

UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS
CURSO GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
HELDER NERES SOUZA

**PRINCIPAIS CONCEITOS DE SOA E SUA IMPLEMENTAÇÃO
ATRAVÉS DE WEB-SERVICES**

VITÓRIA DA CONQUISTA
2011

HELDER NERES SOUZA

PRINCIPAIS CONCEITOS DE SOA E SUA IMPLEMENTAÇÃO
ATRAVÉS DE WEB-SERVICES

VITÓRIA DA CONQUISTA
2011

HELDER NERES SOUZA

PRINCIPAIS CONCEITOS DE SOA E SUA IMPLEMENTAÇÃO
ATRAVÉS DE WEB-SERVICES

Monografia apresentada ao curso de Ciência da Computação da Universidade Estadual do Sudoeste da Bahia – UESB, como requisito para obtenção do título de Bacharel em Ciência da Computação.

Banca examinadora:

Orientador:

Roque Mendes Prado Trindade
Universidade Estadual do Sudoeste da Bahia – UESB

Membro:

Hélio Santos Lopes
Universidade Estadual do Sudoeste da Bahia – UESB

Membro:

Maísa Soares dos Santos Lopes
Universidade Estadual do Sudoeste da Bahia – UESB

Agradecimentos

Primeiramente agradeço a Deus, que diretamente me auxiliou no desenvolvimento deste trabalho, dando-me ânimo, paciência e não me permitindo desistir de alcançar o objetivo.

Agradeço também aos meus pais que contribuíram suficientemente, apesar de todas as dificuldades, para eu poder concluir hoje este trabalho. Agradeço também a minha querida Kathy, que tanto me auxiliou e incentivou na realização do mesmo.

Por fim, agradeço aos meus colegas da UINFOR por todo o apoio e suporte, principalmente, a Bruno, Fábio e Leilane, que auxiliaram-me na entrega desta monografia.

Resumo

A Orientação a Serviços tem se difundido consideravelmente nos últimos anos. Grandes fornecedores de Software tem vendido soluções baseadas em SOA, com a promessa de diminuir o impacto das contínuas alterações tecnológicas nos negócios. Este trabalho apresenta uma visão geral deste paradigma e da tecnologia mais adequada para utilizá-lo, a tecnologia de Web Services, além de desenvolver uma solução baseada nestas tendências para integrar um novo sistema que está sendo desenvolvido no contexto na UESB, o sistema para avaliação de professores, ao sistema acadêmico.

PARAVRAS-CHAVE: SOA, Web-Services, UESB

Abstract

Service orientation has been widely spread in recent years. Large software vendors have sold SOA-based solutions, with the promise of reducing the impact of ongoing technological changes in business. This work presents an overview of this paradigm and the most appropriate technology to use it, the Web Services technology, and develop a solution based on these trends to integrate a new system being developed in UESB context, the system for teacher evaluation, to academic system.

Keywords: SOA, Web-Services, UESB

Lista de Figuras

Figura 1: A Orientação a Serviços e os paradigmas e tecnologia anteriores.

Figura 2: Representação de Serviço.

Figura 3: Associação entre Serviço e Processo.

Figura 4: Relacionamento entre serviços.

Figura 5: Comunicação entre serviços.

Figura 6: Esquema de um documento WSDL.

Figura 7: Esquema de mensagem SOAP.

Figura 8: Ciclo de vida de um projeto de SOA.

Figura 9: Exemplo de serviço de tarefa.

Figura 10: Exemplo de serviço de entidade.

Figura 11: Exemplo de serviço utilitário.

Figura 12: Camadas de serviços.

Figura 13: Visão Geral do ambiente de sistemas da UESB.

Figura 14: Diagrama de casos de uso da solução.

Figura 15: Camadas de serviço e serviços candidatos.

Figura 16: Camadas de Serviços após o termino da fase de projeto.

Figura 17: Serviços e suas respectivas capacidades.

Figura 18: Diagrama de classes de entidade.

