

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CORNÉLIO PROCÓPIO  
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GIOVANNA MAIOQUE

**SISTEMA GERENCIADOR DE ACHADOS E PERDIDOS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2015

GIOVANNA MAIOQUE

## **SISTEMA GERENCIADOR DE ACHADOS E PERDIDOS**

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

Orientadora: Prof<sup>a</sup>. Adriane Carla Anastácio da Silva

CORNÉLIO PROCÓPIO

2015



Ministério da Educação

**Universidade Tecnológica Federal do Paraná**

Câmpus Cornélio Procópio

Nome da Diretoria

Nome da Coordenação

Nome do Curso



---

---

## **FOLHA DE APROVAÇÃO**

Utilizar o modelo utilizado pelo curso – elemento obrigatório

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por ter me dado inteligência, força e persistência, principalmente nos momentos mais difíceis.

Agradeço aos meus pais, Sidney e Rosane, por sempre terem lutado para que eu tivesse uma educação de qualidade e também pela dedicação durante todo o período da faculdade.

Agradeço minhas avós, Nair e Alba, que incondicionalmente me deram apoio e sempre estiveram pedindo a Deus a força para que eu pudesse concluir esta etapa.

Agradeço ao meu namorado, João Henrique, pela paciência e apoio durante todos esses anos, por ter me estendido a mão todas às vezes, em que eu quis desistir e me incentivado a continuar.

Agradeço a minha professora orientadora, Adriane Anastácio, por ter aceitado o meu convite de ser minha orientadora e me guiar por todo este trabalho.

Aos meus amigos, em especial Viviane Gobetti, Elivelton Mendes e Gabriel Guimarães, que sempre me auxiliaram em minhas inúmeras dúvidas e compartilharam alegrias, tristezas, dependências e aprovações.

Enfim, a todos os que por algum motivo contribuíram para a realização deste trabalho.

*“Que vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”*

**Charles Chaplin.**

## RESUMO

MAIOQUE, Giovanna. **Sistema Gerenciador de Achados e Perdidos**. 2015. 42 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Diariamente os alunos perdem e encontram objetos por toda a universidade. Visando mitigar esse problema, foi desenvolvido em um sistema web, para auxiliar no controle e na rápida divulgação dos itens encontrados e perdidos dentro do espaço da universidade. O sistema disponibiliza uma ferramenta de busca para facilitar o acesso dos usuários ao banco de dados. Para desenvolver este projeto foi utilizado o processo de desenvolvimento Scrum Solo, na implementação foi utilizada a linguagem PHP e o banco de dados MySQL, para o armazenamento de dados.

**Palavras-chave:** software de gerenciamento; achados e perdidos; busca de objetos.

## **ABSTRACT**

MAIOQUE, Giovanna. **Sistema Gerenciador de Achados e Perdidos**. 2015. 42 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Every day the students lost and found objects throughout the university. To mitigate this problem, it was developed into a web system to assist in the control and rapid dissemination of items found and lost within the university space. The system provides a search engine to facilitate user access to the database. To develop this project we used the development process Scrum Solo, implementation we used the PHP language and the MySQL database for data storage.

**Keywords:** management software ; lost and found; search objects.

## LISTA DE FIGURAS

Figura 1 - Scrum Solo.....	18
Figura 2 - Modelo de arquitetura MVC (Model-View-Controller).....	21
Figura 3 - Diagrama de Casos de Uso.....	<b>Error! Bookmark not defined.</b>
Figura 4 - Diagrama de Pacotes.....	24
Figura 5 - Diagrama de Classes do pacote View.....	25
Figura 6 - Diagramas de Classes do pacote <i>Model</i> .....	26
Figura 7 - Diagrama de Classes do pacote Controle.....	27
Figura 8 - Diagrama de sequência Enviar mensagem.....	28
Figura 9 - Modelo Entidade Relacionamento (MER).....	29
Figura 10 - Tela Inicial.....	31
Figura 11 - Tela pós busca.....	31
Figura 12 - Tela de um objeto Achado.....	32
Figura 13 - Tela de login.....	32
Figura 14 - Tela pós <i>login</i> .....	33
Figura 15 - Tela da aba ADM.....	33
Figura 16 - Tela de busca de usuários.....	34
Figura 17 - Tela de busca de objetos.....	34
Figura 18 - Tela administração de objeto.....	35
Figura 19 - Tela de envio de mensagem.....	35
Figura 20 - Tela de cadastro de usuários.....	36
Figura 21 - Tela de cadastro de objeto.....	36

## LISTA DE SIGLAS

CPTM	Companhia Paulista de Trens Metropolitanos
CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
DER	Diagrama de Modelagem Entidade-Relacionamento
FACCAMP	Faculdade Campo Limpo Paulista
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transport Protocol</i>
IBM	<i>International Business Machines</i>
IDE	<i>Integrated Development Environment</i>
MVC	<i>Model View Controller</i>
PHP	<i>Hypertext Preprocessor</i>
SGBD	Sistema Gerenciador de Banco de Dados
TCC	Trabalho de Conclusão de Curso
TDAH	Transtorno de Déficit de Atenção
UML	Unified Modeling Language
UTFPR	Universidade Tecnológica Federal do Paraná

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>10</b>
1.2 JUSTIFICATIVA.....	11
1.3 OBJETIVO GERAL.....	11
1.3.1 Funcionalidades do Sistema.....	11
1.4 ORGANIZAÇÃO DO TRABALHO.....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>14</b>
2.1 BUSCA OTIMIZADA.....	14
2.2 OBJETOS PERDIDOS .....	15
2.3 DEPARTAMENTOS DE ACHADOS E PERDIDOS .....	15
2.4 TRABALHOS RELACIONADOS .....	16
<b>3 METODOLOGIA.....</b>	<b>17</b>
<b>3.1 MODELO DE PROCESSO DE DESENVOLVIMENTO .....</b>	<b>17</b>
3.1.1 Métodos Ágeis .....	17
3.1.2 Scrum Solo .....	17
3.2 FERRAMENTAS E TECNOLOGIAS UTILIZADAS .....	19
3.2.1 Modelagem.....	19
3.2.2 Implementação.....	20
3.2.3 Banco de Dados .....	21
3.3 ARQUITETURA DO SISTEMA.....	21
<b>4 MODELAGEM E ESTRUTURA DO SISTEMA.....</b>	<b>23</b>
4.1 DIAGRAMA DE CASO DE USO .....	23
4.2 DIAGRAMA DE PACOTES .....	24
4.3 DIAGRAMAS DE CLASSE.....	25
4.4 DIAGRAMA DE SEQUÊNCIA.....	28
4.5 MODELO ENTIDADE-RELACIONAMENTO.....	29
<b>5 RESULTADOS OBTIDOS .....</b>	<b>31</b>
<b>6 CRONOGRAMA REALIZADO .....</b>	<b>37</b>
<b>7 CONSIDERAÇÕES FINAIS.....</b>	<b>38</b>
<b>REFERÊNCIAS.....</b>	<b>39</b>

## 1 INTRODUÇÃO

Em diversos locais é muito comum pessoas perderem objetos a todo o momento e serem encontrados por alguém que quer devolvê-los. A Companhia Paulista de Trens Metropolitanos (CPTM) recebe cerca de 6.500 objetos perdidos mensalmente, aproximadamente 80% foram devolvidos graças a informações nos objetos que identificam o dono. Os funcionários da CPTM entram em contato com o dono informando que encontraram o objeto perdido. Deste modo, a pessoa entra em contato para haver uma devolução. Contudo, muitas pessoas não tentam recuperar o que foi perdido (SILVA, 2012).

A CPTM mantém um sistema visando à ligação entre quem perdeu e encontrou os objetos. Os objetos ficam guardados em setores por tempo determinado pelas instituições.

O sistema de Achados e Perdidos da Universidade de São Paulo, do aluno Everton Topan da Silva e seu orientador Dr. Alfredo Goldman, localiza objetos perdidos dentro da universidade (SILVA, 2013). No entanto, o sistema não possui um controle sobre onde os objetos encontrados estão armazenados, na Universidade de São Paulo não há um setor responsável pela armazenagem do objeto. Neste sistema, o próprio aluno que encontra o objeto fica de posse até que o dono visualize o objeto no sistema e entre em contato. Observamos a desvantagem do aluno que não conseguir entrar em contato, pessoalmente, com o aluno que encontrou o objeto ou poder não encontrar o objeto no sistema, considerando também, o uso de palavras incorretas na busca.

O sistema gerenciador de achados e perdidos visa suprir a necessidade de um software no setor da Universidade Tecnológica Federal do Paraná (UTFPR). Atualmente, os objetos ficam armazenados no Departamento de Recursos Didáticos, em uma prateleira, caso o dono apareça, ele retira o objeto sem nenhum controle do setor. O gerenciador de achados e perdidos desenvolvido possui um controle sobre quem achou, quem resgatou, o local e a data onde o objeto foi encontrado.

## 1.2 JUSTIFICATIVA

O sistema *web* implementado visa atender as necessidades dos alunos, professores e servidores, que perdem seus objetos dentro da UTFPR, Campus Cornélio Procópio, centralizando a procura e entrega dos objetos, em um só local.

O sistema permite que os objetos perdidos sejam buscados e encontrados com maior praticidade e agilidade. Ainda, oferece ao setor responsável um maior controle, no caso da CPTM, os objetos são arquivados de forma manual, em ordem alfabética, no caso de documentos e papéis, dentro de envelopes, causando demora em sua procura. O objeto perdido, como acontece na Faculdade Campo Limpo Paulista (FACCAMP), permanece por 90 dias, e depois será doado para instituições assistenciais, exceto os documentos.

O sistema poderá ser utilizado para atender inclusive outros campus da universidade, em muitos casos com as características similares, recolhem os objetivos esquecidos e o armazenam em um setor ou departamento. Deste modo, pretendemos melhorar a rotina dos departamentos responsáveis dos respectivos locais.

## 1.3 OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver um software para realizar o gerenciamento de objetos achados e perdidos na UTFPR, Campus Cornélio Procópio.

### 1.3.1 Funcionalidades do Sistema

- **Create, Read, Update e Delete (CRUD) de Usuários:** permite ao usuário se cadastrar, ler, atualizar e deletar seus dados pessoais. Visando o controle sobre sua conta. Os usuários são divididos em: usuário comum com

funcionalidades restritas para seu cadastro e usuário administrador com acesso a todas as funcionalidades do sistema. Os usuários comuns não poderão excluir suas contas.

