

**UNIVERSIDADE FEDERAL DO ACRE CENTRO DE CIÊNCIAS
EXATAS E TECNOLÓGICAS
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**SISTEMA DE CONTROLE DE VISITANTES DE
PRESÍDIOS “SCVP”**

Rio Branco

2018

GÉLIO MARCOS VITAL DE SOUZA
RITA DA COSTA PINTO

SISTEMA DE CONTROLE DE VISITANTES DE PRESÍDIOS “SCVP”

Relatório de estágio apresentado como exigência parcial para obtenção do grau de bacharel em Sistemas de Informação da Universidade Federal do Acre.

Prof. Orientador: Jean Gonzaga Souza de Oliveira, M.Sc.

Rio Branco

2018

TERMO DE APROVAÇÃO

GÉLIO MARCOS VITAL DE SOUZA

RITA DA COSTA PINTO

SISTEMA DE CONTROLE DE VISITANTES DE PRESIDIOS “SCVP”

Este relatório de estágio supervisionado foi apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Acre, sendo aprovado pela banca constituída pelo professor orientador e membro abaixo mencionado.

Compuseram a banca:

Prof. Orientador Jean Gonzaga Souza de Oliveira, M.Sc.

Curso de Bacharelado em Sistemas de Informação

Prof. Wilker Luis Gadelha Maia, M.Sc

Curso de Bacharelado em Sistemas de Informação

“Dedicamos este trabalho aos colegas do curso de Sistemas de Informação, companheiros neste e em outros desafios ao longo de nossas vidas acadêmicas. “

AGRADECIMENTOS

Agradecemos em especial aos nossos familiares por todo o apoio para chegarmos até aqui.

Ao Instituto de Administração Penitenciário do Acre (IAPEN/AC) e a Coordenadoria de Tecnologia da Informação e Comunicação, sob supervisão do coordenador responsável, Leandro Ruy Batista da Silva.

À todos os professores do curso de Sistemas de Informação, em especial ao professor Jean Gonzaga pela orientação no estágio e à professora Laura Sarkis pela paciência que teve, a grande ajuda nas dicas tanto no Estágio Supervisionado quanto no Relatório de Estágio.

Aos colegas do curso de Sistemas de Informação, que muitos contribuíram com ajuda preciosa.

“Uma pessoa inteligente aprende com os seus erros, uma pessoa sábia aprende com os erros dos outros. ” (Augusto Cury)

RESUMO

O presente relatório tem por objetivo apresentar as atividades realizadas durante a disciplina de Estágio Supervisionado do curso de Sistemas de Informação da Universidade Federal do Acre, que realizado na autarquia Instituto de Administração Penitenciário do Acre (IAPEN/AC). São descritos os conceitos e métodos de análises, técnicas e ferramentas de desenvolvimento de software, para a criação de um sistema de cadastro de visitantes dos presídios, afim de facilitar e aprimorar o controle de fluxo de visitantes dos reeducandos das unidades prisionais. O sistema foi desenvolvido utilizando o SGBD Mysql, programação em banco de dados com o SQL para fazer o controle e cadastro do sistema, a ferramenta Delphi, para criação de telas, exibição de dados, geração de relatórios e definição de layouts. Todos os conceitos, métodos e ferramentas utilizados para o desenvolvimento desta aplicação, serão descritos ao decorrer do relatório.

Palavras-chave: Engenharia de Software. MySql. Delphi7

ABSTRACT

This report aims to present activities carried out during the course of Supervised Internship of the Information Systems course of the Federal University of Acre (UFAC), it was in the company Instituto de Administração Penitenciário do Acre (IAPEN/AC). The concepts and methods of analysis, techniques and software development tools are described for the creation of a registry system of visitors to prisons, in order to facilitate and improve the control of the flow of visitors of the reeducados of the prison units. The system was developed using the MySql SGBD, database programming with SQL to make the control and registry of the system, the Delphi7 tool, for creating screens, displaying data, generating reports and defining layouts. All the concepts, methods and tools used to develop this application will be described in the course of the report.

Key-words: Software Engineering. MySql. Delphi7

LISTA DE FIGURAS

FIGURA 2.1 – Modelo Cascata	19
FIGURA 2.2 – Modelo Espiral	20
FIGURA 2.3 – Modelo Concorrente	21
FIGURA 2.4 – O paradigma da prototipação	23
FIGURA 2.5 – Ferramenta MySQL Workbench	24
FIGURA 2.6 – Exemplo de Diagrama de Classe	27
FIGURA 2.7 – Diagrama de Caso de uso	28
FIGURA 2.8 – Tela do ASTAH	29
FIGURA 2.9 – o ambiente Delphi7	30
FIGURA 3.1 – Sede do Instituto do Administrativo no Acre IAPEN	34
FIGURA 3.2 – Organograma do IAPEN	35
FIGURA 3.3 – Diagrama de Caso de uso	37
FIGURA 3.4 – Exemplo de Diagrama de Classe	38
FIGURA 3.5 – Tela de login	39
FIGURA 3.6 – protótipo tela principal	40
FIGURA 3.7 – protótipo tela de cadastro	41
FIGURA 3.8 – Código comunicação com Banco de Dados	41
FIGURA 3.9 – função validar locação do usuário	42
FIGURA 3.10– classe do formulário de login	43
FIGURA 3.11– Código código nível de usuário	44
FIGURA 3.12– Código código nível de usuário	44
FIGURA 3.13– Código código nível de usuário	45
FIGURA 3.14– Código de Cadastro de Usuário	45

LISTA DE TABELAS

TABELA 3.1 – Tabela quadro IAPEN	32
--	----

Sumário

1	INTRODUÇÃO	p. 12
1.1	PROBLEMA DA PESQUISA	p. 13
1.2	OBJETIVOS DO ESTÁGIO	p. 13
1.2.1	OBJETIVOS GERAL	p. 13
1.2.2	OBJETIVOS ESPECÍFICOS	p. 13
1.3	JUSTIFICATIVA DA PESQUISA	p. 14
1.4	METODOLOGIA	p. 14
1.5	ORGANIZAÇÃO DO ESTUDO	p. 15
2	FUNDAMENTAÇÃO TEÓRICA	p. 16
2.1	ENGENHARIA DE SOFTWARE	p. 16
2.1.1	Processos de Software	p. 17
2.2	ENGENHARIA DOS REQUISITOS	p. 21
2.2.1	ENTREVISTA	p. 22
2.3	PROTOTIPAÇÃO	p. 22
2.4	MYSQL	p. 23
2.4.1	MySQL Workbench	p. 24
2.5	BANDO DE DADOS	p. 24
2.6	UML	p. 26
2.6.1	Diagrama de classe	p. 27
2.6.2	Diagrama de Casos de Uso	p. 28
2.7	ASTAH	p. 29

2.8	DELPHI7	p. 29
3	ESTÁGIO SUPERVISIONADO	p. 31
3.1	EMPRESA	p. 31
3.1.1	Monitoramento eletrônico	p. 33
3.1.2	Reinserção pelo trabalho	p. 33
3.1.3	ESTRUTURA ORGANIZACIONAL	p. 35
3.2	ATIVIDADES DESENVOLVIDAS	p. 35
3.2.1	IDENTIFICAÇÃO DO PROBLEMA	p. 35
3.2.2	LEVANTAMENTO DOS REQUISITOS	p. 36
3.2.3	MODELAGEM DO SISTEMA	p. 36
3.2.4	PROTOTIPAÇÃO	p. 39
3.2.5	Implementação da Solução em Delphi7	p. 41
3.2.6	Testes e Validação	p. 45
3.3	RESULTADOS OBTIDOS	p. 46
3.4	CONCLUSÃO	p. 46
4	CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES	p. 47
4.1	CONSIDERAÇÕES FINAIS	p. 47
4.2	RECOMENDAÇÕES	p. 48
	Referências	p. 49

1 INTRODUÇÃO

O presente relatório tem por objetivo exibir as atividades desenvolvidas no Instituto IAPEN, em cumprimento à matéria de estágio supervisionado, do curso de Sistemas de Informação da Universidade Federal do Acre (UFAC). O IAPEN é uma instituição estatal e tem suas principais atividades na administração dos presídios estaduais do estado do Estado do Acre.

O projeto tem como objetivo desenvolver uma aplicação que substitua o existente, pois o mesmo não apresenta os requisitos necessários ao usuário. O novo sistema contará com o controle dos visitantes dos presídios, históricos de entrada e saída de cada visitante, e dos reeducandos monitorados, como impressão de carteiras com código de barras assim, torna-se mais ágil e fácil o trabalho das equipes de administração dos presídios.

O sistema existente, feito na plataforma web, apresenta falhas em regras de controle dos visitantes, funcionalidades desnecessárias, exige muito trabalho manual da parte do usuário e apresenta lentidão no carregamento dos dados. No fluxo atual, muitas vezes o usuário deve fazer o controle desses visitantes em planilhas excel, posteriormente, limitar o número de visitante por reeducandos e enviar via e-mail para cada unidade prisional, devido a lentidão em criar diariamente essas planilhas a mesma não apresenta controle e atualização da previsão quando sua origem sofre alguma alteração. Ao cadastrar um novo reeducando o usuário deve informá-la manualmente na planilha. Outro problema apresentado, se um visitante for penalizado pego em flagrante durante sua visita uma nova planilha deveria ser feita relatando cada detalhe do flagrante para assim sua carteira de visitante ser anulada. Com base nos pontos apresentados acima, foi proposto desenvolvimento de um sistema desktop que atenda todas as necessidades dos usuários, utilizando a ferramenta Delphi7.

O novo sistema de controle de visitantes permitirá que os usuários façam consultas mais rápidas, com todos os detalhes necessários e de forma totalmente automatizada. A partir do momento em que o usuário fizer o cadastro do visitante, o sistema criará, automaticamente, suas carteiras de visitas com código de barras, e o usuário fará vínculo desse visitante com o reeducando posteriormente estipulando o tipo de vínculo.

O sistema apresentará funcionalidades novas para diferenciação de “histórico de visita”, edição facilitada de dados, além de um relatório gerados em excel de conferência para usuários administrativos.

1.1 PROBLEMA DA PESQUISA

O problema consiste no fato de todo o processo ser feito de forma manual e por esta razão não proporcionar nenhuma confiabilidade no controle dos visitantes e dos reeducando. Uma das principais queixas do cliente é a falta de controle dos reeducandos que estavam sendo monitorados, pois tudo era feito em planilhas e qualquer um podia ter acesso e muitas vezes os dados eram facilmente alterados. Um bom exemplo disso era o grande número de carteiras falsas de visitantes. Isso gerava muita confusão com os coordenadores dos presídios.

Considerando o cenário descrito, surge o seguinte questionamento: Como automatizar o controle de visitantes e reeducandos, garantindo maior agilidade e confiabilidade ao IAPEN?

1.2 OBJETIVOS DO ESTÁGIO

O objetivo geral da pesquisa bem como seus objetivos específicos será apresentado nesta seção.

1.2.1 OBJETIVOS GERAL

Desenvolver um sistema para o cadastro e controle dos reeducados e seus visitantes para o Instituto de Administração Penitenciário do Acre (IAPEN).

1.2.2 OBJETIVOS ESPECÍFICOS

- Elaborar estudo sobre Sistemas de Controle Visitantes e Reeducandos;
- Fazer o levantamento de requisitos do sistema;
- Gerar protótipos para o cliente;
- Codificar o sistema;
- Criar uma base de dados;

- Emitir relatórios de: todos os visitantes, reeducandos, quantidade de visitas, data de vencimento das carteiras;
- Implantar o sistema no ambiente de trabalho do cliente;

1.3 JUSTIFICATIVA DA PESQUISA

Atualmente sistemas de computação estão sendo cada vez mais introduzidos nas organizações, devido a quantidade de processos e aumento na demanda que cada vez mais são introduzidos nas empresas. O surgimento dos computadores e seu poder de processamento cada vez maior, juntamente com o crescente aumento no número de pessoas, torna-se indispensável o uso de diferentes tipos de sistemas de computadores para realizar as tarefas que antigamente eram realizadas de forma manual.

As subseções a seguir apresentam os conceitos necessários para realizar as etapas de produção do Sistema de Controle de Visitantes de Presídio.

1.4 METODOLOGIA

A metodologia proposta consistiu em executar algumas atividades distribuídas nas seguintes etapas:

1. Na primeira etapa do desenvolvimento do sistema foi realizado um levantamento de requisitos, para identificar todas as necessidades do cliente através de entrevistas com a equipe TI do IAPEN. Após a identificação dos requisitos foi elaborado um documento de requisitos.
2. Na segunda etapa foi a modelagem utilizando UML para mostrar ao cliente.
3. Na terceira etapa a construção de um sistema que atendesse as necessidades do cliente.
4. Na quarta etapa foram o teste de validação do sistema com o cliente para saber se atendia suas necessidades.

Nesta seção, são abordadas as tecnologias, ferramentas e técnicas que foram utilizadas para o desenvolvimento das atividades de análise e desenvolvimento para atendimento deste estágio. O desenvolvimento da parte estrutural do sistema foi feito utilizando a tecnologia Delphi7 e a implementação foram feitas via procedimentos em banco de dados utilizando a ferramenta MySql workbench, versão 6.3, e os dados armazenados no banco de dados MYSQL. Nos próximos tópicos serão abordadas todas as técnicas e tecnologias que auxiliaram no desenvolvimento do software; elicitação de requisitos, prototipação e desenvolvimento, respectivamente. O diagrama de casos de uso do presente sistema foi criado utilizando a ferramenta Astah Community. Ainda antes de começar o desenvolvimento, de fato, foi feita a prototipação do sistema para ilustrar um modelo próximo à realidade. Foi utilizada a ferramenta Balsamiq Mockups.

1.5 ORGANIZAÇÃO DO ESTUDO

O específico estudo está organizado em quatro capítulos.

1. Na primeira seção deste relatório são apresentadas desde tecnologias, embasamento teórico, metodologia de desenvolvimento, conceito de prototipação de software, de bancos de dados e de programação.
2. Na segunda seção serão apresentadas matérias, técnicas e métodos utilizados para o desenvolvimento da aplicação. Serão apresentadas as ferramentas de desenvolvimento e suas linguagens, bem como as de modelagem e prototipação.
3. Na terceira seção serão apresentados os resultados obtidos na análise e desenvolvimento da aplicação, seguidos pela descrição de dificuldades encontradas no projeto e conclusões finais.

Por fim, serão apresentados os artefatos criados para desenvolvimento do projeto e as telas do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta capítulo serão mostrados os conceitos que ajudaram na elaboração e realização do estágio. Serão abordados os conceitos de processo de criação de software como também a elucidação dos requisitos, os fundamentos da verificação e validação dos artefatos, comunicação com o banco de dados MySql, ainda, a linguagem de programação de Object Pascal usando a ferramenta Delphi7 utilizando seus recursos para comunicação com banco de dados MySQL.

2.1 ENGENHARIA DE SOFTWARE

Engenharia de software é uma área do conhecimento da computação voltada para a especificação, desenvolvimento e manutenção de sistemas de software aplicando tecnologias e práticas de gerência de projetos, objetivando organização, produtividade e qualidade. Ela possui três fases, segundo Pressman (1995 p. 46), definição, desenvolvimento e manutenção. Todo o desenvolvimento estaria envolvido dentro dessas três grandes fases.

A engenharia de software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (SOMMERVILLE, 2003).

PRESSMAN (1995) trata a engenharia de software como uma “tecnologia em camadas”. Qualquer que seja a iniciativa de engenharia de software deve ser apoiada por um compromisso com a qualidade. Acima da camada da qualidade encontram-se os processos, logo acima, os métodos e, acima destes, as ferramentas. Ao decorrer da história da engenharia de software foram sendo construídas ferramentas computadorizadas para apoiar o desenvolvimento. Essas iniciativas avançaram bastante, mas ainda assim necessitam da intervenção humana. Foram concebidos vários modelos de processos de software e nenhum pode ser considerado o ideal, devido às suas divergências. Entretanto, para SOMMERVILLE (2003), todos compartilham de atividades fundamentais como especificação, projeto e implementação, validação e evolução.

2.1.1 Processos de Software

Um processo de software pode ser entendido como um conjunto de atividades exigidas para desenvolver um sistema de software. A utilização de um processo de software é algo primordial para o sucesso e evolução de um software. PAULA FILHO (2003) descreve um processo como um conjunto de passos ordenados, constituídos por atividades, métodos, práticas e transformações, usado para atingir uma meta. Essas atividades são, em sua maioria, executadas por desenvolvedores sistemas. Existe 4 atividades fundamentais no processo de software: 1. Especificação do Software – definição de requisitos e análise de requisitos - a funcionalidade do software e as restrições em sua operação devem ser definidas.

2. Desenvolvimento do Software – projeto e implementação - o software deve ser desenvolvido de modo que atenda a suas especificações.

