



WESLEY GOMES DA SILVA

**DESENVOLVIMENTO DE SERVIÇOS DESTINADO A
CONTRATAÇÃO DE PRESTADORES DE SERVIÇOS UTILIZANDO
SOA – ARQUITETURA ORIENTADA A SERVIÇOS**

Palmas – TO

2014

Wesley Gomes da Silva

**DESENVOLVIMENTO DE SERVIÇOS DESTINADO A
CONTRATAÇÃO DE PRESTADORES DE SERVIÇOS UTILIZANDO
SOA – ARQUITETURA ORIENTADA A SERVIÇOS**

Trabalho de Conclusão do Curso de Sistemas de Informação da Faculdade Católica do Tocantins, apresentado com parte dos requisitos para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: M. Sc. Marco Antônio Firmino de Sousa.

Palmas – TO

2014

Wesley Gomes da Silva

**DESENVOLVIMENTO DE SERVIÇOS DESTINADO A
CONTRATAÇÃO DE PRESTADORES DE SERVIÇOS UTILIZANDO
SOA – ARQUITETURA ORIENTADA A SERVIÇOS**

Esta monografia foi julgada adequada pra obtenção do diploma de Bacharel em Sistemas de Informação do curso de graduação em Sistemas de Informação da Faculdade Católica do Tocantins.

Banca Examinadora

Assinatura do Orientador

Membros da Banca Examinadora

_____, ____/____/____

Local e data de aprovação:

Nota: _____

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!

Em todas as coisas, o sucesso
depende de preparação prévia!

RESUMO

A adoção de Arquitetura Orientada a Serviços - SOA tem se tornado prática frequente em organizações que buscam resolver problemas de flexibilidade, mudanças e de integração entre suas aplicações, SOA chega com o objetivo de compartilhar informações entre aplicações distintas através de serviços bem definidos e trazendo consigo vantagens como reutilização dos serviços criados em outras aplicações, facilidade de manutenção dos serviços, integração com outros serviços assim disponibilizando uma padronização. O objetivo deste trabalho foi criar um mecanismo para o controle de prestadores de serviços através da aplicação de SOA, assim introduzindo uma nova visão na concepção de desenvolvimento de aplicações web. Em paralelo a aplicação de SOA no trabalho, foi desenvolvido um *Web Service* baseado nos protocolos SOAP, UDDI, WSDL e XML que implementam interações remotas entre os aplicativos sob redes baseadas no protocolo IP, garantindo assim o armazenamento dos serviços disponibilizados pelo projeto e disponibilizando-os para consumo das diversas aplicações clientes.

Palavra-chave: Arquitetura Orientada a Serviço, Web Service, Busca de Serviços.

ABSTRACT

ÍNDICE DE FIGURAS

Figura 1. Modelo Operacional Triangular SOA (MARZULLO, 2009).	16
Figura 2. Estrutura de um Envelope SOAP (MARZULLO, 2009).	19
Figura 3. Estrutura de um WSDL (ERL, 2011).	20
Figura 4. Interface web do GetNinjas (GETNINJAS - 2011).	24
Figura 5. Interface web do Recomind.net (RECOMIND.NET - 2011).	25
Figura 6. Diagrama de Atividades.....	29
Figura 7. Diagrama de Componentes.	30
Figura 8. Diagrama de Sequência de Cadastro de Cliente.	31
Figura 9. Diagrama de Sequência de Cadastro de Serviço.....	32
Figura 10. Diagrama do Banco de Dados.	33

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface

CEO - Executive Officer

FACTO - Faculdade Católica do Tocantins

HTTP - Hypertext Transfer Protocol

IBGE - Instituto Brasileiro de Geografia e Estatística

IDE - Integrated Development Environment

NTI - Núcleo de Tecnologia da Informação

PIB - Produto interno bruto

SGBD - Sistema de Gerenciamento de Banco de Dados

SOA - Arquitetura Orientada a Serviços

SOAP - Simple Object Access Protocol

TI - Tecnologia da Informação

UDDI - Universal Description, Discovery and Integration

UML - Unified Modeling Language

WSDL - Web Services Description Language

XML - eXtensible Markup Language

SINE - Sistema Nacional de Emprego

ÍNDICE DE TABELAS

SUMÁRIO

1. INTRODUÇÃO	13
1.1. DEFINIÇÃO DO PROBLEMA.....	14
1.2. JUSTIFICATIVA.....	14
1.3. OBJETIVO.....	14
1.3.1 OBJETIVOS ESPECÍFICOS.....	14
1.4. MOTIVAÇÃO	15
1.5. ESTRUTURA DA MONOGRAFIA.....	15
2 REFERENCIAL TEÓRICO	16
2.1 ARQUITETURA ORIENTADA A SERVIÇO	16
2.2 WEB SERVICES	18
2.2.1 SOAP.....	19
2.2.2 WSDL	20
2.2.3 UDDI.....	21
2.3 CONTRATAÇÃO DE SERVIÇOS NO BRASIL	21
2.4 TRABALHOS RELACIONADOS.....	23
3 METODOLOGIA	26
4 DESENVOLVIMENTO	28
4.1 VISÃO GERAL DA FERRAMENTA.....	28
4.2 REQUISITOS FUNCIONAIS	28
4.3 MODELAGEM.....	29
4.3.1 Diagrama de Atividade	29
4.3.2 Diagrama de Componentes	30
4.3.3 Diagrama de Sequência.....	31
4.3.4 Diagrama do Banco de Dados.....	33
5 CONSIDERAÇÕES FINAIS	34
6 REFERÊNCIA	35

APÊNDICE A	36
APÊNDICE B	37
APÊNDICE C	41

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). De acordo com (LUGER, 2004), 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. DEFINIÇÃO DO PROBLEMA

O problema consiste em desenvolver um sistema que disponibilize serviços para que aplicações criadas em plataformas diferentes possam consumir tais serviços tendo em consideração um custo baixo ou gratuito. Existem sistemas como a GetNinjas, Iguanafix, Recomend.net que são capazes de fornecer informações úteis e em tempo real sobre prestadores de serviços, serviços ofertados, qualificações de serviços e localização, porém essas aplicações não são utilizadas na região norte, em específico no estado do Tocantins, pelo fato de ser uma região pequena em termos de prestação de serviços se comparada com outros estados com São Paulo e Rio de Janeiro, com isso pretende-se desenvolver uma ferramenta para atender as necessidades da região norte em específico o Tocantins.

