

WESLEY GOMES DA SILVA

**PROJETO DE UM SISTEMA DE INFORMAÇÃO DESTINADO A
CONTRATAÇÃO DE PRESTADORES DE SERVIÇOS**

Palmas – TO

2014

Wesley Gomes da Silva

PROJETO DE UM SISTEMA DE INFORMAÇÃO DESTINADO A CONTRATAÇÃO DE PRESTADORES DE SERVIÇOS

Projeto apresentado como requisito parcial para aprovação na disciplina de Estágio Supervisionado II do Curso de Sistemas de Informação, da Faculdade Católica do Tocantins (FACTO), sob a orientação do professor M. Sc. **Marco Antônio Firmino de Sousa.**

Palmas – TO

2014

AGRADECIMENTOS

Agradeço primeiramente a Deus o meu grande guia, por ter me abençoado com a chance de cursar em uma grande faculdade como a CATÓLICA DO TOCANTINS - FACTO e ter recebido subsídios tão ricos durante essa jornada.

Ao professor Marco Antônio Firmino de Sousa pela orientação e motivação para realização deste trabalho, e por exercer tão bem o dom de repassar seus conhecimentos além de ser uma referência aos alunos.

Agradeço em especial a minha família, pai, mãe e irmã, por estar ao meu lado na busca por este sonho e também aos meus amigos que souberam conviver e respeitar ainda que nem sempre compartilhássemos as mesmas ideias. E por tudo, a saudade há de ficar.

Aos meus professores que durante esse tempo que passou transmitiram não só conhecimento teórico, mas lições de vida me engrandecendo como pessoa e profissional.

A todos, agradeço por tudo e muito obrigado!

SUMÁRIO

| | |
|---|----|
| 1. INTRODUÇÃO | 9 |
| 1.1. OBJETIVOS | 10 |
| 1.1.1 Objetivo Geral | 10 |
| 1.1.2 Objetivos Específicos | 10 |
| 1.2 JUSTIFICATIVA..... | 10 |
| 1.3 MOTIVAÇÃO | 10 |
| 1.4 METODOLÓGICA CIENTÍFICA..... | 11 |
| 2 REFERENCIAL TEÓRICO..... | 12 |
| 2.1 MINERAÇÃO DE DADOS..... | 12 |
| 2.2 BANCO DE DADOS..... | 14 |
| 2.3 TRABALHOS RELACIONADOS | 16 |
| 3 DESENVOLVIMENTO | 19 |
| 3.1 REQUISITOS FUNCIONAIS..... | 19 |
| 3.2 REQUISITOS NÃO-FUNCIONAIS..... | 23 |
| 3.3 IMPLEMENTAÇÃO DO BANCO DE DADOS | 28 |
| 3.4 DICIONÁRIO DE DADOS | 30 |
| 3.5 MODELAGEM UML..... | 38 |
| 3.5.1 DIAGRAMA DE ATIVIDADE | 40 |
| 3.5.2 DIAGRAMA DE COMPONENTES..... | 42 |
| 3.5.3 DIAGRAMA DE SEQUÊNCIA | 43 |
| 4 CONSIDERAÇÕES FINAIS | 45 |
| 5 REFERÊNCIA | 46 |
| APÊNDICE A..... | 48 |
| APÊNDICE B | 53 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1 - Etapas que compõem o processo de KDD | 14 |
| Figura 2 - Diagrama simplificado das fases do projeto de um banco de dados | 16 |
| Figura 3 - Interface web do GetNinjas | 19 |
| Figura 4 - Interface web do Recomind.net | 20 |
| Figura 5 - Modelagem do banco de dados | 32 |
| Figura 6 - Diagrama de caso de uso do Cliente | 41 |
| Figura 7 - Diagrama de caso de uso do Prestador de Serviço | 43 |
| Figura 8 - Diagrama de caso de uso do Administrador | 43 |
| Figura 9 - Diagrama de caso de uso do Sistema | 43 |
| Figura 10 - Diagrama de Atividade | 43 |
| Figura 11 - Diagrama de Componentes | 43 |
| Figura 12 - Diagrama de sequência de Cadastro de Cliente | 43 |
| Figura 13 - Diagrama de sequência de Cadastro de Serviço | 43 |

ÍNDICE DE TABELAS

| | |
|--|----|
| Tabela 1 – Requisitos Funcionais | 22 |
| Tabela 2 – Requisitos não-funcionais de Segurança | 26 |
| Tabela 3 – Requisitos não-funcionais de Performance | 27 |
| Tabela 4 – Requisitos não-funcionais de Usabilidade | 28 |
| Tabela 5 – Requisitos não-funcionais de Manutenibilidade | 29 |
| Tabela 6 – Requisitos não-funcionais de Documentação | 30 |
| Tabela 7 – Cidade | 33 |
| Tabela 8 – Uf | 34 |
| Tabela 9 – Tipo_autenticação | 34 |
| Tabela 10 – Endereço | 34 |
| Tabela 11 – Usuário | 35 |
| Tabela 12 – Status_serviço | 36 |
| Tabela 13 – Serviço | 36 |
| Tabela 14 – Situacao_serviço | 37 |
| Tabela 15 – Avaliação_serviço | 38 |
| Tabela 16 – Contrato_serviço_has_serviço | 38 |
| Tabela 17 – Contrato_serviço | 39 |
| Tabela 18 – Serviço_has_categoria_serviço | 39 |
| Tabela 19 – Categoria_serviço | 40 |
| Tabela 20 – Sub_categoria | 40 |

LISTA DE ABREVIATURAS E SIGLAS

IBGE - Instituto Brasileiro de Geografia e Estatística

PIB - Produto interno bruto

CEO - Chief Executive Officer

SBC - Sociedade Brasileira de Computação

SQL - Structured Query Language

KDD - Knowledge Discovery in Databases

SGBD - Sistema de Gerenciamento de Banco de Dados

ER – Entidade Relacionamento

UML – Unified Modeling Language

XHTML – eXtensible Hypertext Markup Language

CRUD - Create, Read, Update e Delete

RESUMO

Este trabalho apresenta o projeto de um sistema destinado a encontrar e facilitar a contratação de um prestador de serviço. Através do projeto proposto pretendemos reunir em um mesmo espaço profissionais liberais como: advogados, programadores, médicos, músicos, designers, confeitores, mecânicos, terapeutas etc. Oferecendo aos usuários acesso a recursos e dados informativos como: avaliações de serviços prestados, localização de um prestador de serviços mais próximo, diminuindo o tempo de espera dos clientes e possibilitando uma melhor contratação do serviço.

Palavra-chave: Busca de serviços, Mineração de dados, Banco de dados.

1. INTRODUÇÃO

As famosas páginas amarelas ajudaram muita gente a encontrar os mais diversos serviços no passado, de pintores e eletricitas a professores particulares e detetives. Hoje quem cumpre esse papel é a internet. Mas como saber se os serviços anunciados são mesmo confiáveis ou de qualidade?

Pensando em resolver este problema, este trabalho propõe o desenvolvimento de uma ferramenta que ajuda o consumidor na hora de encontrar e avaliar diferentes serviços sistematizando todo um processo de contratação de um profissional.

O desafio de inovar em serviços seja nas empresas que os prestam (seguradoras, empresas de tecnologia da informação, laboratórios de análises clínicas, escolas, circos); seja em uma indústria que faz pós-venda ou na oferta de serviços complementares a produtos, a realidade é que a renda gerada em serviços há muito superou a dos produtos em uma economia como a brasileira. Segundo o Instituto Brasileiro de Geografia e Estatística - IBGE, em 2012, a atividade de serviços respondeu por 68,5% do Produto Interno Bruto - PIB e proporcionou mais de 78% dos empregos formais do país.

O objetivo do sistema é ser uma *marketplace* para prestadores de serviços, que facilite a vida de quem está procurando profissionais de qualidade, bons preços e satisfazer uma determinada necessidade ou desejo de um cliente, poupando tempo na contratação de um serviço.

A principal finalidade dessa aplicação é executar funções que, caso um ser humano fosse executar, seriam consideradas inteligentes. Podemos pensar em algumas características básicas desse sistema, como a capacidade de raciocínio (aplicar regras lógicas a um conjunto de dados disponíveis para chegar a uma conclusão), aprendizagem (aprender com os erros e acertos de forma que futuramente possa agir de maneira mais eficaz), reconhecer padrões (padrões de comportamento) e inferência (capacidade de conseguir aplicar o raciocínio nas situações do nosso cotidiano). Um sistema que possua uma ou mais dessas características pode ser considerado um sistema inteligente.

Espera-se para esse sistema que ele possua capacidade de realizar associações de forma que a busca seja a mais adequada à necessidade de quem pesquisa o serviço. Estas

associações possibilitarão que os usuários encontrem o serviço não apenas pelo termo da busca, mas por termos equivalentes como: regionalismos e sinônimos.

1.1. OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver uma proposta de um sistema como meio de auxílio na contratação de um prestador de serviço, capaz de fornecer informações úteis e em tempo real sobre serviços ofertados, qualificações de prestadores de serviços e localização.

1.1.2 Objetivos Específicos

- Realizar um estudo de tecnologias necessárias.
- Levantar e descrever os requisitos funcionais da aplicação;
- Modelar, por meio da UML, a ferramenta proposta;
- Desenvolver o banco de dados do sistema;

1.2 JUSTIFICATIVA

Aplicações inteligentes que nos forneçam informações precisas, rápidas são indispensáveis devido a gestão do tempo cada vez mais difícil, a criação ou melhoramento de ferramentas que nos auxiliem no meio pessoal ou profissional far-se-ão necessárias.

1.3 MOTIVAÇÃO

Melhorar a vida das pessoas com o auxílio da tecnologia, tornando mais fácil a contratação de um serviço, como por exemplo: fornecer prestadores de serviços mais próximos de sua localização, visualização de comentários referentes aos serviços prestados de determinado prestador, auxílio na pesquisa dos serviços. Enfim uma ferramenta que auxiliem na vida das pessoas dispondo de informações úteis e inteligentes de maneira fácil e rápida.

1.4 METODOLÓGICA CIENTÍFICA

A metodologia utilizada para a realização deste trabalho está dividida nas seguintes etapas:

- Pesquisa;
- Modelagem;
- Documentação;

Na etapa de Pesquisa foi realizada uma revisão bibliográfica baseada na leitura de textos disponíveis em livros e artigos técnico-científicos. Realizou-se então uma análise das ferramentas já disponíveis, elaborando tabelas contendo todos os requisitos funcionais e não-funcionais de todos os sistemas analisados incluindo o sistema proposto. Visando determinar um diferencial entre a ferramenta a ser desenvolvida e as ferramentas já existentes, a etapa de pesquisa busca organizar os dados de forma que se possa comparar os sistemas e extrair características interessantes de cada um.

A etapa de Modelagem descreve o projeto do banco de dados do sistema, seu dicionário de dados com o intuito de detalhar minuciosamente cada campo de cada tabela e alguns diagramas em modelo UML (Unified Modeling Language), para caracterização formal do sistema.

A etapa de Documentação é desenvolvida tanto na pesquisa, quanto na modelagem, durante a organização de dados, tabulação de dados, confecção dos diagramas que modela o sistema.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os principais conceitos estudados para o desenvolvimento do trabalho proposto como: mineração de dados e banco de dados. Realiza uma descrição sobre algumas aplicações de mesmo contexto para caracterizar o diferencial do projeto proposto.

2.1 MINERAÇÃO DE DADOS

A área de mineração de dados (Data Mining) surgiu no final da década de oitenta e focaliza a extração de conhecimento a partir de grandes volumes de dados usando computador. A utilização dessa metodologia faz-se necessária para a construção do diferencial da aplicação proposta, proporcionando ao usuário efetuar uma pesquisa não apenas pelo termo explicitado, mas por associações de termos sinônimos, regionalismos e correlacionados.

