



CENTRO DE ESTUDOS SUPERIORES DE ITAITUBA - LTDA
FACULDADE DE ITAITUBA - FAI
CURSO DE PÓS GRADUAÇÃO ESPECIALIZAÇÃO LATO SENSU EM DOCÊNCIA
PARA O MAGISTÉRIO SUPERIOR

ADRIANA DE SOUZA COIMBRA SANTOS

SOFTWARE PARA A IGREJA EVANGÉLICA
ASSEMBLEIA DE DEUS EM ITAITUBA: Resultados Práticos da
Implantação do Sistema SIEADI

Itaituba-Pa

2014

ADRIANA DE SOUZA COIMBRA SANTOS

**SOFTWARE PARA A IGREJA EVANGÉLICA
ASSEMBLEIA DE DEUS EM ITAITUBA: Resultados Práticos da
Implantação do Sistema SIEADI**

Projeto apresentado à Faculdade de Itaituba para obtenção da nota no Curso de Pós Graduação Especialização Lato Sensu em Docência Para o Magistério Superio.

Orientador: Dr^a Djalmira de Sá Almeida Barros.

Itaituba-Pa

2014

Dados Internacionais de Catalogação na Publicação (CIP)

- S237s Santos, Adriana de Souza Coimbra
 Software para a igreja evangélica Assembleia de Deus em Itaituba. / Adriana de Souza
 Coimbra Santos, Tiales Aguiar Ferreira. 2014.
 100 f. il; color, 31 cm.
 TAC (Graduação) – Instituto Federal do Pará. Campus Itaituba, Coordenação de
 Informática – Curso de Tecnologia em análises e desenvolvimento de sistemas, Itaituba,
 2014.
 Orientadora Prof^a. Dr^a. Djalmira de Sá Almeida Barros.
 Inclui bibliografia.
1. Desenvolvimento de Software. 2. Banco de Dados. 3. Engenharia de Software. I. Barros,
Djalmira de Sá Almeida, orientadora. II. Ferreira, Tiales Aguiar. III. Título
- CDD:005.18115**

ADRIANA DE SOUZA COIMBRA SANTOS

SOFTWARE PARA A IGREJA EVANGÉLICA
ASSEMBLEIA DE DEUS EM ITAITUBA: Resultados Práticos da
Implantação do Sistema SIEADI

Projeto apresentado à Faculdade de Itaituba para obtenção da nota no Curso de Pós Graduação Especialização Lato Sensu em Docência Para o Magistério Superio.

Orientador: Dr^a Djalмира de Sá Almeida Barros.

BANCA EXAMINADORA

Presidente: _____
Prof^a Dr^a Djalмира de Sá Almeida Barros

Avaliador: _____
Prof. Doutor Francisco Cláudio Sousa

Avaliador: _____
Prof. Me Elines dos Santos Batista

Resultado: APROVADO

A Deus por nos conceder entendimento, e estar conosco em todos os momentos por meio da fé para atingirmos esse sonho. A todos os nossos familiares que direta e indiretamente nos ajudaram nessa conquista na conclusão de mais uma etapa de nossa formação profissional.

AGRADECIMENTOS

A nossa orientadora. Prof^a Dr^a Djalмира de Sá Almeida Barros; Aos irmãos em Cristo Jesus que muito oraram por nós em intercessão diante de Deus para que tivéssemos força nessa árdua caminhada; Ao nosso querido Pastor Nivaldo dos Anjos, Presidente da Igreja Evangélica Assembleia de Deus em Itaituba; Pastor Altair Galvão pelo apoio e orações. Tudo é possível aquele que crer.

RESUMO

Este Projeto trate-se do Desenvolvimento de um *Software* para a Igreja Assembléia de Deus em Itaituba – SIEADI, vinculada ao curso de Análise e Desenvolvimento de Sistemas nas disciplinas de “Engenharia de Software” e “Banco de Dados”. Onde foi detectado que essa Igreja Evangélica não possuía um *Software* em seu âmbito administrativo, todo processo ocorrido dentro da instituição era feito de forma manual. O *software* para a Assembleia de Deus irá dispor de funcionalidades para operações realizadas pelo administrador do sistema. Entre as questões que norteiam este trabalho destacam-se: O que é uma Igreja Evangélica? Como fazer um software para suprir as necessidades da instituição? Em que medida o programa pode ser um instrumento de auxílio dentro da empresa? O Sistema tem como objetivo: realizar o controle administrativo da entidade. Além desse objetivo geral, foram traçados alguns objetivos específicos: cadastrar os Membros da Igreja através de um *software* no âmbito administrativo. Foi feita uma pesquisa teórica onde são abordadas as principais dificuldades no processo administrativo na Igreja Evangélica Assembleia de Deus em Itaituba, e uma pesquisa de campo nas dependências da Igreja para verificar como é realizado o controle dos fiéis da entidade. A metodologia envolveu um estudo de caso na Igreja e um questionário com questões abertas em que os entrevistados expuseram suas opiniões. Foi aplicada uma pesquisa de abordagem qualitativa para entender os fenômenos ocorridos dentro do ambiente organizacional, levando em consideração as opiniões dos participantes deste ambiente para a qualidade do cadastro informatizado e eficiente controle da membresia.

Palavras - chave: Análise e Desenvolvimento de Software. Banco de Dados. Engenharia de Software.

ABSTRACT

Treat yourself to a Proposed Development of a Software for the Assembly of God Church in Itaituba - SIEADI, linked to the course of analysis and systems development in the disciplines of " Software Engineering " and " Database " . This Evangelical Church did not have software on your administrative level, every process occurring within the institution was done manually. The software for the Assembly of God will have features that correspond to the operations performed by the system administrator, where he can update the software and register of secretaries to access the system , the operations performed by the secretaries will be permitted from the register held by the system administrator . Among the questions that guide this work are: What is an Evangelical? How to make a software to meet the needs of the institution? The extent to which the program can be an instrument of aid within the company? The system aims to: perform administrative control of the entity. Beyond this general objective, specific objectives were set: registering Church Members and develop software at the administrative level of the church member to register . A theoretical research which addresses the main difficulties in the administrative proceedings in the Assembly of God Evangelical Church in Itaituba was taken and performed a field survey on the premises of the Church as it is conducted to verify the users' control. The methodology involved a case study in the Church and a questionnaire with open questions to which respondents expressed their opinions . A qualitative approach was applied to understand the phenomena occurring within the organizational environment, taking into account the views of participants in this environment for quality control and efficient computerized register of members .

Keywords: Analysis and Development of a Software. Software Engineering. Database.

LISTA DE FIGURAS

Figura 1	- Modelo ER – Entidade Relacionamento.	26
Figura 2	- Tela de Splash	39
Figura 3	- Tela Principal	40
Figura 4	- Tela de Ajuda	41
Figura 5	- Formulário Cadastro de Membros.	41
Figura 6	- Formulário Cadastro de Congregações	42
Figura 7	- Formulário para Consulta de Congregações.	44
Figura 8	- Tela para Consulta de Membros	44
Figura 9	- Foto Prateleira de arquivos	52

LISTA DE SIGLAS

CEP	- Código de Endereçamento Postal	36
CPF	- Cadastro de Pessoa Física	42
DIPJ	- Declaração Imposto de Renda Pessoa Jurídica da Secretaria da Receita Federal.	30
ER	- Entidade Relacionamento.	25
ERP	- Enterprise Resource Planning.	33
IBGE	- Instituto Brasileiro de Geografia e Estatística.	29
RG	- Registro Geral	42
SGBD	- Sistemas Gerenciamento de Bancos de Dados.	24
SIEADI	- Software Informatizado para a Igreja Evangélica Assembleia de Deus em Itaituba.	12
SIGI	- Sistema Informatizado de Gerenciamento de Igrejas.	30
SPT	- Sistema de processamento de transações.	22
UF	- Unidade Federativa	42
TI	- Tecnologia da Informação	47

SUMÁRIO

INTRODUÇÃO.....	13
1 SISTEMA INFORMATIZADO	15
1.1 CONCEITO.....	15
1.2 PROCESSOS DE SOFTWARE	17
1.2.1 Modelo de Processo de Software	17
1.2.2 Projeto de Arquitetura	18
1.3 ENGENHARIA DE REQUISITOS	20
1.3.1 Requisitos funcionais e não funcionais	20
1.4 CONTROLE FINANCEIRO	21
1.4.1 Função Financeira	21
1.5 BANCO DE DADOS	23
1.5.1 Introdução a Banco de Dados	23
1.5.2 Modelo de Dados	25
1.5.3 Modelo Entidade-Relacionamento	26
2 PESQUISAS E TRABALHOS JÁ EXISTENTES	28
2.1 O QUE É UMA IGREJA EVANGÉLICA?	28
2.1.1 A Organização da Igreja Primitiva	28
2.2 DESCRIÇÃO DO RAMO DE NEGÓCIO DA EMPRESA	30
2.3 SOFTWARES EXISTENTES PARA GESTÃO DE IGREJAS	31
2.4 SOFTWARE IMPLANTADO NA IGREJA ASSEMBLEIA DE DEUS	34
2.5 ANÁLISE DO PROGRAMA IMPLANTADO: VANTAGENS	35
3 DESENVOLVIMENTO DA APLICAÇÃO	36
3.1 TECNOLOGIAS, FERRAMENTAS E LINGUAGENS UTILIZADAS NO DESENVOLVIMENTO DO SIEADI	36
3.2 INTERFACES DO SISTEMA	40
3.3 IMPLANTAÇÃO DO SIEADI: ANTES E DEPOIS	45
4. RESULTADOS PRÁTICOS DA IMPLANTAÇÃO DO SISTEMA SIEADI.....	47
4.1 A IMPORTANCIA DA TI NAS ORGANIZAÇÕES.....	47
4.2 ECONOMIA DE TEMPO.....	50
4.3 REDUÇÃO DE ESPAÇO.....	52
4.4 TOMADA DE DECISÃO.....	54
4.5 REDUÇÃO DE CUSTOS COM RECURSOS HUMANOS.....	56
BIBLIOGRAFIA.....	60
ANEXOS	65

INTRODUÇÃO

Esta é a descrição e Análise do Desenvolvimento de um Sistema informatizado para a Igreja Evangélica Assembleia de Deus em Itaituba: Resultados Práticos da Implantação do Sistema SIEADI.

O tema apresentado é vinculado ao Curso de Pós Graduação Especialização Lato Sensu em Docência Para o Magistério Superior

A Igreja Evangélica Assembleia de Deus em Itaituba, não possuía um *Software* em seu âmbito administrativo para cadastro de membros, todo processo ocorrido dentro da instituição para esse fim era feito de forma manual. A implantação de um programa de cadastro de membros visa fornecer ferramentas onde o usuário possa ter uma melhor gestão dos recursos humanos, e uma disponibilização rápida e fácil, com a possibilidade de cadastro dos membros e consulta a quantidade dos fies pertencentes a Igreja.

Questiona-se neste estudo: A Igreja Evangélica Assembleia de Deus em Itaituba possui um sistema Informatizado? Como criar um *software* que vai ser útil na administração da entidade? Quais as pesquisas e trabalhos já existentes na área? O custo do *software* impede o requisitante de viabilizar a instalação do mesmo? Dentro do ambiente investigado existe um *software* que supra as carências da Igreja?

O *Software* para a Igreja Evangélica Assembleia de Deus em Itaituba – SIEADI irá dispor de funcionalidades que correspondam às operações realizadas pelo administrador do sistema, onde ele poderá realizar atualizações do *software*, e cadastro de secretários para acesso ao sistema; as operações realizadas pelos secretários serão permitidas a partir do cadastro realizado pelo administrador do sistema.

Entre as questões que norteiam este trabalho destacam-se: O que é um banco de dados de uma Igreja Evangélica? Como fazer um *software* para suprir as necessidades da instituição? Em que medida o programa pode ser um instrumento de auxilio dentro da empresa?

O Sistema Informatizado para a Igreja Evangélica Assembleia de Deus em Itaituba – SIEADI tem como objetivo: realizar o controle administrativo da entidade. Além desse objetivo geral, foram traçados alguns objetivos específicos: Cadastrar os

Membros da Igreja; Verificar o Setor de Recursos Humanos; Propor a Contabilidade de membros; Desenvolver um *software* no âmbito administrativo da Igreja para cadastrar membros;

Para este trabalho foi feita uma pesquisa teórica onde são abordadas as principais dificuldades no processo administrativo da Igreja Evangélica Assembleia de Deus em Itaituba. Para sequência do estudo, foi realizada também uma pesquisa de campo nas dependências da Igreja com o objetivo de se verificar como é realizado o controle dos usuários.

Os procedimentos metodológicos são: uma pesquisa bibliográfica e uma pesquisa de campo aplicada na modalidade qualitativa, com apresentação e descrição das telas e códigos da fonte, para mostrar como proceder para explicar o sistema de informação do SIEADI e armazenamento dos seus dados.

O trabalho está organizado em quatro capítulos: o primeiro, explicando a base conceitual sobre o sistema informatizado; o segundo apresenta pesquisas e trabalhos já existentes na área de informatização de igrejas; o terceiro, mostra passo a passo o funcionamento do sistema SIEADI, bem como o ambiente da secretaria da igreja antes e depois da implantação do sistema, e o quarto aponta os resultados práticos da implantação do sistema, no que tange a tempo, espaço, tomadas de decisões, redução e custos com recursos humanos.

1 SISTEMA INFORMATIZADO

1.1 CONCEITOS

O mundo atual não poderia existir sem o *software*. Infraestruturas e serviços nacionais são controlados por sistemas computacionais, e a maioria dos produtos elétricos inclui um computador e um *software* que o controla. A manufatura e a distribuição industriais são totalmente informatizadas, assim como o sistema financeiro. A área de entretenimento, incluindo a indústria da música, jogos de computador, cinema e televisão fazem uso intensivo de *software*. Portanto, a Engenharia de *Software* é essencial para o funcionamento de sociedades nacionais e internacionais.

Segundo Sommerville (2011), os sistemas de *software* são abstratos e intangíveis. Eles não são restringidos pelas propriedades dos materiais, nem governados pelas leis da física ou pelos processos de manufatura. Isso simplifica a Engenharia de *Software*, porque não tem limites naturais para o potencial do *software*. Muitas pessoas pensam que *software* é simplesmente outra palavra para programa de computador. No entanto, quando falamos de Engenharia de *Software*, não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessárias para fazer esse programa operar corretamente.

Todos os sistemas que envolvem interação com bancos de dados compartilhados podem ser considerados sistemas de informação baseados em transações que permite acesso controlado a uma grande base de informações.

Um sistema de *software* desenvolvido profissionalmente é, com frequência, mais do que apenas um programa; ele consiste em uma série de programas separados e arquivos de configuração que são usados para configurar esses programas. Isso pode incluir documentação do sistema, que descreve a sua estrutura; documentação do usuário, que explica como usar o sistema, e sites para usuários baixarem a informação recente do produto. Inúmeras pessoas escrevem programas. Pessoas envolvidas com negócios escrevem programas em planilhas para exemplificar seu trabalho; cientistas e engenheiros escrevem programas para processar seus dados experimentais; e há aqueles que escrevem programas como

hobby, para seu próprio interesse e diversão. No entanto, a maior parte do desenvolvimento de *software* é uma atividade profissional, em que o *software* é desenvolvido para propósitos específicos de negócio, para inclusão em outros dispositivos ou como produtos de *software* como sistemas da informação, sistemas CAD etc. O *software* profissional, o que é usado por alguém além do seu desenvolvedor, é normalmente criado por equipes, em vez de indivíduos. Ele é mantido e alterado durante a vida.

Existem vários tipos de sistemas de *softwares*, desde os mais simples sistemas embutidos até os sistemas de informação complexos, de alcance mundial. Não faz sentido procurar notações, métodos ou técnicas universais para a engenharia de *software*, porque diferentes tipos de *softwares* exigem abordagens diferentes.

Desenvolver um sistema de informação corporativo é totalmente diferente de desenvolver um controlador para um instrumento científico. Nenhum desses sistemas tem muito em comum com um jogo computacional com gráficos intensos, porém todas essas aplicações precisam de Engenharia de *Software*, embora não necessitem da mesma técnica.

Para Humberto Martins Beneduzzi e João Ariberto Metz (Editora do Livro Técnico 2010), a demanda por *software* nunca foi tão grande quanto hoje e tende a crescer cada vez mais. Por isso, o desenvolvimento de *software* é uma das áreas que mais ganhou espaço no mercado, nos últimos tempos.

Conforme Gerson Rissetti e Sandra Puga (2009), um programa é um conjunto de instruções escrito em uma determinada linguagem que diz ao computador o que deve ser feito. Existem muitas formas e diferentes tipos de linguagens de programação, cada qual com uma finalidade especificada. Elas podem ser classificadas em níveis de dispositivos e lógica digital, chamada baixo nível, até o de linguagem orientada a problemas, chamado de alto nível.

Pressman (2006) destaca que a Engenharia de Software é uma tecnologia em camadas, que tem como base a qualidade, se alicerça em processo, dispõe de métodos que fornecem a técnica de “como fazer” e utiliza apoio automatizado de ferramentas.

Atualmente, por menor e mais simples que seja um Sistema de Informação, ele precisará ter a capacidade de armazenar e recuperar dados rapidamente; segundo Angelotti (2010), a área de Banco de Dados é de grande importância no mundo da Informática, uma vez que a informação é um bem precioso e deve ser armazenada de forma coerente e adequada. Quando você utiliza o Orkut, por exemplo, armazenam em um sistema de banco de dados suas informações pessoais, recados de amigos, depoimentos, etc. Se você sair da sua página no Orkut e entrar nela novamente, as informações ainda estarão lá. Isso ocorre porque essas informações foram armazenadas em um banco de dados, e, portanto, podem ser recuperadas no momento oportuno, quando o usuário solicitar.

1.2 PROCESSOS DE SOFTWARES

1.2.1 Modelos de Processos de *Softwares*

Um processo de software é um conjunto de atividades relacionadas que levam a produção de um produto de *software*. Essas atividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem de programação como Java ou C, no entanto, aplicações de negócios não são necessariamente desenvolvidas dessa forma. Atualmente novos *softwares* de negócios são desenvolvidos por meio da extensão e modificação de sistemas existentes ou por meio da configuração e integração de prateleira ou componentes do sistema.

Existem muitos processos de *software* diferentes, mais todos devem incluir quatro atividades fundamentais para a engenharia de *software*:

1. Especificação de Software. A funcionalidade do *software* e as restrições a seu funcionamento devem ser definidas.

2. Projeto e implementação de software. O *software* deve ser produzido para atender as especificações.

3. Validação de software. O *software* deve ser validado para garantir que atenda as demandas dos clientes.

4. Evolução de software. O *software* deve evoluir para atender as necessidades de mudança dos clientes.

Um modelo de processo de *software* é uma representação simplificada de um processo de *software*. Cada modelo representa uma perspectiva particular de um processo e, portanto, fornece informações parciais sobre ele.

Processos reais de *softwares* são intercalados com sequências de atividades técnicas, de colaboração e de gerência, com o intuito de especificar, projetar, implementar e testar um sistema de *software*. Os desenvolvedores de *softwares* usam uma variedade de diferentes ferramentas de *softwares* em seu trabalho. As ferramentas são especialmente úteis para apoiar a edição de diferentes tipos de documentos e para gerenciar o imenso volume de informações detalhadas que é gerado em um projeto de grande porte.

As quatro atividades básicas do processo - especificação, desenvolvimento, validação e evolução – são organizadas de forma diferente conforme o processo de desenvolvimento.

1.2.2 Projeto de Arquitetura

O projeto de arquitetura é um processo criativo no qual você projeta uma organização de sistemas para satisfazer aos requisitos funcionais e não funcionais de um sistema. Por ser um processo criativo, as atividades no âmbito do processo dependem do tipo de sistema a ser desenvolvido, a formação e experiência do arquiteto de sistemas e os requisitos específicos para o sistema. Por isso, é útil pensar em projeto de arquitetura como uma série de decisões, em vez de uma sequência de atividades.

Durante o processo de projetos de arquitetura, os arquitetos de sistemas precisam tomar uma série de decisões estruturais que afetam profundamente o sistema e o seu processo de desenvolvimentos. Com base em seus conhecimentos e experiências, eles precisam considerar as seguintes questões fundamentais sobre o sistema:

1. Existe uma arquitetura genérica de aplicação que pode atuar como um modelo para o sistema que está sendo projetado?
2. Como o sistema será distribuído por meio de um número de núcleos ou processadores?

3. Que padrões ou estilos de arquitetura podem ser usados?
4. Qual será a abordagem fundamental para se estruturar o sistema?
5. Como os componentes estruturais do sistema serão decompostos em subcomponentes?
6. Que estratégia será usada para controlar o funcionamento dos componentes do sistema?
7. Qual a melhor organização de arquiteturas para satisfazer os requisitos não funcionais do sistema?
8. Como o projeto de arquitetura será avaliado?
9. Como a arquitetura do sistema deve ser documentada?

A arquitetura de um sistema de *software* pode se basear em um determinado padrão ou estilo de arquitetura. Um padrão de arquitetura é uma descrição de uma organização do sistema (GARLAN E SHAW, 1993), como uma organização cliente-servidor ou uma arquitetura em camadas. Os padrões de arquitetura capturam a essência de uma arquitetura que tem sido usada em diferentes sistemas de *software*. Ao tomar decisões sobre a arquitetura de um sistema, você deve conhecer os padrões comuns, bem como saber onde eles podem ser usados e quais são seus pontos fortes e fracos.

A ideia de Garlan e Shaw, de um estilo de arquitetura (estilo e padrão passaram a significar a mesma coisa), abrange as questões 4 e 6 da lista anteriormente apresentada. Você precisa escolher a estrutura mais adequada, como cliente-servidor ou em camadas, que permitirá o cumprimento dos requisitos do sistema. Para decompor em subcomponentes. Devido à estreita relação entre os requisitos não funcionais e a arquitetura do *software*, o estilo e a estrutura da arquitetura particular que você escolhe para um sistema devem depender dos requisitos não funcionais do sistema, como: Desempenho, Proteção, Segurança, Disponibilidade e Manutenção.

Sistemas de aplicações são destinados a atender uma necessidade organizacional ou de negócio. Todas as empresas têm muito em comum – elas precisam contratar pessoas, emitir faturas, manter a contabilidade e assim por diante, em todos os sistemas que envolvem interação com bancos de dados compartilhados.

Para Pfleeger (2004), Programação envolve uma boa dose de criatividade, não importa qual linguagem seja utilizada, cada componente de programação envolve, pelo menos, três aspectos principais: estrutura de controle, algoritmos e estruturas de dados. O padrão mais importante é a necessidade de correspondência direta entre os componentes de projeto do programa e os de código do programa. Todo o processo de projeto é de pouco valor, se a modularidade não for trazida para o código. As características do projeto, tais como o pequeno grau de acoplamento, elevado grau de coesão e interfaces bem definidas, também deverão ser características do programa, de maneira a que algoritmos, funções, interfaces e estruturas de dados possam ser rastreados facilmente do projeto para o código, e vice-versa.

1.3 ENGENHARIA DE REQUISITOS

1.3.1 Requisitos Funcionais e Não Funcionais

Os requisitos de *software* são frequentemente classificados como funcionais – declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações, em alguns casos os requisitos funcionais podem explicar o que o sistema não deve fazer. E não funcionais – que são restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas por normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

Conforme Ramos (2000) existe relatos de que muitas vezes os sistemas de informação não atendem às necessidades dos usuários. Um problema que está muitas vezes relacionado com a adoção de aplicações que não satisfazem os requisitos para elas definidos. Além disso, os custos de desenvolvimento usualmente excedem os benefícios esperados com sua utilização, a partir da constatação de que a aplicação não atende às necessidades, expectativas e ou interesses das entidades (organizações e grupos) afetadas pela sua adoção.

Para Furnival (1995), os casos de projetos para sistemas de informação automatizados que fracassam são muitos, em geral resultando em não uso, sub uso ou até em sabotagem do sistema, pois esse não atinge os objetivos para os quais foi projetado, ou os faz de uma forma diferente da prevista.

De acordo com Resende (2005), os sistemas de informação, independentemente de seu nível ou classificação, objetivam auxiliar os processos de tomada de decisões nas organizações. Se os sistemas de informação não propuserem a atender a esse objetivo, sua existência não será significativa para a organização.

Segundo Vianna (2004), a tarefa de implantação é uma tarefa com um elevado nível de complexidade, e o seu sucesso depende fortemente dos requisitos do software. A postura mais frequente encontrada entre os analistas de sistemas é a de especialistas aptos a modelar as necessidades em termos de informação dos usuários de sistemas. O resultado do seu trabalho é inquestionável: corresponde ao atendimento das necessidades em termos de informação dos usuários. Falhas no levantamento dos requisitos dos sistemas fatalmente são atribuídas às dificuldades dos usuários em “saber o que necessitam” ou em “transmitir o que desejam”.

1.4 CONTROLE FINANCEIRO

1.4.1 Função Financeira

A função financeira está ligada às áreas que necessitam de tomadas de decisões na empresa. O papel do administrador financeiro vem sofrendo notáveis inovações. Antigamente, a função financeira compreendia apenas em gerir a manutenção de registros das operações, da preparação e emissão de relatórios para o controle das atividades já executadas, na administração de recursos de caixa e nos meios para obtenção de recursos para pagar as contas.

Atualmente, a função ultrapassou barreiras, evoluiu, a ponto de preocupar-se com os objetivos gerais da empresa. A preocupação na emissão e preparação de relatórios das atividades passadas, para avaliar o impacto futuro dos possíveis resultados das decisões tomadas em todas as áreas. As pessoas de todos os

departamentos da empresa – contabilidade, produção, vendas precisam interagir com o pessoal de finanças para realizar seus trabalhos.