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. OBJETIVO

Criar um sistema de informação destinado a contratação de prestadores de serviços utilizando SOA – Arquitetura Orientada a Serviços.

1.3.1 OBJETIVOS ESPECÍFICOS

- Criar o banco de dados para o armazenamento e consumo das informações;
- Modelar a ferramenta proposta;
- Criar serviços que possam ser consumidos por diferentes plataformas;
- Desenvolver uma aplicação de teste com o objetivo de demonstrar o consumo dos serviços desenvolvidos.

1.4. MOTIVAÇÃO

Melhorar a vida das pessoas com o auxílio da tecnologia, tornando mais fácil a contratação de serviços, informar prestadores de serviços mais próximos de sua localização, visualização de comentários referentes aos serviços prestados, 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.

1.5. ESTRUTURA DA MONOGRAFIA

A seção 2(dois) desse trabalho mostrará o referencial teórico que aborda os principais conceitos que envolvem a utilização de SOA e trabalhos relacionados. A metodologia será apresentada na seção 3(três), descrevendo os métodos e procedimentos utilizados neste trabalho. A visão do sistema e a codificação serão expostas na seção 4(quatro) apresentando com detalhes o processo de construção dos serviços, bem como os principais trechos de código utilizados no processo. E, por fim, a seção 5(cinco) apresenta informações sobre as conclusões e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Nessa seção serão abordados todos os conceitos que fundamentam a criação de serviços utilizando SOA, desde o processo de utilização até o seu funcionamento.

2.1 ARQUITETURA ORIENTADA A SERVIÇO

Ao desvincular o domínio de negócio de tecnologias e modelos específicos, como linguagens ou sistemas operacionais, SOA oferece à organização a chance de acompanhar as mudanças exigidas por seu contexto de negócio sem que isso sobrecarregue o uso dos recursos de Tecnologia da Informação (TI). De acordo com (MARZULLO, 2009) dentro dessa abordagem, podemos identificar benefícios como separação de responsabilidades, organização lógica e facilidade de uso.

Tratando de aplicações baseadas em serviços, SOA é composto de três elementos que representam papéis distintos de interação: o consumidor do serviço, o prestador de serviço e o registro do serviço. Esses três elementos compõem o que chamamos de modelo operacional triangular.

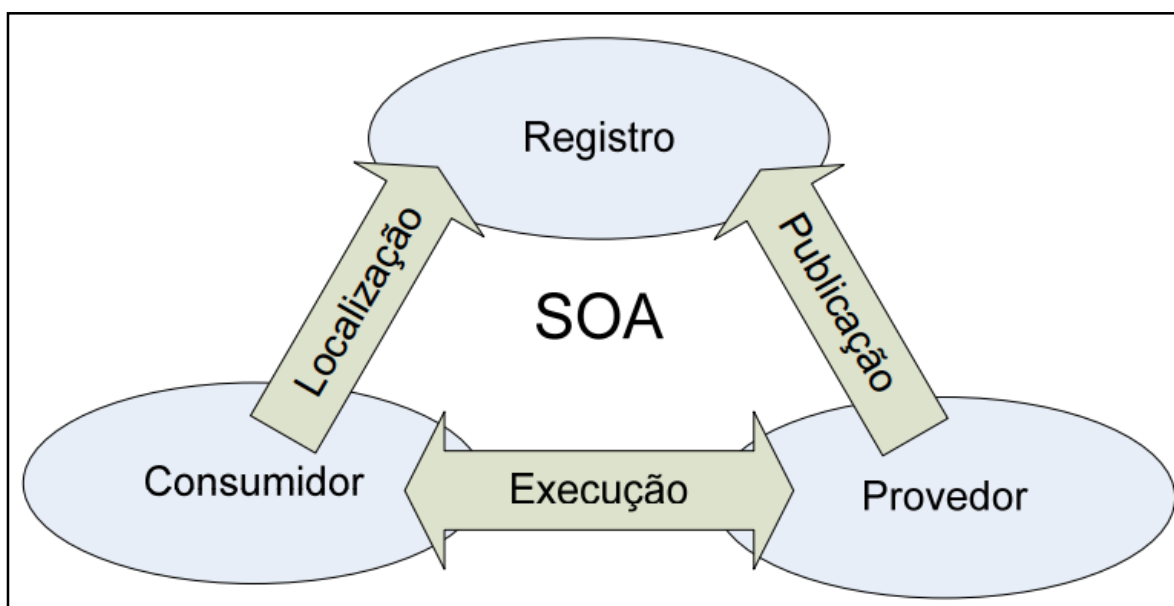


Figura 1. Modelo Operacional Triangular SOA (MARZULLO, 2009).

De acordo com Figura 1, o modelo organizacional de uma arquitetura orientada a serviços se comportam da seguinte forma:

- **Provedor do Serviço:** aquele que oferece o serviço, responsável pela infraestrutura do acesso, determinando todo o seu comportamento.
- **Consumidor do Serviço:** aquele que consome o serviço, responsável pelo comportamento daquele que representa o cliente da organização.
- **Registro do Serviço:** mecanismo que permite ao provedor de serviço cadastrar seus serviços e ao consumidor encontrá-los, normalmente esses registros contêm informações sobre o negócio, informações técnicas com linguagens, tecnologias utilizadas e informações sobre o serviço em si.

De acordo com (KUMAR, 2012), a forma de integrar os dados através dos serviços é um meio de chegar a SOA. Com essa arquitetura, é possível dizer que surgiu uma nova forma de pensar em TI, desaparecendo a ideia de sistemas monolíticos e surgindo a ideia de processos de negócios. Utilizando SOA, os processos podem ser alterados de maneira rápida e eficiente. O tempo de desenvolvimento utilizando em ambientes que passam por problemas de negócios, tais como: gestão de processos, migração de sistemas legados, integração de sistemas e fusão de empresas é encurtado, pois o desenvolvimento das aplicações não é iniciado do zero, aproveitando boa parte do que já existe e fazendo o uso de umas das suas características, que é a reutilização de código.