Na literatura podem ser encontradas diversas características da área, de acordo com Zhou (2003) apud REZENDE, SOLANGE OLIVEIRA (2003), o autor evidencia a caracterização de três perspectivas da área:

1 - Sobre a perspectiva de Banco de Dados, a mineração de dados é "o processo de descoberta de conhecimento interessante em grandes quantidades de dados armazenados em Bases de Dados, *Data Warehouses* ou outros repositórios de dados";

2 - Sobre a perspectiva da área de Aprendizado de Máquina é caracterizada como a "extração de conhecimento implícito, previamente desconhecido e potencialmente útil a partir de dados";

3 - Sobre a perspectiva de área de Estatística é "a análise de conjuntos de dados supervisionados, normalmente em grandes quantidades, para encontrar relacionamentos inesperados e resumir os dados em novas formas que são compreensíveis e úteis para o proprietário dos dados".

Na literatura existem opiniões divergentes a respeito dos termos Mineração de Dados e KDD (Knowledge Discovery in Databases). De acordo com W. Frawley (1992) apud REZENDE, SOLANGE OLIVEIRA (2003), KDD é a "extração de conhecimento

previamente desconhecida, implícita e potencialmente útil, a partir de dados". Existem autores que consideram os termos sinônimos artigo (Fayyad, Piatesky-Shapiro, & Smyth 1996b; Mitchell 1999; Wei 2003) apud REZENDE, SOLANGE OLIVEIRA (2003), enquanto outros consideram a Mineração de Dados apenas um dos passos do processo de KDD, embora seja o passo principal de todo o processo (S. Mitra 2002; I. Sarafis 2002) apud REZENDE, SOLANGE OLIVEIRA (2003).

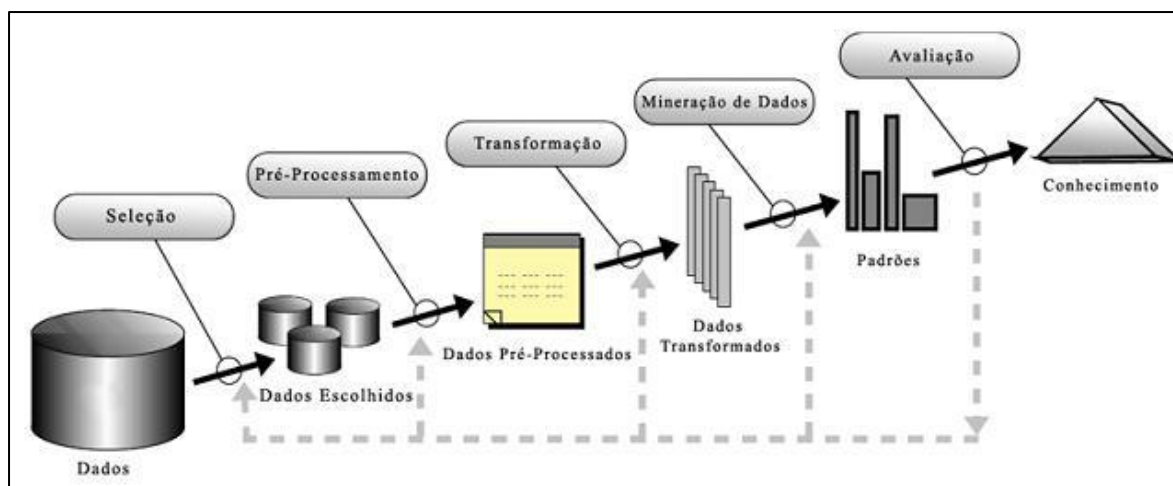


Figura 1 – Etapas que compõem o processo de KDD (Fayyad et al. - 1996).

De acordo com Figura 1, podem se dividir as etapas do KDD da seguinte forma:

- Seleção: é a etapa de agrupamento (organizado) dos dados;
- Pré-processamento: neste momento os dados passam por uma adequação. Ao final do processo, devem possuir o formato correto e não apresentar duplicidade, entre outras características;
- Transformação: é a etapa de armazenamento dos dados de forma a facilitar o uso das técnicas de Data Mining;
- Mineração de Dados: é a principal atividade do conhecimento, aplicando, para este fim, algoritmos de descoberta de padrões;
- Interpretação e avaliação: esta fase consiste em interpretar os dados gerados e verificar se possuem alguma validade para o problema proposto;

2.2 BANCO DE DADOS

Os bancos de dados e suas tecnologias estão provocando um grande impacto no crescimento do uso de computadores. É importante é vital destacar seu papel em quase todas as áreas em que os computadores são utilizados, incluindo negócios, comércio eletrônico, engenharia, medicina, direito, educação e as ciências da informação.

Para (RAMEZ e NAVATHE, 2005), um Sistema de Gerenciamento de Banco de Dados - SGBD é um conjunto de programas de computador que permite aos usuários criar e controlar um banco de dados. Portanto, o SGBD, é um sistema de software com propósito geral que busca facilitar o processo de definição, construção, manipulação e distribuição de dados para vários usuários e aplicativos.

Mas para atender às diversas necessidades do mercado e pesquisas, existem SGBDs com propósitos bem definidos, como para banco de dados espaciais, temporais, multimídias, de tempo real entre outros.

O processo de projeto de banco de dados pode conter algumas fases. Uma dessas fases mais importantes é a modelagem conceitual de banco de dados, pois é nessa fase que se procura elaborar uma visão integrada de todos os dados e processos de uma organização, totalmente independente do sistema computacional que será utilizado para solucionar os problemas dessa organização.

Um modelo de dados conceitual de alto nível bastante popular, o modelo Entidade-Relacionamento - ER, pode ser usado na fase de modelagem conceitual para modelar as estruturas e restrições de um banco de dados.

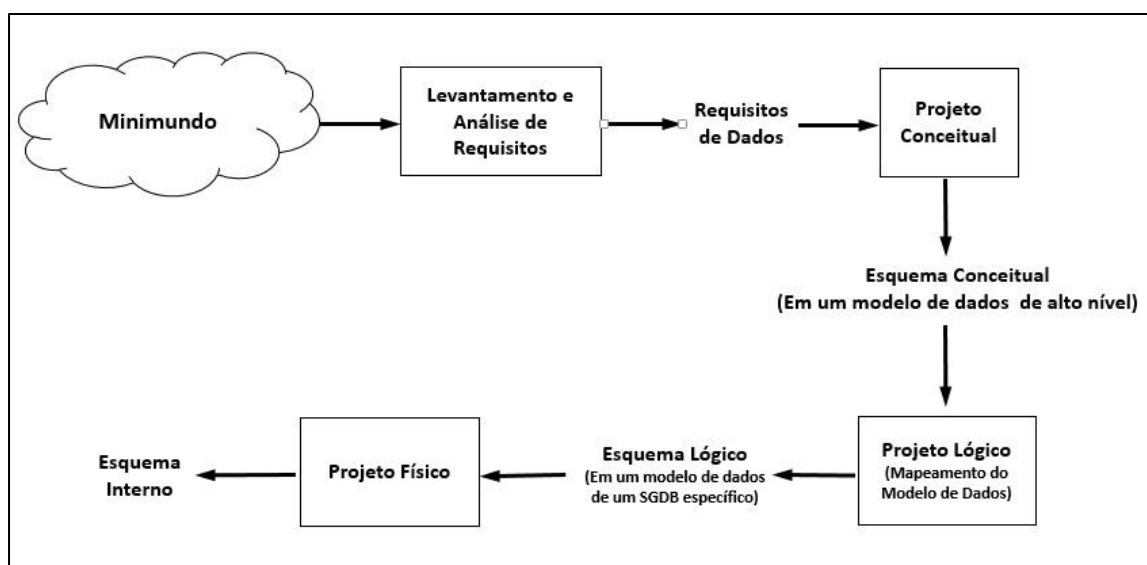


Figura 2 – Diagrama simplificado das fases do projeto de um banco de dados (RAMEZ e NAVATHE, 2005).

Os passos do processo mostrado na Figura 2 são detalhados a seguir:

- O primeiro passo é o levantamento e análise de requisitos. Durante essa fase, o projetista entrevista os possíveis usuários do banco de dados para entender e documentar seus requisitos de dados. O resultado dessa fase é o registro preciso e mais completo possível dos requisitos do usuário.
- Depois que os requisitos foram levantados e analisados, o próximo passo é criar um esquema conceitual para o banco de dados usando um modelo de dados conceitual (ou de alto nível). Essa fase é chamada projeto conceitual. O esquema conceitual é uma descrição dos requisitos de dados dos usuários, expressos através de conceitos fornecidos pelo modelo de dados de alto nível. Esses conceitos não incluem detalhes de implementação em computador, portanto, normalmente, eles são mais fáceis de serem compreendidos e podem ser utilizados na comunicação com os usuários não-técnicos. O esquema conceitual de alto nível também pode ser usado como uma referência para assegurar que todos os requisitos de dados do usuário sejam atendidos. Essa abordagem permite que os projetistas de banco de dados se concentrem na especificação das propriedades do dado, sem se preocupar com os detalhes de armazenamento.

- A próxima fase é a programação real do banco de dados usando um SGBD comercial. A grande maioria dos SGBDs comerciais atuais usa modelo de dados de implementação (modelo relacional ou modelo de banco de dados objeto-relacional) de maneira que o esquema conceitual seja transformado de um modelo de dados de alto nível em um modelo de dados de implementação. Essa fase é definida como projeto lógico.
- O último passo é a fase do projeto físico, durante a qual são definidas as estruturas de armazenamento interno, índices, caminhos de acesso e organização de arquivo para os arquivos do banco de dados.

2.3 TRABALHOS RELACIONADOS

A lista seguinte descreve sistemas relacionados ao mesmo contexto da aplicação proposta:

- **GetNinjas:** No ar desde maio de 2011, nele é possível encontrar profissionais de diferentes ramos. O sistema lucra com uma porcentagem de cada contrato fechado.
- **Iguanafix:** Acesso a todos os detalhes de profissionais liberais ou empresas. Propostas de orçamentos em menos de 24 horas e possibilidade de comparações de preços.
- **Recomind.net:** O Recomind busca facilitar a contratação de profissionais como faxineiras, pedreiros, eletricitas, mecânicos, médicos e cabeleireiras. A aplicação funciona como um caderno de endereços e telefones com características de rede social.
- **Bougue:** O objetivo é fornecer informações para que os usuários que precisam contratar um serviço possam, por exemplo, comparar propostas. Além das propostas, o usuário pode receber também opiniões de quem já usou os serviços do prestador.
- **ClickARQ:** Um site de concorrência criativa em decoração, arquitetura e design de interiores, ajuda a usuários encontrarem arquitetos. Os clientes solicitam projetos de design de interiores e decoração, e arquitetos e designers enviam suas propostas.

Abaixo dois dos sistemas citados com requisitos mais detalhados, os parâmetros de escolha para uma análise mais aprofundada foram: quantidades de usuários ativos, organização de informações, interface intuitiva, plataformas suportadas e prêmios conquistados.

2.3.1 GetNinjas

GetNinjas aplicação web e mobile, com uma plataforma simples e amigável para que pessoas com habilidade de realizar um serviço específico consigam oferecer seu trabalho para o público em geral e, em contrapartida, para que público possa encontrar de maneira rápida, prática e conveniente os serviços que gostaria, é possível encontrar profissionais de diferentes ramos, como fotógrafos, babás, assistentes técnicos e diaristas, fornece informações de acordo com sua região, os prestadores de serviço anunciam gratuitamente no portal, que, em compensação, ficam com uma porcentagem de cada contrato fechado. Até a elaboração desse trabalho, segundo Eduardo L’Hotellier, Chief Executive Officer - CEO do GetNinjas, a plataforma reúne mais de 40 mil profissionais cadastrados em 4.000 cidades brasileiras e movimentou cerca R\$ 8 milhões em negócios para os profissionais desde seu lançamento, o site recebe uma média de 1.000 orçamentos por dia. Só no Estado de São Paulo, por exemplo, são aproximadamente 10 mil orçamentos por mês. Conquistou prêmios como Startup do ano de 2012 promovido pela TheNextWeb Startup e o reconhecimento do Jornal americano The New Work Times em 2011. Sua interface gráfica está observada na Figura 3.