- **CRUD de Objetos perdidos:** Permite ao usuário, desde que *logado* no sistema, cadastrar um objeto perdido, consultar, atualizar e excluir o mesmo. O usuário comum, após cadastrar um objeto, somente pode consultá-lo. O usuário administrador tem acesso pleno aos objetos criados por ele e por outros usuários.
- **CRUD de Objetos encontrados:** Permite ao usuário, desde que *logado* no sistema, cadastrar um objeto encontrado, consultar, atualizar e excluir o mesmo. O usuário comum, após criar um objeto, somente pode consultá-lo. O usuário administrador tem acesso pleno aos objetos criados por ele e por outros usuários.
- **Comunicar ao usuário que um objeto foi encontrado:** O sistema permite o administrador enviar uma mensagem ao usuário que criou um objeto perdido, avisando-o que o objeto foi encontrado. O dono pode se dirigir ao setor responsável para retirá-lo.
- **Buscar um objeto:** O sistema permite ao usuário buscar um objeto perdido ou encontrado no banco de dados, antes de cadastrá-lo, ou realizar uma consulta.

#### 1.4 ORGANIZAÇÃO DO TRABALHO

O segundo capítulo é destinado à fundamentação teórica abordada neste trabalho: busca inteligente, objetos perdidos e uma breve explicação sobre os setores de achados e perdidos.

O terceiro capítulo apresenta à metodologia que foi aplicada ao projeto, descrevendo o modelo de processo de desenvolvimento Scrum Solo, as tecnologias e suas respectivas ferramentas necessárias para: modelagem (UML e Astah), implementação (PHP, CSS, HTML, javascript), servidor web (Apache), banco de dados (MySQL). O capítulo aborda ainda a arquitetura MVC, necessária para a elaboração da ferramenta.

O quarto capítulo aborda os *sprints* (tarefas) que foram realizados como criação do diagrama de casos de uso, diagrama de pacotes, diagrama de classes, diagrama de sequência e modelagem do banco de dados, por meio do diagrama de modelagem entidade-relacionamento. Apresenta a estrutura do sistema.

No quinto capítulo encontram-se os resultados obtidos, a partir da apresentação e descrição das telas do sistema.

O sexto capítulo apresenta o cronograma das atividades realizadas. E nos últimos capítulos encontram-se as considerações finais e as referências utilizadas.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta a fundamentação teórica, trazendo os principais tópicos que permeiam o trabalho.

### 2.1 BUSCA OTIMIZADA

Diante de um grande volume de dados disponíveis na *web*, a ferramenta de busca otimizada se torna praticamente obrigatória à sistemas onde o usuário requer a busca de informações específicas, a partir de palavras chaves.

Segundo Satnam (2009) uma consulta requer uma palavra chave ou uma frase, resultado de uma busca para recuperar documentos relevantes à consulta. A lista de documentos é geralmente devolvida na ordem de relevância que eles apresentam na consulta.

No entendimento de Zaninelli e Catarino (2004) utilizando-se de palavras chaves, operadores e outros recursos, a estratégia de busca forme uma expressão para recuperar as informações relevantes na base de dados. Sendo que para Carneiro (2009) uma expressão mal escrita pode gerar custos e desempenho insuficiente pela utilização de caminhos inadequados do otimizador.

Para Choo( 2003) recuperar uma informação com a finalidade de produzir novo conhecimento ou tomar decisões, pessoais ou administrativas, permite tornar acessível ao usuário as informações desejadas, para que sejam atendidas suas necessidades.

Todas as estratégias para recuperar de forma inteligente os dados são baseadas em uma *query* (busca), comparando-as com documentos armazenados no banco de dados (BEPPLER, 2002).

Na ferramenta construída foi utilizado o conceito de busca inteligente, cada vez que um objeto for perdido ou encontrado, o usuário necessita fazer uma busca, a partir de uma descrição consistente, o usuário poderá encontrar o máximo de resultados compatíveis com sua busca, de maneira rápida e precisa.

## 2.2 OBJETOS PERDIDOS

O Dr. Norton Sayeg (2014) afirma que perder objetos é um processo comum à medida que as pessoas vão envelhecendo. Observamos, que não é um comportamento comum, apenas em idosos, todavia, ocorre em pessoas de qualquer idade. As causas do esquecimento de objetos podem estar relacionadas a alguns medicamentos, a ansiedade, ao uso de álcool, entre outros fatores, incluindo, desde uma simples distração até uma doença grave, como depressão, mal de Alzheimer ou Transtorno de Déficit de Atenção (TDAH).

Uma alternativa mais próxima da realidade e de possível instalação num ambiente universitário é o gerenciamento destes objetos perdidos, a partir de ferramentas web ou de aplicativos para dispositivo móvel, depositados em departamentos de achados e perdidos das próprias universidades.

## 2.3 DEPARTAMENTOS DE ACHADOS E PERDIDOS

Diariamente, são perdidos muitos objetos, em locais como: metrô, aeroportos, shoppings e lugares com grande circulação de pessoas. Nem todos os lugares contam com setores especializados em receber os objetos encontrados, catalogá-los e entregar aos donos.

O Aeroporto de Guarulhos (2015), possui um setor que recebe os objetos encontrados listando-os em um formulário próprio e enviando-os a Infraero para a guarda. Após 3 meses todos os objetos são encaminhados a instituições cadastradas. Quando um objeto é perdido o proprietário deve entrar em contato com um funcionário da Infraero com as características, data e horário aproximado que o objeto foi perdido e a empresa entra em contato caso o objeto seja encontrado. Para o resgate o proprietário deve comparecer ao balcão de informações com documento com foto e solicitar a devolução.

Apresentamos em nosso trabalho algumas alternativas existentes, contudo, consideramos casos particulares de perda de objetos com tratamentos diferenciados a cada caso.

## 2.4 TRABALHOS RELACIONADOS

Uma alternativa criada para solucionar o problema de perda de objetos foi a criação de um *gadget* (equipamento) chamado Tile, utilizado em qualquer objeto, quando o dono não sabe onde ele está, aciona o chip integrado ao Tile, via iPhone, e o aplicativo indica se está perto ou não do objeto, em casos que o sistema não consegue localizar, é possível utilizar um sinal sonoro (TILE, 2014).

Observamos que perder objetos é comum no dia a dia das pessoas. O fato de ter seus objetos conectados a aparelhos, como o Tile, é uma prática quase que inviável, visto que, não poderá ser instalado na totalidade, dificultando sua utilização.

No caso do metrô de São Paulo, ele disponibiliza um sistema em que podem ser encontrados objetos pelos quais os donos podem ser identificados, como documentos, livros, entre outros (METRÔ SP, 2014). Neste sistema o usuário entra com o nome que deseja procurar e verifica se há a ocorrência deste nome nos registros. Os objetos, com identificação ou não, podem ser procurados nas estações e se encontram disponíveis por 60 dias antes de serem doados.

No sistema dos Correios é disponibilizada a função de encontrar documentos perdidos, realizando a busca pelo nome do proprietário e o boletim de ocorrência relativo à perda do documento (CORREIOS, 2014). Contudo, não permite a uma pessoa que tenha encontrado o objeto, registrá-lo via internet. Neste caso, a pessoa deve se dirigir a uma agência dos Correios, mais próxima.

Na Universidade de São Paulo (USP) os alunos que encontrarem um objeto devem se dirigir a guarda universitária para que seja feito um boletim da ocorrência. Caso o dono apareça é feito um termo de responsabilidade pelo recebimento (PUSPLQ,2014). A USP possui um *software* livre que auxilia na busca pelos objetos ou donos dos objetos perdidos, contudo, encontra-se na fase inicial e algumas funcionalidades ainda não funcionam como propostas.

### 3 METODOLOGIA

Nesta seção são apresentados o modelo de processo de desenvolvimento, as ferramentas, as tecnologias e a arquitetura da ferramenta. A opção pela escolha de cada uma é justificada a partir das características da ferramenta desenvolvida, um Sistema Gerenciador de Achados e Perdidos.

#### 3.1 MODELO DE PROCESSO DE DESENVOLVIMENTO

As atividades para o desenvolvimento do sistema proposto serão baseadas na metodologia ágil *Scrum Solo*, adaptada a projetos desenvolvidos por apenas um responsável, para todas as suas fases.

##### 3.1.1 Métodos Ágeis

A metodologia ágil tem como pontos chaves evitar gastos como documentação excessiva, diretamente centrada nas pessoas e no seu potencial de produção, conduzindo o projeto ao sucesso.

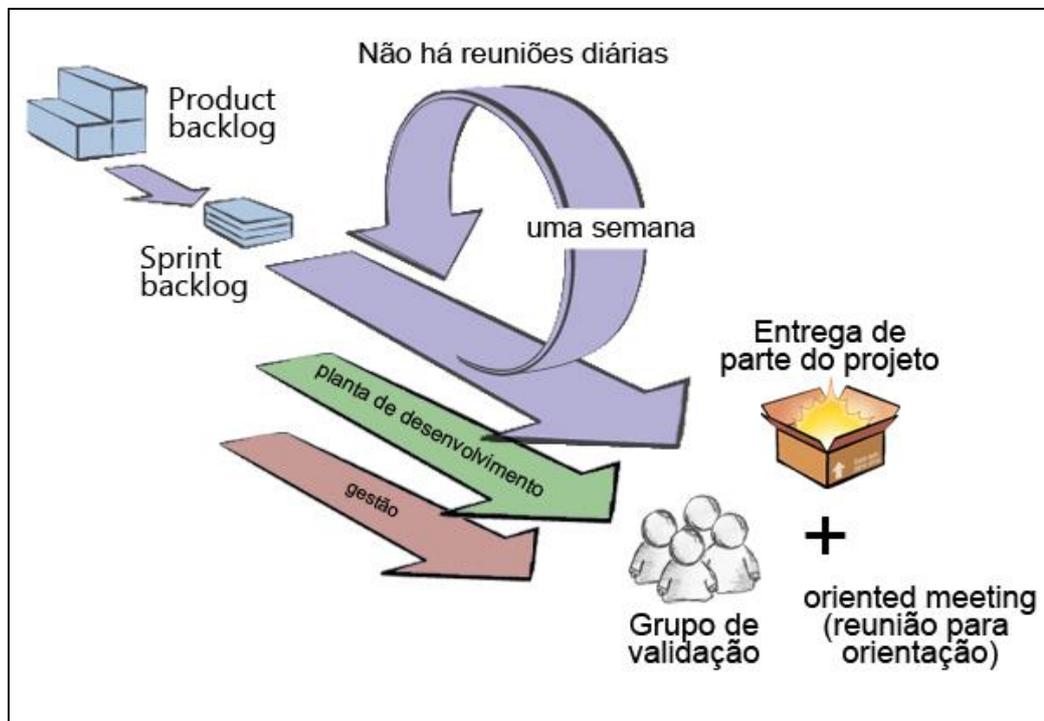
A utilização da metodologia ágil grega agilidade à equipe, tornando-as auto gerenciáveis, com objetivo a qualidade no produto final (VAZ; FERREIRA, 2013).