Segundo (KUMAR, 2012), arquitetura orientada a serviço se torna adequada para a criação de programas em forma de serviços de apoio podendo acontecer à interação através da internet. O conteúdo “serviço” refere-se principalmente às operações envolvendo negócios. Um sistema que tem como proposta utilizar SOA é formado por vários serviços, apresentando geralmente baixo acoplamento por natureza, para caso necessário, novos serviços possam ser adicionados ou os existentes possam ser modificados de maneira rápida, de acordo com a necessidade e dinamismo dos negócios.

Segundo (MARZULLO, 2009), no contexto de SOA, a definição de serviço é:

“Um serviço é um tipo de relacionamento (contrato) entre um provedor e um consumidor, sendo esse provedor se compromete em realizar determinadas tarefas com resultados pré-estabelecidos para um

consumidor, e que, por sua vez, se compromete a usar o serviço da forma contratada.”

Antes de concluir com os conceitos, há a necessidade de se definirem mais dois componentes fundamentais para o entendimento da arquitetura SOA: o repositório de serviços e o barramento de serviços. O repositório de serviços é a entidade que provê facilidades para o descobrimento dos serviços disponíveis na arquitetura, sobretudo daqueles fora do escopo temporal e funcional do sistema de informação, ou melhor, do processo de desenvolvimento do nosso sistema. Ele fornece informações como: localização virtual, provedor, taxas, limitações técnicas, aspectos de segurança, entre outras. O barramento de serviços é o meio utilizado para conexão entre todos os participantes da SOA como: serviços e aplicações *frontends*, engloba uma grande diversidade de produtos e conceitos.

De acordo com (MARZULLO, 2009), barramento de serviços deve ser definida de maneira a permitir a integração de aplicações desenvolvidas em diferentes linguagens e plataformas e deve estar alinhada com as atividades comuns em um ciclo de vida SOA.

2.2 WEB SERVICES

De acordo com (MARZULLO, 2009), um *Web Service* representa a materialização da ideia de um serviço que é disponibilizado na internet e que pode ser acessado em qualquer lugar do planeta. Representa uma lógica de negócio que permite que um ou mais clientes enviem requisições de um tipo bem definido de informação e recebam respostas síncronas ou assíncronas.

Os *Web Services* dispõem de soluções viáveis e interessantes se tratando de interoperabilidade entre sistemas totalmente ou parcialmente baseados no modelo cliente-servidor ou até mesmo outros modelos arquiteturais adotados em sistemas distribuídos. Ao usarmos, é possível construir sistemas com reaproveitamento de componentes disseminados no mercado, como banco de dados, serviços comerciais e muito mais (SHARP, 2011).

A demanda por soluções de negócios integradas e distribuídas cresce a cada dia, e a teoria de *web services* vem ao encontro dessas necessidades para dar uma solução definitiva quanto a forma de se materializar novas estratégias de negócio orientada a

serviços. Portanto, o que antes era visto com um repositório de conteúdo, agora se estabeleceu como um repositório de serviços, criando soluções distribuídas e descentralizadas.

Uma definição técnica de *web services* poderia ser como um serviço disponibilizado na internet, descrito via WSDL (Web Services Description Language), registrado via UDDI (Universal Description, Discovery and Integration), acessado utilizando SOAP (Simple Object Access Protocol) e com os dados transmitidos sendo representados em XML (eXtensible Markup Language). A seguir, encontra-se uma breve explicação de algumas tecnologias citadas na definição anterior:

2.2.1 SOAP

Segundo (MARZULLO, 2009) SOAP é um protocolo para troca de informações em ambiente distribuído. É baseado em definições XML. Esse protocolo encapsula as chamadas e retornos aos métodos dos *web services*, sendo utilizado, principalmente, sobre HTTP (Hypertext Transfer Protocol). A Figura 2 ilustra a estrutura do protocolo SOAP.



Figura 2. Estrutura de um Envelope SOAP (MARZULLO, 2009).

Como mostrado na Figura 2, o elemento envelope é obrigatório, ele define a raiz da mensagem e determina como o documento XML é transportado em uma mensagem SOAP e como deve ser traduzida pelo *web service* no serviço real. O conceito de *Header* é definido como um cabeçalho opcional. Ele transporta informações adicionais, por

exemplo, se a mensagem deve ser processada por um determinado nó intermediário. Quando utilizado, o *Header* deve ser o primeiro elemento do Envelope. Por fim, o *Body* é um elemento obrigatório que irá armazenar o documento a ser transmitido. O elemento *Body* pode conter um elemento opcional *Fault*, usado para indicar códigos e mensagens de erro que podem ocorrer.

2.2.2 WSDL

Segundo (SAUDATE, 2013), WSDL descreve a interface do serviço de forma estruturada e padronizada usando XML. Ela permite, através da definição de um vocabulário em XML, a possibilidade de descrever serviços e a troca de mensagens. Mais especificamente é responsável por prover as informações necessárias para a invocação do *web service*, como sua localização, operações disponíveis e suas assinaturas. Na Figura 3 é demonstrada a estrutura de um WSDL.



Figura 3. Estrutura de um WSDL (ERL, 2011).

- *Types*: Age como um container para definir os tipos de dados usados dentro da mensagem. Permite usar *XML Schema* para definir as estruturas de dados.

- *Message*: permite descrever as mensagens que são trocadas entre o serviço e o consumidor do serviço. Uma mensagem pode possuir várias partes, sendo que cada parte possui um nome e um tipo de dados.
- *PortType*: define o local em que está hospedado o *web service*, identifica o grupo de ações que podem ser executadas num único ponto de acesso e utiliza o elemento *operation* para representar uma operação dentro desse ponto de acesso.
- *Binding*: define o formato da mensagem e o protocolo de comunicação do elemento *Types*.
- *Service e Ports*: Aparece no final do arquivo WSDL e identifica o serviço que é composto.

2.2.3 UDDI

Um UDDI contém informações categorizadas sobre os serviços e as funcionalidades que eles oferecem, e permite a associação desses serviços com suas informações técnicas, geralmente definidas usando WSDL. Como dito anteriormente, o arquivo de descrição em WSDL descreve as funcionalidades do *web service*, a forma de comunicação e sua localização. A especificação UDDI define uma API (Application Programming Interface) baseada em mensagens SOAP, com uma descrição em WSDL do próprio *web service* do servidor de registro. A maioria dos servidores de registro UDDI também provê uma interface de navegação por browser. De acordo com (CALIENDO, 2010), o UDDI é similar a um motor de busca (como o Google), tem como objetivo ser um mediador do serviço, permitindo que os clientes requisitantes encontrem um fornecedor que melhor atenda suas necessidades.

