

1 INTRODUÇÃO

Devido a quantidade de redes sociais existentes atualmente, é difícil para uma equipe de marketing gerenciar diversas campanhas e agrupar essas informações de forma a conseguir analisar o impacto de como a empresa esta sendo vista na rede. A proposta deste trabalho é o desenvolvimento de uma aplicação que irá integrar algumas funcionalidades de CRM com a captura desses contatos, baseados em regra de cada campanha nas redes sociais. Com a integração desses dados captados em diferentes tipos de rede será possível apresentar informações estatísticas para equipe de marketing ou comercial da empresa com a relevância de suas campanhas por meio de análise de tagueamento, análise de sentimento, trending topics, engajamento, tendências e ROI.

Desenvolver um software usando exclusivamente tecnologias baseadas em software livre para unificar a comunicação entre uma empresa e todos os seus “seguidores” em suas contas nas redes sociais e medir sua popularidade por meio de campanhas de marketing, disponibilizar dados analíticos provenientes dessas redes e gerenciar campanhas.

O IBRAMERC (Instituto Brasileiro de Inteligência de Mercado) realizou uma pesquisa sobre a presença e uso das redes sociais com 251 empresas de médio e grande porte e constatou: 65% destas empresas já utilizam as redes sociais como ferramenta de comunicação com clientes e prospects. A onda do momento é o Twitter segundo Richard Lowenthal, diretor executivo do IBRAMERC, diferente de outras, estas ferramenta demanda menos esforço de atualização (PROSPECÇÃO...2011).

A figura a seguir demonstra em forma de gráfico o resultado da pesquisa do IBRAMERC com dados das redes sociais mais relevantes para as empresas atualmente.

FIGURA 1 – PRESENÇA NAS REDES SOCIAIS



FONTE: IBRAMERC (2011).

Mesmo com uma alta adesão no mundo corporativo, as redes sociais ainda são vistas como um canal complementar ou um meio de comunicação muito importante, mas não fundamental. Isso porque a utilização destas ferramentas ainda está em fase de testes e as empresas buscam entender como isso realmente pode agregar valor ao negócio(*id.*,2011).

Algumas dessas empresas já utilizam ferramentas automatizadas de análise e gerenciamento de seus contatos nas redes, entre elas estão o TweetDeck, ScoutLabs, HootSuite, Google Analytics e Google Trends. Mas nenhuma dessas ferramentas realiza em conjunto as rotinas de CRM e análise.

A figura a seguir apresenta mais dados da pesquisa da ABRAMERC sobre como as empresas usam as redes sociais.

FIGURA 2 – UTILIZAÇÃO DAS REDES SOCIAIS POR EMPRESAS



FONTE: IBRAMERC (2011).

2 PLANO DE PROJETO

2.1 OBJETIVOS DO PLANO DE PROJETO

Os objetivos desta sessão é determinar algumas condições para que o projeto seja iniciado, desenvolvido e completado. Esta sessão descreve o projeto de forma a quantificar o tempo, o esforço e o orçamento que o mesmo estará submetido.

2.2 VISÃO GERAL DO PROJETO

Esta sessão está dividida entre objetivo do projeto, custo benefício, o qual define todos os custos do projeto, sejam eles custos de desenvolvimento, custos de implantação, e preço do projeto, define também a organização do projeto, ou seja, a definição dos papéis dos envolvidos no projeto e os processos de gerenciamento que serão utilizados, os quais estão sub-divididos em gerenciamento de escopo, tempo e risco.

2.2.1 Objetivo do Projeto

O objetivo do projeto é desenvolver um aplicativo web para facilitar e unificar a interação com contatos de diversas redes sociais, além de gerenciar esses contatos, prospectando para possíveis negócios e/ou suporte.

2.2.2 Custo Benefício

2.2.2.1 Custo de desenvolvimento do projeto

Com relação aos softwares e ferramentas de desenvolvimento, como premissa serão utilizadas somente soluções de software livre, sendo assim, tanto as ferramentas de desenvolvimento, quanto as tecnologias de software que serão utilizadas no projeto não possuem nenhum custo.

O hardware utilizado para desenvolver a aplicação, bem como realizar a hospedagem será hardware pertencente aos membros do projeto, no total serão usados 2 servidores.

2.2.2.2 Custo de implantação do produto

O projeto será desenvolvido usando o conceito de aplicação na nuvem, podendo ser escalável sob demanda. O acesso a aplicação se dará a partir de dispositivos com conexão à internet, eliminando custos com infraestrutura para o cliente.

Não haverá processo de implantação, por ser um sistema multiusuário na nuvem, basta realizar o cadastro no site, e iniciar o uso.

Por se tratar de uma solução livre de código aberto, o sistema não possui limitações de funcionalidades, mas cada rede social define um limite de consultas para o uso gratuito da mesma.

2.2.2.3 Preço do projeto

O preço total do projeto é de R\$13.240,00 sendo este valor dividido pelos itens abaixo:

QUADRO 1 – PREÇO DO PROJETO

Tipo	Indicador	Valor
Recursos humanos	200 * R\$30,00 (Horas trabalhadas * Valor da Hora)	R\$6.000,00
2 Servidores	R\$2.000 p/ servidor	R\$4.000,00
Banda Larga	R\$100 p/mês cada servidor	R\$1.800,00
Energia elétrica	R\$80 p/mês cada servidor	R\$1.440,00

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

2.2.2.4 Custo benefício para o cliente

O usuário terá em mãos uma ferramenta integrada de relacionamento com seus contatos, e prospecção de *leads* de forma automatizada, sem limitações de funcionalidades com grandes chances de expandir sua rede de negócios e/ou relacionamento de forma eficiente e automatizada.

2.3 ORGANIZAÇÃO DO PROJETO

2.3.1 Estrutura Organizacional do Projeto

QUADRO 2 – ESTRUTURA ORGANIZACIONAL DO PROJETO

Envolvido	Papel
Evandro Klimpel Balmant	Levantamento de requisitos, implementação de conexão com as API's das redes sociais.
Wiliam Fernando Stancke	Levantamento de requisitos, modelagem de banco de dados, definição da arquitetura do sistema.

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

2.4 PROCESSOS DE GERENCIAMENTO DO PROJETO

2.4.1 Gerenciamento do Escopo

Com relação ao gerenciamento do escopo do projeto, inicialmente serão levantados requisitos com base em experiência pessoal com redes sociais, e posteriormente levantados com um especialista em marketing, sendo este, responsável pelo aceite dos requisitos, validando as estratégias de marketing do produto desenvolvido.

O tratamento a mudança de requisitos ou mesmo a inclusão de novos requisitos será tratado de forma a agrupar conjuntos de requisitos e desenvolvidos em novas iterações.

2.4.2 Gerenciamento do Tempo

É utilizado o processo de engenharia de software *Iconix*, e devido a opção por esta metodologia o projeto será dividido em 4 iterações, e cada iteração possuindo as disciplinas de requisitos, análise/desenho preliminar, desenho detalhado e implementação.

Inicialmente está previsto 4 iterações para o desenvolvimento do projeto, entretanto, caso surjam novos requisitos durante o desenvolvimento, os mesmos serão agrupados e desenvolvidos em novas iterações.

2.4.2.1 Cronograma

O quadro a seguir mostra de forma geral todas as iterações, nome da iteração, horas trabalhadas para cada iteração, duração em dias, data de início, data de término e recursos humanos alocados para as referidas atividades.

QUADRO 3 – CRONOGRAMA DO PROJETO

Iteração	Horas	Duração	Início	Término	Recursos Humanos
1	30h	30 dias	05/03/2011	10/05/2011	Todos do projeto
2	50h	50 dias	10/05/2011	15/07/2011	Todos do projeto
3	100h	115 dias	16/07/2001	30/10/2011	Todos do projeto
4	20h	30 dias	01/11/2011	25/11/2011	Todos do projeto

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

2.4.3 Gerenciamento de Riscos

Segue relação dos riscos identificados que podem levar ao insucesso do projeto. Para cada risco será apresentada uma ação para mitigar ou eliminar o respectivo risco, bem como o responsável pela execução desta ação.

QUADRO 4 – GERENCIAMENTO DE RISCOS

Risco	Ação	Responsável
Desconhecimento das tecnologias utilizadas.	Estudar as tecnologias	Todos os envolvidos no projeto
APIs das redes sociais não fornecerem dados suficientes para a implementação.	Optar por outras APIs e/ou eliminar a escolha por determinada API.	Evandro Klimpel Balmant
Concorrência	Realizar estudo do modelo de negócios da concorrência.	Wilian Fernando Stancke

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

3 DESCRIÇÃO DO PROJETO

3.1 OBJETIVOS DA DESCRIÇÃO DO PROJETO

Esta sessão descreve quais os problemas identificados no cenário atual e as formas de solução destes problemas identificando as partes interessadas e afetadas, descrevendo também os requisitos funcionais e não-funcionais do projeto.

3.2 DESCRIÇÃO DO SISTEMA ATUAL

Não existe sistema atual que integre todas as redes sociais de forma unificada, tendo o cliente que conectar em cada uma delas em separado para visualizar as interações com o cliente, possuindo cada sistema uma forma diferente de relacionamento com o usuário.

3.2.1 Descrição dos problemas

O quadro a seguir descreve cada problema identificado informado o problema, quem é o afetado, qual é o impacto do problema e qual será a solução adotada.

QUADRO 5 – DESCRIÇÃO DOS PROBLEMAS

Problema	Afetado	Impacto	Solução
Gerenciamento de múltiplas contas em redes sociais.	Utilizadores de redes sociais de forma geral.	Perda de produtividade e falta de integração.	Desenvolvimento de aplicação que conecta nas APIs da redes sociais informadas.
Perda de histórico de relações com o cliente.	Profissionais de marketing que lançam campanhas através das redes sociais.	Dificuldade em de argumentos negociação.	Desenvolvimento de interface de fácil acesso a estes históricos, independente da rede social.

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

3.2.2 Ambiente do usuário

O acesso à aplicação se dará a partir de qualquer dispositivo que possua acesso a internet através de um navegador web.

3.2.3 Resumo das partes interessadas

Esta sessão apresenta em forma de tabela a relação dos interessados no projeto, informando o seu nome, descrição e responsabilidades. Além dos próprios usuários das redes sociais haverá um especialista em marketing que irá fornecer informações necessárias para as estratégias de marketing utilizadas no sistema.

QUADRO 6 – PARTES INTERESSADAS

Nome	Descrição	Responsabilidades
Utilizadores	Utilizadores das redes sociais	Fornecer dados para análise do projeto.
Evandro Klimpel Balmant	Desenvolvedor de software	Desenvolver a solução
Wilian Fernando Stancke	Desenvolvedor de software	Desenvolver a solução

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

3.2.4 Resumo dos usuários

Profissionais do departamento de marketing de uma empresa ou qualquer utilizador de rede social que deseja avaliar a popularidade de sua campanha.

3.3 Descrição do sistema proposto

Sistema web onde o usuário tenha controle de suas contas em redes sociais de forma unificada, podendo criar campanhas e analisar sua performance de popularidade, publicar conteúdo em várias contas também de forma única, e manter históricos de relacionamento com os contatos.

3.4 REQUISITOS DO PRODUTO

3.4.1 Requisitos Funcionais

3.4.1.1 RF001 – Campanhas de marketing

Descrição: Usuário deverá ter a possibilidade de criar campanhas de marketing e consultar *leads* nas redes sociais.

Por quê?: Para criar parâmetros de pesquisa para fornecimento de dados estatísticos de análise de relacionamento.

Quem?: Profissionais de marketing

Quando?: Quando solicitado

Como?: Selecionando as redes vinculadas a campanha e definindo parâmetros da mesma.

Onde?: Através de módulo de campanhas

3.4.1.2 RF002 – Relatórios

Descrição: Sistema deve possuir diversos tipos de relatórios de prospecção de marketing e popularidade das campanhas de marketing.

Por quê?: Medir nível de popularidade e satisfação da rede de contatos.

Quem?: Analista de marketing

Quando?: Quando solicitado

Como?: Consultando o banco de dados de histórico de relacionamento com os contatos

Onde?: Através de módulo na aplicação

3.4.1.3 RF003 – CRM

Descrição: Sistema deverá manter históricos de relacionamento com cliente.

Por quê?: Para consultas posteriores.

Quem?: Banco de dados.

Quando?: A todo momento.

Como?: Através de banco de dados não-relacional.

Onde?: Em toda funcionalidade que exija persistência dos dados.

3.4.1.4 RF004 – Integração com as API's

Descrição: API's de redes sociais deverão se comunicar com sistema sem interferência direta dos usuários envolvidos, rotinas devem ser feitas de forma automatizada.

Por quê?: Automatizar o processo de publicação de conteúdo e comunicação.

Quem?: A camada de fronteira da aplicação

Quando?: Quando solicitado

Como?: Por meio de web-services ou requisições XML/JSON/HTTP/REST.

Onde?: Em todas as funcionalidades de comunicação

3.4.1.5 RF005 – Publicação de conteúdo em múltiplas contas

Descrição: Sistema deverá dispor de recursos de publicação de conteúdo em várias contas das redes sociais de forma facilidade e unificada.