3. Validação do software – integração e teste - o software tem de ser validado para que possa ter a garantia que ele faz o que o cliente deseja.

4. Evolução do software – o software deve evoluir para atender às necessidades mutáveis do cliente.

Conforme SOMMERVILLE (2003), um processo de software é definido como um conjunto de atividades realizadas, e resultados associados que conduzem à produção de um produto de software. Cada processo de software é complexo e cada um depende do julgamento humano como em qualquer processo intelectual. Por isso, existe uma grande diversidade de processos de software, nenhum ideal, desenvolvidos de maneiras diferentes por cada organização de acordo com suas necessidades.

Entretanto, todo processo de software existe atividades fundamentais comuns:

1. Especificação do software – são as funcionalidades e as restrições de operação do software são definidas:
 - Engenharia de Sistema: estabelecimento de uma solução geral para o problema, envolvendo questões extra-software;
 - Análise de Requisitos: é o levantamento das necessidades do software a ser implementado. Essa Análise tem por objetivo produzir uma especificação de requisitos, onde convencionalmente é um documento;
 - Especificação de Sistema: seria a descrição funcional do sistema. Se desejado pode incluir um plano de testes para verificar adequação.

2. Projeto e implementação do software - o software é desenvolvido e codificado de acordo com cada especificação:

- Projeto Arquitetural: é criado um modelo conceitual para o sistema, composto de módulos mais ou menos independentes;
- Projeto de Interface: é nesse caso que cada módulo tem sua interface de comunicação estudada e definida;
- Projeto Detalhado: é onde os módulos em si são identificados e, possivelmente, traduzidos para pseudo-código;
- Codificação: é a implementação em si do sistema em uma linguagem de computador.

3. Validação do software – onde o software é avaliado para verificar se está em conformidade com as necessidades do cliente:

- Teste de Unidade e Módulo: é feita a realização de testes para verificar a presença de erros e comportamento adequado das funções e módulos básicos do sistema;
- Integração: a reunião dos diferentes módulos em um produto de software homogêneo, e a verificação da interação entre estes quando operando em conjunto.

4. Evolução do software - o software é modificado para atender as novas exigências do cliente:

- Nesta fase, o software de uma forma geral entra em um ciclo iterativo que abrange todas as fases anteriores.

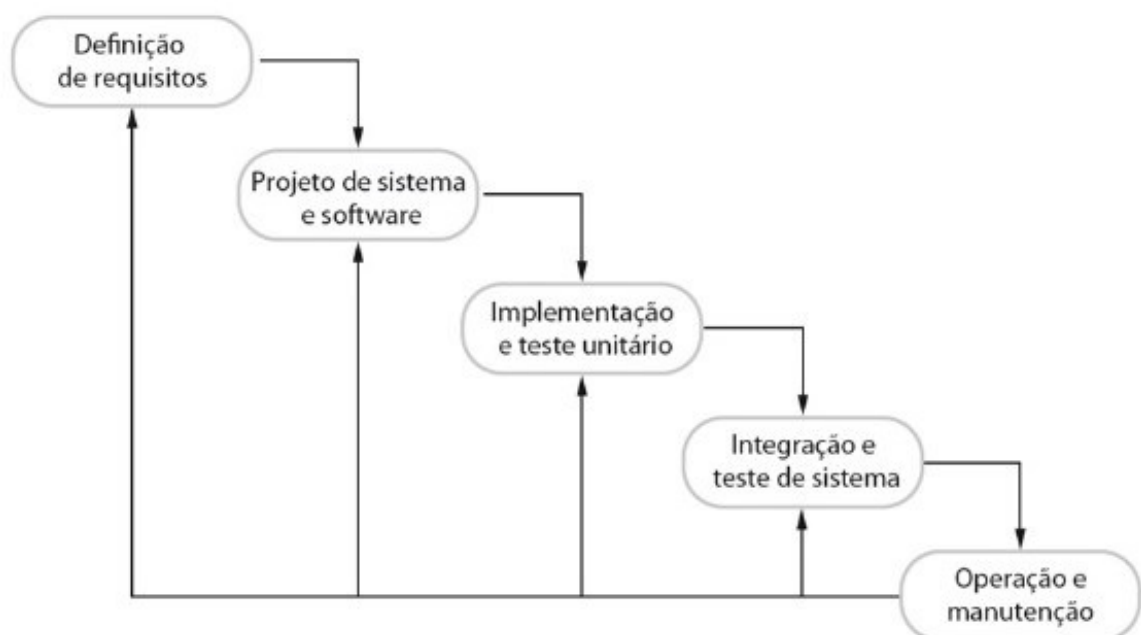
Nessas atividades é que se definem o que é chamamos de “ciclo de vida” do software. A escolha da abordagem e da forma de desenvolvimento de cada atividade é uma importante tarefa designada ao engenheiro de software e depende de sua estratégia de desenvolvimento. Usando outras palavras, o modelo é uma filosofia do andamento das fases, e não uma descrição de como cada atividade deve ser executada.

De acordo com SOMMERVILLE (2003) e PRESSMAN (1995), existem alguns modelos de processo de software. Os mais conhecidos são:

Modelo sequencial linear ou modelo em cascata:

Este modelo foi idealizado em 1970 e tem como característica principal a sequencialidade das atividades: sugere um tratamento ordenado e sistemático ao desenvolvimento do software. Cada fase transcorre completamente e seus produtos são vistos como entrada para a nova fase; o software é criado em um longo processo e entregue ao final deste. O autor sugere laços de feedback, que permitem realimentar fases anteriores do processo, de uma forma geral ao modelo cascata é considerada um modelo linear. Críticas ao modelo Cascata sugerem a inadequação deste a processos reais; em geral, há muito intercâmbio de informações entre as fases, e raramente ocorrem projetos onde não há concorrência das fases em si. Além disso, o modelo Cascata não leva em consideração questões modernas importantes ao desenvolvimento: prototipação, aquisição de software e alterações constantes nos requisitos, a figura 2.1 demonstra o modelo cascata.

Figura 2.1: Modelo Cascata



Fonte: Sommerville (2007, p. 44)

Modelo espiral

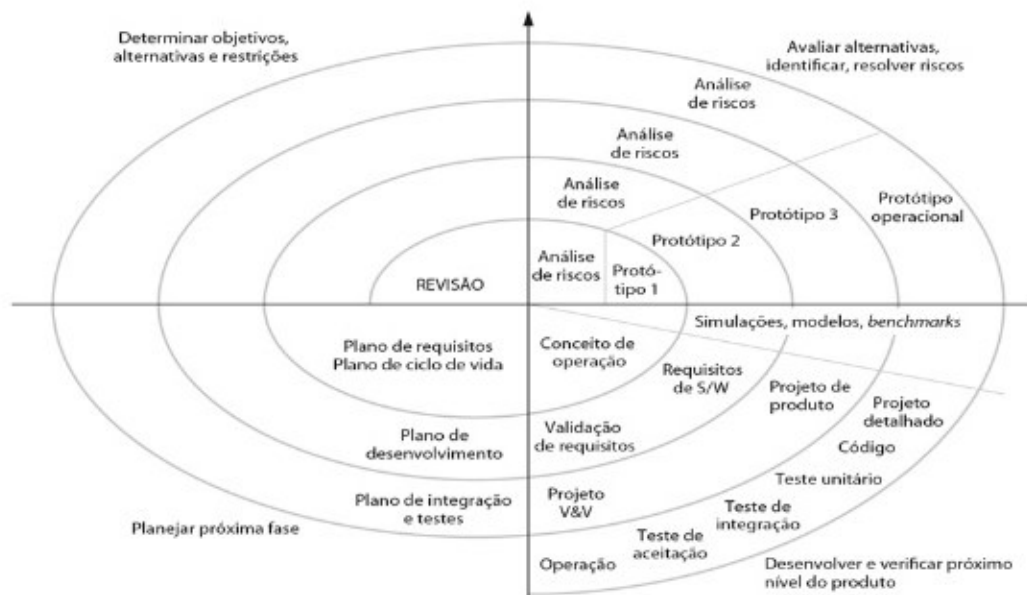
O modelo espiral é uma forma elaborada do modelo cascata, introduzido por um artigo publicado na IEEE Computer em maio de 1988. Foi sugerido um modelo evolucionário para o desenvolvimento de software, baseado em uma sequência de fases que culminam em versões incrementais do software. O modelo espiral define quatro importantes atividades:

- Determinação dos objetivos: definição do que será desenvolvido, restrições impostas à aplicação, tais como desempenho, funcionalidade, capacidade de acomodar mudanças, meios alternativos de implementação;
- Análise de risco: análise das alternativas e identificação/resolução dos riscos. Uma vez avaliados os riscos, podem-se construir protótipos para verificar se estes são realmente robustos para servir de base para a evolução futura do sistema;
- Desenvolvimento: detalhe do projeto, codificação, integração;
- Planejamento e próxima iteração: avaliação dos resultados pelo cliente, entrega ao cliente.

Normalmente, os modelos incrementais têm por objetivo comum lidar melhor com um conjunto de requisitos incertos ou sujeitos a alterações. O modelo espiral é o que é mais bem adequado se parece a projetos reais que a modelo cascata.

O modelo espiral assume a existência de alguma sequência entre as fases: ficando sem suporte as fases que ocorrem simultaneamente, ou as que necessitam de intercomunicação contínua para operarem.

Figura 2.2: Modelo Espiral



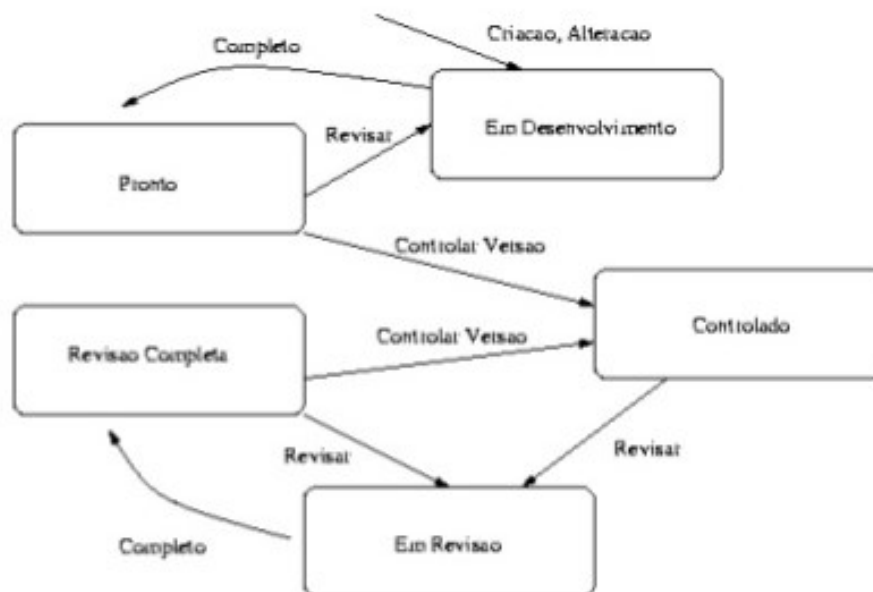
Fonte: Sommerville (2007, p. 44)

Modelo concorrente

Esse modelo tende a considerar um outro ponto importante: seria que as fases de um processo de desenvolvimento não ocorrem sequencialmente, e sim, concorrentemente. O mecanismo pelo qual o processo ocorre é baseado em eventos que sinalizam alterações de estado dentro de cada fase. Esse modelo mostra as atividades simultâneas de todos os membros da equipe de desenvolvimento, e os eventos que alteram o estado são gerados por necessidades do usuário, decisões da gerência, e resultados de revisões técnicas.

Tendo por exemplo, a fase de Especificação pode estar em um dentre muitos estados: o desenvolvimento, completo, revisado, e controlado no repositório. A criação, e toda revisão à especificação original, ativa a fase de Desenvolvimento, de forma que há um ciclo constante entre os estados.

Figura 2.3: Modelo Concorrente



Fonte: <http://slideplayer.com.br/slide/1804199/>

2.2 ENGENHARIA DOS REQUISITOS

Para compreender as dificuldades dos usuários com o sistema existente, bem como entender as necessidades para o novo sistema, considerando que a empresa colaborou na conversa entre analistas e usuários, não necessitando de reuniões formais, foi feita uma entrevista informal, porém seguindo um pequeno roteiro para ter um ponto de partida e manter o foco da conversa.

Em complemento a entrevista, foram feitos os casos de uso para que sejam ilustradas as ações dos usuários em relação ao sistema. Ambas as técnicas são definidas nos próximos tópicos.

2.2.1 ENTREVISTA

Segundo Sommerville (2007), entrevistas formais ou informais com as pessoas interessadas para o desenvolvimento do sistema, faz parte da maioria dos processos de engenharia de software. O autor coloca as entrevistas como úteis para obter um entendimento geral sobre o que deve-se fazer, como as pessoas interessadas podem interagir com o sistema e as dificuldades que enfrentam ao utilizar os sistemas atuais. A entrevista é uma técnica simples onde na maioria dos casos gera bons resultados.

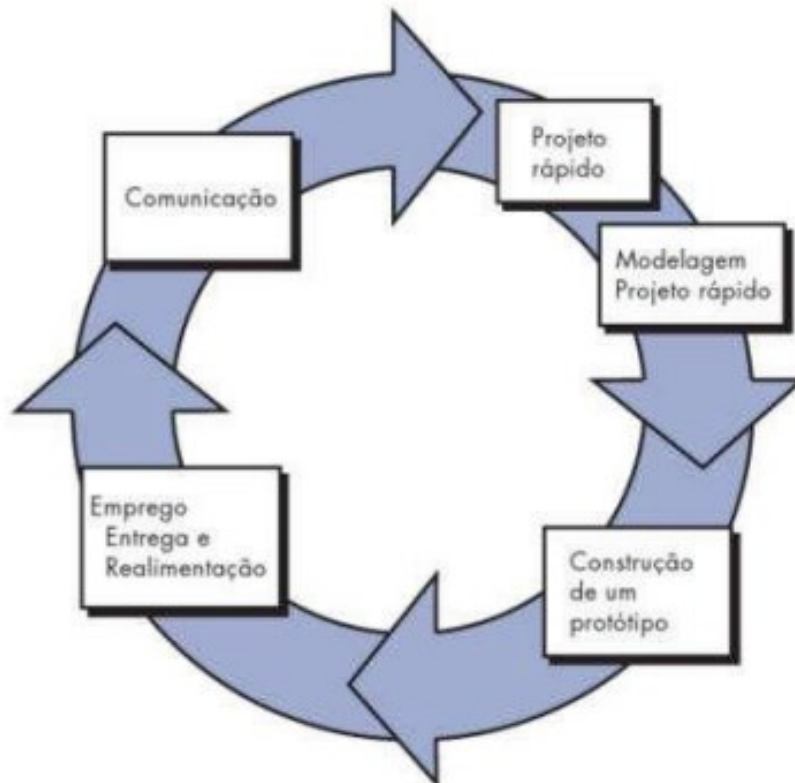
2.3 PROTOTIPAÇÃO

A prototipação é uma das técnicas para elucidar de forma objetiva os requisitos, Segundo Guedes (2009), a prototipação é uma técnica bastante conhecida e de fácil aplicação, que consiste em criar rapidamente um rascunho de baixa fidelidade do que seria o sistema quando ele estivesse finalizado. Um protótipo, normalmente, define pouco mais do que a interface do projeto a ser criado, mostrando como as informações seriam inseridas e recuperadas no sistema, demonstrando alguns exemplos com valores fictícios dos quais seriam os resultados apresentados pelo sistema, principalmente na geração de relatórios. O autor nos esclarece que o processo de prototipação, em muitos casos, evitar que seja descoberto que o software não atende as necessidades do cliente. A prototipação pode, também, mostrar falhas de comunicação na coleta dos requisitos.

Sommerville (2007) define um protótipo como uma versão inicial de um sistema de software usado para demonstrar conceitos, experimentar opções de projeto e, geralmente, conhecer mais sobre o problema e suas possíveis soluções.

Segundo Pressman (2011), O paradigma da prototipação, figura 2.4, é uma sequência de ações que começam com a Comunicação, seguida pelo Projeto Rápido, Modelagem de Projeto rápido, Construção de um protótipo e fechando a sequência o Emprego, Entrega e Realimentação.

Figura 2.4: O paradigma da prototipação



fonte <https://centraldaengenharia.wordpress.com>

fundamentado os conceitos do planejamento e a modelagem do projeto, são explanados, abaixo, a definição das ferramentas utilizadas no desenvolvimento do sistema.

