

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS

Vinicius Corrêa Menezes

Relatório de Estágio

**Desenvolvimento de um banco de dados para
classificação de imagens de Shearografia**

Florianópolis
28 de novembro de 2017

Vinicius Corrêa Menezes

Desenvolvimento de um banco de dados para classificação de imagens de Shearografia

Relatório submetido à Universidade Federal de Santa Catarina como requisito para aprovação na disciplina **DAS 5501: Estágio em Controle e Automação Industrial** do curso de Graduação em Engenharia de Controle e Automação.

Orientador(es): Prof. Armando Albertazzi Gonçalves Junior e Prof. Carlos Barros Montez

Florianópolis

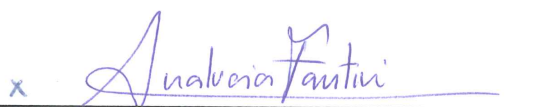
28 de novembro de 2017

Vinicius Corrêa Menezes

Desenvolvimento de um banco de dados para classificação de imagens de Shearografia

Este relatório de estágio foi julgado no contexto da disciplina DAS5501: Estágio em Controle e Automação Industrial e **APROVADO** na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, de de

x 

Orientador no Local de Trabalho

LabMetro, UFSC
Armando Albertazzi Gonçalves Jr, Dr. Eng.
Professor
LABMETRO/EMC/CTC/UFSC



Orientador

Universidade Federal de Santa Catarina

Resumo

A Shearografia é uma técnica óptica interferométrica capaz de medir campos de pequenas deformações ocorridas nas superfícies dos materiais. Ao contrário das demais técnicas interferométricas, a Shearografia é robusta o suficiente para ser operada fora do ambiente laboratorial. Apresenta aspectos positivos, como grande sensibilidade às deformações possibilitando a inspeção não destrutiva, sem contato, e propicia considerável velocidade na inspeção. Nos últimos anos, o LabMetro, através de projetos financiados pela Petrobras, tem utilizado a Shearografia para detectar e avaliar defeitos em revestimentos e reparos feitos em materiais compósitos e em estruturas de materiais compósitos aplicados na indústria do petróleo, gás e energia. O sucesso dos resultados obtidos em campo motivou a criação de um banco de dados para classificação das imagens obtidas através da técnica de Shearografia. A ideia é cadastrar, e assim poder relacionar, os diversos parâmetros envolvidos numa inspeção, como por exemplo: tipos de revestimento, espessuras de revestimento, formas e níveis de carregamento, tipos de defeitos, etc. O banco de dados será inicialmente desenvolvido apenas para consulta, mas já poderá oferecer inúmeras vantagens, como: (a) a centralização das informações em um servidor; (b) a possibilidade de consulta remota às imagens de Shearografia; (c) o acesso às informações sobre inspeções passadas; (d) a possibilidade de re-processar imagens antigas, caso seja do interesse ou seja identificado um novo algoritmo; (e) a disponibilização da informação para toda equipe de forma fácil, segura e organizada. Além de todas as vantagens já citadas, existe ainda um objetivo maior que motiva a criação de um banco de dados. Espera-se, num futuro próximo, utilizar as informações cadastradas no banco para alimentar algoritmos de redes neurais, os quais poderão auxiliar na identificação de defeitos em imagens de Shearografia.

Palavras-chave: Shearografia. Banco de dados. Inspeção. Redes Neurais.

Abstract

Shearography is an optical interferometric technique capable of measuring fields of small deformations occurred on the surfaces of materials. Unlike other interferometric techniques, Shearography is robust enough to be operated outside the laboratory environment. It presents positive aspects, such as high sensitivity to deformations, allowing non-destructive inspection, without contact, and provides considerable speed in inspection. In recent years, LabMetro, through projects financed by Petrobras, has used Shearography to detect and evaluate defects in coatings and repairs made in composite materials and structures of composite materials applied in the oil, gas and energy industry. The success of the results obtained in the field motivated the creation of a database to classify the images obtained through the technique of Shearography. The idea is to register, and thus be able to relate, the various parameters involved in an inspection, such as: types of coating, coating thicknesses, forms and loading levels, types of defects, etc. The database will initially be developed for consultation only, but may already offer numerous advantages, such as: (a) centralizing information on a server; (b) the possibility of remote consultation of Shearography images; (c) access to information on past inspections; (d) the possibility of re-processing old images, if it is of interest or a new algorithm is identified; (e) making the information available to the whole team in an easy, safe and organized way. In addition to all the advantages already mentioned, there is still a greater objective that motivates the creation of a database. In the near future, it is expected to use the information registered in the database to feed neural network algorithms, which may assist in the identification of defects in Shearography images.

Keywords: Shearography. Database. Inspection. Neural Network.

Lista de Figuras

Figura 1 – Sistema de Shearografia	9
Figura 2 – Resultado da Shearografia	10
Figura 3 – Sistema de banco de dados	18
Figura 4 – Toast Notifications	27
Figura 5 – MySQL Workbench - Tela inicial	29
Figura 6 – MySQL Workbench - Modelos	29
Figura 7 – MySQL Workbench - Modelo do sistema	30
Figura 8 – Ferramenta Astah	32
Figura 9 – Balsamiq Mockups - Exemplo de protótipo	33
Figura 10 – Diagrama de caso de uso - Sistema ShicDB	39
Figura 11 – Protótipo ShicDB - Tela Importar Imagem	40
Figura 12 – Modelo Entidade Relacionamento - ShicDB	42
Figura 13 – Modelo Lógico - Tabela Usuário	43
Figura 14 – Tela de Login - ShicDB	46
Figura 15 – Tela Home - ShicDB	46
Figura 16 – Tela de Cadastro - ShicDB	47
Figura 17 – Tela Home com menu aberto - ShicDB	48
Figura 18 – Tela de Importar Imagem - ShicDB	49
Figura 19 – Tela de Importar Imagem após primeiro cadastro - ShicDB	50
Figura 20 – Tela de Pesquisar - ShicDB	51
Figura 21 – Sincronização de dados referentes às imagens - ShicDB	51
Figura 22 – Tela de Pesquisar com imagem selecionada - ShicDB	52
Figura 23 – Tela de Editar - ShicDB	53
Figura 24 – Confirmação de exclusão do registro selecionado - ShicDB	54
Figura 25 – Tela de Upload - ShicDB	54
Figura 26 – Tela de Gerenciar usuários - ShicDB	56
Figura 27 – Tela de Gerenciar informações - ShicDB	56

Sumário

1	INTRODUÇÃO	9
1.1	Motivação	9
1.2	Objetivo geral	11
1.2.1	Objetivos específicos	11
1.3	Métodos	11
1.4	Descrição do laboratório	12
1.4.1	História	12
1.4.2	Missão	13
1.4.3	Visão	13
1.5	Organização do relatório	13
2	REVISÃO DA LITERATURA	15
2.1	Engenharia de software	15
2.2	Engenharia de requisitos	15
2.3	Prototipação	16
2.4	Banco de dados	17
3	MATERIAS, TÉCNICAS E MÉTODOS	21
3.1	Scrum	21
3.1.1	Vocabulário utilizado pelo Scrum	22
3.1.2	Scrum Solo	22
3.2	.NET Framework e Linguagem C#	23
3.3	Visual Studio	23
3.4	WPF e XAML	24
3.5	MahApps.Metro	24
3.6	Padrão MVVM	25
3.7	Entity Framework	25
3.8	RestSharp	26
3.9	Toast Notifications	26
3.10	MySQL	27
3.11	MySQL Workbench	28
3.12	PHP	28
3.13	Visual Studio Code	30

3.14	JSON	31
3.15	Postman	31
3.16	Filezilla	31
3.17	GIT	31
3.18	Astah	32
3.19	Balsamiq Mockups	32
4	PROJETO E IMPLEMENTAÇÃO	35
4.1	Engenharia de requisitos	35
4.1.1	Entrevista	35
4.1.2	Documentação de requisitos de software	36
4.1.2.1	Objetivo do documento	36
4.1.2.2	Escopo do sistema	36
4.1.2.3	Descrição geral do sistema	36
4.1.2.4	Perspectiva do sistema	37
4.1.2.5	Características dos usuários	37
4.1.2.6	Requisitos	37
4.1.2.6.1	Requisitos funcionais (RF)	37
4.1.2.6.2	Requisitos não funcionais (NF)	38
4.1.3	Diagramas de caso de uso	39
4.2	Protótipo	40
4.3	Modelagem do banco de dados	41
4.3.1	Modelo entidade relacionamento (ER)	41
4.3.2	Modelo lógico	41
5	RESULTADOS	45
5.1	Princípios do sistema	45
5.2	Funcionalidades do sistema	45
5.2.1	Login	45
5.2.2	Cadastro	47
5.2.3	Home	47
5.2.4	Importar Imagem	48
5.2.5	Pesquisar	50
5.2.5.1	Editar	52
5.2.5.2	Excluir	53
5.2.6	Upload	53
5.2.7	Download	55

5.2.8	Gerenciar usuários	55
5.2.9	Gerenciar informações	55
5.3	Funcionalidades do servidor	57
5.3.1	Autenticação	57
5.3.2	Sincronizar informações	57
5.3.3	Pesquisar imagens	57
5.3.4	Download de imagens	58
5.3.5	Excluir imagem	58
5.3.6	Atualizar imagem	58
5.3.7	Upload de imagens	58
5.3.8	Gerenciar informações	59
5.3.9	Gerenciar usuários	59
6	CONSIDERAÇÕES FINAIS	61
	REFERÊNCIAS	63

1 Introdução

Neste capítulo é apresentada a motivação do trabalho realizado, seus objetivos e os métodos utilizados.

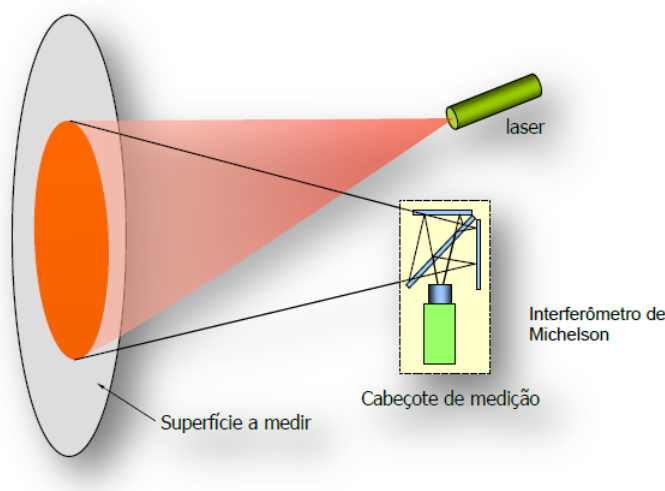
1.1 Motivação

A metrologia óptica está se destacando cada vez mais entre os métodos de medição. Diversos sensores destinados a medição de grandezas físicas e químicas são construídos baseados em princípios ópticos.

A Shearografia é uma das técnicas, dentro da metrologia óptica, em estudo no Laboratório de Metrologia e Automatização (LabMetro). Ela é uma técnica interferométrica reconhecida pela Sociedade Americana de Ensaaios Não-destrutivos ASNT (do inglês: American Society for Nondestructive Testing) que permite realizar medições de pequenas deformações em estruturas fabricadas e/ou revestidas por material compósito.

A Shearografia é uma técnica de inspeção óptica baseada nos princípios da interferometria. A configuração mais comum é composta por uma fonte de luz laser, um Interferômetro de Michelson [1] e uma câmera, conforme mostra a Figura 1.

Figura 1 – Sistema de Shearografia



Fonte: Referência [2]

A câmera observa a estrutura iluminada através do interferômetro de Michelson, o qual gera uma “dupla imagem”, ou imagem de interferência com deslocamento lateral. Essa

imagem é formada no CCD da câmera pela sobreposição de duas imagens deslocadas lateralmente pelo Interferômetro de Michelson. Para medição com Shearografia, um conjunto de imagens de interferência são adquiridas antes e após um carregamento aplicado na estrutura a ser inspecionada. Esse carregamento deve ser capaz de provocar pequenas deformações na estrutura. As imagens com e sem carregamento são processadas através de algoritmos específicos e geram como resultado uma imagem de diferenças de fase. A Figura 2 mostra o resultado da medição de um corpo de prova, no qual foi utilizado carregamento térmico.

Figura 2 – Resultado da Shearografia



Fonte: Referência [3]

A Shearografia revela, através das imagens processadas, a presença de defeitos (delaminações, trincas, furos, etc) na estrutura inspecionada. A Figura 2 mostra um exemplo dos padrões de franjas características da Shearografia.

A técnica tem apresentado bons resultados na inspeção de estruturas metálicas revestidas em material compósito e pode ser considerada uma potente ferramenta para ser aplicada em ensaios não-destrutivos.

A inspeção com Shearografia em campo gera bons resultados devido a elevada sensibilidade para medir pequenas deformações e sua relativa imunidade a perturbações, pois consegue tolerar níveis moderados de vibrações [3].

O sucesso dos resultados obtidos em campo utilizando esta técnica motivou o desenvolvimento deste trabalho, o qual busca a criação de um banco de dados para classificação das imagens obtidas através da Shearografia. A ideia central do trabalho é desenvolver um

sistema de cadastro destas imagens, e assim poder relacionar, os diversos parâmetros envolvidos numa inspeção, como por exemplo: tipos de revestimento, espessuras de revestimento, formas e níveis de carregamento, tipos de defeitos, etc.

1.2 Objetivo geral

O objetivo geral deste trabalho é desenvolver um sistema que salve as diversas informações relacionadas às imagens obtidas através da Shearografia em um banco de dados. O desenvolvimento deste sistema visa facilitar o acesso às imagens que atualmente se encontram espalhadas em diversos dispositivos, em outras palavras, o sistema visa centralizar as informações em um servidor onde possibilitará aos usuários do sistema acessar estas informações de forma fácil, segura e organizada.

1.2.1 Objetivos específicos

Em termos específicos, os objetivos do sistema que será desenvolvido durante o estágio são:

- Importar imagens processadas pela técnica de Shearografia e cadastrar informações relacionadas às mesmas
- Restringir o acesso à certas funções do sistema por tipo de usuário
- Realizar sincronização com o servidor
- Filtrar imagens de acordo com as informações desejadas

1.3 Métodos

Para atingir os objetivos propostos neste estágio, o trabalho foi dividido em dez etapas:

1. Estudo e seleção do banco de dados utilizado no desenvolvimento
2. Estudo e levantamento dos parâmetros envolvidos em uma inspeção de Shearografia
3. Entrevista com equipe de projeto para identificação das necessidades
4. Projeto de software, incluindo o projeto de uma interface amigável para o usuário

5. Implementação das rotinas de software, às quais abrangem as seguintes tarefas: conexão à base de dados, restrição de acesso por tipo de usuário, cadastramento, leitura e gravação das informações
6. Desenvolvimento de uma versão piloto para testes de validação
7. Cadastramento, no banco de dados, de informações já existentes
8. Testes de validação e desempenho do banco de dados
9. Otimizações
10. Documentação contendo detalhes do desenvolvimento

1.4 Descrição do laboratório

1.4.1 História

O LabMetro faz parte do Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina. Seu histórico teve início no ano de 1970, quando, o então professor dessa Universidade, Jaroslav Kozel, introduziu a disciplina de Metrologia no curso de pós-graduação em Engenharia Mecânica, devido a uma procura por profissionais com experiência nessa área. Em 1976, o mesmo professor criou, agregando alguns equipamentos e adquirindo outros, o Laboratório de Metrologia. Este laboratório foi o início do que mais tarde se tornou o LabMetro [4]. Os anos se passaram, o Laboratório cresceu e, em 1979, com o retorno do Professor Carlos Alberto Schneider, que estava em doutoramento na Alemanha, foi criado, juntamente com os professores Kozel e Hermann Lücke, o Laboratório de Metrologia e Medição Mecânica (LMMM). Em 1980, em virtude do trágico falecimento do Professor Kozel, o Professor Schneider assumiu a coordenação do LMMM, para em 1981 transformá-lo, em uma ação estratégica, no LabMetro - Laboratório de Metrologia e Automatização.

