

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO**

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano e Prof. Luiz Angelo Valota Francisco

**USO DO BANCO DE DADOS NA ADMINISTRAÇÃO DE UMA
INSTITUIÇÃO DE LONGA PERMANÊNCIA**

Aluno: Beatriz Moreira Oliveira

Prontuário: 160321

São João da Boa Vista – SP

2019

Resumo

Este trabalho apresenta um pouco do projeto ‘Gerações’, criado para auxiliar e facilitar a administração de uma Instituição de Longa Permanência. Ele foi pensado para atender a crescente população idosa no município de São João da Boa Vista e região. Com foco no módulo de controle administrativo, é apresentado o processo de desenvolvimento do seu banco de dados, descrevendo cinco etapas principais: o levantamento de requisitos, a criação do diagrama e documentação dos casos de uso, a criação do modelo entidade-relacionamento, o modelo relacional e, por fim, a criação do modelo físico do banco de dados.

A dissertação apresenta a importância da existência de um sistema organizado, prático e de fácil acesso para o gerenciamento das instituições, destacando como o uso do banco de dados abrange e facilita alterações ou inserção de novos dados, assim como a visualização deles.

Palavras-chave: Banco de Dados; Instituição de Longa Permanência; MER; organização de dados.

Sumário

1	Introdução	7
1.1	Contextualização / Motivação	7
1.2	Objetivo Geral da Pesquisa	9
1.3	Objetivos Específicos	9
1.4	Estrutura do Documento	10
2	Desenvolvimento	11
2.1	Levantamento Bibliográfico	11
2.1.1.	Requisitos	11
2.1.2.	Dado vs Informação	11
2.1.3.	Banco de Dados	11
2.1.4.	Sistema de Banco de Dados	12
2.1.5.	SGBD's	13
2.1.6.	MER	13
2.1.7.	Modelo Relacional	15
2.2	Etapas para desenvolvimento da pesquisa	16
2.2.1	Objetivo específico 1: Identificação dos requisitos	16
2.2.2	Objetivo específico 2: Apresentar o Diagrama de Casos de Uso	21
2.2.3	Objetivo específico 3: Ferramenta brModelo	23
2.2.4	Objetivo específico 4: Modelagem Entidade-Relacionamento	24
2.2.5	Objetivo específico 5: Modelo Relacional	25
2.2.6	Objetivo específico 6: Modelo Físico Produzido para o Banco de Dados	26
3	Conclusões e Recomendações	28
4	Referências Bibliográficas	31

Índice de Ilustrações

Figura 1 - Pirâmide etária de São João da Boa Vista [1]	7
Figura 2 - Diagrama de Visão Geral do Módulo 08.....	8
Figura 3 - Representação simplificada de um sistema de banco de dados [9]	13
Figura 4 - Exemplo de MER do Módulo 08	14
Figura 5 - Entidades no MER.....	14
Figura 6 - Representação de atributos no MER	14
Figura 7 - Exemplo de Relacionamento	15
Figura 8 - Exemplo de Modelo Relacional do Módulo 08.....	15
Figura 9 - Diagrama de Caso de uso do Módulo 08.....	22
Figura 10 - Interface do brModelo	23
Figura 11 – Conversão de modelos e outras funcionalidades	24
Figura 12 - Modelo Entidade-Relacionamento do Módulo 08	25
Figura 13 - Modelo Relacional do Módulo 08.....	26
Figura 14 - Modelo Físico do Módulo 08	27

Índice de Tabelas

Tabela 1 - Requisitos Funcionais do Módulo 08.....	16
--	----

ÍNDICE DE ABREVIATURAS E SIGLAS

ANVISA	Agência Nacional de Vigilância Sanitária
DBA	Administrador de Banco de Dados (Data Base Administrator)
IBGE	Instituto Brasileiro de Geografia e Estatística
IDH	Índice de Desenvolvimento Humano
IFSP	Instituto Federal de Educação Ciência e Tecnologia de São Paulo
MCU	Modelo de Casos de Uso
MER	Modelo Entidade-Relacionamento
PDS	Prática de Desenvolvimento de Sistemas
SGDB	Sistema de Gerenciamento de Banco de Dados

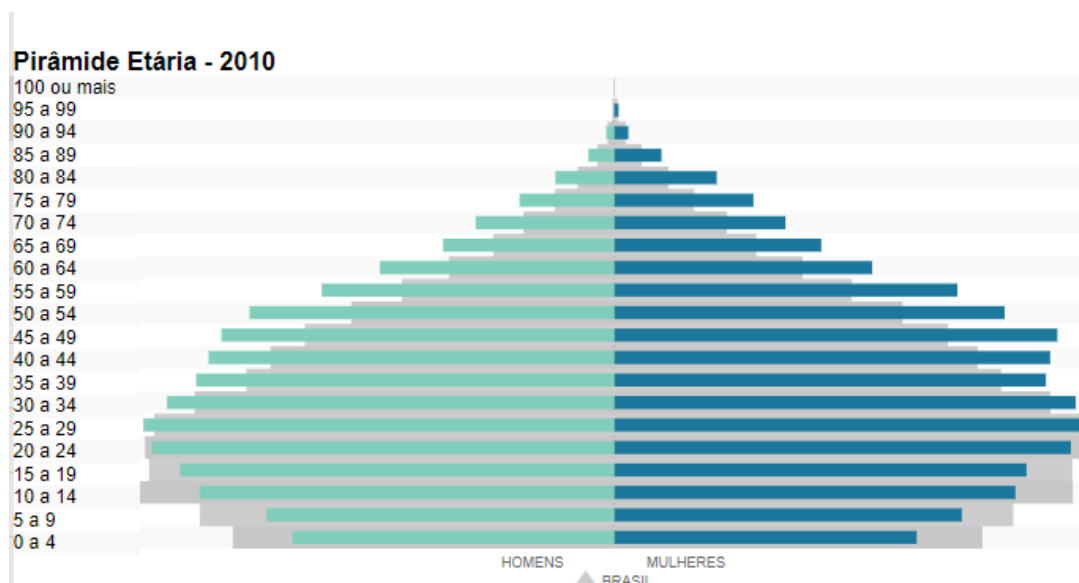
1 Introdução

1.1 Contextualização / Motivação

O município de São João da Boa Vista, presente no estado de São Paulo, conta com aproximadamente 90.637 habitantes, segundo a estimativa feita pelo IBGE no ano de 2018. O Índice de Desenvolvimento Humano (IDH) da população é de 0.797, sendo qualificado como 28º município com maior qualidade de vida no ranking do estado. [1]

O município também apareceu em uma pesquisa de 2017 sobre “As 40 melhores pequenas cidades para envelhecer”, sendo classificado em primeiro lugar [2]. Entre alguns aspectos utilizados para definir e classificar as cidades apresentadas nesse ranking, estão presentes tópicos de interesse da população, como saúde, segurança e qualidade de vida. Como consequência da qualificação em todos os requisitos, é possível observar uma alta quantidade de idosos presentes no município.

Figura 1 - Pirâmide etária de São João da Boa Vista [1]



Observando a quantificação da população idosa presente em na cidade, apesar dos requisitos da pesquisa e o seu IDH, percebe-se a necessidade de uma maior preocupação com a saúde dessa geração, contando com locais que ofereçam cuidados e assistência às famílias e idosos.

Segundo a Anvisa, Instituições de Longa Permanência são entendidas como instituições governamentais ou não-governamentais, que atuam como uma residência coletiva para pessoas da terceira idade, igual ou superior a 60 anos, em situação de carência financeira ou familiar, ou ainda

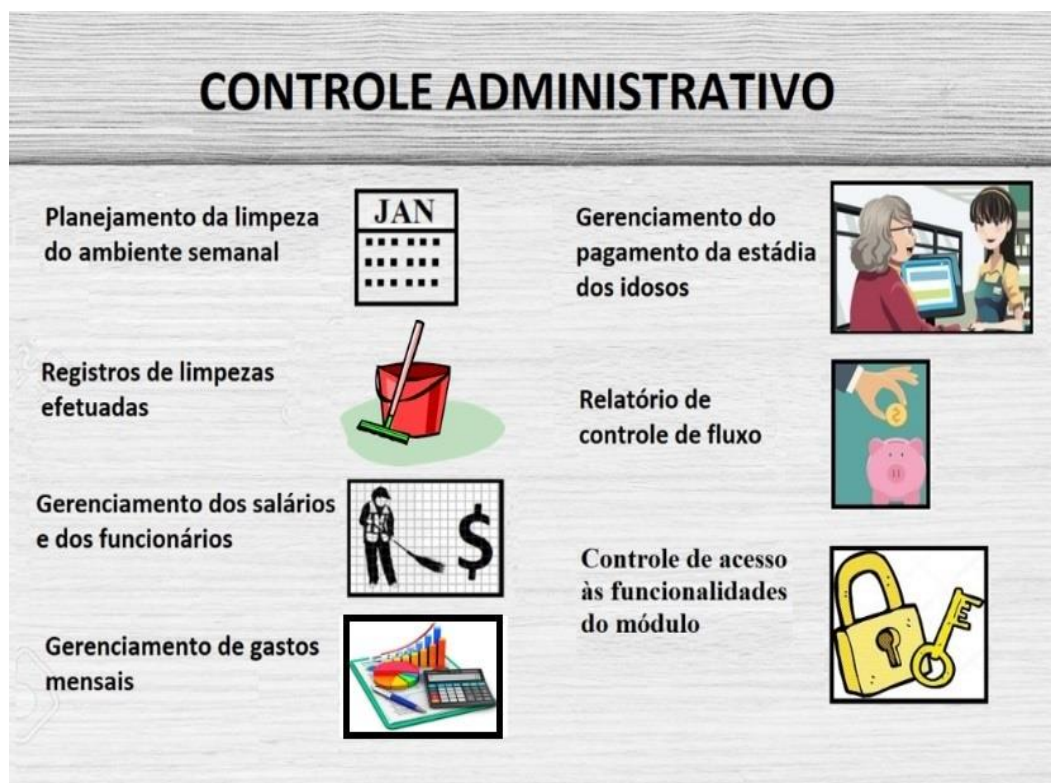
que demandem cuidados prolongados de pessoas especializadas. [3] A cidade em questão, assim como a região, não possui muitas instituições assim, principalmente governamentais.