2.4 MYSQL

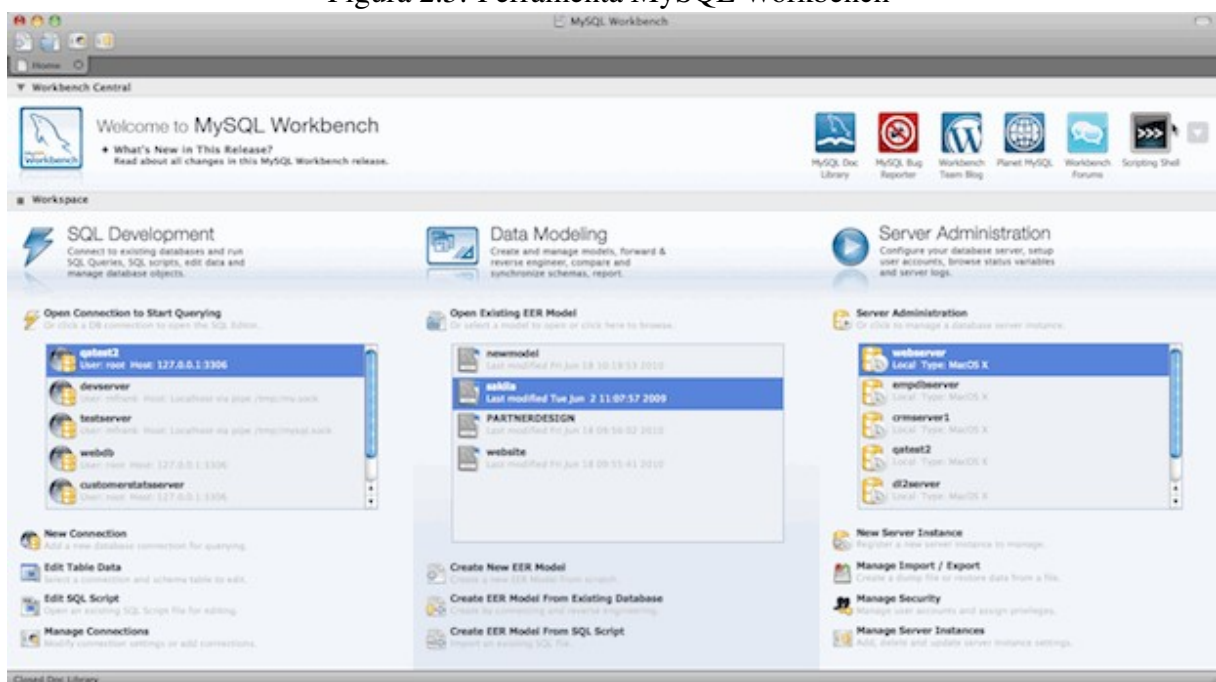
O MySQL é um sistema gerenciador de banco de dados relacional de código aberto. Utilizando a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

O sistema foi desenvolvido pela empresa sueca MySQL AB, em maio de 1995. Atualmente, a Oracle, embora tenha mantido a versão para a comunidade, tornou seu uso mais restrito e os desenvolvedores criaram, então, o projeto MariaDB para continuar desenvolvendo o código da versão 5.1 do MySQL, de forma totalmente aberta e gratuita. O MariaDB pretende manter compatibilidade com as versões lançadas pela Oracle.

2.4.1 MySQL Workbench

O MySQL Workbench é uma ferramenta visual para design, desenvolvimento e administração de base de dados MySQL. Essa ferramenta é originária do DBDesigner. A ferramenta é muito boa e completa. Possui as opções de abrir conexão, editar dados, editar scripts SQL, gerenciar conexões, novo modelo de dados, modelo de dados a partir da base (engenharia reversa), modelo de dados de um script SQL, criação de instâncias de servidor, importação/exportação de base, gerenciamento de segurança e gerenciamento de instâncias.

Figura 2.5: Ferramenta MySQL Workbench



Fonte: Próprio

2.5 BANDO DE DADOS

Os bancos de dados e os sistemas de bancos de dados se tornaram componentes essenciais no cotidiano da sociedade moderna. No decorrer do dia, a maioria de nós se depara com atividades que envolvem alguma interação com os bancos de dados. Por exemplo, se formos ao banco para efetuarmos um depósito ou retirar dinheiro, se fizermos reservas em um hotel ou para a compra de passagens aéreas, se acessarmos o catálogo de uma biblioteca informatizada para consultar uma bibliografia, ou se comprarmos produtos — como livros, brinquedos ou computadores — de um fornecedor por intermédio de sua página Web, muito provavelmente essas atividades envolverão uma pessoa ou um programa de computador que acessará um banco de dados. Até mesmo os produtos adquiridos em supermercados, em muitos casos, atualmente,

incluem uma atualização automática do banco de dados que mantém o controle do estoque disponível nesses estabelecimentos.

Essas interações são exemplos do que podemos denominar aplicações tradicionais de banco de dados, no qual a maioria das informações que são armazenadas e acessadas apresenta-se em formatos textual ou numérico. Nos últimos anos, os avanços tecnológicos geraram aplicações inovadoras e interessantes dos sistemas de banco de dados.

Os bancos de dados e a sua tecnologia estão provocando um grande impacto no crescimento do uso de computadores. É viável 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. Um banco de dados informatizado é usualmente mantido e acessado por meio de um software conhecido como SGBD¹, que é muitas vezes o termo banco de dados é usado como sinônimo de SGDB. Um SGBD - Sistema de Gerenciamento de Banco de Dados é uma coleção de programas que permitem ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades.

Outro fato bastante comum e plenamente justificado, é a diferença entre os termos “banco de dados” e os sistemas que o gerenciam. Segue uma definição segundo Navathe (2011, p.3):

Um banco de dados e uma coleção de dados relacionados. Os dados são fatos que podem ser gravados e que possuem um significado implícito. Por exemplo, considere nomes, números telefônicos e endereços de pessoas que você conhece. Esses dados podem ter sido escritos em uma agenda de telefones ou armazenados em um computador, por meio de programas como o Microsoft Access ou Excel. Essas informações são uma coleção de dados com um significado implícito, conseqüentemente, um banco de dados.

Os bancos de dados podem ter coleções de dados relacionados, com diversos tamanhos. Indo de uma pequena agenda onde contem números e contatos de pessoas até um índice grande de páginas e buscas relacionadas, ou mensagens e informações compartilhada entre muitos usuários de uma determinada rede social.

De forma computacional, existe um modo de software especializado que é criado especificamente com o intuito de gerenciar estas coleções de dados: os sistemas gerenciadores de banco de dados – de modo popular é reconhecido pela sigla SGBD (ou DBMS – DataBase Management Systems, na sigla original em inglês). Segue mais uma definição de Navathe (2011, p.3),

¹ Sistema Gerenciador de Banco de Dados

sobre o termo:

Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre vários usuários e aplicações. A definição de um banco de dados implica especificar os tipos de dados, as estruturas e as restrições para os dados a serem armazenados em um banco de dados

A construção de um banco de dados é o processo de armazenar os dados em alguma mídia apropriada controlada pelo SGBD. A manipulação inclui algumas funções, como pesquisas em banco de dados para recuperar um dado específico, atualização do banco para refletir as mudanças no mundo e gerar os relatórios dos dados. O compartilhamento permite aos múltiplos usuários e programas acessar, de forma concorrente, o banco de dados. Outras funções importantes do SGBD são a proteção e a manutenção do banco de dados por longos períodos. Sua função é a proteção do sistema contra o mau funcionamento ou falhas (crashes) no hardware ou software, e segurança contra acessos não autorizados ou maliciosos. Um banco de dados típico pode ter um ciclo de vida de muitos anos, então, os SGBD devem ser capazes de manter um sistema de banco de dados que permita a evolução dos requisitos que se alteram ao longo do tempo.

2.6 UML

UML (Unified Modeling Language). Pela definição de seu nome, vemos que a UML é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos. Os diagramas UML são usados para expressar ideias que serão utilizadas posteriormente no sistema que estiver sendo projetado, estabelecendo uma camada maior de abstração para observar todos os possíveis participantes daquele sistema. Diagramas UML capturam visões que formam novos tipos de diagramas com diferentes tipos de elementos, os mais comuns são: diagrama de classes e diagrama de caso de uso (LARMAN, 2003).

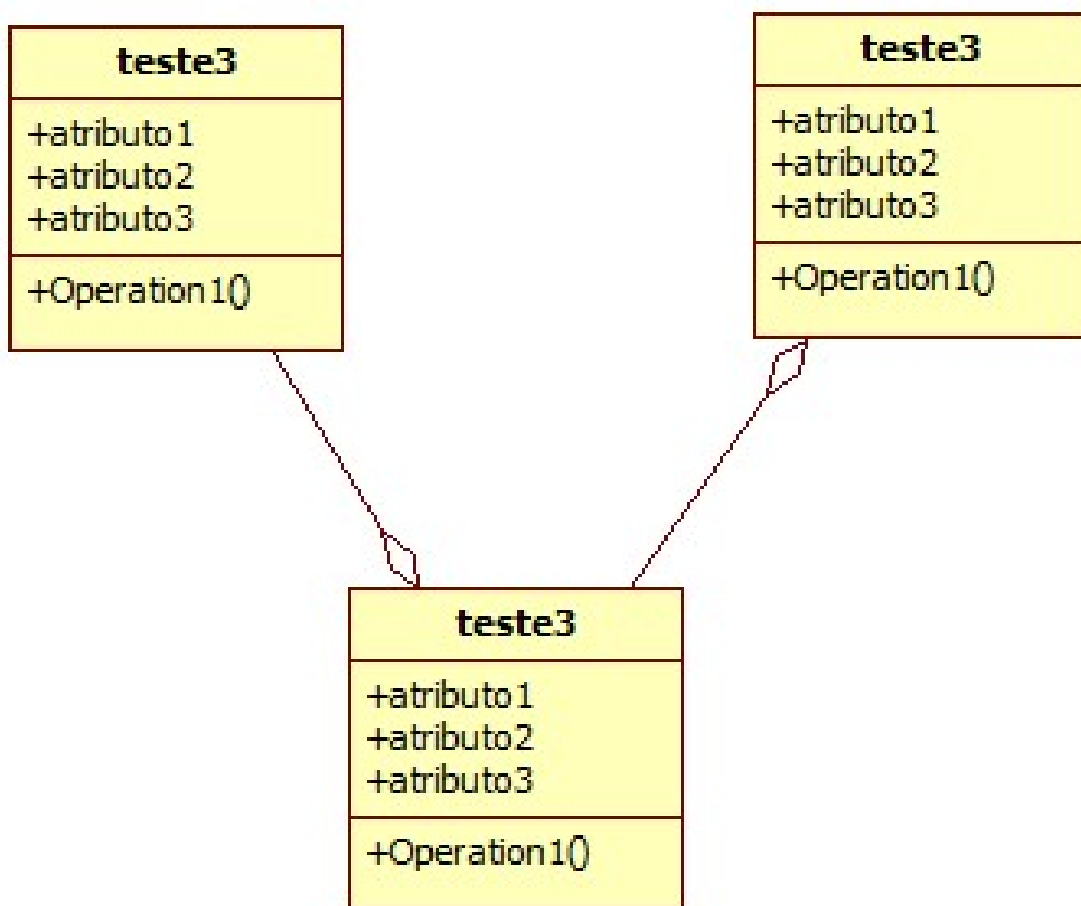
Diagrama de classes da UML são encontrados com maior frequência na modelagem de sistemas orientados a objetos. Eles são usados para modelar a visão estática do projeto de um sistema. Na maioria dos casos, isso envolve a modelagem do vocabulário do sistema, a modelagem de colaborações e/ou a modelagem de esquemas (BOOCH, 1999). Além disso, diagramas de classes oferecem suporte para os requisitos funcionais (os serviços/funcionalidades que o

sistema deverá fornecer aos usuários finais) de um sistema. Através do diagrama de classes é possível obter uma visão lógica de um esquema de banco de dados (BOOCH,1999).

2.6.1 Diagrama de classe

de Acordo com (MELO,2002) Em programação, um diagrama de classes é uma representação da estrutura e a interação entre as classes que servem de modelo para objetos. Portanto conclui-se, de maneira simples, que diagrama de classe seria um conjunto de objetos com as mesmas características, podem assim conhecer e identificar objetos posteriormente agrupá-los, de forma a encontrar suas respectivas classes. Na UML de diagrama de classe, uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos A Figura 2.5 demonstra um exemplo simples de um diagrama de classe.

Figura 2.6: Exemplo de Diagrama de Classe



Fonte: Próprio

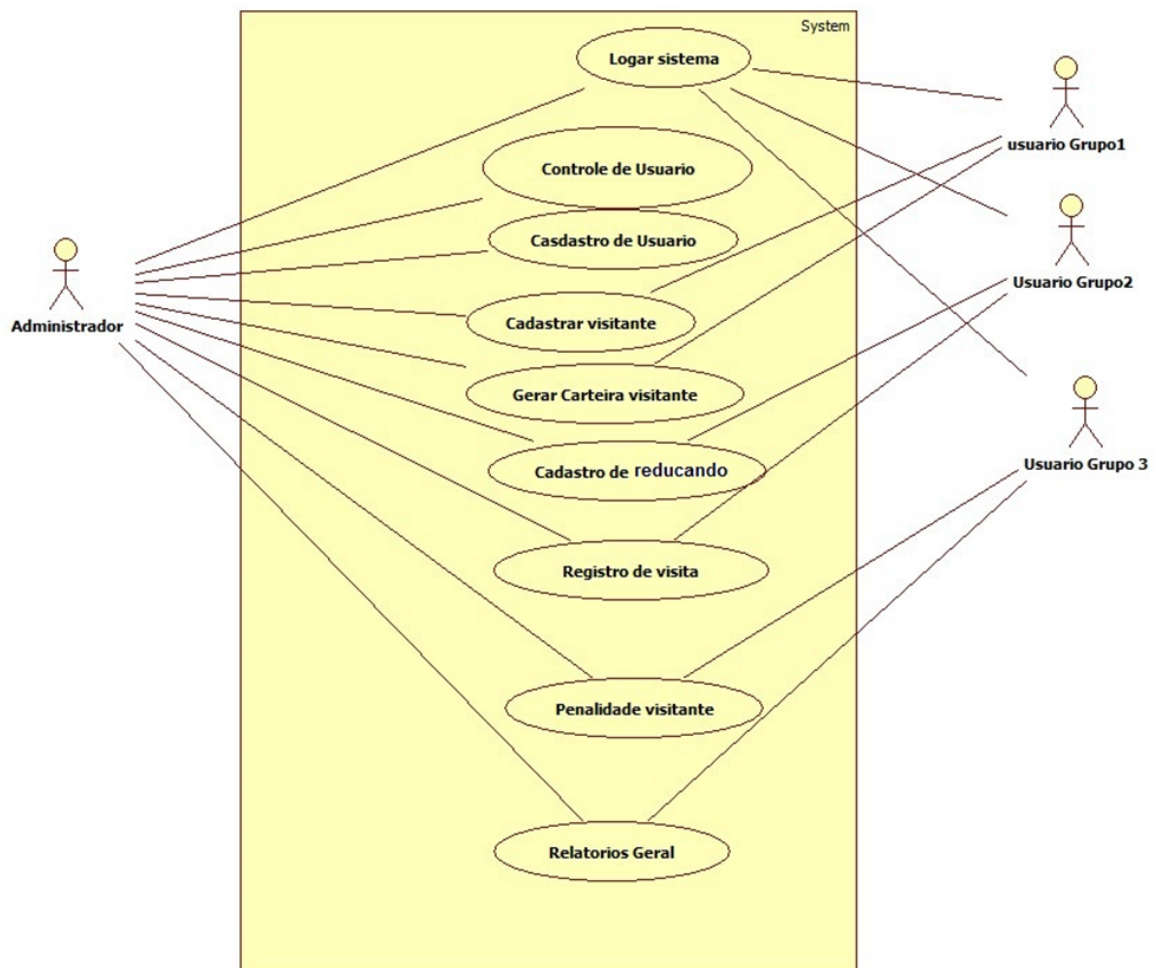
2.6.2 Diagrama de Casos de Uso

Diagramas de Casos de Uso são compostos basicamente por quatro partes:

- Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.
- Ator: É um tipo de usuário do sistema.
- Caso de Uso: É uma funcionalidade realizada pelo ator (usuário).
- Comunicação: é a interação entre um ator com um caso de uso.

A Figura 2.6 demonstra um exemplo simples de um diagrama de caso de uso.