Figura 3. Interface web do GetNinjas (GETNINJAS - 2011).

2.3.2 Recomind.net

Recomind.net aplicação web e mobile, apresenta produtos e serviços baseado nas indicações da rede de contatos do usuário, conta com a integração com o facebook, através dele o usuário tem um espaço para preencher as informações sobre o profissional ou serviço que procura. A partir disso, a requisição ficará listada para que outros usuários possam respondê-la. Além disso, o usuário terá a opção de postar as informações no mural de sua rede social, para que seus amigos possam indicar um prestador de serviço que se adeque as suas necessidades.

Um serviço Multiplataforma roda em tablets, smartphones Android, Iphone e computadores. Um dos vencedores da edição 2011 do Desafio Buscapé: Sua Ideia Vale Um Milhão. O concurso premiou a startup com um investimento de R\$ 300 mil, atualmente a aplicação pertence ao grupo Buscapé company. A Figura 4 apresenta sua interface gráfica.



Figura 4. Interface web do Recomind.net (RECOMIND.NET - 2011).

3 DESENVOLVIMENTO

Este capítulo apresenta a implementação do sistema proposto relacionando os conceitos e teorias que o fundamentam. Métodos e ferramentas que foram utilizadas no desenvolvimento da proposta são brevemente descritos.

3.1 REQUISITOS FUNCIONAIS

Segundo Somerville (2011, p. 59), requisitos funcionais são declarações de funções que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Com isso, abaixo uma tabela contendo todos os requisitos funcionais dos sistemas relacionados e do sistema proposto.

Tabela 1 – Requisitos Funcionais

| ID | Requisitos | Software Proposto | GetNinjas | Iguanafix | Recomind.net | Bougue | ClickARQ |
|-----------|---|--------------------------|------------------|------------------|---------------------|---------------|-----------------|
| RFC0001 | Cadastrar Usuário | X | X | X | X | X | X |
| RFC0002 | Cadastrar Serviço | X | X | X | X | X | X |
| RFC0003 | Cadastrar Empresa | - | - | X | - | - | - |
| RFC0004 | Buscar Serviço | X | X | X | X | X | X |
| RFC0005 | Alterar Perfil de Usuário | X | X | X | X | X | X |
| RFC0006 | Alterar Perfil de Serviço Prestado | X | X | X | X | X | X |
| RFC0007 | Remover Usuário | X | X | X | X | X | X |
| RFC0008 | Remover Serviço | X | X | X | X | X | X |
| RFC0009 | Localização do Prestador de Serviço mais próximo. | X | - | - | X | - | - |
| RFC0010 | Avaliação do Serviço Prestado | X | X | X | X | X | X |
| RFC0011 | Suporte a Dúvidas | X | X | X | X | X | X |
| RFC0012 | Autenticação com Facebook | X | X | X | X | X | - |
| RFC0013 | Gamificação de Avaliação | X | X | X | X | X | - |

Para um melhor entendimento da Tabela 1, é feito uma descrição mais detalhada dos requisitos funcionais:

- **RFC0001 - Cadastrar Usuário:** Requisito responsável por cadastrar todos os usuários que solicitem e/ou que pretendem ofertar algum tipo de serviço, essa condição é utilizada em todos os sistemas abordados.
- **RFC0002 - Cadastrar Serviço:** Requisito responsável por cadastrar todos os serviços prestados no sistema, este quesito é adotado por todos os sistemas levantados, entretanto para o sistema Iguanafix é necessário primeiro a execução do requisito RFC0003.
- **RFC0003 - Cadastrar Empresa:** Requisito responsável por cadastrar uma empresa prestadora de serviços, essa condição é utilizada somente no sistema Iguanafix pelo fato do sistema fazer uma avaliação do prestador de serviço antes de sua liberação, com isso a empresa prestadora de serviço garante algumas vantagens como: ter uma lista segura de profissionais certificados, gerando mais confiança aos clientes.
- **RFC0004 - Buscar Serviço:** Requisito responsável por buscar qualquer tipo de serviço no sistema como: pesquisar um serviço de eletricista, um pedreiro, uma diarista, este item é utilizado por todos os sistemas abordados.
- **RFC0005 - Alterar Perfil de Usuário:** Requisito responsável por fazer alterações no perfil de usuários como: alterar nome, trocar foto, alterar endereço, este quesito é utilizado por todos os sistemas levantados.
- **RFC0006 - Alterar Perfil de Serviço Prestado:** Requisito responsável por fazer alterações no serviço ofertado como: alterar o nome do serviço, alterar o valor do serviço, este quesito é adotado por todos os sistemas levantados, entretanto para o sistema Iguanafix é necessário primeiro a execução do requisito RFC0003.
- **RFC0007 - Remover Usuário:** Requisito responsável por remover qualquer usuário devidamente cadastrado no sistema, onde o próprio usuário poderá excluir sua conta, excluindo todos os dados do mesmo, esta condição é aplicada por todos os sistemas abordados.
- **RFC0008 - Remover Serviço:** Requisito responsável por remover qualquer serviço devidamente cadastrado no sistema, o próprio prestador de serviço poderá excluir o serviço, este quesito é adotado por todos os sistemas levantados.

- **RFC0009 - Localização do Prestador de Serviço mais próximo:** Requisito responsável por exibir a localização exata do prestador de serviço mais próximo do usuário que efetuou a pesquisa, esta condição é aplicada apenas no sistema Recomend.net, o desenvolvimento desse requisito faz parte do diferencial do projeto proposto, devido ao grau de detalhamento que exibirá as informações.
- **RFC0010 - Avaliação do Serviço Prestado:** Requisito responsável por avaliar o serviço prestado através de comentários, onde após o serviço pronto, o usuário solicitante poderá fazer algum comentário a respeito do mesmo, este quesito é abordado por todos os sistemas levantados.
- **RFC0011 - Suporte a Dúvidas:** Requisito responsável por dar suporte a todos os usuários, através de e-mails e/ou telefone, esse item é utilizado por todos os sistemas abordados.
- **RFC0012 - Autenticação com Facebook:** Requisito responsável por agilizar o processo de autenticação de usuários através do facebook, é compartilhar informações através da rede social, quesito que apenas o sistema ClickARQ não utiliza, pelo fato que os usuários deverão fornecer informações pessoais como nome, endereço, número de telefone, bem como outros dados pessoais relativos aos meios de pagamento utilizados, como: número de cartão de crédito, todas as informações pessoais fornecidas voluntariamente ou através de procedimentos automatizados estão sujeitas a medidas de segurança para impedir acesso, uso e divulgação não autorizados, estão sujeitas às disposições presentes na Política de Privacidade do Site.
- **RFC0013 - Gamificação de Avaliação:** Requisito responsável por gamificar a forma de avaliar os prestadores de serviços como: avaliações de classificação com estrelas para os melhores serviços prestados, quesito que apenas o sistema ClickARQ não utiliza, fato de não ter uma funcionalidade diferente do requisito RFC0010, o desenvolvimento desse requisito faz parte do diferencial do projeto proposto onde os prestadores que obtiverem mais avaliações positivas desse tipo terá vantagens especiais como: exibição na página principal da aplicação por mais tempo, descontos exclusivos na utilização do sistema.

3.2 REQUISITOS NÃO-FUNCIONAIS

Além dos requisitos funcionais existem aqueles que não têm relação direta com as funções do software. Suas relações podem estar ligadas a propriedades da organização, segurança, desempenho, facilidades de uso e outras. Esses são os requisitos não-funcionais. Tais requisitos também são definidos por Sommerville (2011, p. 59), ao qual diz que são restrições sobre os serviços ou as funções oferecidas pelo sistema. Entre eles destacam-se restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, entre outros. Para tal, foram identificados requisitos não-funcionais, logo abaixo tabelas representando todos os requisitos dos softwares analisados e da aplicação proposta.

Tabela 2 – Requisitos não-funcionais de Segurança

| ID | Requisitos | Software Proposto | GetNinjas | Iguanafix | Recomind.net | Bougue | Click ARQ |
|------------|--|-------------------|-----------|-----------|--------------|--------|-----------|
| RNF/SEG-01 | O sistema deverá restringir o acesso às informações pessoais dos usuários é prestadores de serviços cadastrados. | X | X | X | X | X | X |
| RNF/SEG-02 | Utilizar técnicas de backup de dados. | X | X | X | X | X | X |
| RNF/SEG-03 | Utilizar técnicas de restauração de dados em caso de falhas. | X | X | X | X | X | X |
| RNF/SEG-04 | Utilizar mecanismos de proteção contra escrita, que previnam alterações indevidas e mantenham a integridade dos dados armazenados. | X | X | X | X | X | X |

Para um melhor entendimento da Tabela 2, é feita uma descrição detalhada dos requisitos de segurança:

- **RNF/SEG-01:** Requisito responsável por restringir o acesso a pessoas não autorizadas a informações pessoais de usuários como: nome, CPF, RG, endereço, telefone, etc., através de privilégios e métodos para o bloqueio ao sistema, esse item é utilizado por todos os sistemas abordados.
- **RNF/SEG-02:** Requisito responsável por executar o backup semanal de todos os dados do sistema em algum servidor diferente do que hospeda a aplicação, de preferência em um local externo a organização, esse quesito é abordado por todos os sistemas levantados.
- **RNF/SEG-03:** Requisito responsável por garantir que os dados sejam restaurados em casos de falhas técnicas e roubos de servidores, essa condição é utilizada por todos os sistemas abordados.
- **RNF/SEG-04:** Requisito responsável por proteger os dados de alterações indevidas por pessoas não autorizadas, esse quesito é utilizado por todas as aplicações abordadas.

Tabela 3 – Requisitos não-funcionais de Performance

| ID | Requisitos | Software Proposto | GetNinjas | Iguanafix | Recomind.net | Bougue | Click ARQ |
|------------|---|-------------------|-----------|-----------|--------------|--------|-----------|
| RNF/PER-05 | O banco de dados deve ser atualizado em tempo real. | X | X | X | X | X | X |
| RNF/PER-06 | O tempo de retorno das consultas (isto é, o intervalo de tempo entre qualquer consulta e seu resultado) não pode ser maior do que 7 segundos. | X | - | - | X | - | - |
| RNF/PER-07 | O sistema deve permitir acesso de qualquer local que disponha de um computador com acesso à Internet. | X | X | X | X | X | X |

Para um melhor entendimento da Tabela 3, é feito uma descrição detalhada dos requisitos de performance:

- **RNF/PER-05:** Requisito responsável pela atualização em tempo real dos dados cadastrado no banco de dados, esse quesito é utilizado por todos os softwares abordados.
- **RNF/PER-06:** Requisito responsável por exibir o resultado de uma busca com o tempo inferior a 7 segundos, evitando assim transtorno para usuários que aguardam por um resultado, está condição e aplicada apenas no sistema Recomind.net, o desenvolvimento desse requisito faz parte do diferencial da aplicação proposta, devido a forma de implementação do banco de dados, garantindo rapidez em consultas complexas.
- **RNF/PER-07:** Por se tratar de uma aplicação totalmente web, este requisito é responsável pelo acesso ao sistema de qualquer computador com acesso a internet, essa condição é utilizada em todas as aplicações abordadas.