Segundo Groppelli e Nikbakht (2002), “a meta principal de um administrador financeiro é maximizar a riqueza dos acionistas. A maximização dos lucros é uma meta de curto prazo que não se remete, ela própria, a alguns fatores-chave, um dos quais é um nível aceitável de risco”.

O administrador financeiro deve ter o cuidado de fazer sempre uma crítica de sua dinâmica de trabalho. O mesmo não pode se acomodar na simples posição de autorizador de gastos, assinar cheques e ir ao banco para cobrir rombos na conta.

Para o Prof. Administrador Sergio Junger (2012), Especialista em Gestão de Tecnologia da Informação e Mestrando em Administração Estratégica, os Sistemas de Informação tem como um dos grandes desafios assegurar agilidade da informação, a qualidade, que são mandatórios para seus gestores e as corporações. A informação é tudo na administração. Todos concordam e confirmam, entretanto, é unânime em dizer que um conjunto de características necessárias para que essa ferramenta de trabalho atenda realmente às necessidades dos gestores com agilidade – disponível no tempo certo e confiabilidade – coerente e correta. A computação corporativa tem uma linha evolutiva particular, tendo como uma de suas principais metas possibilitarem que a informação tenha esse conjunto de características, criando novos e melhores instrumentos de apoio à tomada de decisão.

O principal benefício proporcionado pela tecnologia aos Sistemas de Informação é a habilidade de processar um enorme volume de dados e informações, simultaneamente, tornando a disponibilização destas, praticamente imediata. Porém, toda essa potencialidade nada resolverá se os sistemas (rotinas, processos, métodos) não permanecerem muito bem analisados e coordenados. Informatizar com sistemas ruins gera novos problemas e nenhuma solução, além de encobrir as presumíveis causas dessas falhas. Tais ocorrências infelizmente são bastante comuns nas organizações, uma vez que não há entendimento quanto à análise de sistemas operacional-corporativos e programação desses sistemas.

A pertinência do desenvolvimento dos sistemas da organização e sua análise é uma competência inerente aos administradores, incumbindo à área técnica, tão somente a automatização/informatização dos sistemas. A eficiência dos sistemas não é mensurada pela informatização, mas pela eficácia e a qualidade dos métodos, certificando a informação almejada, confiável e no momento adequado.

Tanto o desenvolvimento dos Sistemas de Informação, quanto a sua análise, necessitam ser executados de forma independente, pois depois de desenvolvidos os processos basta transmiti-los à máquina. Essencialmente, deve ser considerada a importância do administrador nesse processo, que é fundamental para a organização. Mais do que um modismo, a tecnologia deve ser compreendida como um instrumento, metodologia garantindo qualidade, competitividade, redução de custos e principalmente, satisfazer os anseios e pretensões dos clientes, o exato motivo da existência das organizações.

No caso do controle financeiro da Igreja Assembleia de Deus, já existe um programa informatizado que é da competência do Administrador, tratando exclusivamente de receitas, despesas e capitação de recursos para manutenção dos serviços da Igreja. Por essa razão, neste estudo a meta é implantar um software apenas para cadastro e controle de usuários.

Uma das ramificações de sistemas de informação é o Sistema Procedimentos e Transações, este tem por objetivo realizar e gravar as transações de rotinas diárias necessárias para conduzir uma organização.

O sistema de informação torna os processos de uma empresa mais confiável e menos burocrático. De acordo com Laudon e Laudon (2004), "SPTs são os sistemas integrados básicos que atendem ao nível operacional da organização", os autores afirmam que um SPT "é um sistema computadorizado que realiza e registra as transações rotineiras necessárias ao funcionamento da empresa".

Conforme Stair e Reynolds (2006), "SPT é um conjunto organizado de pessoas, procedimentos, software, base de dados e dispositivos para registrar transações completas de negócios".

1.5 BANCO DE DADOS.

1.5.1 Introdução a Banco de Dados

Os Bancos de Dados e a sua tecnologia estão provocando um grande impacto no crescimento do uso de computadores. É inevitável afirmar que eles representam um papel crítico em quase todas as áreas em que os computadores são utilizados incluindo negócios, comércio eletrônico, engenharia, medicina, direito, educação, e as ciências da informação, para citar apenas algumas delas. A palavra banco de dados é tão comumente utilizada que, primeiro devemos defini-la. Nossa definição inicial é bastante genérica.

Para Elmasri e Navathe (2005), um banco de dados é uma coleção de dados relacionados. Os dados são fatos que podem ser gravados e que possuem um significado implícito. Um banco de dados pode ser de qualquer tamanho e de complexidade variável onde as principais características de um banco de dados são as seguintes:

- Natureza auto descritiva do sistema de banco de dados.
- Isolamento entre os programas e os dados, a abstração de dados.
- Suporte para múltiplas visões dos dados.
- Compartilhamento de dados e processamento de transações de multiusuários.

O que são dados? O que são informações?

Dados são arquivos que podem ser texto, imagens, vídeos e diversas outras coisas. Quando organizamos estes dados com um objetivo, eles passam a ser informações, pois servem a um propósito.

O termo 'banco de dados' foi definido por diversos autores:

Um banco de dados é uma coleção de dados operacionais armazenados usados pelos sistemas de uma determinada aplicação. (DATE, 2004).

Um banco de dados é uma coleção de dados que contém informação de um particular empreendimento. (KORTH, 2006).

Uma definição mais precisa de Banco de Dados não existe formalmente, contudo pode-se dizer que:

a) Um Banco de Dados é uma coleção logicamente coerente de dados com um determinado significado inerente. Isto significa que um conjunto aleatório de dados não pode ser considerado um Banco de Dados.

b) Um Banco de Dados representa aspectos de uma parte restrita do mundo real, denominado de minimundo. Alterações que ocorra no minimundo são refletidas no Banco de Dados.

c) Bancos de dados é um conjunto coerente e lógico de dados relacionados que possuem significância intrínseca. Esses dados representam aspectos do mundo real e devem ser mantidos para atender os requisitos da entidade em questão (OLIVEIRA,2002).

No caso de Igrejas evangélicas, assim como em qualquer empreendimento, Banco de Dados é o conjunto das informações inerentes à atividade e ações que elas executam, tanto no registro como no controle de pessoas, identificação, endereço, situação e participação.

1.5.2 Modelos de Dados.

Um modelo de dados é um conjunto de conceitos que podem ser usados para descrever a estrutura e as operações em um banco de dados ELMASRI e NAVATHE (1994). O modelo busca sistematizar o entendimento que é desenvolvido a respeito de objetos e fenômenos que serão representados em um sistema informatizado. Os objetos e fenômenos reais, no entanto, são complexos demais para permitir uma representação completa, considerando os recursos à disposição dos sistemas gerenciadores de bancos de dados (SGBD) atuais. Desta forma, é necessário construir uma abstração dos objetos e fenômenos do mundo real, de modo a obter uma forma de representação conveniente, embora simplificada, que seja adequada às finalidades das aplicações do banco de dados.

Elmasri e Navathe (2005) afirmam que uma característica fundamental do uso de bancos de dados é que permitem a abstração dos dados, ocultando detalhes do armazenamento de dados, que são desnecessários para a maioria dos usuários de bancos de dados. Um modelo de dados fornece o significado necessário para admitir essa abstração. Por estrutura de um banco de dados entendemos os tipos de

dados, relacionamentos e restrições que devem suportar os dados. A maioria dos modelos também inclui uma série de operações básicas para a recuperação e atualizações no banco de dados.

Além das operações básicas fornecidas pelo modelo de dados, esta se tornando comum incluir conceitos para especificar o aspecto dinâmico ou o comportamento de uma aplicação de banco de dados. Isso permite que o projetista do banco de dados especifique uma série de operações definidas pelos usuários, que são permitidas nos objetos do banco de dados.

Ainda com Elmasri e Navathe (2005), nas categorias de modelos de dados, os modelos de alto nível ou modelos de dados conceituais, possuem conceitos que descrevem os dados como os usuários os percebem, enquanto os de baixo nível, ou modelos de dados físicos, contem conceitos que descrevem os detalhes de como os dados estão armazenados no computador. Os conceitos provenientes dos modelos de dados de baixo nível geralmente são significativos para os especialistas em computadores, mas não o são para os usuários finais.

1.5.3 Modelo Entidade-Relacionamento

O modelo Entidade-Relacionamento (ER) Chen (1976) é um dos primeiros modelos de dados semânticos. Ele utiliza apenas três tipos construtores básicos: entidade (conjunto de entidades), relacionamento (conjunto de relacionamentos) e atributo.

Uma entidade é uma representação abstrata de um objeto do mundo real, que possui uma existência independente e sobre a qual se deseja guardar e recuperar informações. Pode ser algo concreto como uma pessoa ou abstrato como um cargo. Uma entidade que tem sua existência dependente de outra é chamada de entidade fraca.

Um relacionamento é uma associação entre duas ou mais entidades. No caso de relacionamentos binários, estes podem ter sua cardinalidade expressa por 1:1, 1:N, N:1 ou N:N, indicando o número de vezes que uma entidade pode participar do relacionamento. Um atributo é uma propriedade que descreve uma

entidade ou um relacionamento. Um atributo identificador identifica unicamente uma entidade, conforme mostrado na Figura 1:

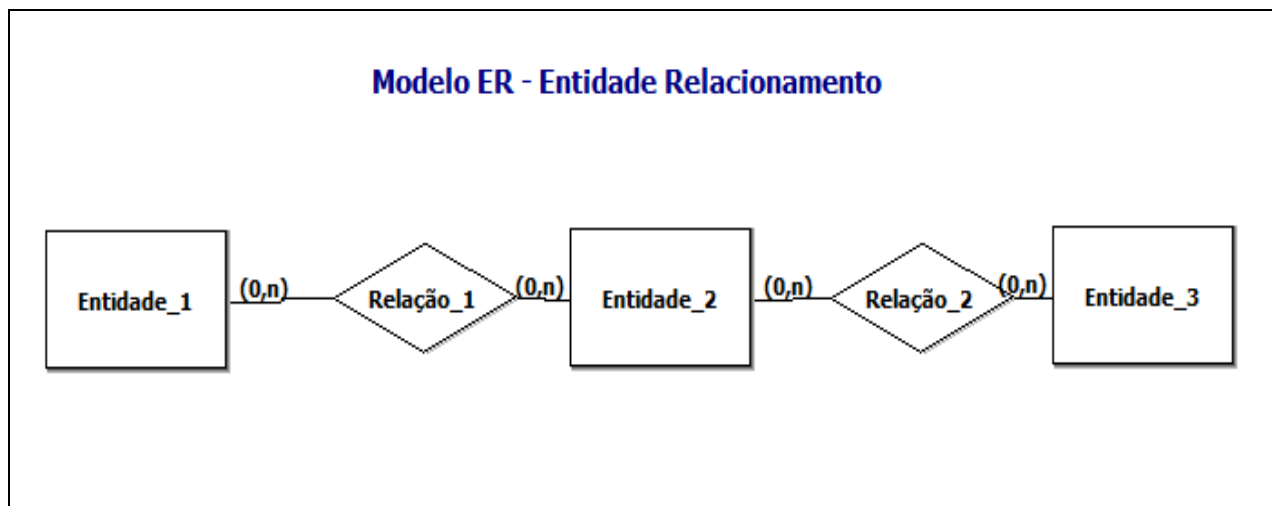


Figura 1: Modelo ER – Entidade Relacionamento.

Fonte: Dados da pesquisa, 2014.

2 PESQUISAS E TRABALHOS JÁ EXISTENTES

2.1 O QUE É UMA IGREJA EVANGÉLICA?

Dentre algumas definições de igreja, a que melhor define é a que afirma que se trata de um templo cristão, onde é um local de pregação dos ensinamentos de Jesus Cristo, que obedece os princípios da ética cristã, colocada também como um conjunto de fiéis unidos pela mesma fé que seguem as mesmas doutrinas religiosas.

No site Significados (2011), encontra-se a informação da Sociologia em que Igreja é um grupo religioso organizado e institucionalizado. É uma sociedade cujos membros representam da mesma maneira o mundo sagrado e suas relações com o mundo profano. Como grupo, uma igreja abrange uma comunidade dos que creem e, geralmente um corpo de sacerdotes, hierarquizado ou não. Como instituição, a igreja representa um sistema de preceitos dogmáticos, ritos e crenças.

Conforme o artigo da Comunidade Cristã Jesus para o Mundo (2005 p. 14), o cristianismo originou-se no mundo Mediterrâneo, sendo o maior e mais importante centro de civilização de então. Herdeiro que era da longa história judaica e tendo seu início nos anos de maior vigor do Império Romano, o Cristianismo gozava de todos os benefícios que o império oferecia aos seus cidadãos. O grande fundador da Igreja Primitiva foi Jesus Cristo, o Filho de Deus, enviado para a redenção do homem.

Então podemos entender que a igreja existe não só fisicamente, mas também espiritualmente, entre os cristãos, pois a igreja é local de oração e adoração do Senhor Jesus Cristo.

2.1.1 A Organização da Igreja Primitiva.

A igreja existe em dois níveis. Um deles é um organismo eterno, invisível, bíblico, que é consolidado em um corpo pelo Espírito Santo. O outro nível é o da organização temporal, histórica, visível, humana. O primeiro é o fim, o segundo os meios. O desenvolvimento da Igreja como uma organização foi iniciado pelos apóstolos sob a direção do Espírito Santo. Todo corpo organizado precisa da divisão

de funções e a especialização da liderança para que possa funcionar eficientemente. Uma liturgia para conduzir o culto da igreja de modo ordenado é outra consequência lógica do crescimento da igreja como organização. O objetivo final da igreja como um organismo que cultua é a conquista da qualidade de vida. (COMUNIDADE CRISTÃ JESUS PARA O MUNDO, 2005, p. 20).

Uma descoberta revolucionária: No início do ano 1512, enquanto lia a Epístola de Paulo aos Romanos, Lutero encontrou a declaração revolucionária: “*O justo viverá pela fé*” (Rm 1.17). Estas palavras incendiaram-lhe a mente com vislumbres da verdade que procurava há tanto tempo. Descobriu que a salvação lhe pertencia, simples e unicamente por fé na obra que Cristo consumou na Cruz. E foi com o intuito de melhor compreender esta verdade, que empreendeu cuidadoso estudo das Escrituras, e das obras de grandes mestres cristãos, como Agostinho e Anselmo. Lutero tinha convicção da verdade que descobriu nas Escrituras e da magnitude das decisões que tomaria a partir daí na defesa da mesma. Essa experiência trouxe novo impulso à sua vida religiosa- impulso necessário ao seu trabalho de reformar a igreja que havia sido paganizada. (COMUNIDADE CRISTÃ JESUS PARA O MUNDO, 2005, P. 20). A partir de então começaram-se as reformas protestantes em todo o mundo dando início a criação do protestantismo.

No site Brasil Escola (2014) Protestantismo é, ao lado do Catolicismo, um dos grandes ramos do Cristianismo. O nome “protestante” provém dos protestos dos cristãos do século XVI contra as práticas da Igreja Católica, já citados anteriormente. No Brasil, o termo “protestante” é substituído por “evangélico”, retirando a conotação polêmica da palavra e dando uma característica mais positiva e universal.

O movimento protestante surgiu na tentativa de Reforma da Igreja Católica iniciada pelo monge agostiniano Martinho Lutero, no século XVI. Os motivos para esse rompimento incluíram principalmente as práticas ilegítimas da Igreja Católica, além da divergência em relação a outros princípios católicos, como a adoração de imagens, o celibato, as missas em latim, a autoridade do Papa.

Para o protestantismo, a salvação é dada através da graça e bondade de Deus, na qual cada pessoa pode se relacionar diretamente com seu Criador, sem a necessidade de um intermediário.

Os protestantes defendem a crença de que a única autoridade a ser seguida é a Palavra de Deus, presente na Bíblia Sagrada. Desta forma, através da ação do Espírito Santo, os cristãos, ao lerem a Bíblia, têm uma maior harmonia com Deus. Por esse motivo, a partir da Reforma Protestante, a Bíblia foi traduzida para diversas línguas e distribuída sem restrições para as pessoas.

O protestantismo pode ser subdividido em ramos, como o luteranismo, calvinismo, anglicanismo, etc. Atualmente, costuma-se classificar as igrejas protestantes em pentecostais e neopentecostais.

No Brasil, o protestantismo foi trazido pelos holandeses entre os anos de 1624 e 1625, tendo sido propagado principalmente entre os índios. Pesquisas recentes mostram o crescimento do protestantismo no Brasil: em 1970, o Censo do IBGE registrava cerca de 4,8 milhões de evangélicos; em 1980, 7,9 milhões; em 1991, 13,7 milhões; em 2000, 26,1 milhões.

2.2 DESCRIÇÃO DO RAMO DE NEGÓCIO DA IGREJA

No mundo, as Assembleias surgiram quase simultaneamente nos Estados Unidos (1914) e no Brasil (1911), na segunda década no século XX como Igreja Pentecostal.

A Assembleia de Deus foi fundada em 18 de junho de 1911, em Belém (PA), pelos missionários suecos Daniel Berg e Gunnar Vingren. É a primeira igreja pentecostal do país, ou seja, que acredita nas manifestações atuais do Espírito Santo tal qual acontecia nos dias dos apóstolos. A obra iniciada em Belém se espalhou para todos os estados do Brasil e diversos países do mundo. Após 98 anos de atividades, a Assembleia de Deus é hoje a maior igreja evangélica do Brasil, com cerca de 20 milhões de membros.

A Assembleia de Deus possui hoje em Belém 100 mil membros, mais de 400 templos e quase 700 pastores, além de manter 31 missionários em outros países. A igreja tem sua missão claramente definida: “Cheios do Espírito Santo, cumprimos nossa missão bíblica de Adoração a Deus, Comunhão com os irmãos, Evangelização dos povos, Discipulado dos salvos, Vigilância e Oração até Jesus voltar”.

Desde 1997 a Assembleia de Deus em Belém tem como líder o Pastor Samuel Câmara, que preside a diretoria da igreja. A organização dos trabalhos em Belém é feita através de 34 coordenações, que abrangem todos os templos da cidade. A igreja possui Conselho Fiscal constituído, bem como assessorias nas áreas jurídica, administrativa, social, eventos e comunicação. A Assembleia de Deus também desenvolve programas sociais como forma de demonstrar o amor de Deus para todos os homens. A igreja conta ainda com a Rede Boas Novas de Comunicação para divulgar a mensagem do evangelho através dos meios de comunicação de massa, cumprindo a Grande Comissão deixada por Cristo à sua Igreja.

A Igreja Evangélica Assembleia de Deus em Itaituba-PA, está localizada à Avenida Antônio de Pádua Gomes, nº 335, Bairro Bela Vista, CEP: 68180-120.

Ramo de atividade ou ramo de negócio nada mais é do que, a área de mercado em que a empresa irá ou pretende atuar. Para se estabelecerem de qual ramo de negócio as empresas fazem parte, deve-se antes analisá-las e, a partir de uma visão abrangente. Então dividi-la em setores menores, a fim de que se defina a orientação da empresa em relação ao seu consumidor, pois a partir daí se definirá o ramo de negócio da empresa.

O ramo de negócio em que se enquadram as igrejas são os de templos e instituições religiosas, o qual no Brasil as igrejas evangélicas vêm crescendo substancialmente.

O número de evangélicos no Brasil aumentou 61,45% em 10 anos, segundo dados do Censo Demográfico divulgado pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Em 2000, cerca de 26,2 milhões se disseram evangélicos, ou 15,4% da população. Em 2010, eles passaram a ser 42,3 milhões, ou 22,2% dos brasileiros. Em 1991, o percentual de evangélicos era de 9% e, em 1980, de 6,6%. Mesmo com o crescimento de evangélicos, o país ainda segue com maioria católica. Segundo o IBGE, o número de católicos foi de 123,3 milhões em 2010, cerca de 64,6% da população. No levantamento feito em 2000, eles eram 124,9 milhões, ou 73,6% dos brasileiros. A queda foi de 1,3%. A queda do percentual de católicos é histórica, de acordo com o instituto. Até 1970, em quase 100 anos, a queda foi de 7,9 pontos percentuais: o número de católicos em 1872 (ano do primeiro Censo) representava 99,7% da população e passou a 91,8% em 1970. O Nordeste ainda mantém o maior percentual de católicos, com 72,2% em 2010. Apesar de ser a região do país com maior concentração do grupo religioso, a população nordestina católica sofreu queda. Em 2000, o percentual era de 79,9%. No Sul, o IBGE também identificou redução do percentual de católicos, saindo de 77,4% para 70,1% nos censos de 2000 e de

2010, respectivamente. A maior redução foi registrada pelo instituto no Norte, passando de 71,3% da população em 2000 para 60,6% em 2010. O IBGE registrou que, ao mesmo tempo em que o número de católicos caiu no Norte e no Nordeste, o número de evangélicos cresceu com maior volume nas duas regiões. A representatividade no Norte saiu de 19,8% (2000) para 28,5% (2010). No Nordeste, o aumento de evangélicos foi menor, saindo de 10,3% para 16,4%, se comparados os Censos de 2000 e de 2010, respectivamente. (G1, 2012).

Como podemos ver, há uma grande oportunidade de crescimento para este tipo de empreendimento no Brasil e principalmente em nossa região, pois como vimos os números de evangélicos só tende a crescer e a região norte e a quem vem apresentando o maior crescimento desta religião nos últimos 10 anos, por este motivo é de grande importância que este empreendimento esteja preparado para receber esse grande número de fiéis.

2.3 SOFTWARES EXISTENTES PARA GESTÃO DE IGREJAS

No mercado de *software* devido à grande facilidade de troca de informação através da internet, encontram-se no mercado, vários *softwares* para igrejas, tanto pagos como gratuitos, que atendem necessidades básicas e específicas das igrejas interessadas neste produto, os quais estarão sendo demonstrados logo abaixo alguns do encontrados no mercado com nomes e especificações.

O SIGI é um exemplo de administração de igrejas evangélicas, versão 6.8. É um programa para administração de igrejas evangélicas tradicionais, controle de financeiro, secretaria e contabilidade.

O SIGI - Sistema Informatizado de Gerenciamento de Igrejas Evangélicas é um *software* para administração de igrejas evangélicas que possui cadastro para ficha de membros, aniversariantes, cargos ministeriais, controle por situação do membro, impressão de credenciais, diversos modelos de cartas e etiquetas. Também faz toda a Contabilidade e está em acordo com a legislação fiscal, permitindo a exportação de arquivo para DIPJ - Declaração de Imposto de Renda Pessoa Jurídica da Secretaria da Receita Federal.

O programa controla agenda de compromissos da igreja, pastores, obreiros e missionários, além de contar com vários gráficos. Com ele, você pode cadastra

congregações, escola dominical, acervo, lançamentos de caixa, bancos, fornecedores, contas a pagar, orçamento, dízimos, ofertas, razão, balancete, diário, ficha financeira de dízimistas e patrimônio.

Funcionalidades: Cadastro de membros; Credencial de membros; Modelos de atas; Agendas; Crianças; Missionários; Visitantes; Emissão de cartas; Acervo; Certificados de batismo; Congregações; Escola bíblica dominical; Cargo ministerial; Funções que exerce; Lançamentos de caixa; Bancos; Dízimos e ofertas; Fornecedores; Contas a pagar; Orçamento; Contabilidade; Patrimônio; Mala direta; Etiquetas; Gráficos; Cartas; Centro de custos. Roda em 95/98/NT/ME/XP/Vista/7.

O *Software Ecclesia Soft 2012* é outro programa desenvolvido para responder a exigência da maioria das Igrejas Evangélicas no Brasil, principalmente nos setores de registro e controle de dados. No estudo em questão, esses setores envolvem a administração, a secretaria, a tesouraria e a Escola Dominical em Igrejas Evangélicas.

Secretaria: Registo de Membros, congregado e crianças; Consulta de Membros, congregados e crianças; Gestão de cargos eclesiásticos; Atribuição de cargos; Agenda; Listagem de Aniversariantes; Listagem de Obreiros em serviço; Cartão de Membro; Carta de Mudança; Carta de Recomendação; Certificado de Batismo; Certificado de Dedicção.

Tesouraria: Lançamento de Entradas; Lançamento de Saídas; Gestão de categorias das entradas e saídas; Mapas contábilísticos; Folha de cálculo; Resumo dos lançamentos; Impressão de recibos; Análise de Caixa (gráfico).

Escola dominical: Inscrição e anulação de matrículas; Gestão de turmas de aula; Gestão de períodos de aulas; Gestão de material de cada turma; Aulas; Presenças e materiais presentes; Relatório geral; Gráfico geral e por turmas.

Administração: Gestão dos utilizadores e acessos ao programa; Gestão das caixas da tesouraria; Licenciamento.

O *Software Church 12* é também uma ferramenta de gerenciamento eclesiástico que permite gerenciar e aumentar a produtividade de sua Igreja. Por ser um sistema *client/server*, pode ser executado diretamente no seu navegador, permitindo dessa forma, flexibilidade, mobilidade. O *Church* suporta os principais Sistemas Operacionais do mercado.

Há outros cenários onde o Sistema pode ser instalado e/ou utilizado. Rede Local (*Lan*) - o sistema pode ser instalado unicamente em um servidor e a partir dele rodar nos computadores ligados a este servidor formando assim uma local área network (rede local). Sem a necessidade de instalar o sistema em cada estação de trabalho, basta ser instalado unicamente no servidor da rede que todos os computadores ligados a rede terão acesso ao sistema.

Internet (*Wan*), o sistema roda 100% na internet e pode ser rapidamente colocado na web com total segurança e confiabilidade dos dados.