A partir de então, a ênfase dos trabalhos desenvolvidos no LabMetro passou a ser a Metrologia Mecânica, embora vários princípios sejam envolvidos. Passaram a ser abordados desde aspectos ligados à avaliação da incerteza de medição, em diferentes níveis de profundidade, ao desenvolvimento de técnicas e sistemas de medição avançados.

O LabMetro está fortemente envolvido em projetos de pesquisa e de extensão, sempre com o foco na metrologia. Alguns projetos envolvem a criação, o aperfeiçoamento, a avaliação e otimização de sistemas de medição destinados a atender certas classes de aplicações ou para atender particularidades. Outros projetos têm foco na gestão da metrologia e na garantia da qualidade metrológica ou da produção. A parceria com empresas é uma componente presente em quase todos os projetos desenvolvidos.

Atualmente o LabMetro ocupa uma área de 980 m^2 dentro do prédio da Fundação CERTI, com quem compartilha um moderno prédio de quatro andares que reúne uma das melhores infraestruturas em metrologia do país.

1.4.2 Missão

Formar pós-graduados de alto-nível, desenvolver e aplicar conhecimentos científicos e tecnológicos em prol do avanço da Metrologia no Brasil.

1.4.3 Visão

Manter-se reconhecido internacionalmente pelo elevado nível de excelência e formará os melhores, mais disputados e mais bem remunerados profissionais em Metrologia do mercado de trabalho.

1.5 Organização do relatório

Este relatório de estágio está organizado da seguinte maneira:

O Capítulo 2 tratará resumidamente de alguns conceitos, técnicas e fundamentação teórica, básicos para compreensão das atividades desenvolvidas durante o estágio.

O Capítulo 3 abordará as tecnologias, ferramentas e técnicas que foram utilizadas para o desenvolvimento do software desktop e do servidor.

O Capítulo 4 abordará mais detalhadamente o processo de levantamento dos requisitos para o desenvolvimento do sistema.

O Capítulo 5 descreverá os resultados obtidos ao final deste estágio.

2 Revisão da literatura

Nesta seção serão abordados alguns conceitos, técnicas e fundamentação teórica, básicos para compreensão das atividades desenvolvidas durante o estágio. Os conhecimentos adquiridos durante o curso foram a base para o desenvolvimento das atividades descritas neste trabalho.

2.1 Engenharia de software

Segundo Sommerville [5] a engenharia de software tem por objetivo apoiar o desenvolvimento de um software, incluindo técnicas, projeto e evolução dos sistemas. É uma área da engenharia que se preocupa com os aspectos da produção de um software.

A engenharia de software surgiu para corrigir problemas com relação ao desenvolvimento de projetos de software. A partir desse surgimento modelos de processos e guia de desenvolvimentos foram criados para otimização do processo de desenvolvimento do software [6].

O cientista de computação Friedrich Ludwig Bauer em 1969 disse que a engenharia de software “é a criação e a utilização de sólidos princípios de engenharia a fim de obter software econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais”. Já a IEEE desenvolveu uma definição mais abrangente que é “aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção do software, isto é a aplicação da engenharia ao software” [7].

Para Pressman [6]: “a engenharia de software é uma tecnologia em camadas que deve estar fundamentada em um comprometimento organizacional com a qualidade, para promover um aperfeiçoamento contínuo dos processos. Onde a base da engenharia de software é a camada de processos, que mantém as camadas de tecnologia coesas e possibilita o desenvolvimento de software de forma racional e dentro do prazo, que é relacionado com o controle do gerenciamento de projetos de software e estabelece o contexto no qual são aplicados métodos técnicos. Assim, fornecendo as informações técnicas necessárias para o desenvolvimento de um software”.

2.2 Engenharia de requisitos

A engenharia de requisitos é uma ação da engenharia de software que inicia durante a atividade de comunicação e continua na de modelagem. Ela deve ser adaptada às necessidades

do processo, do projeto, do produto e das pessoas que estão realizando o trabalho.

Segundo Sommerville [5] a engenharia de requisitos é o conjunto de descrições do que o sistema deve fazer; os serviços que serão oferecidos e as restrições de seu funcionamento. Esses requisitos refletem as necessidades dos clientes para que o sistema possa atender às funcionalidades específicas.

Por sua vez, Pressman [6] descreve que a engenharia de requisitos fornece mecanismo apropriado para entender aquilo que o cliente deseja, analisando as necessidades, avaliando a viabilidade, negociando uma solução razoável, especificando a solução sem ambiguidade, validando a especificação e gerenciando as necessidades à medida que são transformadas em um sistema operacional.

2.3 Prototipação

A prototipação ajuda a entender qual o propósito do que foi desenvolvido, o negócio do cliente, a proposição de melhorias e a minimização de riscos e maximização de lucros.

De acordo com Engholm [8], o protótipo da solução é um instrumento extremamente útil e importante para validação de requisitos. Esses protótipos servem para demonstrar o sistema, como será a navegação entre as interfaces, os relatórios previstos e assim por diante, reproduzindo o comportamento do futuro sistema a ser implementado.

Outra vantagem do protótipo, é que ele permite que sejam revelados erros cometidos nos requisitos propostos inicialmente, pois, com o passar do tempo, o usuário muitas vezes percebe que estava com uma visão incorreta ou incompleta do que realmente necessitava. Desta forma, a especificação do sistema pode ser modificada para o correto entendimento dos requisitos.

Segundo Sommerville [5], “um protótipo é uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções.” Este processo tem como meta facilitar o entendimento dos requisitos, apresentando conceitos e funcionalidades do software. Desta forma, podemos elucidar melhor e sermos mais assertivos no que o cliente realmente quer antecipadamente, o que consequentemente aumentará sua percepção de valor.

Segundo Guedes [9], prototipação ilustra como seriam as telas do sistema, apresentando informações de como seria inserida as informações e como seria apresentada as informações do sistema para o usuário. Ele ainda diz que com a prototipação se evita um longo tempo de desenvolvimento e implantação, evitando que o sistema não atenda as necessidades do cliente.

Por sua vez Pressman [6], diz que a prototipação começa com a comunicação com o cliente. Reuniões com stakeholders para definir os objetivos gerais do software.

Para Nascimento [10] são grandes as vantagens da prototipação:

- Baixa demanda de tempo para desenvolvimento e consequentemente, baixo custo
- Não requer conhecimentos avançados em softwares de edição gráfica
- Facilita a visualização do produto para o cliente desde a fase inicial
- Possibilita receber o feedback do cliente em tempo ágil
- Facilita o levantamento de requisitos e funcionalidades
- Possibilita estimar de forma mais precisa a complexidade e tempo e desenvolvimento
- Possibilita a realização testes de interações
- Reduz os esforços de desenvolvimento

Finalizando, para Ogedebe e Jacob [11] a prototipação é usada como ferramenta de comunicação entre desenvolvedores e cliente, permitindo que os clientes interajam com o protótipo do sistema, para que tenha todas as necessidades atendidas.

2.4 Banco de dados

Silberschatz, Korth e Sudarshan [12] definem banco de dados como uma coleção lógica de dados relacionados e coerentes que possuem significado implícito e representam aspectos do mundo real.

Em outras palavras banco de dados é uma coleção de dados inter-relacionados representando informações sobre um específico domínio, sempre que possível agrupar informações que se relacionam e tratam do mesmo assunto, pode-se dizer que é um banco de dados.

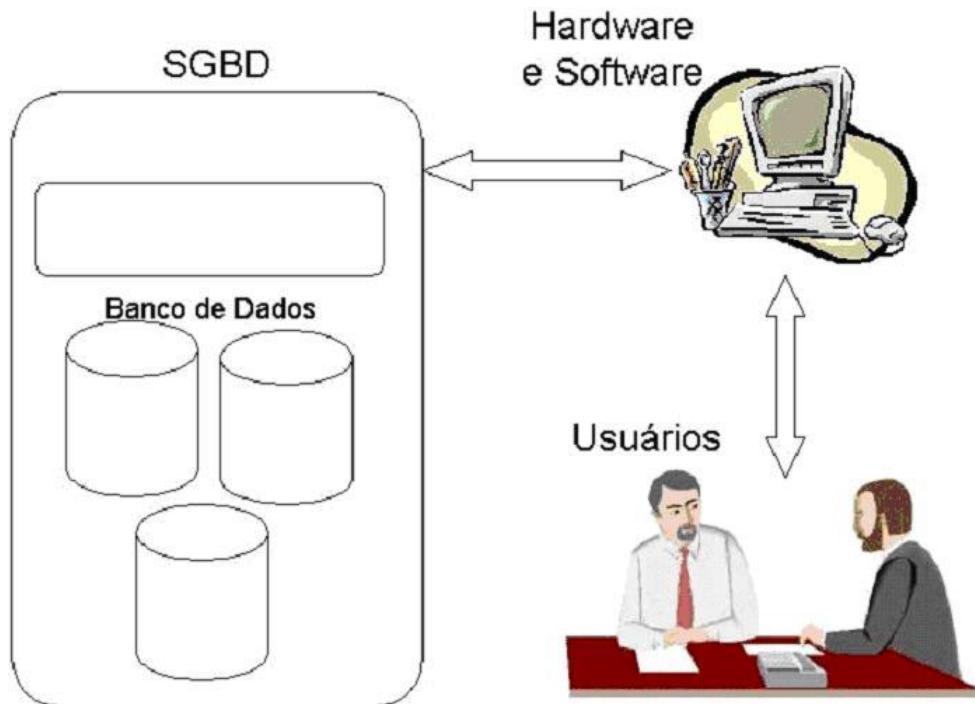
Por exemplo, a lista de nomes, números telefônicos e endereço de pessoas que conhecemos, esses dados podem ser escritos em uma agenda de telefones e armazenado em um computador. Essas informações são uma coleção de dados com um significado implícito, consequentemente banco de dados.

Por sua vez, Date [13] conceituou que sistema de banco de dados é um conjunto de componentes básicos: dados, hardware, software e usuários. Conforme ilustrado na Figura 3.

Existem diferentes modelos de banco de dados para Silberschatz, Korth e Sudarshan [12], esses modelos são:

Modelo relacional: Esse modelo relacional usa uma coleção de tabelas para representar os dados e as relações entre eles, onde cada tabela possui diversas colunas, e cada

Figura 3 – Sistema de banco de dados



Fonte: Referência [13]

coluna possui um nome específico, modelo relacional é um exemplo de modelo baseado em registros, esse modelo é o mais usado nas maiorias dos sistemas de banco de dados atuais.

Modelo de entidade/relacionamento: O modelo entidade/relacionamento (ER) é baseado em uma percepção de um mundo real que consiste em uma coleção de objetos básicos.

Modelo de dados baseado em objeto: Modelo de dados baseado em objeto pode ser visto como uma extensão ao modelo ER com noções de encapsulamento, métodos (funções) e identidade de objeto. O modelo de dados relacional de objeto combina recursos do modelo de dados orientado a objeto e do modelo de dados relacional.

Modelo de dados semiestruturado: Modelo de dados semi estruturado permite a especificação dos dados que itens de dados individuais do mesmo tipo possam ter diferentes conjuntos de atributos.

Normalmente os bancos de dados são operados por SGDB (Sistema Gerenciador de banco de Dados), segundo Elmasri e Navathe [14] SGDB é uma coleção de programas que permitem o usuário manipular as informações, criar, manter um banco de dados e interagir com o usuário.

O SGDB é, portanto, um sistema de software que facilita os processos de definição, construção, manipulação e compartilhamento de banco de dados entre vários usuários e

aplicação.

Segundo Silberschatz, Korth e Susarshan [12], um sistema de banco de dados fornece uma linguagem de definição de dados para especificar o esquema de banco de dados DDL (Data Definition Language) e uma linguagem de manipulação de dados para expressar as consultas e atualizações de banco de dados DML (Data Manipulation Language), essas duas linguagens formam partes de uma única linguagem de banco de dados, a linguagem SQL (Structured Query Language).

A SQL é uma linguagem de consulta estruturada padrão do banco de dados, SQL foi desenvolvido pela IBM no início dos anos 70, que tinha por objetivo demonstrar a viabilidade da implementação do modelo relacional. SQL é uma linguagem universal para todos os bancos de dados. A linguagem SQL é um padrão ANSI (American National Standards Institute), para ser compatível com o padrão ANSI a linguagem deve possuir pelo menos, os comandos principais como (Select, Update, Delete, Insert), assim SQL tornou-se a linguagem mais utilizada para os SGBD.

3 Materias, Técnicas e Métodos

Neste capítulo serão abordadas as tecnologias, ferramentas e técnicas que foram utilizadas para o desenvolvimento das atividades que compuseram o estágio. O desenvolvimento da parte estrutural do sistema desktop foi feito utilizando a tecnologia C#, o desenvolvimento da parte do servidor foi feito utilizando a tecnologia PHP e em ambas as partes os dados foram armazenados utilizando o banco de dados MySQL. Nas próximas seções serão abordadas todas as bibliotecas, metodologias, frameworks e tecnologias que auxiliaram no desenvolvimento do software.

3.1 Scrum

Para o desenvolvimento do sistema foi adotado o framework ágil de desenvolvimento de software Scrum. O Scrum foi escolhido por ser mais simples em relação às outras metodologias ágeis e de fácil adaptação já que todo o sistema foi desenvolvido por apenas um desenvolvedor.

O Scrum foi concebido em 1993 por Jeff Sutherland da necessidade de encontrar uma metodologia que abordasse o problema do desenvolvimento de software de uma forma não tradicional [15], em 1995 Ken Schwaber refinou a metodologia baseado na sua própria experiência no desenvolvimento de sistemas e processos.

O Scrum é um conjunto de atividades e práticas de gestão que devem ser seguidas para garantir que no final do projeto seja entregue um produto como especificado pelo cliente. Ela segue o paradigma iterativo e incremental onde a cada iteração pequenas partes do sistema são entregues. Isso é importante pois garante que teremos partes funcionais do sistema e essas passarão por todas as atividades até a implantação. As principais características do Scrum segundo Ferreira [15] são:

- É um processo ágil para gerenciar e controlar o desenvolvimento de projetos
- É um wrapper para outras práticas de engenharia de software; é um processo que controla o caos resultante de necessidades e interesses conflitantes
- É uma forma de aumentar a comunicação e maximizar a cooperação
- É uma forma de detectar e remover qualquer impedimento que atrapalhe o desenvolvimento de um produto
- É escalável desde projetos pequenos até grandes projetos em toda empresa

3.1.1 Vocabulário utilizado pelo Scrum

Time Scrum: No Time Scrum não existe um líder para ditar as regras do desenvolvimento, o time é autogerido. Por isso, a opinião de todos os envolvidos é importante para o progresso do projeto.