Por quê?: Para agilizar o processos de divulgação das campanhas de marketing.

Quem?: Profissionais de marketing

Quando?: A qualquer momento que desejar.

Como?: Selecionando as redes sociais de destino informando a mensagem e publicando a campanha.

Onde?: Em módulo específico da funcionalidade.

3.4.2 Requisitos não funcionais

Esta sessão descreve os requisitos não-funcionais do projeto que estão classificados em portabilidade, manutenibilidade, eficiência, desempenho, usabilidade, produtividade, extensibilidade e escalabilidade. Para cada requisito é dado um identificador, qual o tipo do requisito e uma breve descrição.

3.4.2.1 RNF 001 – Ambiente Web

Tipo: Portabilidade

Descrição: Sistema deverá ser feito inteiramente para ambiente Web e rodar sem variação de funcionalidade em qualquer navegador recente.

3.4.1.2 RNF 002 – Transferência do Sistema

Tipo: Portabilidade

Descrição: Esforço exigido para transferir o sistema de um ambiente de hardware e / ou software para outro deverá ser mínimo ou inexistente.

3.4.1.3 RNF 003 – Linguagem de Programação

Tipo: Portabilidade

Descrição: Utilização de linguagem de programação Python pois a mesma é multiplataforma, podendo ser portada para qualquer plataforma sem mudanças no código.

3.4.1.4 RNF 004 – Especificação e Documentação

Tipo: Manutenibilidade

Descrição: Deverá ser feita documentação e especificação de requisitos dentro da norma estabelecida e padronizada (deverá ser utilizado padrão UML 2.0 para modelos e diagramas).

3.4.1.5 RNF 005 – Limpeza de código

Tipo: Manutenibilidade

Descrição: Codificação deverá respeitar convenções da linguagem Python definidas pela PEP8 (ROSSUM, 2011, p. 1) para manter a legibilidade do código e a padronização da linguagem.

3.4.1.6 RNF 006– Funcionamento

Tipo: Manutenibilidade

Descrição: Sistema deve manter o seu funcionamento parcial, ou seja, pelo menos 50% das operações devem estar funcionando dada a ocorrência de uma falha de hardware ou software.

3.4.1.7 RNF 007 – Recursos de computação

Tipo: Eficiência

Descrição: Quantidade de recursos de computação e de código exigido para que o programa execute a sua função deverá ser enxuta.

3.4.1.8 RNF 008 – Tempo de resposta

Tipo: Desempenho

Descrição: Velocidade de processamento e tempo de resposta tem de ser de no máximo 5 segundos entre requisição e resposta.

3.4.1.9 RNF 009 – Facilidade de uso

Tipo: Usabilidade

Descrição: Esforço físico e cognitivo do usuário durante o processo de interação deve ser curta (Tempo < 10min por caso de uso), sistema deve possuir recursos visuais bem posicionados, dicas e tooltips de ajuda.

3.4.1.10 RNF 010 – Produção

Tipo: Produtividade

Descrição: Sistema deverá impreterivelmente ser mais produtivo ao usuário do que seria se não o utilizasse, ou seja, deve ser feito comparativo de tempo entre operações semelhantes feitas com e sem o sistema e verificar se o tempo através do sistema é menor.

3.4.1.11 RNF 011 – Navegabilidade

Tipo: Produtividade

Descrição: Navegabilidade deverá ser rápida e facilitada, valendo-se de atalhos, caminhos alternativos, mapa de site entre outros recursos.

3.4.1.12 RNF 012 – Modularização

Tipo: Extensibilidade

Descrição: Software deve ser desenvolvido em formato modularizado para que posteriormente desenvolvedores acrescentem novas redes sociais ao sistema e/ou novas funcionalidades.

3.4.1.13 RNF 013 – Concorrência

Tipo: Escalabilidade

Descrição: Software deverá manter mesmo desempenho quando há um aumento no número de usuários e/ou de requisições simultâneas, para isto deve ser feito uso de clusterização, grid computacional ou outros mecanismos que assegurem o desempenho do sistema.

3.4.1.14 RNF 014 – Segurança dos dados

Tipo: Segurança

Descrição: Não poderá ser realizado acesso direto ao banco de dados por parte dos usuários, sendo necessário acessar os dados através do servidor de aplicação.

4 ARQUITETURA

4.1 OBJETIVOS DA ARQUITETURA

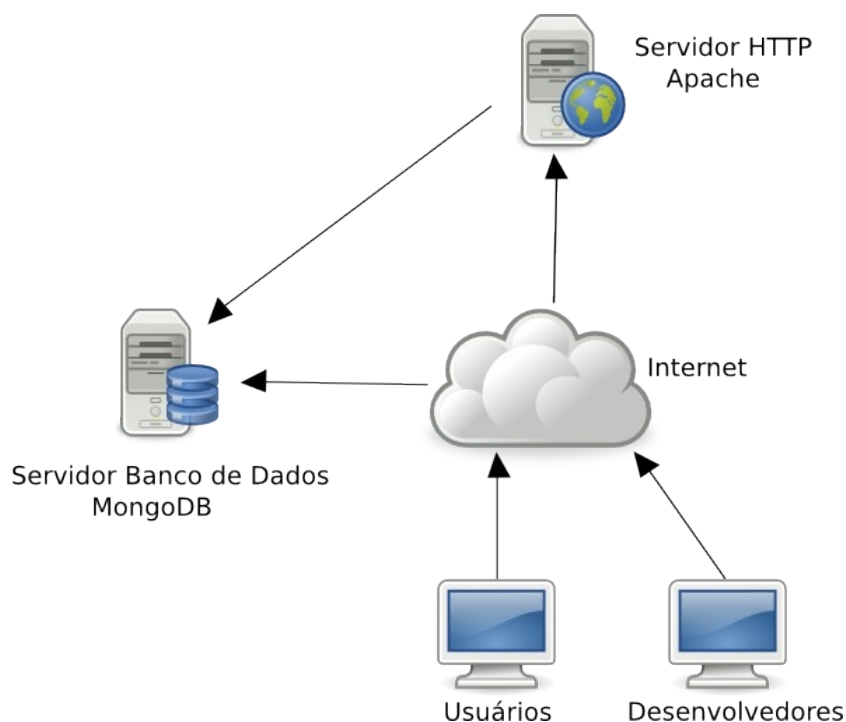
A arquitetura mostra as instalações que serão utilizadas no projeto. Será dividido em três seções, sendo elas, softwares de aplicação, visão física e visão lógica.

4.2 VISÃO FÍSICA

Nesta sessão são descritos os principais componentes do sistema juntamente com os detalhes de configuração utilizado nos equipamentos.

Segue diagrama de modelo físico do projeto:

FIGURA 3 – VISÃO DOS COMPONENTES DO SISTEMA



NOTA: Figura criada pela equipe de desenvolvimento deste projeto.

4.2.1 Descrição dos principais componentes

Servidor HTTP Apache: Servidor HTTP instalado em servidor GNU/Linux Debian 6 (Squeeze), será responsável por hospedar a aplicação e através dele será feito o acesso ao banco de dados para os *hosts* usuários.

Servidor Banco de Dados MongoDB: Servidor de banco de dados instalado em servidor FreeBSD, será responsável por persistir os dados do sistema, estará separado fisicamente do servidor de aplicação para oferecer maior segurança aos dados persistidos.

Usuários: *Hosts* responsáveis por executar a aplicação, os *hosts* usuários não terão acesso direto ao servidor de banco de dados.

Desenvolvedores: *Hosts* responsáveis por realizar o desenvolvimento do sistema, seu acesso poderá ocorrer para o servidor de aplicação ou para banco de dados sem limitação de acesso.

4.3 SOFTWARE DE APLICAÇÃO

Esta sessão descreve quais os softwares que serão utilizados para o desenvolvimento do sistema e sua função, bem como a justificativa para a escolha de sua utilização.

4.3.1 Ferramentas de modelagem

Astah Community: Sucessor do Jude, é uma ferramenta CASE multiplataforma, gratuita que auxilia na elaboração de diagramas da UML. Será utilizado para criação de modelos e diagramas da UML versão 2.0 (ASTAH , 2011). Foi optado pela utilização desta ferramenta de modelagem pois a mesma é distribuída de forma livre e gratuita e segue com rigor os modelos da UML propostos pela OMG (*Object Management Group*).

4.3.2 Ambientes de desenvolvimento integrado

Eclipse IDE: Ambiente de desenvolvimento integrado (IDE), gratuito e de código aberto, auxilia desenvolvedores a escrever, compilar, depurar e instalar aplicações (ABOUT...2011). Criado pela empresa IBM e doado para a comunidade de software livre distribuída através de licença EPL (Eclipse Public License), será utilizado para auxiliar no desenvolvimento dos códigos-fonte do sistema.

4.3.3 Editores de texto

VIm: Versão melhorada do editor VI, é um editor de texto baseado em console. Acompanha a maioria das distribuições linux é distribuído através de licença GPL (GNU *General Public License*), será utilizado para escrever códigos-fonte do sistema.

Cream: Editor de texto visual baseado no VIm (*Vi Improved*) , possui melhorias e customizações para auxiliar na criação de códigos-fonte em diversas linguagens. Distribuído através de licença GPL (GNU *General Public License*), será utilizado para escrever códigos-fonte do sistema.

4.3.4 Servidores de bancos de dados

MongoDB: Banco de dados não relacional (NoSQL), multiplataforma, de código aberto, de alta performance, sem esquemas, orientado à documentos, mantido pela empresa 10gen e distribuído sob licença GNU AGPL v3.0 (*Affero General Public License*) (10GEN , 2011), será utilizado para persistir os dados da aplicação. Foi optado pela utilização deste banco de dados pois além do mesmo contemplar em totalidade o requisito não-funcional RNF 014, possui vasta documentação e é utilizado por empresas consagradas no mercado, como The New York Times, CERN (*European Organization for nuclear research*), Chicago Tribune entre outros.

4.3.5 Servidores de aplicação

Apache HTTP Server: Servidor Web, multiplataforma, baseado em software livre, mantido pela Apache Software Foundation e distribuído sob licença AL (*Apache License*) (APACHE, 2011), será utilizado para servir a aplicação com o uso de módulos WSGI (*Web Server Gateway Interface*) para interpretar scripts Python, foi optado pela utilização deste servidor devido o mesmo já ser um servidor altamente utilizado no mercado e o mesmo possuir documentação abrangente.

4.4 VISÃO LÓGICA

Esta sessão descreve a visão lógica do sistema, ou seja, qual paradigma será utilizado no projeto, qual linguagem de programação, os

frameworks utilizados e os mecanismos arquiteturais para persistência, distribuição, segurança, entre outros.

4.4.1 Paradigma de programação

Será utilizado o paradigma da orientação a objetos para a modelagem e o desenvolvimento da aplicação pois a orientação a objetos reduz a complexidade através de melhoria do grau de abstração, melhoria da produtividade, maximizando a reutilização de código por parte dos desenvolvedores e melhor gerenciamento, através da quebra da aplicação em unidades menores e melhor definidas.

4.4.2 Linguagem de programação

Para o desenvolvimento do sistema foi utilizada a linguagem de programação Python, que é uma linguagem livre e multiplataforma, isso significa que os programas escritos em uma plataforma serão executados na maioria dos sistemas operacionais existentes, sem nenhuma modificação (PYTHON, 2011), é também uma linguagem multiparadigma, e a orientação a objetos é completamente suportada, sendo compatível com a escolha do paradigma escolhido.

Python possui uma extensa biblioteca padrão, a qual contém classes, métodos e funções para realizar essencialmente qualquer tarefa, desde acesso a bancos de dados à interfaces gráficas com o usuário(*id.*, 2011).

4.4.3 Frameworks

Para auxílio no projeto juntamente com a biblioteca padrão Python, será utilizado o *framework* Django, um *framework* de desenvolvimento rápido para a web, de código aberto, distribuído sob licença BSD, e que utiliza princípios de reaproveitamento de código fazendo com que o desenvolvedor aproveite ao máximo o código já feito, evitando a repetição (DJANGO ,2011).

4.4.4 Arquitetura

A arquitetura do sistema será desenvolvida utilizando MTV (Model-Template-View), uma arquitetura em camadas semelhante ao MVC (Model-View-Controller), mas com pequenas variações, é uma arquitetura que separa a aplicação em camadas bem definidas as lógicas de controle, dados e apresentação.