Lista de Tabelas

Tabela 1: Caracterização dos principais sistemas da UESB

Lista de Abreviaturas

AOP	Aspect Oriented Programming
API	Application Programming Interface
BPM	Business Process Management
EAI	Enterprise Application Integration
HTTP	Hiper Text Transfer Protocol
IDE	Integrated Development Environment
Java EE	Java Enterprise Edition
RPC	Remote Procedure Call
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
UESB	Universidade Estadual do Sudoeste da Bahia
UINFOR	Unidade de Informática da UESB
WSDL	Web Service Description Language
XML	Extensible Markup Language

Sumário

1.	Introdução.....	12
1.1	Objetivos.....	12
1.1.1.	Objetivo Geral.....	12
1.1.2.	Objetivos Específicos.....	13
1.2.	Justificativa.....	13
1.3.	Metodologia.....	14
1.3.1.	Classificação quanto a natureza da pesquisa.....	14
1.3.2.	Classificação quanto a abordagem do problema.....	14
1.3.3.	Classificação quanto aos objetivos.....	14
1.3.4.	Classificação quanto aos procedimentos técnicos.....	15
1.3.5.	Procedimentos realizadas no trabalho.....	15
1.4.	Organização do trabalho.....	16
2.	Conceitos de Orientação a Serviços.....	17
2.1.	Visão Geral de SOA.....	17
2.2.	Arquitetura de Software.....	18
2.3.	Orientação a Serviços.....	19
2.3.1	Serviço.....	19
2.3.2	Serviços encapsulam lógica de negócio.....	21
2.3.3.	Serviços se relacionam.....	21
2.3.4.	Serviços se comunicam.....	22
2.4.	Princípios da Orientação a Serviços.....	23
2.5.	Comparação entre Orientação a Serviços e Orientação a Objetos.....	26
2.6.	Arquitetura Orientada a Serviços.....	27
2.6.1.	Benefícios esperados ao se utilizar SOA.....	29
3.	A tecnologia de Web Services.....	32
3.1.	Visão Geral.....	32
3.2.	Componentes da tecnologia de Web Services.....	33
3.2.1.	Serviço.....	33
3.2.2.	Descrição de Serviço.....	34
3.2.3.	Mensagens.....	35
4.	Processo de entrega de SOA.....	37
4.1.	Camadas de Serviços.....	38
4.1.1.	Serviços de Tarefa.....	39
4.1.2.	Serviço de Entidade.....	39
4.1.3.	Serviço Utilitário.....	40
4.2.	Processos de análise e projeto Orientados a Serviço.....	41
5.	Desenvolvimento da solução.....	43
5.1.	Contextualização.....	43
5.2.	Análise da solução.....	45
5.2.1	Requisitos.....	46
5.2.2.	Lógica de automação já existente.....	47
5.2.3.	Camadas de serviços e serviços candidatos.....	47
5.3.	Projeto da solução.....	48
5.3.1.	Escolha da tecnologia.....	48
5.3.2.	Projeto de camadas e de serviços.....	49
5.4.	Construção da solução.....	51
5.4.1.	Arquitetura.....	51
5.4.2.	Testes e resultado do processo de construção.....	52

6.	Conclusão.....	54
7.	Referências.....	55

1. Introdução

No ambiente inter organizacional atual, tornou-se essencial às empresas responderem de forma efetiva e rápida as constantes mudanças. Do ponto de vista administrativo é necessário se oferecer um produto de qualidade ao número máximo de clientes possível e com baixo custo. Para sobreviverem em tal ambiente, as empresas melhoram os seus processos, investem em inovação, fazem parcerias, etc.

Neste contexto, presume-se, atualmente, que a Tecnologia da Informação(TI) é uma aliada estratégica dos negócios, auxiliando as organizações a alcançarem seus objetivos. Isto é normalmente conhecido como alinhamento entre TI e negócio e tem como objetivo fazer com que a TI consiga responder de maneira ágil e eficaz as necessidades que o negócio venha a ter.

A Arquitetura Orientada a Serviços(SOA) surge neste contexto como resultado da evolução de vários paradigmas e de melhores práticas de TI nas organizações. Em suma, é um modo de se ver várias soluções de automação como um só sistema distribuído que deve atender as necessidades do negócio de maneira rápida, eficaz, barata e que viabilize às organizações serem menos dependentes de aspectos tecnológicos.

A implementação de uma SOA está dissociada de qualquer tecnologia, no entanto, a tecnologia que se mostra mais adequada e relacionada aos conceitos de Orientação a Serviços é tecnologia de Web Services. Esta tecnologia é utilizada para viabilizar a comunicação entre software desenvolvido em diferentes plataformas de desenvolvimento e cada vez mais passa a ser utilizada para desenvolvimento de sistemas distribuídos.

Este trabalho apresenta os principais conceitos destas duas tendências, SOA e Web Services assim como, para exemplificar suas aplicações, desenvolve uma solução para a integração entre o sistema acadêmico da UESB(SAGRES) e um novo sistema, atualmente em desenvolvimento, responsável por automatizar o procedimento de avaliação de professores por parte dos alunos. Este novo sistema é referenciado no trabalho como sistema de avaliação de professores.