O município também conta com um campus do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), que oferece cursos gratuitos para a população. Uma das modalidades existentes na instituição é o curso Técnico Integrado ao Ensino Médio, se divide entre o Técnico em Informática e Técnico em Eletrônica. [4]. Anualmente os alunos do 4º que cursam informática, desenvolvem sistemas na matéria “Prática de Desenvolvimento de Sistemas (PDS)”. Esses projetos têm como objetivo atender às necessidades específicas do município e região. Em 2019, o sistema em questão está sendo desenvolvido para atender a população idosa.

O projeto ‘Gerações’ tem como objetivo criar um sistema que auxilie a administração de uma Instituição de Longa Permanência, facilitando o acompanhamento dos idosos pelos profissionais responsáveis do local e pelos familiares, através de perfis designados a cada tipo de usuário. A ideia surgiu a partir de uma pesquisa realizada com uma das instituições presentes na cidade e região, que mostrou a necessidade de alguma ferramenta que facilitasse o armazenamento.

O projeto foi dividido em nove módulos, sendo um deles o de Controle Administrativo, foco deste trabalho. Esse módulo é responsável pela administração de finanças da instituição, como salários e gastos gerais existentes, além da gestão de limpeza. Segue abaixo uma imagem com resumo das principais funções do módulo:

Figura 2 - Diagrama de Visão Geral do Módulo 08



A fim de registrar as informações necessárias para cadastro de funcionários e contabilização da receita mensal e anual da instituição, é necessário o uso de um banco de dados. Essa ferramenta é essencial para uma melhor e mais prática gestão de dados que são importantes para a instituição, e que até então eram registrados em planilhas, que dificultam não só o acesso como a confiabilidade e organização das informações.

1.2 Objetivo Geral da Pesquisa

Esse documento possui como principais objetivos apresentar o projeto ‘Gerações’ (descrito no tópico 1.1) e o uso do banco de dados na área administrativa de uma Instituição de Longa Permanência. O modelo de referência para essa dissertação, é o banco de dados desenvolvido pelo Módulo 08, de Controle Administrativo.

Como apresentado no tópico anterior, o objetivo principal do projeto ‘Gerações’ é auxiliar na administração de uma Instituição de Longa Permanência. O banco de dados será utilizado para armazenar dados de usuários comuns, funcionários, medicamentos, finanças e vários outros. Isso será feito por meio da inserção de dados no banco de todos os módulos integrado, que foi desenvolvido pelos DBAs do projeto. Será necessário adaptar as informações à instituição, coletando dados de acordo com o seu funcionamento.

1.3 Objetivos Específicos

No tópico 1.1 houve uma breve introdução do projeto de sistema do ano de 2019, o ‘Gerações’. Ao longo do documento serão apresentados dados em forma de textos, figuras e tabelas, sendo majoritariamente o conteúdo desenvolvido durante o processo de criação do sistema.

As etapas que serão apresentadas envolvem diretamente o banco de dados, começando pelo levantamento de requisitos dos módulos, que envolve a coleta de informações e interpretação delas para representação em forma de dados; o diagrama de casos de uso, que apresenta as funções que cada módulo deve possuir; o MER (Modelo Entidade-Relacionamento) do módulo 8, assim como o seu modelo relacional e físico.

Também será apresentado o principal programa necessário para o desenvolvimento de todos os modelos, o brModelo. Os dados foram criados pelos alunos envolvidos com o Módulo 08 – Controle Administrativo do projeto, e serão utilizados para explicar o uso do banco de dados como ferramenta de auxílio na gestão de uma Instituição de Longa Permanência.

1.4 Estrutura do Documento

Capítulo 1: Introdução – Nesse capítulo serão apresentados dados gerais do município de São João da Boa Vista, a instituição de ensino IFSP, a disciplina PDS, o projeto ‘Gerações’ e a importância do seu banco de dados.

Capítulo 2: Desenvolvimento – Apresentação de alguns conceitos relacionados ao banco de dados, etapas para o processo de desenvolvimento do banco e descrição de algumas ferramentas utilizadas.

Capítulo 3: Conclusões e Recomendações – Conclusão e resultados da pesquisa; retoma os objetivos principais dessa dissertação e apresenta etapas de desenvolvimento do projeto ‘Gerações’.

2 Desenvolvimento

A criação de um banco de dados envolve um longo processo, desde o levantamento de requisitos que atendam às necessidades do cliente, até a manipulação e organização das informações em forma de dados. Para passar por todas as etapas desse processo, é essencial que haja conhecimento de diversos conceitos e ferramentas.

2.1 Levantamento Bibliográfico

Os conceitos e informações presentes nos tópicos abaixo foram retirados de materiais físicos e virtuais destinados ao ensino sobre o banco de dados e seu processo de criação.

2.1.1. Requisitos

Para entender a diferença entre dado e informação, é preciso ter em mente que o processo de criação do banco de dados envolve estudar o ambiente e/ou necessidade do cliente, realizando um levantamento de requisitos a fim de criar o melhor modelo que se adeque ao seu cenário.

O valor de algo tem origem de suas características. Para um software, elas podem ser divididas em funcionais, que servem para orientar os desenvolvedores em relação a como o sistema deverá funcionar a partir dos comandos do usuário, e os não funcionais, que se limitam a quantificar características específicas do comportamento; características não funcionais podem descrever elementos visuais e aspectos preferenciais em relação ao produto.

Os requisitos são então as características que irão definir como o produto final deve se apresentar para entrar nos critérios de aprovação do cliente. [5]

2.1.2. Dado vs Informação

Em síntese, dados podem ser entendidos como uma “matéria prima” que precisa ser manipulada, processada, a fim de se obter informações concretas e claras; não possuem significado por si só. [6]

Portanto, informação são os dados após serem processados. Segundo o site BInaprática, “informação é a ordenação e organização dos dados de forma a transmitir significado e compreensão dentro de um determinado contexto”. [7]

2.1.3. Banco de Dados

O objetivo de um banco de dados é armazenar dados de um cliente/empresa de maneira organizada e de fácil acesso. Imagine que tenhamos uma grande empresa que necessite do registro

de produção, estoque e vendas. Sem o uso do banco de dados, essas informações podem ser registradas de maneira incorreta e complicada, gerando uma redundância de dados; isso ocorre quando uma determinada informação é registrada repetidamente em um mesmo dispositivo. Há dois tipos de redundância:

- Controlada: ocorre quando o sistema está ciente das diversas cópias existentes de uma mesma informação e consegue conciliá-las de maneira que não seja prejudicial; pode ser utilizada propositalmente para testar e melhorar a confiabilidade de um sistema.
- Não controlada: ocorre quando o sistema não é o responsável pela sincronia da manutenção das versões existentes da mesma informação, e sim o usuário;

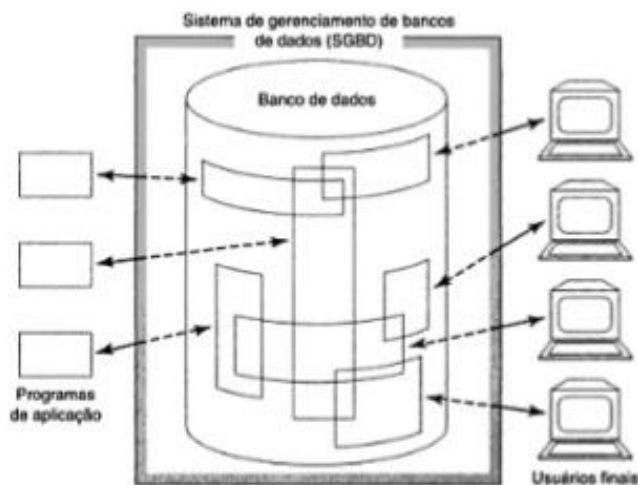
A redundância não controlada precisa ser evitada para que não ocorram casos em que a mesma informação seja armazenada repetidamente, além de impedir que haja uma inconsistência nos dados (quando os dados são erroneamente modificados).