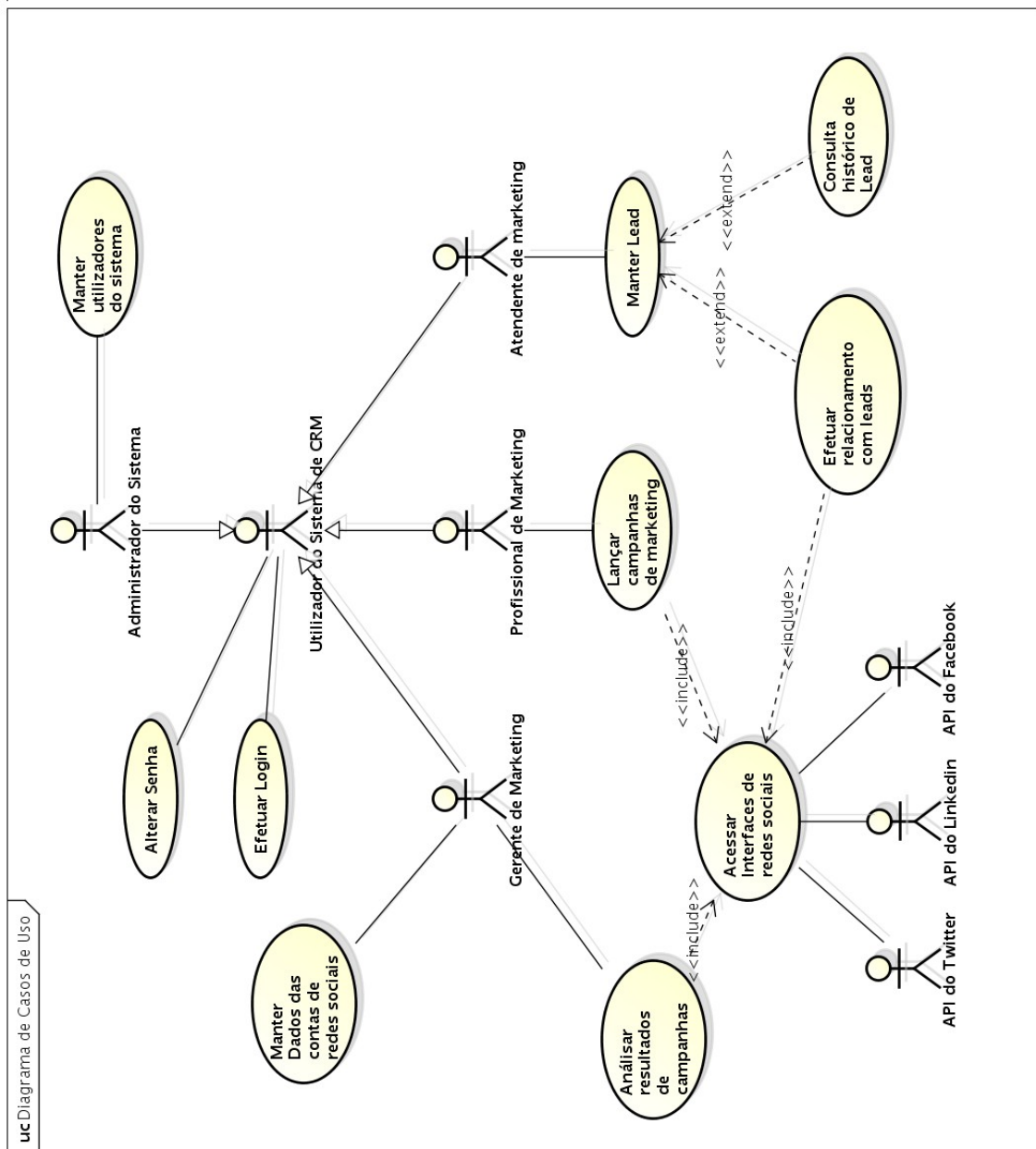
4.4.5 Persistência

Para a persistência será utilizado o banco de dados MongoDB, um banco não-relacional orientado a documentos, de alta performance e escalabilidade, com esquemas flexíveis, e armazenamento em coleções de pares chave-valor (MONGODB, 2011) , a opção da utilização deste tipo de banco de dados se deve a alta escalabilidade, performance e flexibilidade requerida pela aplicação a ser desenvolvida.

5 MODELO DE CASOS DE USO

5.1 DIAGRAMA DE CASOS DE USO

FIGURA 4 – DIAGRAMA DE CASOS DE USO



NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

5.2 ATORES

Utilizador do Sistema de CRM: Ator que engloba todos os utilizadores do sistema de Social CRM.

Gerente de Marketing: Gerente de marketing, responsável por fazer a análise dos resultados de campanhas de redes sociais, bem como manter os dados das mesmas.

Administrador do Sistema: Podendo ser qualquer papel dentro da organização, mas que possua a função da administração da aplicação, pode ser um gerente de T.I, suporte técnico ou gerência de marketing.

Profissional de marketing: Ator responsável por lançar as campanhas de marketing em redes sociais.

Atendente de marketing: Atendente responsável por realizar o relacionamento direto com os leads, e manter os dados cadastrais dos mesmos.

API do Twitter: API externa a qual possui webservices específicos para pesquisas, buscas e interações com a rede twitter.

API do Facebook: API externa a qual possui webservices específicos para pesquisas, buscas e interações com a rede facebook.

API do LinkedIn: API externa a qual possui webservices específicos para pesquisas, buscas e interações com a rede LinkedIn.

6 COMPOSIÇÃO DOS CASOS DE USO

6.1 MANTER UTILIZADORES DO SISTEMA

Descrição: Caso de uso das operações CRUD dos utilizadores do sistema de CRM.

Pré-condição: Não se aplica.

Fluxo Normal:

1 – Visualizar grid.

O caso de uso inicia quando ator entra em módulo de utilizadores, sistema apresenta grid com utilizadores cadastrados, mostrando os dados de grupo e login, ocultando as senhas, e aguarda próximo passo.

2 – Escolher opção de cadastro.

Ator pressiona botão de cadastro, sistema abre dialogo de cadastramento de utilizadores contendo lista de grupos, entrada de nome, login e senha, e aguarda próximo passo.

3 – Selecionar grupo do utilizador.

Ator seleciona em lista o grupo a qual o utilizador está inserido, sistema mantém o dado no campo e aguarda próximo passo.

4 – Inserir nome do utilizador.

O ator insere em campo de entrada o nome do utilizador, sistema mantém o dado no campo e aguarda próximo passo.

5 – Inserir login do utilizador.

O ator insere em campo de entrada o login do utilizador, sistema mantém o dado no campo e aguarda próximo passo.

6 – Inserir senha do utilizador.

O ator insere em campo de entrada o login do utilizador, sistema mantém o dado no campo com caracteres especiais e aguarda próximo passo.

7 – Salvar.

O ator pressiona botão de salvamento, sistema grava os dados em banco de dados, mostra mensagem de sucesso e o caso de uso termina.

Fluxos Alternativos:

1 – Banco de dados indisponível.

Durante a execução do passo 7 do fluxo básico, “Salvar, ou os passos 2, 3 e 4 dos fluxos alternativos caso o banco de dados esteja indisponível, é apresentada mensagem de erro e o caso de uso termina.

2 – Consultar dados de utilizador.

Durante a execução do passo 1, “Visualizar grid”, após exibida uma grid com todas as redes sociais cadastradas no banco de dados, ator visualiza o conteúdo, sistema manter a grid aberta e o caso de uso termina.

3 – Excluir utilizador.

Durante a execução do passo 1, “Visualizar grid”, após exibida a grid com todos os utilizadores cadastrados, ator seleciona um dos utilizadores, seleciona botão de exclusão, sistema abre dialogo de confirmação, ator pressiona botão de confirmação, mostra mensagem de conclusão e o caso de uso termina.

4 – Alterar dados utilizador.

Durante a execução do passo 1, “Visualizar grid”, após exibida a grid com todos os utilizadores cadastrados, ator seleciona uma dos utilizadores, seleciona botão de alteração, sistema habilita os campos da grid de grupo, login, senha, e botão de alteração, ator preenche os campos do dialogo, pressiona botão de alteração, sistema efetua a alteração dos dados, mostra mensagem de conclusão e o caso de uso termina.

6.2 EFETUAR LOGIN

Descrição: Caso de uso para acesso de atores ao sistema de CRM.

Pré-condição: Não se aplica.

Fluxo Normal:

1 – Informar login.

O caso de uso inicia quando ator informa login, sistema mantém dado no campo e aguarda próximo passo.

2 – Informar Senha.

Ator informa senha, sistema mantém senha em caracteres especiais e aguarda próximo passo.

3 - Entrar .

Ator pressiona botão entrar, sistema efetua validação do usuário, o loga no sistema e o caso de uso termina

Fluxos Alternativos:

1 – Banco de dados indisponível.

Durante a execução do passo 3, “Entrar”, caso o banco de dados esteja indisponível, sistema exibe mensagem de erro e caso de uso termina.

2 – Login inválido.

Durante a execução do passo 3, “Entrar”, caso o login for inválido, sistema exibe mensagem de erro e caso de uso retorna para passo 1.

3 – Senha inválida.

Durante a execução do passo 3, “entrar”, caso a senha seja inválida, sistema exibe mensagem de erro e caso de uso retorna para passo 2.

Pós-condição: Efetuado login no sistema.

6.3 ALTERAR SENHA

Descrição: Caso de uso para alteração de senha de atores.

Pré-condição: Não se aplica.

Fluxo Normal:

1 – Informar senha antiga.

O caso de uso inicia quando ator informa senha, sistema mantém senha em caracteres especiais e aguarda próximo passo.

2 – Informar senha atual.

Ator informa senha, sistema mantém senha em caracteres especiais e aguarda próximo passo.

3 - Salvar.

Ator pressiona botão de salvamento, sistema efetua validação de senha antiga em banco de dados, grava nova senha e o caso de uso termina

Fluxos Alternativos:

1 – Banco de dados indisponível.

Durante a execução do passo 4 , “Salvar”, caso o banco de dados esteja indisponível, sistema exibe mensagem de erro e caso de uso termina.

2 – Senha antiga inválida.

Durante a execução do passo 4, “Salvar”, caso a senha antiga for inválida, sistema exibe mensagem de erro e caso de uso retorna para passo 1.

3 – Senha e formato inválido.

Durante a execução do passo 4, “Salvar”, caso a senha esteja em formato inválido, sistema exibe mensagem de erro e caso de uso retorna para passo 2.

Pós-condição: Senha alterada com sucesso.

6.4 MANTER DADOS DAS CONTAS DE REDES SOCIAIS

Descrição: Casos de uso para as operações CRUD dos dados das contas das redes sociais, ou seja, as contas as quais serão enviadas as campanhas de marketing.

Pré-condição: Não se aplica.

Fluxo Normal:

1 – Visualizar grid

O caso de uso inicia quando ator visualiza uma grid com todas as redes sociais cadastradas no banco de dados, se não houver nenhum registro no banco de dados a grid é exibida vazia, somente com os botões de acesso, sistema aguarda próximo passo.

2 – Pressionar botão de inclusão rede social

Ator pressiona botão de inclusão, sistema abre dialogo contendo seletor de rede social, campos de login, campo de senha da rede social e botão de finalização.

3 – Selecionar tipo de rede social.

O caso inicia quando ator seleciona o tipo de rede social em lista, entre elas Twitter, LinkedIn e Facebook, após a seleção, sistema aguarda próximo passo.

4 – Inserir login da rede social.

Ator insere os dados de login em campo de entrada, sistema mantém cursor no campo e aguarda próximo passo.

5 – Inserir senha da rede social.

Ator insere os dados de senha em campo de entrada, sistema mantém a entrada em caracteres especiais e aguarda próximo passo.

6 – Salvar.

Ator pressiona botão de salvamento, sistema salva os dados no banco de dados, apresenta mensagem de confirmação e caso de uso termina.

Fluxos Alternativos:

1 – Banco de dados indisponível.

Durante a execução do passo 6 do fluxo básico, “Salvar, ou os passos 2, 3 e 4 dos fluxos alternativos caso o banco de dados esteja indisponível, é apresentada mensagem de erro e o caso de uso termina.

2 – Consultar dados de rede social.

Durante a execução do passo 1, “Selecionar tipo de rede social”, após exibida uma grid com todas as redes sociais cadastradas no banco de dados, ator visualiza o conteúdo, sistema manter a grid aberta e o caso de uso termina.

3 – Excluir rede social.

Durante a execução do passo 1, “Selecionar tipo de rede social”, após exibida a grid com todas as redes sociais cadastradas, ator seleciona uma das redes sociais, seleciona botão de exclusão, sistema abre dialogo de confirmação, ator pressiona botão de confirmação, mostra mensagem de conclusão e o caso de uso termina.

4 – Alterar dados de rede social.

Durante a execução do passo 1, “Selecionar tipo de rede social”, após exibida a grid com todas as redes sociais cadastradas, ator seleciona uma das redes sociais, seleciona botão de alteração, sistema libera campos de grid com campos de entrada de login, senha, e botão de alteração, ator preenche os campos do dialogo, pressiona botão de alteração , sistema efetua a alteração dos dados, mostra mensagem de conclusão e o caso de uso termina.

6.5 MANTER LEAD

Descrição: Casos de uso para operações CRUD em leads.

Pré-condição: Não se aplica.

Fluxo Normal:

1 – Visualizar *grid*.

O caso de uso inicia quando ator visualiza uma *grid* com todas os leads cadastrados no banco de dados, se não houver nenhum registro no banco de dados a *grid* é exibida vazia, contendo somente os botões de acesso, sistema aguarda próximo passo.

