos importantes empregados na programação, como por exemplo: evento *OnCreate* do formulário principal; configurações do sistema por meio de arquivo INI; e, instruções SQL empregadas para organizar conjuntos de dados usados em formulários de cadastros e relatórios; foram desenvolvidos formulários padrões, como por exemplo o formulário padrão de cadastros, *untBaseCadastro*.

4.4.1 Instruções de Programação

O Quadro 11, apresenta 3 (três) instruções de programação implementadas no sistema proposto:

1. Função “DataToSQL”: utilizada para converter um valor do tipo data para o respectivo valor no formato texto com formatação SQL (é preciso utilizar o padrão americano, incluir aspas..). Utilizada ou invocada em filtros de instruções SQL como, por exemplo, na chamada *sWhere := sWhere + Properties.Items.Items[ItemIndex].Value + ' = ' + DataToSql(edtPesquisa.Text)*, onde a variável *sWhere*, do tipo texto, irá concatenar o valor retornado pelo filtro de data especificado. Procedimento do evento *OnCreate* do formulário principal: verifica a localização do banco de dados, e se o caminho do banco de dados for vazio, será solicitado para o usuário localizar onde está o banco de dados do sistema. Para esta verificação o sistema irá carregar o arquivo CFG.ini, que está localizado juntamente com o executável do sistema. Um arquivo INI, abreviação para *inicialization*, é um arquivo texto usado para armazenar/fornecer configurações pessoais para sistemas/usuários.
2. Procedimento utilizado para criar (método *create*) e abrir (método *show*) a tela de cadastro de pessoas: pacientes e acompanhantes (Tabela 2).

```
// [ 1 ] Função para converte valores do tipo data para formato texto
function DataToSql( Data : Double ): String; overload;
begin
  Result := #39 + FormatDateTime( 'mm/dd/yyyy', Data ) + #39;
end;
```

Quadro 11- Exemplos da programação Delphi utilizada no desenvolvimento do sistema proposto.

```

// [ 2 ] Procedimento de evento OnCreate do formulário principal:
// carrega informações na inicialização do sistema.
procedure TfrmPrincipal.FormCreate(Sender: TObject);
var
  ArqIni : TIniFile;
  sPath : string;
  i: Integer;
begin
  ArqIni := TIniFile.Create( ExtractFilePath( Application.ExeName ) +
    '\CFG.ini' );
  sPath := ArqIni.ReadString( 'Database', 'Path', '' );
  if sPath = '' then begin
    if Mensagem( 'Atenção', 'Não foi possível localizar o banco de dados,
deseja localiza-lo agora?', mbYesNo, mtWarning ) = ID_YES then begin
      if OPD.Execute then begin
        sPath := OPD.FileName;
        dm.Con.Params.Values[ 'Database' ] := sPath;
        try
          dm.Con.Open;
          ArqIni.WriteString( 'Database', 'Path', sPath );
        except on e:Exception do begin
          Mensagem( 'Erro de Conexão', 'Ocorreu algum problema de conectar
ao banco de dados.' + #13 + #10 + e.Message, mbOKCancel, mtError );
          Close;
        end;
      end;
    end;
  end;
  else
    Close;
  end;
  Close;
end;
else begin
  dm.Con.Params.Values[ 'Database' ] := sPath;
  dm.Con.Open;
end;
ArqIni.Free;
// FreeAndNil( sPath );
CarregaRelatorios;
end;

// [ 3 ] Abre (show) a tela de cadastro de Pessoas.
procedure TfrmPrincipal.dxBBarLargeButton1Click(Sender: TObject);
var
  frm : TfrmCadPessoas;
begin
  frm := TfrmCadPessoas.Create( application );
  frm.Show;
end;

```

Quadro 11 (cont.) - Exemplos de programação Delphi utilizada no desenvolvimento do sistema proposto.

A Figura 16 apresenta o formulário padrão usado como base para o desenvolvimento de cadastros. Essa forma de trabalho emprega o conceito de herança, da orientação a objetos, e é utilizada para o reaproveitamento dos códigos

elaborados e a redução do tempo de programação, isso devido às funções comuns existentes entre os formulários que serão criados a partir deste.