Para evitar essas situações que possam causar grandes prejuízos às empresas, o ideal é que haja um compartilhamento de dados, permitindo que eles sejam acessados por vários sistemas quando necessário. O banco de dados pode ser definido então como um conjunto de dados integrados e sistematizados que tem por objetivo atender a uma comunidade de usuários, [8] ou ainda uma coleção de dados persistentes que pode ser usado em um sistema de uma empresa. [9]

2.1.4. Sistema de Banco de Dados

Um sistema de banco de dados é, resumidamente, uma manutenção de registros realizada de maneira computadorizada. Sua principal finalidade é permitir que os usuários busquem, alterem e excluam digitalmente os dados armazenados, quando assim for solicitado. Seu funcionamento envolve quatro componentes principais: dados, hardware, software e usuários. [9]

Figura 3 - Representação simplificada de um sistema de banco de dados [9]



2.1.5. SGBD's

Sistema de Gerenciamento de Banco de Dados ou Gerenciador de Banco de Dados, são os nomes atribuídos aos softwares que fazem a comunicação entre os usuários e o banco de dados físico [9]. É ele que incorpora funções como definição, alteração e recuperação em um banco de dados. [8] Todas as requisições feitas pelos usuários são processadas pelo SGDB. Sua função principal, segundo C. J. DATE, é “isolar os usuários do banco de dados dos detalhes no nível de hardware”; isso significa que o usuário pode visualizar o banco de dados de maneira complexa e permite que, a partir disso, ele execute diferentes operações.

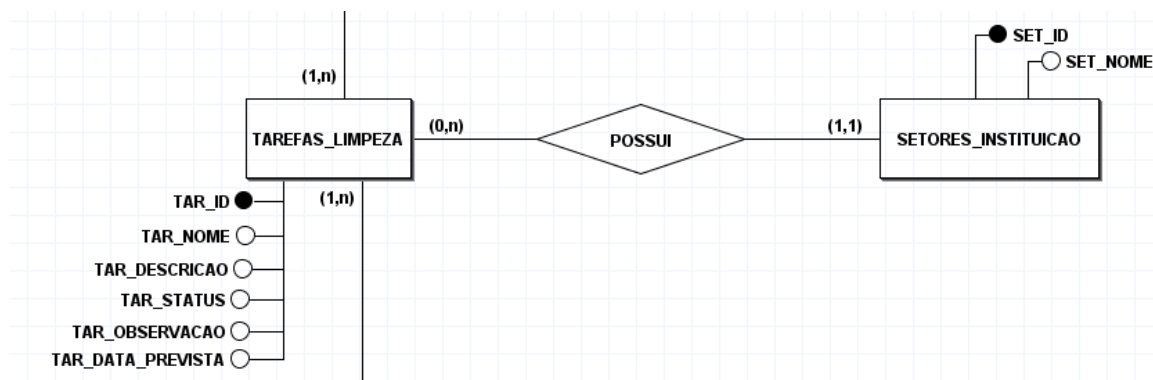
Quando se utiliza o termo usuários, pode-se referir a três tipos diferentes:

- Os programadores de aplicações, que utilizam linguagens de programação para inserir o uso do banco de dados em um sistema, como C/C++, Java, COBOL.
- Os usuários finais, que são os que acessam o banco de dados de maneira interativa, a partir dos sistemas desenvolvidos pelos programadores de aplicações.
- Os DBAs, Administradores de Banco de Dados, são as pessoas que fornecem suporte técnico para implementar as decisões estratégicas e normas criadas no processo de desenvolvimento do banco de dados. [9]

2.1.6. MER

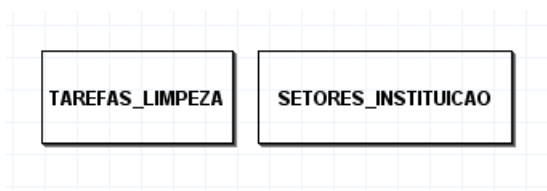
Um banco de dados pode ser descrito em vários modelos diferentes. O modelo entidade-relacionamento é abstrato, geralmente apresentado em formato de diagramas que são representados em entidades e relacionamentos. [8]

Figura 4 - Exemplo de MER do Módulo 08



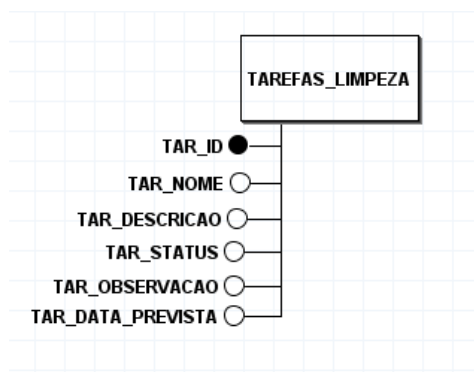
O conceito de entidade pode ser definido como um “conjunto de objetos sobre os quais deseja-se manter as informações no banco de dados”; elas são os alvos principais que irão possuir atributos para descrevê-las. No MER, a entidade é representada por um retângulo. [8]

Figura 5 - Entidades no MER



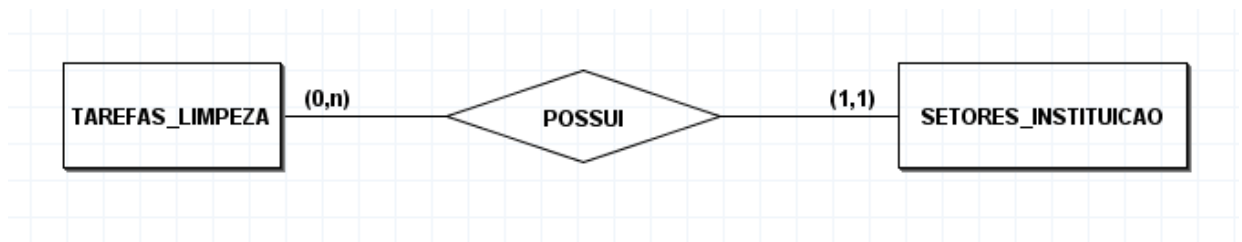
As entidades possuem dados que devem ser associados a elas; eles são chamados de atributos. Em geral, eles não possuem uma representação gráfica para evitar excesso de informações visuais, portanto é apresentado no modelo com um nome que foi atribuído a ele. [8]

Figura 6 - Representação de atributos no MER



Os relacionamentos são conjuntos de associações que acontecem entre objetos, no caso, as entidades. No modelo entidade-relacionamento, as relações são representadas por losangos [8]. Na figura abaixo, é possível compreender que as tarefas de limpeza de uma instituição (entidade) possuem (relacionamento) setores da mesma (entidade).

Figura 7 - Exemplo de Relacionamento



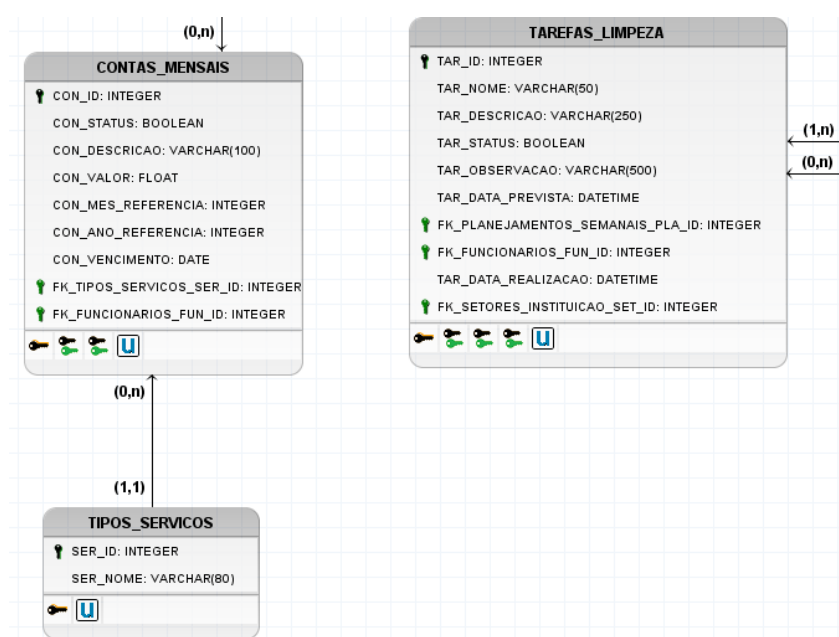
Um aspecto importante que deve ser considerado ao criar relacionamentos entre entidades, é a quantidade de ocorrências em que uma entidade está associada a outra: a essa propriedade se atribui o nome de cardinalidade. Devem ser levadas em consideração as cardinalidades mínimas e máximas em cada relacionamento [8]; no exemplo da figura 7, as tarefas de limpeza estão associadas obrigatoriamente (mínimo) a um setor e só podem ser atribuídas a ele (máximo).

2.1.7. Modelo Relacional

O modelo relacional de dados consiste na apresentação de dados em linhas dispostas em tabelas. Ele possui três aspectos principais:

- Estrutural: todas as informações são apresentadas em formato de tabelas e somente tabelas.
- Integridade: as tabelas devem atender a algumas restrições de integridade, como a definição correta das chaves primárias e estrangeiras.
- Manipulador: são os operadores que o usuário pode utilizar para gerenciar as tabelas.

Figura 8 - Exemplo de Modelo Relacional do Módulo 08



Dos aspectos manipuladores, existem três principais operadores, que são os de restrição, projeção e junção; a restrição ou seleção retira linhas de uma tabela, a projeção extrai colunas e a junção une duas tabelas baseada em valores comuns.

2.2 Etapas para desenvolvimento da pesquisa

Ao longo do desenvolvimento do projeto ‘Gerações’, algumas etapas foram realizadas para que o banco de dados pudesse ser desenvolvido

2.2.1 Objetivo específico 1: Identificação dos requisitos

Para dar início ao projeto, cada módulo fez um levantamento de requisitos de acordo com as suas funções. O documento de requisitos gerado serve de orientação para todas as etapas seguintes do projeto, portanto ele foi feito com atenção e modificado sempre que necessário.