2 – Selecionar opção de cadastro.

Ator seleciona opção de relacionamento de cadastro, sistema abre dialogo de cadastramento e aguarda próximo passo.

3 – Incluir dados cadastrais.

Ator insere os dados de nome, email, endereço do twitter, endereço do facebook e endereço do linkedin, sistema mantém os dados nos campos e aguarda próximo passo.

5 – Salvar *lead*.

Após o ator ter selecionado opção de salvamento, o sistema persiste os dados em banco de dados, apresenta mensagem de confirmação e o caso de uso termina.

Fluxos Alternativos:

1 - Consultar *lead*.

Durante a execução do passo 1, “Visualizar *grid*”, caso o ator opte por somente visulizar o *lead*, sistema mantém os dados na *grid* e o caso de uso termina.

2 - Alterar *lead*.

Durante a execução do passo 1, “Visualizar *grid*”, caso o ator opte por alterar um *lead*, ator pressiona botão de alteração, sistema abre dialogo com campos preenchidos, ator modifica os campos que deseja, sistema mantém os dados nos campos, ator pressiona botão de confirmação, sistema altera estes dados no banco de dados, apresenta mensagem de confirmação e o caso de uso termina.

3 - Excluir *lead*.

Durante a execução do passo 1, “Visualizar *grid*”, caso o ator opte por excluir um *lead*, ator pressiona botão de exclusão, sistema abre dialogo com pergunta de confirmação, ator pressiona botão de confirmação e o caso de uso termina.

4 - Erro em banco de dados

Durante a execução do passo 4 do fluxo básico, “Salvar *lead*”, ou os passos 1, 2, 3 do fluxo alternativo caso o banco de dados esteja indisponível, é apresentada mensagem de erro e o caso de uso termina.

Pontos de extensão:

Ponto de extensão 1: O ponto de extensão “Efetuar relacionamento com *leads*” ocorre no fluxo básico depois do passo 4, “Salvar *Lead*”, quando em vez do ator pressionar botão de salvar *lead*, pressionar botão de salvar e efetuar relacionamento com *lead* ou no passo 1 do fluxo alternativo “Consultar *Lead*”, quando ator seleciona opção para efetuar relacionamento com *lead*.

Ponto de extensão 2: O ponto de extensão “Consulta histórico de *Lead*” ocorre no fluxo alternativo no passo 1, “Consultar *Lead*”, quando o ator visualiza o *lead* e pressiona o botão ver histórico de *lead*.

6.6 CONSULTA HISTÓRICO DE *LEAD*

Descrição: Caso de uso de extensão para consulta de histórico de leads.

Pré-condição: Ter escolhido opção ver histórico de lead em caso de uso base.

Fluxo Normal:

1 – Procurar lead.

O caso de uso inicia quando ator seleciona o *lead*, sistema abre *grid* com todos os relacionamentos feitos com este *lead*.

2 – Visualizar histórico do *lead*.

Ator visualiza a lista de histórico contendo a data e hora do contato, e sistema aguarda próximo passo.

3 – Selecionar detalhe do histórico.

Ator seleciona item do histórico, sistema abre dialogo contendo os detalhes deste contato com o *lead* como data do contato, hora do contato, mensagem enviada e qual rede social foi feito o contato e aguarda próximo passo.

3 – Fechar

Ator seleciona opção de fechamento da interface de histórico, sistema fecha a interface e o caso de uso termina.

Fluxos Alternativos:

4 - Erro em banco de dados

Durante a execução do passo 1 do fluxo básico, "Procurar *lead*", caso o banco de dados esteja indisponível, é apresentada mensagem de erro e o caso de uso termina.

6.7 EFETUAR RELACIONAMENTO COM LEADS

Descrição: Casos de uso de extensão para relacionamento com os *leads* afetados pelas campanhas de marketing disparadas.

Pré-condição: Ter escolhido opção de relacionamento com *lead* em caso de uso base.

Fluxo Normal:

1 – Selecionar campanha.

O caso de uso inicia quando ator seleciona campanha em lista através do identificador, sistema realiza a busca de todos os *leads* relacionados com esta campanha e aguarda próximo passo.

2 – Visualizar *leads*.

Após o ator ter selecionado a campanha e o sistema efetuar a busca dos *leads*, os mesmos são populados em *grid* contendo os dados do *lead* e aguarda próximo passo.

3 – Seleciona *lead*.

Ator seleciona *lead* pressionando botão de relacionamento, sistema abre dialogo contendo campos de lista de seleção de forma de contato, e campo de texto para escrita da mensagem, e aguarda próximo passo.

4 – Seleciona forma de contato.

Ator seleciona a forma de contato baseada na rede social a qual o *lead* está contido, e baseado nesta rede social sistema habilita forma de relacionamento com o *lead*, sendo eles *scrap* para o Facebook, tweet para o twitter ou e-mail para o LinkedIn, após esta escolha sistema mantém dados nos campos e aguarda próximo passo.

5 – Escreve mensagem.

Ator escreve mensagem em campo de texto tendo em vista o tipo de forma de contato escolhida, sistema mantém os dados nos campos e aguarda próxima interação.

6 – Envia mensagem

Ator pressiona botão de envio de mensagem, sistema realiza próximo passo.

7 – Incluir: “Acessar Interfaces de redes sociais”.

8 – Fechar.

Após realização do envio da mensagem, sistema apresenta mensagem de confirmação, ator seleciona opção de fechamento, sistema fecha interface e o caso de uso termina.

Fluxos Alternativos:

1 - Erro em banco de dados.

Durante a execução do passo 6 do fluxo básico, “Enviar mensagem”, caso o banco de dados esteja indisponível para sistema poder persistir os dados históricos, é apresentada mensagem de erro e o caso de uso termina.

1 – Redes sociais indisponíveis.

Durante a execução do passo 6 do fluxo básico, “Enviar mensagem”, caso alguma interface de rede social esteja indisponível, é apresentada mensagem de erro referente a rede social, porém sistema efetua o relacionamento nas outras redes sociais disponíveis e o caso de uso termina.

6.8 LANÇAR CAMPANHAS DE MARKETING

Descrição: Caso de uso para lançamento de campanhas para as redes sociais.

Pré-condição: Possuir contas de redes sociais cadastradas no sistema.

Fluxo Normal:

1– Selecionar redes sociais.

O caso de uso inicia quando ator seleciona através de caixa de seleção quais redes sociais a campanha será disparada e aguarda próximo passo.

2 – Atribuir identificador.

Ator insere em campo de texto um identificador ou palavra-chave para posterior localização, sistema mantém cursor no campo e aguarda próximo passo.

3 – Escrever descrição da campanha de marketing.

Ator escreve descrição da campanha de marketing a ser lida pelos consumidores, se selecionada rede twitter sistema restringe a descrição a 140 caracteres, para os demais não existe restrição, após inserção da descrição sistema aguarda próximo passo.

4 – Disparar campanha.

Ator pressiona botão de disparo de campanha, sistema persiste os dados para guardar histórico da ação, e realiza próximo passo.

5 – Incluir: “Acessar Interfaces de redes sociais”.

6 – Fechar

Após acesso interfaces com redes sociais, sistema mostra aviso de campanha disparada, ator seleciona opção de fechamento, sistema fecha a interface e o caso de uso termina.

Fluxos Alternativos:

1 – Banco de dados indisponível.

Durante a execução do passo 4, “Disparar campanha”, caso o banco de dados esteja indisponível, sistema exibe mensagem de erro e caso de uso termina.

2 – Sem redes sociais cadastradas.

Ao iniciar o caso de uso, caso não tenham sido cadastradas contas de redes sociais, exibe mensagem de erro e caso de uso termina.

3 – Interfaces com redes sociais não acessíveis.

Durante a execução do caso de uso 4, “Disparar campanha”, caso não seja possível acessar alguma rede social, sistema apresenta erro referente a rede social mas continua a interação com as redes sociais que não apresentam erro de acesso e o caso de uso termina.

Pós-condição: Campanha lançada com sucesso nas redes sociais.

6.9 ANALISAR RESULTADOS DE CAMPANHAS

Descrição: Casos de uso para realização a análise das campanhas, bem como emissão de relatórios.

Pré-condição: Existir pelo menos uma campanha cadastrada.

Fluxo Normal:

1 – Buscar campanha.

O caso de uso iniciar quando ator posiciona cursor em campo e começa a digitar o identificador e o sistema de forma assíncrona busca em banco de dados os identificadores relacionados as letras digitadas, usuário escolhe uma campanha e sistema aguarda próximo passo.

2 – Escolher a campanha.

Após escolhido uma campanha, ator pressiona botão de busca, sistema efetua a busca dos resultados no banco de dados e aguarda próximo passo.

3 – Incluir: “Acessar Interfaces de redes sociais”.

4 – Analisar dados da campanha.

Após os resultados serem exibidos na tela, ator analisa os dados formados por tabela, gráficos em pizza e gráficos em barra e sistema aguarda próximo passo.

5 – Emitir relatório.

Ator pressiona botão de impressão, sistema abre dialogo para ator selecionar a impressora, após selecionado sistema acessa a impressora e imprime os relatórios de resultados da campanhas.

6 – Fechar.

Ator seleciona botão de fechamento, sistema fecha a interface e o caso de uso termina.

Fluxos Alternativos:

1 – Banco de dados indisponível.

Durante a execução do passo 1 do fluxo básico, “Buscar campanha”, caso o banco de dados esteja indisponível, é apresentada mensagem de erro e o caso de uso termina.

2 – Impressora indisponível.

Durante a execução do passo 5 do fluxo básico, “Emitir relatório”, caso a impressora esteja indisponível, é apresentada mensagem de erro e o caso de uso retorna para passo 4, “fechar”.

3 – Não emitir relatório.

Durante a execução do passo 5 do fluxo básico, “Emitir relatório”, caso o ator opte por não emitir o relatório, a impressora não é acionada e o caso de uso retorna para passo 4, “fechar”.

6.10 ACESSAR INTERFACES DE REDES SOCIAIS

Descrição: Casos de uso para manipulação das sessões, interfaces e dados das redes sociais.

Pré-condição: Existir pelo menos uma conta de rede social cadastrada.

Fluxo Normal:

1 – Consultar dados de conexão com rede social.

O caso de uso inicia quando sistema recebe ativação externa, verifica em ativação qual a rede social necessita lançar a campanha, acessa banco de dados, consulta os dados de conexão desta rede social e aguarda próximo passo.

2 – Estabelecer conexão com rede social.

Sistema estabelece conexão com rede social escolhida, através de API específicas desta rede social e aguarda próximo passo.

3 – Lançar campanha na rede social.

Sistema ainda com a conexão aberta com a API, lança a campanha na rede social e aguarda próximo passo.

4 – Fechar conexão com a rede social.

Sistema efetua o retorno da confirmação de lançamento de campanha concluída, fecha a conexão com a rede social e o caso de uso termina.

Fluxos Alternativos:

1 – Consultar resultados de campanhas.

No passo 3, “Lançar campanha na rede social”, se caso de uso base optou por consultar os resultados das campanhas, sistema consulta os resultados obtidos pela campanha através da API da rede social, retorna estes dados, fecha a conexão com a API e o caso de uso termina.

2 - Efetuar relacionamento com *leads*.

No passo 3 do fluxo básico, “Lançar campanha na rede social”, se caso de uso base optou por efetuar o relacionamento com os *leads*, sistema envia o relacionamento através da API da rede social, retorna dados de confirmação, fecha a conexão com a API e o caso de uso termina.

3 – Erro em banco de dados

No passo 1 do fluxo básico, “Consultar dados de conexão com rede social”, caso o banco de dados esteja indisponível, é exibido mensagem de erro de conexão com banco de dados e o caso de uso termina.