1.1. Objetivos

Descrever os objetivos possibilita comunicar a proposta da pesquisa, ou seja, a quais resultados se quer chegar com a mesma.(SILVA e MENEZES, 2005)

1.1.1. Objetivo Geral

Este trabalho tem como objetivo geral apresentar os principais conceitos referentes a SOA e utilizar tais conceitos no desenvolvimento de uma solução para a integração entre determinados sistemas na Unidade de Informática da UESB(UINFOR).

1.1.2. Objetivos Específicos

Os objetivos específicos deste trabalho são os seguintes:

1. Revisão da Literatura;
2. Descrição dos principais conceitos associados com SOA.
3. Descrição dos principais conceitos referentes a tecnologia de Web Services.
4. Desenvolvimento de uma solução baseada em SOA e Web Services que deve permitir a interação entre o sistema acadêmico e o sistema de avaliação de professores da UESB.

1.2. Justificativa

A Arquitetura Orientada a Serviços tem se tornado um padrão na indústria, no que diz respeito a sistemas distribuídos. Muitos pesquisadores da área de SOA afirmam que as corporações devem implantar este padrão com o intuito de obterem maior capacidade de competirem no mercado atual. O SOA-Consortium(2011) apresenta uma lista de casos de sucesso de adoção de SOA, mostrando as principais características na implantação deste casos. Todos estes fatores de certa forma justificam o desenvolvimento deste trabalho pois mostram como SOA tem se tornado uma realidade concreta nas organizações e tem auxiliado a estas organizações terem uma TI que, realmente, está alinhada com o negócio que mantêm. Abordar tal conhecimento vem a ser importante para um crescimento profissional e para a compreensão dos rumos pelo qual o desenvolvimento de sistemas tem seguido.

Uma outro característica que justifica o desenvolvimento deste trabalho vem a ser a dificuldade em encontrar material introdutório, em língua portuguesa, a respeito desse novo modelo. Existe material sobre SOA, no entanto, é voltado, em sua maioria, a quem já tem um conhecimento básico.

Por fim, poder aplicar na UESB os conceitos referentes a este novo paradigma e auxiliar à esta organização com uma nova experiência em desenvolvimento de sistemas distribuídos é também um motivo pelo qual este trabalho tem sido desenvolvido.

1.3. Metodologia

Silva e Menezes(2005) definem pesquisa como “um conjunto de ações, propostas para encontrar a solução para um problema, que têm por base procedimentos racionais e sistemáticos”. As mesmas autoras afirmam que existem algumas formas clássicas de se classificar as pesquisas. A seguir este trabalho será apresentado de acordo com estas classificações.

1.3.1. Classificação quanto a natureza da pesquisa

Quanto a natureza, existem duas classificações para a pesquisa. A pesquisa **básica** é aquela na qual não se tem como objetivo gerar conhecimento sem aplicação prática prevista. Já a pesquisa **aplicada** tem como objetivo a geração de conhecimento que pode ser aplicado na prática e dirigido a solução de um problema específico.(SILVA e MENEZES, 2005)

Este trabalho, por objetivar gerar conhecimento referente a SOA e desenvolver uma solução para um específico problema da UESB, é classificado como pesquisa aplicada.

1.3.2. Classificação quanto a abordagem do problema

Quanto a abordagem do problema, existem duas classificações para as pesquisas. A primeira é a **quantitativa** que considera que tudo pode ser representado em números. Este tipo de pesquisa requer o uso de métodos estatísticos. Já a segunda, a **qualitativa**, considera que existe entre o mundo objetivo e a subjetividade do sujeito um vínculo que não pode ser traduzido em números.

Este trabalho é classificado como pesquisa qualitativa pois não tem como objetivo gerar valores quantitativos em sua análise de SOA e no desenvolvimento do estudo de caso.

1.3.3. Classificação quanto aos objetivos

Quanto aos objetivos, este trabalho pode ser considerado de pesquisa **exploratória**, uma vez que busca proporcionar uma maior familiaridade com o problema tornando-o mais explícito(SILVA e MENEZES, 2005). Para isto, ele é desenvolvido através de pesquisa bibliográfica sobre SOA e estudo de caso, no ambiente de sistemas de informação da UESB.