Desta forma, o UDDI é uma interface web, que determina serviços concedendo a descrição e descoberta de negócios, sendo disponibilizado o acesso e o gerenciamento destes serviços.

2.3 CONTRATAÇÃO DE SERVIÇOS NO BRASIL

O Processo de informalidade dos trabalhadores por conta própria pode ser representado como microempresários, desejamos apontar que eles estão criando uma

ocupação no mercado de bens, principalmente na prestação de serviços, com o objetivo de se auto-empregar. O que caracteriza esse grupo, especialmente aqueles que operam com baixo nível de produtividade com relação às empresas capitalistas, é que compreende indivíduos com pouco nível de capital físico ou humano, que são simultaneamente patrões e empregados de si mesmos.

Fala como funciona a contratação de prestadores de serviços no brasil, mostrar comparações e vantagens do mercado.

Falta terminar....

2.4 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.

2.4.1 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 4.



Figura 4. Interface web do GetNinjas (GETNINJAS - 2011).

2.4.2 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 5 apresenta sua interface gráfica.



Figura 5. Interface web do Recomind.net (RECOMIND.NET - 2011).

3 METODOLOGIA

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

- Pesquisa e coleta de dados;
- Modelagem;
- Desenvolvimento;
- Documentação;

Na etapa de Pesquisa e coleta de dados, foi realizada uma revisão bibliográfica baseada na leitura de textos disponíveis em livros e artigos técnico-científicos, realizou então um questionário destinado a coleta de dados sobre os prestadores de serviço e clientes, por meio da ferramenta Google Docs¹ que, além de permitir a criação de documentos, planilhas, apresentações, e desenhos, oferece o recurso de criar formulários online, onde um ou mais usuário que acessam o endereço virtual do formulário podem preencher e enviar suas respostas que estão disponíveis ao proprietário e colaboradores quando houver. Tais resultados serviram para validação dos requisitos levantados até o momento e, possivelmente, a descoberta de outros, o questionário da pesquisa juntamente com alguns gráficos é apresentado no apêndice A deste trabalho, a etapa de pesquisa busca organizar os dados de forma que se possa comparar os sistemas de mesmo contexto e extrair características interessantes de cada um.

A etapa de Modelagem descreve os principais diagramas UML (Unified Modeling Language) para caracterização formal do sistema proposto. Segundo (BOOCH, G. and RUMBAUGH, J. and JACOBSON, 2006), UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas de software. A UML proporciona uma forma padrão para a preparação de planos de arquitetura de projetos de sistemas, os diagramas ilustram graficamente a arquitetura, estrutura, interações e comportamento do sistema proporcionando uma abstração de vários contextos inerentes ao software. Para a explicação do banco de dados, foi criado um dicionário de dados com o intuito de detalhar minuciosamente cada campo de cada tabela do banco de dados.

¹Google Docs. Disponível em <https://docs.google.com/>

Na etapa de Desenvolvimento foram realizadas todas as configurações de ambiente necessárias para o desenvolvimento do *Web Service* e criação dos serviços necessários para compor a aplicação proposta.

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

Todos os experimentos foram elaborados em locais de apoio como o Núcleo de Tecnologia da Informação (NTI), ligado ao curso de Sistemas de Informação da Faculdade Católica do Tocantins (FACTO). O tempo para realização do projeto teve cronograma próprio definido no pré-projeto da pesquisa. Os materiais essenciais para a realização deste trabalho se baseiam em ferramentas gratuitas e versões comunitárias. Utilizou-se a Integrated Development Environment (IDE) ECLIPSE LUNA para desenvolvimento dos serviços, do *Web Service* e do sistema web para consumo dos serviços oferecidos. O banco de dados adotado para o ambiente de produção do sistema foi o MySQL. Foi utilizado o Sistema de Gerenciamento de Banco de Dados (SGDB) MySQL Workbench 6.0. A ferramenta Workbench possui as vantagens de ser gratuita, e ser compatível com todos os sistemas operacionais, possui uma documentação bem detalhada e com uma linguagem simples, permite fazer a engenharia reversa caso necessário e exporta os dados em vários formatos, como por exemplo, PNG e PDF. Nesse projeto, a ferramenta MySQL Workbench foi utilizada para a criação das tabelas e seus respectivos dados, e também para a sincronização do banco de dados. Por fim, na criação do *Web Service* foi utilizado o Apache Axis e configurado o TOMCAT 7.0 como servidor web.

4 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.

4.1 VISÃO GERAL DA FERRAMENTA

Os serviços desenvolvidos durante este trabalho é uma proposta para melhorar a acessibilidade de informações sobre prestadores de serviços no Tocantins.

Todos os dias, milhares de pessoas precisam de profissionais como: pintores, eletricitas, domesticas, jardineiros entre outros, como encontrar esse tipo de profissional de forma rápida fácil e confiável? Esta realidade está presente na maioria das cidades brasileiras. Neste cenário a cidade Palmas, capital do estado do Tocantins não deixa de estar incluída, pois é notória tal situação no cotidiano de boa parte de seus cidadãos.

Quando um usuário necessita de informações concernentes a prestadores de serviços qualquer pessoa é consultada a fim de extrair alguma informação útil para sua necessidade o que pode não lhe informar dados precisos e de confiança. Quando alguém não é abordado para fins de consulta de informações, o usuário só poderá adquirir informações em órgão público como o Sistema Nacional de Emprego - SINE. Com essa realidade apresentada, não há nada mais conveniente do que obter informações de prestadores de serviços, qualificações de serviços, localização de prestadores entre outros, sem importar o lugar e o momento. O sistema proposto visa à disponibilização de informações dessas naturezas, além de recursos que auxiliarão qualquer pessoa a localizar um prestador de serviço com mais facilidade e garantia.

4.2 REQUISITOS FUNCIONAIS

4.3 MODELAGEM

Afim de demonstrar graficamente a estrutura dos requisitos do software, nessa seção serão abordados diagrama de componentes, atividade, sequência e diagrama do banco de dados, por conter diversos diagramas de caso de uso ele é apresentado no apêndice A deste trabalho juntamente com os casos de uso expandidos.

4.3.1 Diagrama de Atividade

De acordo com (GUEDES, 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.