##### 3.1.2 Scrum Solo

O *Scrum* vem sendo utilizado com sucesso para o gerenciamento de desenvolvimento iterativo e incremental de softwares. Segundo Ferreira *et ali* (2005), as principais características do *Scrum* são:

- Processo ágil para gerenciamento e controle do desenvolvimento de projetos;
- Processo que mantém o controle diante dos conflitos gerados diante de interesses e necessidades conflitantes
- Procura manter uma cooperação entre os envolvidos e aumentar a comunicação;
- Processo flexível para pequenos e grandes projetos.

O *Scrum Solo* é uma adaptação do Scrum, criado com intuito de desenvolver o processo com apenas um programador, as características principais do *Scrum*, busca auxiliar o programador a planejar e executar as atividades, sem gerar altos custos. A Figura 1 ilustra o *Scrum Solo*, segundo Fabri (2012).



**Figura 1 - Scrum Solo.**  
**Fonte: Fabri (2012).**

Para Fabri (2012) o *Scrum Solo* é uma versão adaptada do *Scrum*, que utiliza alguns de seus princípios como as reuniões para *sprint backlog*, onde os requisitos selecionados para um log são realizados semanalmente, com a participação do aluno e do professor orientador; e o *Product backlog* os requisitos necessários para o software que podem ser alterados ao longo do projeto, acordados entre todos nas reuniões. As retrospectivas de *sprints* são realizadas

durante as reuniões semanais do aluno-programador com o orientador, considerando-o que é o único responsável por todos os cargos no *Scrum Solo*.

A escolha pelo *Scrum Solo* considerou as características da metodologia ágil, atendendo as necessidades de apenas uma pessoa, para desempenhar todas as funções, no desenvolvimento do sistema proposto.

## 3.2 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

As ferramentas e tecnologias que foram utilizadas no desenvolvimento do Sistema Gerenciador de Achados e Perdidos estão descritas nesta seção, dividida em modelagem, implementação e banco de dados.

### 3.2.1 Modelagem

A UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para a modelagem de sistemas computacionais utilizando o paradigma de orientação a objetos (GUEDES, 2005).

O Sistema Gerenciador de Achados e Perdidos tem como objetivo utilizar o conjunto de notações UML, que permite ao desenvolvedor uma pré-visualização do sistema em diagramas padronizados.

O *Astah Community* possui várias funções como elaborar diagramas, exportar e imprimir imagens dos diagramas, além de possui uma interface de fácil uso (ALMEIDA, 2013). Este *software* foi escolhido por ser uma ferramenta gratuita, que possui as funcionalidades necessárias para a elaboração dos diagramas de maneira fácil de utilizar, como na elaboração dos diagramas UML e modelagem de banco de dados relacional.

### 3.2.2 Implementação

O HTML (*HyperText Markup Language*) é uma linguagem utilizada para criar documentos na *World Wide Web* (WWW). Segundo Soares (2013) utiliza *tags* para estruturar as páginas *web*, motivo que foi definida para este projeto.

O CSS (*Cascading Style Sheets*) foi criado para estilizar as páginas do HTML, mostrando ao navegador como serão posicionados e decorados os elementos. Este fato foi considerado para estilizar as páginas criadas neste trabalho. Com ele é possível formatar textos, figuras, cor de fundo, entre outros recursos que envolvem a estética da página *web*. (SOARES, 2013).

O PHP (*Hypertext Preprocessor*) é uma linguagem específica para *web*, de código fonte aberto, *open source*, ou seja, pode ser baixado, reescrito e aberto. Outra característica importante o código PHP pode ser mesclado com o HTML, onde o código PHP é interpretado e gera um HTML, apresentado a saída ao visitante do *website* em seu navegador (WELLING, 2002). A opção pelo PHP neste projeto considerou a simplicidade na criação de páginas *web*.

A linguagem de programação *JavaScript* é baseada em Java, desenvolvida pela Netscape Communications, capacita seus produtos em *scripting*. A linguagem que controla diversos elementos da página HTML quando inserida no código (MEDEIROS; OLIVEIRA; SILVA, 2008). Neste projeto foi utilizada como plataforma de integração com o PHP, para a criação das páginas interativas do sistema.

O Apache é um software livre e de uso gratuito. É um servidor http (*hyper text transport protocol*) robusto que permite criar domínios com segurança (BRAMBILA; MAHLMANN, 2006). O Apache fez a comunicação entre o navegador e a aplicação, recebendo as requisições dos navegadores, processando e enviando os resultados aos navegadores que fizeram a requisição no sistema proposto.

A IDE Eclipse foi utilizada para implementação, desenvolvido pela IBM e doada para todos. É uma plataforma de código aberto para desenvolvimento que pode ser utilizada para criar *plu-ings* para várias linguagens como C/C++, PHP, Java, COBOL, entre outras (LUCKOW; MELO, 2010). A utilização da IDE Eclipse se deu por conta dela permitir uma flexibilidade maior em comparação às IDEs que são utilizadas para implementação e por ser gratuita.

### 3.2.3 Banco de Dados

O MySQL é um banco de dados rápido, de distribuição gratuita e simples. Atualmente, é o mais popular de sua categoria, utilizado para pequenas e grandes requisições, necessita de poucos recursos do hardware em comparação com outros SGBDs disponíveis no mercado (OLIVEIRO, 2001).

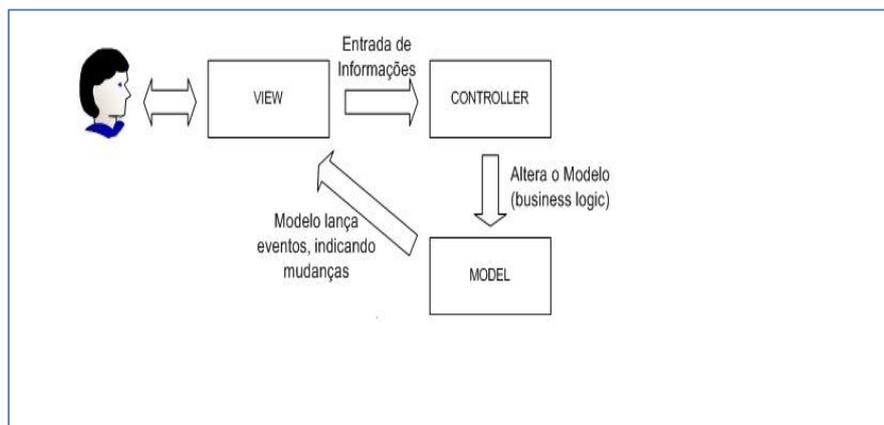
O banco de dados MySQL permite que o software fique mais rápido e fácil de gerenciar, suportando diferentes ambientes, ferramentas, entre outros. Ele foi utilizado para armazenamento de dados do projeto.

### 3.3 ARQUITETURA DO SISTEMA

A arquitetura apresenta a ligação do *software* com seus componentes, propriedades externas e caso houver, ligação com outros *softwares*.

No sistema implementado a arquitetura a ser utilizada é do padrão MVC (*Model-View-Controller*), o sistema é dividido em três partes, caso ocorra mudança em uma camada, as outras camadas não são afetadas.

A Figura 2 ilustra o funcionamento do padrão MVC com base em seus componentes principais.



**Figura 2 - Modelo de arquitetura MVC (Model-View-Controller).**  
Fonte: ALMEIDA (2014).

Observe que o modelo separa a interface da navegação e do comportamento.

De acordo com Junior e Fortes (2007), o modelo MVC possui: o componente *Model* (modelo) possui toda a lógica de negócio específica para o sistema, contendo, todos os objetos e dados externos; o componente *View* (visão) possui a apresentação da lógica com as funções da interface; e o *Controler* (controlador) responsável pelo acesso ao *Model* e a *View*, coordenando os dados que trafegam entre eles.