Scrum Master: O Scrum Master não é o líder do time já que o scrum não possui um, mas sim a pessoa com bom conhecimento técnico em Scrum que organiza o projeto facilitando o andamento e garantindo que todos cumpram suas tarefas.

Product Owner: É o responsável por traduzir as histórias do cliente em requisitos para que a equipe possa trabalhar no Sprint. Equipe de desenvolvimento: São aqueles que participarão na codificação do Produto.

Eventos: São pequenas conversas com tempo fixo, objetiva e ágil.

Sprint: É um ciclo de desenvolvimento ou em outras palavras um tempo determinado para entregar uma tarefa. No final de cada Sprint é entregue algo concreto.

Reunião de planejamento: Reuniões realizadas para definir os objetivos do Sprint. Para calcular a quantidade máxima de horas de cada reunião multiplica-se cada semana do Sprint por 2, se um Sprint tem duração de 1 semana, por exemplo, a reunião terá no máximo 2 horas de duração.

Reunião diária: Reunião com no máximo 15 minutos para revisar o andamento do projeto.

Revisão da Sprint: Revisar os itens desenvolvidos e mostrar as novas funcionalidades.

Retrospectiva da Sprint: Serve para fazer um balanço geral da Sprint e verificar o que funcionou bem, as melhorias que tem que ser feitas, os erros e o que será feito para melhorar.

Product backlog: É uma lista com tudo que o cliente desejaria como funcionalidade. É criado pelo product owner e sofre mudanças a cada solicitação do cliente.

Sprint backlog: É uma lista com o que deve ser feito na próxima Sprint.

3.1.2 Scrum Solo

Scrum solo não é uma nova metodologia, é apenas um denominação dada a uma adaptação do Scrum para projetos que possuem apenas um desenvolvedor. Assim, a pessoa envolvida no projeto tem a liberdade de remover algumas coisas desnecessárias do Scrum quando se está trabalhando sozinho.

Durante o desenvolvimento do sistema, foi utilizada a seguinte adaptação do Scrum:

- Uma lista das funcionalidades do sistema (Product backlog)

- Algumas funcionalidades selecionadas para desenvolver (Sprint backlog)
- Controle das atividades utilizando um quadro Kanban
- Cada produto entregue no final da Sprint era validado por bolsistas do projeto *SHIC* e pelo orientador do estágio
- Reuniões com o pessoal do projeto *SHIC* para reavaliar o Product backlog

3.2 .NET Framework e Linguagem C#

Para o desenvolvimento do sistema desktop adotou-se a plataforma de desenvolvimento Microsoft .NET. Lançado em 2002 o .NET consiste em uma iniciativa da Microsoft, que visa unificar o desenvolvimento e execução de sistemas em uma única plataforma. Com sua evolução rápida e constante, ao longo dos anos ganhou muitos adeptos, formando uma comunidade com milhares de desenvolvedores em todo mundo, fato que coloca o C#, principal linguagem do .NET, entre as linguagens mais utilizadas.

Essa escolha foi motivada principalmente pela necessidade de migração para uma plataforma robusta e moderna, pela complexidade do projeto e também pela experiência do estagiário. A segurança, compatibilidade, e um ambiente único de desenvolvimento foram características requeridas pelo projeto e que estão presentes no .NET. Além disso a necessidade de coletar informações dos sistemas operacionais e dos hardwares onde as ferramentas serão instaladas, reforçou a escolha, já que o framework oferece uma extensa API que abstrai a integração com o sistema operacional, permitindo extrair todas essas informações com facilidade.

Outra definição importante foi a versão do .NET Framework, uma vez que sua instalação de se torna obrigatória nas máquinas que irão executar as ferramentas. Neste ponto optou-se pela versão 4.5.2 do framework.

3.3 Visual Studio

Microsoft Visual Studio é um conjunto completo de ferramentas para construir aplicativos desktop e aplicativos Web empresariais desenvolvidos por equipes. Além de criar aplicativos individuais de alto desempenho, é possível utilizar ferramentas de desenvolvimento baseados em componentes do Visual Studio e outras tecnologias para simplificar o trabalho em equipe no projeto, desenvolvimento e implantação de soluções empresariais.

O Microsoft Visual Studio é um produto da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# e J# (J Sharp).

A ferramenta Visual Studio é protegida por uma licença proprietária da Microsoft, porém existe uma versão Community voltada para estudantes, que pode ser utilizada sem qualquer custo.

3.4 WPF e XAML

XAML (Extensible Application Markup Language) é uma linguagem de marcação desenvolvida pela Microsoft e baseada em XML (eXtensible Markup Language). Esta tecnologia é utilizada para o desenvolvimento de interfaces gráficas para aplicativos, sua funcionalidade é semelhante a linguagem HTML (HyperText Markup Language) que é utilizada para apresentação visual de páginas Web. Para desenvolvimento de interfaces gráficas, com XAML, pode-se utilizar ferramentas de desenho visual, como o Visual Studio.

Apesar de possuir características semelhantes a HTML, XAML não é sucessora desta linguagem, mas sim parte do projeto WPF, uma categoria de recursos do Framework .NET 3.5 da Microsoft que lida com a apresentação visual de aplicativos baseado em páginas Web e no Windows. Aplicações baseadas em WPF podem ser executadas em qualquer versão do Windows, desde que tenha instalado o Framework .NET a partir da versão 3.5.

Em resumo, o WPF usa XAML para construir interfaces de usuário, possibilitando a inserção de vários elementos, como controles, textos, imagens, formas, animações dentre outros componentes, e tudo isso sem precisar utilizar qualquer linguagem de programação. No entanto, como XAML é uma linguagem declarativa, caso o usuário precise adicionar regras lógicas a seu aplicativo é necessário a adição de código extra que deve ser escrito em um arquivo à parte. Esta separação em camadas permite que desenvolvedores e designers possam trabalhar em conjunto, sem atrasar o progresso do outro. Outra peculiaridade interessante dos arquivos XAML é que eles sempre são associados a um arquivo de código C# que fornece suporte para execução de operações lógicas, característica que as linguagens declarativas são incapazes de realizarem.

3.5 MahApps.Metro

O estilo Metro ficou conhecido com o lançamento do Windows 8 e do Windows Phone 7, sistemas operacionais cujas interfaces visuais contavam com várias diferenças se comparadas aos seus antecessores.

O que inicialmente causou estranhamento em muitos usuários, devido ao visual “quadrado e sem brilho”, logo se tornou tendência de design e com algumas variações avançou entre as plataformas web, desktop e mobile.

Em aplicações WPF, o estilo pode ser alterado com certa facilidade graças à flexibilidade e recursos oferecidos pelo framework. Esse visual pode ser obtido com a customização das formas e cores dos controles. Porém, não precisamos realizar toda essa formatação visual a partir do zero, graças à biblioteca MahApps.Metro, que já traz uma série de recursos para dar às aplicações uma interface elegante e profissional, semelhante aos aplicativos modernos do Windows.

3.6 Padrão MVVM

O padrão de projeto MVVM (Model-View-ViewModel) é muito semelhante em alguns aspectos aos padrões MVC (Model-View-Controller) e MVP (Model-View-Presenter). Foi criado por engenheiros da Microsoft e inicialmente utilizado em aplicações Silverlight e WPF (Windows Presentation Foundation). Este padrão tem como objetivo facilitar o desacoplamento das responsabilidades da interface do usuário e a lógica de negócios durante o processo de desenvolvimento de software.

Além disso, cada camada tem sua responsabilidade bem definida e essa separação fornece grandes benefícios, sendo um deles a facilidade de testar. Outra grande vantagem associada ao padrão MVVM é a reutilização de código, uma vez que a mesma ViewModel pode ser utilizada por mais de uma interface.

Uma ViewModel, diferentemente do que se possa imaginar é uma classe não visual que existe para dar vida a uma View, isto é, ela simplesmente fornece os dados e executa as ações que a View necessita. Já a View é a UI e não executa nenhuma operação no sistema, apenas apresenta informações ao usuário e responde à essa interação de forma visual. A comunicação entre essas classes dar-se por meio dos data bindings que são responsáveis por “notificarem” a View sobre alguma alteração nos dados ou os atualizar quando são modificados na View.

3.7 Entity Framework

O Entity Framework é um conjunto de tecnologias no ADO.NET que dão suporte ao desenvolvimento de aplicativos de software orientados a dados. Os arquitetos e desenvolvedores de aplicativos orientados a dados lutam com a necessidade de realizar dois objetivos muito diferentes. Precisam modelar as entidades, as relações e a lógica dos problemas de negócios que estão solucionando e também precisam trabalhar com os mecanismos de dados

usados para armazenar e recuperar os dados. Os dados podem se estender por vários sistemas de armazenamento, cada um com seus próprios protocolos. Mesmo aplicativos que trabalhem com um único sistema de armazenamento devem balancear os requisitos do sistema de armazenamento com os requisitos de gravação eficiente e código de aplicativo que possa ser mantido.

O Entity Framework permite que os desenvolvedores trabalhar com dados na forma de objetos específicos de domínio e propriedades, como clientes e endereços de clientes, sem precisar se preocupar com as tabelas de banco de dados e colunas onde esses dados são armazenados subjacentes. Com o Entity Framework, os desenvolvedores podem trabalhar em um nível mais alto de abstração quando eles lidam com dados e podem criar e manter aplicativos orientados a dados com menos código do que em aplicativos tradicionais.

3.8 RestSharp

O modelo conhecido como REST (sigla de “Representational State Transfer”) tem sido cada vez mais utilizado na implementação de serviços em diferentes plataformas, correspondendo a uma das bases empregadas na concepção de tecnologias como o ASP.NET Web API e o Node.js. Levando em conta tal cenário, é inegável que em algum momento desenvolvedores .NET precisem consumir dados disponibilizados por aplicações deste tipo.

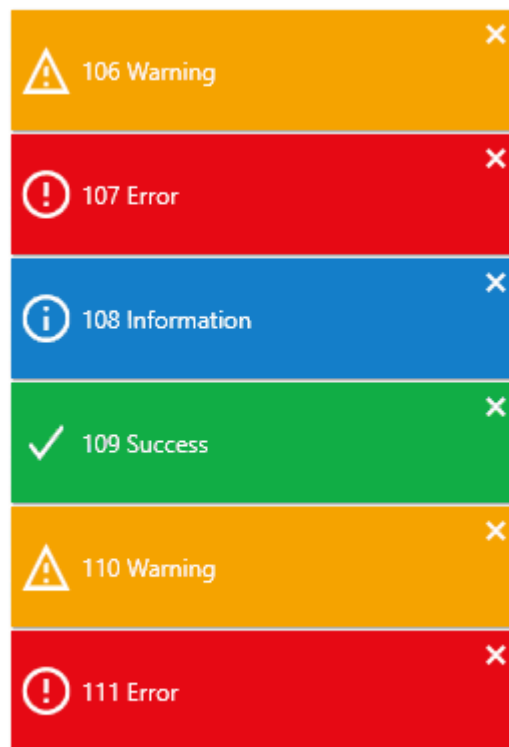
Uma solução simples e que pode atender de forma satisfatória a este tipo de demanda é a biblioteca RestSharp, um projeto open source que disponibiliza diversos recursos com vistas a facilitar a interação com serviços RESTful.

Esta biblioteca será bastante útil para realizar a comunicação entre o sistema desktop e o servidor.

3.9 Toast Notifications

A biblioteca Toast Notifications nos permite criar e exibir notificações ricas em aplicativos WPF. É altamente configurável com um conjunto de opções incorporadas, como posições, comportamentos, temas e muitos outros. É extensível, oferece possibilidade de criar notificações personalizadas e interativas de forma simples, conforme pode ser visto na Figura 4.

Figura 4 – Toast Notifications



Fonte: Arquivo pessoal

3.10 MySQL

MySQL é um sistema de gerenciamento de banco de dados SGDB que utiliza uma linguagem SQL (Structure Query Language) de código aberto, é multitarefa que suporta acessos diferentes, diversos programas clientes e bibliotecas, ferramentas administrativas e diversas interfaces de programação (API's), é desenvolvido, distribuído e tem suporte da MySQL AB. O MySQL é conhecido por sua facilidade de uso, sua interface simples, e também sua capacidade de rodar em vários sistemas operacionais [16].

O MySQL em relação a outros bancos de dados do mesmo porte tem uma maior facilidade para a programação, tem funções mais simples e pode ser totalmente modificado.

A conectividade, velocidade, e segurança fazem com que o MySQL seja altamente adaptável para acessar bancos de dados na Internet.

3.11 MySQL Workbench

MySQL Workbench é uma ferramenta de gerenciamento de banco de dados, possui uma interface gráfica para modelagem de dados, ambiente para desenvolvimento da linguagem SQL e ferramentas de administração para configuração de servidores, administração de usuários, backup e etc.

Esta ferramenta permite que você crie, modele, gere e gerencie banco de dados, podendo criar modelos ER (Entidade Relacionamento) complexos. O MySQL Workbench permite que o desenvolvedor faça engenharia reversa em um banco de dados existente ou em um aplicativo empacotado para obter uma visão melhor do seu design de banco de dados. Não só o MySQL Workbench pode avançar e reverter os bancos de dados existentes, mas também pode importar scripts SQL para construir modelos e exportar modelos que podem ser executados posteriormente [16].

O MySQL Workbench oferece ferramentas visuais para criar, executar e otimizar consultas SQL. O SQL Editor fornece realce de sintaxe de cor, auto-completar, reutilização de trechos de SQL e histórico de execução do SQL.

O MySQL Workbench fornece um conjunto de ferramentas para melhorar o desempenho dos aplicativos MySQL. Os desenvolvedores podem visualizar rapidamente os principais indicadores de desempenho usando o Performance Dashboard. Os relatórios de desempenho fornecem fácil identificação e acesso a pontos de acesso, declarações SQL de alto custo e muito mais [16].