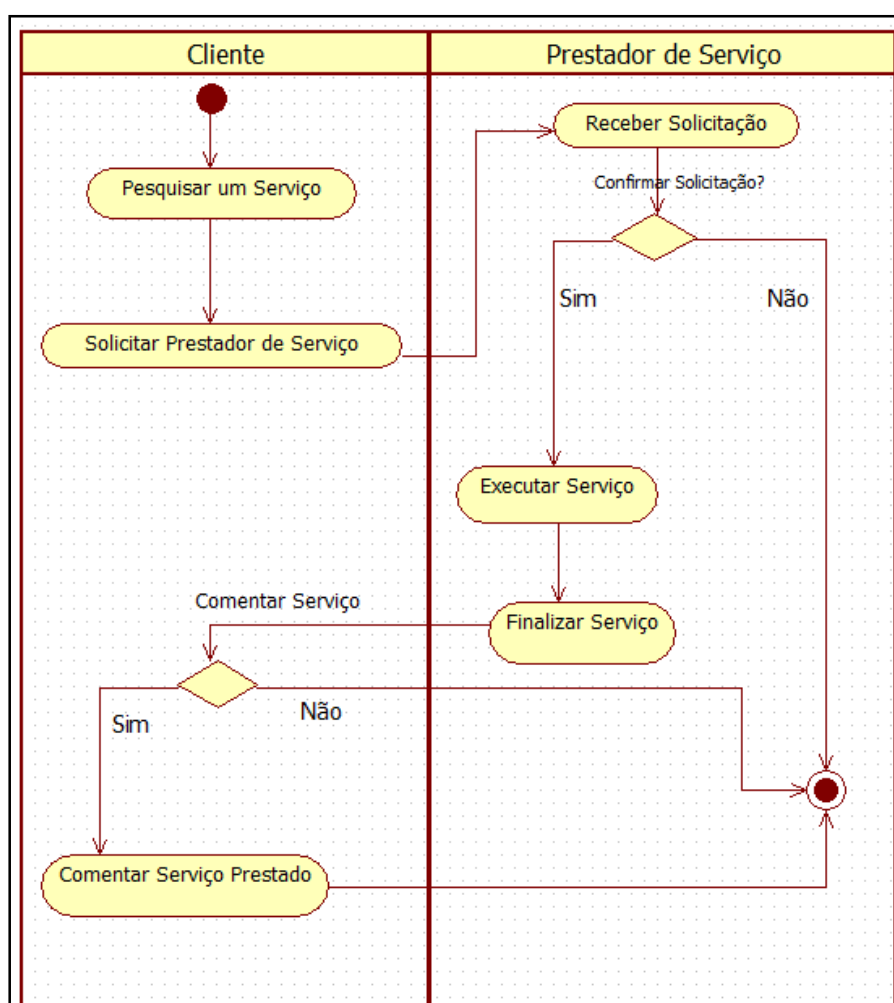


Figura 6. Diagrama de Atividades.

A Figura 6 apresenta o diagrama necessário para contratação de um serviço com os seguintes passos:

- O cliente pesquisa um determinado serviço;
- Com o resultado da pesquisa é 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;

4.3.2 Diagrama de Componentes

Figura 7 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.

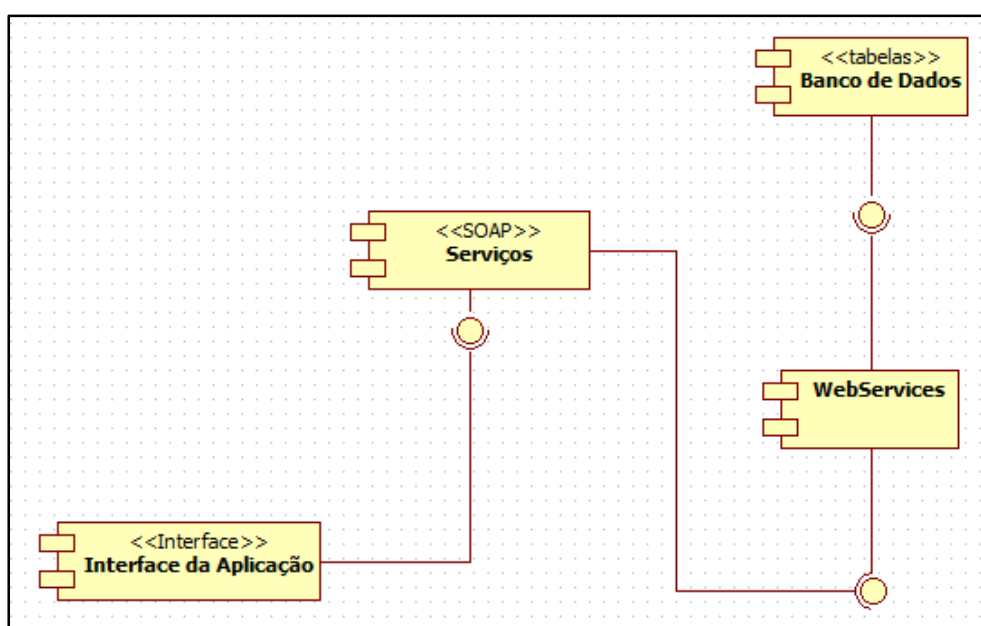


Figura 7. Diagrama de Componentes.

A Figura 7 descreve o diagrama da seguinte forma:

- O componente banco de dados é implementado de acordo com os requisitos levantados ao longo do trabalho. É responsável por persistir e recuperar os dados;
- Através do componente *Web Services* é feita a comunicação entre o banco de dados e os serviços implementados;
- O componente Serviços é responsável pelo desenvolvimento de todos os serviços necessários para a utilização através de outras interfaces;
- Componente Interface da Aplicação é o executável que requer uma interface de acesso, este processo de padronização tem em vista a exibição de páginas Web em diversos dispositivos (computadores, smartphone, televisão, tablets, etc);

4.3.3 Diagrama de Sequência

De acordo com (GUEDES, 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.