Figura 2.7: Diagrama de Caso de uso



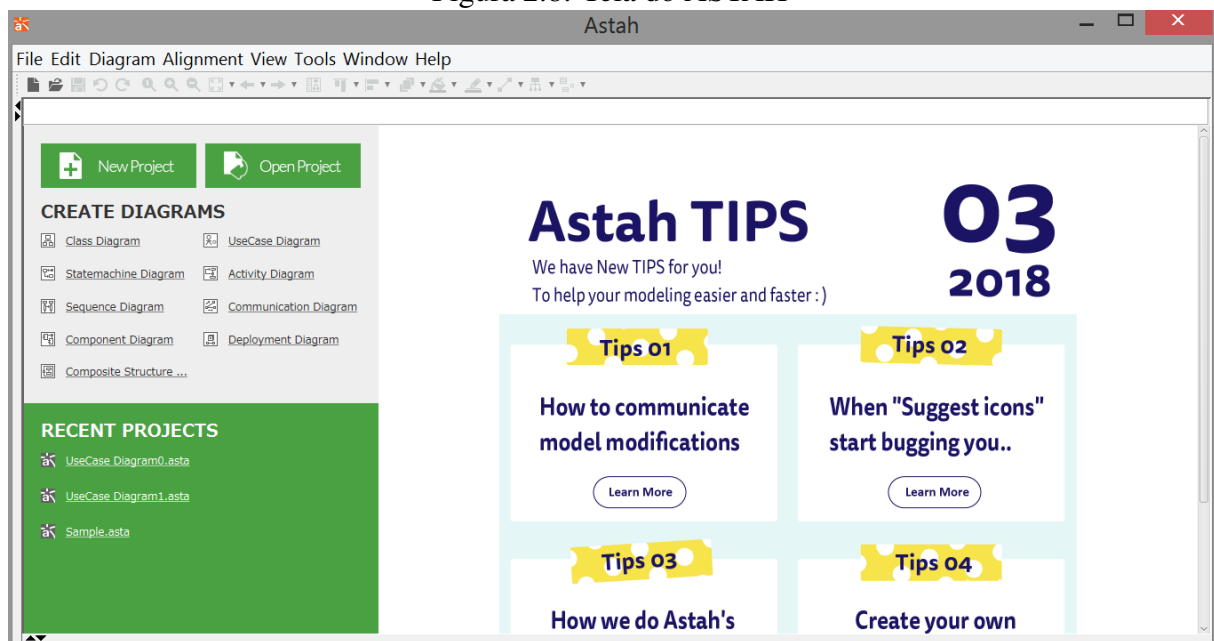
fonte: Elaboração Própria

2.7 ASTAH

Astah é uma ferramenta de modelagem UML. O software foi desenvolvido no Japão na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possua JVM (Máquina Virtual Java). JUDE (Ambiente para Desenvolvedores UML e Java) obteve o prêmio “Produto de Software do Ano 2006”, pela Agência de Promoção de Informação Tecnológica no Japão. Anteriormente conhecido como JUDE, ele funciona nas plataformas Windows, Mac e Linux.

A ferramenta Astah foi utilizada na modelagem dos diagramas de classe e de casos de uso, essa ferramenta já é bastante consolidada, voltada para a modelagem de sistemas utilizando a UML.

Figura 2.8: Tela do ASTAH



fonte: Elaboração Própria

2.8 DELPHI7

O Delphi7 foi lançado em 1995 pela Borland (empresa de ferramentas de desenvolvimento).

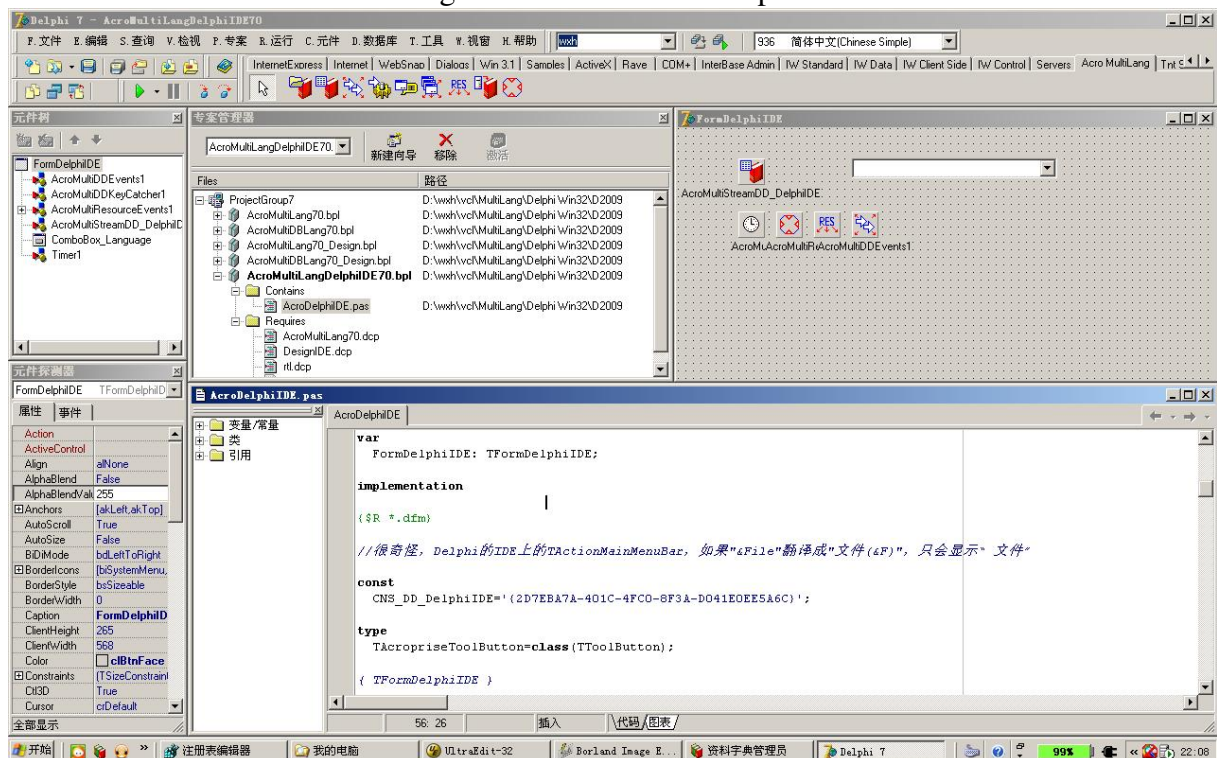
É uma ferramenta de desenvolvimento visual que utiliza uma linguagem de programação totalmente orientada a objeto, o Object Pascal, que possibilita a criação de novos componentes a partir dos existentes na biblioteca do Delphi7, criação de novas classes e objetos, a transmissão das propriedades e métodos de uma classe existente, etc.

É uma ferramenta de desenvolvimento visual que utiliza uma linguagem de programação totalmente orientada a objeto, o Object Pascal, que possibilita a criação de novos componentes a partir dos existentes na biblioteca do Delphi7, criação de novas classes e objetos, a transmissão das propriedades e métodos de uma classe existente, etc.

Por ser uma ferramenta de desenvolvimento visual, o Delphi7 automatiza o trabalho de criação de aplicativos, pois auxilia a construção manual de todas as partes do aplicativo, ou seja, através de um clique do mouse sobre os componentes que serão adicionados aos formulários. Isso gera uma redução do tempo de desenvolvimento e manutenção dos programas e, além disso, um aumento na sua eficiência.

O Delphi compila os programas em códigos nativos, gerando um arquivo executável (.EXE) independente de outros arquivos e de bibliotecas de vínculo dinâmico (arquivos .DLL). Em outras palavras, o programador, ou projetista, pode fornecer aos usuários um único arquivo, EXE, e escrever programas de instalação simples. (MANZANO et al.,1998).

Figura 2.9: o ambiente Delphi7



Fonte: Delphi 2010 IDE Look and feel like Delphi 7

3 ESTÁGIO SUPERVISIONADO

O estágio supervisionado foi realizado no Instituto de Administração Penitenciário do Acre, na Coordenadoria de Tecnologia da Informação e Comunicação, sob supervisão do coordenador responsável, Leandro Ruy Batista da Silva.

Para atingir o objetivo de desenvolver uma solução que traga maior eficiência ao processo de ajuda no controle dos reeducandos e seus visitantes, foram utilizadas atividades relacionadas engenharia de requisitos, prototipação, modelagem e desenvolvimento do sistema, dentre outras.

Ao fim do estágio a solução foi implantada e encontra-se atualmente em fase de teste, atendendo à unidade do IAPEN no município de Rio Branco.

3.1 EMPRESA

O INSTITUTO DE ADMINISTRAÇÃO PENITENCIÁRIA (IAPEN) foi criado pela lei nº 1908 em 31 de julho 2007 que se constitui numa entidade Autárquica dotada com personalidade jurídica de direito público interno, possui autonomia administrativa, financeira e patrimonial. Tem sua finalidade, planejar, programar, executar, coordenar, fiscalizar e executar políticas de diretrizes prisionais. Hoje, o Instituto está vinculado à Secretaria de Justiça e Direitos Humanos.

O IAPEN tem por finalidade garantir a integridade física e moral de seus reeducandos, afim de exercer a democracia e os direitos humanos.

Tabela 3.1: Tabela quadro IAPEN

Estabelecimentos Penais	Masculino	Feminino	Total
Penitenciária	11	1	12
Colônia Agrícola, Industrial ou Similar	-	-	-
Centro de Observação Criminológica e Triagem	-	-	-
Hospitais de Custódia e Tratamento Psiquiátrico	-	-	-
Total	11	1	12

font

IAPEN-AC

- O Estado possui 12 estabelecimentos penais, divididos da seguinte maneira:
- Não existem Casa de Albergado no Estado do Acre.
- No Estado não existem estabelecimentos penais terceirizados.
- Não existe experiência na utilização do método Apac – Associação de Proteção ao Condenado no Estado.
- As delegacias das cidades de Tarauacá e Feijó são as únicas do Estado que custodiam presos.
- O Sistema Penitenciário do Estado do Acre não presta informações sobre a inclusão de presos estrangeiros aos Consulados.
- O único Consulado que mantém contato com as unidades do Acre é o Consulado do Peru.
- O Consulado do Peru, em época natalina, fornece kits higiene, roupas e panetones; fora dessa época do ano, não oferece outra assistência material ao preso peruano.
- O Consulado Peruano firmou convênio com a Faculdade FIB-FAAO, onde estagiários na área de direito atendem os presos estrangeiros.
- No Acre, com base no mês de janeiro de 2008, existiam 6 presos, todos brasileiros, portadores de deficiência, sendo 1 tetraplégico, 1 com atrofia em membro superior, 1 com deficiência visual e 3 com deficiência em membros inferiores.

- Atualmente nenhum preso cumpre pena no Regime Disciplinar Diferenciado.
- A Unidade Antônio Amaro Alves, situada na capital, possui uma ala, com 13 celas, para o cumprimento do RDD.
- A unidade do Estado custodia presos em cumprimento de pena no regime fechado, semi-aberto e provisório.
- Com relação às visitas sociais e íntimas:
- Frequência de realização das visitas sociais:
- As visitas sociais são realizadas aos sábados e domingos.
- Número máximo de visitantes por preso:
- visitantes, não sendo computados, nesse número, as crianças.
- Tempo de duração:
- O tempo de duração da visita social é de, no máximo, 7 horas.
- Visita íntima:
- Ocorrem semanalmente, às quartas feiras.
- Não existem locais apropriados para a realização das visitas íntimas. Essas visitas ocorrem, atualmente, nas próprias celas.

3.1.1 Monitoramento eletrônico

Além disso, o IAPEN, juntamente com a Vara de Execuções Penais do Tribunal de Justiça do Acre (TJAC), mantém cerca de 800 reeducandos no sistema de monitoramento eletrônico, o que representa uma economia real nos gastos com pessoas em cumprimento de pena.

3.1.2 Reinserção pelo trabalho

O IAPEN realiza uma série de atividades profissionalizantes para envolver aqueles que cumprem penas e têm bom comportamento, buscando reinseri-los na sociedade. No ano de 2017, cerca de 180 reeducandos foram certificados em diversos cursos.

Figura 3.1: Sede do Instituto do Administrativo no Acre IAPEN

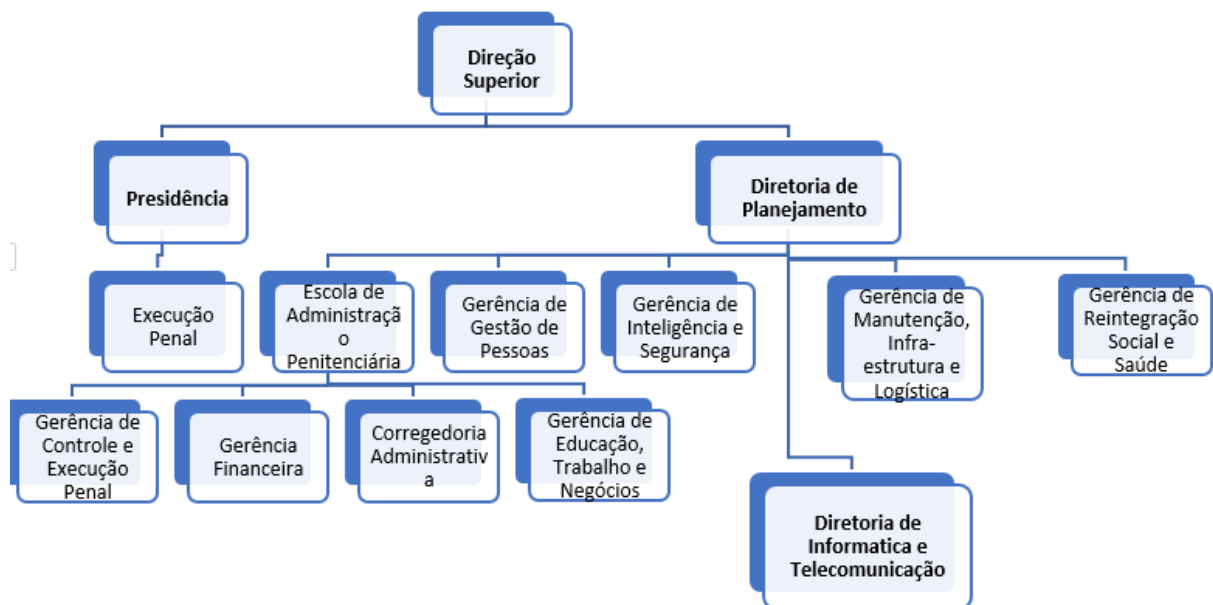


<http://ac.gov.br/wps/portal>

O setor no qual foi desenvolvido o nosso estágio supervisionado foi na Diretoria de Informática e Telecomunicação, que é o setor encarregado a prestar suporte técnico aos sistemas já implantados.

3.1.3 ESTRUTURA ORGANIZACIONAL

Figura 3.2: Organograma do IAPEN



Adaptado do pdsp_{ac}(2016/2017)

3.2 ATIVIDADES DESENVOLVIDAS

As atividades desenvolvidas durante o estágio supervisionado, foram realizadas desta forma:

- a Identificação do problema;
- b Levantamento de requisitos;
- c Modelagem do sistema;
- d Implementação da solução em Delphi7;
- e Prototipação;
- f Testes e validação.

3.2.1 IDENTIFICAÇÃO DO PROBLEMA

A identificação do problema da pesquisa surgiu a partir da constatação feita pela Diretoria de TI do IAPEN/AC de que haviam muitas falhas com o atual sistema de controle dos reeducando e seus visitantes.

Através de uma reunião com os responsáveis do atual sistema foram coletadas informações e assim identificar as necessidades do cliente. Na reunião foram descritos como era feito o controle dos reeducando e seus visitantes e relatadas as principais falhas.

3.2.2 LEVANTAMENTO DOS REQUISITOS

Com a primeira reunião conseguiu-se obter a versão inicial do documento de requisitos, identificando todos requisitos funcionais e não funcionais de acordo com a necessidades básicas do cliente, além de estudos de viabilidade, restrições e de ferramentas a serem usadas no desenvolvimento do software. Com a implementação do software houve a necessidade de adicionar novos requisitos, e com isso o documento de requisitos passou por várias versões.

O documento de requisitos completo é encontrado no APÊNDICE A.