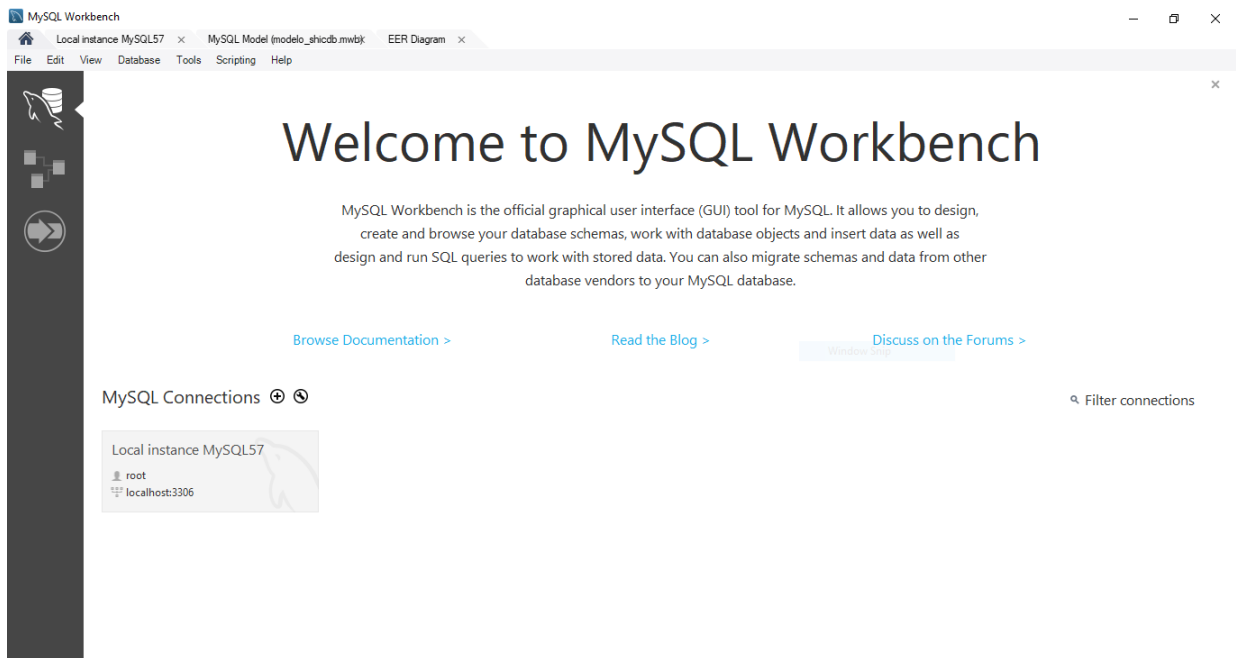
Considerando essas características o MySQL Workbench foi escolhido para o desenvolvimento desse projeto, para criação e manipulação do banco de dados e seus registros. Nas Figuras 5, 6 e 7, temos algumas telas do MySQL Workbench.

3.12 PHP

O PHP (Hypertext Preprocessor), foi concebido em 1994 como resultado do trabalho desenvolvido por Rasmus Lerdorf. Em 2002 já era utilizado por mais de nove milhões de domínios em todo mundo, em 2012 esse número estava próximo dos um bilhão de domínios. Inicialmente o PHP foi denominado Personal Home Page, mas posteriormente foi alterado para Hypertext Preprocessor, nome esse escolhido por meio de votação realizada na comunidade do PHP [17].

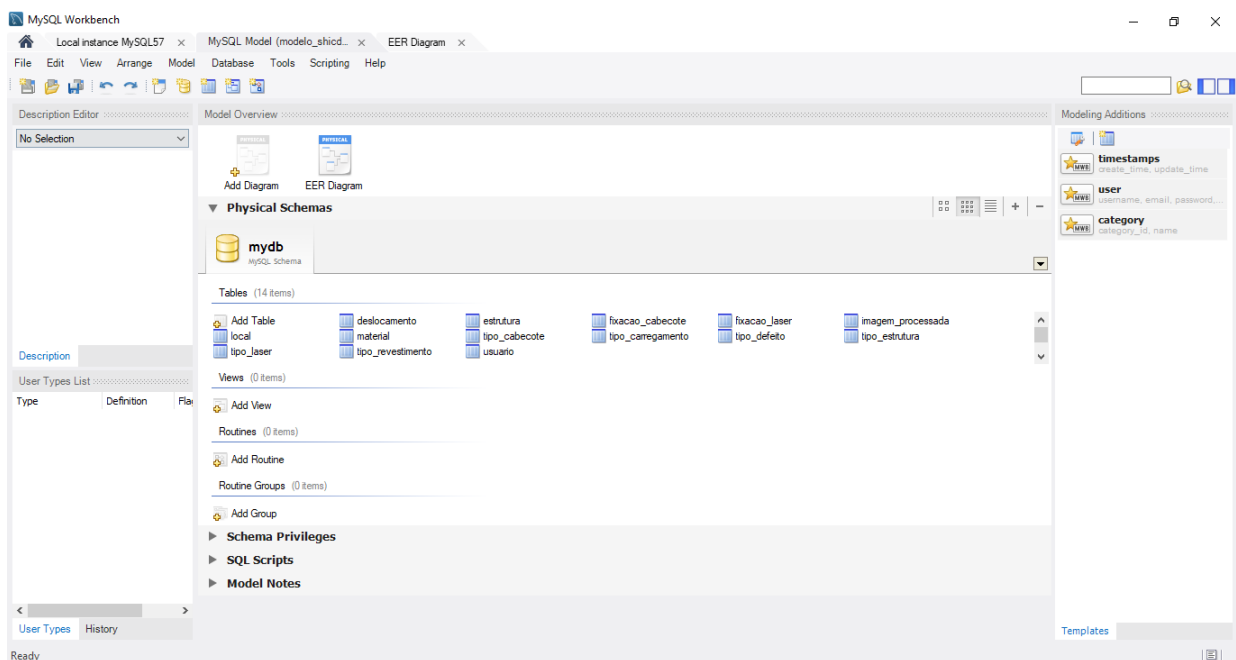
O PHP é uma linguagem de script embutido em HTML (Hypertext Markup Language) no servidor, projetada especificamente para o desenvolvimento Web. É possível embutir/inserir PHP dentro nas páginas HTML, pois o código PHP será executado/interpretado pelo servidor. O PHP é considerado um módulo oficial do servidor HTTP Apache, isso signi-

Figura 5 – MySQL Workbench - Tela inicial



Fonte: Arquivo pessoal

Figura 6 – MySQL Workbench - Modelos



Fonte: Arquivo pessoal

PHP, Java, HTML, R, CSS, SQL, Markdown, TypeScript, LESS, SASS, JSON, XML e Python, assim como muitos outros formatos de arquivos comuns.

Ele é gratuito e open source, com seu código disponibilizado no GitHub, e isso permite que você contribua com seu desenvolvimento.

3.14 JSON

JSON (JavaScript Object Notation) é um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações Web devido a sua capacidade de estruturar informações de uma forma bem mais compacta do que a conseguida pelo modelo XML, tornando mais rápido o parsing dessas informações. Isto explica o fato de o JSON ter sido adotado por empresas como Google e Yahoo, cujas aplicações precisam transmitir grandes volumes de dados.

3.15 Postman

O Postman é um software usado para emular requisições web. Assim, sem precisar que a requisição venha do sistema, é possível testar a qualidade do código produzido, e se está atendendo a necessidade em questão. As requisições do Postman são feitas usando a linguagem de marcação JSON, por onde são passadas as informações para o backend.

3.16 Filezilla

O FileZilla é um aplicativo de código aberto, para Windows, Mac e Linux, recomendado para quem está começando a mexer com internet agora e precisa enviar arquivos para algum servidor através do protocolo FTP (File Transfer Protocol). Com interface completamente amigável, torna-se muito mais fácil realizar o upload de arquivos para o servidor.

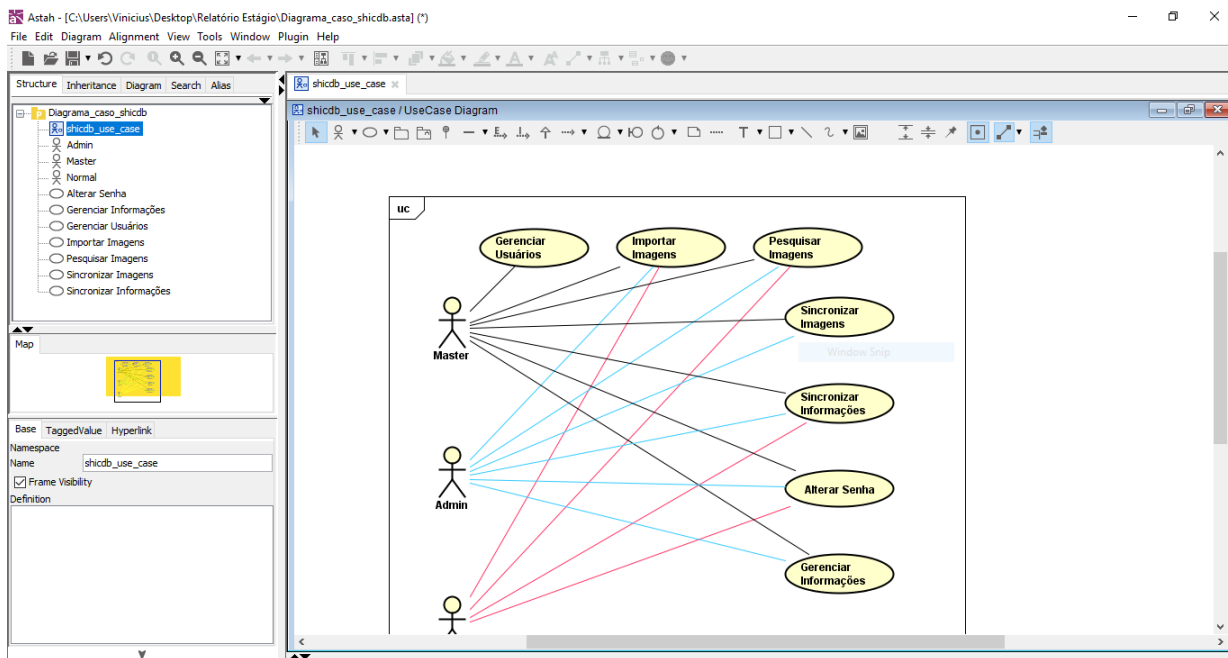
3.17 GIT

GIT é um sistema para controle de versões de arquivos, muito usado para versionamento de código, com seu uso, diversas pessoas podem trabalhar simultaneamente no mesmo projeto, alterando e ou criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas [18]. Para versionar o código deste projeto foi utilizado o GIT juntamente ao Gitlab para que o mesmo pudesse ser acessado facilmente de qualquer máquina.

3.18 Astah

Astah é uma ferramenta de modelagem UML (Unified Modeling Language). Esta ferramenta foi desenvolvida na plataforma Java, o que garante sua portabilidade para qualquer plataforma que possui JVM (Java Virtual Machine). Na Figura 8, vemos um exemplo do uso da ferramenta Astah.

Figura 8 – Ferramenta Astah



Fonte: Arquivo pessoal

Com uma interface simples, ela possibilita qualquer pessoa a projetar um sistema complexo, a ferramenta é utilizada nos diagramas dinâmicos, voltada para modelagem de sistema.

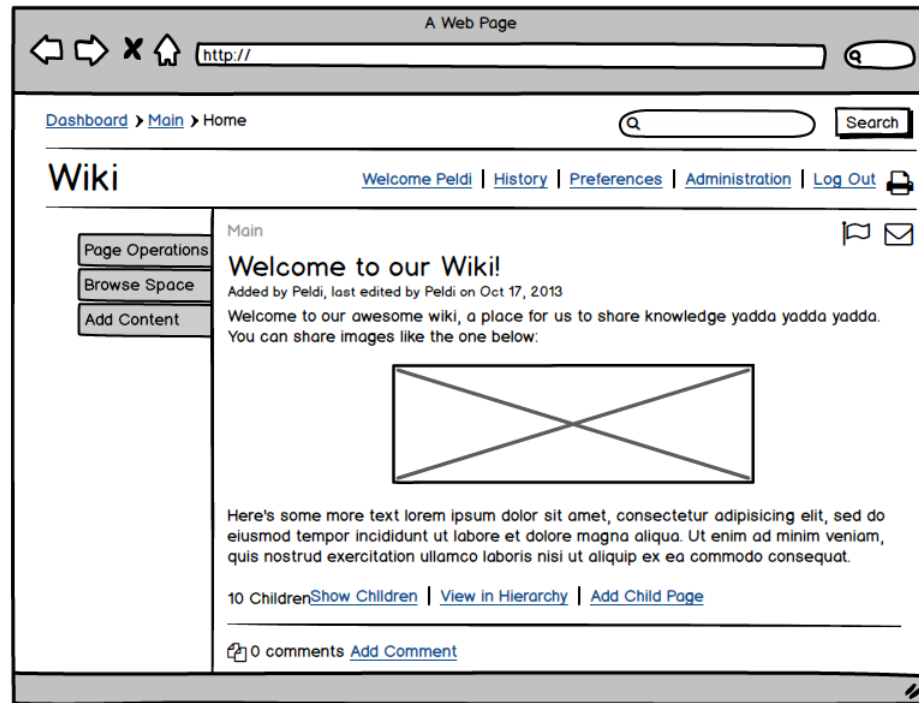
3.19 Balsamiq Mockups

Balsamiq Mockups é uma ferramenta de design de interface de usuário para criação de wireframes. Pode ser usado para gerar esboços das ideias do produto, para facilitar a discussão e compreensão antes de escrever qualquer código [19].

Com o Balsamiq é possível gerar protótipos de baixa fidelidade no qual pode ser visto um exemplo na Figura 9, de maneira rápida. Esses protótipos contêm os principais elementos da interface do sistema, facilitando assim a compreensão de como o sistema deve

se comportar. Os protótipos construídos no Balsamiq são protótipos descartáveis, ou seja, não serão utilizados em outras fases do processo de desenvolvimento, exceto para consulta.

Figura 9 – Balsamiq Mockups - Exemplo de protótipo



Fonte: Balsamiq [19]

A ferramenta tem como forma de trabalho a técnica drag-and-drop, que consiste em arrastar um componente e soltá-lo onde se deseja. O Balsamiq oferece os principais componentes utilizados em interfaces, como botões, painéis, menus, tabelas, diferentes tipos de telas e outros.

4 Projeto e Implementação

O estágio supervisionado tinha como intuito desenvolver um sistema para o controle de imagens de Shearografia. Neste capítulo será abordado mais detalhadamente o processo de levantamento dos requisitos e de desenvolvimento do sistema.

4.1 Engenharia de requisitos

Vimos anteriormente quando falamos de engenharia de requisitos, que nessa etapa buscamos técnicas para colhermos requisitos do sistema a ser desenvolvido. Assim definimos um conjunto de medidas a serem extraídas informações para auxiliar o desenvolvimento do sistema a ser criado, para atingir os objetivos como: qualidade de software, produtividade no desenvolvimento, permitindo que o desenvolvedor atenda os prazos estabelecidos, qualidades e as funcionalidades do sistema que o cliente almeja.

4.1.1 Entrevista

Durante o levantamento de requisitos do sistema foi realizada uma entrevista semiestruturada com o objetivo de obter informações necessárias para o desenvolvimento do software. A entrevista realizada teve como objetivo entender como funciona o negócio do cliente e entender qual a necessidade e o objetivo do sistema.

Durante a entrevista foram realizadas perguntas abertas para entender a perspectiva que o sistema ocasionaria para o cliente, em uma outra etapa durante a entrevista foi feito um questionário para documentar o que o cliente esperava do sistema, e qual seria as funcionalidades que o sistema deveria ter.

A coleta de requisitos junto ao cliente iniciou-se por meio de um brainstorm onde o cliente apresentou aspectos iniciais das necessidades e ainda o brainstorm guiou a elaboração dos tópicos da entrevista semiestruturada e do questionário aplicado.

Após o brainstorm, foi elaborado um questionário com perguntas abertas que serviu de roteiro para a realização de uma entrevista semiestruturada para coleta de requisitos. A entrevista teve por objetivo elucidar alguns pontos relacionados ao sistema, a citar:

- Saber a funcionalidade do sistema
- Quais serão os tipos de usuários do sistema
- Qual será o aspecto visual do sistema

- Informações que o sistema deve conter
- Restrições do sistema

Para esclarecer esses pontos, elaborou-se questões como:

- Quais as funcionalidades que o sistema precisa ter?
- Quem poderá ter acesso ao sistema? Quais são as restrições para cada tipo de usuário?
- Quais funcionalidades tem prioridade?
- O sistema deve ser compatível com qual ou quais sistemas operacionais?