A tabela abaixo apresenta, resumidamente, as principais exigências de cada requisito funcional:

Tabela 1 - Requisitos Funcionais do Módulo 08

Identificador	Descrição do Requisito
RF #01	<p>O sistema deve permitir que o representante da equipe de limpeza possa criar (insert), editar (update) e excluir (delete from table) tarefas no planejamento semanal da limpeza. Este usuário também poderá fazer (se estiver logado) uma marcação nas tarefas que já foram realizadas através de uma caixa de seleção que estará localizada logo no início da descrição.</p> <p>O planejamento semanal deverá ser criado pelo funcionário da limpeza que tenha sido escolhido como representante pela equipe. Para ele poder cadastrar o planejamento deverá aparecer uma tela com os seguintes campos nesta respectiva ordem:</p> <ul style="list-style-type: none">• Tabela: Planejamento Semanal• Nome da tarefa*: Varchar (50);• Descrição da tarefa*: Varchar (250);• Data de realização: Date-time;• Status da tarefa* (concluída ou não): Boolean (deverá indicar se está selecionado ou não).• Funcionário responsável pelo setor*: Varchar (100);• Nome do setor*: Varchar (50);• Observações: Varchar (500); <p>Abaixo dos campos de preenchimento deverá ter um botão (button) “Cadastrar” e um</p>

	<p>“Cancelar”. O botão Cadastrar deverá registrar as informações no banco de dados permitindo que elas sejam acessadas pelo usuário em uma pesquisa. O botão Cancelar deverá fechar a janela.</p> <p>O sistema deverá ter uma página de pesquisa onde será possível pesquisar pelas tarefas do dia de um funcionário específico visualizando isto a partir de uma tabela. Para acessar essa página o usuário deverá clicar na segunda opção do menu retrátil.</p> <p>Para realizar a pesquisa o usuário precisa preencher os campos “Data de início” (Date-time); “Data de término” (Date-time); “Setor” (button dropdown do bootstrap). Na frente desses campos deve haver um botão “Pesquisar” (select), quando o usuário clicar nele, o sistema deverá apresentar uma tabela contendo 5 colunas com os seguintes dados na respectiva ordem:</p> <ul style="list-style-type: none"> • A primeira linha da tabela conterá os nomes dos dados: “código”, “nome da tarefa”, “data da tarefa”, “setor” e “configurações”; • A primeira coluna deverá ser ocupada pelos códigos (autoincrement); • Nome da tarefa • Data de realização; • Nome do setor; • Dos ícones impressos na tela sendo eles para editar e outro para deletar.
RF #02	<p>Planejamento semanal: O sistema deverá conter uma opção do menu retrátil para a limpeza. Ao clicar nesse item o usuário deve ser direcionado para uma página onde deverá aparecer todos os setores da instituição que serão limpos e permitir que o gestor e os outros funcionários visualizem o planejamento semanal de cada um dos setores que o funcionário da limpeza representante da equipe criou (select). O planejamento será feito a partir de uma tabela, que aparecerá no formato de lista para o usuário. Nessa lista aparecerá o nome do funcionário responsável pela limpeza do setor e a tarefa que deve ser realizada (indicando se devem haver produtos específicos que não podem ser utilizados, se há algum cuidado especial com o setor, o que não deve ser movido, etc).</p> <ul style="list-style-type: none"> • O menu receberá o nome “controle administrativo” deverá aparecer quando o cursor do mouse passar em cima dele; <p>Na tabela deve aparecer, respectivamente:</p> <ul style="list-style-type: none"> • Nome do setor*: Varchar (50); • Data de limpeza: Date; • Descrição de como deve ser realizada a limpeza*: Varchar (250); • Funcionário responsável pelo setor: Varchar (100). <p>O funcionário deve logar no sistema e acessar a janela de limpezas para poder marcar uma tarefa como realizada. O sistema deverá imprimir na tela uma tabela com todas as tarefas</p>

	<p>que estão destinadas a este funcionário específico.</p> <p>O gestor tem acesso para visualizar todos os planejamentos semanais.</p>
RF #03	<p>O sistema deverá notificar o gestor quando todas as alas forem selecionadas (boolean) para que um novo planejamento seja feito. A mensagem deve aparecer como um pop-up na tela (utilizando o recurso Dismissing), quando ele acessar a aba de planejamento. Para que o gestor e o representante da limpeza saibam que existe um aviso, deverá aparecer uma bolinha de notificação na frente do nome da ala que teve o planejamento finalizado.</p>
RF #04	<p>O sistema deverá exibir a ficha de todos os funcionários previamente cadastrados, decorrente a sua categoria; essa ficha deve aparecer no formato de uma tabela que exiba todos os dados descritos abaixo. Deve também permitir que o gestor busque pelas fichas de cada funcionário, pesquisando pelo código de identificação ou nome completo, exibindo também sua função/cargo na instituição e seus respectivos salários. A barra de pesquisa deve estar no cabeçalho da própria tabela.</p> <ul style="list-style-type: none"> • Tabela: Funcionários; • Prontuário: Integer e autoincremento (USADO ID NOS MODELOS MER) • Nome do funcionário*: Varchar (100); • CPF* (XXX.XXX.XXX-XX): Inteiro (11), pontos e hífen automáticos; • Data de Nascimento* (DD/MM/AAAA): Date; • Salário*: Float; • Cargo*: Varchar (80); • Data de entrada na empresa* (DD/MM/AAAA): Date; • Status* (ativo ou inativo): Boolean; • Bônus salarial: Float; • Desconto salarial: Float. <p>O sistema também deverá permitir que o gestor ordene a busca dos funcionários de forma que os inativos (quem já não trabalha mais na empresa) apareçam primeiro. O usuário também poderá ordenar a busca (select na tabela de Funcionários) por ordem alfabética (A-Z / Z-A).</p>
RF #05	<p>O sistema deverá conter um segundo item de menu que receberá o nome 'Orçamentos'. Ao selecionar esse item, o usuário deverá ser direcionado a uma página exclusiva para o orçamento dos funcionários. O gestor deverá criar (insert) e/ou editar (update) os salários dos funcionários. Ele poderá adicionar (update) um bônus aos seus salários (variável presente no RF#04) ou realizar um desconto em caso de falta não justificada. Se ele pesquisar o nome do funcionário por CPF ou Nome Completo, deverá aparecer uma ficha do funcionário com suas informações. A ficha deverá ser em formato de tabela, contendo nome*, CPF*, salário*, data de entrada na empresa*, cargo*, data de nascimento*, status*</p>

	<p>(se está trabalhando, de férias, de licença ou afastado). Sempre que forem alterados o Status, Salário, o Bônus Salarial ou o Desconto Salarial, deve ser registrado a data/hora destas alterações, pois isto impacta diretamente na geração do fluxo de caixa mensal da instituição. O histórico deve ficar acessível para pesquisa do gestor.</p>
RF #06	<p>O sistema deverá permitir que o gestor possa criar (insert), editar (update) e buscar (select) os pagamentos mensais ocorridos na instituição (água, energia, gastos com alimentação, entre outros). Além disso, o gestor deverá adicionar a data de vencimento de cada pagamento, dessa forma ele poderá ser avisado caso não pague a conta dentro do prazo. A mensagem de aviso deverá aparecer como um pop-up e a bolinha de notificação deverá aparecer seguindo o mesmo esquema da notificação da tarefa descrito no RF#03.</p> <ul style="list-style-type: none"> • Tabela: Contas Mensais; • Nome da conta*: Varchar (50); • Descrição: Varchar (100); • Valor*: Float; • Data de vencimento* (DD/MM/AAAA): Date; • Status* (pago ou não): Boolean. <p>O sistema deve permitir que o gestor busque pelos pagamentos dos serviços da instituição. O sistema deve apresentar ao usuário os seguintes campos de preenchimento do filtro de pesquisa: nome do serviço, status (pago/devendo) e um botão ‘Pesquisar’;</p> <p>Ao realizar a pesquisa o sistema deverá imprimir na tela uma tabela contendo 5 colunas, sendo elas: o código (autoincremento), o nome do serviço, a data de vencimento, o status e na última dois ícones representando as opções “editar” e “registrar” (pedir uma confirmação se o gestor realmente pagou esta dívida e se a confirmação for “Sim” o sistema deverá registrar a data/hora no banco de dados).</p>
RF #07	<p>O gestor deve cadastrar o prontuário do funcionário, o sistema deverá permitir que o gestor crie (insert), edite (update) e busque (select na tabela de pagamentos) o pagamento dos pacientes da instituição, pesquisando por prontuário ou nome completo do idoso. Ao pesquisar um paciente, deverá aparecer sua ficha completa, seguindo o padrão de tabelas dos requisitos anteriores, apresentando todas as informações que foram salvas no cadastro, incluindo o seu histórico de pagamento.</p> <ul style="list-style-type: none"> • Tabela: Pagamento de Estadia dos Idosos; • Nome do Idoso*: Varchar (80); • Tipo de plano* (mensal, anual): Varchar (20); • Responsável pelo idoso*; • Tipo de pagamento: Varchar (50) (Ex.: Dinheiro, Depósito, Cheque, etc..); • Valor*: Float;