Tabela 4 – Requisitos não-funcionais de Usabilidade

| ID | Requisitos | Software Proposto | GetNinjas | Iguanafix | Recomind.net | Bougue | Click ARQ |
|------------|--|-------------------|-----------|-----------|--------------|--------|-----------|
| RNF/USA-08 | O sistema será operado em ambiente totalmente web. | X | X | X | X | X | X |
| RNF/USA-09 | O sistema usará uma interface intuitiva que permitirá a utilização do sistema com toda sua potencialidade em um curto espaço de tempo e beneficiará o trabalho dos usuários. | X | X | X | X | X | - |
| RNF/USA-10 | As mensagens de erros devem ser claras e concisas para o entendimento perfeito do usuário do sistema. | X | X | X | X | X | X |

Para um melhor entendimento da Tabela 4, é feita uma descrição detalhada dos requisitos de usabilidade:

- **RNF/USA-08:** Requisito garante que a aplicação seja operada somente em ambiente web, essa condição é utilizada em todos os sistemas levantados.
- **RNF/USA-09:** Requisito responsável pela utilização de interfaces intuitiva, garantindo que o usuário entenda todos os passos para uma boa utilização do sistema, potencializando todas as vantagens que beneficia os usuários, esse item é utilizado por todos os sistemas abordados exceto o ClickARQ, análise feita na usabilidade dessa aplicação constatou uma má usabilidade das interfaces para o processo de sua utilização.
- **RNF/USA-10:** Requisito responsável por exibir de forma clara para os usuários, mensagens de erros como: preenchimento indevido de campos, essa condição é utilizada em todos os sistemas abordados.

Tabela 5 – Requisitos não-funcionais de Manutenabilidade

| ID | Requisitos | Software Proposto | GetNinjas | Iguanafix | Recomind.net | Bougue | Click ARQ |
|------------|---|-------------------|-----------|-----------|--------------|--------|-----------|
| RNF/MAN-11 | O sistema será dividido em módulos, de modo que as atualizações serão feitas mais rapidamente e sem a necessidade de longos períodos de atualização onde o sistema ficaria impossibilitado de operar. | X | X | X | X | X | X |

Para um melhor entendimento da Tabela 5, é feito uma descrição detalhada do requisito de manutenabilidade:

- **RNF/MAN-11:** Requisito responsável por modularizar a aplicação, de modo que as atualizações futuras possam ser feitas apenas naquele modulo que a necessite, evitando que a aplicação fique longo período impossibilitado de operar, essa condição é utilizada em todos os sistemas levantados.

Tabela 6 – Requisitos não-funcionais de Documentação

| ID | Requisitos | Software Proposto | GetNinjas | Iguanafix | Recomind.net | Bougue | Click ARQ |
|------------|--|-------------------|-----------|-----------|--------------|--------|-----------|
| RNF/DOC-12 | O sistema possuir um manual de uso a fim de auxiliar os diferentes tipos de usuário. O mesmo explicará detalhadamente como proceder na realização das funções requisitadas para a aplicação. | X | X | X | X | X | X |
| RNF/DOC-13 | O sistema possuir vídeos demonstrativos do funcionamento da ferramenta. | X | X | - | - | - | X |

Para um melhor entendimento da Tabela 6, é feito uma descrição detalhada dos requisitos de documentação:

- **RNF/DOC-12:** Requisito responsável por disponibilizar um documento do software com todas as informações possíveis como: tutoriais de cadastros e consultas, preços de utilização, informações de contato, etc., de forma a maximizar o uso da ferramenta web, essa condição é utilizada em todos os sistemas levantados.
- **RNF/DOC-13:** Requisito responsável por exibir vídeos informativos da ferramenta, para usuários leigos, buscando auxiliar na utilização da ferramenta, a análise de usabilidade constatou que essa condição é utilizada apenas nos sistemas: GetNinjas e ClickARQ, o desenvolvimento desse requisito faz parte do diferencial do projeto proposto, devido a uma forma diferente de demonstrar o funcionamento da aplicação através de vídeos interativos.

3.3 IMPLEMENTAÇÃO DO BANCO DE DADOS

Em geral quando se projeta um banco de dados relacional buscamos estabelecer relações entre entidades, como tabelas, de modo que nos permita armazenar dados sem redundância desnecessária, além de permitir recuperar os dados com pouco custo. Com base nas pesquisas feitas anteriormente, constatou-se que o MySQL é a melhor escolha, pois contém todos os requisitos necessários ao desenvolvimento do aplicativo proposto.

De acordo com (RAMEZ e NAVATHE, 2005) as vantagens na utilização de um padrão para a modelagem de dados é extremamente necessária. A criação de um modelo único faz-se necessário seguindo os seguintes critérios:

- Tabelas serão nomeadas sempre no singular, sem acentos e no lugar do espaço será usado o caractere underline (_) entre os nomes das mesmas, sendo que as letras serão minúsculas;
- Todas as tabelas terão uma chave primária com auto incremento para identificar um registro específico;
- Os relacionamentos estabelecidos com chaves estrangeiras serão nomeados com o prefixo “fk” seguido de “_” e o nome da chave primária da outra tabela concatenado com nome da tabela relacionada;

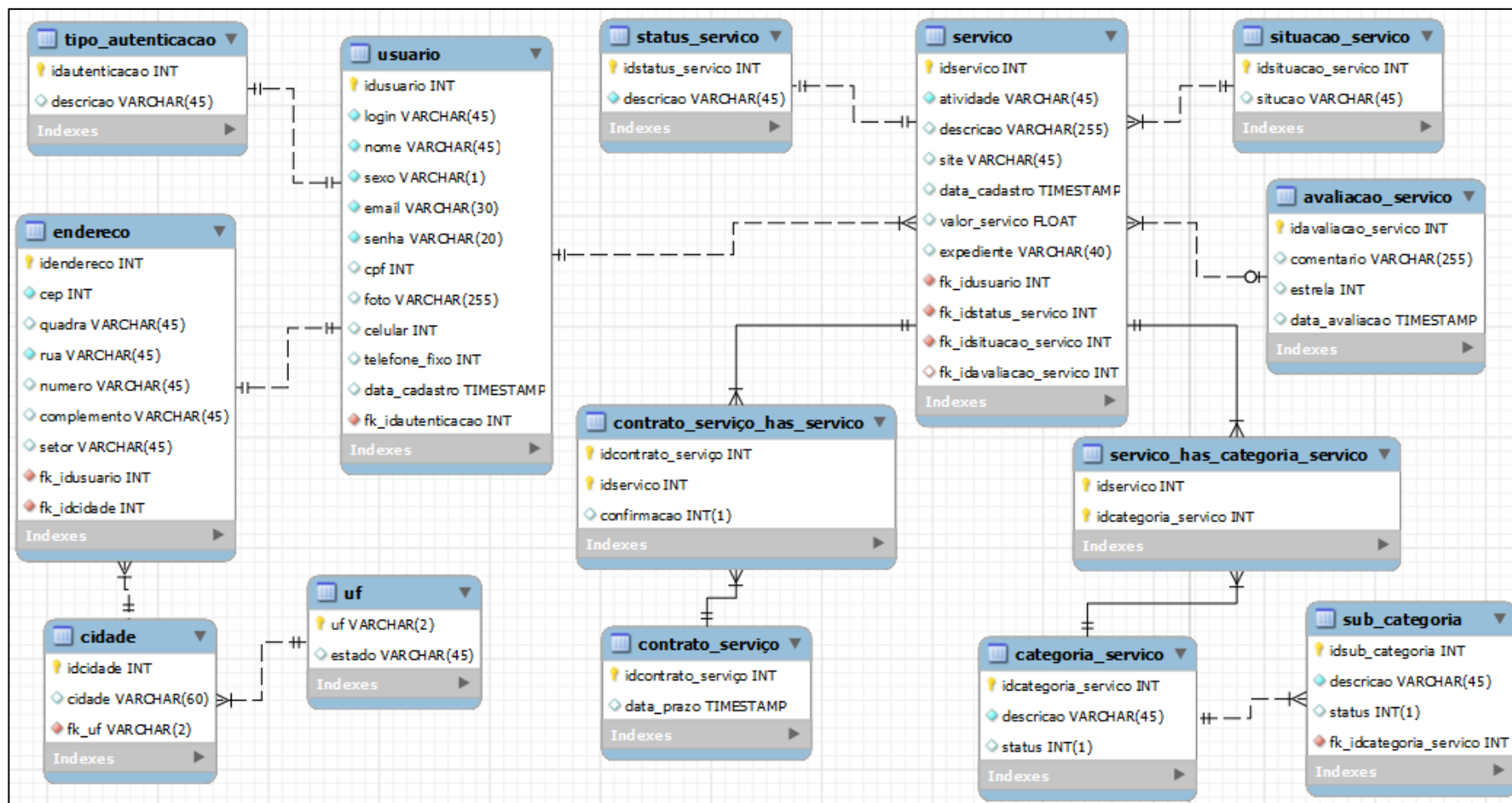


Figura 5. Modelagem do banco de dados.

A Figura 5 ilustra a modelagem do banco de dados para um entendimento técnico, para uma melhor compreensão humana sobre como os dados estão relacionados é necessário que se crie um dicionário de dados, o tópico 3.2 detalha cada tabela do banco de dados de acordo com os requisitos descritos na seção 2.4 e 2.5.

3.4 DICIONÁRIO DE DADOS

Dentro do contexto de SGBD, um dicionário de dados é um grupo de tabelas, habilitadas apenas para leitura ou consulta, ou seja, é uma base de dados, propriamente dita, que entre outras coisas, mantém as seguintes informações:

- Definição precisa sobre elementos de dados;
- Descrição de objetos;
- Restrições de integridade;
- Estrutura geral da base de dados;
- Informação de verificação;
- Alocações de espaço;
- Índices;

Abaixo a descrição de cada tabela contendo o dicionário de dados.

| Tabela 7: cidade | | | | |
|---|-------------|-------------|----------------|---|
| Chave Primária | Nome | Tipo | Tamanho | Descrição |
| PK | idcidade | INT | - | Chave primaria da tabela cidade, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | cidade | VARCHAR | 60 | Campo responsável por armazenar o nome da cidade. |
| FK | fk_uf | INT | 2 | Campo chave estrangeira tem finalidade de pegar o id do estado que está na tabela de uf. |
| Obs.: essa tabela surgiu através do requisito RFC0002 descrito na seção 2.4. | | | | |

A Tabela 7 descreve de forma detalhada todos os campos da tabela cidade do banco de dados é relaciona com o requisito que a implementa.

| Tabela 8: uf | | | | |
|---|-------------|-------------|----------------|--|
| Chave Primária | Nome | Tipo | Tamanho | Descrição |
| PK | iduf | VARCHAR | 2 | Chave primaria da tabela uf, tem por finalidade identificar a posição dos dados cadastrados. |
| | estado | VARCHAR | 45 | Campo responsável por armazenar o nome do estado. |
| Obs.: essa tabela surgiu através do requisito RFC0002 descrito na seção 2.4. | | | | |

A Tabela 8 descreve de forma detalhada todos os campos da tabela uf do banco de dados é relaciona com o requisito que a implementa.

| Tabela 9: tipo_autenticacao | | | | |
|--|----------------|-------------|----------------|--|
| Chave Primária | Nome | Tipo | Tamanho | Descrição |
| PK | idautenticacao | INT | - | Chave primaria da tabela tipo_autenticacao, tem por finalidade identificar a posição dos dados cadastrados. |
| | descricao | VARCHAR | 45 | Campo responsável por armazenar o nome do tipo de autenticação no sistema como exemplo: facebook, twitter e local. |
| Obs.: essa tabela surgiu através do requisito RFC00012 descrito na seção 2.4. | | | | |

A Tabela 9 descreve de forma detalhada todos os campos da tabela tipo_autenticacao do banco de dados é relaciona com o requisito que a implementa.