4.1.2 Documentação de requisitos de software

Esta etapa envolve as atividades de determinar os objetivos do software e as restrições associadas ao mesmo. Feito esta análise do sistema e delimitado o escopo do software, os requisitos do software devem ser analisados e especificados.

4.1.2.1 Objetivo do documento

A documentação dos requisitos de software tem com o objetivo descrever e especificar os requisitos funcionais e não funcionais do sistema denominado *ShicDB*, para atender todas as necessidades do cliente, especificando de uma maneira clara e objetiva para que o produto seja feito pelo desenvolvedor conforme o cliente deseja.

4.1.2.2 Escopo do sistema

O sistema tem como objetivos principais realizar operações de: cadastramento, sincronização e filtragem das imagens obtidas através da Shearografia. O sistema poderá ser acessado por qualquer pessoa que possua cadastro, tendo algumas funções limitadas de acordo com o tipo de usuário logado.

4.1.2.3 Descrição geral do sistema

Descrição de solução: Criar um sistema eficaz que busca a satisfação de nosso cliente, para que ele tenha um gerenciamento das imagens processadas obtidas com a técnica de Shearografia.

Benefícios: Centralização de informações, consulta remota às imagens de Shearografia, acesso às informações sobre inspeções passadas, possibilidade de reprocessar imagens antigas, disponibilizar informações de maneira fácil, segura e organizada.

4.1.2.4 Perspectiva do sistema

Todas as telas do sistema serão de simples acesso e de clareza para que seja fácil utilizá-lo. O sistema será utilizado em plataforma desktop em máquinas com sistema operacional Windows.

4.1.2.5 Características dos usuários

- Os usuários que possuem privilégio *Master* podem acessar todas as funcionalidades do sistema.
- Os usuários que possuem privilégio *Admin* podem acessar todas as funcionalidades disponíveis para usuários *Master*, exceto a opção de *Gerenciar usuários*.
- Os demais usuários possuem privilégio *Normal* e não podem acessar as seguintes funcionalidades:
 - *Upload* de novas imagens para o servidor
 - *Gerenciar usuários*
 - *Gerenciar informações*, ou seja, criar, editar ou deletar certas informações

4.1.2.6 Requisitos

Nessa etapa de elaboração do documento de requisitos contém a especificação de requisitos para o sistema *ShicDB*.

4.1.2.6.1 Requisitos funcionais (RF)

- RF01 - O sistema terá 3 tipos de privilégio para os usuários: *Master*, *Admin* e *Normal*.
- RF02 - O usuário *Master* terá acesso total do sistema.
- RF03 - Usuários com privilégios *Admin* e *Normal* terão acessos restritos à certas funcionalidades do sistema.
- RF04 - Usuários com privilégio *Master* podem importar imagens processadas, filtrar imagens cadastradas, realizar upload de novas imagens para o servidor, realizar sincronização de informações do servidor para o sistema desktop, gerenciar usuários e gerenciar informações gerais.

- RF05 - Usuários com privilégio *Admin* podem importar imagens processadas, filtrar imagens cadastradas, realizar upload de novas imagens para o servidor, realizar sincronização de informações do servidor para o sistema desktop e gerenciar informações gerais.
- RF06 - Usuários com privilégio *Normal* podem importar imagens processadas, filtrar imagens cadastradas, realizar sincronização de informações do servidor para o sistema desktop.
- RF07 - O sistema deverá ter opção de mudar senha.
- RF08 - O sistema deverá validar as informações antes do usuário importar nova imagem.
- RF09 - O sistema deverá filtrar as imagens de acordo com as opções selecionadas em tempo real.
- RF10 - O sistema deverá ter opção de edição e exclusão de imagens já cadastradas.
- RF11 - O sistema deverá ter opção de realizar o upload das imagens cadastradas no sistema desktop para o servidor, tornando-as disponíveis para os demais usuários.
- RF12 - O sistema deverá ter opção de gerenciamento de usuários, permitindo alteração de privilégios ou exclusão.
- RF13 - O sistema deverá ter opção de gerenciamento de informações referentes às imagens processadas.

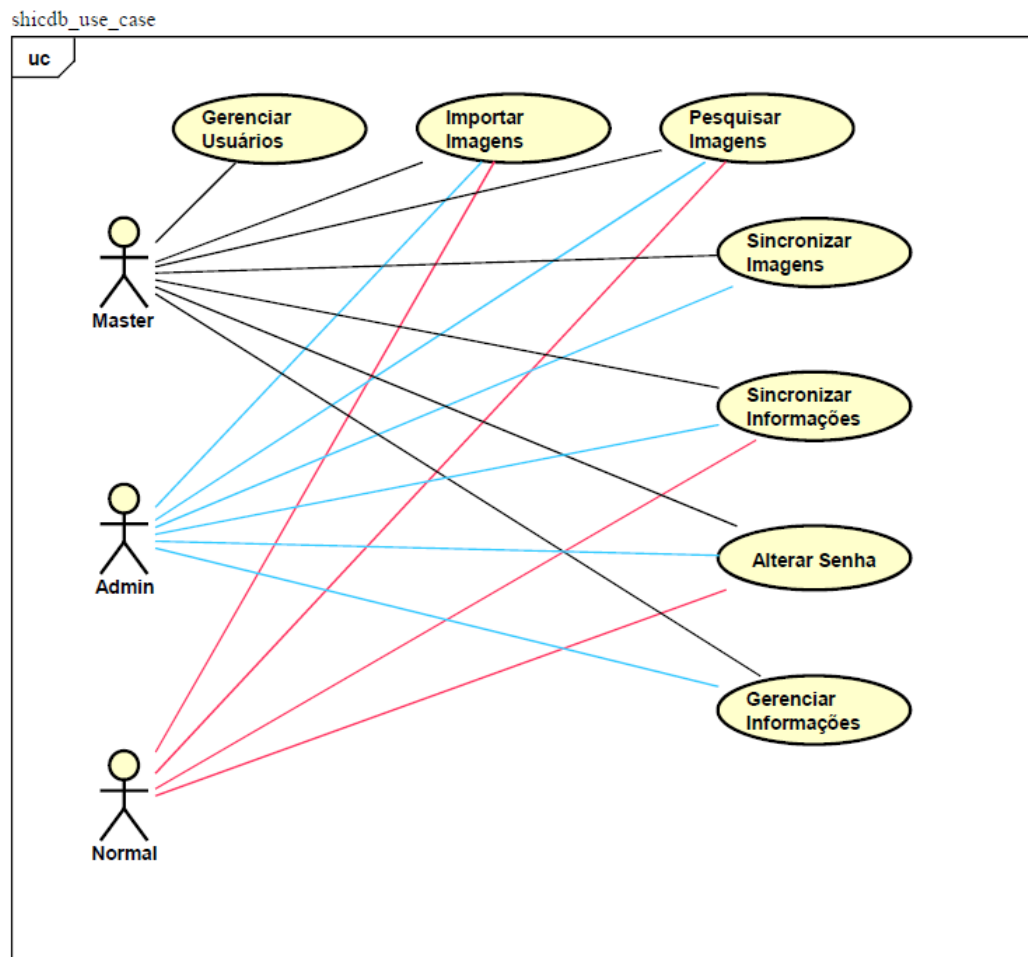
4.1.2.6.2 Requisitos não funcionais (NF)

- NF01 - Será necessário a utilização do sistema operacional Windows 7 ou superior para executar o sistema.
- NF02 - Interface do sistema deverá ser de fácil entendimento e intuitiva para o usuário.
- NF03 - Utilização de linguagem C# para o desenvolvimento do sistema desktop.
- NF04 - Todas as operações que envolvem comunicação com o servidor deverão ter mensagens indicando o resultado.
- NF05 - Caso o usuário esqueça a senha, um e-mail com informações para mudança de senha deve ser enviado.

4.1.3 Diagramas de caso de uso

Na elaboração dos diagramas de casos de usos será exibido uma visão geral da funcionalidade do sistema e como os usuários irão utilizar o sistema. No diagrama foi descrito as principais funcionalidades do sistema e como que o usuário irá se relacionar com o sistema. A Figura 10, demonstra o caso de uso do sistema utilizando o Software Astah.

Figura 10 – Diagrama de caso de uso - Sistema ShicDB



Fonte: Arquivo pessoal

O diagrama descreve as funções do sistema, representando um usuário com privilégio *Master*, *Admin* ou *Normal* acessando informações do sistema.

Descreve-se nesse sistema que os usuários têm a possibilidade de acessar o sistema por um login e senha, afim de obter o acesso ao sistema.

Como pode ser visto, todos os tipos de usuário possuem acesso às seguintes funcionalidades:

- Importar imagens
- Pesquisar imagens
- Sincronizar informações
- Alterar senha

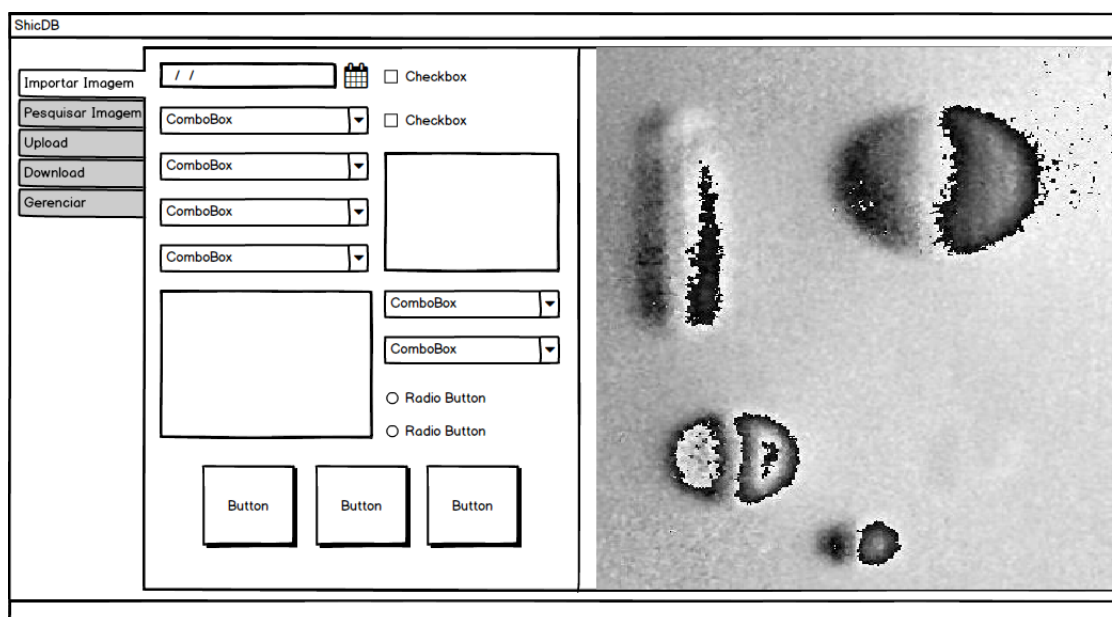
Sendo outras funções disponíveis de acordo com o privilégio dado a cada usuário.

4.2 Protótipo

A etapa de prototipação foi realizada com a intenção de fornecer um esboço das telas do sistema, para ter uma ideia do que seria o sistema. Foram criados protótipos simples de baixo nível técnico, que serviu apenas para ter a interpretação e compreensão de como que seriam as telas do sistema e sua funcionalidade.

Somente algumas telas foram construídas apresentando as funcionalidades e exemplificando o caminho que teria que seguir para ter acesso a determinada função no sistema, tudo feito o mais simples possível para apenas ter uma ideia de que seria o sistema. A Figura 11 demonstra como seria o protótipo da tela *Importar Imagem*.

Figura 11 – Protótipo ShicDB - Tela Importar Imagem



Fonte: Arquivo pessoal

4.3 Modelagem do banco de dados

Nesta seção será apresentado como o banco de dados para o projeto foi desenvolvido e estruturado. Foram utilizados o modelo entidade-relacionamento e o modelo lógico.

4.3.1 Modelo entidade relacionamento (ER)

Todo o processo de criação do banco de dados, aconteceu fazendo uso da ferramenta MySQL WorkBench. Primeiramente, foi desenvolvido o DER (Diagrama de entidade-relacionamento), que nada mais é que uma representação visual de todas as entidades/tabelas do banco de dados e suas cardinalidades.

O modelo entidade-relacionamento é considerado a base da área moderna da tecnologia de banco de dados, ou seja, um projeto que usa uma base de dados precisa ter o modelo entidade-relacionamento bem definido [13].

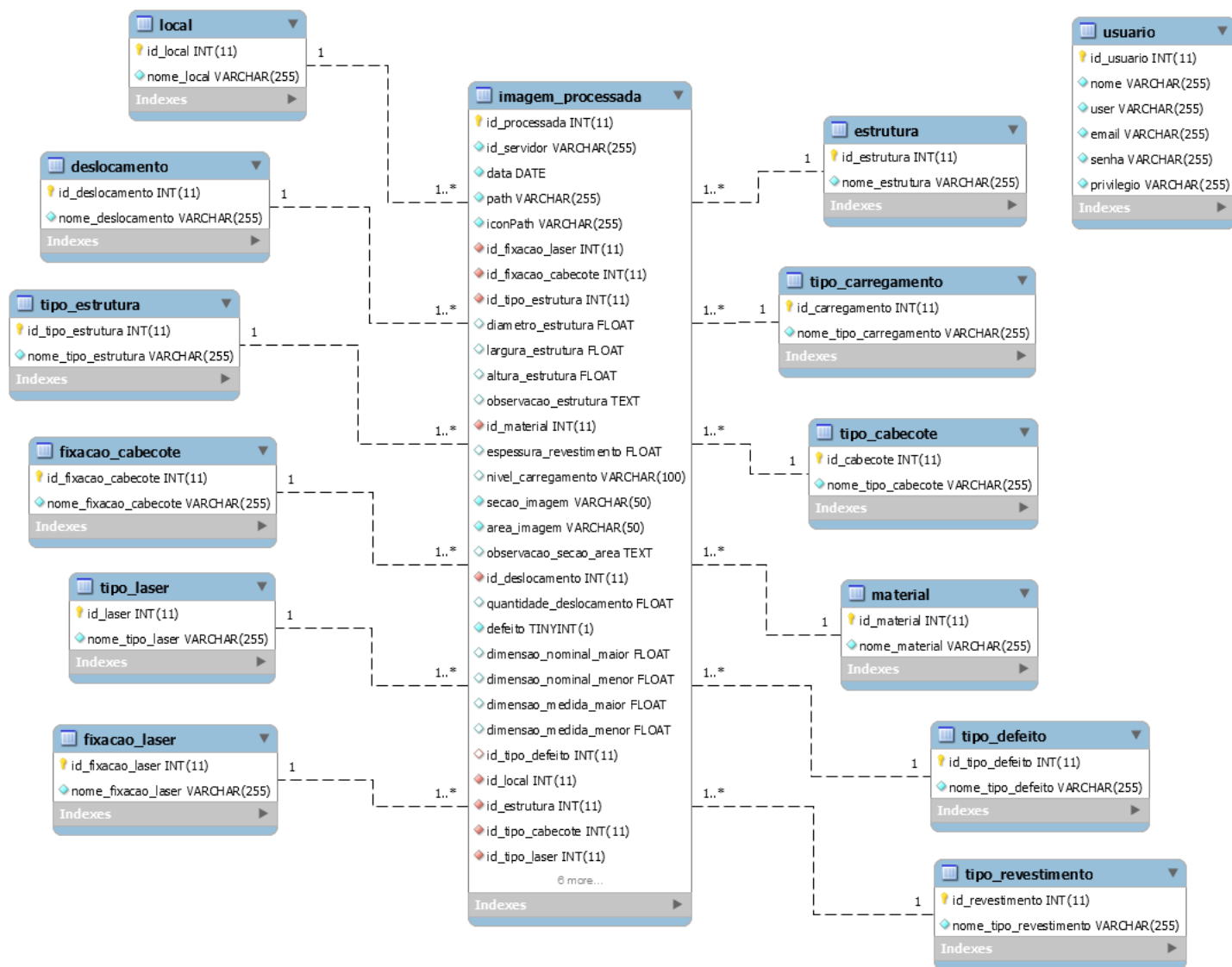
A Figura 12 apresenta como o diagrama entidade relacionamento foi desenvolvido para atender o proposto no projeto.