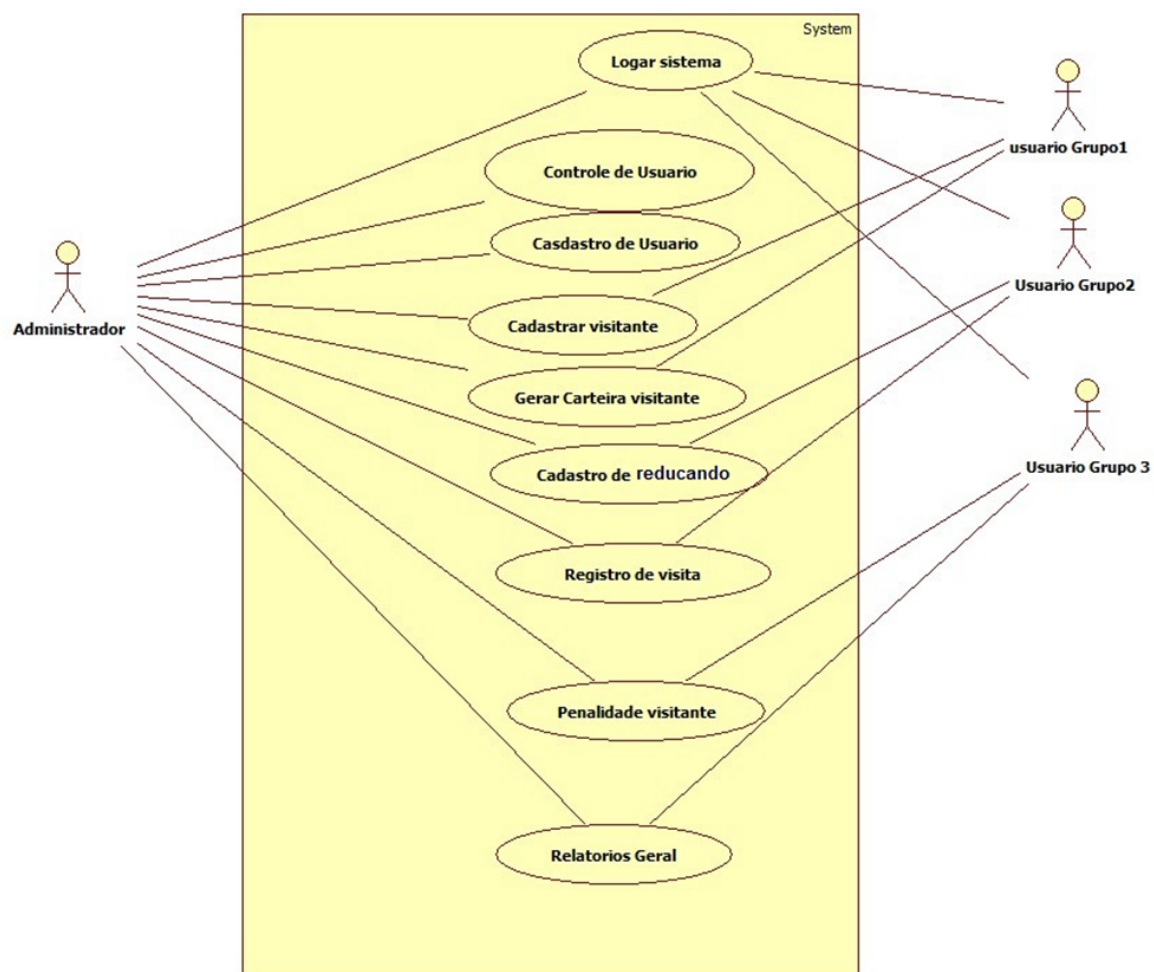
3.2.3 MODELAGEM DO SISTEMA

Na implementação do software foram feitos os diagramas usando a linguagem UML. Com os diagramas de implementação é possível ver relacionamentos entre os componentes de software e hardware no sistema, e com isso podemos validar suas funcionalidades.

A figura 3.3 mostra o diagrama de caso de uso e cada fluxo de tarefas que serão realizadas pelos atores, o sistema tem o administrador e seus usuários e cada um, o administrador exerce todas as tarefas e libera as tarefas dos usuários de acordo com a função de cada um como relatado nos requisitos funcionais do documento de requisitos.

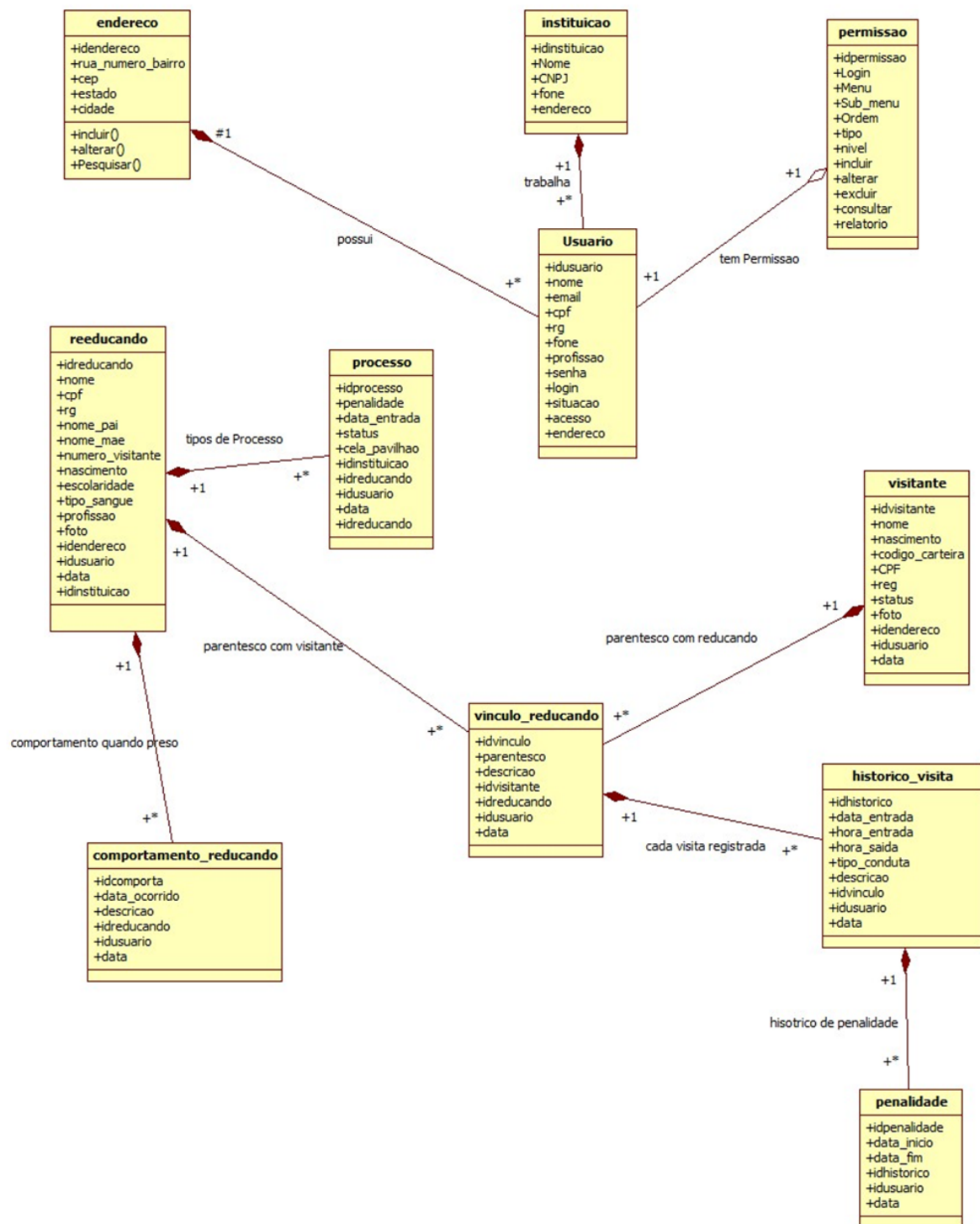
Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes (Melo, 2004).

Figura 3.3: Diagrama de Caso de uso



fonte: Elaboração Própria

Figura 3.4: Exemplo de Diagrama de Classe



Fonte: Elaboração Própria

3.2.4 PROTOTIPAÇÃO

Com a coleta de todos os requisitos funcionais e não funcionais foi iniciado a prototipação do software. Foram feitos alguns rascunhos levando em consideração tudo o que o IAPEN/AC (cliente) necessitava, assim surgiram os primeiros protótipos de baixa fidelidades para ajudar na demonstração ao cliente de como ficaria a interface do sistema, os desenhos continham as paletas mostrando os botões, caixa de texto e ícones, etc.

Finalmente foi iniciada a prototipação, que ao longo do projeto foram apresentadas as fases do seu desenvolvimento até atingir a forma final das interfaces.

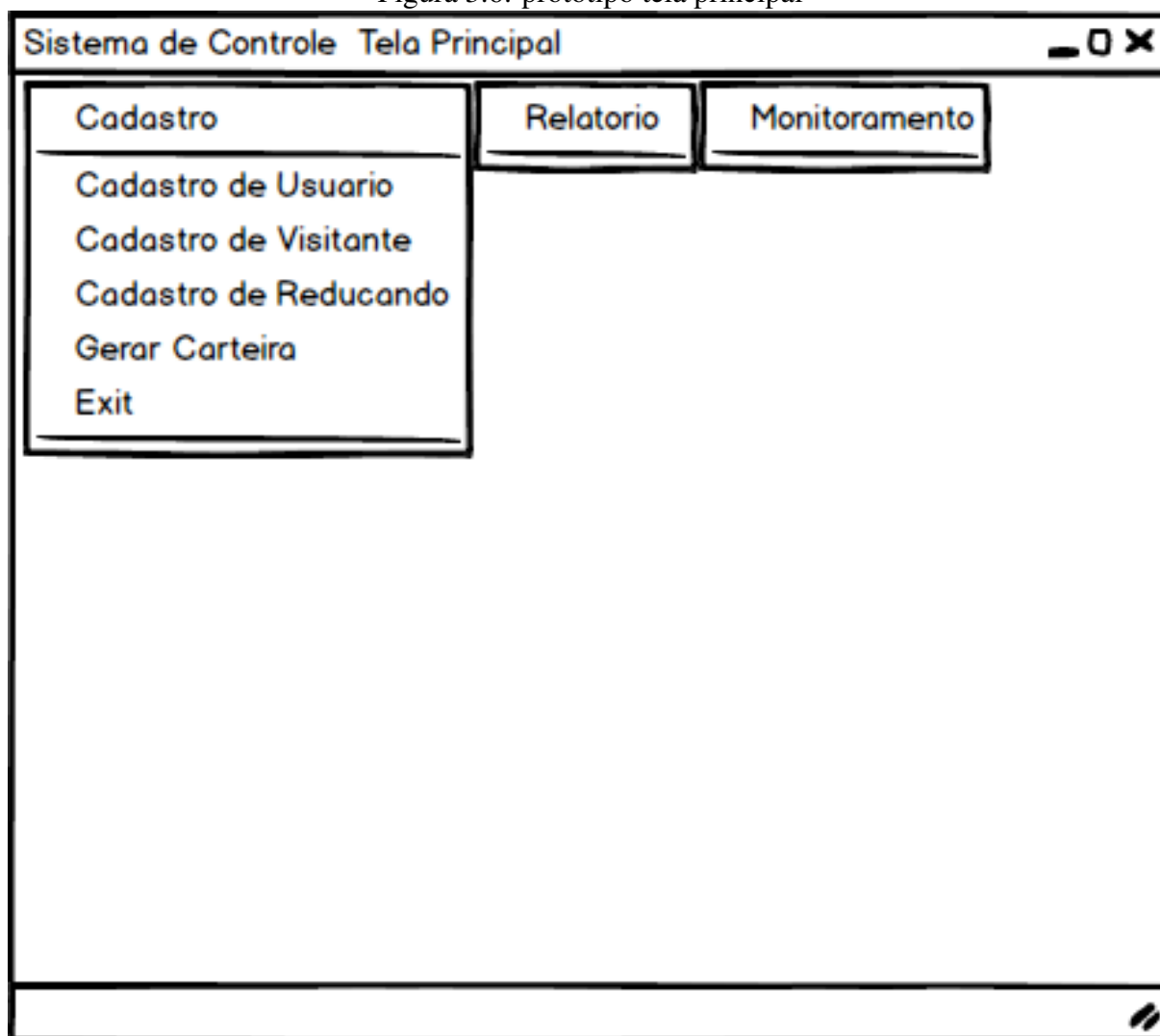
As figuras 3.5, 3.6 e 3.7 mostra o a criação dos protótipos de baixa fidelidade.

Figura 3.5: Tela de login

O protótipo da tela de login é apresentado dentro de uma moldura retangular. No topo, há uma barra de título com o texto "Tela de Login". Abaixo, o formulário contém duas seções principais: "Usuario" e "Senha". Cada seção possui um campo de entrada de texto retangular. Na base do formulário, há uma barra contendo três botões retangulares: "Entrar", "suorte" e "Sair".

Fonte: Elaboração própria

Figura 3.6: protótipo tela principal



Fonte: Elaboração própria

Figura 3.7: protótipo tela de cadastro

O protótipo da tela de cadastro, intitulada 'cadastros', apresenta uma interface simples. No topo, há uma barra de título com o nome da tela e ícones de minimizar, maximizar e fechar. Abaixo da barra, há uma barra de menu com quatro botões: 'Novo', 'Salvar', 'Editar' e 'Procura'. O corpo da tela é dividido em duas seções principais. À esquerda, há uma grande área vazia, provavelmente destinada a uma lista de registros. À direita, há campos de entrada para os dados do cadastro: um campo longo para 'Nome', e dois campos menores para 'CPF' e 'RG'. Abaixo desses campos, há uma barra horizontal, possivelmente para uma barra de rolagem ou uma barra de status.

Fonte: Elaboração própria

3.2.5 Implementação da Solução em Delphi7

Após a coleta de todos os artefatos foi dado início a implementação do sistema, o sistema foi desenvolvido IDE Delphi7 em linguagem Objeto Pascal.

Primeiro foram criados as funções responsáveis pela comunicação com SGBD e fornecendo operações com CRUD¹, com a utilização de componentes ZEUS², a figura 3.8, mostra a função de comunicação com banco de dados.

Figura 3.8: Código comunicação com Banco de Dados

```
function TForm1.conexao: Boolean;
var
  config: TIniFile;
  porta: integer;
begin
  config:=TIniFile.Create('C:\Program Files (x86)\SGE\config.ini');
  if not FileExists('C:\Program Files (x86)\SGE\config.ini') then
  begin
    ShowMessage('arquivo config.ini nao existe! tente criar, no local C:\Program Files (x86)\SGE\config.ini ');
  end
  else
  begin
    try
      Datamodule3.connect.HostName:=Criptor.Descriptor(config.ReadString('connection1','hostname',''),108); //componente ZEUS metod hostname
      Datamodule3.connect.Database:=Criptor.Descriptor(config.ReadString('connection1','database',''),108); //componente ZEUS metod base de dados
      Datamodule3.connect.Password:=Criptor.Descriptor(config.ReadString('connection1','password',''),108); //componente ZEUS metod password
      Datamodule3.connect.User:=Criptor.Descriptor(config.ReadString('connection1','user',''),108); //componente ZEUS metod user
      Datamodule3.connect.Port:=strtoint(Criptor.Descriptor(config.ReadString('connection1','port',''),108)); //componente ZEUS metod port
      Datamodule3.connect.Protocol:=Criptor.Descriptor(config.ReadString('connection1','protocolo',''),108); //componente ZEUS metod tipo de banco
      Result:=true;
      config.Free;
    except
      showmessage('Erro na Conexao com Banco de dados, verifique o arquivo config.ini');
      Result:=false;
      config.Free;
    end;
  end;
end;
```

Fonte: Elaboração própria

¹ Refere-se a quatro operações básicas utilizadas em banco de dados relacionais ou em interface, sendo criação, consulta, atualização e destruição de dados.

² Componente para conexão nativas com banco de dados MySql entre Outros

Uma vez estabelecidas todas as comunicações com o banco de dados, foi necessário, determinar uma função para validar o login em senha de usuário, ao utilizar o sistema a figura 3.9 mostra com detalhes essa função.

Figura 3.9: função validar locação do usuário

```
function TForm1.loginValido(const usuario, senha: String): Boolean;
begin
  with datamodule3.connect, datamodule3.login do
    begin
      if not datamodule3.connect.Connected then //Caso o componente ConLOGIN não esteja conectado ao BD
        begin
          datamodule3.connect.Connected := True;
          datamodule3.login.Close;
          datamodule3.login.SQL.Clear;
          datamodule3.login.SQL.Add('SELECT Count(*) as numero, Login, Email, acesso,
            idusuario, senha FROM tbusuario WHERE UPPER(login)= '
            + QuotedStr(AnsiUpperCase(Trim(usuario))) + ' AND senha = '
            + QuotedStr(AnsiUpperCase(Trim(senha))) + ' AND Situacao = '
            + QuotedStr(AnsiUpperCase(Trim('A'))));
          datamodule3.login.Open;
          if datamodule3.login.Fields[0].AsInteger > 0 then
            begin
              Datamodule3.permission.Active := False;
              datamodule3.permission.close;
              Datamodule3.permission.SQL.Clear;
              Datamodule3.permission.SQL.Add('select * from tbPermissao');
              Datamodule3.permission.SQL.Add('where Login = '+QuotedStr(Trim(usuario))+');
              Datamodule3.permission.SQL.Add(' and Situacao = '+QuotedStr('A')+');
              Datamodule3.permission.SQL.Add('ORDER BY Ordem');
              Datamodule3.permission.ExecSQL;
              Datamodule3.permission.Open;
              Result:= true;
            end
          else
            begin
              result:=false
            end;
          end;
        end;
      end;
    end;
  end;
```

Fonte: Elaboração própria

O passo seguinte foi a criação da classe de Formulários ou interface gráfica de usuários GUI³, e todas as telas de cadastros, consultas, monitoramentos. Todas as telas foram construídas com base nos protótipos de baixa fidelidade. A camada de visão foi toda baseada no componentes nativos do Delphi7 responsável pela criação das GUI, as figura 3.10, mostra o código de implementação para a criação de interface do formulário principal.