| Tabela 10: endereco | | | | |
|----------------------------|-------------|-------------|----------------|---|
| Chave Primária | Nome | Tipo | Tamanho | Descrição |
| PK | idendereco | INT | - | Chave primaria da tabela endereco, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | cep | INT | - | Campo responsável por armazenar o cep do endereço. |
| | quadra | VARCHAR | 45 | Campo responsável por armazenar a quadra do endereço. |
| | rua | VARCHAR | 45 | Campo responsável por armazenar a rua do endereço. |
| | numero | VARCHAR | 45 | Campo responsável por armazenar o numero do endereço. |
| | complemento | VARCHAR | 45 | Campo responsável por armazenar |

| | | | | |
|---|--------------|---------|----|--|
| | | | | algum ponto de referência, para fácil identificação do seu endereço. |
| | setor | VARCHAR | 45 | Campo responsável por armazenar o setor do endereço. |
| FK | fk_idusuario | INT | - | Campo chave estrangeira contem a referência do id da tabela usuário. |
| FK | fk_idcidade | INT | - | Campo chave estrangeira contem a referência do id da tabela cidade. |
| Obs.: essa tabela surgiu através do requisito RFC0002 descrito na seção 2.4. | | | | |

A Tabela **10** descreve de forma detalhada todos os campos da tabela endereço do banco de dados é relaciona com o requisito que o implementa.

| Tabela 11: usuario | | | | |
|---------------------------|---------------|-------------|----------------|--|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idusuario | INT | - | Chave primaria da tabela usuario, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | login | VARCHAR | 45 | Campo responsável por armazenar o login de um usuário, necessário para ser identificado no sistema e ter acesso as funcionalidades do mesmo. |
| | nome | VARCHAR | 45 | Campo responsável por armazenar o nome do cliente. |
| | sexo | VARCHAR | 1 | Campo responsável por armazenar o sexo da pessoa apenas com uma letra M para masculino ou F para feminino. |
| | email | VARCHAR | 30 | Campo responsável por armazenar o email do cliente. |
| | senha | VARCHAR | 20 | Campo responsável por armazenar a senha do usuário, para ter acesso ao sistema. |
| | cpf | INT | - | Campo responsável por armazenar o cpf do prestador de serviço. |
| | foto | VARCHAR | 255 | Campo responsável por armazenar o caminho da foto do perfil do prestador de serviço. |
| | celular | INT | - | Campo responsável por armazenar o celular do prestador de serviço. |
| | telefone_fixo | INT | - | Campo responsável por armazenar o telefone fixo do prestador de serviço. |
| | data_cadastro | TIMESTAMP | - | Campo responsável por armazenar a data do cadastro do usuário. O campo Default está definido como |

| | | | | |
|--|-------------------|-----|---|--|
| | | | | now(), método responsável por pegar a data e hora exata do cadastro. |
| FK | fk_idautenticacao | INT | - | Campo que contem a referência do id da tabela tipo_autenticacao. |
| Obs.: essa tabela surgiu através dos requisitos RFC0001, RFC0002 e RFC0012 descrito na seção 2.4. | | | | |

A Tabela **11** descreve de forma detalhada todos os campos da tabela usuário do banco de dados é relaciona com os requisitos que o implementam.

| Tabela 12: status_servico | | | | |
|---|------------------|-------------|----------------|---|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idstatus_servico | INT | - | Chave primaria da tabela status_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | descricao | VARCHAR | 45 | Campo responsável por armazenar qual o status se encontra o serviço ofertado. |
| Obs.: essa tabela surgiu através do requisito RFC0004 descrito na seção 2.4. | | | | |

A Tabela **12** descreve de forma detalhada todos os campos da tabela status_servico do banco de dados é relaciona com o requisito que o implementa.

| Tabela 13: servico | | | | |
|---------------------------|---------------|-------------|----------------|--|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idservico | INT | - | Chave primaria da tabela servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | atividade | VARCHAR | 45 | Campo responsável por armazenar o nome do serviço ofertado. |
| | descricao | VARCHAR | 255 | Campo responsável por armazenar uma descrição do serviço informado. |
| | site | VARCHAR | 45 | Campo responsável por armazenar o endereço virtual do prestador de serviço como: site, facebook, twitter etc. |
| | data_cadastro | TIMESTAMP | - | Campo responsável por |

| | | | | |
|---|------------------------|---------|----|--|
| | | | | armazenar a data do cadastro do serviço. Default está definido como now(), método responsável por pegar a data e hora exata do cadastro. |
| | valor_servico | FLOAT | - | Campo responsável por armazenar o valor do serviço ofertado. |
| | expediente | VARCHAR | 40 | Campo responsável por armazenar os dias disponíveis para realizar o serviço ofertado. |
| FK | fk_idusuario | INT | - | Campo que contem a referência do id da tabela usuario. |
| FK | fk_status_servico | INT | - | Campo que contem a referência do id da tabela status_servico. |
| FK | fk_idsituacao_servico | INT | - | Campo que contem a referência do id da tabela situação_servico. |
| FK | fk_idavaliacao_servico | INT | - | Campo que contem a referência do id da tabela avaliacao_servico. |
| Obs.: essa tabela surgiu através do requisito RFC0002 descrito na seção 2.4. | | | | |

A Tabela **13** descreve de forma detalhada todos os campos da tabela serviço do banco de dados é relaciona com o requisito que o implementa.

| Tabela 14: situacao_servico | | | | |
|---|--------------------|-------------|----------------|---|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idsituacao_servico | INT | - | Chave primaria da tabela situacao_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | situacao | VARCHAR | 45 | Campo responsável por armazenar qual a situação que o serviço se encontra exemplo: em aberto, finalizado, em andamento. |
| Obs.: essa tabela surgiu através do requisito RFC0010 descrito na seção 2.4. | | | | |

A Tabela **14** descreve de forma detalhada todos os campos da tabela situação_servico do banco de dados é relaciona com o requisito que o implementa.

| Tabela 15: avaliacao_servico | | | | |
|-------------------------------------|---------------------|-------------|----------------|--|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idavaliacao_servico | INT | - | Chave primaria da tabela avaliacao_servico, autoincremento |

| | | | | |
|---|----------------|-----------|-----|---|
| | | | | tem por finalidade identificar a posição dos dados cadastrados. |
| | comentario | VARCHAR | 255 | Campo responsável por armazenar comentários sobre um determinado serviço. |
| | estrela | INT | - | Campo responsável por armazenar estrelas como outra forma de avaliação. |
| | data_avaliacao | TIMESTAMP | - | Campo responsável por armazenar a data do cadastro da avaliação do serviço. Default está definido como now(), método responsável por pegar a data e hora exata do cadastro. |
| Obs.: essa tabela surgiu através dos requisitos RFC0013 e RFC0010 descrito na seção 2.4. | | | | |

A Tabela **15** descreve de forma detalhada todos os campos da tabela avaliação_servico do banco de dados e relaciona com os requisitos que o implementam.

| Tabela 16: contrato_servico_has_servico | | | | |
|---|--------------------|-------------|----------------|---|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idcontrato_servico | INT | - | Chave primaria da tabela contrato_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| PK | idservico | INT | - | Chave primaria da tabela servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | confirmacao | INT | 1 | Campo responsável por armazenar a situação do contrato de serviço. |
| Obs.: essa tabela surgiu através da ligação de muitos para muitos entre a tabela contrato_servico e servico. | | | | |

A Tabela **16** descreve de forma detalhada todos os campos da tabela contrato_servico_has_servico do banco de dados.

| Tabela 17: contrato_servico | | | | |
|------------------------------------|--------------------|-------------|----------------|---|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idcontrato_servico | INT | - | Chave primaria da tabela contrato_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |

| | | | | |
|--|------------|-----------|---|--|
| | data_prazo | TIMESTAMP | - | Campo responsável por armazenar a data do contrato do serviço. Default está definido como now(), método responsável por pegar a data e hora exata do cadastro. |
| Obs.: essa tabela surgiu através do requisito RFC004 descrito na seção 2.4. | | | | |

A Tabela **17** descreve de forma detalhada todos os campos da tabela contrato_servico do banco de dados e relaciona com o requisito que o implementa.

| Tabela 18: servico_has_categoria_servico | | | | |
|--|---------------------|-------------|----------------|--|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idservico | INT | - | Chave primaria da tabela servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| PK | idcategoria_servico | INT | - | Chave primaria da tabela categoria_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| Obs.: essa tabela surgiu através da ligação de muitos para muitos entre a tabela servico e categoria_servico. | | | | |

A Tabela **18** descreve de forma detalhada todos os campos da tabela servico_has_categoria_servico do banco de dados.

| Tabela 19: categoria_servico | | | | |
|-------------------------------------|---------------------|-------------|----------------|--|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idcategoria_servico | INT | - | Chave primaria da tabela categoria_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | descicao | VARCHAR | 45 | Campo responsável por armazenar uma descrição da categoria do serviço. |
| | status | INT | 1 | Campo responsável por armazenar |

| | | | | |
|--|--|--|--|--|
| | | | | um status da categoria do serviço, informação necessária para administradores do sistema, para que possa atribuir categorias aos serviços cadastrados. |
| Obs.: essa tabela surgiu através do requisito RFC002 descrito na seção 2.4. | | | | |

A Tabela **19** descreve de forma detalhada todos os campos da tabela categoria_servico do banco de dados é relaciona com o requisito que o implementa.

| Tabela 20: sub_categoria | | | | |
|--|------------------------|-------------|----------------|--|
| Chave Primaria | Nome | Tipo | Tamanho | Descrição |
| PK | idsub_categoria | INT | - | Chave primaria da tabela sub_categoria, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| | descricao | VARCHAR | 45 | Campo responsável por armazenar uma descrição da sub categoria do serviço cadastrado. |
| | status | INT | 1 | Campo responsável por armazenar um status da categoria do serviço, informação necessária para administradores do sistema, para que possa atribuir categorias aos serviços cadastrados. |
| FK | fk_idcategoria_servico | INT | - | Chave primaria da tabela categoria_servico, autoincremento tem por finalidade identificar a posição dos dados cadastrados. |
| Obs.: essa tabela surgiu através do requisito RFC002 descrito na seção 2.4. | | | | |

A Tabela **20** descreve de forma detalhada todos os campos da tabela sub_categoria do banco de dados é relaciona com o requisito que a implementa.

3.5 MODELAGEM UML

Segundo Booch, G. and Rumbaugh, J. and Jacobson - (2006), UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas.

Os diagramas como parte da modelagem de um sistema de informação, ilustram graficamente a arquitetura, estrutura, interações e comportamento do sistema proporcionando uma abstração de vários contextos inerentes ao software.

Afim de demonstrar graficamente a estrutura dos requisitos do software, abaixo seguem os diagramas contendo os casos de uso, atividade, componente e sequência.