Intranet e *Extranet* é possível combinar os dois tipos de acesso sem problema, ou seja, o cliente poderá acessar o sistema de uma rede local e facilmente portar seu acesso para internet, colocando-o acessível aos usuários internos da sua organização apenas através do seu portal ou site. Assim sendo, se eles estiverem na rede local acessarão pela mesma, se estiverem fora da rede, acessarão pela internet, mantendo total sigilo e segurança.

Secretaria: Cadastros Principais; Cadastros Secundários; Tabelas Principais; Tabelas Secundárias; Relatórios Principais; Relatórios Parte 1 e Parte 2; Gráficos.

Cadastro de Membros: Com o cadastro de membros tenha controle total sobre sua membresia e facilite o trabalho da secretaria. Além de ser completo, limpo e de fácil navegação, permite fazer buscas com várias opções de filtros de pesquisa.

Reuniões: Reuniões com atas e controle de frequência: Controle de Reuniões integrado com presenças (via cartão com código de barras), Geração da Ata automaticamente e inclusão de definições oriundos das reuniões das filiais. Ao Final da reunião a ata é gerada automaticamente para ser impressa e assinada pelos presentes. Permanecendo disponível para consultas no sistema e reabertura sempre que for necessário. Reuniões possui Pautas e Presenças: Controle de Reuniões possui pautas que podem ser ligadas à determinado tipo de reunião, bem como o controle de presenças integrado ao cadastro da reunião. Os participantes vão chegando e marcando sua presença, o sistema por sua vez atualiza dinamicamente a lista dos presentes na mesma janela de controle da reunião conforme imagem abaixo. Reuniões possuem Reabertura: Controle de Reuniões possui também mecanismo para reabrir uma reunião/ata sempre que for necessário. Para isso basta selecionar a Ata e clicar no botão correspondente que a mesma é

reativada. Emissão de Credencial: As credenciais podem ser editadas e geradas diretamente no sistema, através do complemento, portanto sem o uso de aplicação desktop. Tudo web.

Financeiro: Patrimônio, Folha de Salários de Funcionários;

Recursos Humanos: Cadastro de Funcionários, Cadastro de Pastores;

Escola Bíblica: Departamento Infantil, Biblioteca: Permite cadastrar livros, dvds, cds, apostilas, sermões, Fazer pesquisa por assunto, autor;

Gestão Estratégica: Permite que o Presidente e administradores gerem gráficos e relatórios que apontem níveis de desempenho da organização nas áreas eclesiástica e financeira;

Segurança: Em segurança é possível definir se o usuário vai poder acessar uma, várias ou todas as congregações. Em grupos de acesso você defini qual grupo o usuário irá fazer parte.

Nota-se a grande variedade de sistemas disponível no mercado para atender necessidades diversas, no entanto são necessidades universais para empresas que necessitam deste tipo de sistema, diferente do que se propõe o SIEADI, as igrejas Assembleia de Deus, pois foi realizado um estudo para que se pudesse atender necessidades específicas da igreja, o que acaba o torne diferente dos existentes no mercado.

2.4 SOFTWARE IMPLANTADO NA IGREJA ASSEMBLEIA DE DEUS.

Durante o processo de desenvolvimento do software, foi realizado uma pesquisa de campo com intuito de identificar as necessidades e os possíveis sistemas já utilizados pela igreja Assembleia de Deus em Itaituba Pará; no decorrer deste, identificamos que a empresa já utiliza um *software* de gestão o WHCIMC3 – Comércio versão 3.7, que é um sistema de ERP (*Enterprise Resource Planning*), Sistemas Integrados de Gestão Empresarial, ele é “multiusuário” (pode ser instalado em redes) e “multiempresas” (pode controlar até 99 empresas). Tem funções cadastro de clientes e fornecedores, contas a pagar e receber, caixa, controle bancário, departamentos e geração de gráficos gerenciais.

2.5 ANÁLISE DO PROGRAMA IMPLANTADO: VANTAGENS

Através da análise do programa, identificamos que atende parcialmente as necessidades da igreja, pois não é um *software* para este tipo de atividade, é direcionado para atender empresas que atuam no comércio varejista, no entanto em se tratando das funções financeiras, atendem a necessidade de controle de caixa, de fornecedores e conta a pagar, esse tipo de empreendimento necessita de algumas funções que específicas desta atividade como, por exemplo, o cadastro de dívidistas e ofertantes quando se trata do controle financeiro, necessita também de um cadastro para controle e quantidade de membros e congregações pertencentes a mesma entidade, geração de relatórios tanto financeiros como de recursos humanos.

3 DESENVOLVIMENTO DA APLICAÇÃO

3.1 TECNOLOGIAS, FERRAMENTAS E LINGUAGENS UTILIZADAS NO DESENVOLVIMENTO DO SIEADI.

Para o desenvolvimento de *software* são utilizadas diversas ferramentas, e no desenvolvimento do SIEADI, não foi diferente, no entanto as ferramentas a serem utilizadas no desenvolvimento de um sistema, depende muito das habilidades do desenvolvedor e da aplicabilidade do sistema a ser desenvolvido, pois isso irá direcioná-lo as quais melhores ferramentas a serem utilizadas, devido à grande diversidade de programas para desenvolvimento e de linguagens de programação existentes.

A linguagem de programação utilizada para dar vida ao SIEADI foi a linguagem *Java*, uma linguagem desenvolvida pela Sun Microsystems, objetivando-se aplicações voltadas para produtos eletrônicos de grande consumo, tais como televisões, videocassetes e outros eletrodomésticos. A escolha desse ramo não obteve o sucesso desejado, no entanto, com a popularização da Internet e suas páginas Web, surgia uma nova e interessante aplicação para a linguagem –as *Applets*– proporcionando animação e interatividade aos até então estáticos documentos HTML. As possibilidades para artes gráficas, multimídia e interação passaram a ser maiores, pois a linguagem *Java* permite desenhar figuras, fazer cálculos, enviar mensagem ao usuário, exibir janelas e gráficos, etc.

Java é uma linguagem que herdou muitas de suas características do C++ e implementa o paradigma da Programação Orientada a Objetos. É uma linguagem interpretada, o que a torna independente da plataforma, ou seja, um mesmo programa pode ser executado em qualquer sistema que possua seu interpretador. Em uma rede com vários computadores diferentes, esta independência de arquitetura mostra seu valor. O formato da arquitetura de *Java* concede sólidos benefícios tanto ao cliente quanto ao desenvolvedor: escreve uma vez e vende para qualquer plataforma.

O SGBD (Sistema Gerenciador de Banco de Dados) utilizado para o SIEADI é o MySQL, é um banco de dados de código aberto e o mais popular do mundo,

possibilita a entrega econômica de aplicativos de banco de dados confiáveis, de alto desempenho e escaláveis, com base na Web e incorporados; utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) como interface. É atualmente um dos bancos de dados mais populares.

Foi criado na Suécia por suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius, que têm trabalhado juntos desde a década de 1980. No dia 16 de Janeiro de 2008, a MySQL AB, desenvolvedora do MySQL, foi adquirida pela Sun Microsystems, e em 2009 a Oracle compraria a Sun Microsystems e todos o seus produtos, incluindo o MySQL. Hoje grandes empresas como Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics e Texas Instruments, usam o MySQL em aplicações de missão crítica. O MySQL hoje suporta *Unicode*, *Full Text Indexes*, replicação, *Hot Backup*, GIS, OLAP e muitos outros recursos de banco de dados.

O ambiente de desenvolvimento utilizado para a construção gráfica do SIEADI foi o NetBeans IDE, é um software de código aberto, disponível para Windows, Linux e Mac, que permite desenvolver páginas de Internet e aplicações para dispositivos móveis em linguagens como JavaScript, C, C++, Ruby on Rails, PHP e outras. Seus recursos permitem que determinados códigos ganhem destaque sintática e semanticamente para melhor organização e mais rapidez do trabalho. Uma facilidade oferecida é a criação de testes a fim de eliminar erros nas aplicações em desenvolvimento, pois ele possui suporte com todos os componentes necessários para simular o ambiente. Além disso, seus diagramas e modelos ajudam a acelerar o trabalho. Encontra-se nele também um ambiente avançado com modelos de projetos em C e C++, com bibliotecas de componentes dinâmicos e estáticos.

O NetBeans IDE é excelente ferramenta para desenvolvedores em qualquer nível de conhecimento, pois suas ferramentas facilitam para os iniciantes e trazem recursos avançados para os profissionais. Ele vem em inglês, possui versão gratuita para testes e é compatível com Mac OS X 10.5 ou superior, Linux e também, com o Windows 2000, 2003, XP, Vista, 7 e 8.

Outro exemplo de programa utilizado é o IText, uma biblioteca JAVA para manipulação de documentos em PDF, que é um formato de arquivo aberto,

originalmente criado pela Adobe. Este documento é independente de sistemas de software ou hardware e atualmente é uma parte essencial da web.

O iText é livre e de código-fonte aberto sob a licença Affero General Public License (AGPL). Atualmente existem duas versões do iText: uma versão para Java e outra versão para C#, esta última chamada de iTextSharp. Essa poderosa ferramenta é usada para criar relatórios em pdf, além de uma variedade de outras coisas, como ingressos digitais para companhias aéreas ou shows, boletos, e muito mais. iText também pode ser integrado com uma aplicação para gerar documentos PDF como uma alternativa à papéis impressos para adicionar assinaturas digitais aos documentos. Além disso, podemos concatenar ou separar diferentes documentos.

O *CorelDraw* para criação e edição de imagens pois é um programa de desenho vetorial bidimensional para design gráfico desenvolvido pela *Corel Corporation*, Canadá. É um aplicativo de ilustração vetorial e layout de página que possibilita a criação e a manipulação de vários produtos, como por exemplo: desenhos artísticos, publicitários, logotipos, capas de revistas, livros, CDs, imagens de objetos para aplicação nas páginas de Internet (botões, ícones, animações gráficas) etc.

O *Adobe Photoshop* é um programa profissional de edição de imagem, para Windows e Mac, que possui, recorte de elementos, elaboração de fotomontagem e adição de filtros, pincéis, gradientes e máscaras, etc.

O *software* é voltado a para o aprimoramento de imagens com recursos de alta qualidade, além de profissionais de design, que criam banners, estampas, desenhos, logomarcas e outros projetos ligados às artes.

A utilização das ferramentas citadas foi de grande importância no desenvolvimento do sistema, pois elas atendem as necessidades básicas e dão o suporte necessário no desenvolvimento de um *software*, a utilização de cada um deles tem um fim específico, e ao mesmo tempo com uma diversidade de funções, o que contribuiu substancialmente no desenvolvimento do sistema SIEADI.

3.2 INTERFACES DO SISTEMA

O sistema foi desenvolvido em linguagem Java, visando favorecer a implantação, pois essa linguagem é multiplataforma, ou seja, ela é acessível em vários Sistemas Operacionais como Windows, Linux, Mac e etc...

O SIEADI possui cinco telas interativas com o usuário, sendo a primeira tela a de *Splash*, que contém informações iniciais do sistema. Essa tela de apresentação ou *splash screen*, é a tela inicial exibida ao abrir o *Software*. A maioria dos aplicativos apresentam *splash screens*, geralmente para exibir o logo do aplicativo ou da empresa que o desenvolveu. Também pode ser uma maneira de “distrair” o usuário durante alguns segundos enquanto a aplicação realiza algum processo ou carregamento inicial. Como mostrado na Figura 2:



Figura 2: Tela de Splash.

Fonte: Dados da Pesquisa, 2014.

Tela de *Splash* será o primeiro contato com o usuário, esta tela aparecerá quando o programa estiver carregando.

Após o carregamento da Tela de *Splash*, entra em funcionamento a tela principal, são nela que contém os menus de acesso aos formulários de registros e consultas do Sistema, os menus contidos na tela Principal são: Menu Arquivo, com sub menu Sair, que ao ser clicado o Sistema é encerrado. O Menu Cadastro é composto de dois submenus, onde ficam localizados os formulários para cadastros,

um para cadastros de membros, e o outro para cadastro de congregações. Esta tela estará frequentemente em contato com usuário, por esta razão deve ser simples e de fácil utilização. A Interface com o Usuário é uma parte fundamental de um *software*; é a parte do sistema visível para o usuário, através da qual, ele se comunica para realizar suas tarefas. É importante que elas sejam amigáveis, simples e objetiva facilitando a vida do usuário. Veja a Figura 3:

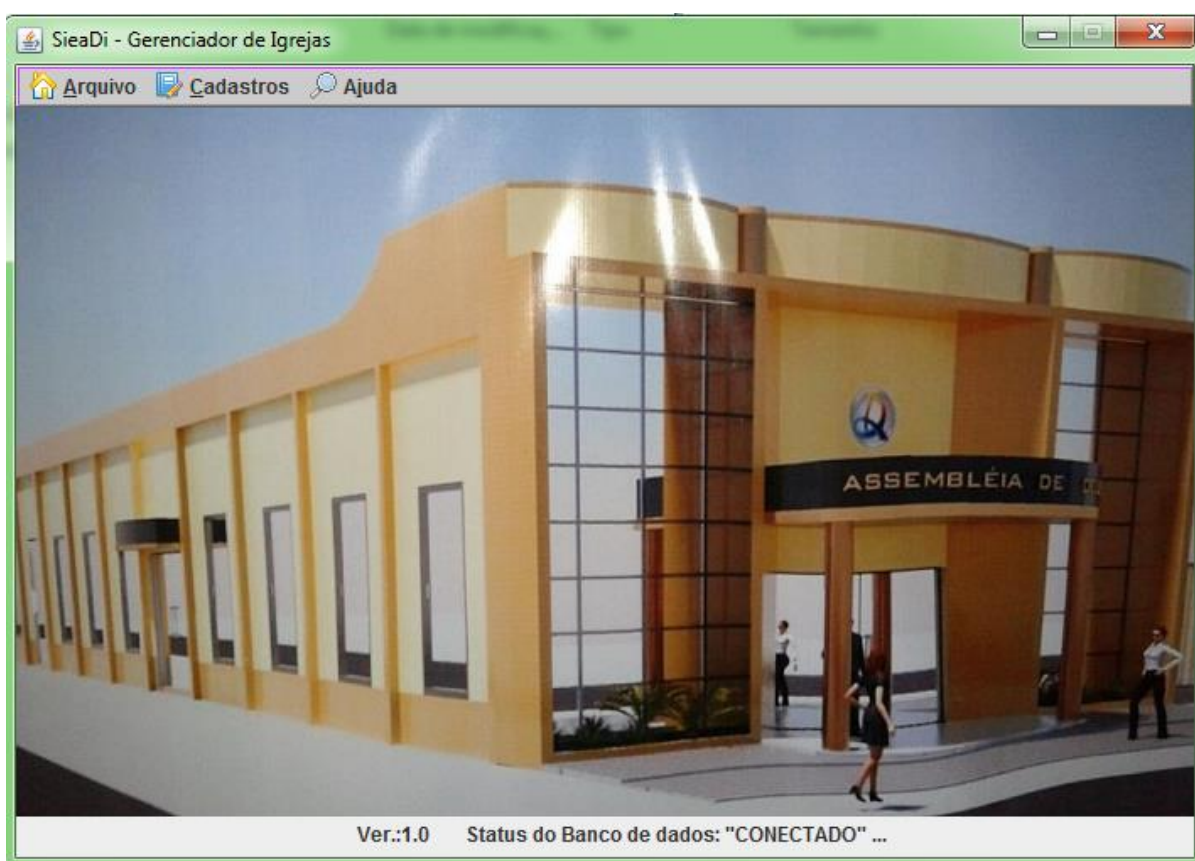


Figura 3: Tela Principal.
Fonte: Dados da pesquisa, 2014.

Nos menus, temos as demais funcionalidades do sistema, incluindo a Tela de Ajuda, conforme Figura 4.



Figura 4: Tela de Ajuda
Fonte: Dados da pesquisa, 2014.

Esta é a Tela de Ajuda, onde o usuário encontrará o nome do software, os desenvolvedores e a versão atual. Em seguida Telas de Cadastros e Consultas, como mostrado nas Figuras 5, 6 e 7.

Figura 5: Formulário Cadastro de Membros.
Fonte: Dados da Pesquisa, 2014.

No formulário de cadastro de membros estão os campos de preenchimento de dados pessoais e dados ministeriais, nos dados pessoais estão: código, nome, endereço, número, bairro, CEP, CPF, RG, UF do RG, Data de Nascimento, Telefone, celular, e-mail, cidade e estado. Nos dados ministeriais: departamento, cargo, nome do pai, nome da mãe, conjugue, data de batismo, UF do local de batismo, local de batismo, naturalidade, profissão, escolaridade, congregação e estado civil; é composto de quatro botões: botão novo, salvar, atualizar, excluir; uma tabela contendo todos os dados cadastrados no sistema e um campo de Pesquisa, onde o usuário ao digitar o nome que deseja extrair dados, o sistema fará a varredura no banco de dados e exibirá na tela o resultado da pesquisa.

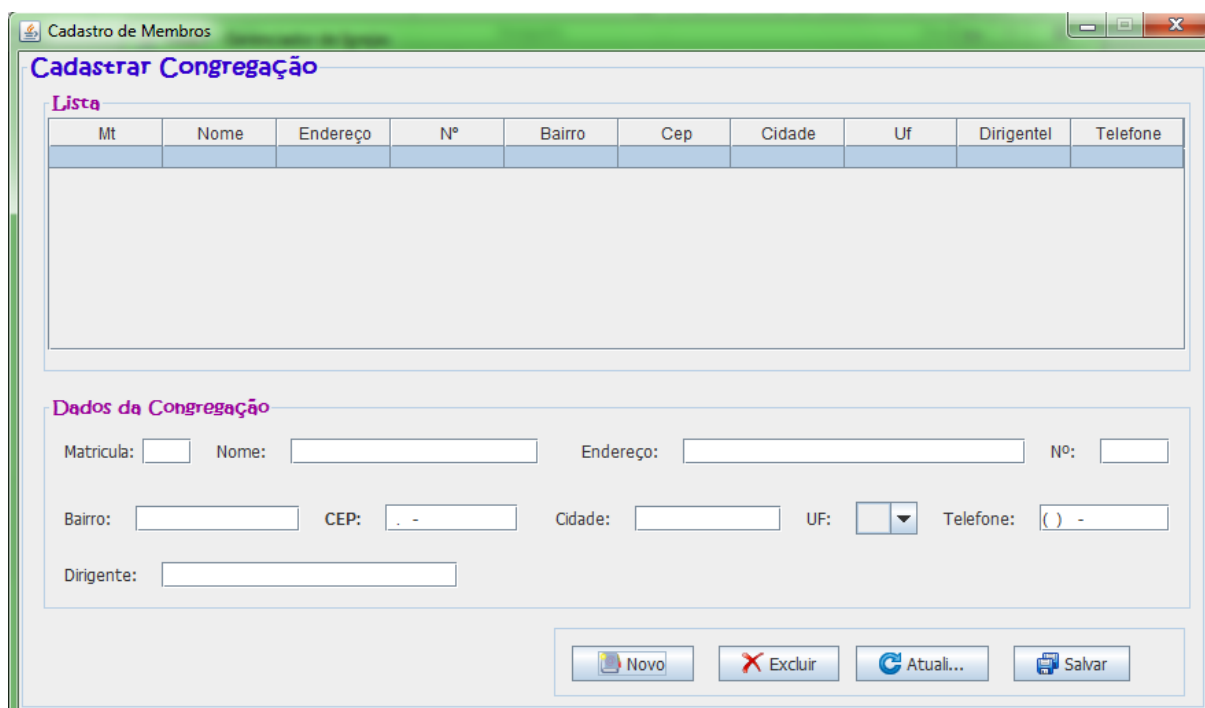


Figura 6: Formulário para Cadastro de Congregações.

Fonte: Dados da Pesquisa, 2014.

Este formulário servirá para realizar o cadastro das congregações existentes na cidade, com nomes de seus dirigentes, endereço e matrícula.

No formulário de cadastro de congregações estão os campos de: matrícula, nome, endereço, número, bairro, CEP e dirigente. Ele é composto de quatro botões: botão novo, salvar, atualizar e excluir; também contém uma tabela contendo todos

os dados cadastrados no sistema e um campo de Pesquisa para fazer a busca da congregação que deseja fazer alterações ou extrair dados.

The screenshot shows a window titled 'Consulta de Membros' with a sub-header 'Consultar Congregação'. Below the header is a search section labeled 'Pesquisar' containing a text box with the label 'Nome:'. Below the search section is a table with the following columns: Mt, Nome, End, N°, Bairro, Cep, Cidade, Uf, Dirigente, and Telefone. The table is currently empty.

Figura 7: Formulário para Consulta de Congregações.

Fonte: Dados da Pesquisa, 2014.

Formulário para consulta e pesquisa de membros cadastrados no sistema, onde serão exibidos os dados dos cadastrados:

The screenshot shows a window titled 'Consulta de Membros' with a sub-header 'Consultar Membro'. Below the header is a search section labeled 'Pesquisar' containing a text box with the label 'Nome:'. Below the search section is a table with the following columns: Cod, No..., En..., Nu..., Bai..., Cep, Cid..., Cpf, Ida..., Tel..., Cel..., Da..., Em..., Es..., Dat..., Uf, Rg, Pr..., No..., No..., Conj, Ca..., De..., Lo..., Uff, Sexo, Co..., and Est... The table is currently empty.

Figura 8: Tela para Consulta de Membros

Fonte: Dados da Pesquisa, 2014

No menu consulta, estão contidas as telas de pesquisa, temos o campo nome, onde ao se digitar o nome desejado, sistema faz a busca no banco de dados e traz as informações cadastradas, mostrando-as na tabela.

3.3 IMPLANTAÇÃO DO SIEADI: ANTES E DEPOIS.

Após identificação do campo pesquisado, no caso, a Igreja Evangélica Assembleia de Deus de Itaituba-PA, constatou-se que a mesma não possuía um sistema informatizado para o cadastro de membros. As Assembleias de Deus brasileiras estão organizadas em forma de árvore, na qual cada Ministério é constituído pela igreja-sede com suas respectivas filiais (congregações e pontos de pregação). O sistema de administração é um misto entre o sistema episcopal e o sistema congregacional, por meio do qual os assuntos são previamente tratados pelo ministério. Atualmente o cadastro de membros e controle da entidade é feito manualmente sem um controle preciso do efetivo. A Igreja Evangélica Assembleia de Deus em Itaituba é composta por uma Matriz (Igreja-Sede) e 35 Congregações (filiais) espalhadas ao longo da cidade, sendo assim avaliado entre 5 a 6 mil membros, congregados e simpatizantes.

O sistema foi desenvolvido tendo em vista que já existe um programa para o controle financeiro da igreja para atender as necessidades financeiras. O SIEADI foi criado pra atender inicialmente as necessidades de cadastro de membros e congregações, para que se pudesse ter um controle sobre a quantidade de membros e congregações existentes na cidade.

Para a instalação do sistema SIEADI na secretaria foi necessário utilizar um computador com Processador: Intel Core 2 Duo, Memória DDR 2, 800 MHz, 2 GB e HD de 160GB, já instalados a Máquina Virtual JAVA - JVM, e o Banco de Dados MySQL.

O sistema que era utilizado na Igreja Assembleia de Deus, antes do estudo, era no formato tradicional, com arquivos de papel e listagens em Excel, além de

fichas do tipo cartão de identificação, preenchidas manualmente, com as informações dos membros.

Com a instalação do sistema SIEADI, o arquivo da secretaria passou a ser virtual, com os dados cadastrados no computador, mas mantendo o arquivo tradicional para, aos poucos, irem se desfazendo da necessidade de utilização do arquivo velho, já que se pretende ampliar as possibilidades do programa, indo além do cadastro de membros, acrescentando numa nova versão a geração e impressão de relatórios, bem como agendamento e pauta de reuniões.

A implantação do programa SIEADI na secretaria da igreja Assembleia de Deus permitirá economia de tempo, de espaço, concentração de diversas ações de registro e controle no mesmo ambiente, facilitando o trabalho de acompanhamento das atividades da instituição, tornando prático o serviço dos auxiliares do setor, além de redução de custos com recursos humanos, pois possibilita que todo o movimento da secretaria seja organizado por uma só pessoa.

4. RESULTADOS PRÁTICOS DA IMPLANTAÇÃO DO SISTEMA SIEADI

4.1 A IMPORTÂNCIA DA TECNOLOGIA DA INFORMAÇÃO NAS ORGANIZAÇÕES

Segundo REZENDE e ABREU (2000), a TI está fundamentada nos seguintes componentes:

- Hardware e seus dispositivos e periféricos;
- Software e seus recursos;
- Sistemas de telecomunicações;
- Gestão de dados e informações.

A união desses componentes eleva a potencialidade de atuação das empresas, agregando valor de mercado e capacidade de gerir as informações de forma eficiente.

O acompanhamento de um software é sem dúvidas uma missão infinita entregue aos responsáveis pela criação ou implantação do mesmo, onde são analisados os pontos críticos e os favoráveis para assim manter o sistema em funcionamento e atualizado no que tange as novas tecnologias que vão surgindo ao longo dos anos.

Para Stoner (1999) somente com informações precisas e na hora certa os administradores podem monitorar o progresso na direção de seus objetivos e transformar os planos em realidade.

Assim, para esse autor as informações devem ser avaliadas segundo quatro fatores:

. **qualidade da informação** - quanto mais precisa a informação, maior sua qualidade e com mais segurança os administradores podem contar com ela no momento de tomar decisões;

. **oportunidade da informação** - para um controle eficaz, a ação corretiva deve ser aplicada antes de ocorrer um desvio muito grande do plano ou do padrão; portanto as informações devem estar disponíveis para a pessoa certa no momento certo;

- . **quantidade da informação** - dificilmente os administradores podem tomar decisões precisas e oportunas sem informações suficientes; contudo é importante que não haja uma inundação de informações, de modo a esconder as coisas importantes;
- . **relevância da informação** - de modo semelhante, a informação que os administradores recebem deve ter relevância para suas responsabilidades e tarefas.