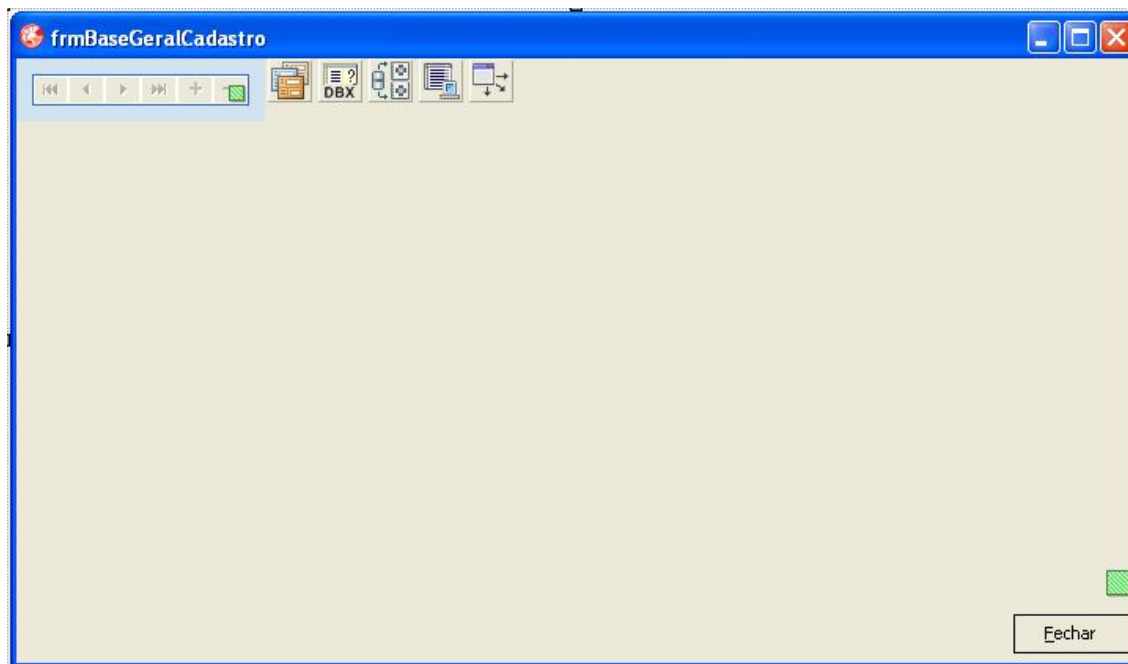


Figura 16 – Formulário, untBaseCadastro, utilizado como base de cadastros

4.4.2 Instruções SQL

O Quadro 12 apresenta 3 (três) instruções SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada), implementadas:

1. SQL utilizado no formulário para localização de pessoas, aberto por meio da aba "Principal" grupo "Cadastros" botão "Pessoas": No resultado da execução desta instrução estarão disponíveis todos os registros da tabela pessoas (Tabela 2) e para cada pessoa é associada as respectivas informações do seu município de origem, tabela municípios (Tabela 1), por meio do relacionamento (*left join*) definido pelo campo "id_municipio". Os registros no resultado serão ordenados pelo campo "id" da tabela pessoas.
2. Utilizado para carregar as informações mantidas na tabela municípios (Tabela 1) que serão utilizadas na tela de cadastro de municípios, carregado por meio da aba "Principal", grupo "Cadastros", botão "Municípios": Os registros no resultado

da execução desta instrução SQL serão ordenados pelo campo “id” da respectiva tabela.

3. Instrução SQL utilizada no relatório para imprimir o comprovante de entrega de *tickets*, aba “Principal”, grupo “Relatórios”, botão “Comprovante Tickets”: A execução desta instrução SQL retorna informações das pessoas (Tabela 2) e a quantidade de *tickets* entregues na estadia (Tabela 3). As tabelas estão ligadas por meio do campo “id” da tabela pessoa, com o campo “id_pessoa” da tabela estadias.

```
// [ 1 ] Localizar Pessoas
select p.*, m.*
from pessoas p
left join municipios m
on (p.id_municipio = m.id)
order by p.id

// [ 2 ] Localizar Município
select *
from Municipios
order by id

// [ 3 ] Comprovante de Entrega de Ticks
select p.*, e.quantidade_tickets, e.Data_Entrada, e.id asCodigo_Estadia
from Pessoas p
left join Estadias e
on (e.id_pessoa = p.id)
where e.id = :id
```

Quadro 12- Exemplos de instruções SQL implementadas.