³Conhecida como Interface Gráfica com Usuário (Graphical User Interface – GUI), é onde os resultados são apresentados em modo gráfico.

Figura 3.10: classe do formulário de login

```

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, IniFiles, StdCtrls, Buttons, ExtCtrls, ComCtrls, Menus, ToolWin,
  ImgList, TreeSeguranca, UCDataConnector, jpeg, UCZEOSConn, UCBase;

type
  TForm2 = class(TForm)
    MainMenu1: TMainMenu;
    StatusBar1: TStatusBar;
    ImageList1: TImageList;
    Image1: TImage;
    procedure BitBtn1Click(Sender: TObject);
    procedure Image1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure ToolButton6Click(Sender: TObject);
    procedure Action1Execute(Sender: TObject);
    procedure Usuario1Click(Sender: TObject);
    procedure Visitante1Click(Sender: TObject);
    procedure Sair1Click(Sender: TObject);
    procedure reeducando1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure consultacriada1Click(Sender: TObject);
    procedure sql1click(Sender: TObject);
    procedure carteira1click(Sender: TObject);
    procedure AlterarSenha1click(Sender: TObject);
    procedure visitantereeducao1click(Sender: TObject);
    procedure InativaVisitante1click(Sender: TObject);
    procedure Instituicao1click(Sender: TObject);
  private
    procedure primeiroacesso;
    function retornos(form, nome: String): Boolean;
  { Private declarations }
  public

```

Fonte: Elaboração própria

após fazer a tela de login, foi criado o formulário pai, onde o mesmo fica responsável de validar o nível de acesso do usuário a figura 3.11, 3.12 e 3.13 mostra o trecho do código

Figura 3.11: Código código nível de usuário

```

procedure TForm2.FormActivate(Sender: TObject);
var
//criação das variaveis
Cont1, Cont2,Cont3,cont4,cont5,cont6 : integer;
vMenu, vItem : TMenuItem;
salva, xy,s : string;
begin
//coloca a query na posição consulta feita ao logar o usuario e salva tos num array
Datamodule3.permission.First;
salva:='';
while not Datamodule3.permission.Eof do // enquanto não chegar no fim faça o loop
begin
with Datamodule3.permission do
begin
if (length(trim(FieldByName('Formulario').AsString)) > 0) then
AddPermissao(FieldByName('Formulario').AsString, // pega todo os formulario e depois monta o menu
FieldByName('Incluir').AsBoolean,
FieldByName('Alterar').AsBoolean,
FieldByName('Excluir').AsBoolean,
FieldByName('Consultar').AsBoolean,
FieldByName('Reativar').AsBoolean);
end;
if (Datamodule3.permission.FieldByName('Tipo').AsInteger <> 2) then
begin
if salva <> Datamodule3.permission.FieldByName('Menu').AsString then
begin
salva:=Datamodule3.permission.FieldByName('Menu').AsString;
vMenu:=addMenu(nil, Trim(salva), '');
end
else
begin
if Datamodule3.permission.FieldByName('Tipo').AsFloat = 0 then
vItem:=addMenu(vMenu, Trim(Datamodule3.permission.FieldByName('Sub_Menu').AsString) , '')
else
if Datamodule3.permission.FieldByName('Nivel').AsFloat = 2 then
AddMenu(vItem, Trim(Datamodule3.permission.FieldByName('Sub_Menu').AsString) ,

```

Fonte: Elaboração própria

Figura 3.12: Código código nível de usuário

```

begin
if Datamodule3.permission.FieldByName('Tipo').AsFloat = 0 then
vItem:=addMenu(vMenu, Trim(Datamodule3.permission.FieldByName('Sub_Menu').AsString) , '')
else
if Datamodule3.permission.FieldByName('Nivel').AsFloat = 2 then
AddMenu(vItem, Trim(Datamodule3.permission.FieldByName('Sub_Menu').AsString) ,
('mn'+Trim(Datamodule3.permission.FieldByName('Formulario').AsString)))
else
AddMenu(vMenu, Trim(Datamodule3.permission.FieldByName('Sub_Menu').AsString) ,
('mn'+Trim(Datamodule3.permission.FieldByName('Formulario').AsString)));
end;
end;
Datamodule3.permission.Next;
end;
AddMenu(nil, 'Sair', '');
for cont1 := 0 to Form2.MainMenu1.Items.Count -1 do begin
s := Form2.MainMenu1.Items[cont1].Caption;
Delete(s, pos('&',Form2.MainMenu1.Items[cont1].Caption),1);
//node1 := trvGeral.Items.Add(nil, s);
for cont2 := 0 to Form2.MainMenu1.Items[cont1].Count - 1 do begin
s := Form2.MainMenu1.Items[cont1].Items[cont2].Caption;
Delete(s, Pos('&', Form2.MainMenu1.Items[cont1].Items[cont2].Caption), 1);
if s <> '-' then begin
//node2 := trvGeral.Items.AddChild(node1, s);
for cont3 := 0 to Form2.MainMenu1.Items[cont1].Items[cont2].Count - 1 do begin
s := Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Caption;
Delete(s, Pos('&', Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Caption), 1);
if s <> '-' then begin
//node3 := trvGeral.Items.AddChild(node2, s);
for cont4 := 0 to Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Count - 1 do begin
s := Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Caption;
Delete(s, Pos('&', Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Caption), 1);
if s <> '-' then begin
//node4 := trvGeral.Items.AddChild(node3, s);
for cont5 := 0 to Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Count - 1 do begin
s := Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Items[cont5].Caption;

```

Fonte: Elaboração própria

Figura 3.13: Código código nível de usuário

```

if s <> '-' then begin
    //node3 := trvGeral.Items.AddChild(node2, s);
    for cont4 := 0 to Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Count - 1 do begin
        s := Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Caption;
        Delete(s, Pos('&', Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Caption), 1);
        if s <> '-' then begin
            //node4 := trvGeral.Items.AddChild(node3, s);
            for cont5 := 0 to Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Count - 1 do begin
                s := Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Items[cont5].Caption;
                Delete(s, Pos('&', Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Items[cont5].Caption), 1);
                if s <> '-' then begin
                    //node5 := trvGeral.Items.AddChild(node4, s);
                    for cont6 := 0 to Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Items[cont5].Count - 1 do begin
                        s := Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Items[cont5].Items[cont6].Caption;
                        Delete(s, Pos('&', Form2.MainMenu1.Items[cont1].Items[cont2].Items[cont3].Items[cont4].Items[cont5].Items[cont6].Caption), 1);
                        if s <> '-' then
                            //trvGeral.Items.AddChild(node5, s);
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
end;
end;
end;
end;
end;
end;
end;

```

Fonte: Elaboração própria

e por ultimo foram criados os outros formulários de cadastros, as figura 3.14 mostra os código de implementação de cada formulário.

Figura 3.14: Código de Cadastro de Usuário

```

try
DataModule3.cadUsuario.Open;
except on E:Exception do
showmessage('Erro ao selecionar usuario:'+ E.Message);
end;
if DataModule3.cadUsuario.RecordCount <> 0 then
begin
    if DataModule3.cadUsuario.Active then DataModule3.cadUsuario.Close;
    ShowMessage('Usuário Já Cadastrado');
    Abort;
    edNome.SetFocus;
end
else
begin
    if DataModule3.cadUsuario.Active then DataModule3.cadUsuario.Close;
    DataModule3.cadUsuario.SQL.Clear;
    DataModule3.cadUsuario.SQL.Add('INSERT INTO tbUsuario');
    DataModule3.cadUsuario.SQL.Add(' (login, NomeCompleto,email ,Senha, Situacao,cpf,rg,profissao,fone,endereco,identificacao,acesso)');
    login:=AnsiUpperCase(edID.Text);
    DataModule3.cadUsuario.SQL.Add('VALUES ('+QuotedStr(trim(AnsiUpperCase(edID.Text)))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(AnsiUpperCase(edNome.Text)))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(edmail.Text))+','');
    senha:=gerarsenha;
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(MD5(senha)))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(edRg))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(maskedit1.Text))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(edit3.Text))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(edit8.Text))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(maskedit2.Text))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(edit7.Text))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(trim(combobox1.Items[combobox1.Itemindex]))+','');
    DataModule3.cadUsuario.SQL.Add(' '+QuotedStr(inttostr(0))+','');
    try
    if vopf(maskedit1.Text)=true then
    begin
        DataModule3.cadUsuario.ExecSQL;
        gravaPermissao;
        showmessage('usuario cadastrado '+ 'login: '+ login + ' '+ 'senha: '+ senha);
    end;
    end;
end;
end;

```

Fonte: Elaboração própria

3.2.6 Testes e Validação

Durante o desenvolvimento do software foram realizados vários testes funcionais para saber se os objetivos estavam alcançando os resultados esperados pelo cliente e se estava atendendo a todos os requisitos estabelecidos no documento de requisitos.

No primeiro teste não foi obtido muito sucesso, após a instalação do software foram detectados alguns erros, um deles era que ele não salvava o bando de dados criados naquele dia como também não gerava as planilhas em Excel. Esses foram os erros do teste funcional.

Na apresentação de interfaces pedimos para alguns estagiários, para testar realizando alguns cadastros e eles não tiveram dificuldade nenhuma. A interação com usuário foi bem-sucedida.

O teste do software foi de suma importância para poder corrigir os erros e observar se o sistema seria interativo com o usuário. Sabemos que o teste é muito importante para o desenvolvimento de um software antes de colocá-lo para funcionar, pois isso evita problemas no futuro.

3.3 RESULTADOS OBTIDOS

No final de todos os testes o resultado obtido com desenvolvimento do sistema foi satisfatório, pois foram alcançados todos os objetivos descritos no documento de requisitos, cumprindo-se tudo que o cliente esperava. O software realiza cadastro, tanto dos visitantes quanto dos reeducando, emite carteiras com código de barras, além de gerar relatórios, coisas que o sistema anterior não fazia. O sistema ficou mais organizado e mais seguro. Seguro porque, código de barras na carteira evita a entrada do visitante que esteja com a carteira inativa.

3.4 CONCLUSÃO

Conclui-se que todos os objetivos propostos pelo cliente foram alcançados com êxito. Mostrando assim que foram aplicados grande parte dos conhecimentos adquiridos durante o curso de graduação em Sistemas de Informação.

Desenvolver um sistema de cadastro de visitantes e reeducando para o IAPEN/AC foi um excelente desafio servindo como experiência para o mercado de trabalho. Também nos mostrou como lidar com um cliente e trabalhar para uma empresa de grande porte. O sistema foi implantado atingindo todos os objetivos que se esperava, cumprindo-se assim a missão de desenvolver um trabalho acadêmico.

4 CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES

Neste capítulo serão apresentadas as considerações finais sobre as atividades do estágio supervisionado realizado no IAPEN/AC, e as dificuldades encontradas. Por fim, recomendações para trabalhos futuros.

4.1 CONSIDERAÇÕES FINAIS

Este relatório apresentou as atividades desenvolvidas durante o período de estágio supervisionado no Instituto de Administração Penitenciário do Acre (IAPEN/AC), relatando o desenvolvimento de uma solução para o cadastro de visitantes e seus reeducandos.

O estágio nos permitiu aplicar os conhecimentos adquiridos durante a graduação, como desenvolver um software, também nos deu a oportunidade de passar por experiências de realizar trabalhos e problemas reais numa empresa real, como verdadeiros profissionais da área de tecnologia da informação.

Além da aplicação destes conhecimentos adquiridos, foram necessárias várias pesquisas de como funciona o leitor de código de barras e de como aplicar essa funcionalidade ao sistema, pois ele o utiliza para leitura da carteira de visitante.

Passamos por várias dificuldades durante o estágio supervisionado, um deles foi de como criar o código de barras na carteira e depois como integrar isso no sistema.

Mas, o estágio supervisionado foi concluído com êxito. Foram cumpridos todos os objetivos especificados no documento de requisitos, tanto os objetivos específicos quanto os gerais. O sistema foi implantado e testado obtendo o resultado esperado pelo cliente.

Por tanto o problema levantado durante a pesquisa foi solucionado. O problema era gerar e salvar os relatórios em pdf e manter a privacidade dos mesmos. A solução desse problema garante a privacidade desses documentos e sua integridade para que não sejam alterados como era antes, pois somente o administrador tem acesso a eles.

4.2 RECOMENDAÇÕES

O sistema atendeu todas as necessidades que foram especificadas pelo cliente, mas no final foi possível observar que no sistema caberiam outras funcionalidades que poderiam ser implementadas juntas àquelas já desenvolvidas. Uma delas seria o cadastro de biometria de visitantes, podendo assim evitar carteira de um visitante que não esteja cadastrado seja usada por outra pessoa a visitação aos reeducandos.

Uma outra recomendação para trabalhos futuros, seria o envio da senha cadastrada por e-mail evitando assim o transtorno de perder e recuperar depois, assim ajudaria na segurança da informação e autenticidade.

Referências

- [BOOCH1999] BOOCH, G. (1999). *The unified modeling user guide*. [S.I.] Pearson education Índia.
- [DATE2004] DATE, C. (2004). *Introdução a sistemas de banco de dados*. 8.ed. Pearson/Addison Wesley, Rio de Janeiro.
- [de Estado de Segurança Pública2018] de Estado de Segurança Pública, S. (2018). *Secretaria de Estado de Segurança Pública*.
- [Heuser2009] Heuser, C. A. (2009). *Projetos de bancos de dados*. volume 6.ed-dados eletrônicos-Porto Alegre Bookman.
- [José Augusto MANZANO1999] José Augusto MANZANO, S. S. V. M. (1999). *Estudo Dirigido Delphi7*. . Editora Érica Ltda., 2ª Edição, São Paulo.
- [LARMAN2002] LARMAN, C. (2002). *Utilizando UML e Padrões*. 3. ed. Bookman.
- [Melo2002] Melo, A. C. (2002). *Desenvolvendo Aplicações com UML*. 1º Edição, Brasport.
- [NAVATHE2011] NAVATHE, Shamkant B.; ELMASRI, R. (2011). *Sistemas de Banco de Dados*. 6. ed. Pearson - Addison Wesley.
- [PAULA FILHO2003] PAULA FILHO, W. P. (2003). *Engenharia de Software: Fundamentos, Métodos e Padrões*. LTC Editora.
- [PRESSMAN1995] PRESSMAN, R. S. (1995). *Engenharia de Software*. . Makron Books.
- [SOMMERVILLE2003] SOMMERVILLE, I. (2003). *Engenharia de Software*. Addison Wesley.
- [Wallon1999] Wallon, H. (1999). *Afetividade e processo ensino-aprendizagem*. Fenomenologia da percepção. São Paulo: Martins Fontes.

APÊNDICE A – DOCUMENTO DE REQUISITOS

UNIVERSIDADE FEDERAL DO ACRE

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

DOCUMENTO DE REQUISITOS DE SOFTWARE

SISTEMA DE CONTROLE DE VISITANTES DE PRESÍDIOS

VERSÃO 2.1

Desenvolvedores/Analistas

Gelio Marcos Vital Souza

Rita da Costa Pinto

Rio Branco – AC 2018

HISTÓRICO DE ALTERAÇÕES

Data	Versão	Descrição	Autor
16/06/2017	1.0	Versão inicial do Sistema	Gelio e Rita
03/06/2017	1.1	Fazendo os diagramas de classe e de casos de uso	Gelio e Rita
11/08/2017	2.0	Correção de erros e ajustes nos requisitos	Gelio e Rita
30/08/2017	2.1	Validação dos requisitos	Gelio e Rita

1. ANÁLISE DO PROBLEMA

Visa realizar o desenvolvimento de um sistema para o cadastro e controle dos reeducando e seus visitantes para o Instituto de Administração Penitenciário do Acre (IAPEN), de modo a corrigir a deficiência do sistema que é usado no momento no processo de controle. Criar um aplicativo desktop em Delphi, com leitor de código de barra, e gera um arquivo desktop.ini, com dados criptografado para conexão com banco de dados, ip.do banco, nome da base de dados, usuários e senhas.

Outro aplicativo pode se conecta ao banco de dados pelo arquivo desktop.ini ele com as funções de criar e administrar usuários, para controle de usuários no aplicativo. Cria histórico dos reeducandos, data de entrada data de saída em regime de monitoramento. Liberar algumas funções para os usuários do sistema.

2. NECESSIDADES BÁSICAS DO CLIENTE

Necessita de um software para ter um controle dos reeducandos e seus visitantes.

3. ESTUDO DE VIABILIDADE

O desenvolvimento do Sistema de Controle de Visitantes de Presídios do Acre é viável, pois além de suprir a necessidade do cliente atende os requisitos a seguir.

3.1. VIABILIDADE TÉCNICA

Existem todos os recursos técnicos necessários para a execução do projeto atendendo suas especificações pois, o órgão já possui a infraestrutura necessária para o funcionamento do sistema. O sistema será desenvolvido com tecnologias também já utilizadas no órgão.