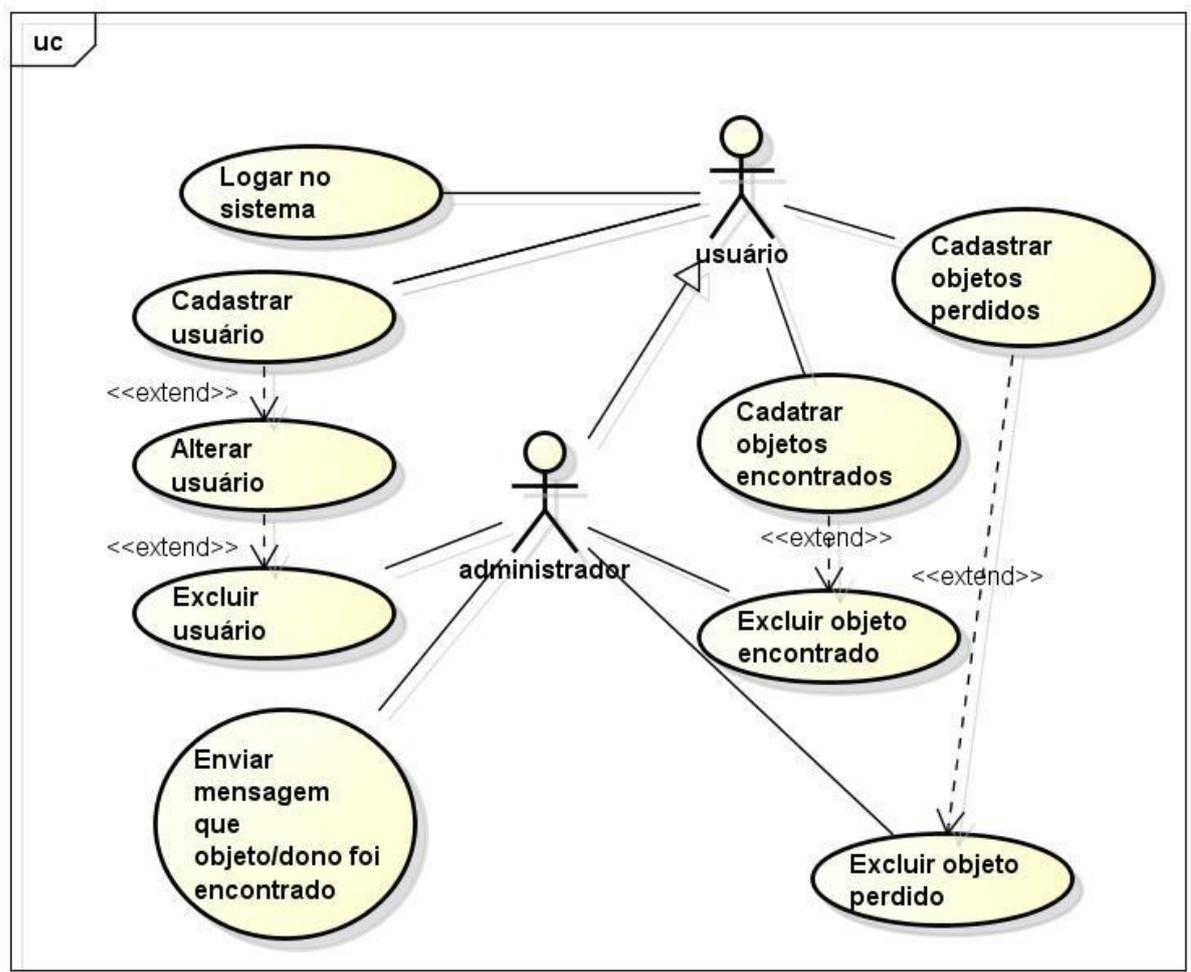
A *View* gera os eventos e o *Controler* modifica o estado do *Model* se necessário. As características do MVC foram decisivas para a escolha desta arquitetura na elaboração do sistema.

## 4 MODELAGEM E ESTRUTURA DO SISTEMA

Este capítulo apresenta os principais diagramas utilizados no desenvolvimento do sistema e sua estrutura.

### 4.1 DIAGRAMA DE CASO DE USO

A Figura 3 apresenta as principais funcionalidades do sistema no Diagrama de Casos de Uso.



powered by Astah

Figura 3 - Diagrama de Casos de Uso.

O diagrama de caso de uso mostra as funcionalidades no qual o usuário tem acesso. As funcionalidades são apresentadas de forma geral, o diagrama de caso de uso foi modelado de forma que qualquer pessoa possa ler e entender as funcionalidades, até mesmo um leigo.

## 4.2 DIAGRAMA DE PACOTES

O Diagrama de Pacotes organiza o sistema, distribuindo-o em classes do projeto, conforme suas características, seguindo a arquitetura MVC. Os pacotes utilizados foram:

- Pacote Controle possui as classes de acesso ao banco de dados, com funcionalidades de inserir, atualizar ou excluir dados de um objeto. Estas classes são responsáveis por modelar o banco de dados, executa comandos quando requeridos pelas classes do pacote *Model*.
- Pacote *Model* é o principal pacote do sistema, possui as regras de negócio. As classes deste pacote processam os eventos que podem gerar alterações no banco de dados, enviando as requisições às classes do pacote Controle ou manipula as classes do pacote *View*.
- Pacote *View* possui as classes que lidam diretamente com o usuário. Elas recebem as ações dos usuários e instanciam as classes do pacote *Model*, solicita alterações ou ações ao banco de dados.

A Figura 4 apresenta o Diagrama de Pacotes do sistema

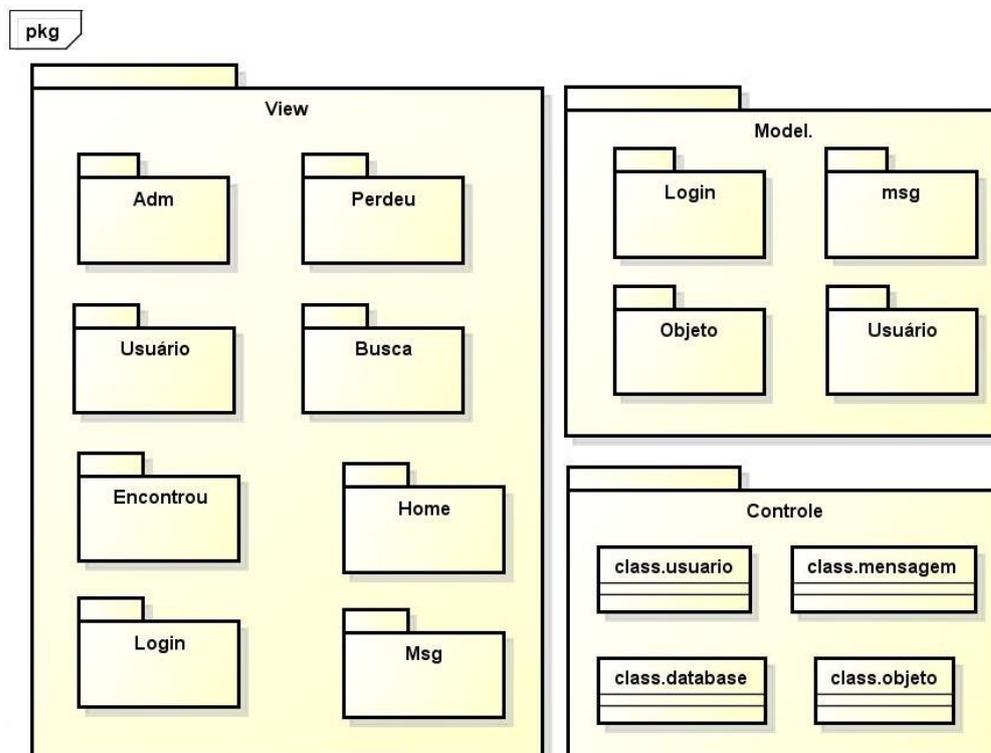


Figura 4 - Diagrama de Pacotes.

### 4.3 DIAGRAMAS DE CLASSE

A Figura 5 apresenta o Diagrama das Classes do pacote *View*.

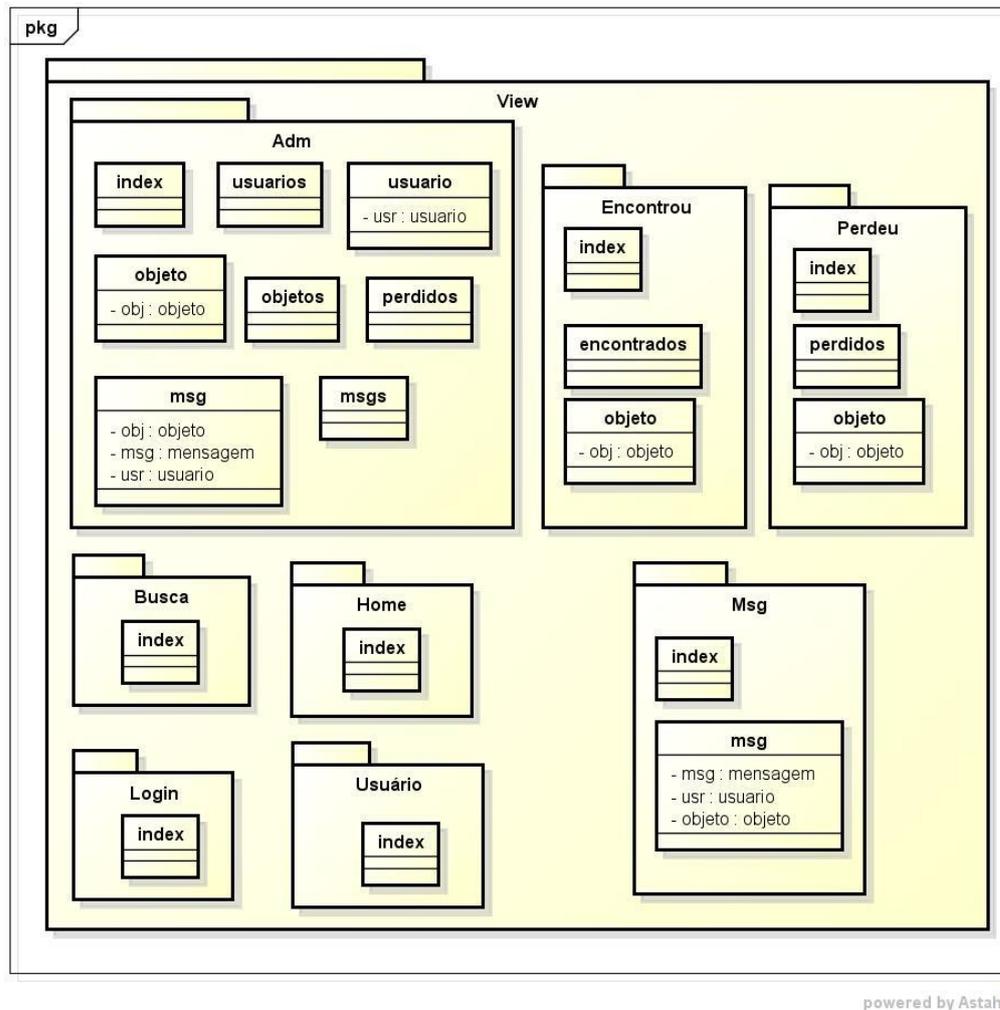


Figura 5 - Diagrama de Classes do pacote *View*.

O pacote *View* apresentado na figura 5 possui os pacotes: *Adm*, *Busca*, *Home*, *Login*, *Usuário*, *Encontrou* e *Perdeu*, com suas respectivas classes de interface. Estas classes possuem código PHP com CSS e HTML. Todos os pacotes possuem sua classe *index*, principal tela mostrada ao usuário, chama as outras classes do pacote *View* quando solicitado.

O pacote *Adm* possui as interfaces referentes ao usuário administrador. A classe *msg* possui a interface onde o administrador envia a mensagem ao usuário. A classe *msgs* mostra todas as mensagens recebidas pelo administrador. As interfaces *objeto*, *perdidos*, *usuário* são utilizadas para inserir dados no banco e as interfaces

objetos e usuário são para excluir e alterar dados no banco, as funcionalidades são exclusivas do usuário administrador.