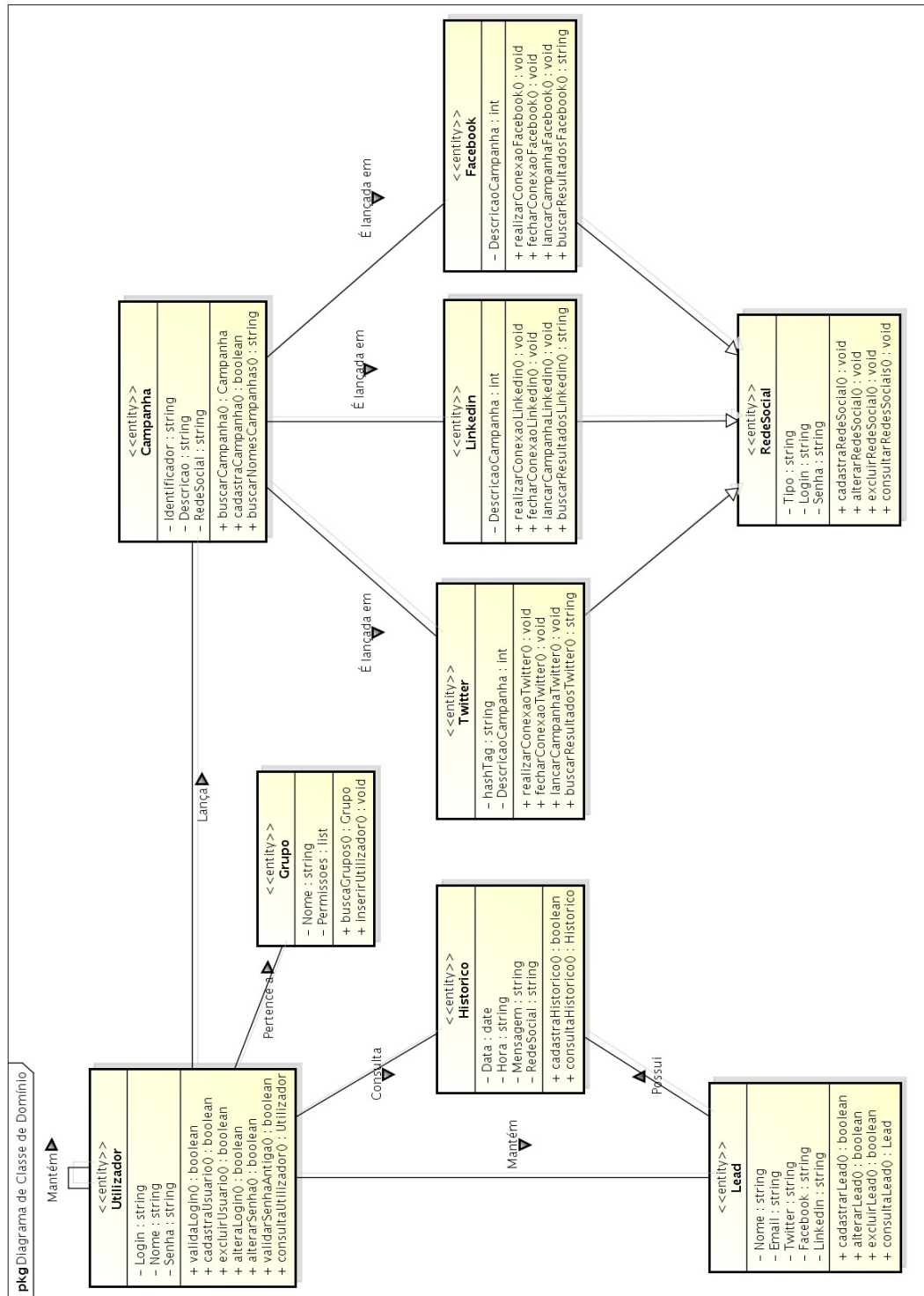
4 – API de rede social inacessível.

No passo 2 do fluxo básico, “Estabelecer conexão com rede social”, caso a API de conexão da rede social esteja indisponível, é exibido mensagem de erro de conexão com API e o caso de uso termina.

7 COMPOSIÇÃO DA REALIZAÇÃO DOS CASOS DE USO

7.1 DIAGRAMA DE CLASSE DE ANÁLISE DE DOMÍNIO

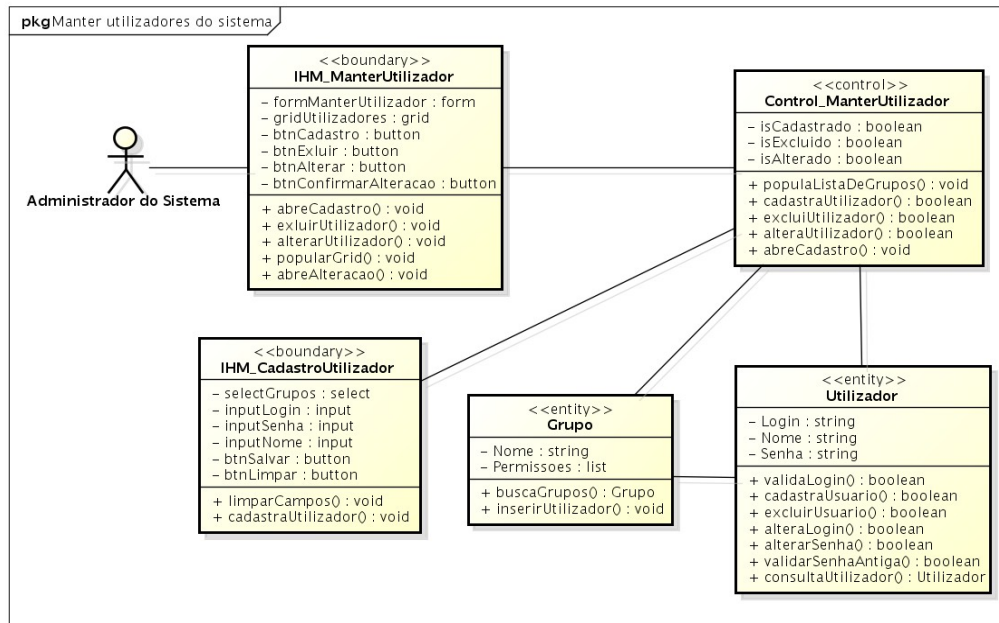
FIGURA 5 – DIAGRAMA DE CLASSE DE ANÁLISE DE DOMÍNIO



NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

7.2 DIAGRAMAS DE CLASSE DE ANÁLISE POR CASO DE USO

FIGURA 6 – DIAGRAMA DE CLASSE - MANTER UTILIZADORES DO SISTEMA



NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 7 – DIAGRAMA DE CLASSE - EFETUAR LOGIN

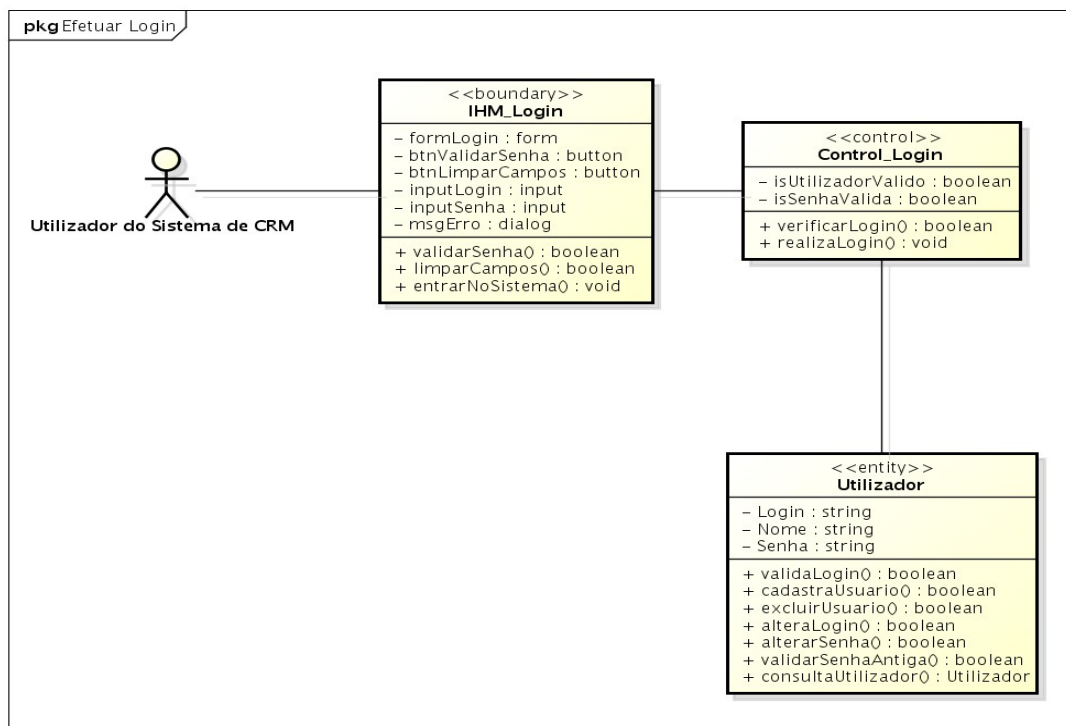
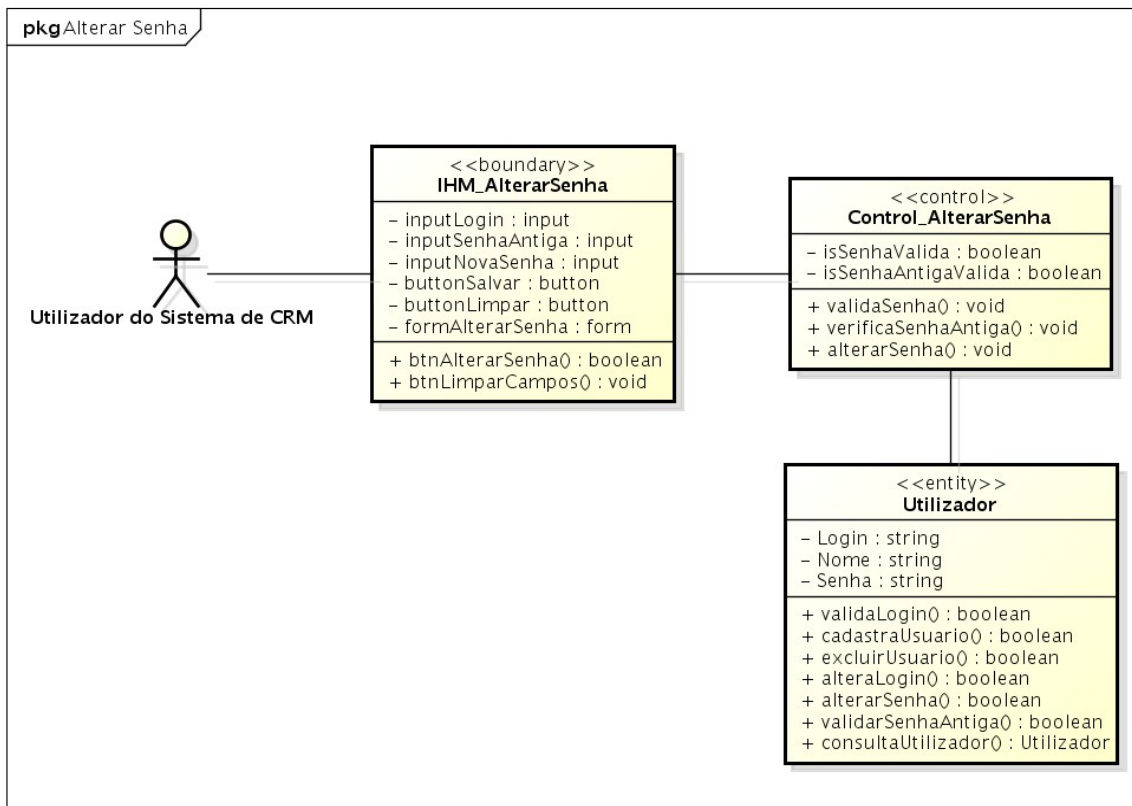
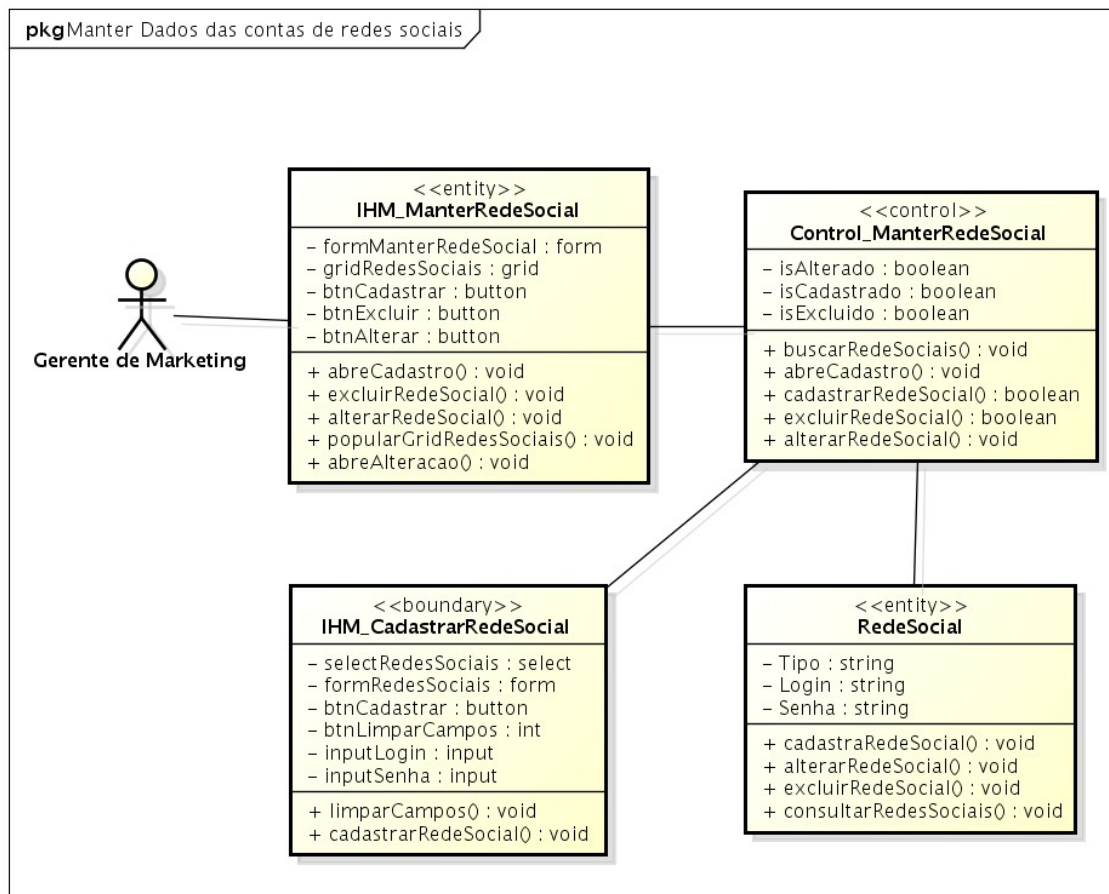