O propósito básico da informação, dentro do contexto organizacional, de acordo com Oliveira (1998), é o de habilitar a empresa a alcançar seus objetivos por meio do uso eficiente dos recursos disponíveis (pessoas, materiais, equipamentos, tecnologia, dinheiro, além da própria informação). Neste sentido, a teoria da informação considera os problemas e as adequações do seu uso efetivo pelos tomadores de decisão.

A eficiência na utilização da informação é medida em relação ao custo para obtê-la e o valor do benefício derivado de seu uso. Associam-se à produção da informação os custos envolvidos na coleta, processamento e distribuição. (Oliveira, 1998, p. 103).

Os benefícios da implantação do sistema de informação - SIEADI, na Igreja Assembleia de Deus em Itaituba, dentro do que Oliveira ressalta, foi utilizado e adquirido, como vantagens, no âmbito organizacional da igreja, agilizando o processo de agilidade antes não favorável no papel organizacional e administrativo da referida igreja, Assembleia de Deus em Itaituba.

Conforme Gonçalves (1993), a relação entre estrutura organizacional e tecnologia tem sido alvo de grande atenção, uma vez que as recentes inovações trazem mudanças radicais nas organizações, já que são capazes de alterar a forma de administrar a empresa ou até mesmo o local de realização do trabalho.

Já Paladini (2002) comenta que os indicadores da qualidade são essencialmente estratégicos e por isso são conhecidos como indicadores de sobrevivência da empresa.

Quando se faz um projeto de software, levam-se em conta todas as necessidades da entidade, procurando assim sanar a cada uma dessas necessidades, para no final proporcionar melhor agilidade em armazenamento de dados e manuseio dos mesmos. E para análise de qualidade usam-se indicadores para mostrar se foram extintos ou não os problemas da instituição e etc...

Carvalho et al (2005) acrescentam ainda que todo indicador é definido em bases quantitativas que o torna um mecanismo mensurável e permite avaliar de forma direta ou indireta o impacto do produto final sobre o consumidor e também avaliar o quanto as melhorias no processo produtivo são relevantes, sob o ponto de vista do consumidor, para a qualidade do produto final.

Para a construção do SIEADI, foi realizado uma pesquisa no âmbito da corporação para obter, as reais necessidades da organização; foi constatado que a maior deficiência era a falta de controle de seus clientes, denominados membros, visto que o lucro direto dessa empresa está ligado ao total de pessoas que contribuem com seus dízimos e ofertas para o funcionamento do estabelecimento. Com esse controle de cadastrados tem-se um resultado de pesquisa em fração de segundos para quaisquer tomadas de decisões. Ao final construiu-se um software que veio sanar as demandas da igreja no que tange ao controle da membresia. Obtivemos êxito no que diz respeito a tempo, espaço, custos e tal tomadas de decisões, conforme:

Para Paladini (1997), satisfeitos os quesitos de custo e tempo, a pequena empresa não pode abrir mão do fator qualidade nos seus sistemas de informação. Se qualidade pode ser definida como adequação ao uso um sistema de informação deve ser adequado às necessidades da pequena empresa.

A igreja usufruiu hoje da vantagem de um sistema de gestão com foco em atividades estratégicas que trarão melhores resultados sem ter que abdicar do cuidado e atenção que a gestão administrativa requer.

4.1.1 Economia de Tempo

Segundo Alfredo Bosi (2011), os meios eletrônicos de informação foram inventados para poupar tempo. Desde a revolução industrial dos fins do século 18, há pelo menos duzentos anos, isto é, há pelo menos sete gerações (um átimo na história do *Homo sapiens*), grande parte do esforço da tecnologia consiste em acelerar os procedimentos de produção de bens culturais, apressando igualmente a sua circulação. No sistema capitalista, esses bens assumem a forma de mercadorias.

Acredita-se que o uso inteligente do tempo, tem sido um dos maiores desafios dentro de muitas ou se não de todas as empresas pelo mundo, devido vivermos em um sistema organizacional em crescentes mudanças, há uma grande necessidade de tal administração.

Segundo Thompson (1991), o capitalismo industrial foi capaz de trazer mudanças em rotinas de trabalho, o que podemos chamar de “nova” disciplina temporal, que em 1967 publicou seu clássico artigo “Time, work-discipline and industrial capitalism”, analisou minuciosamente as mudanças relacionadas a essa nova forma de vida. Para ele, a base do lucro deste sistema é a disciplina no trabalho através do controle do tempo, relacionada ao conjunto da divisão e exploração do tempo de trabalho humano.

Ainda Thompson (1991), lembra que quando o trabalho começa a ser controlado e remunerado pelo tempo (número de horas trabalhadas), reforça-se a associação de tempo a dinheiro, “o tempo começa a ser relacionado a dinheiro: ao dinheiro do empregador”. Nessa conjuntura, muda-se a essência do trabalho: não é mais a tarefa que tem importância, mas sim o “valor” do tempo e o retorno do investimento, alterando assim profundamente as relações de trabalho.

Giddens (1991) reflete sobre as mudanças da modernidade social e vê o período moderno como multidimensional no âmbito das instituições e, baseado nas correntes teóricas tradicionais da Sociologia, que derivam dos escritos de Marx, Durkheim e Weber, o autor considera que os papéis do capitalismo, da

industrialização e da racionalização, podem auxiliar na melhor compreensão do fenômeno social.

Ao se basear nos escritos de Marx, Giddens (1991) acredita que o capitalismo é a força transformadora principal que modela o mundo moderno, concebendo o trabalho humano como mercadoria; nas ideias de Durkheim, o industrialismo, e não exatamente o capitalismo, e a complexa divisão do trabalho, são responsáveis pela rápida transformação da vida social através da exploração industrial da natureza voltada à produção para suprir necessidades humanas. Por último e de acordo com o pensamento de Weber, Giddens (1991) enfatiza a racionalização expressa na tecnologia e organização das atividades humanas (burocracia) no processo de modernização.

Apesar das divergências, cada uma destas tradições de pensamento tem a sua devida importância e, portanto, as características da modernidade apresentadas por cada uma delas não devem ser encaradas como mutuamente exclusivas na análise do movimento social moderno.” (Giddens, 1991, p. 89).

Com o Sistema já em operação, o que gastaria em tempo para cadastro de um membro (visto que era feito de forma manual), com a implantação, deste recurso tecnológico, foi reduzido significativamente, confirmando as teorias acima mencionadas, melhorando os trabalhos desenvolvidos administrativamente na Igreja, que num mundo capitalista tempo significa dinheiro.

Para sobreviver no próximo século e não ser expulsa do mercado, uma empresa precisa fornecer valor superior aos seus clientes. Fornecer valor é dar aos clientes tudo o que eles querem e nada do que não querem; a melhor qualidade e os melhores preços, seja rápido e sirva o cliente com um sorriso! Os clientes atuais, sofisticados e atentos, fazem quatro julgamentos quando decidem fazer negócio com uma organização: (1) qualidade, (2) serviço, (3) custos e (4) tempo (Band, 1997, p. 58).

4.1.2 Redução de Espaço



Figura 9: Arquivos
Fonte: imagens do google

Bitner (1992) considera a evidência física como um fator de impacto sobre as percepções dos clientes e dos funcionários, pois comunicam atributos de qualidade em relação aos serviços e criam expectativas nos clientes e nos funcionários, podendo inclusive, afetar a produtividade, a motivação e a satisfação dos funcionários.

O cenário de serviços pode atrair ou barrar o acesso, pode influenciar a eficácia com que funcionários e clientes executam suas atividades, bem como influenciar a natureza e a qualidade das interações sociais. O cenário, ao comunicar significados, pode causar reações internas, produzindo efeitos sobre as crenças das pessoas em relação ao lugar e as pessoas que se encontram naquele lugar. Da mesma forma, as evidências físicas do cenário que compõem o ambiente físico podem auxiliar as pessoas a qualificar a organização e gerar reações emocionais, que por sua vez geram comportamentos (BITNER, 1992, p.57-71).

A relação entre estrutura organizacional e tecnologia segundo Golçalves (1993), tem sido alvo de grande atenção, uma vez que as recentes inovações trazem mudanças radicais nas organizações, já que são capazes de alterar a forma de administrar a empresa ou até mesmo o local de realização do trabalho.

De acordo com Pontes (2006) Durante a Revolução Industrial os modelos de administração, organização e sociedade passaram por transformações. A economia se baseava fundamentalmente na fabricação de produtos homogêneos produzidos em linha de montagem, com o controle de tempos e movimentos, e pela produção em série. Além da separação entre a execução e o planejamento: onde a execução baseava-se principalmente na fragmentação das funções (trabalho parcelar) e pela construção e consolidação do operário-massa; enquanto a gerência se preocupava com o planejamento do trabalho.

Ao se implantar sistemas informatizados, propõe-se logo de cara a mudança no âmbito físico das organizações, já que o que se ocupa de espaço físico, com prateleiras cheias de arquivos impressos, chega a ser alarmante.

Laudon e Laudon (1999), afirmam que "a razão mais forte pelas quais as empresas constroem os sistemas, é para resolver problemas organizacionais e para reagir a uma mudança no ambiente".

De acordo com Turban; MacLean; Wetherbe (2004), o sistema de informação coleta, processa, armazena analisa e dissemina informações com um determinado objetivo dentro de um contexto e como qualquer outro sistema inclui inputs (dados, instruções) e outputs (relatórios, cálculos). Sistema opera dentro de um ambiente, não necessariamente computadorizado, mesmo que atualmente a maioria os seja, processa os inputs, que são enviados para o usuário ou para outros sistemas. Obtemos assim redução em tempo custos e espaços.

Para MORGAN, 1996; DALE, 2005 o controle organizacional, a implantação de mudanças, bem como a transmissão dos valores culturais da organização, pode ser efetuada com o auxílio de artefatos tangíveis como arquitetura, layout, sinalizações, cores, aparência externa do edifício e materiais utilizados por meio dos seus significados implícitos e da interação de seus significados com a atividade

social e sua re-significação ao longo do tempo, sendo incorporados à subjetividade dos indivíduos que interagem nestes espaços.

Na igreja Assembleia de Deus em Itaituba todos os cadastros de membros eram feitos de forma manual, onde cada um tinha uma pasta contendo ficha com foto e as demais informações pessoais; ao se implantar o SIEADI, tudo isso foi sendo eliminado, já que não é mais preciso imprimir fixas, a não ser para alguma tomada de decisão; e aos poucos as pastas foram evadindo-se das prateleiras com medidas adotadas pela instituição sendo uma delas a de entregar gradativamente essas pastas aos seus respectivos donos. Atualmente e depois da nova tecnologia, a realidade foi de economia de tempo, agilidade, modernidade e aspectos positivos internos, externos e no visual inovador.

4.1.3 TOMADA DE DECISÃO

Segundo Kazmier (1975) Tomada de Decisão é o ato ou efeito de tomar, de decidir, resolução, determinação, deliberação, desembaraço, disposição, coragem capacidade de decidir onde a habilidade em tomar decisões é a chave para o planejamento bem sucedido em todos os níveis da gestão. Isto envolve mais que uma simples seleção de planos de ação que assume pelo menos três fases: diagnóstico; descobertas de alternativas e análises. Para o mesmo o diagnóstico é a primeira fase do Processo de Tomada de Decisão que tem como função de identificar e aclarar o problema.

Conforme Motta (1986), desde o início do século, as empresas foram organizadas a partir dos princípios da Teoria Científica, cujo precursor, Frederick Taylor, preconizava a divisão do trabalho, a disciplina, a racionalização dos métodos e sistemas de trabalho e a padronização da produção. Para garantir que os padrões

de produção fossem atingidos, sugeriu a seleção, o treinamento e o controle dos trabalhadores, incluindo o pagamento pela produção gerada em cada um deles, por meio de um sistema de incentivos monetários.

Corroborando esta afirmativa, Oliveira (1997) enfatiza que os antigos paradigmas tayloristas levaram a profundas distorções nas organizações, que desumanizaram as relações no trabalho e robotizaram os indivíduos, resultando em desmotivação e, conseqüentemente, em baixa produtividade. Neste contexto, os gerentes assumiam o papel de “controladores” encarregados de fiscalizar os funcionários para garantir maior produtividade. Por sua vez, os trabalhadores eram explorados, tornavam-se cada vez mais alienados e não mantinham controle sobre sua produtividade.

Daft (2005) julga que a tomada de decisão é uma das atividades essenciais no contexto gerencial, pois tudo o que o gerente vier a realizar recairá sobre qual o melhor caminho a seguir, portanto eles necessitam de determinadas aptidões para obter êxito na função de gerenciar.

A tomada de decisão é o processo de identificar os problemas e as oportunidades e em seguida solucioná-los. A tomada de decisão envolve esforços antes e depois da escolha real. (DAFT, 2005, p.196).

A Igreja Assembleia de Deus em Itaituba conta com um programa que vem suprir as demandas de tomadas de decisão, oferecendo um sistema de banco de dados contendo todas as informações necessárias cadastradas, uma vez que através de uma consulta é possível ter acesso as informações cadastradas no sistema, facilitando as tomadas de decisões em tempo real e ágil.

4.1.4 REDUÇÃO DE CUSTOS COM RH

Para Claver et al (2001 apud TEIXEIRA JUNIOR; OLIVEIRA, 2003), o sucesso dos sistemas de informação nas organizações está condicionado à união de dados, tecnologia da informação e pessoas, sendo estas últimas estabelecidas na condição de indivíduos dentro da cultura organizacional.

Giurliani (1999) enfatiza que o uso de TI carrega consigo custos 'invisíveis', que são aqueles referentes aos gastos com usuário final, como suporte e treinamento informais (quando alguém interrompe suas atividades para auxiliar um colega com dificuldades referentes ao uso ou obtenção da informação); perda de tempo em atividades extremamente elaboradas para obter uma informação, cujo valor não compensa o trabalho empregado; finalmente, custos decorrentes da perda de produtividade, devido à indisponibilidade da informação (queda do sistema, por exemplo). Da mesma forma, para o autor os custos mais facilmente mensuráveis ('visíveis') seriam aqueles relacionados a *hardware* e *software* (despesas com compra e/ou leasing de equipamentos, *upgrades*, atualizações); gerenciamento (redes, sistemas, banco de dados); suporte (*helpdesk*, treinamento, viagens, manutenção); desenvolvimento (aplicações, testes e documentação, customização e manutenção); e taxas de comunicação (aluguel de linhas de comunicação de dados, taxas de acesso a servidores).

De acordo com o Sebrae (2000), as vantagens que a TI traz para a pequena empresa podem ser divididas em três grupos.

a) Menores custos: a informática, quando bem utilizada, reduz os custos da empresa, porque agiliza e possibilita maior segurança e confiabilidade nos processos, rotinas e controles administrativos; simplifica as tarefas burocráticas; reduz os erros e praticamente elimina a repetição do trabalho.

b) Maior produtividade: possibilita que as pessoas produzam mais, em menos tempo, com menor dispêndio de recursos; permite aproveitar melhor a capacidade produtiva da empresa com o planejamento e o controle da produção; armazena e

localiza imediatamente informações fundamentais para os negócios; agiliza os processos de tomada de decisões em relação a preços, estoques, compras e vendas, entre outros.

c) Maior qualidade: a qualidade dos produtos e serviços é melhorada, pois as tecnologias de informação ajudam a manter o padrão dos produtos dentro das especificações estabelecidas; proporciona melhores condições de trabalho para os empregados; reduz esforços com a burocracia para concentrá-los nas atividades fins da empresa.

Segundo uma análise efetuada por Weber ampliada nas últimas décadas, indica diferentes possibilidades para a relação entre empreendedorismo e religião. Carswell e Rolland (2004) afirmam que há, atualmente, uma diversidade étnica e religiosa cada vez maior, exercendo influências distintas sobre o empreendedorismo. Diante disso, verifica-se que os valores religiosos agem sobre o comportamento dos indivíduos gerando um efeito direto sobre suas motivações e ações, influenciando na propensão ao empreendedorismo e no processo empreendedor (Dana, 2010; Drakopoulou Dodd & Gotsis, 2007).

A igreja, que é a todo instante denominada empresa nessa pesquisa, garantiu economia de custos com recursos humanos, uma vez que implantado o sistema viu-se descartado a necessidade de contratação de mais funcionários pra efetuar as diversas tarefas administrativas, já que o SIEADI pode ser manipulado facilmente por qualquer individuo uma vez treinando. A igreja é um empreendedorismo e como tal pode adqvar-se as demandas do mercado tecnológico

CONCLUSÃO

Neste estudo, para implantar um sistema de *software* na Igreja Assembleia de Deus, primeiro procurou-se responder às seguintes questões: A Igreja Evangélica Assembleia de Deus em Itaituba possui um sistema informatizado? Como criar um *software* que vai ser útil na administração da entidade? Quais as pesquisas e trabalhos já existentes na área? O custo do *software* impede o requisitante de viabilizar a montagem do mesmo? Dentro do ambiente investigado existe um *software* que supra as carências da Igreja?

Com base em uma pesquisa bibliográfica delimitou-se a base conceitual do estudo sobre sistema informatizado, *software* aplicado a entidades religiosas e diferentes tipos de cadastro hoje existentes. Após uma pesquisa de campo, foram destacados os ambientes que compõem o espaço de armazenamento das informações, antes da instalação do Programa e depois da sua implementação, tais como secretaria, tesouraria, cadastro de membros e pauta de reuniões.

Na Secretaria foram verificados os Cadastros Antigos, em formulários de Papel. Depois foram instalados itens dos Cadastros Principais, Cadastros Secundários, Tabelas Principais, Tabelas Secundárias, Relatórios Principais, Relatórios Parte 1 e Parte 2 e Gráficos. Destes, apenas alguns foram selecionados para constar no trabalho.

Para a tesouraria, há um programa específico gerenciado pelo departamento financeiro, razão pela qual não foi possível verificar os dados relacionados a contabilidade, até mesmo porque não foi dado acesso ao funcionamento deste sistema.

Como o cadastro de membros possui o controle total sobre sua membresia optamos por começar a pesquisa de campo por este item para facilitar o trabalho da secretaria, também por ser completo, limpo e de fácil navegação, permitindo fazer buscas com várias opções de filtros de pesquisa.

O trabalho permitiu a colaboração com a Igreja, no sentido de organizar os dados da membresia da entidade, além da oportunidade de aplicação do conhecimento teórico e prático sobre análise e desenvolvimento de *software* adquirido no decorrer do curso de Tecnologia em Análise e Desenvolvimento de

Sistemas do Instituto Federal de Educação Ciência e Tecnologia do Pará-Campus de Itaituba.

O programa implantado destinou-se inicialmente em apenas, cadastrar os membros da igreja que pelos resultados práticos e eficientes obtidos na excelência dos resultados, tanto de tempo, economia e lucro, pretende-se posteriormente, extender, em cursos de pós-graduação, especialização ou mestrado, incrementar uma nova versão do programa com pautas de reuniões, atas e relatórios com tomada de decisões dos gerenciadores do programa. A satisfação com os frutos obtidos e repercussão, resultou em um trabalho de grande valor.

BIBLIOGRAFIA

ALFREDO BOSI. **Considerações Sobre Tempo e Informação**. 2011. Disponível em: <<http://www.iea.usp.br/publicacoes/textos/bosiinternet.pdf>>.

ANGELOTTI, E. S. **Banco de Dados**. 1. ed. Curitiba: Editora do Livro Técnico, 2010.

BAND, William A. **Competências críticas:dez novas idÉias pararevolucionar a empresa**. Rio de Janeiro: Campus, 1999.

BITNER, Mary Jo. **Services capes: the impact of physical surroundings on customers and employees**. Journal of Marketing. v. 56, 1992, p.57-71.

BENEDUZZI, H. M. **Lógica de Programação: Introdução ao Desenvolvimento de Software** /Humberto Martins Beneduzzi; João Ariberto Metz. São Paulo: Editora do livro Técnico, 2010.

BRASIL ESCOLA. **Protestantismo**. 2014. Disponível em: <<http://www.brasilestola.com/religiao/protestantismo.htm>>. Acesso em 20 abril 2014, 18:22:12.

CARVALHO, M.M et al.. **Gestão da Qualidade: teoria e casos**. Rio de Janeiro: Elsevier, 2005.

COMUNIDADE CRISTÃ JESUS PARA O MUNDO, **História da Igreja**. Niterói: 2005. 105 p.

CHEN, P. **The entity-relationship model - toward a unified view of data.** ACM.

DANA, L.-P. (2010). **Entrepreneurship and religion.** Cheltenham: Ed. Elgar.

DATE, C. J. **Introdução a Sistemas de Banco de Dados.** 6ª ed. Elsevier Editora, 2000.

DAFT, Richard L. **Administração.** 6.ed. São Paulo: Pioneira Thomson Learning, 2005.

DRAKOPOULOU D, S. D; & GOTSIS, G. **The interrelationships between entrepreneurship and religion.** The International Journal of Entrepreneurship and Innovation, 8(2),93-104. 2007.

ELMASRI, R.; NAVATHE, S. **Sistemas de Banco de Dados.** 4 ed. Pearson Addison-Wesley, 2005.

FURNIVAL, A. C. **A Participação dos Usuários no Desenvolvimento de Sistemas de Informação.** In; Ciência da Informação – Vol 25 Numero 2, 1995.

GIDDENS, Anthony. **As conseqüências da modernidade.** São Paulo: UNESP, 1991.

GIURLIANI, S. **A difícil arte de fazer contas**. Gestão Empresarial,v.1, n. 3, p.7-16, 1999.

GONÇALVES, J. E. L. **Os impactos das novas tecnologias nas empresas prestadoras de serviço**. Revista Administração de Empresas, v. 34, n. 1, p.63-81, 1993.

GARLAN. D; SHAW. M. **An introduction to software architecture. Advances in Software Engineering and Knowledge Engineering**. 1993.

GONÇALVES, Edson. **Dominando NetBeans**. 1 Ed. Rio de Janeiro: Editora Ciência Moderna Ltda., 2006.

GROPPELLI, A. A.; NIKBAKHT, E. **Administração Financeira**. 2. Ed. Tradução Célio Knipel Moreira. São Paulo: Saraiva 2002.

G1. **Número de evangélicos aumenta 61% em 10 anos, aponta IBGE**. 2012. Disponível em: < <http://g1.globo.com/brasil/noticia/2012/06/numero-de-evangelicos-aumenta-61-em-10-anos-aponta-ibge.html>>. Acesso em: 20 abril de 2014, 19:23:13.

JUNGER, Sérgio. **A Organização e importância do Sistema de Informação**. Universidade de Tecnologia do Minho, Portugal, 2012.

KAZMIER, Leonard. J. **Princípios de Gerência**. 2. ed. Rio de Janeiro: Pallas,1975.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5a ed., Campus, 2006.

LAUDON. K. C; LAUDON. J. P. **Sistemas de Informação Gerenciais: administrando a empresa digital**. 5. Ed. Tradução Arlete Simille Marques. São Paulo: Pearson Prentice Hall, 2004.

LAUDON, KENNETH C.; LAUDON, JANE PRICE. **Sistemas de Informação**. 4. Ed.LTC: Rio de Janeiro 1999.

MOTTA, F. C. P. Teoria das organizações: evolução e crítica. São Paulo: Pioneira, 1986. 112 p.

O'BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da internet**. São Paulo: Saraiva, 2003.

OLIVEIRA, M. **de Energia emocional**. São Paulo : Makron Books, 1997. 220 p.

OLIVEIRA, D. P. R. (1998). **Planejamento Estratégico: conceitos, metodologia e práticas**. São Paulo, Atlas.

OLIVEIRA. C. **Banco de Dados: Curso Prático H.P. de. SQL**. 1a Ed. Novatec Editora Ltda, São Paulo-SP, 2002.

PALADINI, E.P. **Avaliação Estratégica da Qualidade**. São Paulo: Editora Atlas, 2002.

PFLEEGER. S.L. **Engenharia de Software: teoria e prática**. 2^a.ed.São Paulo: Pearson Prentice Hall, 2004.

PRESSMAN. R. **Engenharia de Software**. 2006. McGraw Hill– Processo: Uma visão Genérica.

PUGA. S. **Lógica de Programação e Estruturas de Dados, com Aplicações em Java**. \Sandra Puga, Gerson. – 2. Ed. – São Paulo: Pearson Prentice Hall, 2009.

RAMOS, I. M. P. **Aplicações das Tecnologias de Informação que suportam as Dimensões Estrutural, Social, Política, Simbólica do Trabalho**. Tese de doutorado em Tecnologias e Sistemas de Informação – Universidade do Minho, Departamento de Sistemas de Informação. Guimarães – Portugal, Dezembro de 2000.

REZENDE. D. A. **Sistemas de Informações Organizacionais: guia prático para projetos em cursos de administração, contabilidade e informática**. São Paulo: Atlas, 2005.

REZENDE, Denis Alcides; ABREU, Aline França de. **Tecnologia da informação aplicada a sistemas de informação empresariais: o papel estratégico da informação e dos sistemas de informação nas empresas**. São Paulo: Atlas, 2000.

SANTOS, Meneses Ciro. **Desenvolvimento de Aplicações Comerciais com Java e NetBeans**. Editora Ciência Moderna. Ltda.2010.

SHAW. M; GARLAN. D. **Software Architecture: Perspectives on na Emerging Discipline**. Englewood Cliffs, NJ: Prentice Hall, 1996.

SEBRAE Informática: **solução para a pequena empresa**. Brasília: SEBRAE, 1994.

SIGNIFICADOS. **Significado de Igreja**. 2014. Disponível em: <<http://www.significados.com.br/igreja/>>. Acesso em: 21 de abril de 2014, 20:24:43.