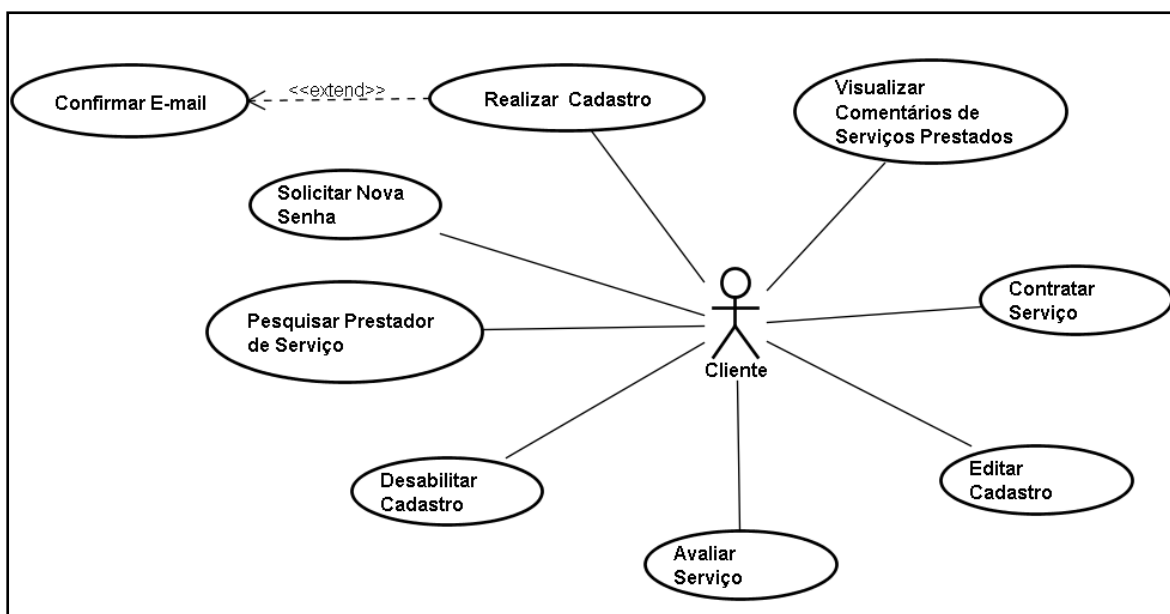


Figura 6. Diagrama de caso de uso do Cliente.

Os casos de uso expandidos dos casos de usos mostrados no diagrama da Figura 6 estão presentes no apêndice A.

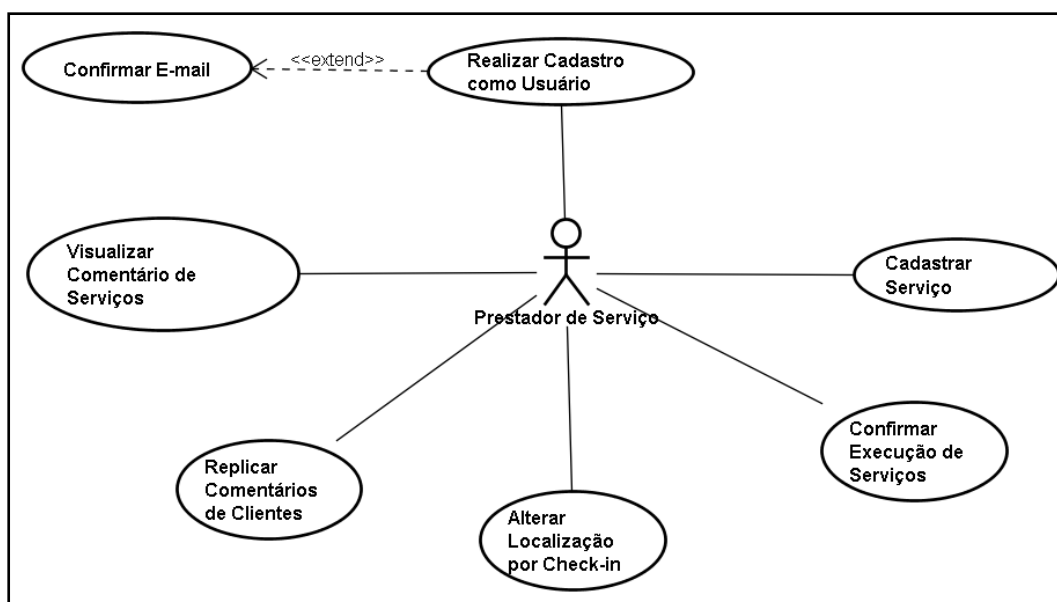


Figura 7. Diagrama de caso de uso do Prestador de Serviço.

Os casos de uso expandidos dos casos de usos mostrados no diagrama da Figura 7 estão presentes no apêndice A.

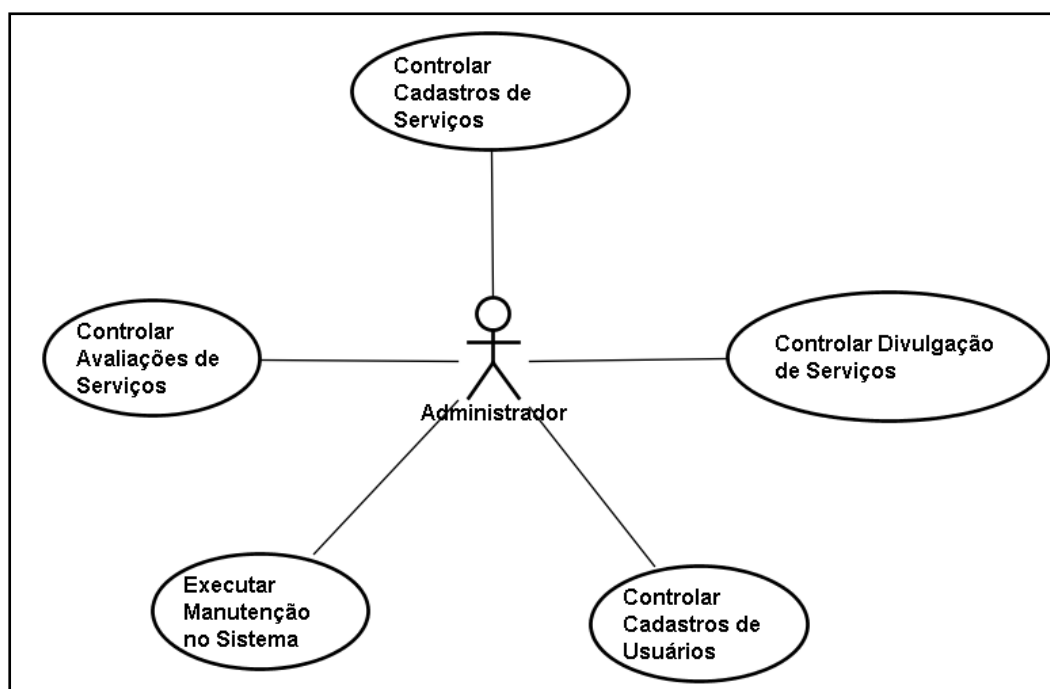


Figura 8. Diagrama de caso de uso do Administrador.

Os casos de uso expandidos dos casos de usos mostrados no diagrama da Figura 8 estão presentes no apêndice A.

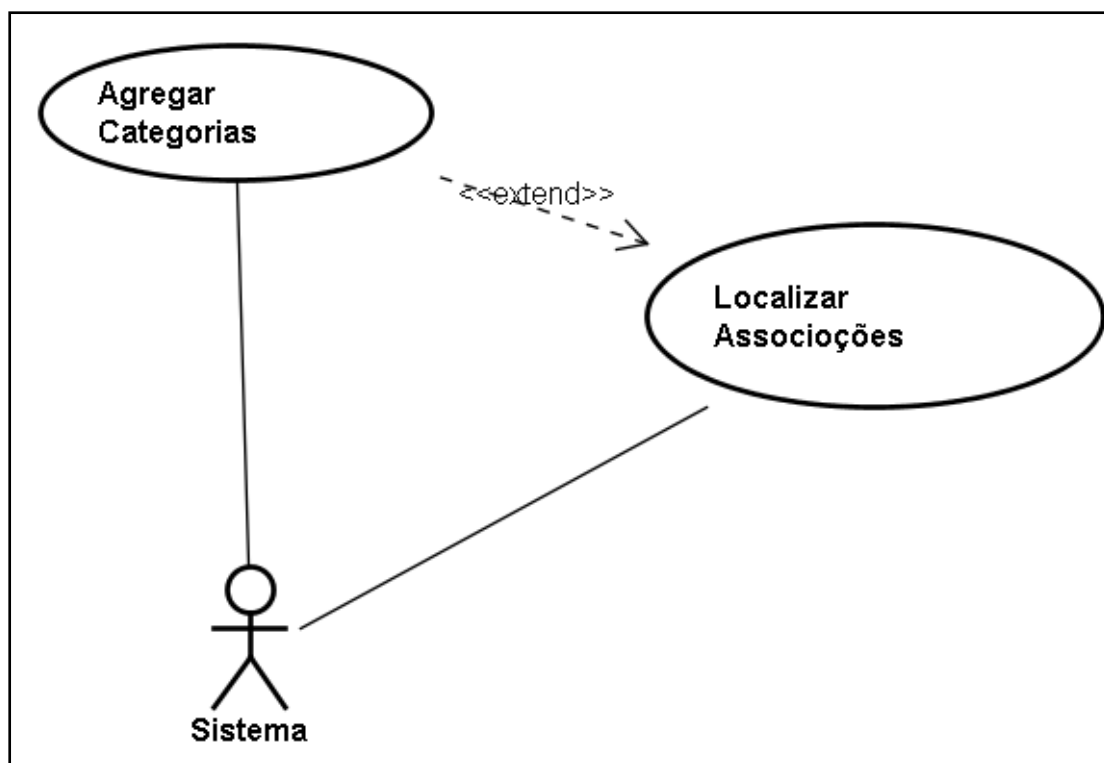


Figura 9. Diagrama de caso de uso do Sistema.

Os casos de uso expandidos dos casos de usos mostrados no diagrama da Figura 9 estão presentes no apêndice A.

3.5.1 DIAGRAMA DE ATIVIDADE

De acordo com GUEDES, G.T.A. (2011), diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle de uma atividade.

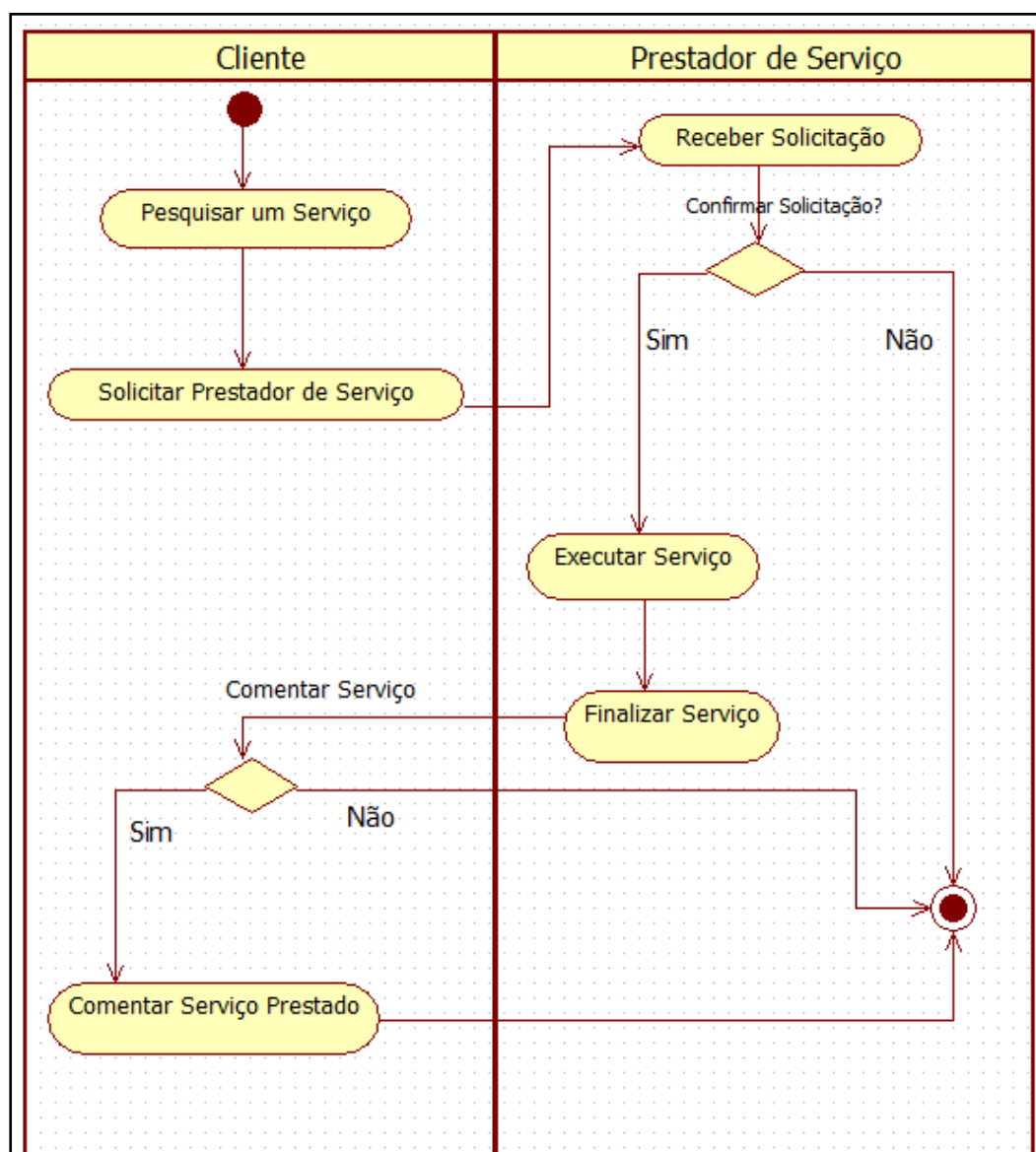


Figura 10. Diagrama de Atividade.