O pacote Busca possui a interface onde o usuário pode realizar a busca de um objeto perdido ou encontrado no banco de dados.

O pacote *Home* possui a página inicial do sistema.

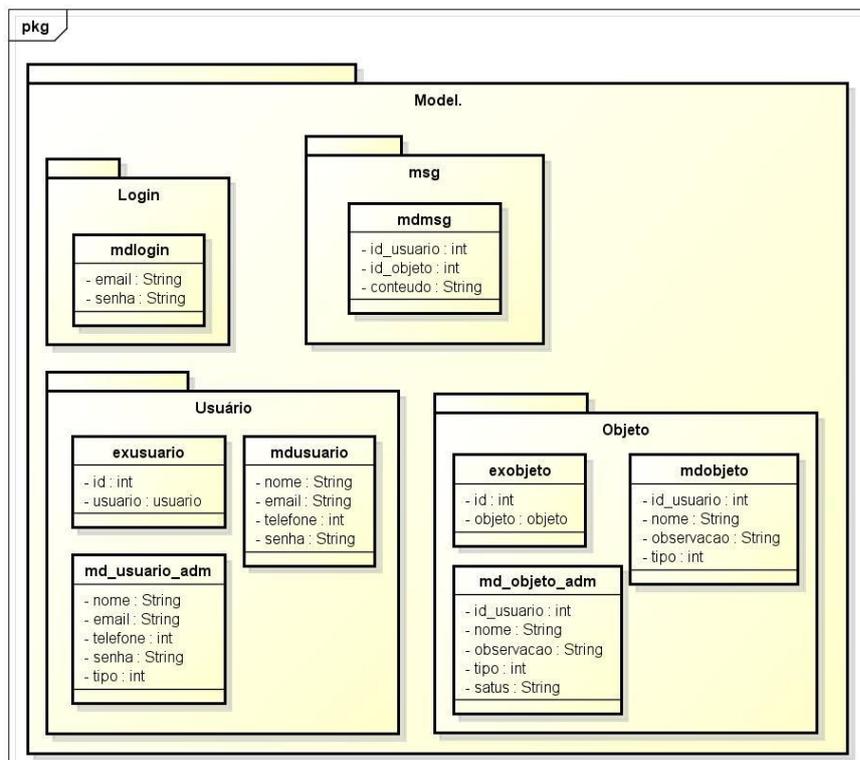
O pacote *Login* possui a tela onde o usuário pode se logar no sistema.

O pacote Encontrou possui na interface Encontrados, uma lista de objetos encontrados cadastrados, a interface objeto mostra um objeto com suas informações e a opção de envio de mensagem ao administrador.

O pacote Perdeu é semelhante ao Encontrou, possuindo as mesmas funcionalidades, mas referindo-se aos objetos perdidos.

O pacote Usuário possui a interface de Cadastro de usuário.

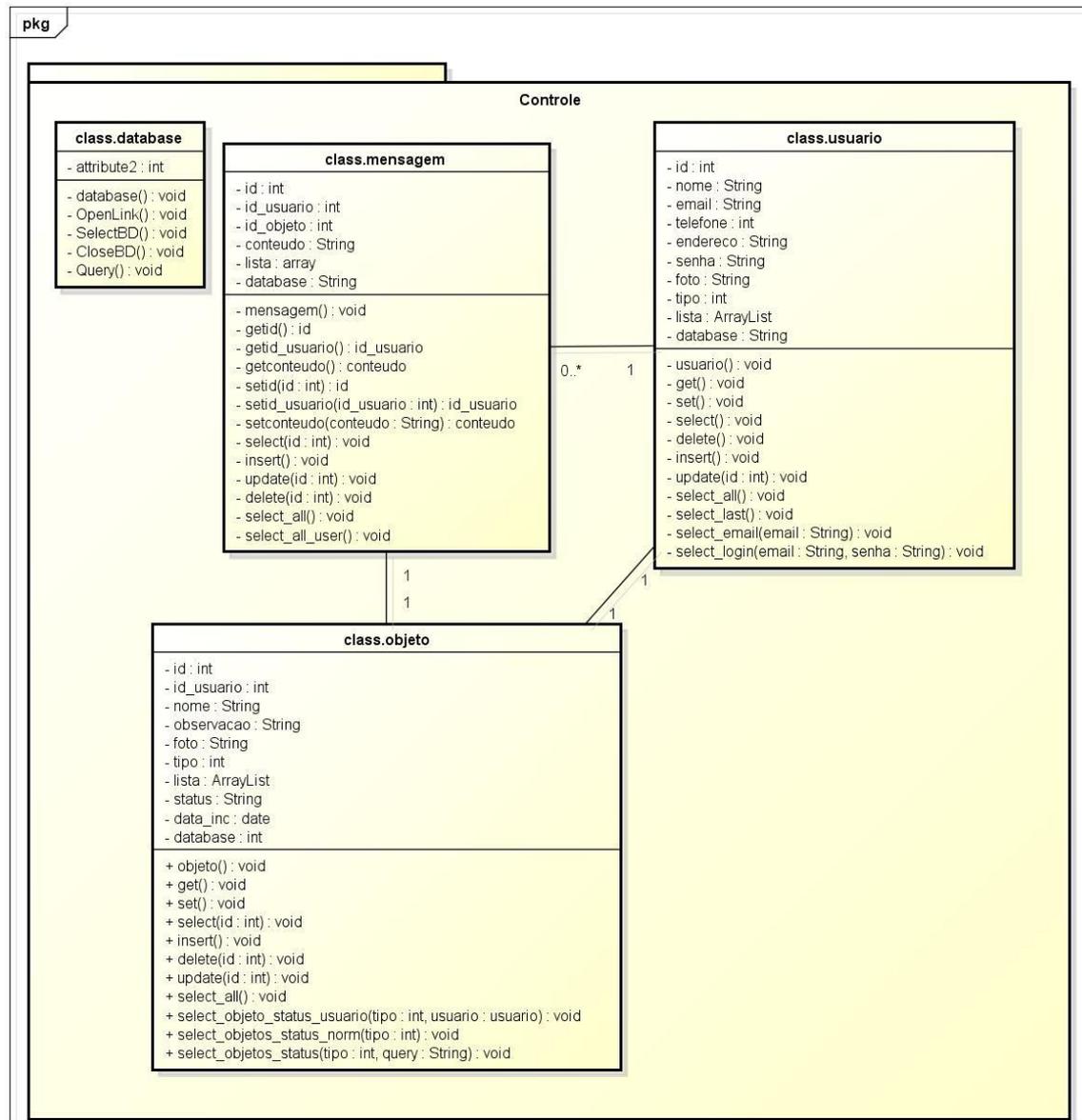
O pacote *Model* apresentado na Figura 6 possui classes que recebem informações do pacote *View*, as processam, decide o que cada entrada significa e as enviam para as classes do pacote *Controler*, determinando o fluxo da apresentação, conforme a regra de negócio a ser utilizada. Estas Classes são responsáveis por instanciar os objetos e determinar o fluxo da aplicação. Há uma classe para cada funcionalidade do sistema.



powered by Astah

Figura 6 - Diagramas de Classes do pacote *Model*.

O pacote *Controler*, representado na Figura 7 possui as classes de persistência, assumindo a conexão com o banco de dados.



powered by Astah

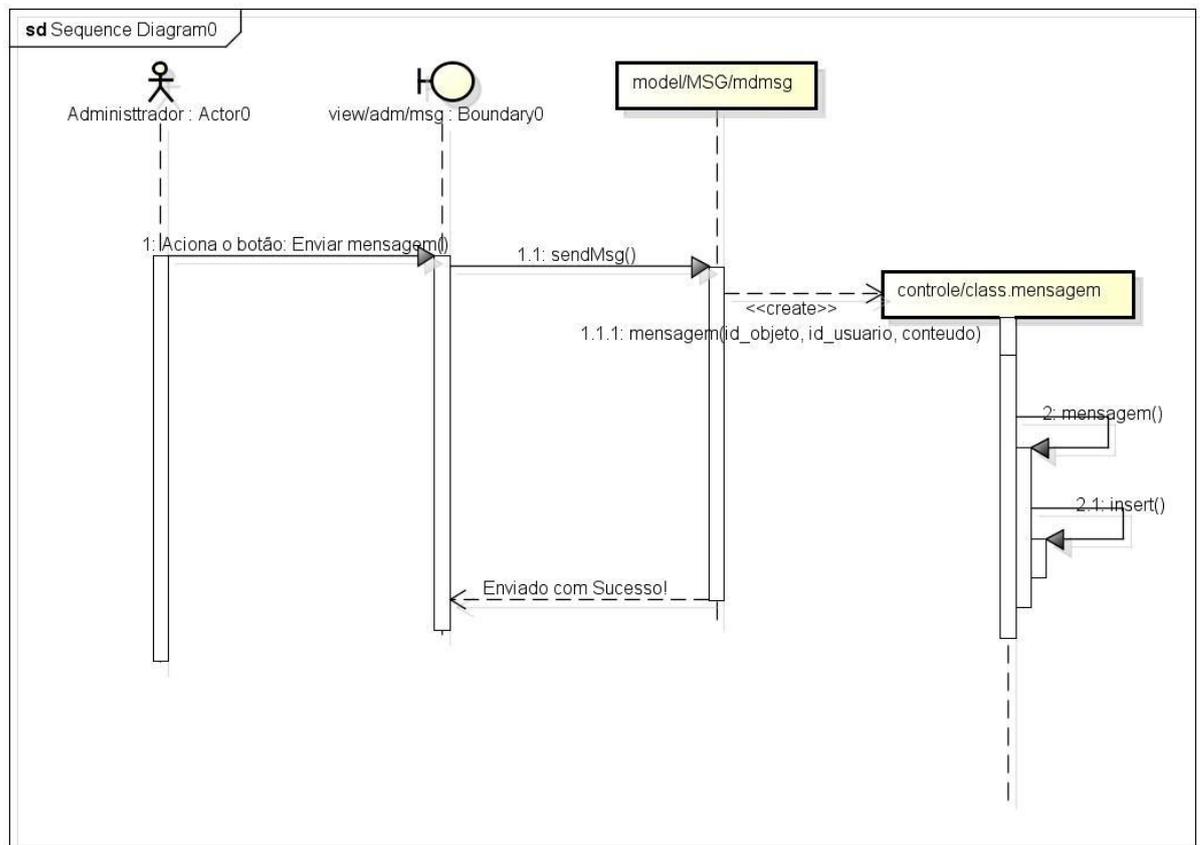
Figura 7 - Diagrama de Classes do pacote Controler.