Figura 8 – DIAGRAMA DE CLASSE - ALTERAR SENHA



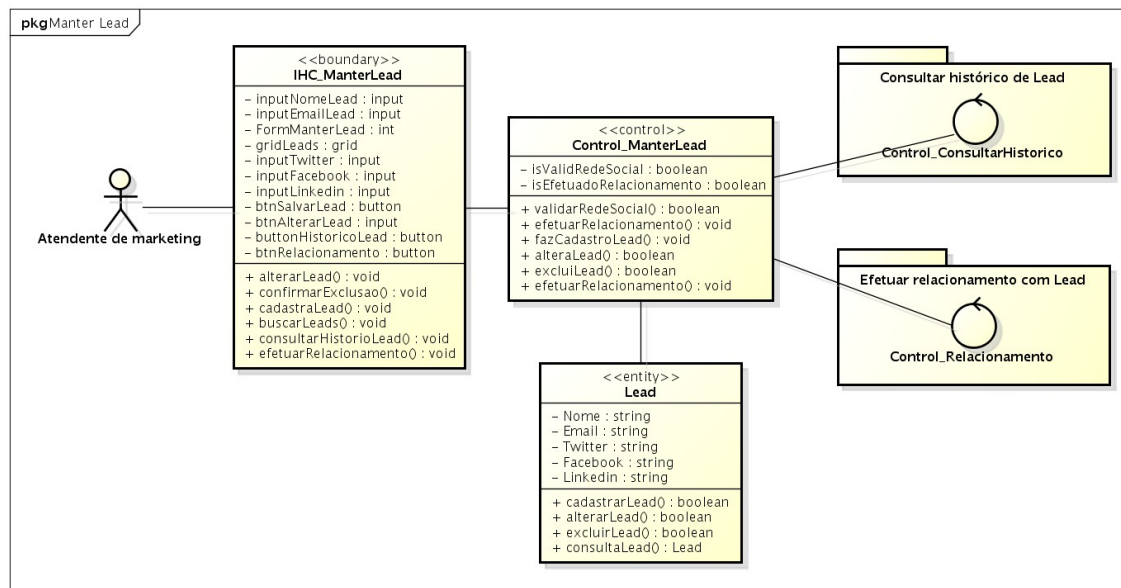
NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 9 – DIAGRAMA DE CLASSE - MANTER DADOS DAS CONTAS DE REDES SOCIAIS



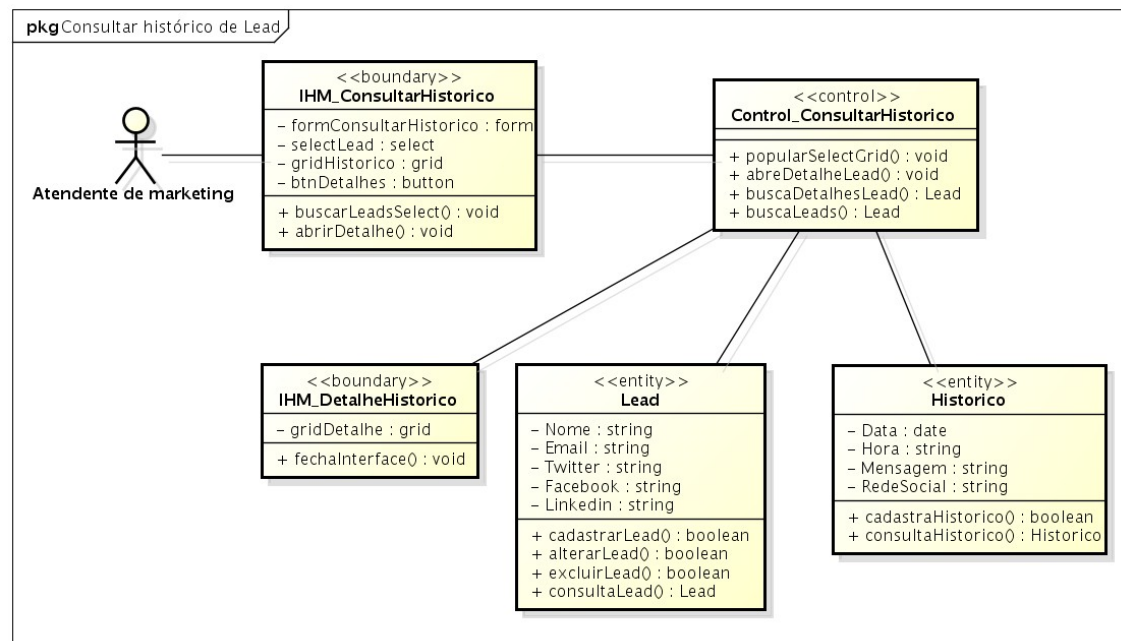
NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 10 – DIAGRAMA DE CLASSE - MANTER LEAD



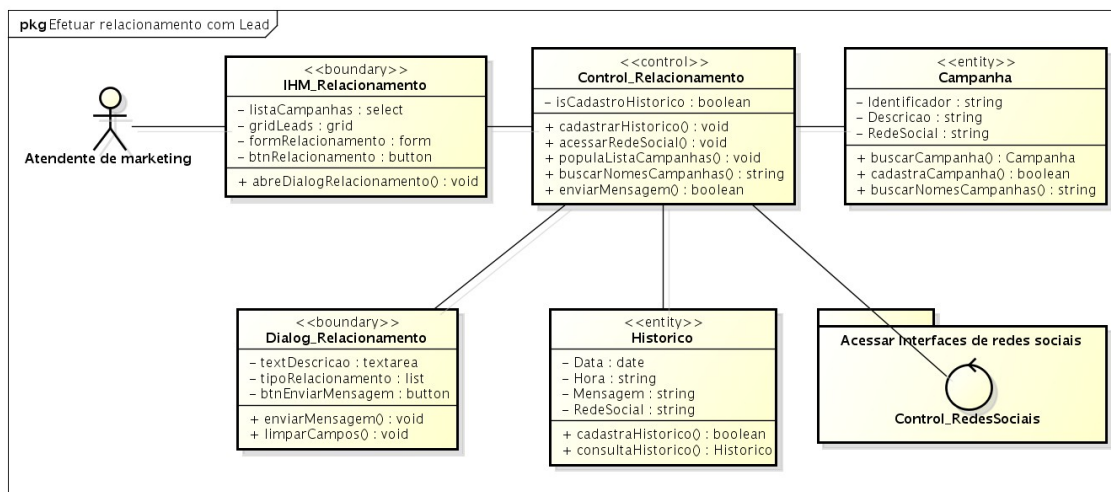
NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

Figura 11 – DIAGRAMA DE CLASSE - CONSULTAR HISTÓRICO DE LEAD



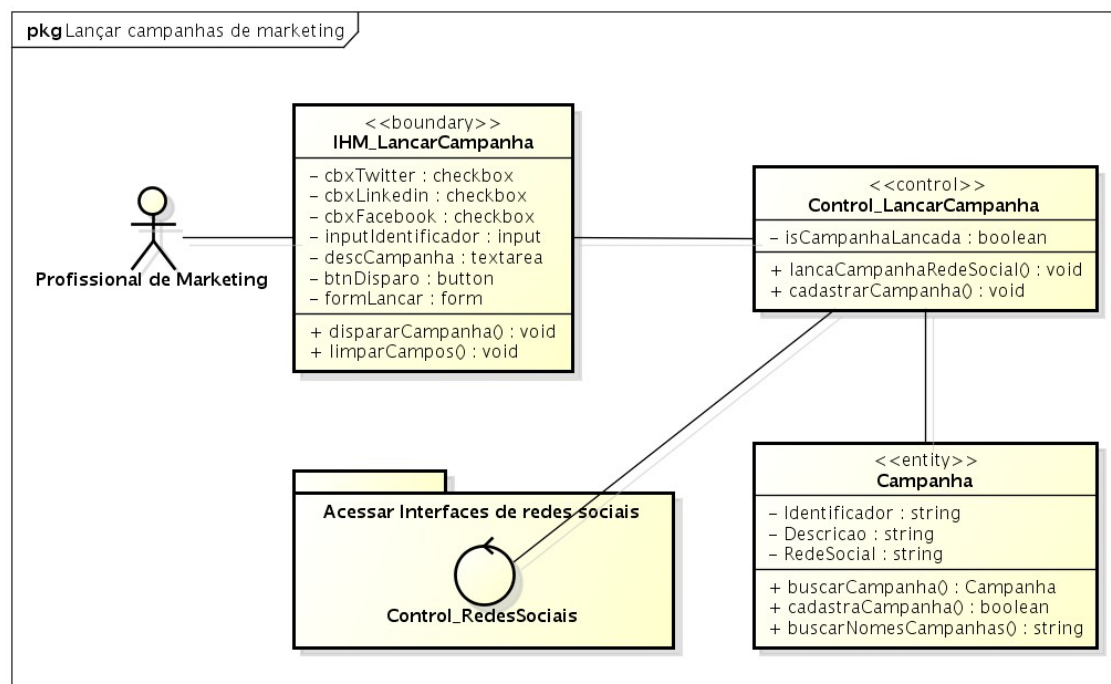
NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 12 – DIAGRAMA DE CLASSE - EFETUAR RELACIONAMENTO COM LEAD



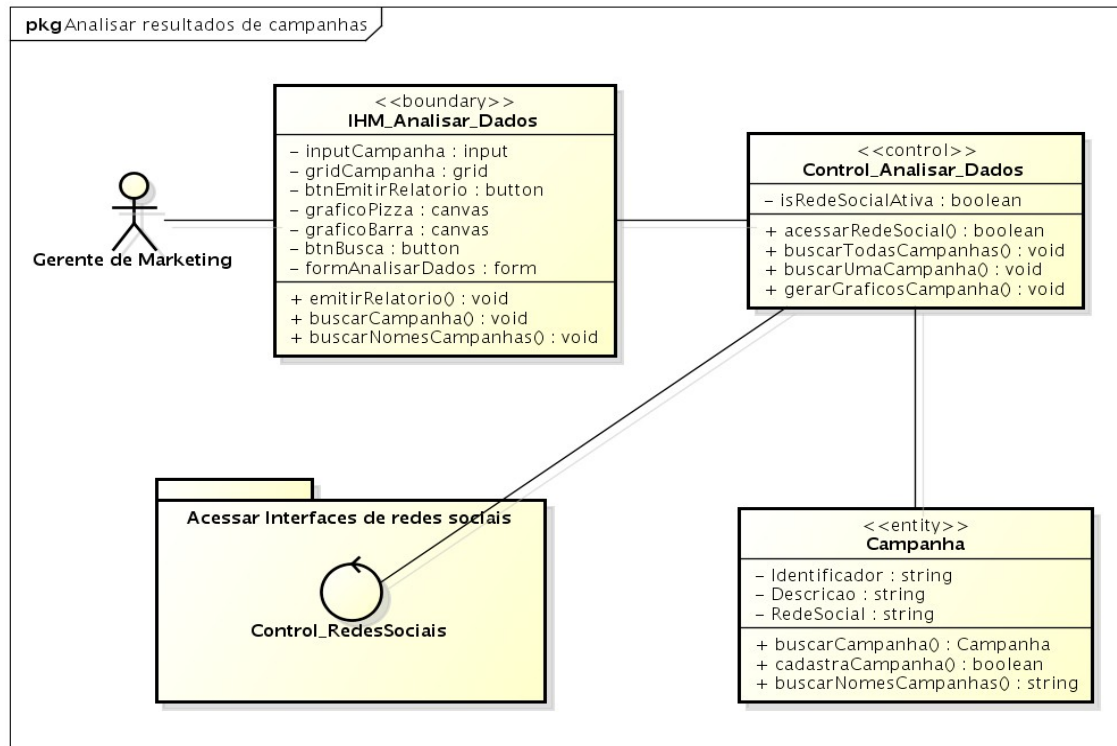
NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 13 – DIAGRAMA DE CLASSE - LANÇAR CAMPANHAS DE MARKETING