Existem também outras duas maneiras de se classificar a pesquisa do ponto de vista de

seus objetivos. É possível se classificar uma pesquisa como pesquisa **descritiva**, a qual visa descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis, ou como pesquisa **explicativa**, que busca identificar os fatores que determinam ou contribuem para a ocorrência de determinados fenômenos.(SILVA e MENEZES, 2005)

1.3.4. Classificação quanto aos procedimentos técnicos

Quanto aos procedimentos técnicos realizados para se desenvolver, este trabalho pode ser considerado um trabalho de **pesquisa bibliográfica**, pois é elaborado de material já publicado sobre SOA, em sua maioria livros, e também um trabalho de estudo de caso pois envolve o estudo de um pequeno processo de desenvolvimento de SOA.

Após classificado o trabalho é necessário se identificar o método pelo qual ele alcançará seus objetivos. De acordo com Silva e Menezes(2005), método científico “é o conjunto de processos ou operações mentais que se devem empregar na investigação”. A seguir o procedimento utilizado para desenvolvê-lo é descrito.

1.3.5. Procedimentos realizados no trabalho

Para se alcançar os objetivos esperados, este trabalho foi desenvolvido de acordo com os seguintes passos:

1. Análise da Literatura: Foram procuradas as principais obras relacionadas a SOA, com o intuito de se obter uma firme base teórica sobre o assunto. Destaca-se aqui a presença de várias obras publicadas por Thomas Erl, pesquisador de SOA e do paradigma de Orientação a Serviços, as quais fornecem considerável sustentação a este trabalho.
2. Desenvolvimento dos capítulo 2, 3 e 4 do trabalho, contendo os principais conceitos referentes, respectivamente, a Orientação a Serviços, Web Services e processo de entrega de SOA.
3. Com a base fornecida pelas atividades anteriores tornou-se então viável o desenvolvimento da solução baseada em SOA e Web Service para a integração entre o SAGRES e o sistema para avaliação de professores. Para isto, foi analisado contextualmente o ambiente de sistemas de informação da UESB, obtendo-se assim uma visão geral deste ambiente. Nesta contextualização

buscou-se, secundariamente, a identificação de características que justificassem ou não a adoção de SOA. Por fim, foi realizado o desenvolvimento da solução, através de um processo de entrega de SOA adaptado as necessidades deste projeto, contendo análise e projeto orientada a serviços e construção efetiva da solução. As atividades descritas neste item estão resumidamente descritas no capítulo 5.

1.4. Organização do trabalho

Para facilitar a compreensão do trabalho como um todo, a seguir está descrita a estrutura do documento. Este documento é composto por 7 capítulos.

No capítulo 1 é apresentada a parte introdutória do trabalho, contendo uma introdução do mesmo, juntamente com seus objetivos, justificativa e metodologia utilizadas para desenvolvê-lo.

Nos capítulos 2, 3 e 4 é apresentada a revisão bibliográfica realizada no trabalho. Nestes capítulos são apresentados, respectivamente, conceitos de Orientação a Serviços, de Web Services e o processo de entrega de SOA.

O capítulo 5 apresenta o desenvolvimento do estudo de caso referente a solução desenvolvida para integração entre o SAGRES e o sistema de avaliação de professores.

O capítulo 6 apresenta as conclusões obtidas no trabalho assim como contribuições trazidas por ele e algumas propostas de trabalhos futuros.

Por fim, o capítulo 7 apresenta as referências utilizadas para desenvolvimento da obra.

2. Conceitos de Orientação a Serviços

É fundamental, para se alcançar os objetivos deste trabalho, apresentar o conjunto de ideias associadas a SOA. É este sistema de ideias o meio pelo qual as corporações esperam alcançar sustentação tecnológica necessária para competirem no ambiente comercial atual.

Sendo assim, este capítulo tem como objetivo apresentar os principais conceitos associados a SOA e que se tornam necessários para o desenvolvimento do trabalho, como um todo.

2.1. Visão Geral de SOA

De acordo com Bean(2010), a origem do termo SOA não está suficientemente clarificada. Segundo o mesmo, é racional admitir que SOA surgiu no meio tecnológico, com a evolução de arquiteturas RPC(Remote Procedure Call) e de protocolos de objetos distribuídos, mas também é necessário admitir que SOA surgiu como um meio de se atender ao negócio, auxiliando-o a alcançar seus objetivos.