A classe *Database* armazena os dados para conexão com o banco de dados, persistidos às outras classes que realizam alterações no banco de dados. As demais classes do pacote *Controler* cria um objeto responsável por sua manipulação. Os métodos destas classes modelam o banco de dados e o comportamento de cada objeto a partir das regras de negócio. Estes métodos se preocupam apenas com o armazenamento, manipulação e geração de dados no banco, combinando os dados recebidos com os métodos que os manipulam.

As coleções de dados `ArrayList` e `Array` foram utilizadas de modo que, a cada vez que o usuário faz uma requisição de listar objetos ou usuários, é retornada uma lista com dados de diversos tipos e tamanhos. Tais coleções foram utilizadas para que os dados fossem retornados de forma mais rápida e eficiente.

#### 4.4 DIAGRAMA DE SEQUÊNCIA

O Diagrama de sequência `Enviar mensagem`, apresentado na Figura 8, apresenta a comunicação entre as classes, permite que o administrador envie uma mensagem a um usuário.



powered by Astah

**Figura 8 - Diagrama de sequência Enviar mensagem.**

Observa-se na Figura os passos do usuário administrador ao enviar a mensagem a um usuário comum. Primeiro o usuário clica no botão `Enviar mensagem`, que aciona a função `sendMsg`, na classe `msg` do pacote `View`. Esta função envia os dados para a classe `mdmsg` do pacote `Model`, responsável por

receber os dados e determinar o fluxo. Deste modo, ela envia os dados para a classe `class.mensagem` no pacote `Controler` que, por sua vez, instancia uma nova mensagem e a insere no banco de dados. Por fim, a classe de interface recebe os dados que as informações foram adicionadas no banco de dados e confirma ao usuário.

#### 4.5 MODELO ENTIDADE-RELACIONAMENTO

A Figura 9 mostra a modelagem do banco de dados, suas tabelas e atributos.

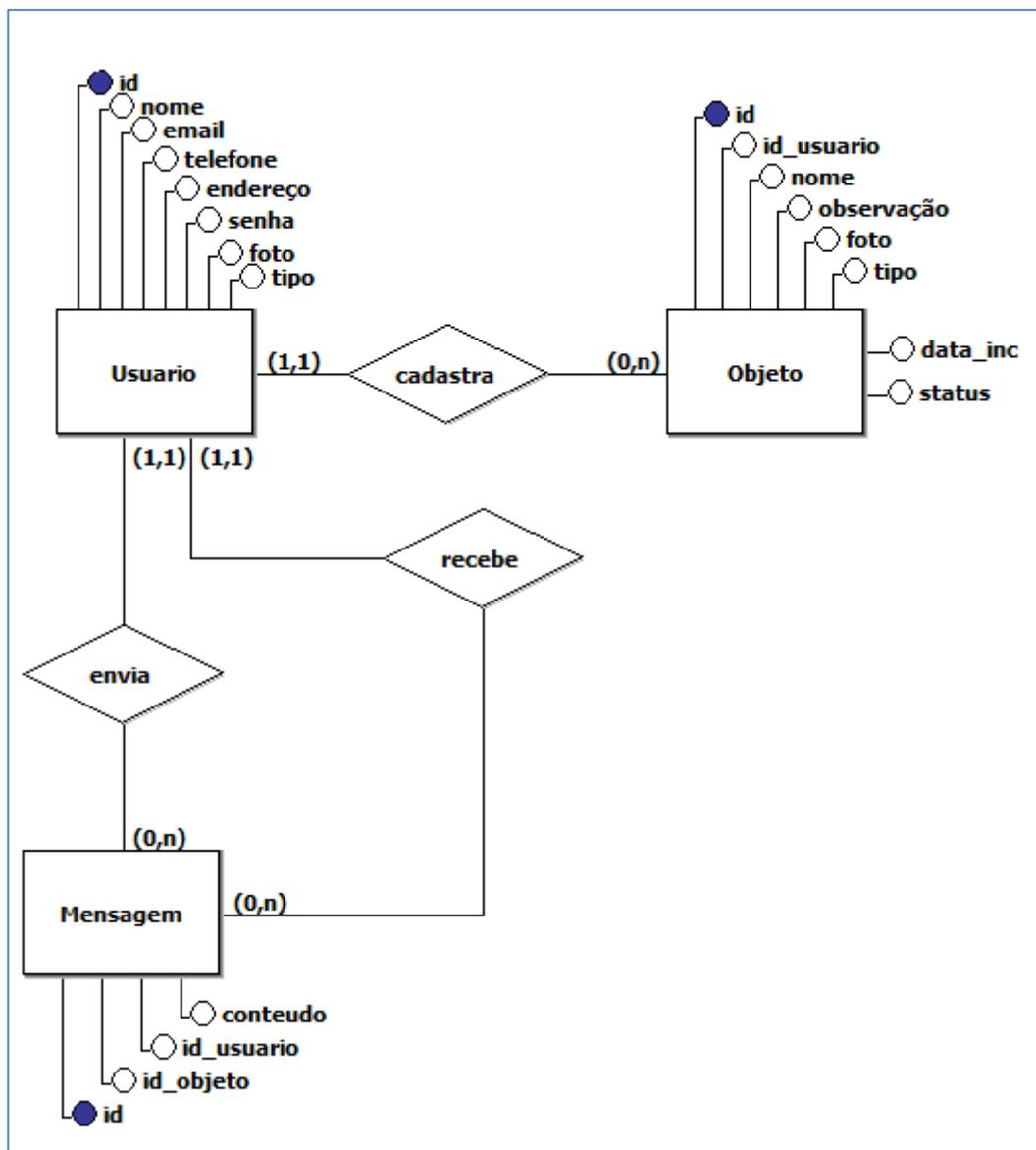


Figura 9 - Modelo Entidade Relacionamento (MER)

O modelo de entidade-relacionamento representa o banco de dados utilizado no sistema no seu modelo conceitual. Os atributos coloridos em azul representam chaves primárias. Cada objeto pode ser cadastrado apenas por uma pessoa, assim como uma mensagem. Um usuário pode criar quantos objetos e mensagem quiser.

O modelo foi criado com base dos conceitos da notação de Peter Chen. Segundo Nogueira (2015) este modelo traz uma modelagem simples da diagramação e dos conceitos que o envolvem onde: os retângulos indicam as entidades que armazenam os dados, os losangos representam os relacionamentos entre estas entidades e os círculos indicam os atributos. Na figura 9 verifica-se que existem 3 entidades com seus respectivos atributos. Pelos losangos conclui-se que estas entidades estão relacionadas, inclusive um relacionamento entre a própria entidade em que pode-se perceber que o Usuário envia a mensagem e ele também pode ser um receptor da entidade mensagem.

## 5 RESULTADOS OBTIDOS

Este capítulo apresenta os resultados obtidos do desenvolvimento do Sistema Gerenciador de Achados e Perdidos.

A Figura 10 apresenta a tela inicial do sistema, onde se encontra os campos para *login*, cadastro de usuário e busca de objetos.



Figura 10 - Tela Inicial.

Caso o usuário resolva fazer uma pesquisa, logado ou não. O resultado aparece como o mostrado na Figura 11.

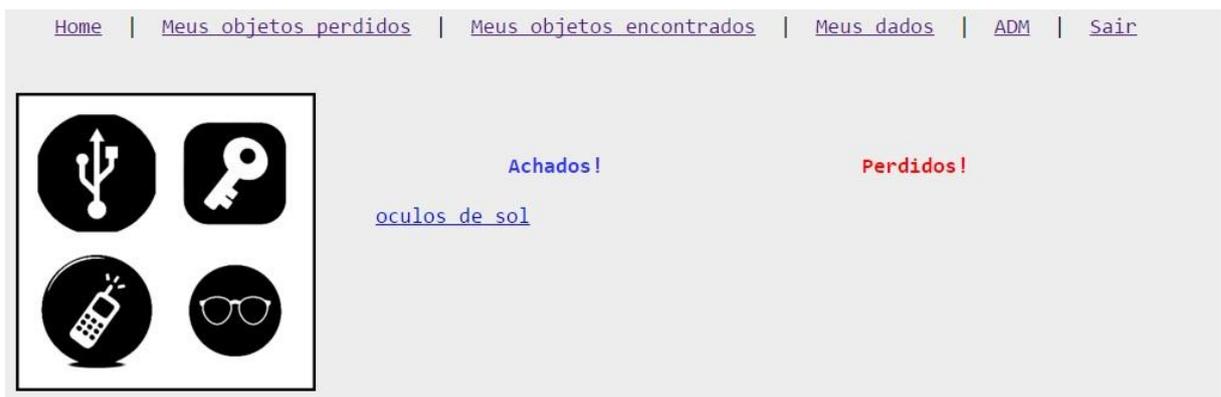


Figura 11 - Tela pós busca

Após realizada a busca, o usuário pode selecionar o objeto desejado e visualizará a imagem da Figura 12, na qual é possível enviar uma mensagem instantânea ao administrador, que posteriormente entrará em contato com o mesmo.



**Figura 12 - Tela de um objeto Achado.**

Caso o usuário não estiver logado no sistema, ele é direcionado para a tela de login, mostrada na Figura 13.



**Figura 13 - Tela de login.**

A tela apresentada na Figura 14 é a tela pós *login* como administrador. Nesta tela é possível visualizar os objetos cadastrados pelo usuário, botões para cadastro de novos objetos e uma aba de apoio, com algumas opções, como por exemplo, a opção ADM que direciona para as funcionalidades exclusivas do administrador.