5 CONCLUSÃO

O trabalho apresentado teve como objetivo o desenvolvimento de um sistema de software destinado ao gerenciamento de uma casa de apoio, sendo necessário para isso o desenvolvimento de cadastros e suas manutenções e relatórios a partir do ambiente de desenvolvimento Delphi XE.

Durante o período de desenvolvimento foram realizadas atividades de implementação de instruções SQL, utilização da ferramenta administrativa para banco de dados Firebird IBExpert e desenvolvimento de aplicação usando a linguagem de programação Delphi no ambiente de desenvolvimento Embarcadero Delphi XE.

Sendo assim, esse trabalho, ao atingir seus objetivos, permitiu ao acadêmico passar pela experiência de realizar todas as principais etapas do desenvolvimento de um software, desde o levantamento dos requisitos até a sua implantação. Isso fez com que um grande aprendizado pudesse ser gerado, servindo de base para o desenvolvimento de outros projetos.

Com relação à casa de apoio, esse trabalho ofereceu uma ferramenta eficiente para dar suporte ao gerenciamento de suas atividades e sem custo, permitindo ao acadêmico contribuir com a sociedade por meio de seus conhecimentos.

Com o sistema já em utilização, o acadêmico recebeu o *feedback* dos usuários, especialmente com relação à usabilidade, permitindo futuras melhorias ao protótipo desenvolvido.

REFERÊNCIAS

BERTOL, Omero Francisco. **Formulários Padrão**. Disponível em: <<http://www.pb.utfpr.edu.br/omero/Delphi/ppt/Form%20Padrao.zip>>, acessado em 19/06/2013.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2. ed. rev. e atual. Rio de Janeiro, RJ: Campus; Elsevier, 2006.

CANTU, Carlos Henrique. **Firebird - O Irmão do InterBase**. Revista SQL Magazine Edição 4, jan/2008. Disponível em: <<http://www.devmedia.com.br/artigo-sql-magazine-04-firebird-o-irmao-do-interbase/7644>>, acessado em 07/06/2013.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier, 2003.

FIREBIRD. Disponível em: <http://www.firebirdsql.org/en/logos/>, acessado em 05/03/2013.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação**. Rio de Janeiro: Livros Técnicos e Científicos, 1999.

LEVENEZ, Eric. **Languages**. Disponível em: <<http://www.levenez.com/lang/lang.pdf>>, acessado em 07/06/2013.

O'BRIEN, James A. **Sistemas de Informação e as Decisões Gerenciais na Era da Internet**. 9ª ed. São Paulo: Editora Saraiva, 2002.

PRESSMAN, R. **Engenharia de software**. Rio de Janeiro: McGrawHil, 2005.

REZENDE, Denis Alcides. **Engenharia de Software e sistemas de informação**. 3ª ed. rev. e ampl. Rio de Janeiro: Brasport, 2005.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. 3ª ed. São Paulo. Editora Pearson Makron Books, 2010.

WAZLAWICK Raul. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 2. ed. Rio de Janeiro: Elsevier, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 8ª ed. Editora Pearson, 2007.

ANEXOS

Anexo 1 – Especificação do processo de check-in fornecido pela colaboradora da Casa de Apoio

1- CHECK-IN:

Objetivo: Registrar Entrada de Pacientes na Casa de Apoio.

Pré-condições: O Paciente deve possuir ao menos um procedimento médico marcada para efetuar CHECK-IN na Casa de Apoio.

Fluxo Principal:

P01 – O sistema carrega os Municípios no campo combobox;
P02 – Atendente seleciona o Município de residência do Paciente;
P03 – Atendente marca opção Conveniado = SIM (A01);
P04 – Atendente marca opção Entrega de Ticket Alimentação = SIM, e informa a quantidade de Tickets entregues (A02);
P05 – Atendente clica no Botão Pesquisar, para selecionar Pessoa já cadastrada (A03);
P06 – O sistema apresenta a tela Pessoas em nova janela;
P07 – Atendente seleciona Pessoa e clica em Concluir;
P08 – O sistema fecha a janela, e apresenta os campos Código, Nome, RG, Data de Nascimento, Telefone Fixo, Celular, Endereço, Número, Complemento, Bairro, CEP, Município e Estado, preenchidos conforme cadastrado da tela Pessoas (A04);
P09 – Atendente informa ou seleciona a data e hora de Entrada do paciente (E01);
P10 – Atendente informa Local e descrição do procedimento médico;
P11 – Atendente informa Data e Hora;
P12 – Atendente Clica no Botão Incluir (A05);
P13 – O sistema apresenta os dados informados na Lista (A06, A07, A08, A22);
P14 – Atendente marca a opção Acompanhante = SIM (A11);
P15 – Atendente clica no Botão Pesquisar, para selecionar Pessoa já cadastrada (A12);
P16 – O sistema apresenta a tela Pessoas em nova janela;
P17 – Atendente seleciona Pessoa e clica em Concluir;
P18 – O sistema fecha a janela, e apresenta os campos Código, Nome, RG, Data de Nascimento, preenchidos conforme cadastrado na tela Pessoas (A13);
P19 – Atendente informa ou seleciona a data e hora de Entrada do Acompanhante (E01);
P20 – Atendente clica no Botão Incluir (A14);
P21 – O sistema apresenta os dados informados na Lista e limpa os campos Dados Acompanhante (A15, A16, A17, A23);
P22 – Atendente clica no Botão Concluir (A20);
P23 – O sistema verifica que todos os dados obrigatórios foram informados (E02, E03, E04, E05, E06, E07, E08, E09);
P24 – O sistema efetua o CHECK-IN com sucesso;
P25 – O sistema apresenta pop-up impressão do comprovante de CHECK-IN;
P26 – O sistema verificar que o campo Entrega de Tickets = SIM, e apresenta pop-up impressão do comprovante de Entrega do Tickets (A21);
P27 – O sistema limpa todos os campos da Tela, e retorna para o passo P01 do Fluxo Principal.

Fluxo Alternativo:

A01 – Município NÃO Conveniado

- 01 – Atendente marca opção Conveniado = NÃO;
- 02 – Prosseguir para o passo P04 do Fluxo Principal.

A02 – Não Entregar os Tickets Alimentação

- 01 – Atendente marca opção Entrega de Tickets = NÃO;
- 02 – Prosseguir para o passo P05 do Fluxo Principal.

A03 – Incluir Paciente

- 01 – Atendente aciona o botão Incluir – Dados do Paciente, para incluir novo paciente;
- 02 – O sistema apresenta a tela Pessoas em nova janela;
- 03 – Atendente inclui dados e clica em Concluir;
- 04 – O sistema fecha a janela, e apresenta os campos Código, Nome, RG, Data de Nascimento, Telefone Fixo, Celular, Endereço, Número, Complemento, Bairro, CEP, Município e Estado, preenchidos conforme cadastrado da tela Pessoas (A04);
- 05 – Prosseguir para o passo P09 do Fluxo Principal.

A04 – Editar Paciente

- 01 – Atendente aciona o botão Editar – Dados do Paciente, para alterar alguma informação incorreta;
- 02 – O sistema apresenta a tela Pessoas em nova Janela;
- 03 – Atendente altera informações e clica em Concluir;
- 04 – O sistema fecha a janela, e apresenta os campos Código, Nome, RG, Data de Nascimento, Telefone Fixo, Celular, Endereço, Número, Complemento, Bairro, CEP, Município e Estado, atualizados conforme alteração na tela Pessoas;
- 05 – Prosseguir para o passo P09 do Fluxo Principal.

A05 – Limpar Procedimentos Médicos

- 01 – Atendente clica no Botão Limpar;
- 02 – O sistema limpa os campos Procedimentos Médicos, exceto a Lista;
- 03 – Retorna para o passo P10 do Fluxo Principal.

A06 – Incluir Procedimentos Médicos

- 01 – Atendente informa local e descrição do Procedimentos Médicos;
- 02 – Atendente informa Data e Hora;
- 03 – Atendente clica no Botão Incluir (A05);
- 04 – O sistema apresenta as informações na Lista e limpa os campos Procedimentos Médicos (A05, A06, A07);
- 05 – Prossegue para o passo P14 do Fluxo Principal.

A07 – Editar Procedimentos Médicos

- 01 – Atendente seleciona um registro na lista e clica no Botão Editar;
- 02 – O sistema apresenta o registro selecionado nos campos Procedimentos Médicos;
- 03 – Atendente altera informações (local, data/Hora);
- 04 – Atendente clica no Botão Salvar (A05);
- 05 – O sistema altera as informações na Lista e limpa os campos Procedimentos Médicos (A06, A07, A08);
- 06 – Prossegue para o passo P14 do Fluxo Principal.