4.3.2 Modelo lógico

Com base no DER, e utilizando as funcionalidade de criação e exportação de script disponível na ferramenta MySQL Workbench, foi desenvolvido o modelo lógico de cada uma das tabelas.

O modelo lógico possibilita um detalhamento maior de cada uma das entidades desenvolvidas, é possível visualizar os tipos de dados, os atributos, chaves primárias e estrangeiras etc . A Figura 13 mostra a tabela usuário e sua estrutura lógica. Os demais modelos não serão mostrados neste relatório pois seguem o mesmo esquema.

Figura 12 – Modelo Entidade Relacionamento - ShicDB



Fonte: Arquivo pessoal

Figura 13 – Modelo Lógico - Tabela Usuário

```
-- -----  
-- Table `mydb`.`usuario`  
-- -----  
CREATE TABLE IF NOT EXISTS `mydb`.`usuario` (  
  `id_usuario` INT(11) NOT NULL,  
  `nome` VARCHAR(255) NOT NULL,  
  `user` VARCHAR(255) NOT NULL,  
  `email` VARCHAR(255) NOT NULL,  
  `senha` VARCHAR(255) NOT NULL,  
  `privilegio` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`id_usuario`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

Fonte: Arquivo pessoal

5 Resultados

O resultado do desenvolvimento deste projeto foi a construção de um sistema capaz de cadastrar imagens de Shearografia assim como suas informações, centralizando estes dados em um servidor, permitindo que cada usuário possa ter acesso às estas informações de qualquer máquina. O nome dado para a ferramenta aqui desenvolvida foi *ShicDB*. As funcionalidades disponíveis no sistema serão descritas nas próximas seções.

5.1 Princípios do sistema

A ideia do sistema é servir como um banco de dados, onde os usuários podem inserir imagens de Shearografia e suas respectivas informações, fornecendo vantagens como: Centralização das informações, consulta remota desses dados, disponibilização das informações de forma organizada, fácil e segura, entre outras.

5.2 Funcionalidades do sistema

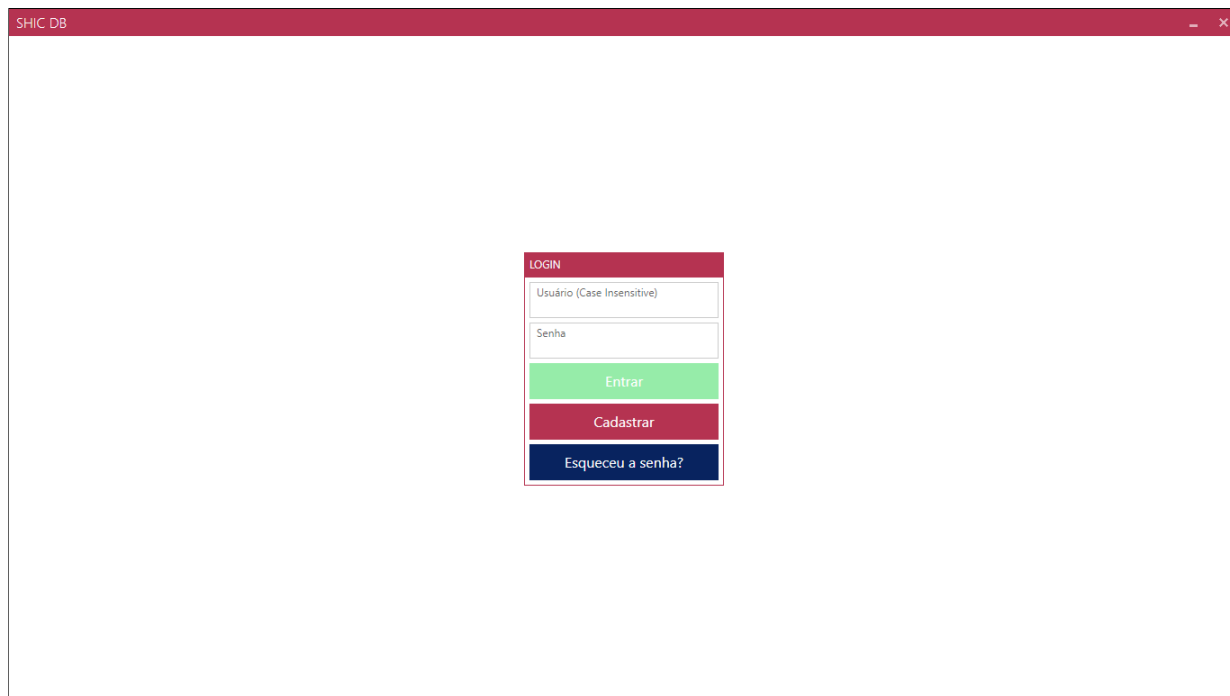
Nesta seção serão descritas as funcionalidades desenvolvidas juntamente com as telas criadas para o sistema.

5.2.1 Login

A Figura 14, ilustra a tela de *Login*, que neste caso é a tela inicial do sistema desenvolvido. Na tentativa do usuário "logar", o sistema verifica se o usuário já possui suas credenciais salvas no banco de dados local, isto é, no banco de dados instalado na máquina onde o sistema está sendo executado. Caso positivo, o sistema autentica o usuário e o leva para a tela *Home* (Figura 15). Caso o sistema não consiga autenticar os dados que o usuário inseriu com os dados contidos no banco de dados local, é feita a tentativa de autenticação com o servidor.

Os dados de login são então enviados e verificados com os dados salvos no servidor, caso exista um usuário cadastrado no servidor com os dados fornecidos, o sistema salva as informações no banco de dados local, assim na próxima vez que este usuário tentar logar nesta mesma máquina, não será necessária a comunicação com o servidor para autenticação. Caso os dados fornecidos não sejam válidos com os dados presentes no servidor, é apresentada uma mensagem de alerta para que o usuário confira suas credenciais.

Figura 14 – Tela de Login - ShicDB



Fonte: Arquivo pessoal

Figura 15 – Tela Home - ShicDB



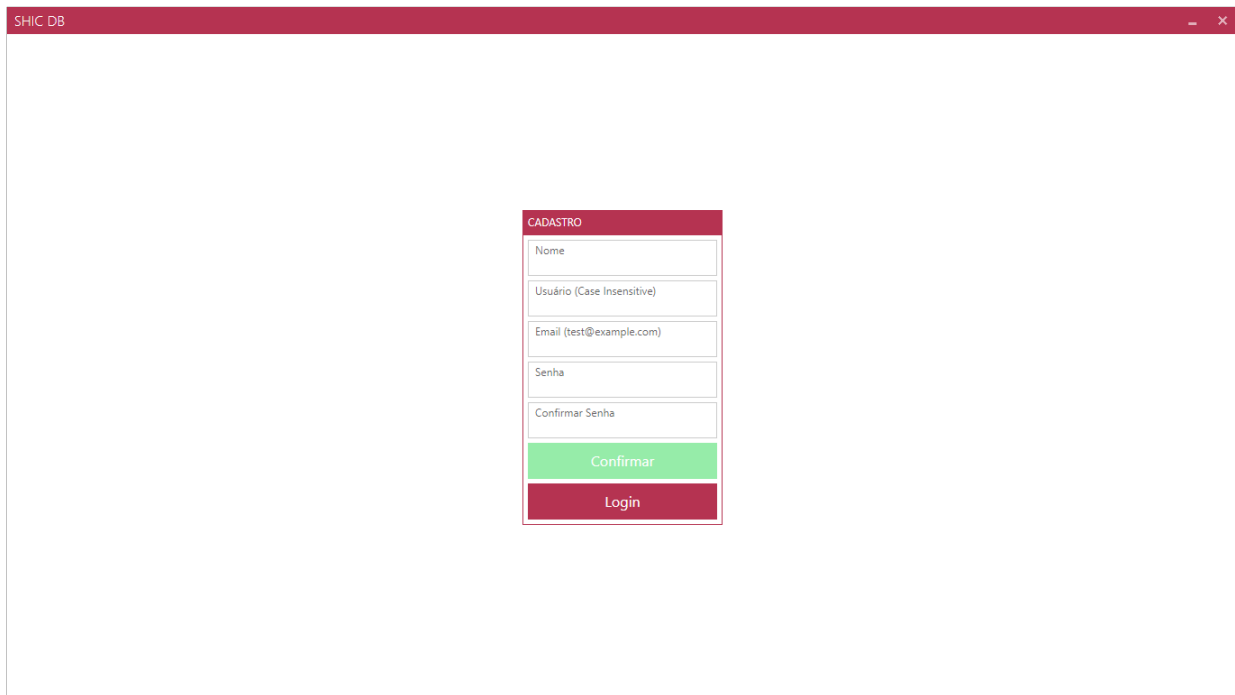
Fonte: Arquivo pessoal

5.2.2 Cadastro

A Figura 16, ilustra a tela de *Cadastro*. Caso um novo usuário deseje se cadastrar para a utilização do sistema, ele deve inserir os seguintes dados: Nome completo, nome de usuário, email e senha. Após a inserção dos dados necessários, o sistema se comunica com o servidor, verificando se já existe alguém com o mesmo email fornecido. Caso negativo, o sistema autentica o usuário, salva suas informações no banco de dados tanto do servidor quanto local e redireciona o usuário para a tela *Home* (Figura 15). Caso positivo, isto é, já exista alguém com o mesmo email cadastrado, um alerta é criado para o utilizador do sistema.

Todos usuários que são cadastrados utilizando a tela de cadastro (Figura 16) são registrados com privilégio *Normal*, caso algum usuário precise de um privilégio superior é necessário notificar um usuário que possua privilégio *Master* para que o mesmo altere esse parâmetro.

Figura 16 – Tela de Cadastro - ShicDB



Fonte: Arquivo pessoal

5.2.3 Home

Após a autenticação, o usuário é redirecionado para a tela *Home* do sistema. Nesta tela é apresentado ao lado esquerdo o menu com as opções disponíveis no sistema, onde algumas são desabilitadas de acordo com o privilégio do usuário. A Figura 17 demonstra

melhor o menu e suas opções. No centro da tela é mostrado o nome do usuário atual, assim como seu privilégio.

Figura 17 – Tela Home com menu aberto - ShicDB



Fonte: Arquivo pessoal

5.2.4 Importar Imagem

A primeira opção do menu lateral conforme pode ser visto na Figura 17 é a de *Importar imagem*. Ao selecionar esta opção, o usuário deve selecionar a imagem a qual deseja cadastrar no sistema e após confirmação da seleção da imagem, ele é redirecionado para a tela de *Importar Imagem* (Figura 18).

Nesta tela (Figura 18) podemos observar que as informações são agrupadas e dispostas em três diferentes seções: *Informações sobre a inspeção*, *Informações sobre a área/seção* e *Informações sobre o defeito*.

As informações que são obrigatórias no cadastro de uma nova imagem são identificadas pela borda vermelha em torno de cada campo. Esta borda também serve como validação, pois ela só desaparece após a inserção de dados válidos.

Alguns campos possuem um botão de adição ao lado que fornece a possibilidade de cadastro de uma nova opção, caso o usuário não ache a opção necessária nas opções já cadastradas. Esses campos são: *Local*, *Estrutura* e *Tipo de revestimento*. As demais

Figura 18 – Tela de Importar Imagem - ShicDB

Fonte: Arquivo pessoal

opções podem ser gerenciadas dependendo do privilégio do usuário selecionando a opção de *Gerenciar informações* conforme é mostrado na Figura 17.

A seção de *Informações sobre o defeito* só é habilitada caso exista um defeito na imagem selecionada, ou seja, somente se o campo *Defeito* é selecionado como verdadeiro.

Após a inserção dos parâmetros obrigatórios e validação dos mesmos, a opção *Salvar* é habilitada. Ao salvar as informações com sucesso, um ícone referente à imagem cadastrada é gerado, onde será utilizado na tela de *Pesquisar* (5.2.5).

As informações da seção *Informações sobre a inspeção* são mantidas, pois normalmente existem diversas imagens em uma mesma inspeção, portanto, essas informações não mudam para estas imagens. Desta maneira, caso o usuário deseje continuar cadastrando imagens de uma mesma inspeção, ele só precisa inserir as informações referentes às outras duas seções (*Informações sobre área/seção* e *Informações sobre o defeito*) e importar uma nova imagem através do botão *Adicionar*. A tela após o primeiro cadastramento pode ser vista na Figura 19.

Todos os cadastros feitos pelo usuário são salvos no banco de dados local, isto é, na máquina onde o sistema está instalado. Sendo assim, estes cadastros ficam pendentes para a realização do upload para o servidor, enquanto esta sincronização não é feita, outros usuários não terão acesso à estas imagens.

Figura 19 – Tela de Importar Imagem após primeiro cadastro - ShicDB

Fonte: Arquivo pessoal

5.2.5 Pesquisar

A segunda opção presente no menu lateral é a de *Pesquisar*. Ao acessar esta opção o usuário é redirecionado para a tela de *Pesquisar* (Figura 20).

Ao acessar a tela de *Pesquisar*, temos primeiramente três seções, sendo duas contendo os parâmetros de pesquisa e a outra contendo uma lista representando os resultados da pesquisa. Esta lista mostra todas as imagens condizentes aos parâmetros da pesquisa, onde cada item possui um ícone e algumas informações referentes à imagem. Caso nenhum parâmetro tenha sido selecionado, todas as imagens cadastradas no banco de dados local são mostradas.

Caso o usuário não possua imagens cadastradas ou queira atualizar as informações do seu banco de dados local referentes às imagens, um botão de atualização é disponibilizado logo acima da seção de *Informações sobre a inspeção*. Ao pressionar este botão, todos os dados referentes às imagens que estão no servidor são transferidos e verificados com os dados salvos no banco de dados local (Figura 21). Todos os dados que são novos em relação aos dados contidos no banco de dados local são salvos juntamente aos ícones que representam cada imagem.

Um exemplo de pesquisa pode ser visto na Figura 22. Ao selecionar um item da lista de resultados, todas as informações referentes à imagem são mostradas, agrupadas em