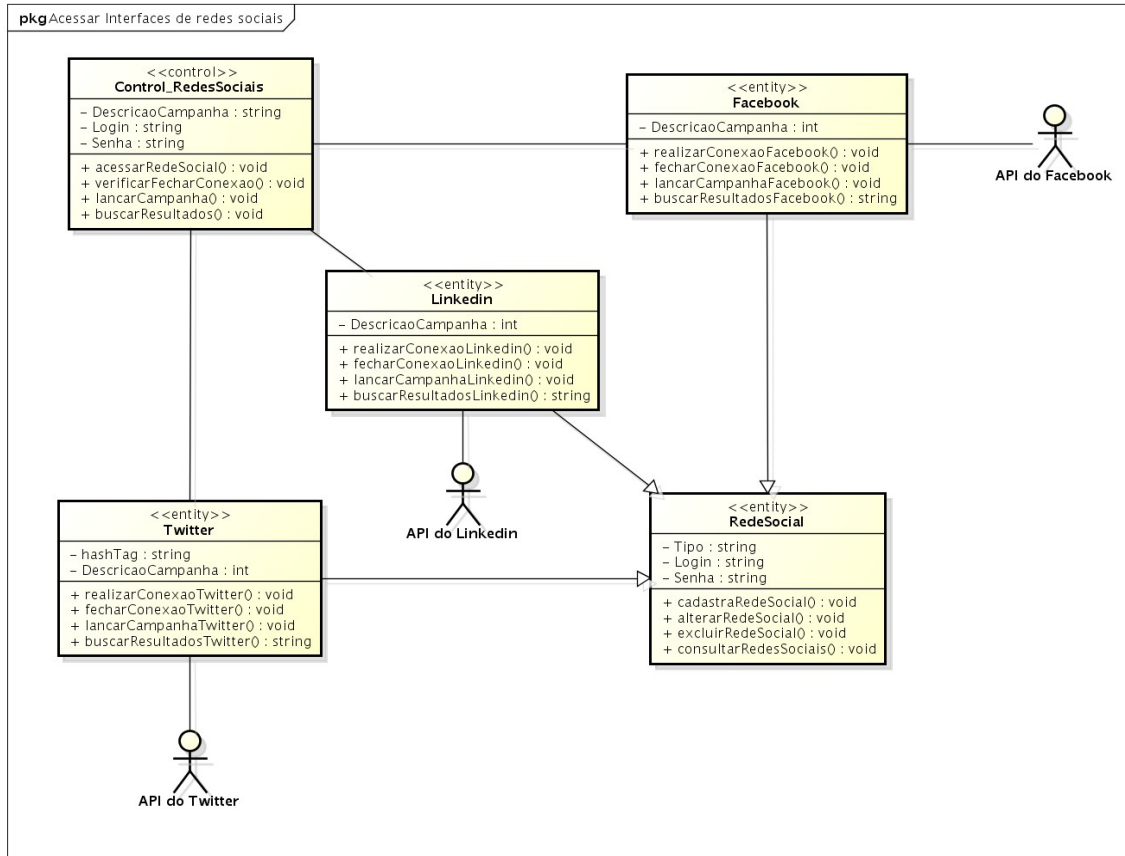
NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 14 – DIAGRAMA DE CLASSE - ANALISAR RESULTADOS DE CAMPANHAS



NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

FIGURA 15 – DIAGRAMA DE CLASSE - ACESSAR INTERFACES DE REDES SOCIAIS

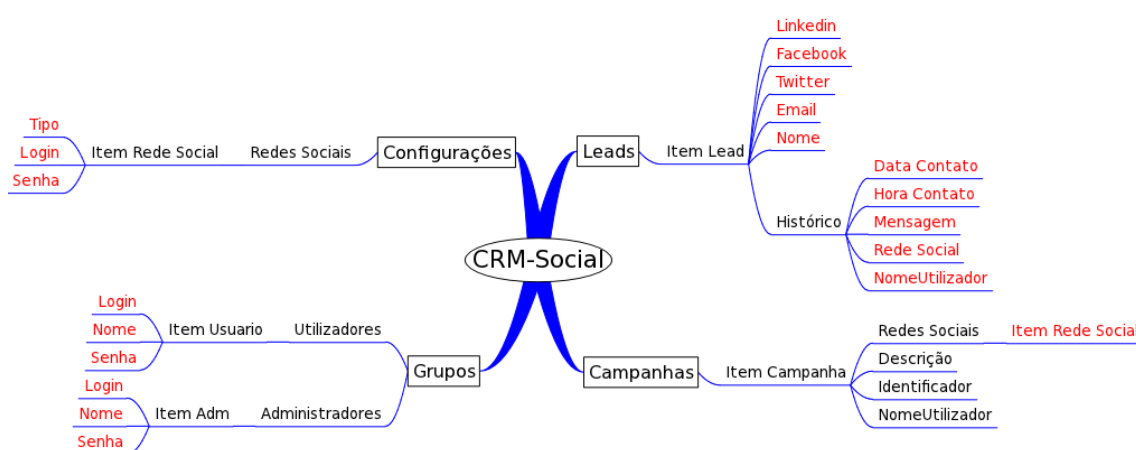


NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

8 MODELO DE DADOS

Será utilizado banco de dados não relacional baseado em documento, devido a esta opção, o modelo de dados é definido através de mapa mental, ou árvore hierárquica. segue figura do modelo de dados:

FIGURA 16 – MODELO DE DADOS NÃO RELACIONAL



NOTA: Diagrama criado pela equipe de desenvolvimento deste projeto.

Através da representação de dados em forma de mapa mental, o modelo apresenta as seguintes representações:

Elipse: Banco de Dados;

Retângulos: Coleções;

Nós de segundo nível: Documentos;

Nós folhas: Chaves.

9 PLANO DE TESTE

Esse tópico tem por finalidade descrever o planejamento geral das atividades de teste bem como descrição dos envolvidos, ambientes de testes e critérios de aceitação dos artefatos a serem testados.

9.1 PLANEJAMENTOS DOS TESTES

Esta sessão descreve o planejamento dos testes, ou seja, quais os objetos a serem testados, quais os estágios, qual a abordagem e qual técnica utilizada para realizar os testes.

Para o projeto serão utilizadas 3 perspectivas de testes, sendo elas: Testes de unidade, testes de integração, teste de sistema e teste de aceite.

9.1.1 Testes de unidade

Os testes de unidade serão realizados pelos próprios desenvolvedores e serão rodados através de ferramentas específicas da linguagem Python de forma automática.

Serão feitos testes unitários para no mínimo 70% dos métodos, e independente da cobertura a ser testada, pelo menos um caso de teste por componente desenvolvido deve ser gerado que procure realizar uma maior cobertura do código.

Segue quadro explicativo dos testes unitários:

QUADRO 7 – TESTES DE UNIDADE

Objeto	Classes (métodos)
Estágio	Fase de implementação
Abordagem	Caixa Branca
Técnica	Automática

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

9.1.2 Testes de integração

Serão construídos testes automáticos através da própria linguagem de programação para testar a integração de pelo menos 50% dos componentes.

Segue quadro explicativo do teste de integração:

QUADRO 8 – TESTES DE INTEGRAÇÃO

Objeto	Componentes e Módulos
Estágio	Fase de testes
Abordagem	Caixa Branca
Técnica	Automática

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

9.1.3 Testes de sistema

Os testes de sistema devem focar na iteração entre os componentes do sistema, para isso será realizado teste manual por parte dos desenvolvedores/testadores, o qual deve contemplar 100% dos casos de uso.

Segue quadro explicativo do teste de sistema:

QUADRO 9 – TESTES DE SISTEMA

Objeto	Fachada
Estágio	Fase de testes
Abordagem	Caixa Preta
Técnica	Manual

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

9.1.4 Testes de aceite

Os testes de aceitação serão realizados pela comunidade de software livre. Na liberação de cada *release*, os usuários serão comunicados pelas redes sociais do projeto, e os mesmos serão convidado a interagir livremente pelo sistema a procura de erros ao mesmo tempo em que verificam os requisitos do sistema.

Segue quadro explicativo do teste de aceite:

QUADRO 10 – TESTES DE ACEITE

Objeto	Interface do Usuário
Estágio	Fase de implantação
Abordagem	Caixa Preta
Técnica	Manual

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

9.2 CRITÉRIOS DE ACEITAÇÃO DOS TESTES

Para cada iteração do ciclo de vida do produto será realizado critério de aceitação de qualidade para os testes. Segue no quadro a seguir os critérios de aceitação para os testes que serão realizados de acordo com medidas de qualidade predefinidas.

Segue quadro com as iterações e critérios de aceite dos testes em cada iteração:

QUADRO 11 – CRITÉRIOS DE ACEITAÇÃO DOS TESTES

Iteração	Unidade	Integração	Sistema	Aceite
1	Testes de unidade implementados para 50% dos componentes.	Testes de integração para 10% dos componentes.	N/A	N/A
2	Testes de integração para os restantes 20% dos componentes. Testes de unidade passando 100%,	Testes de integração para os restantes 40% dos componentes.	N/A	N/A

3	N/A	Testes de integração passando 100%.	Testes de sistema para todos os use cases passando pelo menos 80%.	Teste de aceite validado pelo menos 50% pelos stakeholders.
4	N/A	N/A	Testes de sistema para todos os use cases passando pelo menos 100%.	Teste de aceite validado 100% pelos stakeholders .

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

9.3 CONFIGURAÇÃO E AMBIENTES DE TESTE

Esta sessão descreve a configuração e os ambientes de testes utilizados para efetuar todos os testes realizados no projeto.

Segue quadro de relação de ambientes e configurações de teste:

QUADRO 12 – CONFIGURAÇÃO E AMBIENTES DE TESTES

Recurso	Quantidade	Descrição
Servidor de aplicação	1	Desktop Intel Pentium 4 – 2GB RAM – 300gb HD – GNU/Linux Debian
Servidor de banco de Dados	1	Desktop AMD Atlon 64 – 4GB RAM – 500gb HD – FreeBSD
Computadores para testes	2	Notebook Core 2 Duo 6400 - 4GB RAM – 1 TB HD – GNU/Linux Fedora 15 Noteook – 4gb RAM – 500GB HD –

		GNU/Linux Mint
--	--	----------------

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

9.4 RECURSOS HUMANOS

Esta sessão descreve os recursos humanos, com suas respectivas funções e papéis que estão envolvidos com os testes, tanto para o desenvolvimento dos testes, quanto o aceite dos mesmos.

Segue quadro com relação dos recursos humanos:

QUADRO 13 – RECURSOS HUMANOS

Nome	Função	Papel
Evandro Balmant	Desenvolvedor	Desenvolver e realizar testes de sistema, e elaborar release de teste para aceite.
Wilian Stancke	Desenvolvedor	Desenvolver e realizar testes unitários e de integração e validar critérios de aceitação dos testes.

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

10 CASOS DE TESTE

Esta sessão descreve os casos de teste baseados nos caminhos básicos dos casos de uso. Segue tabela contendo identificador do caso de teste, caso de uso relacionado, dados de entrada, e resultados esperados:

QUADRO 14 – CASOS DE TESTE

Identificador	Caso de uso	Entradas	Resultado esperado:
TC1	MANTER UTILIZADORES DO SISTEMA	Grupo, nome, login e senha.	Cadastro concluído com sucesso.
TC2	EFETUAR LOGIN	Login e senha.	Acesso efetuado no sistema.
TC3	ALTERAR SENHA	Senha antiga e senha atual.	Senha alterada com sucesso.
TC4	MANTER DADOS DAS CONTAS DE REDES SOCIAIS	Tipo de rede social, login e senha.	Rede social cadastrada com sucesso.
TC5	MANTER LEAD	Nome, email, endereço do twitter, endereço do facebook e endereço do linkedin.	Lead cadastrado com sucesso.
TC6	CONSULTA HISTÓRICO DE LEAD	Nome ou identificador do lead.	Interface de visualização do lead.
TC7	EFETUAR RELACIONAMENTO	Lead, forma de contato e	Enviado mensagem para o

	COM LEADS	mensagem.	lead.
TC8	LANÇAR CAMPANHAS DE MARKETING	Tipo de rede social, identificador, descrição da campanha.	Enviado campanha em rede social escolhida.
TC9	ANALISAR RESULTADOS DE CAMPANHAS	Identificador da campanha.	Relatórios dos resultados da campanha.
TC10	ACESSAR INTERFACES DE REDES SOCIAIS	Dados de conexão da rede social.	Dados dos resultados das campanhas.

NOTA: Quadro elaborado pela equipe de desenvolvimento deste projeto.

11 CONCLUSÃO

Baseado na proposta original deste trabalho para o desenvolvimento de uma aplicação web para atender os setores de marketing, conclui-se que é viável implementar esta solução em software seguindo os modelos desenvolvidos e a especificação produzida, ocasionando assim, a solução do problema de integração dos dados captados em diferentes tipos de redes sociais.

Porém para estudo futuro, recomenda-se um melhor detalhamento das classes de análise, desenvolvimento de classes de implementação, desenvolvimento dos diagramas de atividade para cenários complexos da aplicação, e desenvolvimento de casos de testes para cenários de fluxo alternativo dos casos de uso.