A08 – Excluir Procedimentos Médicos

- 01 – Atendente seleciona um registro na lista e clica no Botão Excluir;

- 02 – O sistema solicita confirmação da Exclusão;
- 03 – Atendente confirma exclusão (A09);
- 04 – O sistema exclui da lista o registro selecionado, e verifica que existem registros na lista (A10);
- 05 – Prossegue para o passo P14 do Fluxo Principal.

A09 – Cancelar Exclusão da Procedimentos Médicos

- 01 – Atendente Cancela Exclusão;
- 02 – Retorna para o passo 01 do Fluxo Alternativo Excluir Consulta.

A10 – Desabilitar botões Manutenção da Lista Procedimentos Médicos

- 01 – O sistema verifica que não existem mais registros na lista e retorna para o passo P10 do Fluxo Principal.

A11 – Paciente sem Acompanhante

- 01 – Atendente marca a opção Acompanhante = NÃO;
- 02 – Prosseguir para o passo P22 do Fluxo Principal.

A12 – Incluir Acompanhante

- 01 – Atendente aciona o botão Incluir – Dados do Acompanhante, para incluir novo paciente;
- 02 – O sistema apresenta a tela Pessoas em nova janela;
- 03 – Atendente inclui dados e clica em Concluir;
- 04 – O sistema fecha a janela, e apresenta os campos Código, Nome RG, Data de Nascimento e, preenchidos conforme cadastrado da tela Pessoas (A13);
- 05 – Prosseguir para o passo P19 do Fluxo Principal.

A13 – Editar Acompanhante

- 01 – Atendente aciona o botão Editar – Dados do Acompanhante, para alterar alguma informação incorreta;
- 02 – O sistema apresenta a tela Pessoas em nova janela;
- 03 – Atendente altera dados e clica em Concluir;
- 04 – O sistema fecha a janela, e apresenta os campos Código, Nome RG, Data de Nascimento atualizados conforme alteração na tela Pessoas;
- 05 – Prosseguir para o passo P19 do Fluxo Principal.

A14 – Limpar dados do Acompanhante

- 01 – Atendente clica no Botão Limpar;
- 02 – O sistema limpa os campos Código, Nome RG, Data de Nascimento e Data/Hora de Entrada;
- 03 – Retorna para o passo: P14 do Fluxo Principal.

A15 – Incluir Acompanhante

- 01 – Atendente marca a opção Acompanhante = SIM (A11);
- 02 – Atendente clica no Botão Pesquisar, para selecionar Pessoa já cadastrada ou cadastrar nova Pessoa;
- 03 – O sistema apresenta a tela Pessoas em nova janela;
- 04 – Atendente seleciona Pessoa e clica em Concluir;
- 05 – O sistema fecha a janela, e apresenta os campos Código, Nome, RG, Data de Nascimento, preenchidos conforme cadastrado na tela Pessoas;
- 06 – Atendente informa ou seleciona a data e hora de Entrada do Acompanhante (E01);

- 07 – Atendente clica no Botão Incluir (A14);
- 09 – O sistema apresenta as informações na Lista e limpa os campos Dados Acompanhante (A15, A16, A17);
- 10 – Prossegue para o passo P22 do Fluxo Principal.

A16 – Editar Acompanhante

- 01 – Atendente seleciona um Acompanhante na lista e clica no Botão Editar;
- 02 – O sistema apresenta os campos da lista selecionada, e habilita o Botão Salvar;
- 03 – Atendente altera Acompanhante clicando no botão Pesquisar, Data e/ou Hora de Entrada do Acompanhante;
- 04 – Atendente clica no Botão Salvar (A14);
- 05 – O sistema altera as informações na Lista e limpa os campos Dados Acompanhante (A15, A16, A17);
- 06 – Prossegue para o passo P22 do Fluxo Principal.

A17 – Excluir Acompanhante

- 01 – Atendente seleciona um registro na lista e clica no Botão Excluir;
- 02 – O sistema solicita confirmação da Exclusão;
- 03 – Atendente confirma exclusão (A18);
- 04 – O sistema excluir o registro da lista, e verifica que ainda existem registros na lista (A19);
- 05 – Prossegue para o passo P22 do Fluxo Principal.