STONER, J.A. F.; FREEMAN, R. E. **Administração. As Organizações e a Necessidade de Administração**. Rio de Janeiro.1999. 5ª.ed. trad. CALADO, A. revi. Agrícola de Souza betblem, D. Sc. (UFRJ). p. 4 e p. 5.

SOMMERVILLE, L. **Engenharia de Software**. Tradução Ivan Bosnic e Kalinka G. revisão técnica Kechi Hiramã. 9. Ed. São Paulo: Pearson Prentice Hall, 2011.

TURBAN, E.; MCLEAN, E.; WETHERBE, J. **Tecnologia da Informação Para Gestão**. Tradução de Renate Schinke. 3.ed. Porto Alegre: Bookman, 2004.

THOMPSON, Edward Palmer. **Time, work-discipline and industrial capitalism**. In: Customs in common. London : Merlin Press, 1991. p.352–403.

VIANNA. E. C. C.M. **Estudo e Proposta de Práticas Participativas na Gestão de Requisitos**. Trabalho Final Mestrado Profissional – UEC, Instituto de Computação. Campinas, 2004.

ANEXO: CÓDIGO FONTE DO SOFTWARE SIEADI.

Formulário Splash Screen.

```

1
2 package PRINCIPAL;
3
4 import static java.lang.Thread.sleep;
5 import java.util.logging.Level;
6 import java.util.logging.Logger;
7
8 /**
9  *
10  * @author Adriana Coimbra <dri8_@hotmail.com>
11  */
12 public class TelaSplash extends javax.swing.JFrame {
13     private static final long serialVersionUID = 1L;
14
15     public TelaSplash() {
16         initComponents();
17         initComponents();
18         setResizable(true);
19         setLocationRelativeTo(null);
20
21         new Thread(){
22             @Override
23             public void run(){
24                 for(int i = 0; i<101; i++){
25
26                     try {
27                         sleep(70);
28                         EfeitoVideo.setValue(i);
29
30                     } catch (InterruptedException ex) {
31
32                         Logger.getLogger(TelaSplash.class.getName()).log(Level.SEVERE, null, ex);
33                     }
34
35                 }
36
37                 new PRINCIPAL.SieaDi().setVisible(true);
38                 dispose();
39             }
40         }.start();
41
42     }
43
44     private TelaSplash(Runnable aThis) {
45         throw new UnsupportedOperationException("Not yet implemented");
46     }
47
48     @SuppressWarnings("unchecked")
49     // <editor-fold defaultstate="collapsed" desc="Generated Code">
50     private void initComponents() {
51

```

```

52     EfeitoVideo = new javax.swing.JProgressBar();
53     jLabel1 = new javax.swing.JLabel();
54     jLabel2 = new javax.swing.JLabel();
55
56     setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
57     setTitle("SieaDi Sistema Gerenciador de Igreja");
58     setAutoRequestFocus(false);
59     setBackground(new java.awt.Color(102, 153, 255));
60     setName("TelaSplash");
61     setResizable(false);
62     setUndecorated(true);
63     getContentPane().setLayout(null);
64
65     EfeitoVideo.setBackground(new java.awt.Color(255, 255, 255));
66     EfeitoVideo.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
67     EfeitoVideo.setForeground(new java.awt.Color(0, 0, 0));
68     EfeitoVideo.setBorder(javax.swing.BorderFactory.createCompoundBorder());
69     EfeitoVideo.setString("%");
70     EfeitoVideo.setStringPainted(true);
71     EfeitoVideo.addAncestorListener(new javax.swing.event.AncestorListener() {
72         public void ancestorMoved(javax.swing.event.AncestorEvent evt) {
73         }
74         public void ancestorAdded(javax.swing.event.AncestorEvent evt) {
75             EfeitoVideoAncestorAdded(evt);
76         }
77         public void ancestorRemoved(javax.swing.event.AncestorEvent evt) {
78         }
79     });
80     getContentPane().add(EfeitoVideo);
81     EfeitoVideo.setBounds(0, 230, 470, 10);
82
83     jLabel1.setBackground(new java.awt.Color(0, 153, 255));
84     jLabel1.setFont(new java.awt.Font("Kristen ITC", 1, 10)); // NOI18N
85     jLabel1.setText("SieaDi Software -- Igreja Evangélica Assembleia de Deus em
Itaituba");
86     getContentPane().add(jLabel1);
87     jLabel1.setBounds(0, 240, 470, 20);
88
89     jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGENS/Deus.jpg"))); // NOI18N
90     getContentPane().add(jLabel2);
91     jLabel2.setBounds(0, 0, 470, 230);
92
93     java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
94     setBounds((screenSize.width-470)/2, (screenSize.height-264)/2, 470, 264);
95 } // </editor-fold>
96
97 private void EfeitoVideoAncestorAdded(javax.swing.event.AncestorEvent evt) {
98
99 }
100
101 public static void main(String args[]) {
102
103     java.awt.EventQueue.invokeLater(new Runnable() {
104
105         @Override
106

```

```
107     public void run() {
108
109         new TelaSplash().setVisible(true);
110         new SieaDi().setVisible(false);
111     }
112 }
113
114 });
115 }
116
117 // Variables declaration - do not modify
118 private javax.swing.JProgressBar EfeitoVideo;
119 private javax.swing.JLabel jLabel1;
120 private javax.swing.JLabel jLabel2;
121 // End of variables declaration
122
123 }
124
```

Formulário Principal

```

1
2 package PRINCIPAL;
3
4
5 import javax.swing.JFrame;
6 /**
7  *
8  * @author Adriana Coimbra <dri8@hotmail.com>
9  */
10 public class SieaDi extends javax.swing.JFrame {
11     private static final long serialVersionUID = 1L;
12
13     public SieaDi() {
14         initComponents();
15         setResizable(false);
16         setLocationRelativeTo(null);
17     }
18
19     SieaDi(boolean b) {
20         throw new UnsupportedOperationException(""); //To change body of generated methods,
choose Tools | Templates.
21     }
22
23     @SuppressWarnings("unchecked")
24     // <editor-fold defaultstate="collapsed" desc="Generated Code">
25     private void initComponents() {
26
27         SisIlgPUEntityManager = java.beans.Beans.isDesignTime() ? null :
javax.persistence.Persistence.createEntityManagerFactory("SisIlgPU").createEntityManager();
28         congregacaoQuery = java.beans.Beans.isDesignTime() ? null :
SisIlgPUEntityManager.createQuery("SELECT c FROM Congregacao c");
29         congregacaoList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
congregacaoQuery.getResultList();
30         jToolBar2 = new javax.swing.JToolBar();
31         jSeparator2 = new javax.swing.JToolBar.Separator();
32         jLabel5 = new javax.swing.JLabel();
33         jSeparator1 = new javax.swing.JToolBar.Separator();
34         BarraProgresso = new javax.swing.JProgressBar();
35         jLabel2 = new javax.swing.JLabel();
36         jLabel1 = new javax.swing.JLabel();
37         jLabel3 = new javax.swing.JLabel();
38         Menu = new javax.swing.JMenuBar();
39         MenuArquivo = new javax.swing.JMenu();
40         jSeparator4 = new javax.swing.JPopupMenu.Separator();
41         SubMenuSair = new javax.swing.JMenuItem();
42         MenuCadastro = new javax.swing.JMenu();
43         MenuCadMembros = new javax.swing.JMenuItem();
44         MenuCadCong = new javax.swing.JMenuItem();
45         jMenu1 = new javax.swing.JMenu();
46         jMenuItem1 = new javax.swing.JMenuItem();
47         jMenuItem2 = new javax.swing.JMenuItem();
48         MenuAjuda = new javax.swing.JMenu();
49         SubMenuSobre = new javax.swing.JMenuItem();
50
51         setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

```

```

52 setTitle("SieaDi - Gerenciador de Igrejas");
53 setBackground(new java.awt.Color(204, 204, 255));
54 setName("Frente");
55
56 jToolBar2.setFloatable(false);
57 jToolBar2.setRollover(true);
58 jToolBar2.add(jSeparator2);
59
60 jLabel5.setText("    Status do Banco de dados: !\"CONECTADO!\"    ");
61
62 BarraProgresso.addAncestorListener(new javax.swing.event.AncestorListener() {
63     public void ancestorMoved(javax.swing.event.AncestorEvent evt) {
64     }
65     public void ancestorAdded(javax.swing.event.AncestorEvent evt) {
66         BarraProgressoAncestorAdded(evt);
67     }
68     public void ancestorRemoved(javax.swing.event.AncestorEvent evt) {
69     }
70 });
71
72 jLabel2.setText("Ver.: 0.0.0.1");
73
74 jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGENS/igreja.jpg"))); // NOI18N
75
76 jLabel3.setText("Carregando.");
77
78 Menu.setBackground(new java.awt.Color(204, 204, 204));
79
80 Menu.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, null, new java.awt.Color(204, 51, 255), null, null));
81
82 MenuArquivo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/Arquivo.png"))); // NOI18N
83 MenuArquivo.setMnemonic('A');
84 MenuArquivo.setText("Arquivo");
85 MenuArquivo.add(jSeparator4);
86
87 SubMenuSair.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_W,
java.awt.event.InputEvent.CTRL_MASK));
88 SubMenuSair.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/BOTAO SAIR.png"))); // NOI18N
89 SubMenuSair.setText("Sair");
90 SubMenuSair.addActionListener(new java.awt.event.ActionListener() {
91     public void actionPerformed(java.awt.event.ActionEvent evt) {
92         SubMenuSairActionPerformed(evt);
93     }
94 });
95 MenuArquivo.add(SubMenuSair);
96
97 Menu.add(MenuArquivo);
98
99 MenuCadastro.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/cadastrar.png"))); // NOI18N
100 MenuCadastro.setMnemonic('C');
    MenuCadastro.setText("Cadastros");

```

```

101
102
MenuCadMembros.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_
M, java.awt.event.InputEvent.CTRL_MASK));
103     MenuCadMembros.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/BOTAO PESSOAS.png"))); // NOI18N
104     MenuCadMembros.setText("Membros");
105     MenuCadMembros.addActionListener(new java.awt.event.ActionListener() {
106         public void actionPerformed(java.awt.event.ActionEvent evt) {
107             MenuCadMembrosActionPerformed(evt);
108         }
109     });
110     MenuCadastro.add(MenuCadMembros);
111
112
MenuCadCong.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_R,
java.awt.event.InputEvent.CTRL_MASK));
113     MenuCadCong.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/BOTAO CONGR..png"))); // NOI18N
114     MenuCadCong.setText("Congregação");
115     MenuCadCong.addActionListener(new java.awt.event.ActionListener() {
116         public void actionPerformed(java.awt.event.ActionEvent evt) {
117             MenuCadCongActionPerformed(evt);
118         }
119     });
120     MenuCadastro.add(MenuCadCong);
121
122     Menu.add(MenuCadastro);
123
124     jMenu1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/BOTAO
ABRIR.png"))); // NOI18N
125     jMenu1.setText("Consulta");
126
127
jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
128     jMenuItem1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/BOTAO CONSULTARM.png"))); //
NOI18N
129     jMenuItem1.setText("Consultar Membro");
130     jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
131         public void actionPerformed(java.awt.event.ActionEvent evt) {
132             jMenuItem1ActionPerformed(evt);
133         }
134     });
135     jMenu1.add(jMenuItem1);
136
137
jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_E,
java.awt.event.InputEvent.CTRL_MASK));
138     jMenuItem2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGEMS/BOTAO BUSCAR 3.png"))); // NOI18N
139     jMenuItem2.setText("Consultar Congregação");
140     jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
141         public void actionPerformed(java.awt.event.ActionEvent evt) {
142             jMenuItem2ActionPerformed(evt);
143     }

```

```

144     });
145     jMenu1.add(jMenuItem2);
146
147     Menu.add(jMenu1);
148
149     MenuAjuda.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGENS/Ajuda.png"))); // NOI18N
150     MenuAjuda.setMnemonic('j');
151     MenuAjuda.setText("Ajuda");
152
153     SubMenuSobre.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));
154     SubMenuSobre.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGENS/BOTAO SOBRE.png"))); // NOI18N
155     SubMenuSobre.setText("Sobre");
156     SubMenuSobre.addActionListener(new java.awt.event.ActionListener() {
157         public void actionPerformed(java.awt.event.ActionEvent evt) {
158             SubMenuSobreActionPerformed(evt);
159         }
160     });
161     MenuAjuda.add(SubMenuSobre);
162
163     Menu.add(MenuAjuda);
164
165     setJMenuBar(Menu);
166
167     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
168     getContentPane().setLayout(layout);
169     layout.setHorizontalGroup(
170         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
171             .addGroup(layout.createSequentialGroup()
172                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
173                     .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 723,
javax.swing.GroupLayout.PREFERRED_SIZE))
174                     .addGroup(layout.createSequentialGroup()
175                         .addComponent(jToolBar2, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
176                         .addGroup(layout.createSequentialGroup()
177                             .addComponent(jLabel2)
178                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
179                             .addComponent(jLabel5)
180                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
181                             .addComponent(jLabel3)
182                             .addGap(18, 18, 18)
183                             .addComponent(BarraProgresso,
javax.swing.GroupLayout.PREFERRED_SIZE, 224, javax.swing.GroupLayout.PREFERRED_SIZE)))
184                         .addGap(0, 0, Short.MAX_VALUE)))
185             .addContainerGap())
186 
```

```

190 );
191 layout.setVerticalGroup(
192     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
193     .addGroup(layout.createSequentialGroup()
194         .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 380,
195             javax.swing.GroupLayout.PREFERRED_SIZE)
196         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
197         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
198             .addGroup(layout.createSequentialGroup()
199                 .addComponent(jLabel2)
200                 .addComponent(jLabel5)
201                 .addComponent(jLabel3))
202             .addGap(480, 480, 480)
203             .addComponent(jToolBar2, javax.swing.GroupLayout.PREFERRED_SIZE,
204                 javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
205         .addComponent(BarraProgresso, javax.swing.GroupLayout.PREFERRED_SIZE,
206             javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
207 );
208 getAccessibleContext().setAccessibleName("SieaDi -SOFTWARE Igreja Assembléia de
209 Deus");
210 getAccessibleContext().setAccessibleDescription("");
211 java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
212 setBounds((screenSize.width-705)/2, (screenSize.height-469)/2, 705, 469);
213 } // </editor-fold>
214 private void MenuCadMembrosActionPerformed(java.awt.event.ActionEvent evt) {
215
216     // Chamar Formulario de Membros:
217     new CADASTROSMEMBROS.CadMembro().setVisible(true);
218     JFrame CadMembro = new JFrame();
219     CadMembro.setContentPane(new CADASTROSMEMBROS.CadMembro());
220     setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
221     CadMembro.pack();
222     CadMembro.setVisible(true);
223     CadMembro.setLocationRelativeTo(null);
224     CadMembro.setResizable(false);
225     CadMembro.setTitle("Cadastro de Membros");
226
227 }
228
229 private void SubMenuSobreActionPerformed(java.awt.event.ActionEvent evt) {
230
231     //Chamar Tela de Ajuda
232     Sobre tb = new Sobre(this, true);
233     setLocationRelativeTo(null);
234     tb.setVisible(true);
235 }
236
237 private void BarraProgressoAncestorAdded(javax.swing.event.AncestorEvent evt) {
238     // TODO add your handling code here:
239
240 }
241

```

```

242
243 private void MenuCadCongActionPerformed(java.awt.event.ActionEvent evt) {
244
245     this.setEnabled(false);
246
247     // Chamar Formulario de Congrega  o:
248     new CADAstroMEMBROS.CadCongregacao().setVisible(true);
249     JFrame CadCongregacao = new JFrame();
250     setLocationRelativeTo(null);
251     CadCongregacao.setContentPane(new CADAstroMEMBROS.CadCongregacao());
252     setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
253     CadCongregacao.pack();
254     CadCongregacao.setVisible(true);
255     CadCongregacao.setLocationRelativeTo( null );
256     CadCongregacao.setResizable(false);
257     CadCongregacao.setTitle("Cadastro de Membros");
258 }
259
260 private void SubMenuSairActionPerformed(java.awt.event.ActionEvent evt) {
261     // Fechar o Sistema:
262     System.exit(0);
263 }
264
265 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
266     new CADAstroMEMBROS.ConsultaMembro().setVisible(true);
267     JFrame ConsultaMembro = new JFrame();
268     setLocationRelativeTo(null);
269     ConsultaMembro.setContentPane(new CADAstroMEMBROS.ConsultaMembro());
270     setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
271     ConsultaMembro.pack();
272     ConsultaMembro.setVisible(true);
273     ConsultaMembro.setLocationRelativeTo( null );
274     ConsultaMembro.setResizable(false);
275     ConsultaMembro.setTitle("Consulta de Membros");
276 }
277
278 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
279     new CADAstroMEMBROS.ConsultarCong().setVisible(true);
280     JFrame ConsultarCong = new JFrame();
281     setLocationRelativeTo(null);
282     ConsultarCong.setContentPane(new CADAstroMEMBROS.ConsultarCong());
283     setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
284     ConsultarCong.pack();
285     ConsultarCong.setVisible(true);
286     ConsultarCong.setLocationRelativeTo( null );
287     ConsultarCong.setResizable(false);
288     ConsultarCong.setTitle("Consulta de Membros");
289 }
290
291
292 public static void main(String args[]) {
293
294     java.awt.EventQueue.invokeLater(new Runnable() {
295         public void run() {
296
297             new SieaDi().setVisible(true);
298         }
299     });

```

```

299     });
300 }
301 // Variables declaration - do not modify
302 private javax.swing.JProgressBar BarraProgresso;
303 private javax.swing.JMenuBar Menu;
304 private javax.swing.JMenu MenuAjuda;
305 private javax.swing.JMenu MenuArquivo;
306 private javax.swing.JMenuItem MenuCadCong;
307 private javax.swing.JMenuItem MenuCadMembros;
308 private javax.swing.JMenuItem MenuCadastro;
309 private javax.persistence.EntityManager SislPUEntityManager;
310 private javax.swing.JMenuItem SubMenuSair;
311 private javax.swing.JMenuItem SubMenuSobre;
312 private java.util.List<CADASTROSMEMBROS.Congregacao> congregacaoList;
313 private javax.persistence.Query congregacaoQuery;
314 private javax.swing.JLabel jLabel1;
315 private javax.swing.JLabel jLabel2;
316 private javax.swing.JLabel jLabel3;
317 private javax.swing.JLabel jLabel5;
318 private javax.swing.JMenu jMenu1;
319 private javax.swing.JMenuItem jMenuItem1;
320 private javax.swing.JMenuItem jMenuItem2;
321 private javax.swing.JToolBar.Separator jSeparator1;
322 private javax.swing.JToolBar.Separator jSeparator2;
323 private javax.swing.JPopupMenu.Separator jSeparator4;
324 private javax.swing.JToolBar jToolBar2;
325
326
327 void initApplication() {
328     throw new UnsupportedOperationException("Not yet implemented");
329 }
330 }
331

```

Formulários de Cadastro de Membros

```

1 package CADASTROSMEMBROS;
2
3 import CONEXAO.Conexao;
4 import java.awt.EventQueue;
5 import java.beans.Beans;
6 import java.io.File;
7 import java.io.IOException;
8 import java.sql.Connection;
9 import java.util.ArrayList;
10 import java.util.List;
11 import javax.persistence.RollbackException;
12 import javax.swing.JFrame;
13 import javax.swing.JPanel;
14 import net.sf.jasperreports.engine.*;
15 import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;
16 import net.sf.jasperreports.view.JasperViewer;
17 /**
18  *
19  * @author Adriana Coimbra <dri8@hotmail.com>
20  */
21 public class CadMembros extends JPanel {
22
23     public CadMembros() {
24         initComponents();
25         if (!Beans.isDesignTime()) {
26             entityManager.getTransaction().begin();
27         }
28     }
29
30     @SuppressWarnings("unchecked")
31     // <editor-fold defaultstate="collapsed" desc="Generated Code">
32     private void initComponents() {
33         bindingGroup = new org.jdesktop.beansbinding.BindingGroup();
34
35         entityManager = java.beans.Beans.isDesignTime() ? null :
javax.persistence.Persistence.createEntityManagerFactory("igrejaPU").createEntityManager();
36         query = java.beans.Beans.isDesignTime() ? null : entityManager.createQuery("SELECT m FROM
Membros m");
37         list = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
org.jdesktop.observablecollections.ObservableCollections.observableList(query.getResultList());
38         membrosQuery = java.beans.Beans.isDesignTime() ? null : entityManager.createQuery("SELECT m
FROM Membros m");
39         membrosList = java.beans.Beans.isDesignTime() ? java.util.Collections.emptyList() :
membrosQuery.getResultList();
40         jPanel1 = new javax.swing.JPanel();
41         datanascLabel = new javax.swing.JLabel();
42         enderecoLabel = new javax.swing.JLabel();
43         cidadeField = new javax.swing.JTextField();
44         numeroField = new javax.swing.JTextField();
45         telefoneLabel = new javax.swing.JLabel();
46         codLabel = new javax.swing.JLabel();
47         emailField = new javax.swing.JTextField();
48         nomeField = new javax.swing.JTextField();
49         enderecoField = new javax.swing.JTextField();
50         celularLabel = new javax.swing.JLabel();

```

```

51 cpfLabel = new javax.swing.JLabel();
52 emailLabel = new javax.swing.JLabel();
53 numeroLabel = new javax.swing.JLabel();
54 codField = new javax.swing.JTextField();
55 cidadeLabel = new javax.swing.JLabel();
56 idadeLabel = new javax.swing.JLabel();
57 nomeLabel = new javax.swing.JLabel();
58 uflabel = new javax.swing.JLabel();
59 bairroLabel = new javax.swing.JLabel();
60 bairroField = new javax.swing.JTextField();
61 cepLabel = new javax.swing.JLabel();
62 rgLabel = new javax.swing.JLabel();
63 rgField = new javax.swing.JTextField();
64 sexoLabel = new javax.swing.JLabel();
65 jComboBox1 = new javax.swing.JComboBox();
66 jFormattedTextField2 = new javax.swing.JFormattedTextField();
67 jFormattedTextField3 = new javax.swing.JFormattedTextField();
68 jFormattedTextField4 = new javax.swing.JFormattedTextField();
69 jFormattedTextField5 = new javax.swing.JFormattedTextField();
70 jComboBox3 = new javax.swing.JComboBox();
71 jFormattedTextField6 = new javax.swing.JFormattedTextField();
72 jComboBox8 = new javax.swing.JComboBox();
73 jPanel2 = new javax.swing.JPanel();
74 masterScrollPane = new javax.swing.JScrollPane();
75 masterTable = new javax.swing.JTable();
76 jPanel3 = new javax.swing.JPanel();
77 uffLabel = new javax.swing.JLabel();
78 databatismLabel = new javax.swing.JLabel();
79 escolaridadeLabel = new javax.swing.JLabel();
80 conjField = new javax.swing.JTextField();
81 nomemaeField = new javax.swing.JTextField();
82 localbatismoLabel = new javax.swing.JLabel();
83 cargoLabel = new javax.swing.JLabel();
84 profissaoField = new javax.swing.JTextField();
85 departamentoLabel = new javax.swing.JLabel();
86 nomepaiLabel = new javax.swing.JLabel();
87 nomepaiField = new javax.swing.JTextField();
88 conjLabel = new javax.swing.JLabel();
89 nomemaeLabel = new javax.swing.JLabel();
90 profissaoLabel = new javax.swing.JLabel();
91 jComboBox2 = new javax.swing.JComboBox();
92 jFormattedTextField1 = new javax.swing.JFormattedTextField();
93 jComboBox4 = new javax.swing.JComboBox();
94 jComboBox5 = new javax.swing.JComboBox();
95 jLabel2 = new javax.swing.JLabel();
96 jTextField1 = new javax.swing.JTextField();
97 jLabel3 = new javax.swing.JLabel();
98 jComboBox6 = new javax.swing.JComboBox();
99 jComboBox7 = new javax.swing.JComboBox();
100 localbatismoField = new javax.swing.JTextField();
101 jPanel4 = new javax.swing.JPanel();
102 BotãoNovoMembro = new javax.swing.JButton();
103 deleteButton = new javax.swing.JButton();
104 refreshButton = new javax.swing.JButton();
105 saveButton = new javax.swing.JButton();
106 jPanel5 = new javax.swing.JPanel();
107 jLabel1 = new javax.swing.JLabel();

```

```

108     PesquisarMembro = new javax.swing.JTextField();
109
110     FormListener formListener = new FormListener();
111
112     setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Cadastrar Membros",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Kristen ITC", 1, 16), new
java.awt.Color(0, 102, 204))); // NOI18N
113     setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
114
115     jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Dados Pessoais",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Kristen ITC", 1, 12), new
java.awt.Color(153, 0, 153))); // NOI18N
116
117     datanascLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
118     datanascLabel.setText("Data de Nascimento:");
119
120     enderecoLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
121     enderecoLabel.setText("Endereço:");
122
123     cidadeField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
124
125     org.jdesktop.beansbinding.Binding binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.cidade}"),
cidadeField, org.jdesktop.beansbinding.BeanProperty.create("text"));
126     binding.setSourceUnreadableValue(null);
127     bindingGroup.addBinding(binding);
128     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), cidadeField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
129     bindingGroup.addBinding(binding);
130
131     numeroField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
132
133     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.numero}"),
numeroField, org.jdesktop.beansbinding.BeanProperty.create("text"));
134     binding.setSourceUnreadableValue(null);
135     bindingGroup.addBinding(binding);
136     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), numeroField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
137     bindingGroup.addBinding(binding);
138
139     telefoneLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
140     telefoneLabel.setText("Telefone:");
141
142     codLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
143     codLabel.setText("Codigo:");
144
145     emailField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
146