A Figura 10 apresenta o diagrama necessário para contratação de um serviço com os seguintes passos:

- O cliente pesquisa um determinado serviço;
- Com o resultado da pesquisa e possível visualizar perfis dos prestadores de serviços com comentários de serviços já prestados, assim poderá solicitar um melhor prestador de serviço;
- O prestador recebe a solicitação e decide se deseja aceitar ou não;
- Caso não aceite o processo é finalizado;

- Caso aceite executa o serviço;
- Após o serviço concluído notifica no sistema a conclusão do mesmo;
- O cliente recebe uma notificação para comentar o serviço prestado;
- Caso não aceite o processo é finalizado;
- Caso aceite é feito um comentário sobre a qualidade do serviço prestado;
- Finaliza o processo da aplicação;

3.5.2 DIAGRAMA DE COMPONENTES

De acordo com GUEDES, G.T.A. (2011), a Figura 11 apresenta os componentes do sistema em módulos de código-fonte, formulários, módulos executáveis etc. e determina como tais componentes estarão estruturados e irão interagir para que o sistema funcione de maneira adequada.

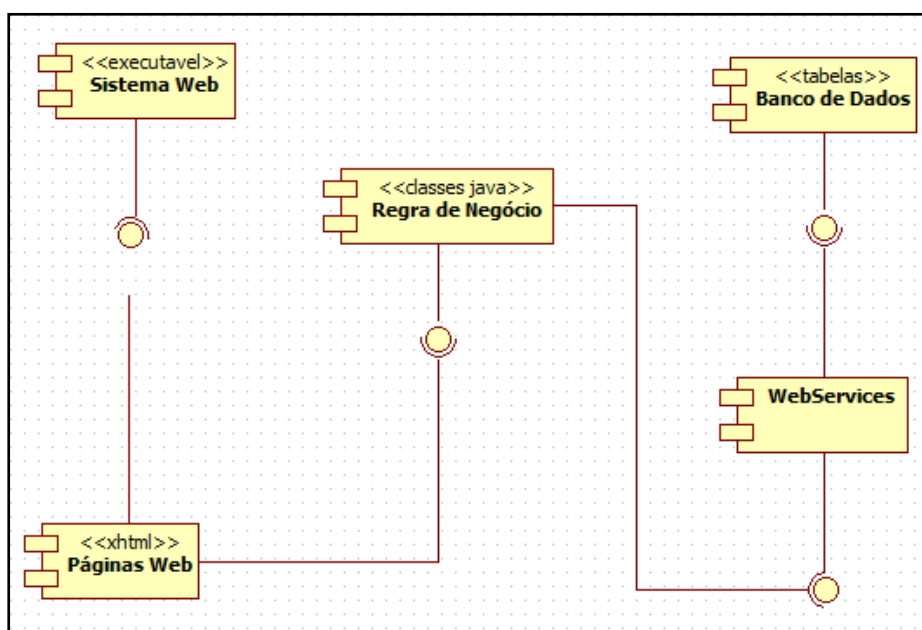


Figura 11. Diagrama de Componentes.

A Figura 11 descreve o diagrama da seguinte forma:

- Componente Sistema Web é o executável que requer uma interface do tipo XHTML (eXtensible Hypertext Markup Language) este processo de padronização tem em vista a exibição de páginas Web em diversos dispositivos (televisão, palm, celular, etc);
- Páginas web é o componente responsável por implementar todas as paginas XHTML e ao mesmo tempo requer uma interface da regra de negócio;

- Regra de Negócio é o componente que implementa as classes java da aplicação, onde é feita toda a regra de negócio através dos métodos CRUD (acrônimo de Create, Read, Update e Delete);
- Através do componente WebServices é feito a comunicação do sistema o banco de dados;
- O componente banco de dados são implementadas de acordo com os requisitos levantados na seção 3.1 e 3.2 deste trabalho todas as tabelas que armazenará os dados;

3.5.3 DIAGRAMA DE SEQUÊNCIA

De acordo com GUEDES, G.T.A. (2011), diagrama de sequência é um diagrama comportamental que preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Em geral, baseia-se em um caso de uso de diagrama de classes para determinar os objetos das classes envolvidas em um processo.

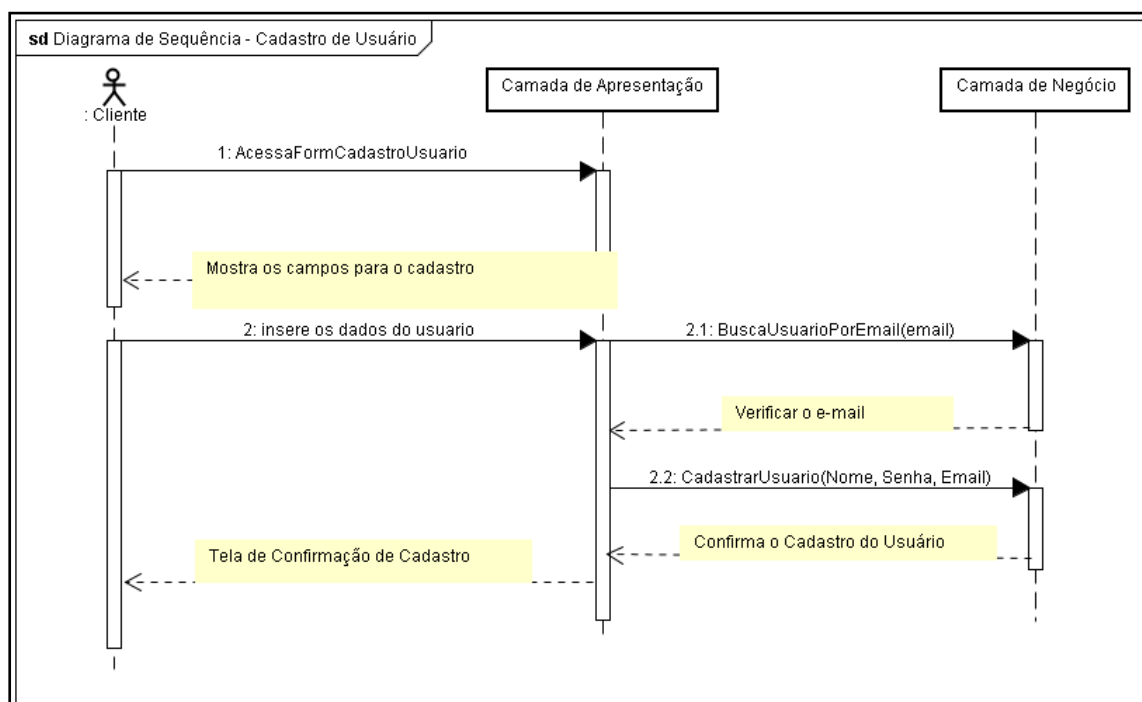


Figura 12. Diagrama de sequência de Cadastro de Cliente.

A Figura 12 exibe o fluxo do diagrama de sequência de cadastro de cliente para um entendimento técnico, onde o usuário acessa o formulário de cadastro, a aplicação exibe os campos para cadastro, o usuário insere os dados requeridos, a camada de negócio busca por um e-mail para verificar se já existe cadastrado, após a verificação efetua o cadastro e por fim exibe uma mensagem de confirmação para o usuário.

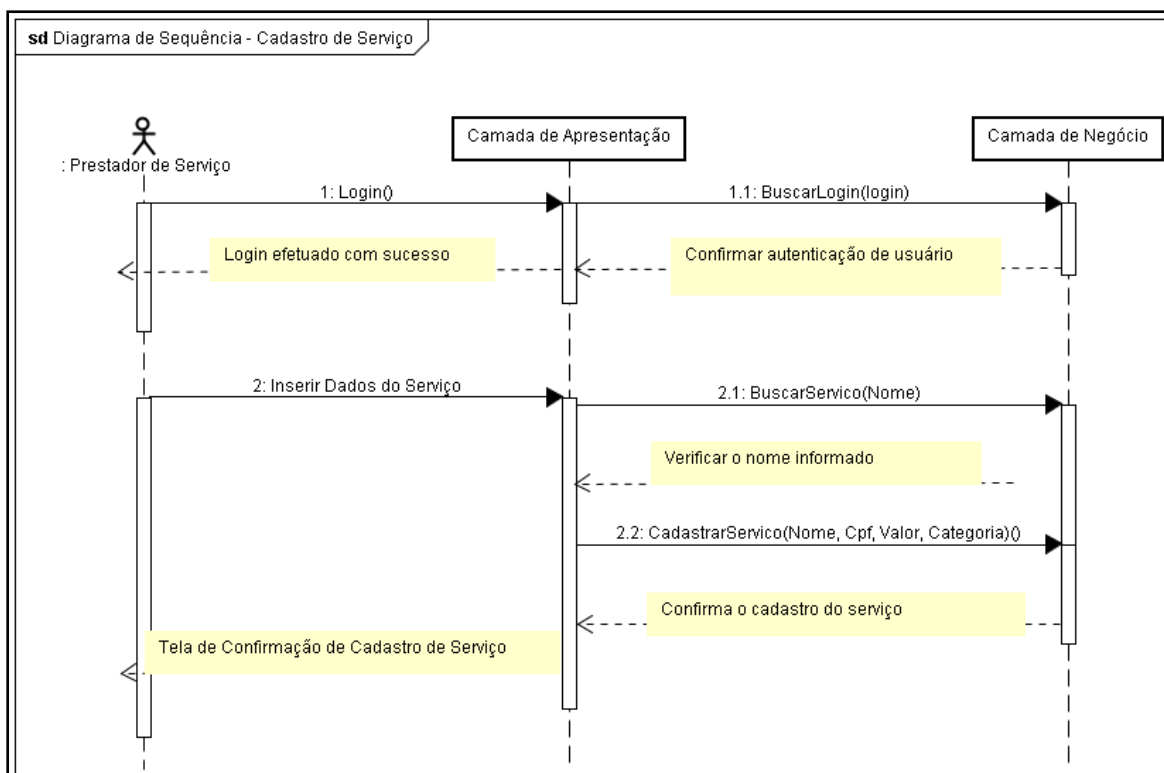


Figura 13. Diagrama de sequência de Cadastro de Serviço.