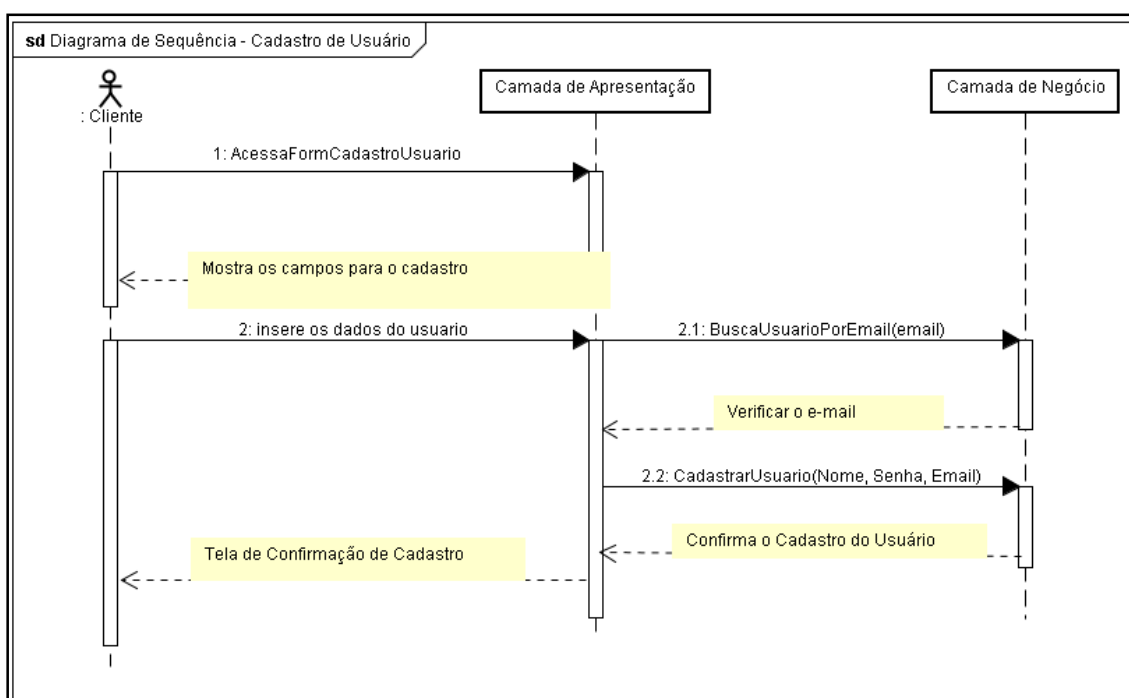


Figura 8. Diagrama de Sequência de Cadastro de Cliente.

A Figura 8 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.

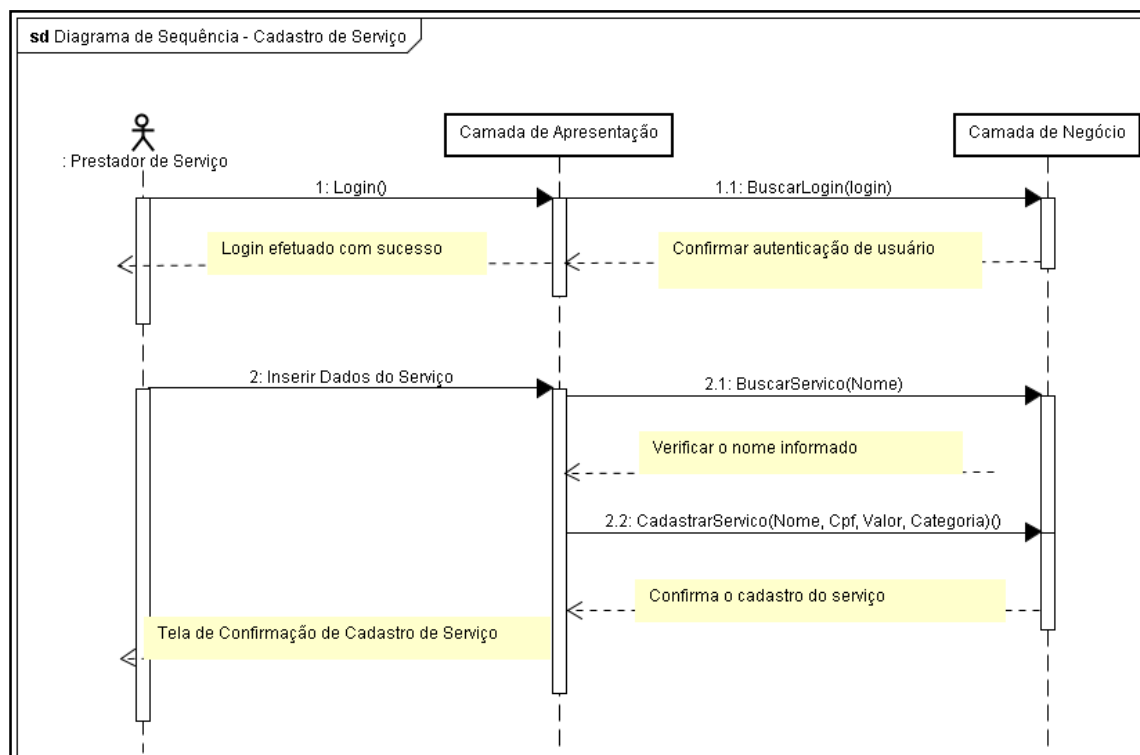


Figura 9. Diagrama de Sequência de Cadastro de Serviço.

A Figura 9 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.

4.3.4 Diagrama do Banco de Dados

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 serão estabelecidos com chaves estrangeiras, que serão nomeadas com o prefixo “fk” seguido de “_” e o nome da chave primária da outra tabela concatenado com nome da tabela relacionada.