```

```

147     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.email}"),
emailField, org.jdesktop.beansbinding.BeanProperty.create("text"));
148     binding.setSourceUnreadableValue(null);
149     bindingGroup.addBinding(binding);
150     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), emailField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
151     bindingGroup.addBinding(binding);
152
153     nomeField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
154
155     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.nome}"),
nomeField, org.jdesktop.beansbinding.BeanProperty.create("text"));
156     binding.setSourceUnreadableValue(null);
157     bindingGroup.addBinding(binding);
158     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), nomeField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
159     bindingGroup.addBinding(binding);
160
161     enderecoField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
162
163     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.endereco}"),
enderecoField, org.jdesktop.beansbinding.BeanProperty.create("text"));
164     binding.setSourceUnreadableValue(null);
165     bindingGroup.addBinding(binding);
166     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
enderecoField, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
167     bindingGroup.addBinding(binding);
168
169     celularLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
170     celularLabel.setText("Celular:");
171
172     cpfLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
173     cpfLabel.setText("CPF:");
174
175     emailLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
176     emailLabel.setText("E-mail:");
177
178     numeroLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
179     numeroLabel.setText("Nº:");
180
181     codField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
182
183     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy

```

```

.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.cod}"),
codField, org.jdesktop.beansbinding.BeanProperty.create("text"));
184     binding.setSourceUnreadableValue(null);
185     bindingGroup.addBinding(binding);
186     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), codField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
187     bindingGroup.addBinding(binding);
188
189     cidadeLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
190     cidadeLabel.setText("Cidade:");
191
192     idadeLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
193     idadeLabel.setText("Idade:");
194
195     nomeLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
196     nomeLabel.setText("Nome:");
197
198     ufLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
199     ufLabel.setText("UF:");
200
201     bairroLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
202     bairroLabel.setText("Bairro:");
203
204     bairroField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
205
206     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.bairro}"),
bairroField, org.jdesktop.beansbinding.BeanProperty.create("text"));
207     binding.setSourceUnreadableValue(null);
208     bindingGroup.addBinding(binding);
209     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), bairroField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
210     bindingGroup.addBinding(binding);
211
212     cepLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
213     cepLabel.setText("CEP:");
214
215     rgLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
216     rgLabel.setText("RG:");
217
218     rgField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
219
220     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.rg}"), rgField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
221     binding.setSourceUnreadableValue(null);
222     bindingGroup.addBinding(binding);
223     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), rgField,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));

```

```

224     bindingGroup.addBinding(binding);
225
226     sexoLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
227     sexoLabel.setText("Sexo:");
228
229     jComboBox1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
230     jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "AC\t", "AL\t",
"AP", "AM\t", "BA\t", "CE\t", "DF\t", "ES\t", "GO\t", "MA\t", "MT\t", "MS\t", "MG\t", "PA\t", "PB\t",
"PR\t", "PE\t", "PI\t", "RJ\t", "RN\t", "RS\t", "RO\t", "RR\t", "SC\t", "SP\t", "SE\t", "TO" }));
231
232     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.uf}"),
jComboBox1, org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
233     bindingGroup.addBinding(binding);
234     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox1, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
235     bindingGroup.addBinding(binding);
236
237     jFormattedTextField2.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.DateFormatter(java.text.DateFormat.getDateInstance(java.text.DateFormat.MEDIUM))));
238
239     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.datanasc}"),
jFormattedTextField2, org.jdesktop.beansbinding.BeanProperty.create("value"));
240     bindingGroup.addBinding(binding);
241     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jFormattedTextField2, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
242     bindingGroup.addBinding(binding);
243
244     try {
245         jFormattedTextField3.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.MaskFormatter("###.###-####")));
246     } catch (java.text.ParseException ex) {
247         ex.printStackTrace();
248     }
249     jFormattedTextField3.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
250
251     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.cep}"),
jFormattedTextField3, org.jdesktop.beansbinding.BeanProperty.create("value"));
252     bindingGroup.addBinding(binding);
253     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jFormattedTextField3, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
254     bindingGroup.addBinding(binding);
255
256     try {
257         jFormattedTextField4.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.MaskFormatter("###.###.###-####")));

```

```

258     } catch (java.text.ParseException ex) {
259         ex.printStackTrace();
260     }
261     jFormattedTextField4.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
262
263     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.cpf}"),
jFormattedTextField4, org.jdesktop.beansbinding.BeanProperty.create("value"));
264     bindingGroup.addBinding(binding);
265     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jFormattedTextField4, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
266     bindingGroup.addBinding(binding);
267
268     try {
269         jFormattedTextField5.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.MaskFormatter("(##)####-####")));
270     } catch (java.text.ParseException ex) {
271         ex.printStackTrace();
272     }
273     jFormattedTextField5.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
274
275     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.telefone}"),
jFormattedTextField5, org.jdesktop.beansbinding.BeanProperty.create("value"));
276     bindingGroup.addBinding(binding);
277     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jFormattedTextField5, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
278     bindingGroup.addBinding(binding);
279
280     jComboBox3.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
281     jComboBox3.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Feminino",
"Masculino" }));
282
283     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.sexo}"),
jComboBox3, org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
284     bindingGroup.addBinding(binding);
285     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox3, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
286     bindingGroup.addBinding(binding);
287
288     try {
289         jFormattedTextField6.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.MaskFormatter("(##)####-####")));
290     } catch (java.text.ParseException ex) {
291         ex.printStackTrace();
292     }
293     jFormattedTextField6.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

```

```

294
295     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.celular}"),
jFormattedTextField6, org.jdesktop.beansbinding.BeanProperty.create("value"));
296     bindingGroup.addBinding(binding);
297     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jFormattedTextField6, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
298     bindingGroup.addBinding(binding);
299
300     jComboBox8.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
301     jComboBox8.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "1", "2", "3", "4",
"5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24",
"25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43",
"44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59", "60", "61", "62",
"63", "64", "65", "66", "67", "68", "69", "70", "71", "72", "73", "74", "75", "76", "77", "78", "78", "80", "81",
"82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92", "93", "94", "95", "96", "97", "98", "99", "100", "
", " " }));
302
303     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.idade}"),
jComboBox8, org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
304     bindingGroup.addBinding(binding);
305     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox8, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
306     bindingGroup.addBinding(binding);
307
308     javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
309     jPanel1.setLayout(jPanel1Layout);
310     jPanel1Layout.setHorizontalGroup(
311         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
312             .addGroup(jPanel1Layout.createSequentialGroup()
313                 .addContainerGap()
314                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
315                     .addGroup(jPanel1Layout.createSequentialGroup()
316                         .addGroup(jPanel1Layout.createSequentialGroup()
317                             .addComponent(codLabel)
318                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
319                             .addComponent(codField, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
320                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
321                             .addComponent(nomeLabel)
322                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
323                             .addComponent(nomeField, javax.swing.GroupLayout.PREFERRED_SIZE, 196,
javax.swing.GroupLayout.PREFERRED_SIZE)
324                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
325                             .addComponent(enderecoLabel)
326                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
327                             .addComponent(enderecoField, javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)
328                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

329         .addComponent(numeroLabel)
330         .addGap(18, 18, 18)
331         .addComponent(numeroField, javax.swing.GroupLayout.PREFERRED_SIZE, 43,
javax.swing.GroupLayout.PREFERRED_SIZE))
332         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
333
334     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
335         .addGroup(jPanel1Layout.createSequentialGroup()
336             .addComponent(cidadeLabel)
337             .addGap(18, 18, 18)
338             .addComponent(cidadeField, javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE)
339             .addGap(18, 18, 18)
340             .addComponent(ufLabel)
341             .addGap(18, 18, 18)
342             .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
343             .addGap(18, 18, 18)
344             .addComponent(telefoneLabel)
345             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
346             .addComponent(jFormattedTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, 113, javax.swing.GroupLayout.PREFERRED_SIZE))
347         .addGroup(jPanel1Layout.createSequentialGroup()
348             .addGap(419, 419, 419)
349             .addComponent(idadeLabel)))
350
351     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
352         .addGroup(jPanel1Layout.createSequentialGroup()
353             .addComponent(celularLabel)
354             .addGap(10, 10, 10)
355             .addComponent(jFormattedTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE, 106, javax.swing.GroupLayout.PREFERRED_SIZE)
356             .addGap(18, 18, 18)
357             .addComponent(emailLabel))
358         .addGroup(jPanel1Layout.createSequentialGroup()
359             .addComponent(jComboBox8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
360             .addGap(18, 18, 18)
361             .addComponent(sexoLabel)
362             .addGap(18, 18, 18)
363             .addComponent(jComboBox3, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)))
364         .addGap(24, 24, 24)))
365     .addGroup(jPanel1Layout.createSequentialGroup()
366         .addComponent(cpfLabel)
367         .addGap(18, 18, 18)
368         .addComponent(jFormattedTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
369         .addGap(21, 21, 21)
370         .addComponent(datanascLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)
371         .addGap(18, 18, 18)
372         .addComponent(jFormattedTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 89,
javax.swing.GroupLayout.PREFERRED_SIZE)))

```

```

372         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
373         .addGroup(jPanel1Layout.createSequentialGroup()
374         .addGap(16, 16, 16)
375         .addComponent(bairroLabel)
376         .addGap(18, 18, 18)
377         .addComponent(bairroField, javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)
378         .addGap(18, 18, 18)
379         .addComponent(cepLabel)
380         .addGap(18, 18, 18)
381         .addComponent(jFormattedTextField3, javax.swing.GroupLayout.DEFAULT_SIZE, 81,
Short.MAX_VALUE))
382         .addGroup(jPanel1Layout.createSequentialGroup()
383         .addGap(18, 18, 18)
384         .addComponent(emailField)
385         .addGap(18, 18, 18)
386         .addComponent(rgLabel)
387         .addGap(18, 18, 18)
388         .addComponent(rgField, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)))
389         .addContainerGap()
390     );
391     jPanel1Layout.setVerticalGroup(
392     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
393     .addGroup(jPanel1Layout.createSequentialGroup()
394     .addContainerGap()
395     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
396     .addComponent(codLabel)
397     .addComponent(codField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
398     .addComponent(nomeLabel)
399     .addComponent(nomeField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
400     .addComponent(enderecoLabel)
401     .addComponent(enderecoField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
402     .addComponent(numeroLabel)
403     .addComponent(numeroField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
404     .addComponent(bairroLabel)
405     .addComponent(bairroField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
406     .addComponent(cepLabel)
407     .addComponent(jFormattedTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
408     .addGap(29, 29, 29)
409     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
410     .addComponent(cidadeLabel)
411     .addComponent(cidadeField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
412     .addComponent(ufLabel)
413     .addComponent(telefoneLabel)
414     .addComponent(celularLabel)
415     .addComponent(emailLabel)
416     .addComponent(emailField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
417     .addComponent(rgLabel)

```

```

418        .addComponent(rgField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
419        .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
420        .addComponent(jFormattedTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
421        .addComponent(jFormattedTextField6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
422        .addGap(34, 34, 34)
423        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
424            .addComponent(cpfLabel)
425            .addComponent(datanascLabel)
426            .addComponent(idadeLabel)
427            .addComponent(sexoLabel)
428            .addComponent(jFormattedTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
429            .addComponent(jFormattedTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
430            .addComponent(jComboBox3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
431            .addComponent(jComboBox8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
432        .addGap(269, 269, 269))
433    );
434
435    add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 230, 1060, 170));
436
437    jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Lista",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Kristen ITC", 1, 12), new
java.awt.Color(153, 0, 153))); // NOI18N
438
439    org.jdesktop.swingbinding.JTableBinding jTableBinding =
org.jdesktop.swingbinding.SwingBindings.createJTableBinding(org.jdesktop.beansbinding.AutoBinding.Update
Strategy.READ_WRITE, list, masterTable);
440    org.jdesktop.swingbinding.JTableBinding.ColumnBinding columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${cod}"));
441    columnBinding.setColumnName("Cod");
442    columnBinding.setColumnClass(Integer.class);
443    columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${nome}"));
444    columnBinding.setColumnName("Nome");
445    columnBinding.setColumnClass(String.class);
446    columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${endereco}"));
447    columnBinding.setColumnName("Endereco");
448    columnBinding.setColumnClass(String.class);
449    columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${numero}"));
450    columnBinding.setColumnName("Numero");
451    columnBinding.setColumnClass(Integer.class);
452    columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${bairro}"));
453    columnBinding.setColumnName("Bairro");
454    columnBinding.setColumnClass(String.class);
455    columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${cep}"));

```

```

456     columnBinding.setColumnName("Cep");
457     columnBinding.setColumnClass(String.class);
458     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${cidade}"));
459     columnBinding.setColumnName("Cidade");
460     columnBinding.setColumnClass(String.class);
461     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${cpf}"));
462     columnBinding.setColumnName("Cpf");
463     columnBinding.setColumnClass(String.class);
464     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${idade}"));
465     columnBinding.setColumnName("Idade");
466     columnBinding.setColumnClass(Short.class);
467     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${telefone}"));
468     columnBinding.setColumnName("Telefone");
469     columnBinding.setColumnClass(String.class);
470     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${celular}"));
471     columnBinding.setColumnName("Celular");
472     columnBinding.setColumnClass(String.class);
473     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${datanasc}"));
474     columnBinding.setColumnName("Datanasc");
475     columnBinding.setColumnClass(java.util.Date.class);
476     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${email}"));
477     columnBinding.setColumnName("Email");
478     columnBinding.setColumnClass(String.class);
479     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${escolaridade}"));
480     columnBinding.setColumnName("Escolaridade");
481     columnBinding.setColumnClass(String.class);
482     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${databatism}"));
483     columnBinding.setColumnName("Databatism");
484     columnBinding.setColumnClass(java.util.Date.class);
485     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${uff}"));
486     columnBinding.setColumnName("Uff");
487     columnBinding.setColumnClass(String.class);
488     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${rg}"));
489     columnBinding.setColumnName("Rg");
490     columnBinding.setColumnClass(String.class);
491     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${uf}"));
492     columnBinding.setColumnName("Uf");
493     columnBinding.setColumnClass(String.class);
494     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${congregacao}"));
495     columnBinding.setColumnName("Congregacao");
496     columnBinding.setColumnClass(String.class);
497     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${estadocivil}"));
498     columnBinding.setColumnName("Estadocivil");

```

```

499     columnBinding.setColumnClass(String.class);
500     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${sexo}"));
501     columnBinding.setColumnName("Sexo");
502     columnBinding.setColumnClass(String.class);
503     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${localbatismo}"));
504     columnBinding.setColumnName("Localbatismo");
505     columnBinding.setColumnClass(String.class);
506     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${departamento}"));
507     columnBinding.setColumnName("Departamento");
508     columnBinding.setColumnClass(String.class);
509     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${cargo}"));
510     columnBinding.setColumnName("Cargo");
511     columnBinding.setColumnClass(String.class);
512     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${conj}"));
513     columnBinding.setColumnName("Conj");
514     columnBinding.setColumnClass(String.class);
515     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${nomemae}"));
516     columnBinding.setColumnName("Nomemae");
517     columnBinding.setColumnClass(String.class);
518     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${nomepai}"));
519     columnBinding.setColumnName("Nomepai");
520     columnBinding.setColumnClass(String.class);
521     columnBinding =
jTableBinding.addColumnBinding(org.jdesktop.beansbinding.ELProperty.create("${profissao}"));
522     columnBinding.setColumnName("Profissao");
523     columnBinding.setColumnClass(String.class);
524     bindingGroup.addBinding(jTableBinding);
525     jTableBinding.bind();
526     masterScrollPane.setViewportView(masterTable);
527     masterTable.getColumnModel().getColumn(9).setResizable(false);
528     masterTable.getColumnModel().getColumn(10).setResizable(false);
529
530     javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
531     jPanel2.setLayout(jPanel2Layout);
532     jPanel2Layout.setHorizontalGroup(
533         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
534             .addComponent(masterScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 1048,
Short.MAX_VALUE)
535     );
536     jPanel2Layout.setVerticalGroup(
537         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
538             .addGroup(jPanel2Layout.createSequentialGroup()
539                 .addComponent(masterScrollPane, javax.swing.GroupLayout.PREFERRED_SIZE, 178,
javax.swing.GroupLayout.PREFERRED_SIZE)
540                 .addGap(0, 2, Short.MAX_VALUE))
541     );
542
543     add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 1060, 210));
544

```

```

545    jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Dados Ministeriais",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Kristen ITC", 1, 12), new
java.awt.Color(153, 0, 153))); // NOI18N
546
547    uffLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
548    uffLabel.setText("UF:");
549
550    databatismLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
551    databatismLabel.setText("Data de Batismo:");
552
553    escolaridadeLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
554    escolaridadeLabel.setText("Escolaridade:");
555
556    conjField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
557
558    binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.conj}"),
conjField, org.jdesktop.beansbinding.BeanProperty.create("text"));
559    binding.setSourceUnreadableValue(null);
560    bindingGroup.addBinding(binding);
561    binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
conjField, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
562    bindingGroup.addBinding(binding);
563
564    nomemaeField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
565
566    binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.nomemae}"),
nomemaeField, org.jdesktop.beansbinding.BeanProperty.create("text"));
567    binding.setSourceUnreadableValue(null);
568    bindingGroup.addBinding(binding);
569    binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
nomemaeField, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
570    bindingGroup.addBinding(binding);
571
572    localbatismoLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
573    localbatismoLabel.setText("Local de Batismo:");
574
575    cargoLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
576    cargoLabel.setText("Cargo:");
577
578    profissaoField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
579
580    binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.profissao}"),
profissaoField, org.jdesktop.beansbinding.BeanProperty.create("text"));
581    binding.setSourceUnreadableValue(null);
582    bindingGroup.addBinding(binding);

```

```

583     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
profissaoField, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
584     bindingGroup.addBinding(binding);
585
586     departamentoLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
587     departamentoLabel.setText("Departamento:");
588
589     nomepaiLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
590     nomepaiLabel.setText("Nome do Pai:");
591
592     nomepaiField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
593
594     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.nomepai}"),
nomepaiField, org.jdesktop.beansbinding.BeanProperty.create("text"));
595     binding.setSourceUnreadableValue(null);
596     bindingGroup.addBinding(binding);
597     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
nomepaiField, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
598     bindingGroup.addBinding(binding);
599
600     conjLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
601     conjLabel.setText("Conjuge:");
602
603     nomemaeLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
604     nomemaeLabel.setText("Nome da Mãe:");
605
606     profissaoLabel.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
607     profissaoLabel.setText("Profissao:");
608
609     jComboBox2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
610     jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "AC\t", "AL\t",
"AP\t", "AM\t", "BA\t", "CE\t", "DF\t", "ES\t", "GO\t", "MA\t", "MT\t", "MS\t", "MG\t", "PA\t", "PB\t",
"PR\t", "PE\t", "PI\t", "RJ\t", "RN\t", "RS\t", "RO\t", "RR\t", "SC\t", "SP\t", "SE\t", "TO" }));
611
612     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.uff}"),
jComboBox2, org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
613     bindingGroup.addBinding(binding);
614     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox2, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
615     bindingGroup.addBinding(binding);
616
617     jFormattedTextField1.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.DateFormatter()));
618
619     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy

```

```

.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.databatism}"),
jFormattedTextField1, org.jdesktop.beansbinding.BeanProperty.create("value"));
620     bindingGroup.addBinding(binding);
621     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jFormattedTextField1, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
622     bindingGroup.addBinding(binding);
623
624     jFormattedTextField1.addActionListener(formListener);
625
626     jComboBox4.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
627     jComboBox4.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "CEMINI",
"ASBAADI", "CEIA", "CÍRCULO DE ORAÇÃO", "COZINHA", "DAADI", "EBD", "INFANTIL", "MANHÃ
DA BENÇÃO", "MINISTÉRIO", "UMADI", " " }));
628
629     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.departamento}"), jComboBox4,
org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
630     bindingGroup.addBinding(binding);
631     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox4, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
632     bindingGroup.addBinding(binding);
633
634     jComboBox5.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
635     jComboBox5.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Pastor", "Pastor
Presidente", "Porteiro", "Presbítero", "Professor", "Secretário", "Tesoureiro", "Cantor", "Congregado",
"Coordenador(a)", "Diaconisa", "Diácono", "Dirigente", "Evangalista", "Missionária", "Músico", "Obreiro",
"Vice-Coordenador(a)", " " }));
636
637     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.cargo}"),
jComboBox5, org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
638     bindingGroup.addBinding(binding);
639     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox5, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
640     bindingGroup.addBinding(binding);
641
642     jLabel2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
643     jLabel2.setText("Congregação:");
644
645     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.congregacao}"), jTextField1,
org.jdesktop.beansbinding.BeanProperty.create("text"));
646     bindingGroup.addBinding(binding);
647     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy

```

```

.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jTextField1, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
648     bindingGroup.addBinding(binding);
649
650     jLabel3.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
651     jLabel3.setText("Estado Civil:");
652
653     jComboBox6.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
654     jComboBox6.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Casado",
"Solteiro", " " }));
655
656     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.estadocivil}"),
jComboBox6, org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
657     bindingGroup.addBinding(binding);
658     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox6, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
659     bindingGroup.addBinding(binding);
660
661     jComboBox7.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
662     jComboBox7.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Primário", "Ensino
Fundamental", "Ensino Médio", "Técnico", "Graduação", "Especialização", "Mestrado", "Doutorado", "PHD", "
" }));
663
664     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.escolaridade}"), jComboBox7,
org.jdesktop.beansbinding.BeanProperty.create("selectedItem"));
665     bindingGroup.addBinding(binding);
666     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
jComboBox7, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
667     bindingGroup.addBinding(binding);
668
669     localbatismoField.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
670
671     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ_WRITE, masterTable,
org.jdesktop.beansbinding.ELProperty.create("${selectedElement.localbatismo}"), localbatismoField,
org.jdesktop.beansbinding.BeanProperty.create("text"));
672     binding.setSourceUnreadableValue(null);
673     bindingGroup.addBinding(binding);
674     binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"),
localbatismoField, org.jdesktop.beansbinding.BeanProperty.create("enabled"));
675     bindingGroup.addBinding(binding);
676
677     javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
678     jPanel3.setLayout(jPanel3Layout);
679     jPanel3Layout.setHorizontalGroup(

```

```

680     jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
681     .addGroup(jPanel3Layout.createSequentialGroup())
682     .addContainerGap()
683     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
684     .addGroup(jPanel3Layout.createSequentialGroup())
685     .addComponent(databatismoLabel)
686     .addGap(18, 18, 18)
687     .addComponent(jFormattedTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE)
688     .addGap(18, 18, 18)
689     .addComponent(localbatismoLabel)
690     .addGap(18, 18, 18)
691     .addComponent(localbatismoField, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
692     .addGap(0, 0, Short.MAX_VALUE))
693     .addGroup(jPanel3Layout.createSequentialGroup())
694     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
695     .addGroup(jPanel3Layout.createSequentialGroup())
696     .addComponent(jLabel2)
697     .addGap(18, 18, 18)
698     .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 135,
javax.swing.GroupLayout.PREFERRED_SIZE)
699     .addGap(18, 18, 18)
700     .addComponent(departamentoLabel))
701     .addGroup(jPanel3Layout.createSequentialGroup())
702     .addComponent(nomemaeLabel)
703     .addGap(18, 18, 18)
704     .addComponent(nomemaeField, javax.swing.GroupLayout.PREFERRED_SIZE, 242,
javax.swing.GroupLayout.PREFERRED_SIZE)))
705     .addGap(8, 8, 8)
706     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
707     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
708     .addGap(10, 10, 10)
709     .addComponent(jLabel3)
710     .addGap(18, 18, 18)
711     .addComponent(jComboBox6, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)
712     .addGap(18, 18, 18)
713     .addComponent(conjLabel)
714     .addGap(18, 18, 18)
715     .addComponent(conjField, javax.swing.GroupLayout.PREFERRED_SIZE, 316,
javax.swing.GroupLayout.PREFERRED_SIZE)
716     .addGap(0, 0, Short.MAX_VALUE))
717     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
718     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
719     .addGroup(jPanel3Layout.createSequentialGroup()
720     .addGap(0, 0, Short.MAX_VALUE)
721     .addComponent(jComboBox4, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
722     .addGap(18, 18, 18)
723     .addComponent(cargoLabel)
724     .addGap(18, 18, 18)