	<p>O sistema deve permitir que o gestor busque pelos pagamentos de estadia dos pacientes da instituição. O sistema deve apresentar ao usuário os seguintes campos de preenchimento do filtro de pesquisa: nome do idoso, tipo de pagamento, o valor, o status (pago/devendo) e um botão ‘Pesquisar’;</p> <p>Ao realizar a pesquisa o sistema deverá imprimir na tela uma tabela contendo 6 colunas, sendo elas: o código (autoincremento), o nome do idoso, o tipo de pagamento, o valor, o status e na última três ícones representando as opções “editar” e “registrar” (pedir uma confirmação se o gestor realmente recebeu este valor e se a confirmação for “Sim” o sistema deverá registrar a data/hora no banco de dados).</p> <p>Quando estiver na página do planejamento, o sistema deve permitir que o gestor faça uma busca ordenada (select) por ‘Planos Semanais’, ‘Planos Mensais’ e ‘Planos Anuais’, aparecendo assim os pacientes que se encaixam em cada categoria.</p>
RF #08	<p>O sistema deverá ter um terceiro item de menu que receberá o nome de doações, contendo o registro de todas as doações realizadas na instituição. O registro deverá conter o tipo da doação, sendo eles: doação monetária (float), alimentícia ou agasalhos.</p> <ul style="list-style-type: none"> • Tabela: Doações; • Nome da doação*: Varchar (50); • Tipo* (roupa, comida, dinheiro): Varchar (50); • Descrição: Varchar (300); • Valor (em caso de doação monetária): Float; <p>O sistema deve permitir que o gestor busque pelos pagamentos de estadia dos pacientes da instituição. O sistema deve apresentar ao usuário os seguintes campos de preenchimento do filtro de pesquisa: nome da doação, tipo da doação e um botão ‘Pesquisar’;</p> <p>Ao realizar a pesquisa o sistema deverá imprimir na tela uma tabela contendo 3 colunas, sendo elas: o código (autoincremento), o nome da doação, o tipo da doação e na última três ícones representando as opções “editar”, “excluir” (neste caso obrigatoriamente o sistema deve pedir uma confirmação) e “registrar” (pedir uma confirmação se a instituição realmente recebeu esta doação e se a confirmação for “Sim” o sistema deverá registrar a data/hora no banco de dados).</p>
RF #09	<p>A quarta opção do menu retrátil deverá abrir uma aba com todas as notificações que o gestor receber, independente do fato dele ter fechado elas, deverão ficar armazenadas na opção ‘Notificações’ do menu retrátil, na frente deverá estar o número de notificações que o gestor possui. No canto superior direito, logo acima das notificações deve ter uma opção de “Limpar” na qual se o gestor clicar deverá excluir todas as notificações desta aba. Se o gestor não quiser excluir tudo, mas apenas algumas específicas ele tem a opção de clicar em um “X” ao lado de cada notificação. Se o gestor limpar todas as notificações o número ao lado da palavra “Notificações” deverá aparecer como “0” (zero).</p>

	As notificações que serão enviadas ao gestor são referentes a contas pendentes da instituição (sendo elas contas mensais, salários de funcionários e atualização do pagamento dos idosos), conclusão do planejamento semanal.
RF #10	<p>O sistema deverá imprimir na tela duas tabelas relacionadas ao fluxo de caixa. A primeira deverá apresentar todas as receitas da empresa e o valor de todas elas, o gestor NÃO vai digitar esses dados, o sistema irá trazê-los do banco de dados. Ao final da tabela o sistema deve fazer um auto soma de todos os valores e imprimir o valor total de receitas na tela.</p> <p>A segunda tabela segue o mesmo esquema, apresentando todas as despesas da instituição e o valor de todas elas. Ao final da tabela o sistema deve fazer um auto soma e imprimir o valor total de despesas na tela.</p> <p>Os valores das despesas devem estar na cor vermelha e os valores das receitas devem estar na cor verde.</p> <p>Ao final das duas tabelas o sistema deve imprimir na tela o valor total de dinheiro na instituição. Se o valor for positivo a cor deve ser verde, mas se o valor for negativo a cor deve ser vermelha.</p>
RF #11	O sistema deverá conter um menu retrátil, de forma que quando o usuário posicionar o mouse sobre este botão deverá aparecer um menu com 4 opções: 'Limpeza', 'Orçamento', 'Doações' e "Notificações". Quando um destes itens de menu for selecionado, o usuário será direcionado a uma página específica em cada caso. Não há um lugar específico no qual esse menu deve ser inserido, pois depende do template escolhido.

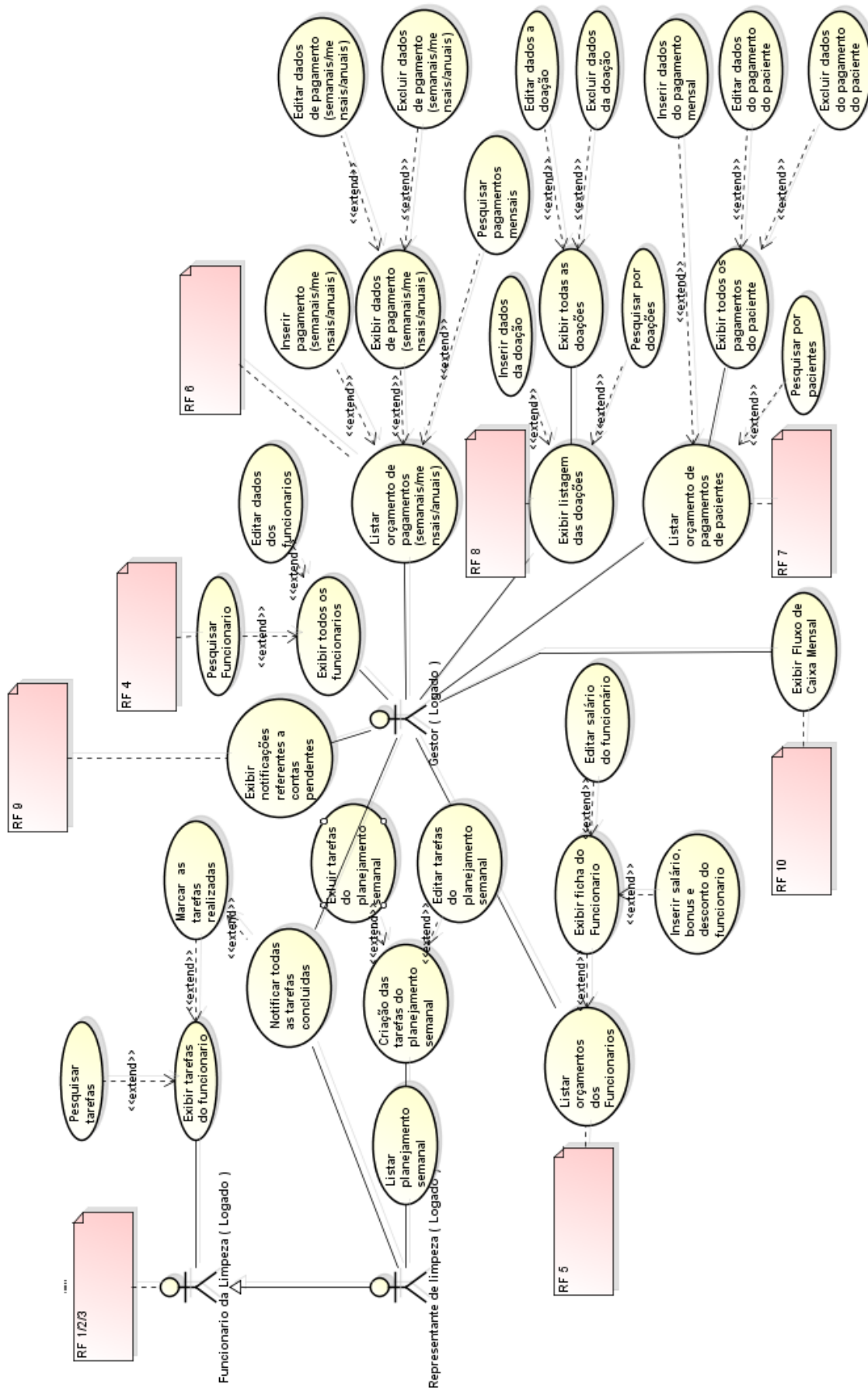
2.2.2 Objetivo específico 2: Apresentar o Diagrama de Casos de Uso

O diagrama de caso de uso, feito por analistas, DBAs e desenvolvedores, apresenta as principais funções do módulo de maneira simples. Ele foi desenvolvido a partir dos requisitos levantados anteriormente.

Para a criação deste caso de uso, foram identificados três principais atores: o gestor, responsável pela administração de funcionários e finanças; o funcionário da limpeza, responsável pela limpeza dos setores na instituição; e o representante de limpeza, que é um funcionário comum, porém é ele quem cadastra as informações do planejamento semanal.

Os casos de uso, interações de cada ator com o sistema, foram separados por requisito.

Figura 9 - Diagrama de Caso de uso do Módulo 08



Dividindo em etapas, a primeira parte do diagrama apresenta os requisitos funcionais 1, 2, 3, e 9, que são referentes à criação, edição, exclusão e pesquisa de tarefas, assim como marcação de tarefas realizadas.

O requisito 5 foi apresentado como os casos de uso listar e exibir funcionários, editar salário e inserir bônus ou desconto salarial.

A segunda parte apresenta os requisitos 4, 6, 7, 8 e 10; os casos de uso criados se dividem entre exibir, editar e pesquisar os funcionários, as ações referentes ao orçamento de entrada e saída da instituição (que são realizadas somente pelo gestor), o gerenciamento das doações e o fluxo de caixa.