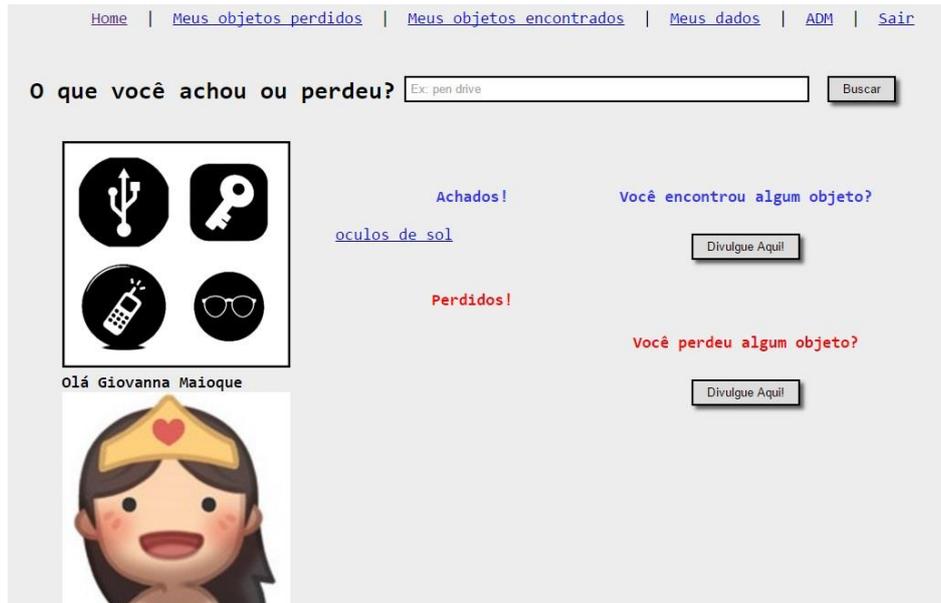


Figura 14 - Tela pós login.

A Figura 15 apresenta na tela as opções mostradas ao administrador, após selecionar a aba ADM. Nela pode ser visualizada as opções Usuários, Objetos, Mensagens, direcionadas às telas de manipulação dos dados e realizar buscas no banco de dados por meio das opções: Relatório objetos e Relatório Usuários.



Figura 15 - Tela da aba ADM.

A Figura 16 representa a tela para realizar buscas de usuário por meio das opções nome ou email. Logo abaixo é apresentado um exemplo de como a busca é visualizada pelo usuário após a pesquisa.

**Figura 16 - Tela de busca de usuários.**

A Figura 17 mostra a tela de Busca de objetos, exclusiva do usuário administrador. Esta busca pode ser realizada por meio dos atributos: nome, número de registro, tipo, status. Também é possível selecionar um intervalo de tempo para realizar as buscas, que encontra todos os objetos registrados no sistema neste intervalo.

**Figura 17 - Tela de busca de objetos**

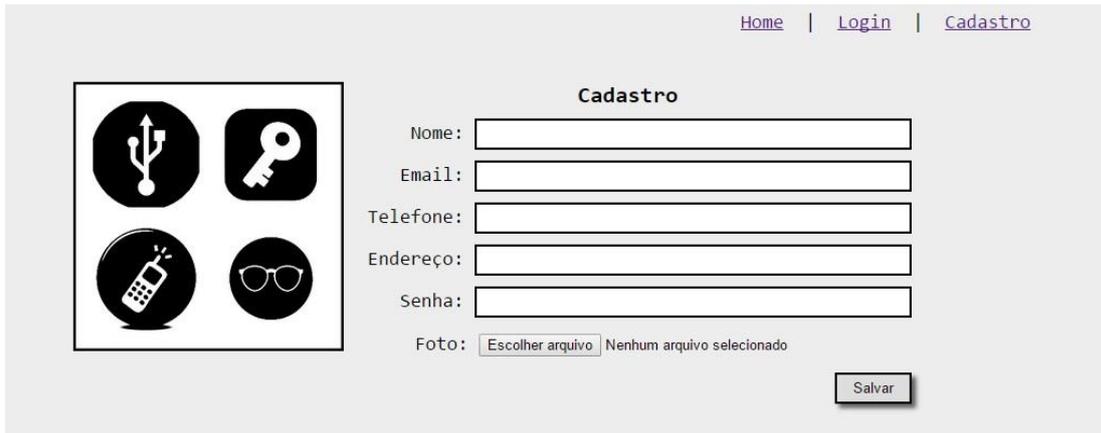
A tela da Figura 18 apresenta a tela para o administrador manipular dados do objeto. Nesta tela o usuário pode alterar qualquer dado do objeto, inclusive, seu estado, que é uma exclusividade do administrador. Os estágios variam entre *ON* (objeto disponível para pesquisa no sistema), *OFF* (objeto devolvido ao dono) e *ESQUECIDO* (objeto há mais de 90 dias no sistema e pode ser doado).

**Figura 18 - Tela administração de objeto.**

A tela da Figura 19 apresenta a tela onde, após o administrador escolher Mensagens, apresentada na Figura 15, aparece uma lista com várias mensagens e quando o administrador escolhe uma, aparece a tela a seguir. O administrador pode responder a mensagem enviada pelo usuário.

**Figura 19 - Tela de envio de mensagem.**

A Figura 20 apresenta a tela de cadastro de usuário, onde são digitados os dados do usuário e permite escolher a foto para utilizar no perfil. O *email* do usuário será utilizado como *login*. O usuário quando é criado, ele é instanciado como usuário comum. O administrador é o único que pode alterar tal configuração



Home | Login | Cadastro

### Cadastro

Nome:

Email:

Telefone:

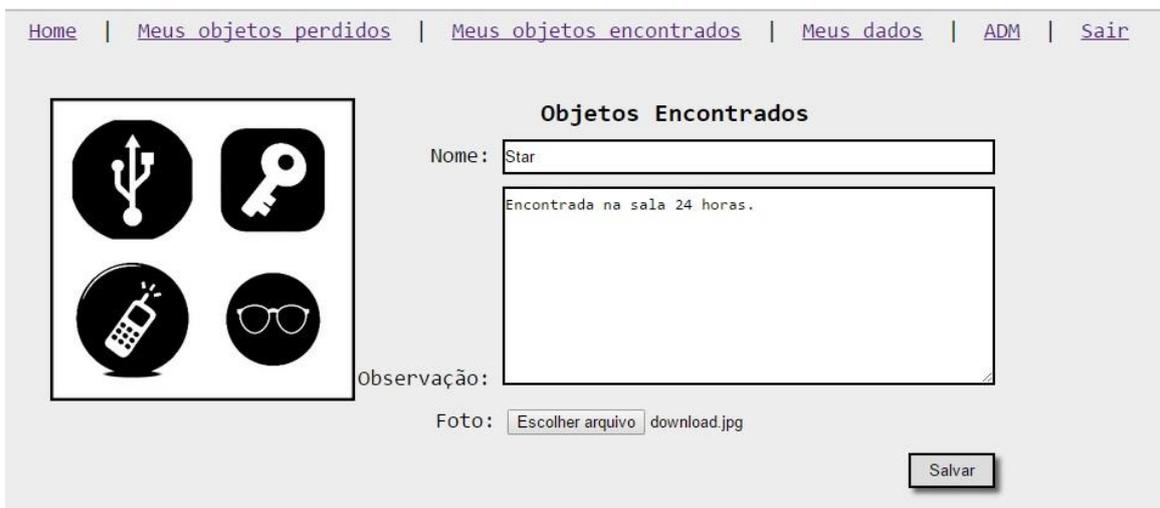
Endereço:

Senha:

Foto:  Nenhum arquivo selecionado

Figura 20 - Tela de cadastro de usuários.

A Figura 21 mostra a tela de cadastro de um novo objeto, onde é possível descrever todas as características do objeto, onde foi encontrado, para que outro usuário possa identificá-lo da maneira mais eficiente.



Home | Meus objetos perdidos | Meus objetos encontrados | Meus dados | ADM | Sair

### Objetos Encontrados

Nome:

Observação:

Foto:  download.jpg

Figura 21 - Tela de cadastro de objeto.

## 6 CRONOGRAMA REALIZADO

<i>Sprints</i>	<b>FUNCIONALIDADES</b>	<b>Tempo Necessário</b>	<b>Início</b>	<b>Término</b>
	<b>Definição e Estudo dos Requisitos Necessários</b>	<b>8 dias</b>	<b>19/03/2014</b>	<b>26/03/2014</b>
1ª	Estudo dos Requisitos	2 dias	19/03/2014	20/03/2014
2ª	Definição dos Requisitos Funcionais	5 dias	21/03/2014	25/03/2014
3ª	Definição dos Requisitos Não Funcionais	1 dia	26/03/2014	26/03/2014
	<b>Modelagem dos Principais Diagramas</b>	<b>11 dias</b>	<b>27/03/2014</b>	<b>06/04/2014</b>
4ª	Modelagem do Diagrama de Caso de Uso	2 dias	27/03/2014	28/03/2014
5ª	Modelagem do Diagrama de Classes	4 dias	29/03/2014	01/04/2014
6ª	Modelagem do Diagrama de Entidade-Relacionamento	3 dias	02/04/2014	04/04/2014
7ª	Modelagem do Diagrama de Sequência	1 dia	05/04/2014	05/04/2014
8ª	Modelagem do Diagrama de Pacotes	1 dia	06/04/2014	06/04/2014
	<b>Escrita da Proposta</b>	<b>62 dias</b>	<b>10/04/2014</b>	<b>01/07/2014</b>
9ª	Fundamentação Teórica	10 dias	10/04/2014	19/04/2014
10ª	Metodologia	10 dias	20/04/2014	29/04/2014
11ª	Protótipos de tela e estruturação da interface gráfica	12 dias	10/05/2014	22/05/2014
12ª	Organização do documento da Proposta	30 dias	01/06/2014	01/07/2014
	<b>Estudo de Programação para Web</b>	<b>150 dias</b>	<b>01/07/2014</b>	<b>01/12/2014</b>
13ª	Programação php	60 dias	01/07/2014	01/09/2014
14ª	HTML e css	60 dias	01/09/2014	01/11/2014
15ª	Dúvidas em geral	30 dias	01/11/2014	01/12/2014
	<b>Defesa da Proposta</b>	<b>1 dia</b>	<b>14/07/2014</b>	<b>14/07/2014</b>
	<b>Implementação do Sistema</b>	<b>150 dias</b>	<b>15/01/2015</b>	<b>15/10/2015</b>
16ª	Implementação do Banco de Dados	30 dias	15/01/2015	15/02/2015
17ª	Implementação das Classes de Modelagem do Sistema	30 dias	16/02/2015	18/03/2015
18ª	Implementação das Classes de Acesso ao Banco de Dados	30 dias	18/03/2015	18/04/2015
19ª	Desenho das Views	10 dias	19/05/2015	29/06/2015
20ª	Implementação das Views	10 dias	30/07/2015	09/08/2015
21ª	Implementação das Funcionalidades	20 dias	09/08/2015	29/08/2015
22ª	Ajustes de implementação em geral	20 dias	30/08/2015	10/09/2015
	<b>Testes</b>	<b>15 dias</b>	<b>15/09/2015</b>	<b>30/09/2015</b>
23ª	Revisão e teste geral do Sistema	10 dias	15/09/2015	25/09/2015
24ª	Correção de Bugs	5 dias	25/09/2015	30/09/2015
	<b>Escrita da Monografia</b>	<b>10 dias</b>	<b>10/09/2015</b>	<b>30/09/2015</b>
25ª	Escrita da Monografia	20 dias	10/09/2014	30/09/2015
	<b>Defesa do Trabalho de Conclusão</b>	<b>1 dia</b>	<b>01/11/2015</b>	<b>01/11/2015</b>

## 7 CONSIDERAÇÕES FINAIS

Desenvolver este sistema foi extremamente desafiador e gratificante. O uso de uma linguagem de programação desconhecida tomou grande parte do cronograma do projeto, obrigando o desenvolvedor a adquirir uma experiência que suprisse suas necessidades.