```

```

725         .addComponent(jComboBox5, javax.swing.GroupLayout.PREFERRED_SIZE,
110, javax.swing.GroupLayout.PREFERRED_SIZE))
726         .addGroup(jPanel3Layout.createSequentialGroup())
727         .addGap(186, 186, 186)
728         .addComponent(uffLabel)
729         .addGap(18, 18, 18)
730         .addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
731         .addGap(0, 0, Short.MAX_VALUE)))
732         .addGap(18, 18, 18)
733
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
734         .addGroup(jPanel3Layout.createSequentialGroup()
735         .addComponent(nomepaiLabel)
736         .addGap(18, 18, 18)
737         .addComponent(nomepaiField))
738         .addGroup(jPanel3Layout.createSequentialGroup()
739         .addComponent(escolaridadeLabel)
740         .addGap(18, 18, 18)
741         .addComponent(jComboBox7, javax.swing.GroupLayout.PREFERRED_SIZE,
102, javax.swing.GroupLayout.PREFERRED_SIZE)
742         .addGap(18, 18, 18)
743         .addComponent(profissaoLabel)
744         .addGap(18, 18, 18)
745         .addComponent(profissaoField, javax.swing.GroupLayout.PREFERRED_SIZE,
114, javax.swing.GroupLayout.PREFERRED_SIZE))))))
746         .addGap(23, 23, 23))
747     );
748     jPanel3Layout.setVerticalGroup(
749     jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
750     .addGroup(jPanel3Layout.createSequentialGroup()
751     .addContainerGap()
752     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
753     .addComponent(departamentoLabel)
754     .addComponent(cargoLabel)
755     .addComponent(escolaridadeLabel)
756     .addComponent(jComboBox4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
757     .addComponent(jComboBox5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
758     .addComponent(profissaoLabel)
759     .addComponent(profissaoField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
760     .addComponent(jLabel2)
761     .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
762     .addComponent(jComboBox7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
763     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 24,
Short.MAX_VALUE)
764     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
765     .addComponent(databatismoLabel)
766     .addComponent(jFormattedTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
767     .addComponent(localbatismoLabel)
768     .addComponent(localbatismoField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

769         .addComponent(uffLabel)
770         .addComponent(jComboBox2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
771         .addComponent(nomepaiLabel)
772         .addComponent(nomepaiField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
773     .addGap(24, 24, 24)
774     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
775         .addComponent(nomemaeLabel)
776         .addComponent(nomemaeField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
777         .addComponent(jLabel3)
778         .addComponent(jComboBox6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
779         .addComponent(conjLabel)
780         .addComponent(conjField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
781     .addContainerGap())
782 );
783
784 add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 410, 1060, -1));
785
786 jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(""));
787
788 BotãoNovoMembro.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
789 BotãoNovoMembro.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGENS/BOTAO NOVO.png"))); // NOI18N
790 BotãoNovoMembro.setText("Novo");
791 BotãoNovoMembro.addActionListener(formListener);
792
793 deleteButton.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
794 deleteButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/IMAGENS/BOTAO
DELETAR.png"))); // NOI18N
795 deleteButton.setText("Excluir");
796
797 binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy
.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement != null}"), deleteButton,
org.jdesktop.beansbinding.BeanProperty.create("enabled"));
798 bindingGroup.addBinding(binding);
799
800 deleteButton.addActionListener(formListener);
801
802 refreshButton.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
803 refreshButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/IMAGENS/BOTAO
ALTERAR.png"))); // NOI18N
804 refreshButton.setText("Atualizar");
805 refreshButton.addActionListener(formListener);
806
807 saveButton.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
808 saveButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/IMAGENS/BOTAO
SALVAR.png"))); // NOI18N
809 saveButton.setText("Salvar");
810 saveButton.addActionListener(formListener);
811
812 javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
813 jPanel4.setLayout(jPanel4Layout);

```

```

814     jPanel4Layout.setHorizontalGroup(
815         jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
816             .addGroup(jPanel4Layout.createSequentialGroup()
817                 .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
818                 .addComponent(BotãoNovoMembro, javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
819             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
820             .addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)
821             .addGap(18, 18, 18)
822             .addComponent(refreshButton, javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)
823             .addGap(18, 18, 18)
824             .addComponent(saveButton, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
825             .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
826         );
827     jPanel4Layout.setVerticalGroup(
828         jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
829             .addGroup(jPanel4Layout.createSequentialGroup()
830                 .addComponent(BotãoNovoMembro, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 37, Short.MAX_VALUE)
831                 .addComponent(refreshButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
832                 .addComponent(saveButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
833                 .addComponent(deleteButton, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
834             );
835     add(jPanel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 590, 450, 50));
836
837     jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Pesquisar",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Kristen ITC", 1, 12), new
java.awt.Color(153, 0, 153))); // NOI18N
838
839     jLabel1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
840     jLabel1.setText("Digite o Nome:");
841
842     PesquisarMembro.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
843     PesquisarMembro.addActionListener(formListener);
844
845     javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
846     jPanel5.setLayout(jPanel5Layout);
847     jPanel5Layout.setHorizontalGroup(
848         jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
849             .addGroup(jPanel5Layout.createSequentialGroup()
850                 .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
851                 .addComponent(jLabel1)
852                 .addGap(18, 18, 18)
853                 .addComponent(PesquisarMembro, javax.swing.GroupLayout.PREFERRED_SIZE, 370,
javax.swing.GroupLayout.PREFERRED_SIZE)
854             .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
855         );
856     jPanel5Layout.setVerticalGroup(
857         jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
858             .addGroup(jPanel5Layout.createSequentialGroup()

```

```

859         .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
860         .addComponent(jLabel1)
861         .addComponent(PesquisarMembro, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
862         .addGap(0, 9, Short.MAX_VALUE))
863     );
864
865     add(jPanel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 580, 500, 60));
866
867     bindingGroup.bind();
868 }
869
870 // Code for dispatching events from components to event handlers.
871
872 private class FormListener implements java.awt.event.ActionListener, java.awt.event.KeyListener {
873     FormListener() {}
874     public void actionPerformed(java.awt.event.ActionEvent evt) {
875         if (evt.getSource() == jFormattedTextField1) {
876             CadMembros.this.jFormattedTextField1ActionPerformed(evt);
877         }
878         else if (evt.getSource() == BotãoNovoMembro) {
879             CadMembros.this.BotãoNovoMembroActionPerformed(evt);
880         }
881         else if (evt.getSource() == deleteButton) {
882             CadMembros.this.deleteButtonActionPerformed(evt);
883         }
884         else if (evt.getSource() == refreshButton) {
885             CadMembros.this.refreshButtonActionPerformed(evt);
886         }
887         else if (evt.getSource() == saveButton) {
888             CadMembros.this.saveButtonActionPerformed(evt);
889         }
890     }
891
892     public void keyPressed(java.awt.event.KeyEvent evt) {
893         if (evt.getSource() == PesquisarMembro) {
894             CadMembros.this.PesquisarMembroKeyPressed(evt);
895         }
896     }
897
898     public void keyReleased(java.awt.event.KeyEvent evt) {
899     }
900
901     public void keyTyped(java.awt.event.KeyEvent evt) {
902     }
903 } // </editor-fold>
904
905 @SuppressWarnings("unchecked")
906 private void refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
907     entityManager.getTransaction().rollback();
908     entityManager.getTransaction().begin();
909     java.util.Collection data = query.getResultList();
910     for (Object entity : data) {
911         entityManager.refresh(entity);
912     }
913     list.clear();
914     list.addAll(data);

```

```

915 }
916
917 private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {
918     int[] selected = masterTable.getSelectedRows();
919     List<CADASTROSMEMBROS.Membros> toRemove = new
ArrayList<CADASTROSMEMBROS.Membros>(selected.length);
920     for (int idx = 0; idx < selected.length; idx++) {
921         CADASTROSMEMBROS.Membros m =
list.get(masterTable.convertRowIndexToModel(selected[idx]));
922         toRemove.add(m);
923         entityManager.remove(m);
924     }
925     list.removeAll(toRemove);
926 }
927
928 private void BotãoNovoMembroActionPerformed(java.awt.event.ActionEvent evt) {
929     CADASTROSMEMBROS.Membros m = new CADASTROSMEMBROS.Membros();
930     entityManager.persist(m);
931     list.add(m);
932     int row = list.size() - 1;
933     masterTable.setRowSelectionInterval(row, row);
934     masterTable.scrollRectToVisible(masterTable.getCellRect(row, 0, true));
935 }
936
937 private void saveButtonActionPerformed(java.awt.event.ActionEvent evt) {
938     try {
939         entityManager.getTransaction().commit();
940         entityManager.getTransaction().begin();
941     } catch (RollbackException rex) {
942         rex.printStackTrace();
943         entityManager.getTransaction().begin();
944         List<CADASTROSMEMBROS.Membros> merged = new
ArrayList<CADASTROSMEMBROS.Membros>(list.size());
945         for (CADASTROSMEMBROS.Membros m : list) {
946             merged.add(entityManager.merge(m));
947         }
948         list.clear();
949         list.addAll(merged);
950     }
951 }
952
953 private void PesquisarMembroKeyPressed(java.awt.event.KeyEvent evt) {
954     String query1 = "Select m From Membros m where m.nome like :nome";
955     query = entityManager.createQuery(query1);
956     query.setParameter("nome", "%" + PesquisarMembro.getText() + "%");
957     list.clear();
958     @SuppressWarnings("unchecked")
959     boolean addAll = list.addAll(query.getResultList());
960 }
961
962 private void jFormattedTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
963
964 }
965
966 // Variables declaration - do not modify
967 private javax.swing.JButton BotãoNovoMembro;
968 private javax.swing.JTextField PesquisarMembro;

```

```

969 private javax.swing.JTextField bairroField;
970 private javax.swing.JLabel bairroLabel;
971 private javax.swing.JLabel cargoLabel;
972 private javax.swing.JLabel celularLabel;
973 private javax.swing.JLabel cepLabel;
974 private javax.swing.JTextField cidadeField;
975 private javax.swing.JLabel cidadeLabel;
976 private javax.swing.JTextField codField;
977 private javax.swing.JLabel codLabel;
978 private javax.swing.JTextField conjField;
979 private javax.swing.JLabel conjLabel;
980 private javax.swing.JLabel cpfLabel;
981 private javax.swing.JLabel databatismLabel;
982 private javax.swing.JLabel datanascLabel;
983 private javax.swing.JButton deleteButton;
984 private javax.swing.JLabel departamentoLabel;
985 private javax.swing.JTextField emailField;
986 private javax.swing.JLabel emailLabel;
987 private javax.swing.JTextField enderecoField;
988 private javax.swing.JLabel enderecoLabel;
989 private javax.persistence.EntityManager entityManager;
990 private javax.swing.JLabel escolaridadeLabel;
991 private javax.swing.JLabel idadeLabel;
992 private javax.swing.JComboBox jComboBox1;
993 private javax.swing.JComboBox jComboBox2;
994 private javax.swing.JComboBox jComboBox3;
995 private javax.swing.JComboBox jComboBox4;
996 private javax.swing.JComboBox jComboBox5;
997 private javax.swing.JComboBox jComboBox6;
998 private javax.swing.JComboBox jComboBox7;
999 private javax.swing.JComboBox jComboBox8;
1000 private javax.swing.JFormattedTextField jFormattedTextField1;
1001 private javax.swing.JFormattedTextField jFormattedTextField2;
1002 private javax.swing.JFormattedTextField jFormattedTextField3;
1003 private javax.swing.JFormattedTextField jFormattedTextField4;
1004 private javax.swing.JFormattedTextField jFormattedTextField5;
1005 private javax.swing.JFormattedTextField jFormattedTextField6;
1006 private javax.swing.JLabel jLabel1;
1007 private javax.swing.JLabel jLabel2;
1008 private javax.swing.JLabel jLabel3;
1009 private javax.swing.JPanel jPanel1;
1010 private javax.swing.JPanel jPanel2;
1011 private javax.swing.JPanel jPanel3;
1012 private javax.swing.JPanel jPanel4;
1013 private javax.swing.JPanel jPanel5;
1014 private javax.swing.JTextField jTextField1;
1015 private java.util.List<CADASTROSMEMBROS.Membros> list;
1016 private javax.swing.JTextField localbatismoField;
1017 private javax.swing.JLabel localbatismoLabel;
1018 private javax.swing.JScrollPane masterScrollPane;
1019 private javax.swing.JTable masterTable;
1020 private java.util.List<CADASTROSMEMBROS.Membros> membrosList;
1021 private javax.persistence.Query membrosQuery;
1022 private javax.swing.JTextField nomeField;
1023 private javax.swing.JLabel nomeLabel;
1024 private javax.swing.JTextField nomemaeField;
1025 private javax.swing.JLabel nomemaeLabel;

```

```

1026 private javax.swing.JTextField nomepaiField;
1027 private javax.swing.JLabel nomepaiLabel;
1028 private javax.swing.JTextField numeroField;
1029 private javax.swing.JLabel numeroLabel;
1030 private javax.swing.JTextField profissaoField;
1031 private javax.swing.JLabel profissaoLabel;
1032 private javax.persistence.Query query;
1033 private javax.swing.JButton refreshButton;
1034 private javax.swing.JTextField rgField;
1035 private javax.swing.JLabel rgLabel;
1036 private javax.swing.JButton saveButton;
1037 private javax.swing.JLabel sexoLabel;
1038 private javax.swing.JLabel telefoneLabel;
1039 private javax.swing.JLabel ufLabel;
1040 private javax.swing.JLabel uffLabel;
1041 private org.jdesktop.beansbinding.BindingGroup bindingGroup;
1042 // End of variables declaration
1043 public static void main(String[] args) {
1044
1045     EventQueue.invokeLater(new Runnable() {
1046
1047         public void run() {
1048             JFrame frame = new JFrame();
1049             frame.setContentPane(new CadMembros());
1050             frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
1051             frame.pack();
1052             frame.setVisible(true);
1053         }
1054     });
1055 }
1056 }
1057

```

Formulário de Cadastro de Congregação

```

1 pacote CADASTROSMEMBROS;
2
3 importação java.awt.EventQueue;
4 de importação java.beans.Beans;
5 import java.io.File;
6 import java.io.IOException;
7 import java.util.ArrayList;
8 import java.util.List;
9 de importação javax.persistence.RollbackException;
10 importação javax.swing.JFrame;
11 importação javax.swing.JPanel;
12 importação net.sf.jasperreports.engine *;
13 importação net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;
14 importação net.sf.jasperreports.view.JasperViewer;
15
16 / **
17 *
18 * @ author Adriana Coimbra <dri8@hotmail.com>
19 */
20 público classe CadCongregacao estende JPanel {
21
22     pública CadCongregacao () {
23         initComponents ();
24         se (! Beans.isDesignTime ()) {
25             entityManager .. getTransaction () começar ();
26         }
27     }
28
29     @ SuppressWarnings ( " unchecked " )
30     // <editor-fold defaultstate="collapsed" desc="Generated code">
31     privadas vazio initComponents () {
32         BindingGroup = new org . JDesktop . beansbinding . BindingGroup ();
33
34         entityManager = java . beans . Beans . isDesignTime () ? null : javax . persistence . Persistence .
createEntityManagerFactory ( " igrejaPU " ) . createEntityManager ();
35         query = java . beans . Beans . isDesignTime () ? null : entityManager . createQuery ( " SELECT c
DE Congregação c " );
36         list = java . beans . Beans . isDesignTime () ? java . util . Collections . emptyList () : org . jdesktop
. observablecollections . ObservableCollections . observableList ( query . getResultList () );
37         jPanel3 = new javax . swing . JPanel ();
38         jPanel1 = new javax . swing . JPanel ();
39         cepcongLabel = new javax . swing . JLabel ();
40         telefoneLabel = new javax . swing . JLabel ();
41         endcongField = new javax . swing . JTextField ();
42         mtcongField = new javax . swing . JTextField ();
43         dirigenteatualLabel = new javax . swing . JLabel ();
44         nomecongLabel = new javax . swing . JLabel ();
45         nucongField = new javax . swing . JTextField ();
46         ufLabel = new javax . swing . JLabel ();
47         nucongLabel = new javax . swing . JLabel ();
48         mtcongLabel = new javax . swing . JLabel ();
49         cidadeLabel = new javax . swing . JLabel ();
50         cidadeField = new javax . swing . JTextField ();
51         bairrocongField = new javax . swing . JTextField ();
52         nomecongField = new javax . swing . JTextField ();

```

```

53 bairrocongLabel = new javax.swing.JLabel();
54 dirigenteatualField = new javax.swing.JTextField();
55 endcongLabel = new javax.swing.JLabel();
56 jComboBox1 = new javax.swing.JComboBox();
57 jFormattedTextField1 = new javax.swing.JFormattedTextField();
58 jFormattedTextField2 = new javax.swing.JFormattedTextField();
59 jPanel2 = new javax.swing.JPanel();
60 refreshButton = new javax.swing.JButton();
61 deleteButton = new javax.swing.JButton();
62 saveButton = new javax.swing.JButton();
63 newButton = new javax.swing.JButton();
64 jPanel4 = new javax.swing.JPanel();
65 PesquisarCongregação = new javax.swing.JTextField();
66 jLabel1 = new javax.swing.JLabel();
67 jPanel5 = new javax.swing.JPanel();
68 masterScrollPane = new javax.swing.JScrollPane();
69 masterTable = new javax.swing.JTable();
70
71 FormListener formListener = new FormListener();
72
73 javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
74 jPanel3.setLayout(jPanel3Layout);
75 jPanel3Layout.setHorizontalGroup(
76     jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
77         .addGap(0, 100, Short.MAX_VALUE)
78 );
79 jPanel3Layout.setVerticalGroup(
80     jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
81         .addGap(0, 100, Short.MAX_VALUE)
82 );
83
84 setBorder(javax.swing.BorderFactory.createTitledBorder(nulo, "Cadastrar ITC", 1, 16)
, novo java.awt.Color(51, 0, 204)); // NOI18N
85 setLayout(novo org.netbeans.lib.awtextra.AbsoluteLayout());
86
87 jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(nulo, "Dados da ITC"
, 1, 12), novo java.awt.Color(153, 0, 153)); // NOI18N
88
89 cepcongLabel.setText("CEP:");
90
91 telefoneLabel.setFont(novo java.awt.Font("Tahoma", 0, 12)); // NOI18N
92 telefoneLabel.setText("Telefone:");
93
94 endcongField.setFont(novo java.awt.Font("Tahoma", 0, 12)); // NOI18N
95
96 org.jdesktop.beansbinding.Binding binding = org.jdesktop.beansbinding.Bindings
.createAutoBinding(org.jdesktop.beansbinding.AutoBinding.UpdateStrategy.READ_WRITE,
masterTable, org.jdesktop.beansbinding.ELProperty.create("${selectedElement.endcong}"),
endcongField, org.jdesktop.beansbinding.BeanProperty.create("text"));
97 binding.setSourceUnreadableValue(null);
98 bindingGroup.addBinding(binding);
99 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create(
"${selectedElement != null}"), endcongField, org.jdesktop.beansbinding.BeanProperty.create(
"enabled"));
100 bindingGroup.addBinding(binding);
101

```

```

102 mtcongField.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
103
104 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement.mtcong}" ), mtcongField, org.jdesktop.beansbinding.BeanProperty.create( "
text" ));
105 binding.setSourceUnreadableValue( null );
106 bindingGroup.addBinding( binding );
107 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create( "
${selectedElement != null}" ), mtcongField, org.jdesktop.beansbinding.BeanProperty.create( "enabled
" ));
108 bindingGroup.addBinding( binding );
109
110 dirigenteatualLabel.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
111 dirigenteatualLabel.setText( "Dirigente:");
112
113 nomecongLabel.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
114 nomecongLabel.setText( "Nome:");
115
116 nucongField.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
117
118 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement.nucong}" ), nucongField, org.jdesktop.beansbinding.BeanProperty.create( "
text" ));
119 binding.setSourceUnreadableValue( null );
120 bindingGroup.addBinding( binding );
121 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create( "
${selectedElement != null}" ), nucongField, org.jdesktop.beansbinding.BeanProperty.create( "enabled
" ));
122 bindingGroup.addBinding( binding );
123
124 ufLabel.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
125 ufLabel.setText( "UF:");
126
127 nucongLabel.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
128 nucongLabel.setText( "N°:");
129
130 mtcongLabel.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
131 mtcongLabel.setText( "Matricula:");
132
133 cidadeLabel.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
134 cidadeLabel.setText( "Cidade:");
135
136 cidadeField.setFont( novo java.awt.Font( "Tahoma", 0, 12)); // NOI18N
137
138 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement.cidade}" ), cidadeField, org.jdesktop.beansbinding.BeanProperty.create( "
text" ));
139 binding.setSourceUnreadableValue( null );
140 bindingGroup.addBinding( binding );
141 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.

```

```

create ( " ${selectedElement != null} " ), cidadeField, org.jdesktop.beansbinding.BeanProperty.create (
" enabled " ));
142 bindingGroup.addBinding ( binding );
143
144 bairrocongField.setFont ( novo java.awt.Font ( " Tahoma ", 0, 12 )); // NOI18N
145
146 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding ( org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create ( " ${selectedElement.bairrocong} " ), bairrocongField, org.jdesktop.beansbinding.BeanProperty.
create ( " text " ));
147 binding.setSourceUnreadableValue ( null );
148 bindingGroup.addBinding ( binding );
149 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding ( org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create ( "
${selectedElement != null} " ), bairrocongField, org.jdesktop.beansbinding.BeanProperty.create ( "
enabled " ));
150 bindingGroup.addBinding ( binding );
151
152 nomecongField.setFont ( novo java.awt.Font ( " Tahoma ", 0, 12 )); // NOI18N
153
154 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding ( org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create ( " ${selectedElement.nomecong} " ), nomecongField, org.jdesktop.beansbinding.BeanProperty.
create ( " text " ));
155 binding.setSourceUnreadableValue ( null );
156 bindingGroup.addBinding ( binding );
157 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding ( org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create ( "
${selectedElement != null} " ), nomecongField, org.jdesktop.beansbinding.BeanProperty.create ( "
enabled " ));
158 bindingGroup.addBinding ( binding );
159
160 bairrocongLabel.setFont ( novo java.awt.Font ( " Tahoma ", 0, 12 )); // NOI18N
161 bairrocongLabel.setText ( " Bairro: " );
162
163 dirigenteatualField.setFont ( novo java.awt.Font ( " Tahoma ", 0, 12 )); // NOI18N
164
165 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding ( org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create ( " ${selectedElement.dirigenteatual} " ), dirigenteatualField, org.jdesktop.beansbinding.
BeanProperty.create ( " text " ));
166 binding.setSourceUnreadableValue ( null );
167 bindingGroup.addBinding ( binding );
168 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding ( org.jdesktop.beansbinding
.AutoBinding.UpdateStrategy.READ, masterTable, org.jdesktop.beansbinding.ELProperty.create ( "
${selectedElement != null} " ), dirigenteatualField, org.jdesktop.beansbinding.BeanProperty.create ( "
enabled " ));
169 bindingGroup.addBinding ( binding );
170
171 endcongLabel.setFont ( novo java.awt.Font ( " Tahoma ", 0, 12 )); // NOI18N
172 endcongLabel.setText ( " endereco: " );
173
174 jComboBox1.setFont ( novo java.awt.Font ( " Tahoma ", 0, 12 )); // NOI18N
175 jComboBox1.setModel ( new javax.swing.DefaultComboBoxModel ( new String [ ] { " AC \t ",
" AL \t ", " AP \t ", " AM \t ", " BA \t ", " CE \t \t ", " DF \t ", " ES \t ", " GO \t ", " MA \t ", " MT \t ",
" MS \t ", " MG \t ", " PA \t ", " PB \t ", " PR \t ", " PE \t ", " PI \t ", " RJ \t ", " RN \t ", " RS \t ",
" RO \t ", " RR \t ", " SC \t ", " SP \t ", " SE \t ", " TO " } ) );

```

```

176
177 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement.uf}" ), jComboBox1, org.jdesktop.beansbinding.BeanProperty.create( "
selectedItem" ));
178 bindingGroup.addBinding( binding );
179 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement != null}" ), jComboBox1, org.jdesktop.beansbinding.BeanProperty.create
( "enabled" ));
180 bindingGroup.addBinding( binding );
181
182 try {
183 jFormattedTextField1.setFormatterFactory( new javax.swing.text.DefaultFormatterFactory(
new javax.swing.text.MaskFormatter( "###.###-###" )));
184 } catch ( java.text.ParseException ex ) {
185 ex.printStackTrace();
186 }
187
188 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement.cepcong}" ), jFormattedTextField1, org.jdesktop.beansbinding.
BeanProperty.create( "value" ));
189 bindingGroup.addBinding( binding );
190 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement != null}" ), jFormattedTextField1, org.jdesktop.beansbinding.BeanProperty
.create( "enabled" ));
191 bindingGroup.addBinding( binding );
192
193 try {
194 jFormattedTextField2.setFormatterFactory( new javax.swing.text.DefaultFormatterFactory(
new javax.swing.text.MaskFormatter( "(###)####-####" )));
195 } catch ( java.text.ParseException ex ) {
196 ex.printStackTrace();
197 }
198
199 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement.telefone}" ), jFormattedTextField2, org.jdesktop.beansbinding.BeanProperty
.create( "value" ));
200 bindingGroup.addBinding( binding );
201 binding = org.jdesktop.beansbinding.Bindings.createAutoBinding( org.jdesktop.beansbinding
. AutoBinding.UpdateStrategy.READ_WRITE, masterTable, org.jdesktop.beansbinding.ELProperty.
create( "${selectedElement != null}" ), jFormattedTextField2, org.jdesktop.beansbinding.BeanProperty
.create( "enabled" ));
202 bindingGroup.addBinding( binding );
203
204 javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout( jPanel1 );
205 jPanel1.setLayout( jPanel1Layout );
206 jPanel1Layout.setHorizontalGroup(
207 jPanel1Layout.createParallelGroup( javax.swing.GroupLayout.Alignment.LEADING )
208 .addGroup( jPanel1Layout.createSequentialGroup()
209 .addContainerGap( )
210 .addGroup( jPanel1Layout.createParallelGroup( javax.swing.GroupLayout.Alignment.
LEADING )
211 .addGroup( jPanel1Layout.createSequentialGroup( )