A Figura 10 demonstra como as tabelas estão relacionadas umas com as outras:

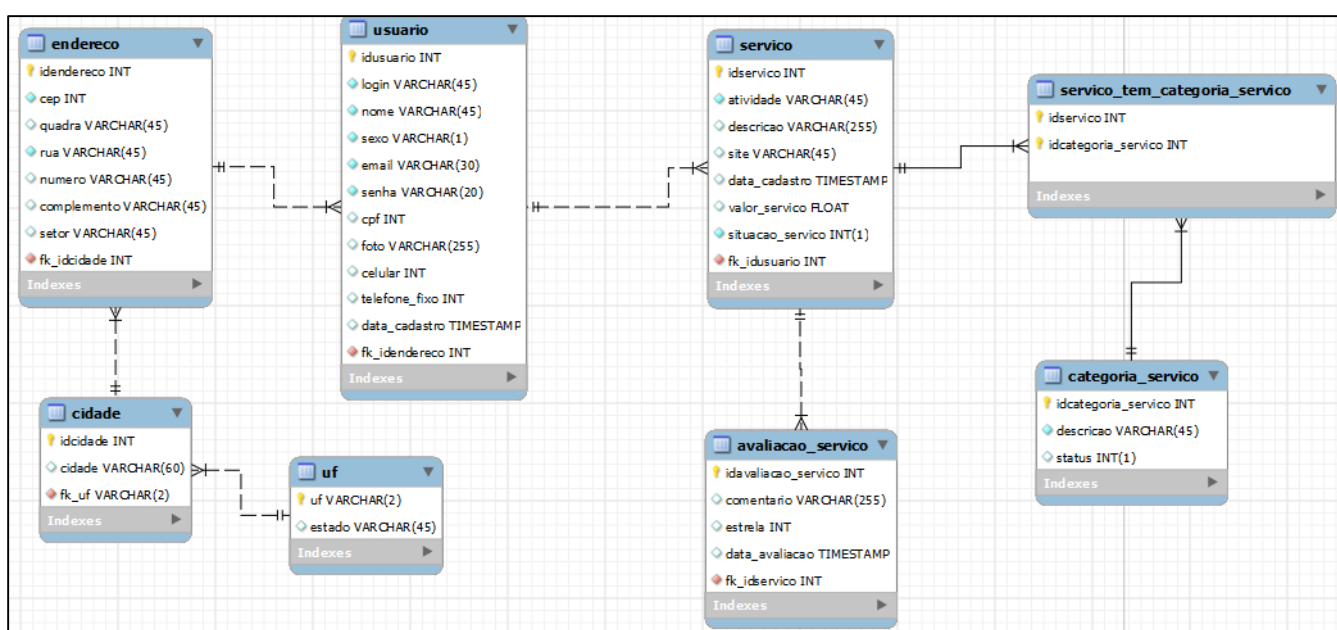


Figura 10. Diagrama do Banco de Dados.

Como mencionado na metodologia, para uma melhor compreensão humana sobre como os dados estão relacionados foi criado um dicionário de dados que é apresentado apêndice C deste trabalho, detalhando cada tabela do banco de dados.

5 CONSIDERAÇÕES FINAIS

Descrever Considerações finais

6 REFERÊNCIA

IBGE. **Instituto Brasileiro de Geografia e Estatística**. Acessado em 25 de Agosto 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.

BOOCH, G. and RUMBAUGH, J. and JACOBSON. (2006). **UML: Guia do usuário** (2ª ed.). Campus – RJ.

MARZULLO, F. P. (2009). **SOA na Prática**. Novatec.

KUMAR, B. V. (2012). **Implementando SOA Usando JAVA™ EE**. Rio de Janeiro: ALTA BOOKS.

SHARP, J. (2011) **Microsoft Visual C# 2010: passo a passo**. Tradução de Tereza Cristina Félix de Sousa e Edson Furmankiewicz. Porto Alegre: Bookman.

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/>>

SAUDATE, A. (2013). **SOA aplicado: Integrando com web services e além**. São Paulo: Casa do Código.

ERL, T. (2011). **Introdução às tecnologias Web Services: SOA, SOAP, WSDL e UDDI**. WebMobile.

CALIENDO, R. F. (2010). **Desenvolvimento de uma Aplicação utilizando SOA: um Estudo de Caso**. pp. 2-9.

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

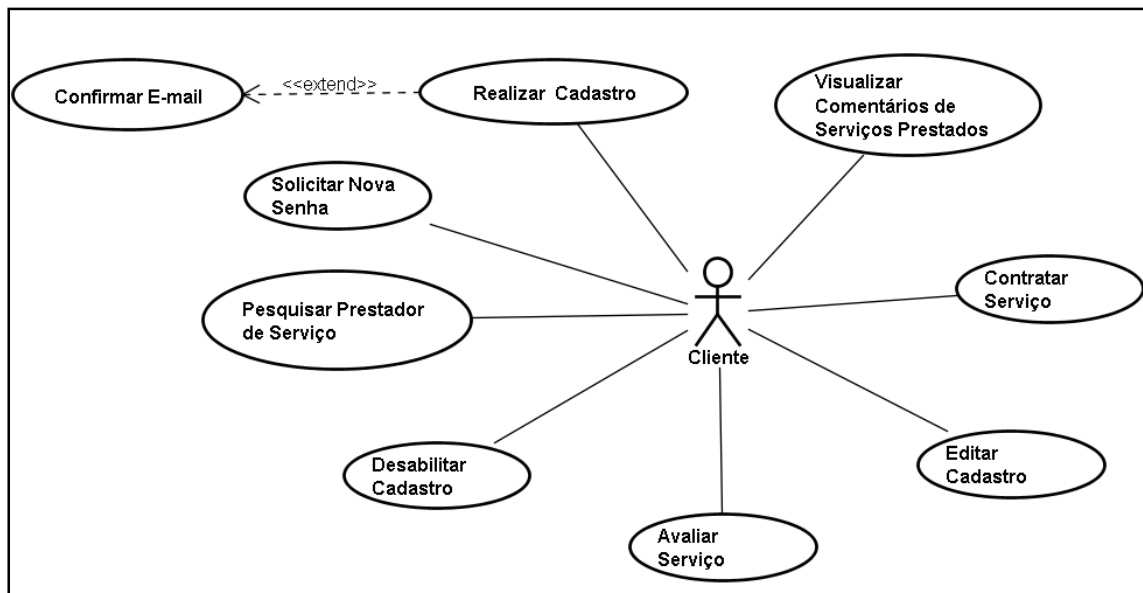
RAMEZ, E.; NAVATHE, S. B. **Sistemas de banco de dados**. 4ª. ed. São Paulo: Pearson Addison Wesley, 2005.