A Figura 13 exibe o fluxo do diagrama de sequência de cadastro de serviço para um entendimento técnico, onde o usuário efetua o login, a camada de apresentação busca por um login válido, autentica o usuário e efetua o login, a aplicação exibe os campos para cadastro de serviço, o usuário insere os dados requeridos, a camada de negócio busca por um serviço para verificar se já existe cadastrado, após a verificação efetua o cadastro e por fim exibe uma mensagem de confirmação de cadastro de serviço para o prestador.

4 CONSIDERAÇÕES FINAIS

Neste trabalho foram abordadas tecnologias para o desenvolvimento do banco de dados do sistema Busca de Serviço, bem como modelos que visam descrever a aplicação proposta. As principais partes do sistema foram modeladas usando diagramas de sequência. A modelagem do banco de dados foi desenhada juntamente com a criação de um dicionário de dados para o detalhamento de cada tabela. Tudo isso, em conformidade com os objetivos traçados na introdução deste trabalho.

Apresentou o processo de mineração de dados, detalhando as atividades que devem ser realizadas desde a compreensão do domínio, pré-processamento dos dados, até a avaliação do conhecimento extraído.

Conforme apresentado, pode-se observar que o processo de extração de conhecimento é bastante complexo e trabalhoso, pois envolve a execução de muitas tarefas, configuração de diversos parâmetros e grande interação com os usuários. Contudo, o sucesso do processo pode trazer uma recompensa valiosa para as organizações.

Com a comparação entre as aplicações de mesmo contexto foi possível observar funcionalidades implementadas, que se adequa ao sistema proposto com algumas modificações, tornando assim uma ferramenta mais completa.

Com isso, pretende-se estudar uma metodologia onde através de algoritmos de mineração implantados ao banco de dados seja possível agregar valores aos perfis dos usuários da aplicação, de forma que possa armazenar e resgatar conhecimento intrínseco ao contexto dos perfis de cada usuário.

5 REFERÊNCIA

IBGE. **Instituto Brasileiro de Geografia e Estatística**. Acessado em 09 de Março de 2014. Disponível em: <<http://www.ibge.gov.br/home>>.

LUGER, GEORGE F. (2004). **Inteligência Artificial: Estruturas e Estratégias para a Solução de Problemas Complexos**. (4ª ed.), Porto Alegre: Bookman.

PEGN – **Pequenas Empresas Grandes & Negócios**, 10 de Junho de 2013. Disponível em: <http://revistapegn.globo.com/Revista/Common/0,,EMI337305-17180,00-STARTUPS+QUE+CONECTAM+CONSUMIDORES+E+PRESTADORES+DE+SERVICOS.html>>. Acessado em 14 de Março de 2014.

GETNINJAS – (2011) – **Aplicação web de mesmo contexto**. Disponível em: <<http://www.getninjas.com.br/>>

RECOMIND.NET – (2011) – **Aplicação web de mesmo contexto**. Disponível em: <<http://www.recomind.net/>>

Zhou, Z. H. (2003). Three perspectives of data mining. **Artificial Intelligence journal** 143(1), 139-146.

W. Frawley, G. Piatetsky-Shapiro, C. M. (1992). Knowledge Discovery in databases: An overview. *AI Magazine*.

Bradley, P., U. Fayyad, & O. Mangasarian (1998). Data mining: Overview and optimization opportunities. Technical Report MSR-TR-98-04, Microsoft Research Report, Redmond, WA.

Fayyad, U., G. Piatetsky-Shapiro, & P. Smyth (1996b). Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp.82–88.

Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM* 42 (11).

Wei, J. M. (2003). Rough set based approach to selection of node. *International Journal of Computational Cognition* 1(2), 25–40.

S. Mitra, S. K. Pal, P. M. (2002). Data mining in soft computing framework: A survey. *IEEE Transactions on Neural Networks* 13(1), 3–14.

I. Sarafis, A. M. S. Zalzala, P. W. T. (2002). A genetic rule-based data clustering toolkit. In *Congress on Evolutionary Computation (CEC)*, Honolulu, USA.

REZENDE, SOLANGE OLIVEIRA (2003). **Mineração de Dados**, São Leopoldo – RS.

RAMEZ, E.; NAVATHE, S. B. **Sistemas de banco de dados**. 4ª. ed. São Paulo: Pearson Addison Wesley, 2005.

SOMMERVILLE, I. (2011). Engenharia de Software (9 ed.). (I. Bosnic, & K. G. de O. Gonçalves, Trans.) São Paulo: Pearson Prentice Hall.

Booch, G. and Rumbaugh, J. and Jacobson. (2006). **UML: Guia do usuário** (2ª ed.). Campus – RJ.

GUEDES, G.T.A. (2011). **UML 2 - UMA ABORDAGEM PRÁTICA** (2 ed.). São Paulo: NOVATEC.

APÊNDICE A

Principais Casos de Uso

Cadastrar Usuário

| | |
|------------------------------------|--|
| Caso de Uso: | Cadastrar Usuário |
| Ator(es): | Cliente |
| Finalidade: | Cadastrar o usuário no sistema |
| Visão Geral: | Permitir que o usuário se autentique no sistema para usufruir de suas funcionalidades. |
| Pré-Condições: | - |
| Pós-Condições: | Redirecionar o usuário para a tela principal da aplicação |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Insere dados. | |
| | 2. Validação de dados. Se for validado com sucesso é redirecionado a pagina principal. |
| Sequência Alternativa | |
| Linha 2 | Se não ocorrer sucesso na validação, o sistema exibe uma mensagem para o usuário. |

Cadastrar Serviço

| | |
|------------------------------------|---|
| Caso de Uso: | Cadastrar Serviço |
| Ator(es): | Prestador de Serviço |
| Finalidade: | Cadastrar um serviço no sistema |
| Visão Geral: | Permitir que o usuário prestador de serviço possa cadastrar um serviço no sistema. |
| Pré-Condições: | Ter o usuário cadastrado no sistema. |
| Pós-Condições: | Redirecionar o usuário para a tela principal da aplicação, e disponibilizar seu serviço para os demais clientes da aplicação. |
| Sequência Típica de Eventos | |

| Ação do Ator | | Resposta do Sistema |
|------------------------------|--|---|
| 1. Insere dados. | | |
| | | 2. Validação de dados. Ser for validado com sucesso é redirecionado a pagina principal. |
| Sequência Alternativa | | |
| Linha 2 | Se não ocorrer sucesso na validação, o sistema exibe uma mensagem para o prestador de serviço. | |

Pesquisar Serviço

| | |
|--|---|
| Caso de Uso: | Pesquisar Serviço |
| Ator(es): | Cliente |
| Finalidade: | Pesquisar um serviço no sistema. |
| Visão Geral: | Permitir que o cliente possa pesquisar um prestador de serviço cadastrado no sistema. |
| Pré-Condições: | Ter o serviço cadastrado no sistema. |
| Pós-Condições: | Exibir resultado da pesquisa para o cliente. |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Informa o estado, cidade e atividade. | |
| | 2. Exibir resultado da pesquisa. |
| Sequência Alternativa | |
| Linha 2 | Nenhum serviço encontrado. |

Contratar Serviço

| | |
|--|---|
| Caso de Uso: | Contratar Serviço |
| Ator(es): | Cliente, Prestador de Serviço. |
| Finalidade: | Contratar um serviço no sistema. |
| Visão Geral: | Permitir que o cliente possa contratar um prestador de serviço cadastrado no sistema. |
| Pré-Condições: | Ter os usuários (cliente, prestador de serviço) e o serviço, cadastrados no sistema. |
| Pós-Condições: | Enviar ao cliente um questionário de avaliação sobre o serviço prestado, e disponibilizar sua avaliação para os demais clientes da aplicação. |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Pesquisar o Serviço. | |
| | 2. Exibir resultado da pesquisa. |
| 3. Solicita Contratação | |
| | 4. Notifica prestador sobre a solicitação. |
| | 5. Informa ao Cliente a aceitação do serviço. |
| 6. Responde questionário de avaliação do serviço prestado. | |
| Sequência Alternativa | |
| Linha 1 | Nenhum serviço encontrado. |
| Linha 4 | Ator: Prestador de serviço não aceita solicitação de serviço. |
| Linha 6 | Ator: Cliente não responde o questionário de avaliação do serviço. |

Avaliar Serviço

| | |
|---------------------|------------------------------|
| Caso de Uso: | Avaliar Serviço |
| Ator(es): | Cliente. |
| Finalidade: | Avaliar um serviço prestado. |

| | |
|--|--|
| Visão Geral: | Permitir que o cliente possa avaliar o serviço prestado. |
| Pré-Condições: | Ter contratado um serviço no sistema. |
| Pós-Condições: | Exibir avaliação do cliente no perfil do prestador de serviço. |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Acessar área do sistema para efetuar avaliação. | |
| | 2. Exibir campos para digitar texto. |
| 3. Efetua avaliação. | |
| | 4. Notifica ao prestador sobre a avaliação feita. |
| Sequência Alternativa | |
| Linha 1 | Ator: Cliente sem privilégios para acessar essa área do sistema. |
| Linha 3 | Ator: Erro na autenticação da avaliação. |

Agregar Categorias

| | |
|--|--|
| Caso de Uso: | Agregar Categorias |
| Ator(es): | Sistema. |
| Finalidade: | Agregar categorias semelhantes a prestadores de serviços. |
| Visão Geral: | Permitir que o sistema agregue de forma automática categorias semelhantes aos perfis dos prestadores de serviços. |
| Pré-Condições: | Ter uma base de dados com prestadores cadastrados. |
| Pós-Condições: | Enriquecer os perfis dos prestadores de serviços de forma automática, com mais informações sobre seu serviço prestado. |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Executar algoritmo de mineração de dados. | |
| | 2. Exibir associações encontradas e notificar através do e-mail os prestadores de serviços sobre uma nova categoria que se enquadra ao seu perfil. |

| | |
|--|---|
| 3. Inserir nova categoria e atualizar o perfil do prestador. | |
| Sequência Alternativa | |
| Linha 1 | Ator: Nenhuma associação encontrada. |
| Linha 2 | Prestador não aceita atualizar seu perfil com a nova categoria. |
| Linha 3 | Ator: Falha ao atualizar o perfil do prestador de serviço. |

Localizar Associações

| | |
|------------------------------------|--|
| Caso de Uso: | Localizar Associações |
| Ator(es): | Cliente, Sistema. |
| Finalidade: | Localizar associações na pesquisa de um serviço. |
| Visão Geral: | Proporcionar aos usuários efetuar pesquisas não apenas pelo termo explicitado, mas por associações de termos sinônimos, regionalismos e correlacionados. |
| Pré-Condições: | Informar os dados no campo de pesquisa. |
| Pós-Condições: | Exibir o resultado esperado para o cliente. |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Informar dados de pesquisa. | |
| | 2. Exibir associações encontradas e exibir resultado da pesquisa. |
| Sequência Alternativa | |
| Linha 1 | Ator: Dados informados incorretamente. |
| Linha 2 | Nenhum serviço encontrado. |

Controlar Cadastros de Usuários

| | |
|---|---|
| Caso de Uso: | Controlar Cadastro de Usuários |
| Ator(es): | Administrador do Sistema. |
| Finalidade: | Controlar todos os usuários cadastrados no sistema. |
| Visão Geral: | Garantir autenticidade dos dados dos usuários cadastrados no sistema. |
| Pré-Condições: | Usuários cadastrados. |
| Pós-Condições: | - |
| Sequência Típica de Eventos | |
| Ação do Ator | Resposta do Sistema |
| 1. Pesquisar todos os usuários cadastrados no sistema. | |
| | 2. Exibir resultado da busca do administrador. |
| 3. Efetuar um contato com o prestador para verificação dos dados cadastrados. | |
| | 4. Contato efetuado com sucesso. |
| Sequência Alternativa | |
| Linha 2 | Nenhum serviço encontrado. |
| Linha 3 | Falha ao efetuar contato com o prestador de serviço. |

APÊNDICE B