```

```

212         .addComponent ( bairrocongLabel )
213         .addGap ( 18 , 18 , 18 )
214         .addComponent ( bairrocongField , javax . swing . GroupLayout . PREFERRED_SIZE ,
120 , javax . swing . GroupLayout . PREFERRED_SIZE )
215         .addGap ( 0 , 0 , Short . MAX_VALUE ) )
216     .addGroup ( jPanel1Layout . createSequentialGroup ( )
217     .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment
. LEADING )
218         .addComponent ( mtcongLabel )
219         .addComponent ( dirigenteatualLabel ) )
220     .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment
. LEADING )
221     .addGroup ( javax . swing . GroupLayout . Alignment . TRAILING , jPanel1Layout .
createSequentialGroup ( )
222     .addGap ( 18 , 18 , 18 )
223     .addComponent ( dirigenteatualField , javax . swing . GroupLayout .
PREFERRED_SIZE , 216 , javax . swing . GroupLayout . PREFERRED_SIZE )
224     .addGap ( 0 , 0 , Short . MAX_VALUE ) )
225     .addGroup ( jPanel1Layout . createSequentialGroup ( )
226     .addGap ( 4 , 4 , 4 )
227     .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout .
Alignment . LEADING )
228         .addGroup ( jPanel1Layout . createSequentialGroup ( )
229         .addComponent ( mtcongField , javax . swing . GroupLayout .
PREFERRED_SIZE , 36 , javax . swing . GroupLayout . PREFERRED_SIZE )
230         .addGap ( 18 , 18 , 18 )
231         .addComponent ( nomecongLabel )
232         .addGap ( 18 , 18 , 18 )
233         .addComponent ( nomecongField , javax . swing . GroupLayout .
PREFERRED_SIZE , 181 , javax . swing . GroupLayout . PREFERRED_SIZE ) )
234     .addGroup ( jPanel1Layout . createSequentialGroup ( )
235     .addGap ( 133 , 133 , 133 )
236     .addComponent ( cepcongLabel )
237     .addGap ( 18 , 18 , 18 )
238     .addComponent ( jFormattedTextField1 , javax . swing . GroupLayout .
PREFERRED_SIZE , 103 , javax . swing . GroupLayout . PREFERRED_SIZE ) ) )
239     .addPreferredGap ( javax . swing . LayoutStyle . ComponentPlacement . RELATED )
240     .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout .
Alignment . LEADING )
241         .addGroup ( jPanel1Layout . createSequentialGroup ( )
242         .addGap ( 0 , 29 , Short . MAX_VALUE )
243         .addComponent ( endcongLabel )
244         .addGap ( 18 , 18 , 18 )
245         .addComponent ( endcongField , javax . swing . GroupLayout .
PREFERRED_SIZE , 250 , javax . swing . GroupLayout . PREFERRED_SIZE )
246         .addGap ( 18 , 18 , 18 )
247         .addComponent ( nucongLabel )
248         .addGap ( 18 , 18 , 18 )
249         .addComponent ( nucongField , javax . swing . GroupLayout .
PREFERRED_SIZE , 51 , javax . swing . GroupLayout . PREFERRED_SIZE ) )
250     .addGroup ( jPanel1Layout . createSequentialGroup ( )
251     .addComponent ( cidadeLabel )
252     .addGap ( 18 , 18 , 18 )
253     .addComponent ( cidadeField , javax . swing . GroupLayout .
PREFERRED_SIZE , 107 , javax . swing . GroupLayout . PREFERRED_SIZE )
254     .addGap ( 18 , 18 , 18 )
255     .addComponent ( ufLabel )

```

```

256         .addGap ( 18 , 18 , 18 )
257         .addComponent ( jComboBox1 , javax . swing . GroupLayout .
PREFERRED_SIZE , javax . swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout .
PREFERRED_SIZE )
258         .addGap ( 18 , 18 , 18 )
259         .addComponent ( telefoneLabel )
260         .addGap ( 18 , 18 , 18 )
261         .addComponent ( jFormattedTextField2 ) ) ) ) ) )
262     .addContainerGap ( ) )
263 );
264 jPanel1Layout . setVerticalGroup (
265     jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment . LEADING )
266     .addGroup ( jPanel1Layout . createSequentialGroup ( )
267         .addContainerGap ( )
268         .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment .
BASELINE )
269             .addComponent ( mtcongLabel )
270             .addComponent ( mtcongField , javax . swing . GroupLayout . PREFERRED_SIZE , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
271             .addComponent ( nomecongLabel )
272             .addComponent ( nomecongField , javax . swing . GroupLayout . PREFERRED_SIZE , javax
. swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
273             .addComponent ( endcongLabel )
274             .addComponent ( endcongField , javax . swing . GroupLayout . PREFERRED_SIZE , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
275             .addComponent ( nucongLabel )
276             .addComponent ( nucongField , javax . swing . GroupLayout . PREFERRED_SIZE , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE ) )
277         .addGap ( 27 , 27 , 27 )
278         .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment .
BASELINE )
279             .addComponent ( bairrocongLabel )
280             .addComponent ( bairrocongField , javax . swing . GroupLayout . PREFERRED_SIZE , javax
. swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
281             .addComponent ( cepcongLabel )
282             .addComponent ( cidadeLabel )
283             .addComponent ( cidadeField , javax . swing . GroupLayout . PREFERRED_SIZE , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
284             .addComponent ( ufLabel )
285             .addComponent ( jComboBox1 , javax . swing . GroupLayout . PREFERRED_SIZE , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
286             .addComponent ( telefoneLabel , javax . swing . GroupLayout . PREFERRED_SIZE , 21 ,
javax . swing . GroupLayout . PREFERRED_SIZE )
287             .addComponent ( jFormattedTextField1 , javax . swing . GroupLayout . PREFERRED_SIZE ,
javax . swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
288             .addComponent ( jFormattedTextField2 , javax . swing . GroupLayout . PREFERRED_SIZE ,
javax . swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE ) )
289         .addPreferredGap ( javax . swing . LayoutStyle . ComponentPlacement . RELATED , 20 , Short
. MAX_VALUE )
290         .addGroup ( jPanel1Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment .
BASELINE )
291             .addComponent ( dirigenteatualLabel )
292             .addComponent ( dirigenteatualField , javax . swing . GroupLayout . PREFERRED_SIZE ,
javax . swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE ) )
293         .addContainerGap ( ) )
294 );
295

```

```

296 add ( jPanel1 , novo org . netbeans . lib . awtextra . AbsoluteConstraints ( 16 , 251 , 840 , - 1 ) );
297
298 jPanel2 . setBorder ( javax . balanço . BorderFactory . createTitledBorder ( "" ) );
299
300 refreshButton . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
301 refreshButton . setIcon ( novo javax . balanço . ImageIcon ( getClass ( ) . getResource ( " / imagens /
Botão Change. png " ) ) ); // NOI18N
302 refreshButton . setText ( " Atualizar " );
303 refreshButton . addActionListener ( formListener );
304
305 DeleteButton . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
306 DeleteButton . setIcon ( novo javax . balanço . ImageIcon ( getClass ( ) . getResource ( " / imagens /
Botão DELETAR. png " ) ) ); // NOI18N
307 DeleteButton . setText ( " Excluir " );
308
309 binding = org . jdesktop . beansbinding . Bindings . createAutoBinding ( org . jdesktop . beansbinding
. AutoBinding . UpdateStrategy . READ , masterTable , org . jdesktop . beansbinding . ELProperty . create ( "
${selectedElement != null} " ) , deleteButton , org . jdesktop . beansbinding . BeanProperty . create ( " enabled
" ) );
310 bindingGroup . addBinding ( binding );
311
312 DeleteButton . addActionListener ( formListener );
313
314 SaveButton . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
315 SaveButton . setIcon ( novo javax . balanço . ImageIcon ( getClass ( ) . getResource ( " / imagens /
Botão SALVAR. png " ) ) ); // NOI18N
316 SaveButton . setText ( " SALVAR " );
317 SaveButton . addActionListener ( formListener );
318
319 newButton . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
320 newButton . setIcon ( novo javax . balanço . ImageIcon ( getClass ( ) . getResource ( " / imagens /
Botão NOVO. png " ) ) ); // NOI18N
321 newButton . setText ( " Novo " );
322 newButton . addActionListener ( formListener );
323
324 javax . swing . GroupLayout jPanel2Layout = new javax . swing . GroupLayout ( jPanel2 );
325 jPanel2 . setLayout ( jPanel2Layout );
326 jPanel2Layout . setHorizontalGroup (
327     jPanel2Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment . LEADING )
328     . addGroup ( jPanel2Layout . createSequentialGroup ( )
329         . addContainerGap ( 14 , Short . MAX_VALUE )
330         . addComponent ( newButton )
331         . addPreferredGap ( javax . swing . LayoutStyle . ComponentPlacement . UNRELATED )
332         . addComponent ( deleteButton )
333         . addGap ( 18 , 18 , 18 )
334         . addComponent ( refreshButton )
335         . addGap ( 18 , 18 , 18 )
336         . addComponent ( saveButton )
337         . addContainerGap ( )
338     );
339
340 jPanel2Layout . linkSize ( javax . swing . SwingConstants . HORIZONTAL , new java . awt .
Component [ ] { deleteButton , refreshButton } );
341
342 jPanel2Layout . setVerticalGroup (
343     jPanel2Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment . LEADING )

```

```

344         .addComponent ( newButton , javax . swing . GroupLayout . DEFAULT_SIZE , 41 , Short .
MAX_VALUE )
345         .addComponent ( deleteButton , javax . swing . GroupLayout . Alignment . TRAILING , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . DEFAULT_SIZE , Short .
MAX_VALUE )
346         .addComponent ( refreshButton , javax . swing . GroupLayout . DEFAULT_SIZE , javax . swing .
GroupLayout . DEFAULT_SIZE , Short . MAX_VALUE )
347         .addComponent ( saveButton , javax . swing . GroupLayout . Alignment . TRAILING , javax .
swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . DEFAULT_SIZE , Short .
MAX_VALUE )
348     );
349
350     add ( jPanel2 , novo org . netbeans . lib . awtextra . AbsoluteConstraints ( 410 , 420 , 440 , - 1 ) );
351
352     jPanel4 . setBorder ( javax . swing . BorderFactory . createTitledBorder ( null , " Pesquisar " , javax .
swing . border . TitledBorder . DEFAULT_JUSTIFICATION , javax . swing . border . TitledBorder .
DEFAULT_POSITION , new java . awt . Font ( " Kristen ITC " , 1 , 12 ) , novo java . awt . Cor ( 153 , 0 ,
153 ) ) ); // NOI18N
353
354     PesquisarCongregaçao . addKeyListener ( formListener );
355
356     jLabel1 . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
357     jLabel1 . setText ( " Digite o Nome: " );
358
359     javax . swing . GroupLayout jPanel4Layout = new javax . swing . GroupLayout ( jPanel4 );
360     jPanel4 . setLayout ( jPanel4Layout );
361     jPanel4Layout . setHorizontalGroup (
362         jPanel4Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment . LEADING )
363         . addGroup ( jPanel4Layout . createSequentialGroup ( )
364             . addContainerGap ( )
365             . addComponent ( jLabel1 )
366             . addPreferredGap ( javax . swing . LayoutStyle . ComponentPlacement . UNRELATED )
367             . addComponent ( PesquisarCongregaçao , javax . swing . GroupLayout . DEFAULT_SIZE , 256
, Short . MAX_VALUE )
368             . addContainerGap ( )
369         );
370     jPanel4Layout . setVerticalGroup (
371         jPanel4Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment . LEADING )
372         . addGroup ( jPanel4Layout . createParallelGroup ( javax . swing . GroupLayout . Alignment .
BASELINE )
373             . addComponent ( PesquisarCongregaçao , javax . swing . GroupLayout . PREFERRED_SIZE ,
javax . swing . GroupLayout . DEFAULT_SIZE , javax . swing . GroupLayout . PREFERRED_SIZE )
374             . addComponent ( jLabel1 )
375         );
376
377     add ( jPanel4 , novo org . netbeans . lib . awtextra . AbsoluteConstraints ( 20 , 420 , 384 , 60 ) );
378
379     jPanel5 . setBorder ( javax . swing . BorderFactory . createTitledBorder ( null , " Lista " , javax .
swing . border . TitledBorder . DEFAULT_JUSTIFICATION , javax . swing . border . TitledBorder .
DEFAULT_POSITION , new java . awt . Font ( " Kristen ITC " , 1 , 12 ) , novo java . awt . Cor ( 153 , 0 ,
153 ) ) ); // NOI18N
380
381     masterScrollPane . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
382
383     masterTable . setFont ( novo java . awt . Font ( " Tahoma " , 0 , 12 ) ); // NOI18N
384

```

```

385     org.jdesktop.swingbinding.JTableBinding jTableBinding = org.jdesktop.swingbinding .
SwingBindings.createJTableBinding ( org.jdesktop.beansbinding.AutoBinding.UpdateStrategy .
READ_WRITE , list , masterTable );
386     org.jdesktop.swingbinding.JTableBinding.ColumnBinding columnBinding = jTableBinding .
addColumnBinding ( org.jdesktop.beansbinding.ELProperty.create ( " ${mtcong} " ) );
387     columnBinding.setColumnName ( " Mt " );
388     columnBinding.setColumnClass ( Long.class );
389     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${nomecong} " ) );
390     columnBinding.setColumnName ( " Nome " );
391     columnBinding.setColumnClass ( String.class );
392     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${endcong} " ) );
393     columnBinding.setColumnName ( " Endereço " );
394     columnBinding.setColumnClass ( String.class );
395     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${nucong} " ) );
396     columnBinding.setColumnName ( " Nº " );
397     columnBinding.setColumnClass ( Integer.class );
398     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${bairrocong} " ) );
399     columnBinding.setColumnName ( " Bairro " );
400     columnBinding.setColumnClass ( String.class );
401     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${cepcong} " ) );
402     columnBinding.setColumnName ( " Cep " );
403     columnBinding.setColumnClass ( String.class );
404     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${cidade} " ) );
405     columnBinding.setColumnName ( " Cidade " );
406     columnBinding.setColumnClass ( String.class );
407     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${uf} " ) );
408     columnBinding.setColumnName ( " Uf " );
409     columnBinding.setColumnClass ( String.class );
410     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${dirigenteatual} " ) );
411     columnBinding.setColumnName ( " Dirigente " );
412     columnBinding.setColumnClass ( String.class );
413     columnBinding = jTableBinding.addColumnBinding ( org.jdesktop.beansbinding.ELProperty .
create ( " ${telefone} " ) );
414     columnBinding.setColumnName ( " Telefone " );
415     columnBinding.setColumnClass ( String.class );
416     bindingGroup.addBinding ( jTableBinding );
417     jTableBinding.bind ();
418     masterScrollPane.setViewportViewView ( masterTable );
419
420     javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout ( jPanel5 );
421     jPanel5.setLayout ( jPanel5Layout );
422     jPanel5Layout.setHorizontalGroup (
423         jPanel5Layout.createParallelGroup ( javax.swing.GroupLayout.Alignment.LEADING )
424         .addComponent ( masterScrollPane , javax.swing.GroupLayout.Alignment.TRAILING , javax
.swing.GroupLayout.DEFAULT_SIZE , 828 , Short.MAX_VALUE )
425     );
426     jPanel5Layout.setVerticalGroup (
427         jPanel5Layout.createParallelGroup ( javax.swing.GroupLayout.Alignment.LEADING )

```

```

428         .addGroup ( javax . swing . GroupLayout . Alignment . TRAILING , jPanel5Layout .
createSequentialGroup ( )
429         .addComponent ( masterScrollPane , javax . swing . GroupLayout . DEFAULT_SIZE , 170 ,
Short . MAX_VALUE )
430         .addContainerGap ( ) )
431     );
432
433     add ( jPanel5 , novo org . netbeans . lib . awtextra . AbsoluteConstraints ( 16 , 29 , 840 , - 1 ) );
434
435     BindingGroup . ligamento ( );
436 }
437
438 // Código para o envio de eventos a partir de componentes para manipuladores de eventos.
439
440 private class FormListener implements java . awt . event . ActionListener , java . awt . event .
KeyListener {
441     FormListener ( ) { }
442     public void actionPerformed ( java . awt . event . ActionEvent evt ) {
443         if ( evt . getSource ( ) == refreshButton ) {
444             CadCongregacao . this . refreshButtonActionPerformed ( evt );
445         }
446         else if ( evt . getSource ( ) == deleteButton ) {
447             CadCongregacao . this . deleteButtonActionPerformed ( evt );
448         }
449         else if ( evt . getSource ( ) == saveButton ) {
450             CadCongregacao . this . saveButtonActionPerformed ( evt );
451         }
452         else if ( evt . getSource ( ) == newButton ) {
453             CadCongregacao . this . newButtonActionPerformed ( evt );
454         }
455     }
456
457     public void keyPressed ( java . awt . event . KeyEvent evt ) {
458         if ( evt . getSource ( ) == PesquisarCongregação ) {
459             CadCongregacao . this . PesquisarCongregaçãoKeyPressed ( evt );
460         }
461     }
462
463     público vazio keyReleased ( java . awt . evento . KeyEvent evt ) {
464     }
465
466     público vazio keyTyped ( java . awt . evento . KeyEvent evt ) {
467     }
468 } // </ editor-fold>
469
470
471
472 @ SuppressWarnings ( " unchecked " )
473 private void refreshButtonActionPerformed ( java . awt . event . ActionEvent evt ) {
474     entityManager . getTransaction().rollback();
475     entityManager .. getTransaction () começar ();
476     dados java.util.Collection = consulta getResultList ();
477     para ( entidade objeto: dados ) {
478         entityManager . atualizar ( entidade );
479     }
480     lista clear ();
481     lista . addAll ( dados );

```

```

482 }
483
484 privada vazio deleteButtonActionPerformed ( java . awt . evento . ActionEvent evt ) {
485     int [] = selecionados masterTable.getSelectedRows ();
486     Lista <CADASTROSMEMBROS.Congregacao> toremove = new ArrayList <> (selected. comprimento );
487     para ( int idx = 0; idx <selecionado. comprimento ; idx ++ ) {
488         CADASTROSMEMBROS.Congregacao c = lista chegar. ( masterTable.convertRowIndexToModel
489             (selecionado [idx]).);
489         toRemove.add (c);
490         entityManager . remover (c);
491     }
492     lista . removeAll (toremove);
493 }
494
495 privada vazio newButtonActionPerformed ( java . awt . evento . ActionEvent evt ) {
496     CADASTROSMEMBROS.Congregacao c = new CADASTROSMEMBROS.Congregacao ();
497     entityManager . persistir (c);
498     lista . adicionar (c);
499     int row = lista size () - 1.;
500     masterTable . setRowSelectionInterval (row, row);
501     masterTable . scrollRectToVisible ( masterTable.getCellRect (linha, 0,. verdadeiro ));
502 }
503
504 privada vazio saveButtonActionPerformed ( java . awt . evento . ActionEvent evt ) {
505     try {
506         entityManager .. getTransaction () commit ();
507         entityManager .. getTransaction () começar ();
508     } catch (rex RollbackException) {
509         rex.printStackTrace ();
510         entityManager .. getTransaction () começar ();
511     }
512     Lista <CADASTROSMEMBROS.Congregacao> fundiu = new ArrayList
513     <CADASTROSMEMBROS.Congregacao> ( lista size ().);
514     para (CADASTROSMEMBROS.Congregacao c: lista ) {
515         merged.add ( entityManager fundir (c).);
516     }
517     lista clear ().;
518     lista . addAll (mesclados);
519 }
520
521 privada vazio PesquisarCongregaçãoKeyPressed ( java . awt . evento . KeyEvent evt ) {
522     Cadeia query1 = " Selecione c Da Congregação c onde c.nomecong gosto: nomecong ";
523     query = entityManager createQuery (query1).;
524     consultas . setParameter ( " nomecong " , " % " + PesquisarCongregação getText () +. " % " );
525     lista clear ().;
526     @ SuppressWarnings ( " unchecked " )
527     boolean addAll = lista addAll ( . consulta getResultList ().);
528 }
529 // declaração de variáveis - não Fim da declaração de variáveis
530 públicas estáticas vazio principais (String [] args) {
531
532     EventQueue. invokeLater ( new Runnable () {
533         público vazio run () {
534             JFrame frame = new JFrame ();
535             frame.setContentPane ( novo CadCongregacao ());

```

```
575 frame.setDefaultCloseOperation (JFrame. EXIT_ON_CLOSE );
576 frame.pack ();
577 frame.setVisible ( verdadeiro );
578         // Este Comando centralização o JFrame na Tela
579 frame.setLocationRelativeTo ( nulo );
580 frame.setTitle ( " Cadastro de Congregação " );
581}
582});
583}
584}
585
```

Formulários de Ajuda

```

1
2 package PRINCIPAL;
3
4 import javax.swing.JFrame;
5 /**
6  *
7  * @author Adriana Coimbra <dri8@hotmail.com>
8  */
9 public class Sobre extends javax.swing.JDialog {
10
11     public Sobre(java.awt.Frame parent, boolean modal) {
12
13         super(parent, modal);
14
15         initComponents();
16         setLocationRelativeTo(null);
17     }
18
19     Sobre() {
20         throw new UnsupportedOperationException("");
21     }
22
23     @SuppressWarnings("unchecked")
24     // <editor-fold defaultstate="collapsed" desc="Generated Code">
25     private void initComponents() {
26
27         jLabel2 = new javax.swing.JLabel();
28         jLabel3 = new javax.swing.JLabel();
29         jLabel4 = new javax.swing.JLabel();
30         jLabel1 = new javax.swing.JLabel();
31         jLabel5 = new javax.swing.JLabel();
32         BotãoFecharSobre = new javax.swing.JButton();
33         jLabel6 = new javax.swing.JLabel();
34
35         setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
36         setTitle("Sobre");
37         setName("Sobre");
38         setResizable(false);
39
40         jLabel2.setFont(new java.awt.Font("Tahoma", 3, 11)); // NOI18N
41         jLabel2.setText("SieaDi-Software Igreja");
42
43         jLabel3.setBackground(new java.awt.Color(153, 0, 153));
44         jLabel3.setFont(new java.awt.Font("Kristen ITC", 1, 12)); // NOI18N
45         jLabel3.setForeground(new java.awt.Color(153, 0, 153));
46         jLabel3.setText("Konourx- Engenharia de Software.");
47
48         jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/IMAGENS/JAVA.png"))); // NOI18N
49
50         jLabel1.setText("Autores: Adriana Coimbra");
51
52         jLabel5.setText("Versão 1.0 ");
53

```

```

54     BotãoFecharSobre.setIcon(new
javafx.swing.ImageIcon(getClass().getResource("/IMAGENS/BOTAO DELETAR.png"))); // NOI18N
55     BotãoFecharSobre.setText("Fechar");
56     BotãoFecharSobre.addActionListener(new java.awt.event.ActionListener() {
57         public void actionPerformed(java.awt.event.ActionEvent evt) {
58             BotãoFecharSobreActionPerformed(evt);
59         }
60     });
61
62     jLabel6.setText("Tiales Aguiar");
63
64     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
65     getContentPane().setLayout(layout);
66     layout.setHorizontalGroup(
67         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
68             .addGroup(layout.createSequentialGroup()
69                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
70                     .addGroup(layout.createSequentialGroup()
71                         .add(ContainerGap())
72                         .add(Component(jLabel3))
73                         .addGroup(layout.createSequentialGroup()
74                             .add(Group(layout.createSequentialGroup()
75                                 .add(Component(jLabel2)))
76                                 .addGap(0, 0, Short.MAX_VALUE))
77                             .add(Group(layout.createSequentialGroup()
78                                 .add(Group(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
79                                     .add(Group(layout.createSequentialGroup()
80                                         .add(ContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
81                                         .add(Component(BotãoFecharSobre))
82                                         .addGap(18, 18, 18))
83                                         .add(Component(jLabel5))
84                                         .add(Group(layout.createSequentialGroup()
85                                             .add(Component(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 119,
86                                                 javax.swing.GroupLayout.PREFERRED_SIZE))
87                                             .add(PreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
88                                             .add(Group(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
89                                                 .addGroup(layout.createSequentialGroup()
90                                                     .add(Component(jLabel1))
91                                                     .addGap(4, 4, 4))
92                                                     .add(Group(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
93                                                         .add(layout.createSequentialGroup()
94                                                             .add(Component(jLabel6))
95                                                             .addGap(21, 21, 21))))))
96                                                         .add(ContainerGap())
97                                                         .add(layout.createSequentialGroup()
98                                                             .add(layout.createParallelGroup(
99                                                                 javax.swing.GroupLayout.Alignment.LEADING)
100                                                                 .add(PreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
101                                                                 .add(Component(jLabel2))
102                                                                 .add(Group(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
103                                                                     .add(Group(layout.createSequentialGroup()
104                                                                         .addGap(41, 41, 41))
105                                                                         .add(Component(jLabel1))
106                                                                         .add(PreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))

```

```

107         .addComponent(jLabel6)
108         .addGap(78, 98, Short.MAX_VALUE))
109         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
110         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
111         .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
112         .addGap(18, 18, 18)))
113         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
114         .addComponent(BotãoFecharSobre)
115         .addComponent(jLabel5))
116         .addContainerGap()
117     );
118
119     pack();
120 } // </editor-fold>
121
122 private void BotãoFecharSobreActionPerformed(java.awt.event.ActionEvent evt) {
123
124     this.hide();
125
126 }
127
128
129 public static void main(String args[]) {
130
131     java.awt.EventQueue.invokeLater(new Runnable() {
132         @Override
133         public void run() {
134             Sobre dialog = new Sobre(new javax.swing.JFrame(), true);
135             dialog.addWindowListener(new java.awt.event.WindowAdapter() {
136
137                 @Override
138                 public void windowClosing(java.awt.event.WindowEvent e) {
139                     System.exit(0);
140                 }
141             });
142             dialog.setVisible(true);
143         }
144     });
145 }
146 // Variables declaration - do not modify
147 private javax.swing.JButton BotãoFecharSobre;
148 private javax.swing.JLabel jLabel1;
149 private javax.swing.JLabel jLabel2;
150 private javax.swing.JLabel jLabel3;
151 private javax.swing.JLabel jLabel4;
152 private javax.swing.JLabel jLabel5;
153 private javax.swing.JLabel jLabel6;
154 // End of variables declaration
155 }
156

```