Figura 20 – Tela de Pesquisar - ShicDB

The screenshot displays the 'Pesquisar' (Search) window in the SHIC DB application. The interface is divided into several sections:

- Left Sidebar:** Contains navigation icons for home, search, and other functions.
- Top Bar:** Labeled 'SHIC DB' and 'Pesquisar'.
- Filters (Left Panel):**
 - INFORMAÇÕES SOBRE A INSPEÇÃO:** Includes fields for 'Data Inicio (dd/mm/yyyy)', 'Data Fim (dd/mm/yyyy)', 'Local', 'Estrutura', 'Tipo Cabeçote', 'Tipo Laser', 'Fixação Laser', 'Fixação Cabeçote', 'Tipo de Estrutura', 'Material', and 'Tipo de Revestimento'.
 - INFORMAÇÕES SOBRE A ÁREA/SEÇÃO:** Includes fields for 'Tipo de Carregamento', 'Deslocamento', 'Defeito', 'Observações', and 'Tipo de Defeito'.
- Results (Right Panel):** Titled 'RESULTADOS', it shows a list of search results. Each result includes a small image of a defect and associated metadata: 'Data: 12/05/2014', 'Local: Houston', 'Estrutura: Tubo 516', 'Cabeçote: 4F', 'Laser: Cobalt Verde 1W', 'Material: Metal / Compósito', and 'Carregamento: Pressão Decrescente'.

Fonte: Arquivo pessoal

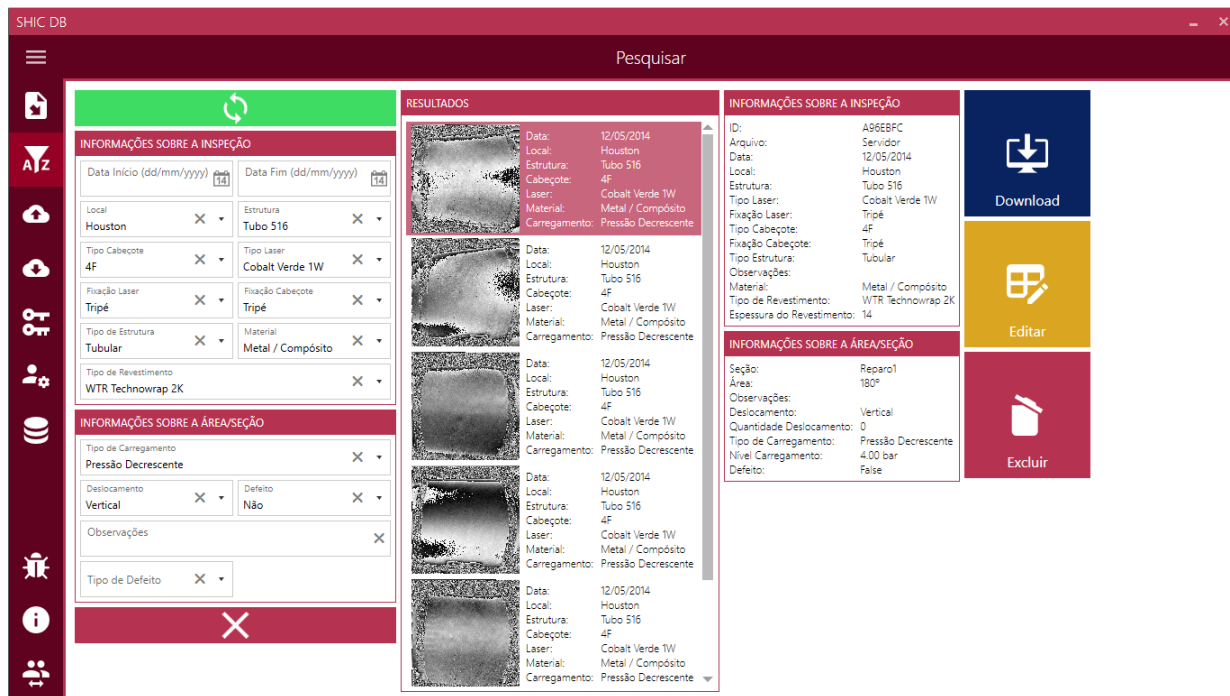
Figura 21 – Sincronização de dados referentes às imagens - ShicDB

This screenshot shows the same 'Pesquisar' window as Figure 20, but with a semi-transparent overlay indicating a synchronization process. The overlay contains the text 'Imagens sincronizadas: 84' (Images synchronized: 84). The background interface, including the filters and results sections, remains visible but is dimmed.

Fonte: Arquivo pessoal

diferentes seções. Caso o item selecionado não possua o arquivo original salvo na máquina, uma opção de *Download* é mostrada ao usuário, caso o item já esteja salvo localmente, uma opção de *Exportar* imagem para um novo local é disponibilizada. Se o usuário possuir privilégio de *Admin* ou *Master*, ele terá mais duas opções referentes à imagem selecionada: *Editar* e *Excluir*.

Figura 22 – Tela de Pesquisar com imagem selecionada - ShicDB



Fonte: Arquivo pessoal

5.2.5.1 Editar

Após realizar a consulta dos registros o sistema fornece a opção de alterar um registro específico. Quando o usuário selecionar o registro que será alterado o sistema irá exibir os campos contendo as informações do registro selecionado, permitindo a alteração das informações ali contidas, como pode ser visto na Figura 23. Como citado anteriormente, esta funcionalidade está disponível apenas para os usuários com privilégio *Admin* ou *Master*.

Figura 23 – Tela de Editar - ShicDB

SHIC DB

Pesquisar

INFORMAÇÕES SOBRE A INSPEÇÃO

Data (dd/mm/yyyy) 12/05/2014

Local Houston

Estrutura Tubo 516

Tipo Laser Cobalt Verde 1W

Fixação Tripé

Tipo Cabecote 4F

Fixação Cabecote

Tipo de Estrutura

Tipo de Revestimento

INFORMAÇÕES SOBRE A ÁREA/SEÇÃO

Seção Reparo1

Área 90°

Observações Open Trial Repair

Deslocamento Vertical

Quantidade 0.0 mm

Tipo de Carregamento Pressão Decrescente

Nível Carregamento 4.00 bar

Defeito Sim

INFORMAÇÕES SOBRE O DEFEITO

Dim. Nominal Menor 0.0 mm

Dim. Nominal Maior 0.0 mm

Dim. Medida Menor 144.0 mm

Dim. Medida Maior

Tipo Quadrado ou Retangular

Salvar

Cancelar

Fonte: Arquivo pessoal

5.2.5.2 Excluir

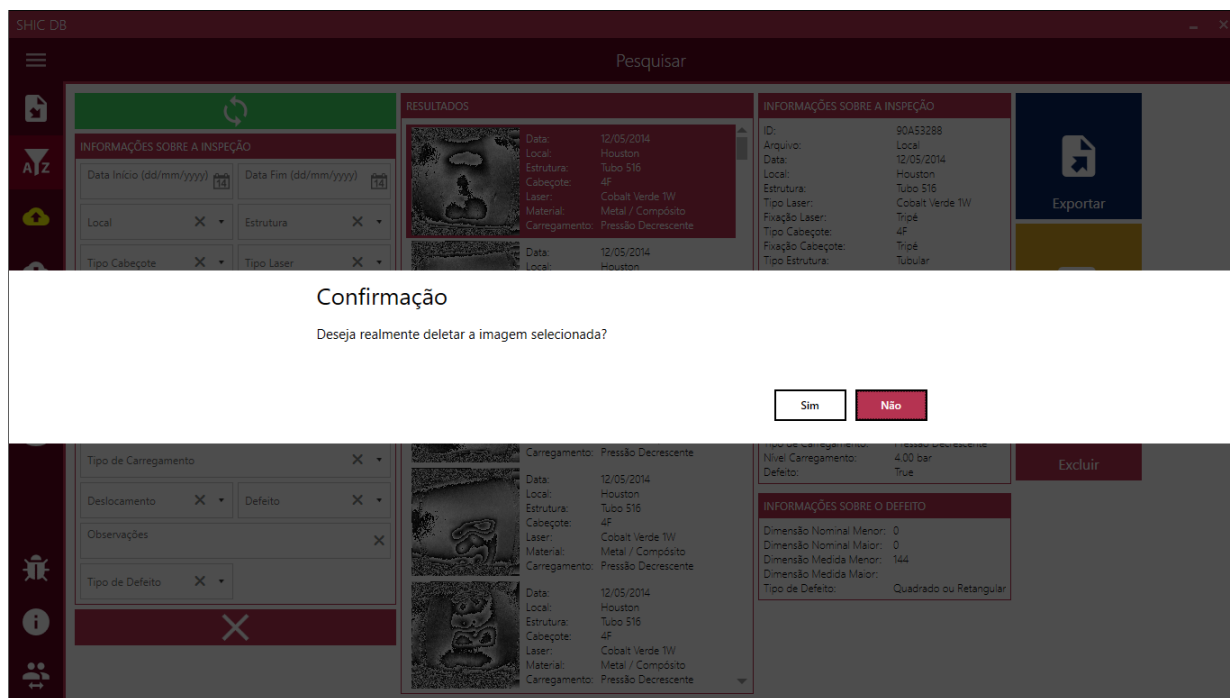
O sistema permite a exclusão dos registros armazenados no banco, como citado anteriormente, esta funcionalidade também está disponível apenas para os usuários com privilégio *Admin* ou *Master*. A exclusão de um registro é feita após consultá-lo e selecionar a opção de *Excluir*. Ao realizar essa ação, o usuário deverá confirmar a exclusão conforme demonstrado na Figura 24.

5.2.6 Upload

A terceira função disponível no menu lateral do sistema é a de *Upload*, isto é, realizar a sincronização dos dados locais referentes às imagens para o servidor. A tela de *Upload* pode ser vista na Figura 25.

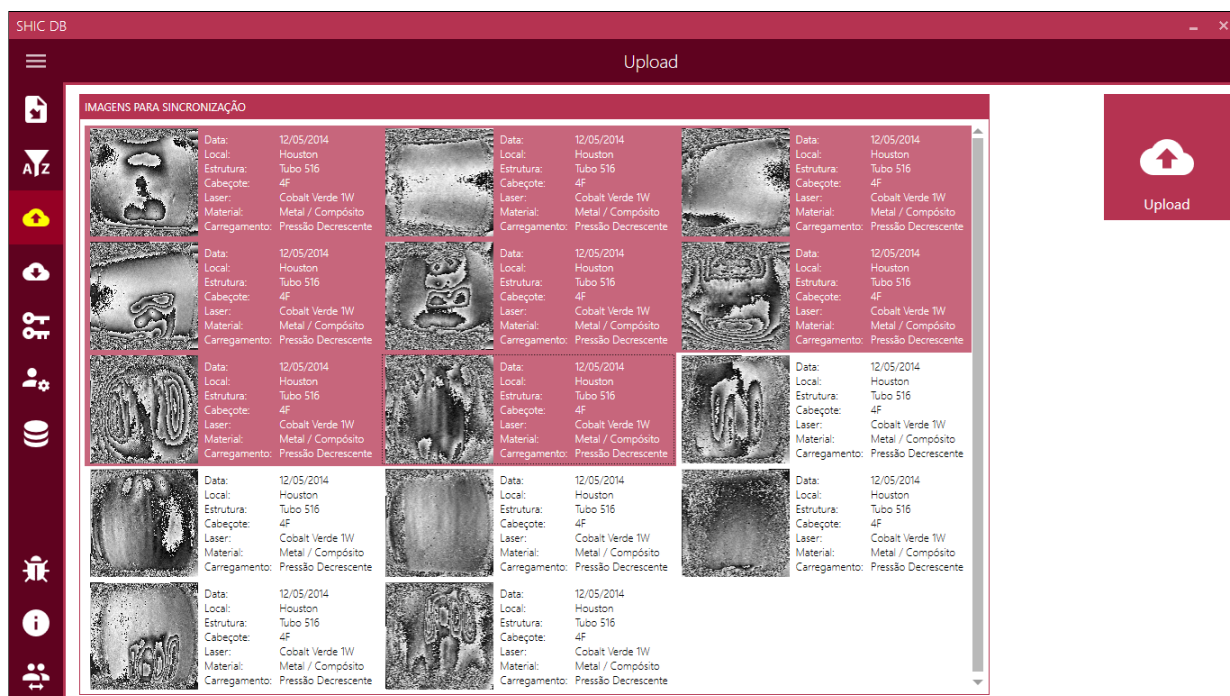
Na tela de *Upload* são listadas todas as imagens que foram cadastradas pelo usuário mas que ainda não foram enviadas para o banco de dados do servidor. Sempre que há alguma imagem que não foi sincronizada, o ícone de *Upload* do menu lateral ficará na cor amarela, indicando que há pendências. Após a sincronização das imagens, elas se tornarão disponíveis para que demais usuários possam ter acesso às mesmas.

Figura 24 – Confirmação de exclusão do registro selecionado - ShicDB



Fonte: Arquivo pessoal

Figura 25 – Tela de Upload - ShicDB



Fonte: Arquivo pessoal

5.2.7 Download

A quarta função disponível é a de *Download*. Esta função não possui uma tela específica e apenas realiza a sincronização das informações que são utilizadas no cadastramento de uma imagem, em outras palavras, atualiza as opções fornecidas para cada parâmetro. Informações estas como:

- Local
- Estrutura
- Tipo de laser
- Material

Vale ressaltar que dados referentes às imagens não são sincronizadas com esta função, sendo somente realizada no botão disponível na tela de *Pesquisar* 5.2.5, conforme mencionado anteriormente.

5.2.8 Gerenciar usuários

A sexta opção no menu lateral é a de *Gerenciar usuários*. Esta opção só é disponível para usuários com privilégio *Master*. Ao acessar esta função, o usuário é redirecionado para a tela de *Gerenciar usuários* conforme mostra a Figura 26.

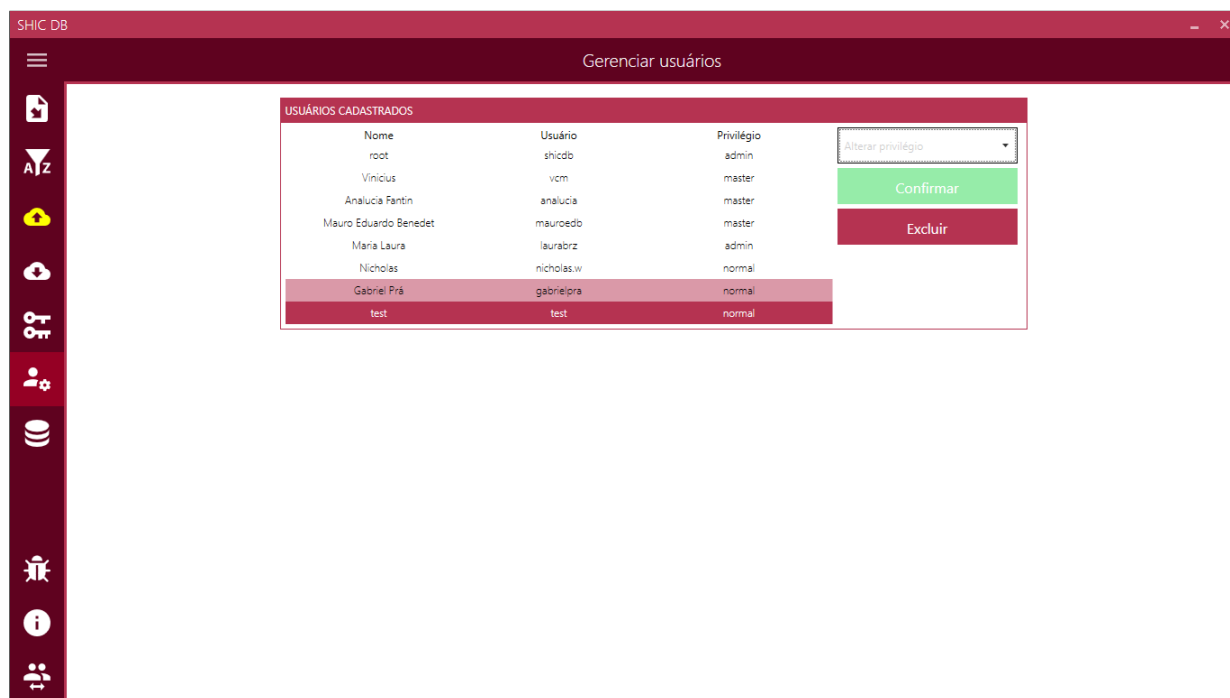
Nesta tela (Figura 26) o usuário consegue visualizar todos os usuários cadastrados no sistema. O usuário *Master* pode realizar duas operações em relação aos usuários cadastrados: *Alterar privilégio* e *Excluir*. Conforme mencionado anteriormente, caso um usuário com privilégio *Normal* ou *Admin* queira trocar de privilégio, é necessário pedir à um usuário *Master* para que este faça esta alteração.

5.2.9 Gerenciar informações

A sétima função disponível é a de *Gerenciar informações*. Ao selecionar esta opção, o usuário é direcionado para a tela de *Gerenciar informações* conforme pode ser visto na Figura 27.

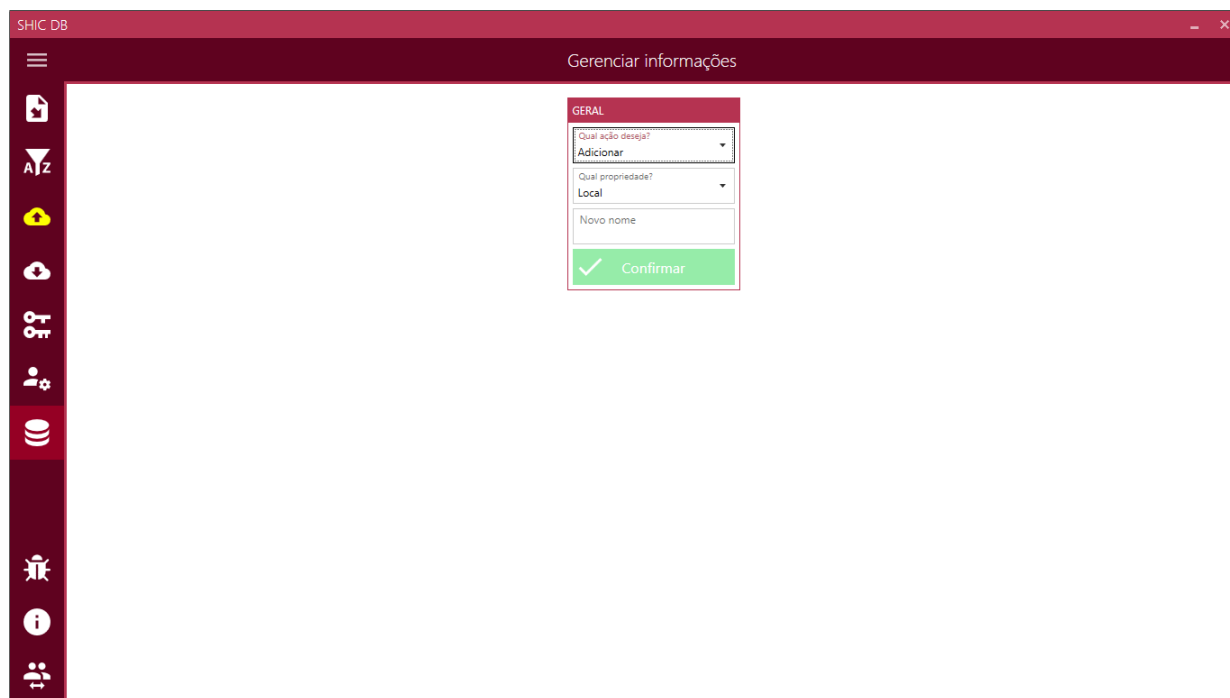
Nesta tela (Figura 27) o usuário consegue realizar operações de adição, edição e exclusão das informações disponíveis no momento de cadastrar novas imagens. Conforme mencionado na seção de *Importar Imagem* 5.2.4, alguns parâmetros só podem ser gerenciados através desta tela, enquanto outros possuem a opção de adição na própria tela de *Importar Imagem* (Figura 18).

Figura 26 – Tela de Gerenciar usuários - ShicDB



Fonte: Arquivo pessoal

Figura 27 – Tela de Gerenciar informações - ShicDB



Fonte: Arquivo pessoal

5.3 Funcionalidades do servidor

Com o intuito de centralizar as informações de maneira fácil e segura para todos os usuários do sistema *ShicDB*, também foi desenvolvido códigos em PHP para rodar em um servidor Web com funções que serão descritas nas próximas seções.

5.3.1 Autenticação

Como mencionado na seção de *Login* (5.2.1), caso a autenticação de um usuário não seja possível utilizando as informações contidas no banco de dados local onde o sistema está sendo executado, uma comunicação é feita com o servidor de modo à tentar autenticar usuário com as informações contidas no banco de dados do servidor.

Já na seção de *Cadastro* (5.2.2), a comunicação com o servidor é sempre necessária de modo à garantir que o novo cadastro de usuário não possua um mesmo email já utilizado anteriormente no sistema.

5.3.2 Sincronizar informações

Sempre que um usuário for autenticado pelo sistema e levado para a tela *Home*, uma comunicação é feita com o servidor de modo à sincronizar todas as informações que aparecem como opção na hora de importar uma imagem. Isto é feito de modo à garantir que o usuário tenha todas as informações atualizadas no momento de cadastrar uma nova imagem.

Lembrando também que essas informações podem ser sincronizados à qualquer momento durante a utilização do sistema através da função *Download* disponível no menu lateral conforme pode ser visto na Figura 17.

5.3.3 Pesquisar imagens

Conforme mencionado na seção de *Pesquisar* (5.2.5), caso o usuário queira atualizar as informações referentes às imagens disponíveis na sua máquina local, basta pressionar o botão de atualização disponível na tela de *Pesquisar*.

Ao realizar esta ação, uma comunicação é feita com o servidor, onde este retorna uma lista contendo todos os registros de imagens disponíveis no banco de dados do servidor, ou seja, o usuário será capaz de acessar todas as informações referentes às imagens que foram sincronizadas com o servidor à partir de qualquer máquina.