2.2.3 Objetivo específico 3: Ferramenta brModelo

Para a criação dos modelos que serão apresentados posteriormente, foi necessário o uso do programa brModelo, uma ferramenta criada em 2005 para modelagem de dados relacionais. Ela utiliza Java e foi programada em código aberto. Possui uma interface de fácil entendimento e permite a conversão de modelos (MER para relacional, relacional para físico), facilitando o processo de desenvolvimento do banco de dados.

Figura 10 - Interface do brModelo

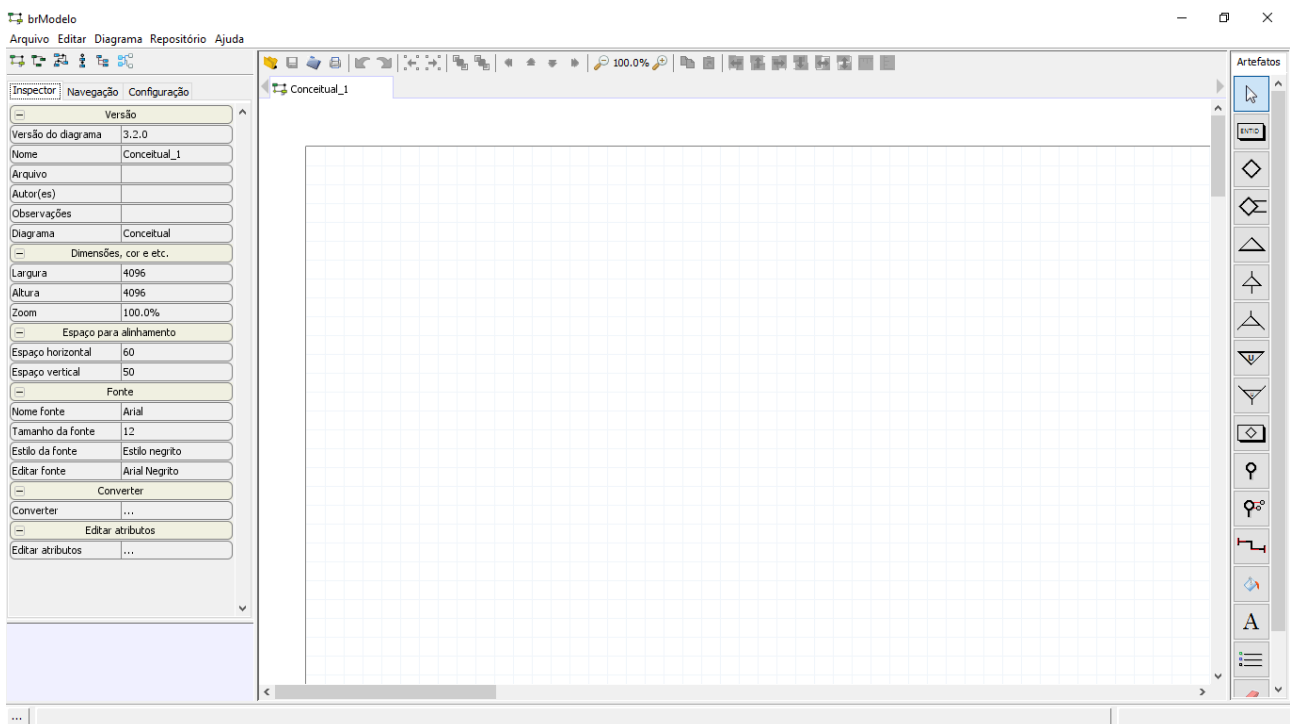
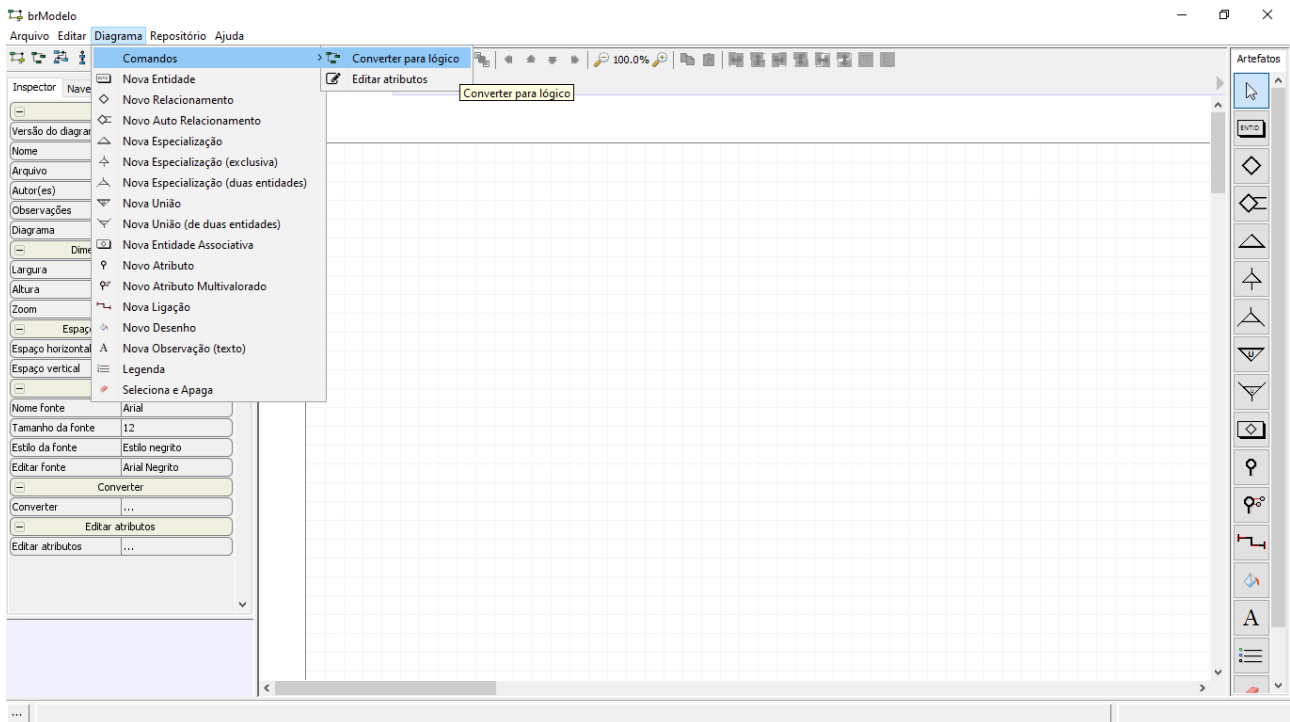


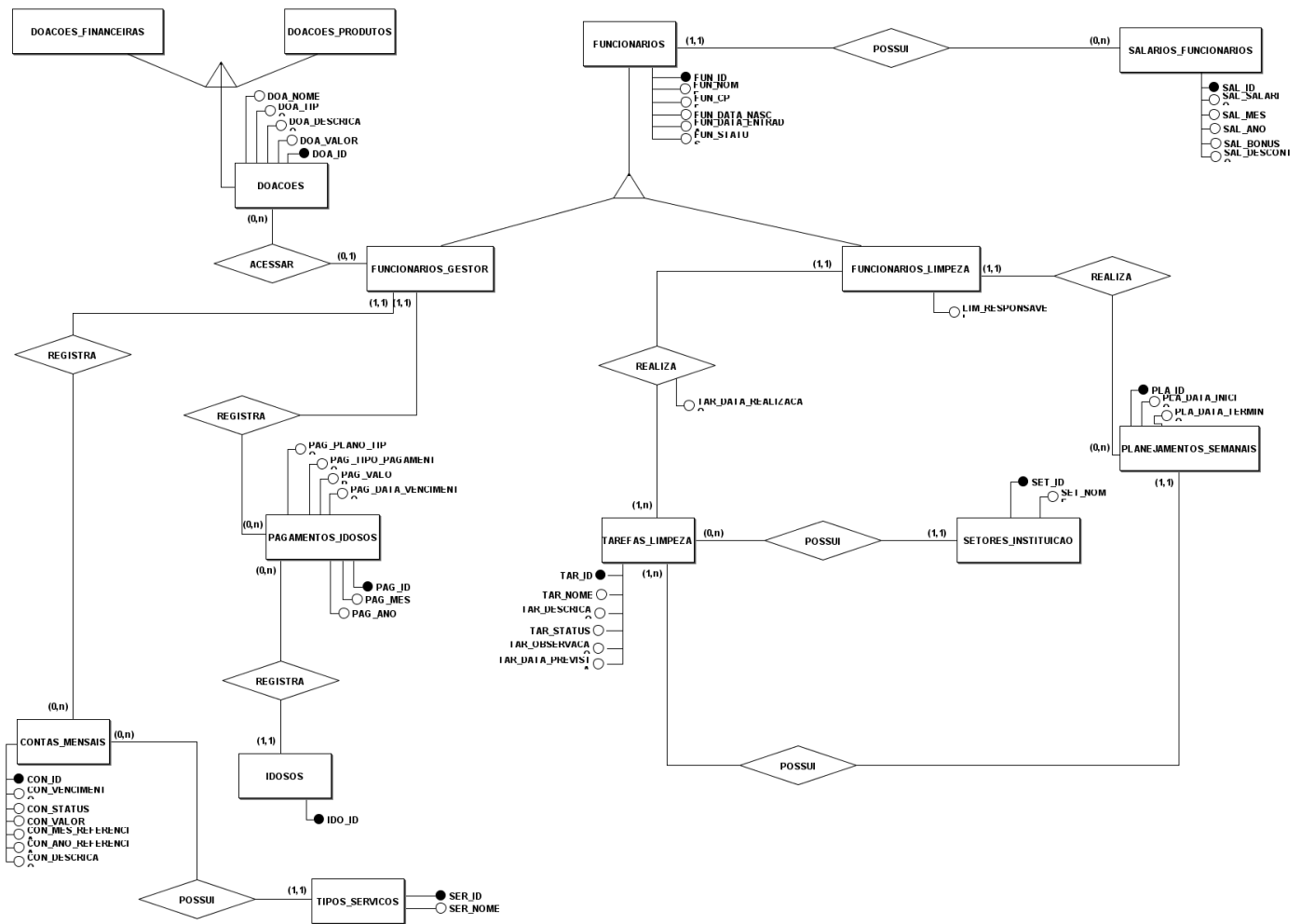
Figura 11 – Conversão de modelos e outras funcionalidades