O trabalho desenvolvido atinge satisfatoriamente os objetivos previstos, permitindo ao setor de Achados e Perdidos da Universidade Tecnológica Federal do Paraná controle e agilidade durante os processos de busca e recuperação de objetos. O sistema possui como diferencial em comparação aos demais produtos existentes por possuir a funcionalidade de alterar o status de um objeto automaticamente após 90 dias cadastrado no banco de dados, permitindo assim que o setor doe os objetos.

O sistema desenvolvido não possui um servidor *web*, executa de forma local. Apenas o administrador possui as funcionalidades de alterar, excluir e modificar o estado dos objetos bem como excluir usuários. O gerenciador não permite manter conversas entre os usuários. Tais características podem ser notadas como limitações do *software*.

Em trabalhos futuros podem ser previstas modificações para melhoria do Sistema Gerenciador de Achados e Perdidos como: permitir que usuário realize *login* por meio de redes sociais (Twitter, Facebook, Google+) e o desenvolvimento do sistema em um aplicativo para *smartphone*.

## REFERÊNCIAS

AEROPORTO DE GUARULHOS. Disponível em: <<http://www.aerportoguarulhos.net/dicas-de-viagem/saiba-mais-sobre-os-achados-e-perdidos-nos-aeroportos-da-infraero>>. Acesso dia 21 nov. 2015.

ALAG, Satnam. **Collective Intelligence in Action**. 1. ed. New York: Manning Publications, 2009. 425 p.

ALMEIDA, Hélder Nunes de. **LogicSort** : Um objeto de aprendizagem para o ensino de algoritmos de ordenação. 2013. Trabalho de Conclusão de Curso (Graduação em Computação) – Universidade Estadual da Paraíba, Centro de Ciências e Tecnologias, Campina Grande, 2013. Disponível em: <<http://dspace.bc.uepb.edu.br:8080/xmlui/bitstream/handle/123456789/2350/PDF%20-%20H%C3%A9lder%20Nunes%20de%20Almeida.pdf?sequence=1>>. Acesso em: 13.05.2014

ALMEIDA, Rodrigo Rebouças de. **Model-View-Controller (MVC)**. Disponível em:<<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>>. Acesso em 15 mai. 2014.

BRAMBILA, Marcelo K.; MAHLMANN, Luiz Gustavo G. **Sistema de Gerenciamento e Controle de Documentos de TCC e Estágio**. 10p. Trabalho de Conclusão de Curso (Graduação) – Curso superior de Sistemas de Informação. Universidade Luterana do Brasil, Guaíba. Disponível em: <<http://guaiba.ulbra.br/seminario/eventos/2006/artigos/sistemas/133.pdf>>. Acesso em: 17 mai. 2014.

CARNEIRO, Alessandro Pinto; MOREIRA, Julinao Lucas; DE FREITAS, André Luis Castro. **Tuning – Técnicas de Otimização de Banco de Dados. Um estudo comparativo: Mysql e Postgresql**. 10p. Trabalho de Graduação. Centro de Ciências Computacionais. Universidade do Rio Grande (FURG), Rio Grande, RS. Disponível em: <<http://repositorio.furg.br/bitstream/handle/1/1692/TUNING.pdf?sequence=1>>. Acesso em 21 nov. 2015.

CORREIOS. **Achados e perdidos**. Disponível em: <<http://www.correios.com.br/para-voce/correios-de-a-a-z/achados-e-perdidos>>. Acesso em: 19 mai. 2014.

FABRI, José Augusto. **Scrum Solo**. Disponível em: <<http://engenhariasoftware.wordpress.com/2012/06/06/scrum-solo/>>. Acesso em: 2 mai. 2014.

FACCAMP. Disponível em: <<http://www.faccamp.br/manualAluno.html>>. Acesso em: 16 mai. 2014.

FERREIRA, Décio; COSTA, Felipe; ALONSO, Filipe; ALVES, Pedro; NUNES, Tiago. **SCRUM: Um Modelo Ágil para Gestão de Projetos de Software**. Disponível em: <[http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es\\_final\\_19.pdf](http://paginas.fe.up.pt/~aaguiar/es/artigos%20finais/es_final_19.pdf)>. Acesso em: 2 mai. 2014.

GUEDES, Gilleanes T. A.; **UML: Uma abordagem prática**. 2 ed., São Paulo: NovaTec, 2005.

JUNIOR, Edson Alves de Oliveira; FORTES, Renata Pontin de Mattos. 2007. 32 p. **Arquitetura de Software na Web Atual: Processamento no Servidor**. Artigo. Universidade de São Paulo. Instituto de Ciências Matemáticas e de Computação, São Carlos, 2007.

LUCKOW, Décio Heizelmann; MELO, Alexandre Altair de. **Programação Java para web**. São Paulo: Novatec Editora, 2010. Disponível em: <<http://editoranovatec.com.br/livros/javaparaweb/capitulo9788575222386.pdf>>. Acesso em: 15 mai. 2014.

MEDEIROS, Levindo C.; OLIVEIRA, Luciano C. S. de.; SILVA, Márcio M.; **Sistema de disponibilização de informações geográficas do estado de Goiás na internet (SIG OnLine)**. 2005. 45 p. Trabalho de Conclusão de Curso (Graduação) – Curso de Tecnologia em Geoprocessamento. Centro Federal de Educação Tecnológica do Estado de Goiás, Goiânia, 2005. Disponível em: <[http://www.seplan.go.gov.br/sieg/downloads/SIG\\_ONLINE.pdf](http://www.seplan.go.gov.br/sieg/downloads/SIG_ONLINE.pdf)>. Acesso em 17 mai. 2014.

NIEDERAUER, Juliano. **Desenvolvendo websites em PHP**. 2. ed. São Paulo: Novatec Editora. 2011. Disponível em: <<http://www.novatec.net.br/livros/php2ed/capitulo9788575222348.pdf>>. Acesso em 15 mai. 2014.

NOGUEIRA, Marcelo. **MER – Modelo entidade relacionamento**. 6p. UNIP – Ciência da Computação. Disponível em: <[http://www.noginfo.com.br/arquivos/SI\\_AAES\\_Parte\\_03\\_MER.pdf](http://www.noginfo.com.br/arquivos/SI_AAES_Parte_03_MER.pdf)>. Acesso em 22 nov. 2015.

NUNES, Otávio. **Achados e perdidos da CPTM recebe 5 mil documentos e objetos por mês**. Disponível em: <<http://portalgeral.com.br/noticias-brasil/achados->

[e-perdidos-da-cptm-recebe-5-mil-documentos-e-objetos-por-mes/](#)>. Acesso em: 16 mai. 2014.

OLIVEIRO, Carlos Antônio José. **Faça um site: PHP4 com Base de Dados MySQL Orientado por Projeto**. São Paulo: Editora Érica, 2001.

SAYEG, Norton. Disponível em: <<http://www.alzheimermed.com.br/perguntas-e-respostas/esquecer-onde-coloquei-as-chaves-oculos-e-um-processo-natural-do-envelhecimento-ou-da-doenca-de-alzh>>. Acesso em: 19 mai. 2014.

SILVA, Diego. **Achados e perdidos da CPTM ganham sistema online**. Disponível em: <<http://cptmemfoco.blogspot.com.br/2012/08/achados-e-perdidos-da-cptm-ganha.html>>. Acesso em: 14 mai. 2014.

SILVA, Everton Topan da. **ACHUSP: Serviço de achados e perdidos para a Universidade de São Paulo**. 2013. 29 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior em Ciências da Computação. Universidade de São Paulo, Instituto de Matemática e Estatística, São Paulo, 2013. Disponível em: <<http://bcc.ime.usp.br/principal/tccs/2012/rec/everton/monografia.pdf>>. Acesso em: 14 mai. 2014.

SOARES, Jonathan. **Guideline de construção de jogos educacionais para web usando HTML5**. 2013. 81 p. Monografia (Graduação) – Curso Superior em Ciência da Computação. Universidade Federal de Pernambuco, Recife, 2013. Disponível em: <<http://www.cin.ufpe.br/~tg/2013-1/js2.pdf>>. Acesso em 17 mai. 2014.

TAVARES, Jânio. **ACHADOS E PERDIDOS - Objetos perdidos se acumulam na Infraero**. Disponível em: <<http://www.folhabv.com.br/noticia.php?id=48726>>. Acesso em: 16 mai. 2014.

TILE. Disponível em: <<http://www.thetileapp.com/#our-app>>. Acesso em: 19 mai. 2014.

VAZ, Noeli A. P.; FERREIRA, Fernando B. **Metodologias ágeis – Scrum**. 2 f. Artigo – III Congresso Acadêmico Científico. Unidade Universitária da Universidade Estadual de Goiás, Porangatu, Goiás, 2013.

WELLING, Luke; THOMSON, Laura. **PHP e MySQL – Desenvolvimento web**. 2002. 676 p. Rio de Janeiro: Campus, 2003.

ZANINELLI, Thais Batista; CATARINO, Maria Elisabete. **Recuperação de informação no banco de dados da Embrapa Soja: estudo do Anfo**. Inf., Londrina. [S.l.], v. 9, n. 1-2, p. 48-66, dez. 2004. ISSN 1981-8920. Disponível em:

<http://www.uel.br/revistas/uel/index.php/informacao/article/view/1726/1477>.  
Acesso em: 21 Nov. 2015.