A18 – Cancelar Exclusão de Acompanhante

- 01 – Atendente Cancela Exclusão;
- 02 – Retorna para o passo 01 do Fluxo Alternativo Excluir Acompanhante.

A19 – Desabilitar botões Manutenção da Lista Acompanhante

- 01 – O sistema verificar que não existem mais registros na lista e retorna para o passo P14 do Fluxo Principal.

A20 – Cancelar CHECK-IN

- 01 – Atendente clica no Botão Cancelar;
- 02 – O sistema limpa todos os campos da tela e retorna para o passo P01 do Fluxo Principal.

A21 – Não foram entregues os Tickets Alimentação

- 01 – O sistema verifica que o campo Entrega de Tickets = NÃO, e prossegue para o passo P27 do Fluxo Principal.

A22 – Visualizar Procedimentos Médicos

- 01 – Atendente seleciona um registro na lista e clica no Botão Visualizar;
- 02 – O sistema apresenta o registro selecionado nos campos Procedimentos Médicos.

A23 – Visualizar Acompanhante

- 01 – Atendente seleciona um registro na lista e clica no Botão Visualizar;
- 02 – O sistema apresenta o registro selecionado nos campos Dados Acompanhante.

Fluxo de Exceções:

E01 – Informar Data de Entrada retroativa inválida.

O sistema verifica que a Data informada é retroativa em mais de 1 dia útil, apresenta a mensagem “Data retroativa deve ser de até 1 dia útil.”, e retorna para o passo P09 do Fluxo Principal.

E02 – Não informar Município.

O sistema verifica que nenhum Município foi selecionado, e apresenta a mensagem “Obrigatório informar o Município.”, e retorna para o passo P25 do Fluxo Principal.

E03 – Não informar Conveniado.

O sistema verifica que nenhuma opção foi marcada para o Chekbox Conveniado, e apresenta a mensagem “Obrigatório informar se o município é ou não Conveniado.”, e retorna para o passo P22 do Fluxo Principal.

E04 – Não informar Entrega de Ticket Alimentação.

O sistema verifica que nenhuma opção foi marcada para Entrega de Ticket Alimentação, e apresenta a mensagem “Obrigatório informar se Ticket Alimentação foi ou não Entregue.”, e retorna para o passo P22 do Fluxo Principal.

E05 – Não informar quantidade de Ticket Alimentação.

O sistema verifica que a opção Entrega de Ticket Alimentação foi marcada com “Sim”, mas 0 ou nenhuma quantidade de tickets foi informada, e apresenta a mensagem “Obrigatório informar a quantidade de Tickets Alimentação entregues.”, e retorna para o passo P22 do Fluxo Principal.

E06 – Não selecionar nenhum Paciente na tela Pessoas.

O sistema verifica que o campo Código do item Dados do Paciente, não foi informado, e apresenta a mensagem “Obrigatório informar o Paciente.”, e retorna para o passo P22do Fluxo Principal.

E07 – Não foi selecionada a Entrada do Paciente.

O sistema verifica que o campo Entrada (data e hora), não foi informada, e apresenta a mensagem “Obrigatório informar a Data e Hora de Entrada do Paciente na Casa de Apoio.”, e retorna para o passo P22 do Fluxo Principal.

E08 – Nenhum Procedimento Médico informado.

O sistema verifica que nenhum procedimento médico foi inserido na Grid, e apresenta a mensagem “Obrigatório informar ao menos um procedimento médico.”, e retorna para o passo P22 do Fluxo Principal.

E09 – Nenhum Acompanhante informado.

O sistema verifica que a opção Acompanhante foi marcada com Sim, mas nenhum Acompanhante foi Inserido na Grid, e apresenta a mensagem “Para a opção Acompanhante SIM, é obrigatório informar ao menos um Acompanhante.”, e retorna para o passo P22 do Fluxo Principal.

Pós-condições: CHECK-IN efetuado. Impressão completa dos comprovantes de CHECK-IN e recebimento da quantidade de Tickets Alimentação.

Dados obrigatórios:

Município, Conveniado, Entrega de Ticket Alimentação, Dados do Paciente, ao menos um procedimento médico (com todos os campos), Acompanhante;

Para a opção Entrega de Ticket Alimentação = Sim, obrigatório informar a quantidade.

Para a opção Acompanhante= Sim, obrigatório informar ao menos um Acompanhante.