2.2.4 Objetivo específico 4: Modelagem Entidade-Relacionamento

O banco de dados foi criado a partir dos documentos de requisito e casos de uso. A criação do modelo entidade-relacionamento envolveu DBAs e analistas, interpretando as funcionalidades do módulo para criar as principais entidades que seriam necessárias para o melhor funcionamento do sistema e armazenamento dos dados. Foram definidas então as entidades: Doações (herdando doações financeiras e produtos), Funcionários (herdando gestor e funcionários de limpeza), Salários, Planejamentos Semanais, Tarefas de Limpeza, Setores, Contas Mensais, Tipos de Serviço, Idosos e Pagamentos dos Idosos. Abaixo o MER do módulo:

Figura 12 - Modelo Entidade-Relacionamento do Módulo 08



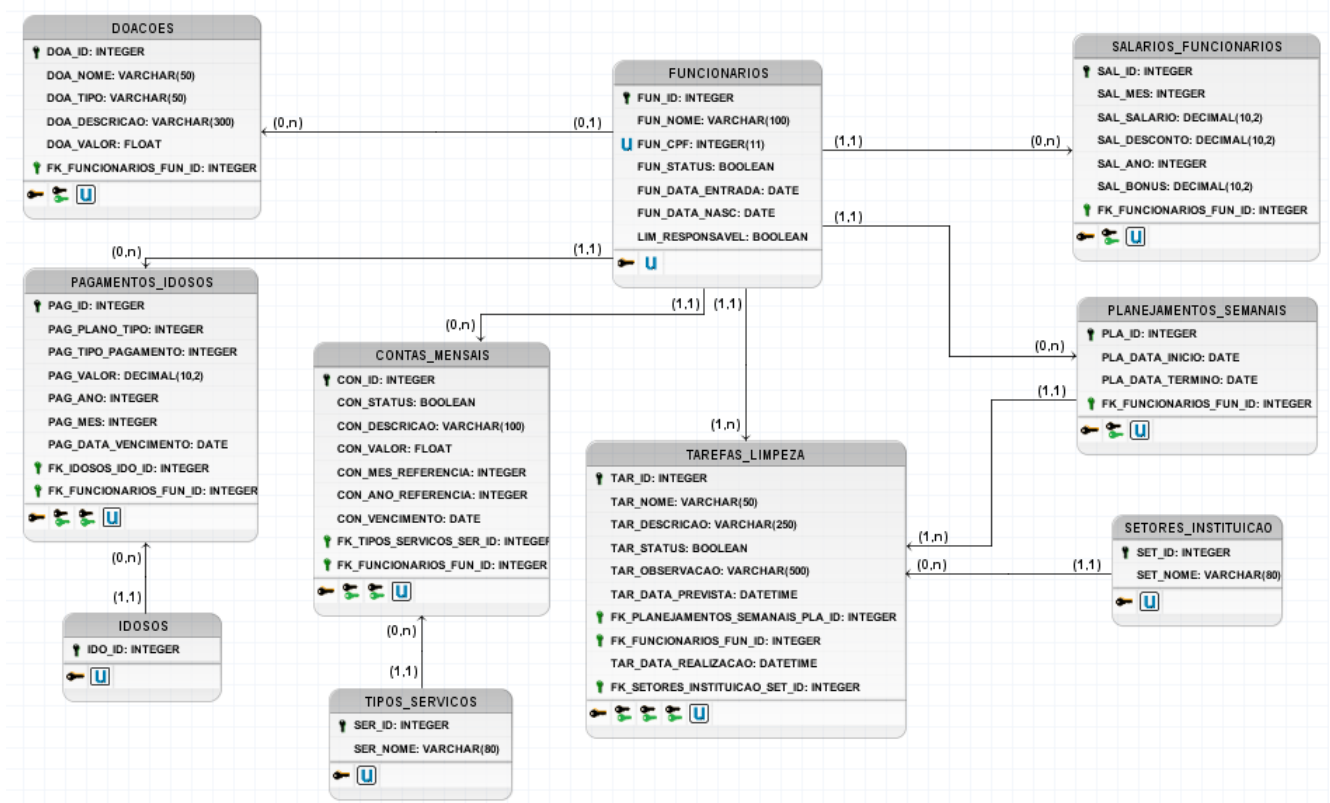
Cada entidade recebeu atributos para cadastrar suas informações essenciais. Para fins de padronização, os nomes de todos os atributos começam com 3 letras relacionadas à entidade (como em “TAR_NOME”, da entidade TAREFAS_LIMPEZA) e em caixa alta. Também foi pedido que as entidades fosse nomeadas no plural.

Esse modelo é essencial para a apresentação do projeto a terceiros, pois apresenta de maneira simples a relação entre as funções (entidades) do sistema e quais informações serão cadastradas, sendo possível uma melhor compreensão do banco de dados e o sistema como um todo.

2.2.5 Objetivo específico 5: Modelo Relacional

O modelo relacional foi gerado a partir do MER e adaptado para corrigir pequenos erros. O diferencial desse modelo é a presença dos tipos de cada atributo, o que serve de orientação para os desenvolvedores e para os próprios DBAs na inserção de dados no banco.

Figura 13 - Modelo Relacional do Módulo 08



A presença de cardinalidades e a apresentação dos dados em formato de tabela o tornam um modelo também essencial para a apresentação do projeto, pois complementa o MER e demonstra de maneira mais clara as relações de dependência de cada entidade. Na conversão, as entidades passam a se conectar a partir das chaves primárias e estrangeiras; nesse modelo, tornam-se: Doações, Funcionários, Salários, Planejamentos Semanais, Tarefas de Limpeza, Setores, Idosos, Pagamentos dos Idosos, Contas Mensais e Tipos de Serviços.

2.2.6 Objetivo específico 6: Modelo Físico Produzido para o Banco de Dados

Após a criação do modelo relacional e a definição dos tipos de cada atributo, é possível gerar o modelo físico. Esse é o modelo final que será usado no script do banco de dados para que ele seja criado e permita a utilização das funções, como inserção, edição e exclusão de dados. Segue abaixo o modelo físico do módulo 8:

Figura 14 - Modelo Físico do Módulo 08

```

CREATE TABLE TAREFAS_LIMPEZA (
    TAR_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    TAR_NOME VARCHAR(50) NOT NULL,
    TAR_DESCRICAO VARCHAR(250) NOT NULL,
    TAR_STATUS BOOLEAN NOT NULL,
    TAR_OBSERVACAO VARCHAR(500),
    TAR_DATA_PREVISTA DATETIME NOT NULL,
    FK_PLANEJAMENTOS_SEMANAIS_PLA_ID INTEGER,
    FK_FUNCIONARIOS_FUN_ID INTEGER,
    TAR_DATA_REALIZACAO DATETIME NOT NULL,
    FK_SETORES_INSTITUICAO_SET_ID INTEGER
);

CREATE TABLE DOACOES (
    DOA_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    DOA_NOME VARCHAR(50) NOT NULL,
    DOA_TIPO VARCHAR(50),
    DOA_DESCRICAO VARCHAR(300),
    DOA_VALOR FLOAT NOT NULL,
    FK_FUNCIONARIOS_FUN_ID INTEGER
);

CREATE TABLE PLANEJAMENTOS_SEMANAIS (
    PLA_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    PLA_DATA_INICIO DATE NOT NULL,
    PLA_DATA_TERMINO DATE NOT NULL,
    FK_FUNCIONARIOS_FUN_ID INTEGER
);

CREATE TABLE CONTAS_MENSAIS (
    CON_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    CON_STATUS BOOLEAN NOT NULL,
    CON_DESCRICAO VARCHAR(100),
    CON_VALOR FLOAT NOT NULL,
    CON_MES_REFERENCIA INTEGER NOT NULL,
    CON_ANO_REFERENCIA INTEGER NOT NULL,
    CON_VENCIMENTO DATE NOT NULL,
    FK_TIPOS_SERVICOS_SER_ID INTEGER,
    FK_FUNCIONARIOS_FUN_ID INTEGER
);

CREATE TABLE IDOSOS (
    IDO_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE
);

CREATE TABLE PAGAMENTOS_IDOSOS (
    PAG_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    PAG_PLANO_TIPO INTEGER NOT NULL,
    PAG_TIPO_PAGAMENTO INTEGER NOT NULL,
    PAG_VALOR DECIMAL(10,2) NOT NULL,
    PAG_ANO INTEGER NOT NULL,
    PAG_MES INTEGER NOT NULL,
    PAG_DATA_VENCIMENTO DATE NOT NULL,
    FK_IDOSOS_IDO_ID INTEGER,
    FK_FUNCIONARIOS_FUN_ID INTEGER
);

CREATE TABLE SETORES_INSTITUICAO (
    SET_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    SET_NOME VARCHAR(80) NOT NULL
);

CREATE TABLE FUNCIONARIOS (
    FUN_ID INTEGER AUTO_INCREMENT PRIMARY KEY,
    FUN_NOME VARCHAR(100) NOT NULL,
    FUN_CPF INTEGER(11) NOT NULL,
    FUN_STATUS BOOLEAN NOT NULL,
    FUN_DATA_ENTRADA DATE NOT NULL,
    FUN_DATA_NASC DATE NOT NULL,
    LIM_RESPONSAVEL BOOLEAN NOT NULL,
    UNIQUE (FUN_ID, FUN_CPF)
);

CREATE TABLE SALARIOS_FUNCIONARIOS (
    SAL_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    SAL_MES INTEGER NOT NULL,
    SAL_SALARIO DECIMAL(10,2) NOT NULL,
    SAL_DESCONTO DECIMAL(10,2) NOT NULL,
    SAL_ANO INTEGER NOT NULL,
    SAL_BONUS DECIMAL(10,2) NOT NULL,
    FK_FUNCIONARIOS_FUN_ID INTEGER
);

CREATE TABLE TIPOS_SERVICOS (
    SER_ID INTEGER AUTO_INCREMENT PRIMARY KEY UNIQUE,
    SER_NOME VARCHAR(80) NOT NULL
);