APÊNDICE A

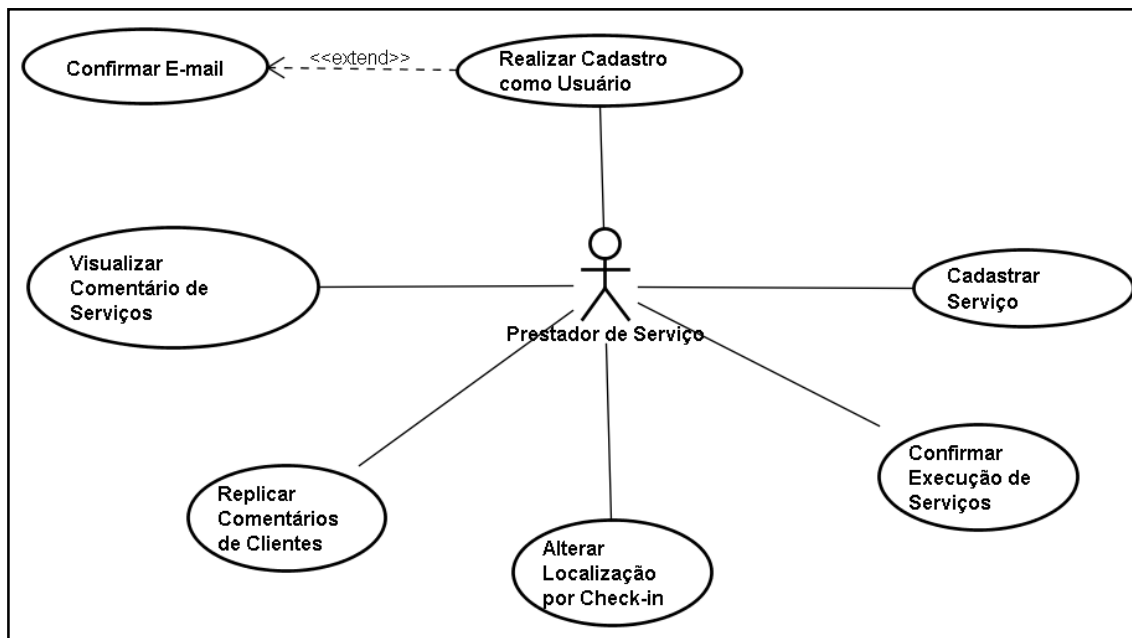
Apresentar o questionário feito é gráfico!

APÊNDICE B

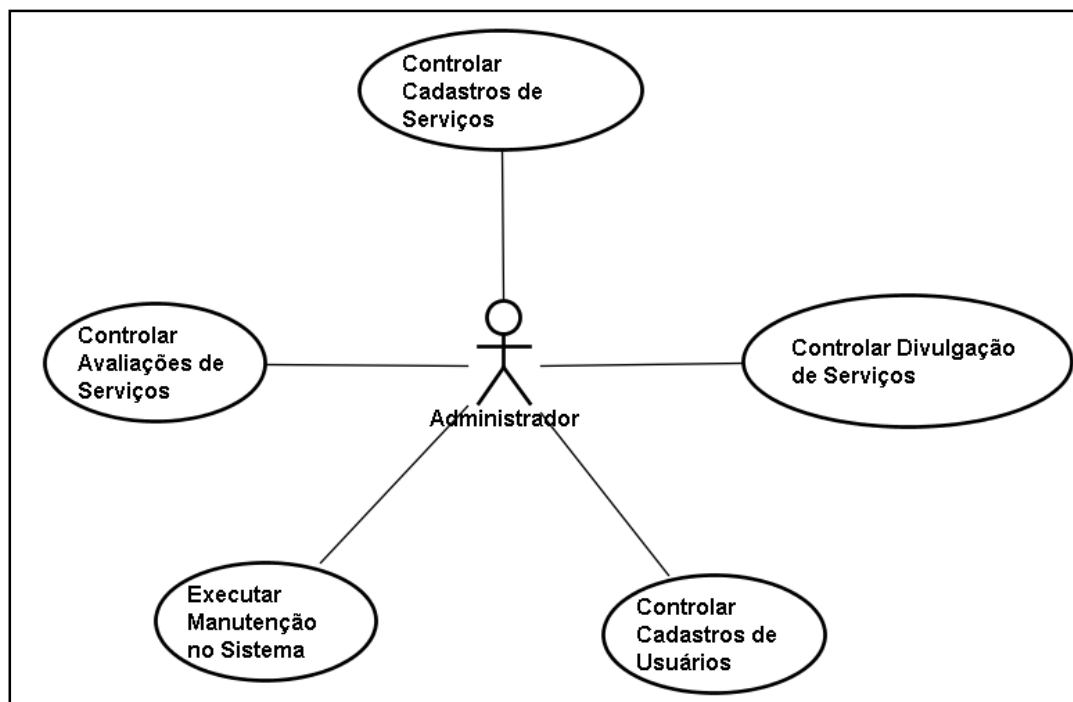
Casos de Uso do Cliente



Casos de Uso do Prestador de Serviço



Casos de Uso do Administrador



Principais Casos de Uso Expandidos

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 página 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. Se for validado com sucesso é redirecionado a página 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.
---------	----------------------------

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 C

Dicionário de Dados

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

Tabela 1: endereco				
Chave Primaria	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 o nome da quadra do endereço.
	rua	VARCHAR	45	Campo responsável por armazenar o nome da rua do endereço.
	numero	VARCHAR	45	Campo responsável por armazenar o número 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 nome do setor do endereço.
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 4.2				

Tabela 2: cidade				
Chave Primaria	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 4.2				

Tabela 3: uf				
Chave Primaria	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 4.2				

Tabela 4: 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 e-mail 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 no banco de dados.
	celular	INT	-	Campo responsável por armazenar o número do celular do prestador de serviço.
	telefone_fixo	INT	-	Campo responsável por armazenar o número de 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_idendereco	INT	-	Campo que contem a referência do id da tabela endereco.
Obs.: essa tabela surgiu através dos requisitos RFC0001, RFC0002 e RFC0012 descrito na seção 4.2				

Tabela 5: 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.
	situação_servico	INT	1	Campo responsável por armazenar qual a situação que o serviço se encontra exemplo: em aberto, finalizado, em andamento.
FK	fk_idusuario	INT	-	Campo que contem a referência do id da tabela usuario.
Obs.: essa tabela surgiu através do requisito RFC0002 descrito na seção 4.2				

Tabela 6: 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.
FK	fk_idservico	INT	-	Campo chave estrangeira contem a referência do id da tabela serviço.
Obs.: essa tabela surgiu através dos requisitos RFC0013 e RFC0010 descrito na seção 4.2				

Tabela 7: servico_tem_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 entra a tabela servico e categoria_servico.				

Tabela 8: 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 4.2				