3.2. VIABILIDADE ECONÔMICA

O projeto não possui investimento econômico, tudo será feito gratuitamente. Também não será necessário investir em infraestrutura de TI, pois os equipamentos atualmente utilizados no órgão são suficientes para implantação da solução.

3.3. VIABILIDADE LEGAL

O projeto não infringe nenhuma lei, portanto todos os recursos utilizados são permitidos pela legislação.

4. MISSÃO DO SOFTWARE

Manter o controle dos visitantes nos presídios, até o momento o sistema existente não cumpre totalmente com as necessidades do cliente, como por exemplo aplicar penalidade aos visitantes, ou mesmo histórico de visitas.

5. LIMITES DO SISTEMA

ID	Funcionalidade	Justificativa
L1	Controlar as visitas nos presídios.	O Sistema será exclusivo para controlar as visitas nos presídios e seus reeducandos.

6. BENEFÍCIOS GERAIS

ID	Benefícios
B1	Facilitar o controle de entrada e saída dos visitantes.
B2	Manter o controle dos reeducandos que estão sendo monitorados.
B3	Identificar as carteiras suspensas dos visitantes.
B4	Gerar relatórios para manter o controle das visitas.

7. RESTRIÇÕES

ID	Restrição	Descrição
R1	O sistema será Desktop	A pedido do cliente pois, nos presídios não tem acesso à internet.

8. ATORES

ID	Atores	Descrição
A1	Usuário Grupo1	Pessoa autorizada a usar o sistema para acesso à funções não gerenciais, cadastro de visitantes e gerar carteiras, geralmente funcionários que trabalham no IAPEN/AC.
A2	Usuário Grupo2	Funcionários do presídio que exerçam alguma função de relevância e que precise usar o sistema para tal, cadastro de reeducandos e controle de visitantes, com carteira.
A3	Usuário Grupo3	Funcionários do presídio ou IAPEN, pré-determinado pelo administrador “cliente” que exerçam alguma função de relevância e que vai usar o sistema para ativar ou desativar o visitante, de acordo com sua conduta.
A4	Administrador	Pessoas de nível administrativo superior que terá acesso a todas as funções disponíveis no sistema.

9. REQUISITOS FUNCIONAIS

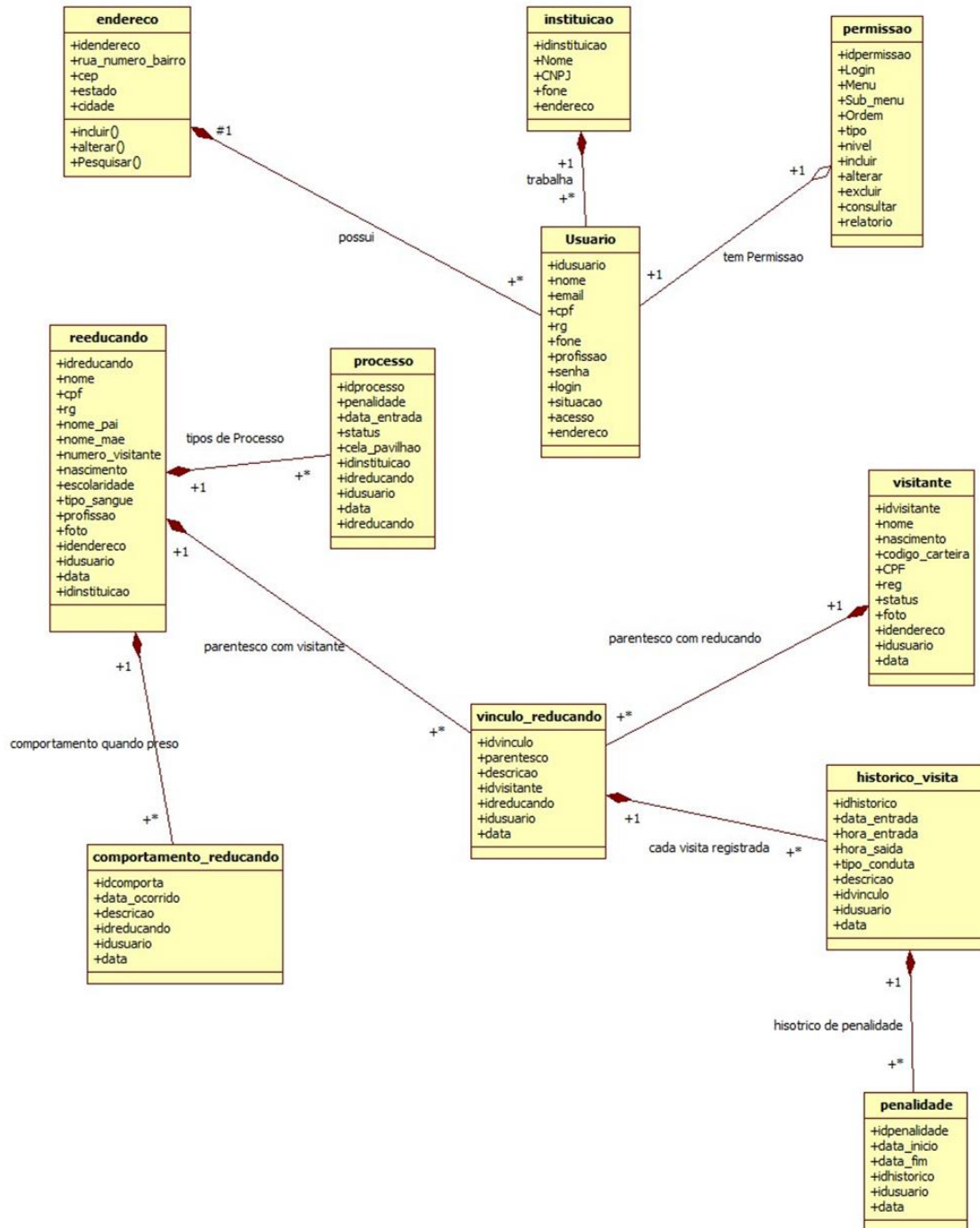
ID	Funcionalidade	Necessidades	Prioridade
RF1	Cadastrar usuário	Terá funções para cadastro, remoção (ativar e desativar usuário) e edição de dados desses usuários, bem como consultas. Os usuários podem ser alocados de acordo com seu grupo de suas funções.	Alta
RF2	Gerenciar os reeducandos	Terá funções para cadastro, e edição de dados desses reeducandos, bem como consultas. O sistema armazena os seguintes dados para cada reeducando: nome, CPF, RG, filiação, número de visitantes, endereço, escolaridade, tipo sanguíneo, profissão e foto.	Alta
RF3	Gerenciar visitantes	Há funções para cadastro, remoção e edição de dados de cada visitante, gerar carteiras funcionais para os visitantes, bem como consultas.	Alta
RF4	Gerenciar histórico de visitas	Terá os seguintes dados: reeducando visitado, data entrada, hora entrada, hora saída, tipo de conduta (normal, carteira falsa, armas, drogas, celulares), descrição (detalhar com clareza as condutas erradas do visitante). Todo tipo de conduta que não é normal, o sistema avisa automaticamente aos Usuário do Grupo 3, para desativar esses visitantes.	Alta
RF5	Gerenciar penalidades	O sistema prevê a inclusão da penalidade ao visitante, com data de início e data de fim e o visitante ficará com a carteira inativa	Alta
RF6	Gerar relatórios	O sistema irá gerar relatórios semanal, mensal ou diário de todas as condutas e penalidades dos visitantes.	Alta

10. REQUISITOS NÃO FUNCIONAIS

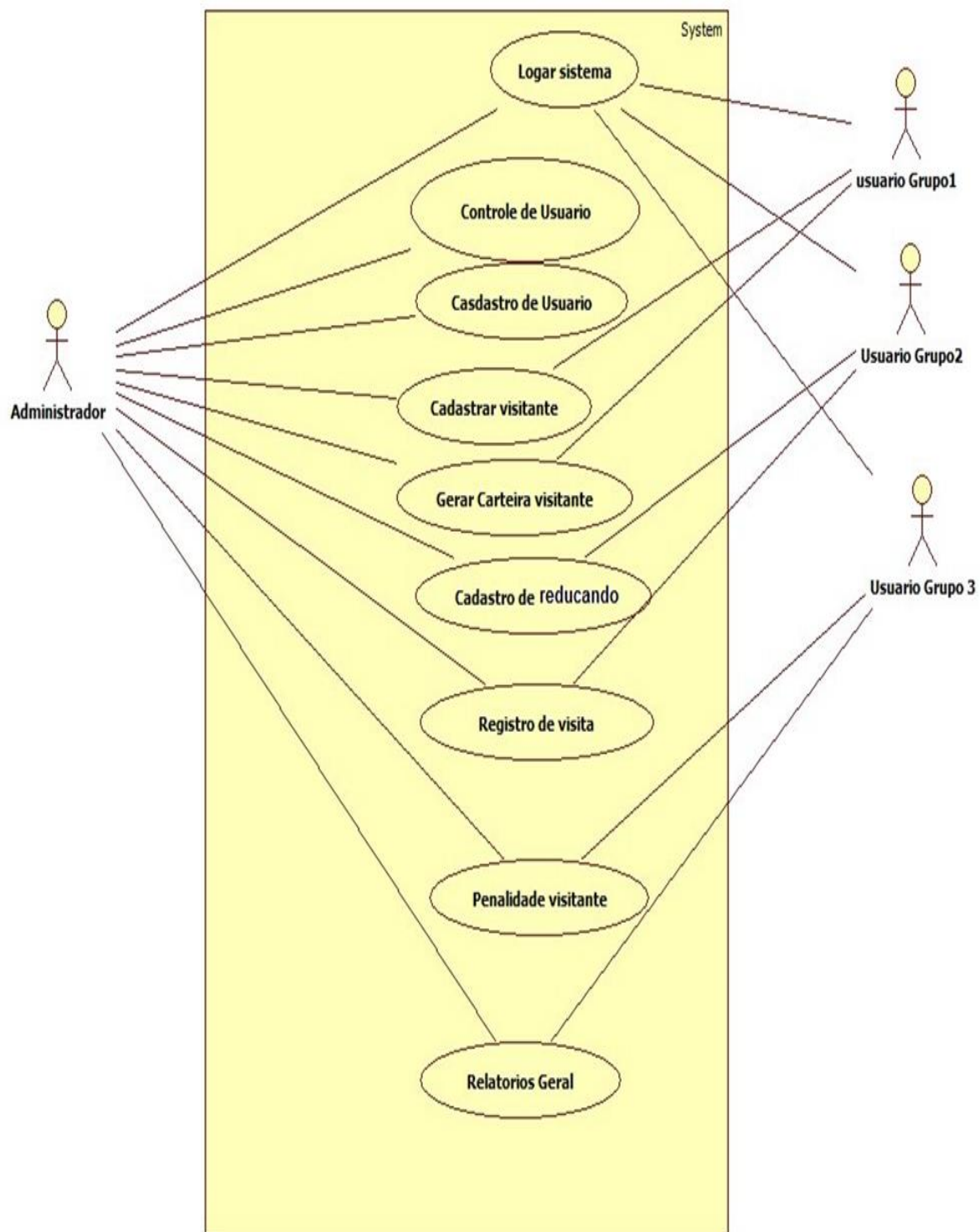
ID	Requisitos	Categoria
NRF1	Interface desktop	Segurança
NRF2	O acesso ao sistema deve ser por meio de autenticação de usuários previamente cadastrados e autorizados.	Segurança
NRF3	O sistema deve ser de fácil utilização, não exigindo de seus operadores conhecimentos avançados de informática.	Usabilidade
RF4	O sistema deve ser acessado em qualquer Sistema Operacional.	Portabilidade

11. DIAGRAMAS

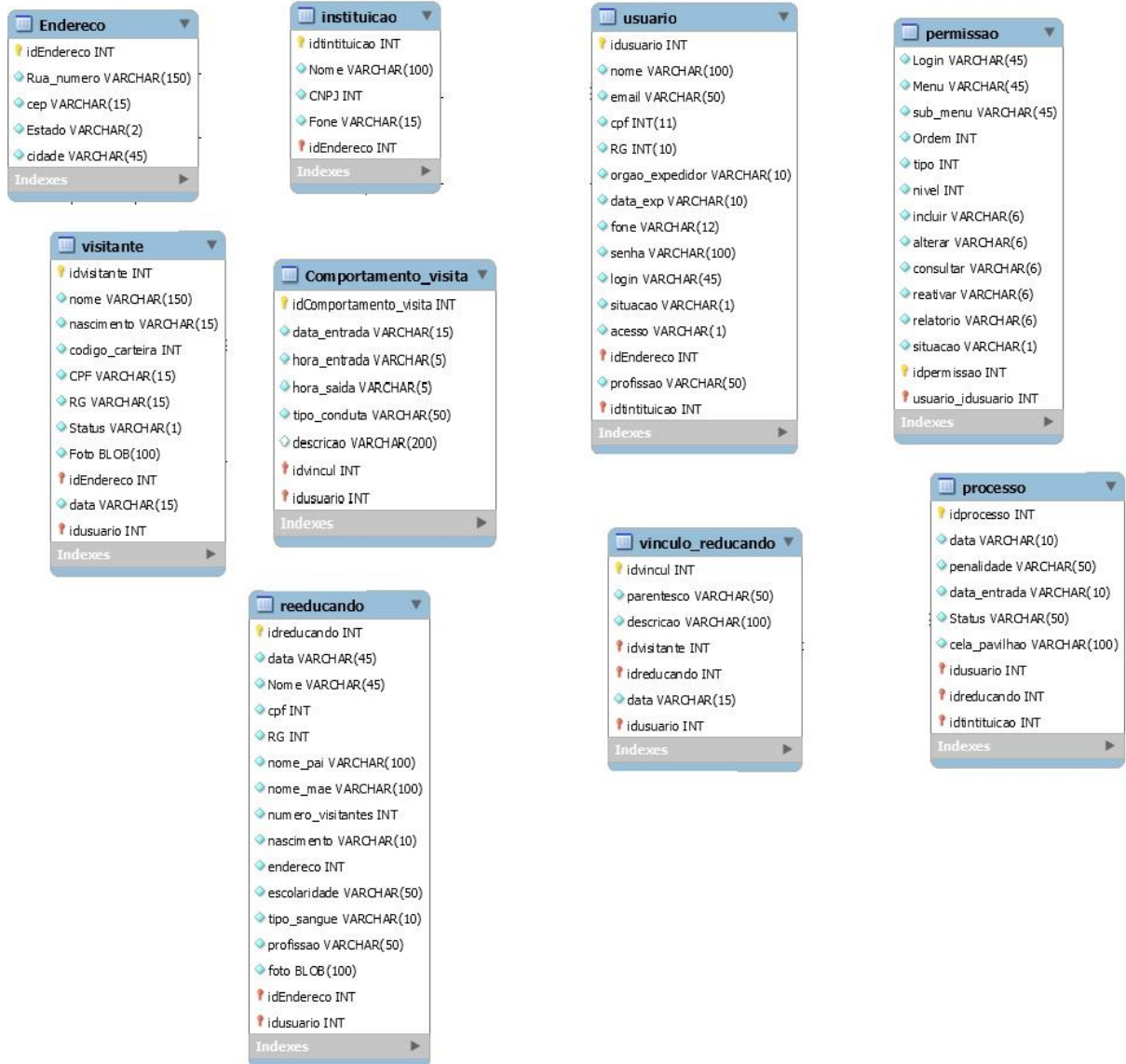
11.1 DIAGRAMAS DE CLASSE



11.2 DIAGRAMAS DE CASO DE USO



11.2 DIAGRAMAS DE ENTIDADES



12. REQUISITOS DE HARDWARE

12.1. CONFIGURAÇÃO MÍNIMA

- Sistema operacional: Windows 7 ou superior
- 512 MB de memória RAM;
- 1 GB de espaço em disco.

12.2. Configuração Recomendada

- Sistema operacional: Windows 10.
- Memória do sistema (RAM): 4 GB ou mais
- Disco: 10 GB de espaço livre ou mais.

13. FERRAMENTAS DE DESENVOLVIMENTO E LICENÇA DE USO

- Delphi7;
- MySQL.