```

Após a execução do banco de dados, é possível conectá-lo ao código de programação e manipulá-lo por meio da interface criada. É nessa etapa do projeto gerações que o sistema passa a ser mais integrado e completo.

3 Conclusões e Recomendações

Com o aumento da população idosa não só no município de São João da Boa Vista, como em todo o país, é necessário atentarmos-nos à saúde e qualidade de vida dessa geração. Para isso, existem e são de extrema importância as Instituições de Longa Permanência. Nessa dissertação foi apresentado o projeto ‘Gerações’, que visa auxiliar a administração dessas instituições por meio de um sistema organizado e de fácil acesso; também foi ilustrado e explicado o banco de dados do módulo administrativo do projeto, que carrega uma parcela fundamental do sistema e é importante para as instituições.

O objetivo geral dessa pesquisa foi mostrar não só a importância, como também a necessidade do uso de um banco de dados para armazenar todas as informações das casas de longa permanência, principalmente ao que se refere aos gastos e receitas do local. Para o desenvolvimento desse banco, foi necessária a conclusão de 5 etapas principais:

A primeira delas envolveu o levantamento de requisitos, um passo de extrema importância pois foi a partir dele que identificamos as principais necessidades do sistema. Para a quantificação e especificação desses requisitos, foi necessário que os integrantes de cada módulo se reunissem e interpretassem a função do mesmo. A partir de diversas discussões, cada requisito foi qualificado como funcional ou não, e foi descrito cada aspecto necessário para que ele fosse cumprido. Para o módulo de controle administrativo, foram levantados ao todo 27 requisitos, sendo 11 deles requisitos funcionais e 16, requisitos não funcionais. Também foram descritas as funcionalidades e informações que deveriam ser apresentadas junto a cada função deles.

Para a criação do caso de uso do módulo administrativo, a segunda etapa, foram identificados três atores principais: o gestor, o funcionário de limpeza e o representante da equipe de limpeza; eles foram escolhidos para atender de forma organizada e correta as restrições de acesso do sistema. Os requisitos foram explorados mais a fundo e deles foram extraídas as sequências do fluxo do sistema, demonstrando como cada ator interage com ele e quais são as consequências de cada ação. Ao todo foram identificados 26 casos de uso, distribuídos entre funções como criar, editar, excluir, listar e pesquisar. O diagrama de casos de uso serviu de orientação para a criação do banco de dados e desenvolvimento de todo o projeto.

Finalizados os documentos gerados por cada etapa, foi necessário o uso da ferramenta brModelo, apresentada no tópico 2.2.3 desta pesquisa. Ela serviu para a criação, organização e integração dos modelos que foram desenvolvidos no módulo de controle administrativo.

Com o brModelo se deu início à terceira etapa, a criação do modelo entidade-relacionamento. Ele foi fundamental para o desenvolvimento de todas as etapas posteriores referentes ao banco de dados. Os DBAs identificaram quais seriam as entidades essenciais para o melhor desempenho do banco de dados, considerando as diversas funções do módulo. Com as entidades definidas, os atributos necessários foram associados a elas, considerando todas as informações que cada uma deveria possuir. Os relacionamentos e cardinalidades foram decididos de acordo com os requisitos funcionais, visando o cumprimento de todas as exigências.

A quarta etapa envolveu a criação do modelo relacional, gerado e adaptado a partir do MER, incluindo outras informações essenciais para DBAs e desenvolvedores, como a união de entidades e classificação dos tipos de atributo (texto, número, lógico e etc).

Para finalizar, a quinta e última etapa foi a criação do modelo físico do banco de dados. Após o esboço e criação dos modelos entidade-relacionamento e relacional, foi gerado o script do banco com todas as informações contidas neles, utilizando o brModelo. Foi a partir do script que o banco pôde ser executado e preenchido. Nele, os atributos são modelados com um nome e o seu tipo, permitindo que sejam manipulados. A população de dados fez parte desse processo pois, para a realização de testes e melhorias, eram necessárias informações registradas no banco; essas informações permitiram aos DBAs verificarem se as relações entre as entidades haviam sido feitas de maneira correta e se não houve conflito nos tipos de atributos.

Foi possível observar ao longo dessa pesquisa a importância de possuir um sistema organizado, considerando os conceitos presentes relacionados ao banco de dados e os resultados obtidos durante o seu processo de desenvolvimento. O volume de dados envolvidos na administração de uma instituição é grande e demanda organização e facilidade de acesso. Por meio do banco de dados em um sistema, é possível manter todas as informações dos funcionários, pacientes, responsáveis e terceiros ao alcance dos gestores das instituições, tornando práticas possíveis alterações ou inserção de novos dados, assim como a visualização deles.

O desenvolvimento do projeto ‘Gerações’ e do banco de dados do módulo de Controle Administrativo envolveu diversas dificuldades, tanto técnicas quanto na compreensão de alguns objetivos gerais. Algumas ferramentas apresentaram erros, como a ferramenta de armazenagem de arquivos SVN (*Apache Subversion*), que ficou indisponível em alguns momentos devido ao grande volume de dados compartilhados. O brModelo também possui falhas que impossibilitaram momentaneamente que os arquivos criados fossem salvos. Ademais, apesar de ter sido cuidadosamente pensado e planejado, ainda há melhorias que podem ser feitas em projetos futuros. O planejamento e ordem das atividades pode ser repensado, a fim de melhorar o progresso.

A criação do sistema ‘Gerações’, assim como essa pesquisa, foram experiências ricas em conhecimento e permitiram a aprendizagem de novos conceitos e técnicas de desenvolvimento do banco de dados.

4 Referências Bibliográficas

- [1] IBGE - Instituto Brasileiro de Geografia e Estatística. **São João da Boa Vista**, 2018. Disponível em: <<https://cidades.ibge.gov.br/brasil/sp/sao-joao-da-boa-vista/panorama>>. Acesso em: 16 de agosto de 2019.
- [2] PRADO, Ana Laura. **As 40 melhores pequenas cidades para envelhecer**, 2017. Disponível em: <<https://exame.abril.com.br/brasil/as-40-melhores-pequenas-cidades-para-envelhecer/>>. Acesso em: 16 de agosto de 2019.
- [3] CAMARANO, Ana Amélia; KANSO, Solange; **As instituições de longa permanência para idosos no Brasil**, 2010. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-30982010000100014> Acesso em 19 de outubro de 2019.
- [4] IFSP – Campus São João da Boa Vista. **Estude aqui**, 2017. Disponível em: <<https://www.sbv.ifsp.edu.br/estude-aqui>>. Acesso em: 16 de agosto de 2019.
- [5] FILHO, Wilson de Pádua Paula. **Engenharia de Software: fundamentos, métodos e padrões**, 2000. Disponível em: <http://aulasprof.6te.net/Arquivos_Aulas/07-Proces_Desen_Soft/Livro_Eng_Soft_Fund_Met_Padrees.pdf> Acesso em 27 de setembro de 2019.
- [6] CRUZ, Lucas. **Diferença entre Dados e Informação**, 2017. Disponível em: <<https://expertdigital.net/diferenca-entre-dados-e-informacao/>>. Acesso em 23 de setembro de 2019
- [7] ELIAS, Diego. **Dados VS Informação: Qual a diferença?** Disponível em: <<https://www.binapratice.com.br/dados-x-informacao>>. Acesso em 23 de setembro de 2019
- [8] HEUSER, Carlos Alberto. **Projeto de banco de dados**, p21-23. 6. ed. Porto Alegre: Bookman, 2009.
- [9] DATE, C J., tradução por Daniel Vieira. **Introdução a sistemas de banco de dados**, p6-12. 8. ed. Rio de Janeiro: Elsevier, 2003 – 19ª reimpressão.