Sempre que uma atualização é feita pelo usuário, além das informações das imagens serem enviadas, o ícone referente à cada uma também é enviado, isto foi pensado de maneira à fornecer uma prévia da imagem original ao usuário sem prejudicar o tempo de resposta

do servidor com o sistema desktop, pois um ícone tem um tamanho muito reduzido quando comparado ao tamanho da imagem original.

5.3.4 Download de imagens

Quando um usuário seleciona uma imagem da lista de resultados na tela de *Pesquisar*, caso esta imagem não possua seu arquivo original salvo na máquina onde o sistema está sendo executado, uma opção de *Download* é disponibilizada ao usuário.

Ao pressionar o botão de *Download*, uma requisição é feita ao servidor com a identificação da imagem selecionada, assim o servidor retorna o arquivo original da imagem e o sistema salva este arquivo na pasta padrão onde todos os arquivos de imagem ficam salvos.

5.3.5 Excluir imagem

Outra opção que fica disponível para usuários com privilégio *Master* ou *Admin* quando certa imagem é selecionada na lista de resultados na tela de *Pesquisar* é a de *Excluir* imagem. Ao realizar esta ação, o sistema envia para o servidor a identificação da imagem, deste modo o registro da imagem é excluída do banco de dados do servidor, não sendo mais possível então acessar dados referentes à tal imagem.

5.3.6 Atualizar imagem

A última opção disponível ao selecionar uma imagem na lista de resultados na tela de *Pesquisar* e para usuários com privilégio *Master* ou *Admin* é a de *Editar*. Ao pressionar o botão de *Editar*, realizar as alterações necessárias e salvar, um envio para o servidor é feito contendo as novas informações da imagem selecionada. Ao receber tais informações de uma imagem já cadastrada, o servidor se encarrega de atualizá-las, deste modo, na próxima vez que um usuário realizar a atualização na tela de *Pesquisar*, os dados referentes à esta imagem também serão atualizados no banco de dados local.

5.3.7 Upload de imagens

O servidor ao receber um requisição de *Upload*, fica encarregado de salvar os registros das imagens no seu banco de dados, assim como salvar os respectivos ícones e arquivos originais das imagens. Deste modo, estas imagens se tornarão acessíveis para quaisquer usuários que tenham o sistema desktop instalado.

5.3.8 Gerenciar informações

Sempre que um usuário realiza operações de *Adicionar*, *Editar* ou *Excluir* informações, uma comunicação é feita com o servidor. No caso de o usuário querer adicionar uma nova informação, o servidor primeiro confere em seu banco de dados se tal informação já existe, caso não exista, a informação é adicionada com sucesso, senão o servidor retorna uma mensagem de erro ao sistema.

Ao realizar a operação de *Editar*, o sistema envia ao servidor, a identificação da informação à ser modificada, deste modo o servidor a localiza em seu banco de dados e faz sua atualização, assim, todos os usuários terão certa informação atualizada na próxima vez que entrarem no sistema ou quando utilizarem a função *Download*.

A ação de excluir também envia a identificação da informação à qual se deseja deletar, deste modo o servidor se encarrega de excluí-la, sendo assim tal informação se torna indisponível para todos os usuários.

5.3.9 Gerenciar usuários

Usuários que possuem privilégio *Master* possuem a opção de *Alterar um usuário*, ou seja, alterar o privilégio de tal usuário e a opção de *Excluir* certo usuário. Para qualquer uma das ações, um envio com o identificador do usuário é feito para o servidor. Caso a opção seja *Alterar usuário*, o servidor se encarrega de localizar o usuário com o identificador recebido e alterar seu privilégio de acordo com o novo privilégio que recebeu do sistema. Caso a opção seja *Excluir*, o servidor deleta o usuário do seu banco de dados, impossibilitando-o de realizar futuras operações no sistema.

6 Considerações Finais

Durante o estágio foram utilizados conceitos do processo de desenvolvimento de software que foram aprendidos durante a graduação. A realização de todas as etapas do processo de desenvolvimento de software, da análise à implementação, contribuiu fornecendo uma visão macro das funções de cada etapa do desenvolvimento, como elas são organizadas cronologicamente e quais artefatos são produzidos em cada uma.

Na execução de cada etapa realizada nesse projeto exigiram-se várias horas de pesquisas e estudos, além dos conhecimentos adquiridos ao longo do curso. Assim, com a conclusão desse estágio percebe-se um grande conhecimento e experiência adquiridos na área de desenvolvimento, sendo que todas as etapas foram muito importantes para a conclusão do trabalho.

No decorrer da etapa de implementação do código fonte do sistema verificou-se que a utilização dos frameworks descritos no trabalho, permite que o desenvolvedor passe mais tempo concentrado na lógica de negócio da aplicação, aumentando significativamente a produtividade.

Durante o projeto de estágio foram desenvolvidas as principais funcionalidades do fluxo principal do sistema, porém devido ao tempo disponível para o projeto, as melhorias e novas funcionalidades do sistema serão analisadas e caso sejam viáveis serão implementadas em trabalhos futuros.

No estado final do sistema, ao término desse estágio, já é possível a sua implantação no ambiente do laboratório substituindo a forma atual, contudo, a fase de implantação demanda outras premissas para que seja realizada, tais como: treinamento do pessoal, período de adaptação com o sistema, entre outras, que fogem do escopo deste trabalho.

Em síntese, a realização do estágio permitiu adquirir experiência em análise e prototipação de sistemas, com o desenvolvimento de várias características orais e escritas, necessárias para comunicar e entender as ideias do cliente e negociar o escopo e entregas em virtude do prazo estabelecido para a conclusão do trabalho. Além do mais, foram consolidadas experiências adquiridas anteriormente com programação e execução de testes, sendo concluído a contento e com aquisição de novas experiências profissionais entendendo como proveitosa a realização deste trabalho.

Referências

- [1] R. A. YOUNG, H. D; FREEDMAN, *Física IV: Ótica e Física Moderna 4*. Pearson, 14 ed., 2016.
- [2] G. J. A. A., M. C., W. D. P., B. R. V., S. P. B., F. R. P., D. E., V. M. R., and H. D., “Detecção de falhas de adesão entre mantas poliméricas e dutos usando interferometria de deslocamento lateral,” in *XXIV CONAEND - Congresso Nacional de Ensaio Não Destrutivos e Inspeção e o IEV 2006 - Conferência Internacional sobre Evaluación de Integridad y Extensión de Vida de Equipos Industriales, 2006, São Paulo*.
- [3] L. D. D., “Avaliação do desempenho de um retardador de cristal líquido para aplicação em shearografia,” dissertação (mestrado) - departamento de engenharia mecânica, UFSC, Florianópolis, 2013.
- [4] L. UFSC. <http://www.labmetro.ufsc.br/>, November 2017.
- [5] S. Ian., *Engenharia de Software*. Rio de Janeiro: McGrawHill, 7 ed., 2011.
- [6] P. R. S., *Engenharia de Software: uma Abordagem Profissional*. Pearson Prentice Hall, 9 ed., 2011.
- [7] M. Higor, “Princípios da engenharia de software.” <http://www.devmedia.com.br/principios-da-engenharia-de-software/29630>, 2017.
- [8] E. J. Hélio, “Engenharia de software na prática,” 2010.
- [9] G. G. T. A., *UML 2 - Uma Abordagem Prática*. Editora Novatec, 2 ed., 2011.
- [10] N. Thiago, “A importância dos prototipos no desenvolvimento de sistemas.” <http://thiagonasc.com/desenvolvimento-web/a-importancia-dos-prototipos-no-desenvolvimento-de-sistemas>, 2017.
- [11] O. M. Peter and J. B. Peter, “Software prototyping: A strategy to use when user lacks data processing experience,” *ARPN Journal of Systems and Software*, vol. Vol 2.
- [12] S. Abraham, K. H. F, and SUDARSHAN, *Sistema de Banco de Dados*. Elsevier, 5 ed., 2011.
- [13] D. C. J., *Introdução a Sistemas de Bancos de Dados*. Elsevier, 8 ed., 2004.
- [14] E. Ramez and N. S. B., *Sistema de Banco de Dados*. Pearson, 7 ed., 2005.

-
- [15] D. FERREIRA, “Um modelo Ágil para gestão de projectos de software.” http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_19.pdf, 2005.
 - [16] MySQL, “What is mysql.” <http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>, 2017.
 - [17] W. Luke and T. Laura, *PHP MySQL desenvolvimento Web*. Elsevier, 3 ed., 2005.
 - [18] D. SCHMITZ, “Tudo que você queria saber sobre git e github, mas tinha vergonha de perguntar.” <http://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar>, 2015.
 - [19] BALSAMIQ, “Mockups application overview..” <https://docs.balsamiq.com/desktop/overview>, 2017.