APÊNDICE B-PROTÓTIPOS DO SISTEMA

PROTÓTIPOS DO SISTEMA

Sistema de Controle de Visitantes de presídios

Versão 2.1

Desenvolvedores/Analistas

Gélio Marcos Vital de Souza

Rita da Costa Pinto

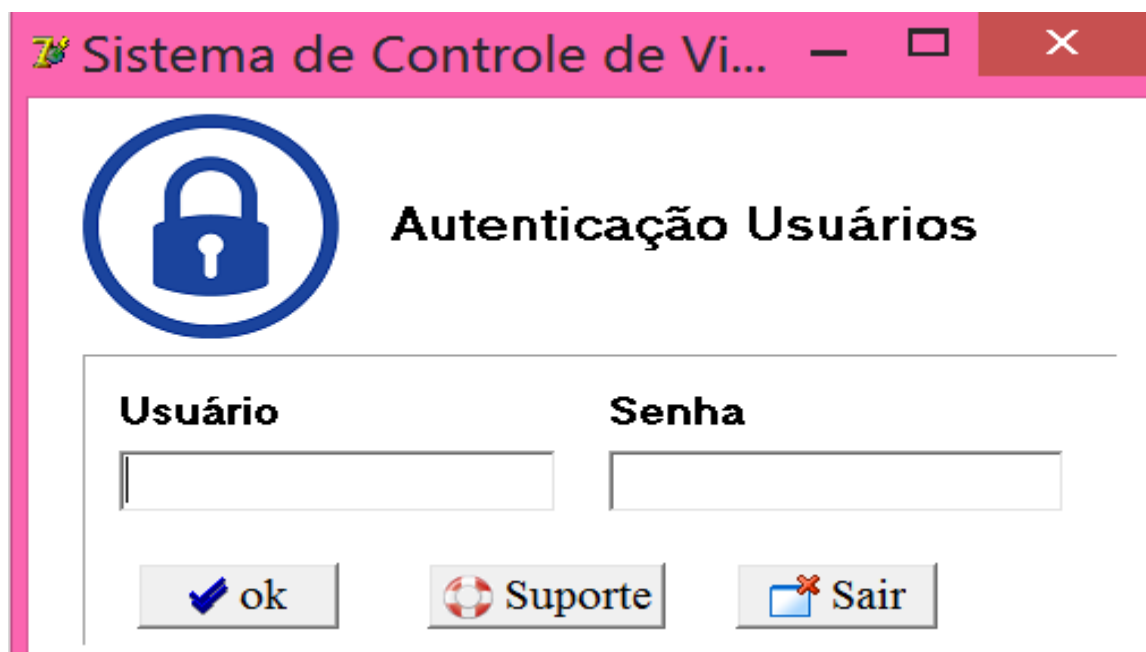
Rio Branco – AC 2018

Todos os protótipos apresentados a seguir, foram desenvolvidos de acordo com o documento de requisitos.

1. Tela de Apresentação do Sistema



2. Tela de Login



3. Tela de Cadastro de Usuários

Cadastro Relatorio Monitoramento Sair

Cadastro de Usuarios

Novo Deletar Editar Pesquisar Salvar Imprimir

Cadastro Usuario | Controle de Permissão

Cadastro de Usuario

Nome Completo:

Email:

CPF: RG: Instituição: Telefone: Função:

Endereço:

Login: ☐ Ativar

idusuario	nomecompleto	login
1	ADMINISTRADOR	ADMIN
2	GELIO VITAL	GELIO
3	vital	vital
4	III IRTA MENTES	III IRTA

4. Tela de Cadastro de Visitante

Cadastro Relatorio Monitoramento Sair

Cadastro de Visitante

Novo Editar Salvar Pesquisar Imprimir Tirar Foto Documentação Fazer Vinculo

FOTO

Nome Completo: ☐ Ativar Visitante

Data Nascimento: CPF ou Registro Nascimento: RG / ORG / UF: Perfil do Visitante: Telefone:

Endereço:

☐ Cadastro Ativo ☐ Cadastro Inativo

Pessoas Sugeridas pelos Reeducandos sem Cadastro:

nome

Cadastros dos Visitantes:

5. Tela de Cadastro de Reeducando

Cadastro Relatorio Monitoramento Sair

Cadastro Reeducando

Novo Editar salvar Pesquisar Imprimir deletar Foto

FOTO

Nome:

CPF: RG /ORG /UF: Data Nascimento: Tipo Sangu: Profissão:

Escolaridade: Endereço: Presídio:

Nome Pai:

Nome da Mãe:

Carregar Foto

Tipo de Regime e Visitante

Nome Visitante | Tipo de Regime | Cancelar Visitante

Nome do Reeducando:

Nome do Visitante:

Clique sobre o nome para carregar visitantes:

Ativar o Windows
Acesse as configurações do computador para
ativar o Windows.

6. Tela de Gerar Carteira

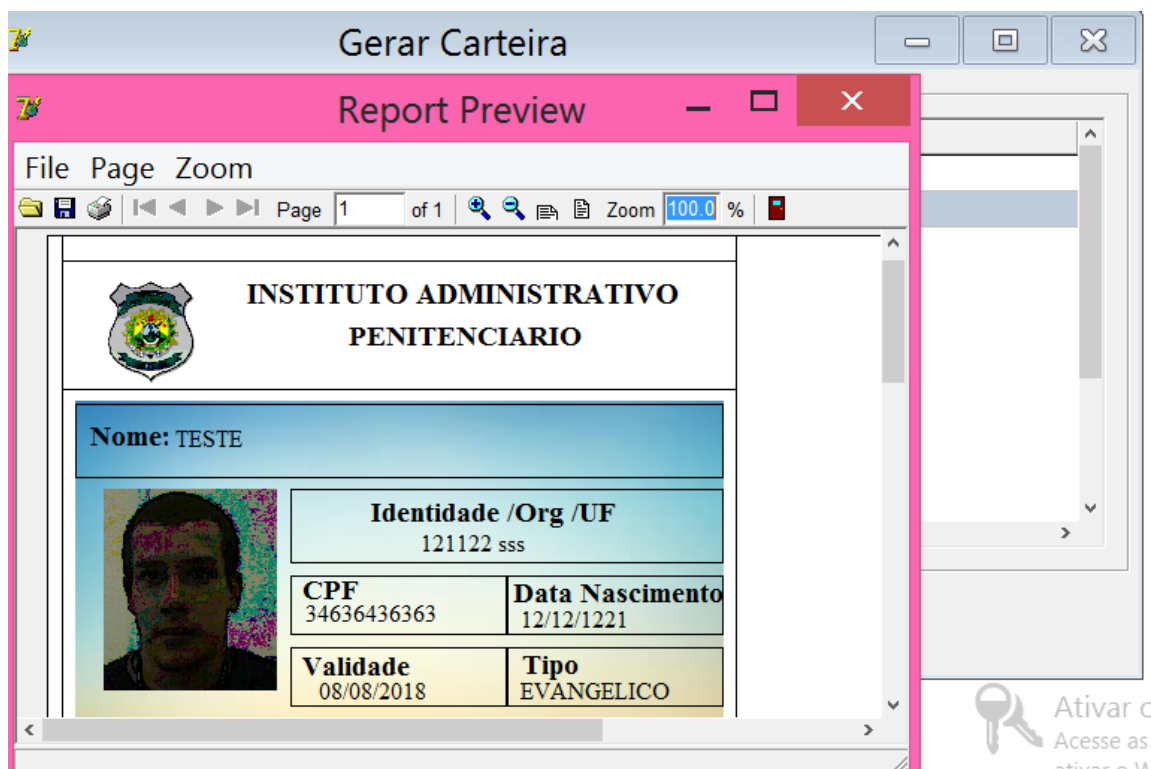
Gerar Carteira

Escolha o Nome do Visitante:

nome
TESTE
TADEU

Imprimir

7. Tela da carteira Gerada



Gerar Carteira

Report Preview

File Page Zoom

Page 1 of 1 Zoom 100.0 %

INSTITUTO ADMINISTRATIVO PENITENCIARIO

Nome: TESTE

Identidade /Org /UF
121122 sss

CPF
34636436363

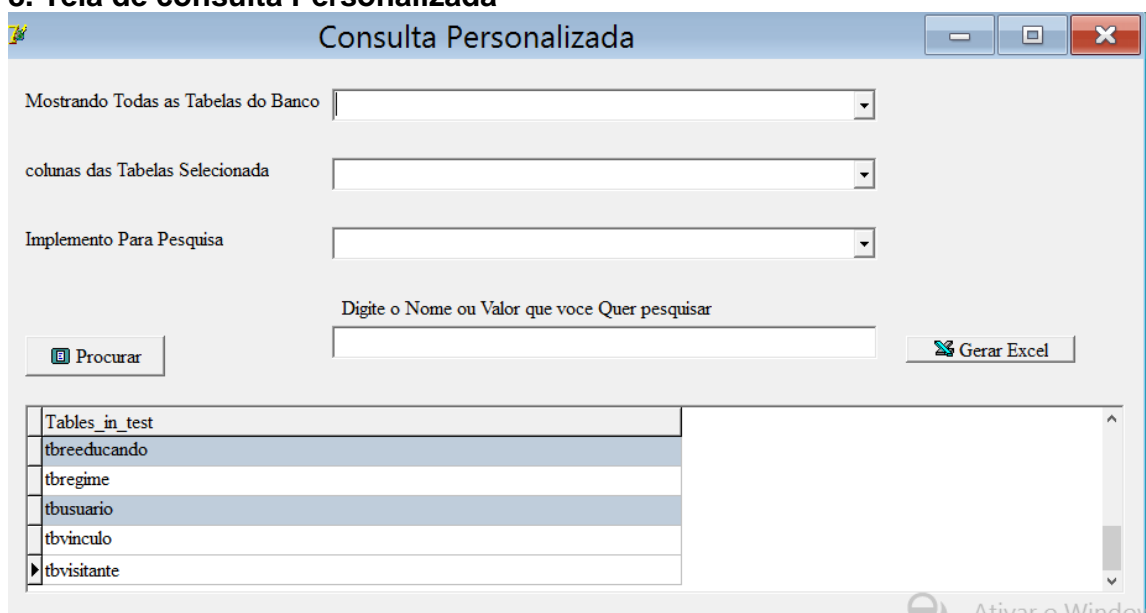
Data Nascimento
12/12/1221

Validade
08/08/2018

Tipo
EVANGELICO

Ativar c
Acesse as
ativar o W

8. Tela de consulta Personalizada



Consulta Personalizada

Mostrando Todas as Tabelas do Banco

colunas das Tabelas Seleccionada

Implemento Para Pesquisa

Digite o Nome ou Valor que voce Quer pesquisar

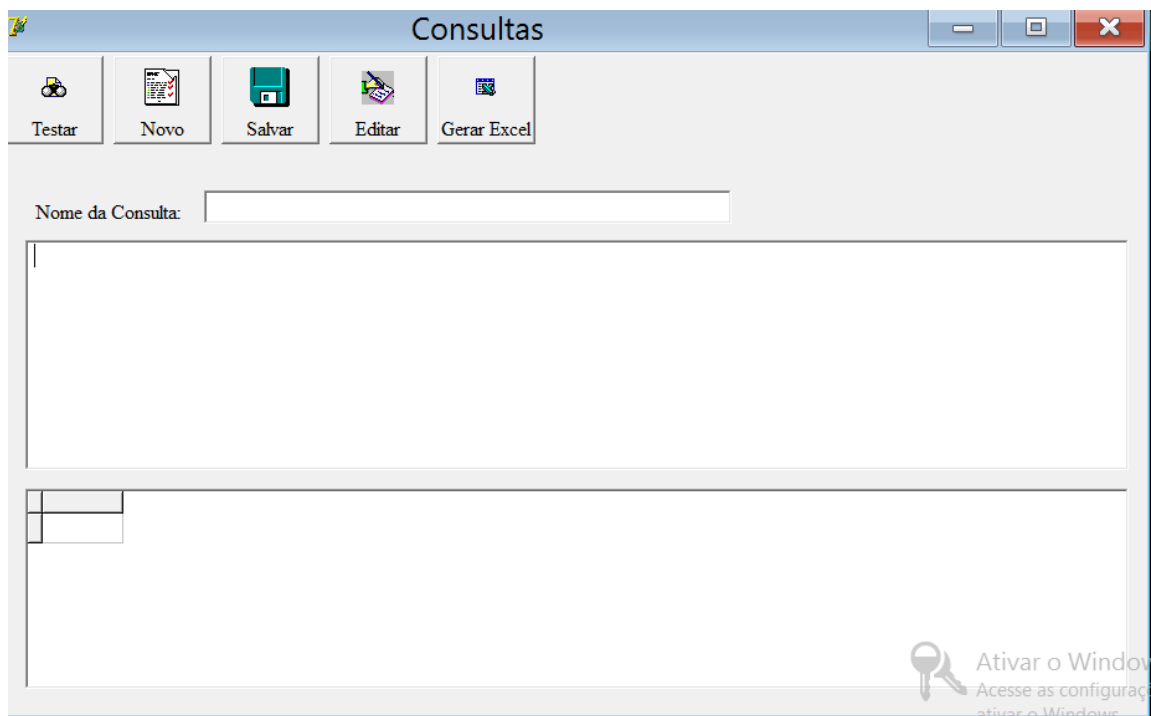
Procurar

Gerar Excel

Tables_in_test
tbreeducando
tbregime
tbusuario
tbvinculo
tbvisitante

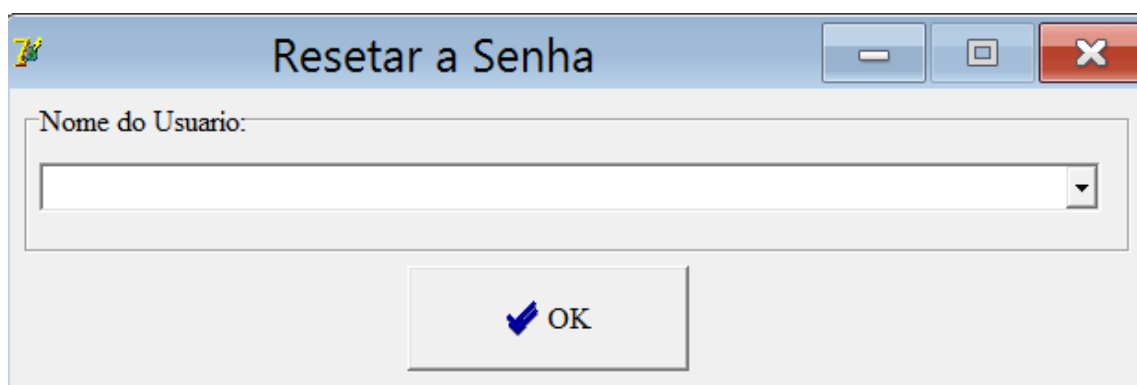
Ativar e Window

9. Tela de Consultas SQL



The screenshot shows a window titled "Consultas" with a standard Windows interface. Below the title bar is a toolbar with five buttons: "Testar" (with a magnifying glass icon), "Novo" (with a document icon), "Salvar" (with a floppy disk icon), "Editar" (with a pencil icon), and "Gerar Excel" (with an Excel icon). Below the toolbar is a label "Nome da Consulta:" followed by a text input field. Below the input field is a large, empty rectangular area, likely for displaying query results or SQL code. In the bottom right corner, there is a watermark that reads "Ativar o Windows" and "Acesse as configurações do Windows".

10. Tela de Resetar a Senha



The screenshot shows a window titled "Resetar a Senha" with a standard Windows interface. Below the title bar is a label "Nome do Usuario:" followed by a text input field with a dropdown arrow on the right. Below the input field is a large, empty rectangular area. At the bottom center, there is a button with a blue checkmark icon and the text "OK".

11. Tela de Monitoramento dos Reeducandos

The 'Monitoramento' window features a title bar with standard Windows controls. Below the title bar, there are two tabs: 'Controle de Entrada' (selected) and 'Controle de Saída'. The main area is divided into several sections:

- FotoGrafia do Visitante:** A large rectangular area for the visitor's photo.
- Codigo de Carteira:** A text input field.
- Escolhar Reeducando:** A dropdown menu.
- Tipo de Visita:** Radio buttons for 'intima', 'Normal', and 'Evangélica'. A checkbox labeled 'Houve Ocorrência' is also present.
- Hora Entrada:** A text input field showing '22:11:16'.
- Data:** A text input field showing '14/03/2018'.
- Descrição da Ocorrência:** A text area for describing the incident.
- Entrada:** A button with a camera icon and the text 'Entrada'.

At the bottom right, there is a Windows activation watermark: 'Ativar o Windows. Acesse as configurações do computador para ativar o Windows.'

12. Tela de Inativar Visitante

The 'Inativar Visitante' window has a title bar with standard Windows controls. It includes an 'Aplicar' button with a checkmark icon in the top left corner. The form contains the following fields:

- Nome do Visitante:** A text input field.
- CPF:** A text input field with a mask.
- RG:** A text input field.
- Data Nascimento:** A text input field.
- Ativar e Desativar Visitante:** A checkbox.
- Descrição da Ocorrência:** A text area.

At the bottom, there is a list box with the header 'nomevisitante'. It contains one entry, 'TADEU', which is currently selected and highlighted in blue.

At the bottom right, there is a Windows activation watermark: 'Ativar o Windows. Acesse as configurações do computador para ativar o Windows.'

Rio Branco-AC, 23 de Março de 2018.

Assinatura do aluno

Assinatura do aluno

Assinatura do Orientador