REFERÊNCIAS

10GEN. <http://www.10gen.com>. Data de acesso: 15/08/2011.

ABOUT the eclipse foundation. <http://www.eclipse.org/org>. Data de acesso: 20/05/2011.

APACHE license, version 2.0 (current). <http://www.apache.org/licenses>. Data de acesso: 27/05/2011.

ASTAH Community faq. <Http://astah.change-vision.com/en/faq/com-faq.html>. Data de acesso: 27/05/2011.

DJANGO documentation. Disponível em: <http://docs.djangoproject.com>. Data de acesso: 20/05/2011.

EDITOR VI. Disponível em: <http://www.infowester.com/linuxvi.php>. Data de acesso: 15/08/2011.

GNU AFFERO GENERAL PUBLIC LICENSE – GNU PROJECT. Disponível em <http://www.gnu.org/licenses/agpl.html>. Data de acesso: 15/08/2011.

GNU PROJECT. Disponível em <http://www.gnu.org>. Data de acesso: 15/08/2011.

ICONIX Process. Disponível em: <http://www.iconixprocess.com/iconix-process>. Data de acesso: 27/05/2011.

PORTAL HSM. Infográfico: As redes sociais preferidas no mundo corporativo. Disponível em: <http://www.hsm.com.br/editorias/infografico-redes-sociais-preferidas-no-mundo-corporativo>. Data de acesso: 17/08/2011.

LEE, Richard C, TEPFENHART, William M. UML e C++: Guia Prático de desenvolvimento orientado a objeto. 1 edição. Makron Books, 2001.

MONGODB. Disponível em: <http://www.mongodb.org>. Data de acesso: 15/04/2011.

OBJECT MANAGEMENT GROUP – UML. Disponível em <http://www.uml.org>. Data de acesso: 15/08/2011.

PYTHON.ORG. Disponível em <http://www.python.org>. Data de acesso: 15/08/2011.

PEP8 Style guide for python code. Disponível em: <http://www.python.org/dev/pepspep-0008>. Data de acesso: 20/05/2011.

PRESSMAN, Roger, LOWE, David. Engenharia Web. 1 edição. LTC, 2009.

PROSPECÇÃO DE CLIENTES. Disponível em <http://prospeccao-de-clientes.com/o-que-e-prospeccao-de-clientes-sabe-a-definicao-ou-o-significado-440.html>. Acessado em 16/08/2011.

SERVIDOR APACHE. Disponível em: http://pt.m.wikipedia.org/wiki/Servidor_Apache. Data de acesso: 16/08/2011.

SERVIDOR WEB. Disponível em: http://pt.m.wikipedia.org/wiki/Servidor_web. Data de acesso: 15/08/2011.

USABILIDOIDO – AFINAL O QUE É USABILIDADE ? Disponível em http://usabilidoido.com.br/afinal_o_que_e_usabilidade.html

GLOSSÁRIO

10gen: Empresa norte-americana criadora e mantenedora do *MongoDB*.

AGPL (GNU Affero General Public License): é uma licença *copyleft* livre para softwares e outros tipos de obras, especialmente concebidos para assegurar a cooperação com a comunidade no caso do *software* servidor de rede.

AL (Apache License): é uma licença para *software* livre (*open source*) de autoria da Apache Software Foundation (ASF).

Análise de sentimento: é a verificação realizada sobre uma publicação para identificar se é uma reclamação ou um elogio, por exemplo.

Apache: é o mais bem sucedido servidor web livre. Em 2010 ele representava 54% de todos os sites e mais de 66% dos milhões de sites mais movimentados.

API (Application Programming Interface): Interface de Programação de Aplicativos, é o conjunto de rotinas e padrões estabelecidos por um software para utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

Árvore hierárquica: é uma forma de representar a natureza hierárquica de uma estrutura em forma gráfica, É chamado de “estrutura de árvore”, porque a representação se assemelha a uma árvore.

Astah Community: é uma ferramenta de modelagem UML gratuita, por ser uma versão *community* possui algumas limitações.

Banco de dados não relacional: Ver “NoSQL”.

Biblioteca padrão: é uma coleção de classes, funções e variáveis escritas na própria linguagem para facilitar o desenvolvimento de aplicações.

BSD: é uma licença de código aberto inicialmente utilizada nos sistemas operacionais do tipo Berkeley Software Distribution (um sistema derivado do Unix).

Campanha: são ações criadas para atingir um objetivo de marketing específico.

Clusterização: é a classificação não-supervisionada de dados, formando agrupamentos ou cluster.

Código aberto: também conhecido por software livre, trata-se de software que respeita as quatro liberdades definidas pela Free Software Foundation, que são usar, copiar, estudar e redistribuir sem restrições.

Código fonte: é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica.

Cream: é um editor de texto livre, disponível como uma extensão para o famoso Vim.

CRM: é um acrônimo de Customer Relationship Management, que pode ser traduzida para o português como gestão de Relacionamento com o Cliente.

CRUD: é um acrônimo de *Create*, *Retrieve*, *Update* e *Delete*, para as quatro operações básicas utilizadas em bancos de dados ou em interface para usuários para criação, consulta, atualização e destruição de dados.

Debian: é o nome de uma distribuição não comercial livre (gratuita e de código fonte aberto) de GNU/Linux (amplamente utilizada) e de um grupo de voluntários que o mantêm à volta do mundo.

Django: é um framework de alto nível escrito em Python que estimula o desenvolvimento rápido e limpo.

Eclipse IDE: é o mais popular ambiente de desenvolvimento integrado de software de código aberto, mantido pela comunidade livre, pode ser estendido com inúmeros plugins.

Engajamento: nível de atração de uma marca ou campanha, quantidade de pessoas que interagem com seu perfil na rede.

EPL (Eclipse Public License): é uma licença de *software open source* usada pela Eclipse Foundation.

Escalável: é o termo usado para descrever a possibilidade de crescimento por motivo de demanda de uso de uma aplicação.

Framework: é uma abstração que une códigos entre vários projetos de *software* provendo uma funcionalidade genérica.

FreeBSD: é um sistema operacional livre do tipo Unix descendente do BSD desenvolvido pela Universidade de Berkeley.

GNU/Linux: é um sistema operacional livre ao estilo Unix, composto por uma série de programas que o deixam interagir com o seu computador

GPL (General Public License): é a designação da licença para software livre com maior utilização na comunidade.

Grid: no contexto do trabalho, *grid* é referenciada como parte da interface do aplicativo, composta por uma tabela paginada com ícones de interação com janelas de adição, edição, visualização e deleção de registros.

Grid computacional: é um serviço para compartilhamento de poder de processamento e capacidade de armazenamento pela Internet, a fim de se otimizar as tarefas realizadas.

Hardware: é a parte física do computador, ou seja, é o conjunto de componentes eletrônicos, circuitos integrados e placas que se comunicam através de barramentos.

Host: consiste em qualquer máquina ou computador conectado a uma rede.

Ferramenta CASE: é uma classificação que abrange todas as ferramentas baseadas em computadores que auxiliam atividades de engenharia de *software*, desde análise de requisitos e modelagem até programação e testes.

Hashtag: são os termos que estão precedidos pelo “#”, esta convenção é utilizada para facilitar o reconhecimento de uma conversa específica.

HTTP (Hypertext Transfer Protocol): é um protocolo de comunicação utilizado para sistemas de informação de *hipermedia* distribuídos e colaborativos.

IDE (Integrated Development Environment): é um ambiente integrado para desenvolvimento de software, ou seja, um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objetivo de agilizar este processo.

Infraestrutura: relacionado a tecnologia o termo infraestrutura abrange redes de computadores, sistemas operacionais, bancos de dados e hardware.

JSON (JavaScript Object Notation): é um formato leve para intercâmbio de dados computacionais. JSON é um subconjunto da notação de um objeto JavaScript, mas seu uso não requer Javascript exclusivamente.

Jude: é uma ferramenta de modelagem UML predecessora do Astah, o Jude esta descontinuado.

Mapa mental: é o nome dado para um tipo de diagrama, voltado para gestão de informações.

Marketing: é o conjunto de técnicas e atividades relacionadas com o fluxo de bens e serviços do produtos para o consumidor.

MongoDB: é uma aplicação de código aberto, de alta performance, sem esquemas, orientado à documentos.

MTV (Model-Template-View): é um padrão de arquitetura de *software*, similar ao MVC, o Django Framework implementa o MTV e define que o “C” de controller fica a cargo do próprio *framework*.

Multiplataforma: é o termo usado para referenciar sistemas capazes de serem executados em diversos sistemas operacionais.

Multiusuário: é o termo usado para definir que um sistema pode ser acessado simultaneamente por múltiplos usuários.

MVC (Model-View-Controller): é um padrão de arquitetura de *software* que visa a separar a lógica de negócio da lógica de apresentação, permitindo o desenvolvimento, teste e manutenção isolado de ambos.

Navegabilidade: é a forma como a informação está organizada, promovendo que o usuário navegue de forma intuitiva e saiba onde está, de onde veio e quais são suas possibilidades futuras de acesso.

Navegador WEB: é o instrumento que permite aos utilizadores da Internet acessar páginas web.

NoSQL: entenda-se “Not only SQL” é um termo genérico para uma classe definida de banco de dados não-relacionais que rompe uma longa história de banco de dados relacionais com propriedades ACID.

Nuvem (Nuvem Computacional): é um modelo de computação em que dados, arquivos e aplicações residem em servidores físicos ou virtuais, acessíveis por meio de uma rede em qualquer dispositivo comparável.

OMG (Object Management Group): é uma organização internacional que aprova padrões abertos para aplicações orientadas a objetos.

Paradigma: é a representação de um padrão a ser seguido.

PEP: é o índice de todas as Python Enhancement Proposals, conhecido como PEPs.

Persistência de dados: significa armazená-los em algum local de modo que possamos recuperá-los posteriormente.

Prospecção: no contexto do projeto, prospecção refere-se à busca de possíveis clientes para o seu produto ou serviço, não de clientes existentes.

Python: é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, de tipagem dinâmica e forte.

Redes sociais: são aplicativos web de comunicação de massa, onde os usuários podem interagir uns com os outros, agrupando-se em comunidades, listas ou círculos.

Release: é uma versão estável de um *software*.

REST (Representational State Transfer): A Transferência de Estado Representacional é uma técnica de engenharia de software para sistemas hipermídia distribuídos. Pretendida como uma imagem do design da aplicação: uma rede de *websites* (um estado virtual), onde o usuário progride com uma aplicação selecionando as ligações (transições do estado), tendo como resultado a página seguinte (que representa o estado seguinte da aplicação) que está sendo transferida ao usuário e apresenta para seu uso.

ROI (Return on Investment): retorno sobre investimento, também chamado de taxa de retorno, é a relação entre dinheiro ganho ou perdido através de um investimento, e o montante de dinheiro investido.

Scrap: é uma mensagem publicada em um perfil de uma rede social.

Script: são partes de código executadas no interior de programas.

Seguidores: São usuários de redes sociais que acompanham determinados assuntos, pessoas ou listas.

Servidores WEB: é um programa de computador responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagem, etc.).

Software: é um produto desenvolvido pela Engenharia de *Software*, e inclui não só o programa de computador propriamente dito, mas também manuais e especificação.

Software livre: é o *software* que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição.

Squeeze: é um codinome para a versão 6 do GNU Debian.

Tagueamento: é a análise dos termos mais usados ou destacados nos conteúdos publicados em redes sociais.

TI (Tecnologia da Informação): é o conjunto de todas as atividades e soluções providas por recursos de computação.

Tooltips: São aquelas informações extras que aparecem ao colocar o ponteiro do mouse sob um link, ou imagem por exemplo.

Trending Topics: é uma lista em tempo real dos nomes mais postados no Twitter pelo mundo todo. Valem para essa lista as hashtags (#) e nomes próprios.

UML (Unified Modeling Language): é a especificação mais usada, e uma forma de modelar não só o estrutura da aplicação, comportamento e arquitetura, mas também de negócios e estrutura de dados.

Usabilidade: é um sinônimo de facilidade de uso. Se um produto é fácil de usar, o usuário tem maior produtividade: aprende mais rápido a usar, memoriza as operações e comete menos erros.

VI: é a sigla para “Visual Interface”. É um editor de texto muito poderoso, a origem desse nome deve-se ao fato de quando o VI foi criado (começo da década de 80), não era comum existirem editores de textos como nos dias de hoje. Naquela época, você digitava um texto mas não podia vê-lo.

VIM (Vi IMproved): é uma versão melhorada do VI, é um dos editores de textos mais utilizados no mundo Unix.

Web-Services: é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes.

WSGI (Web Server Gateway Interface): é uma especificação para comunicação entre servidores de aplicação, ou servidores web, e aplicações web definida na PEP 333. WSGI é um padrão Python e tem como objetivos ser simples e de fácil implementação.

XML: Extensible Markup Language, é uma linguagem de marcação capaz de descrever diversos tipos de dados, seu propósito principal é a facilidade de compartilhamento de informações através da Internet.