Um dos requisitos de negócio mais importantes e que a TI poderia prover, em meados da década de 90, era a agilidade. Com o surgimento, entre outras coisas, da Web e do e-commerce, esse requisito de agilidade tornou-se um diferencial. Empresas que se adaptam rapidamente as mudanças no ambiente tem maior capacidade de vencerem na luta do mercado global.(JOSUTTIS, 2007)

SOA surge neste contexto como a evolução da tecnologia associada ao negócio. Um padrão da indústria que permite as organizações alcançarem níveis adequados de qualidade nas atividades de desenvolvimento de soluções de automação, por meio de um conjunto de requisitos técnicos de desenvolvimento, como reusabilidade e baixo acoplamento.

Atualmente, SOA tem se difundido de modo considerável e muitas das grandes empresas de desenvolvimento de plataformas computacionais fornecem soluções baseadas neste modelo. Outros requisitos surgiram, como a necessidade de comunicar com software de organizações parceiras(federação) e a independência de plataformas de fornecedores(neutralidade de fornecedor), mas a ideia básica, a TI estar intimamente associada ao negócio, é a mesma. Erl(2005) lista o conjunto das características atuais de SOA, as quais denomina SOA Contemporâneo:

- SOA está no centro da plataforma de computação orientada a serviços.
- SOA aumenta a qualidade de serviço.

- SOA é fundamentalmente autônoma.
- SOA é baseada em padrões abertos.
- SOA suporta diversidade de fornecedor.
- SOA promove capacidade de descoberta.
- SOA promove federação.
- SOA promove composabilidade arquitetural.
- SOA promove reusabilidade nativa.
- SOA enfatiza a extensibilidade.
- SOA suporta um paradigma de modelagem de negócio orientado a serviço.
- SOA promove baixo acoplamento por toda a organização.
- SOA promove agilidade organizacional.
- SOA é um bloco de construção.
- SOA é uma evolução.
- SOA está ainda em estado de maturação.
- SOA é um ideal alcançável.

Estas são as características que SOA possui. Por conta de sua extensão, analisar cada uma destas está além do escopo deste trabalho. No entanto, as características mais importantes presentes nesta listagem serão analisadas, em geral, com o desenvolver do mesmo.

2.2. Arquitetura de Software:

É importante, para se ter ideia do que é SOA, compreender de fato o que significa o termo arquitetura quando aplicado ao contexto da Ciência da Computação e da Engenharia de Software.

De acordo com Hurwitz et. al.(2007), quando aplicado a ciência da computação, o termo arquitetura descreve a concepção global e a estrutura de um sistema de computador. Já Bass(2003) afirma que: “The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them”.

Estas definições associam a arquitetura de software a uma visão geral das estruturas que compõe o sistema computacional ou, mais especificamente, o sistema de software. Será este o sentido do termo arquitetura para o restante do trabalho.

2.3. Orientação a Serviços

O conceito de Orientação a Serviços é essencial para a compreensão da nova plataforma de aplicações que está se desenvolvendo atualmente, onde organizações se comunicam, interna e externamente, através de software, principalmente por meio da infraestrutura da Web.

Service-orientation is a design paradigm intended for the creation of solution logic units that are individually shaped so that they can be collectively and repeatedly utilized in support of the realization of the specific strategic goals and benefits associated with SOA and service- oriented computing. (ERL, 2008a)

É também do mesmo autor, a definição de Computação Orientada a Serviços:

Service-oriented computing is an umbrella term that represents a new generation distributed computing platform. As such, it encompasses many things, including its own design paradigm and design principles, design patterns, a distinct architectural model, and related concepts, technologies, and frameworks. (ERL, 2008a)

Logo, a Orientação a Serviços é um paradigma de projeto que busca a consecução de objetivos estratégicos específicos através da criação de unidades de solução lógica conhecidas como serviços e que podem ser coletiva e repetidamente utilizadas.

Erl(2007) afirma que um paradigma de projeto, dentro do contexto de automação de negócio, é uma abordagem que rege o projeto de lógica de solução. Sendo um paradigma de projeto, a Orientação a Serviços pode ser comparada a outros paradigmas, como por exemplo, a Orientação a Objetos ou a Orientação a Aspectos. No entanto, a Orientação a Serviços não “compete” com os paradigmas anteriores, ela se aproveita da experiência obtida com estes paradigmas e de padrões utilizados na indústria para formular seus princípios. Na Figura 1 é apresentado um esboço do relacionamento entre a Orientação a Serviços e os paradigmas anteriores.

Segundo Erl(2008a) a Orientação a Serviços é focada, primariamente, em oito princípios. Estes princípios serão descritos no item 2.3 deste trabalho. A seguir serão apresentadas as entidades básicas da Orientação a Serviços.

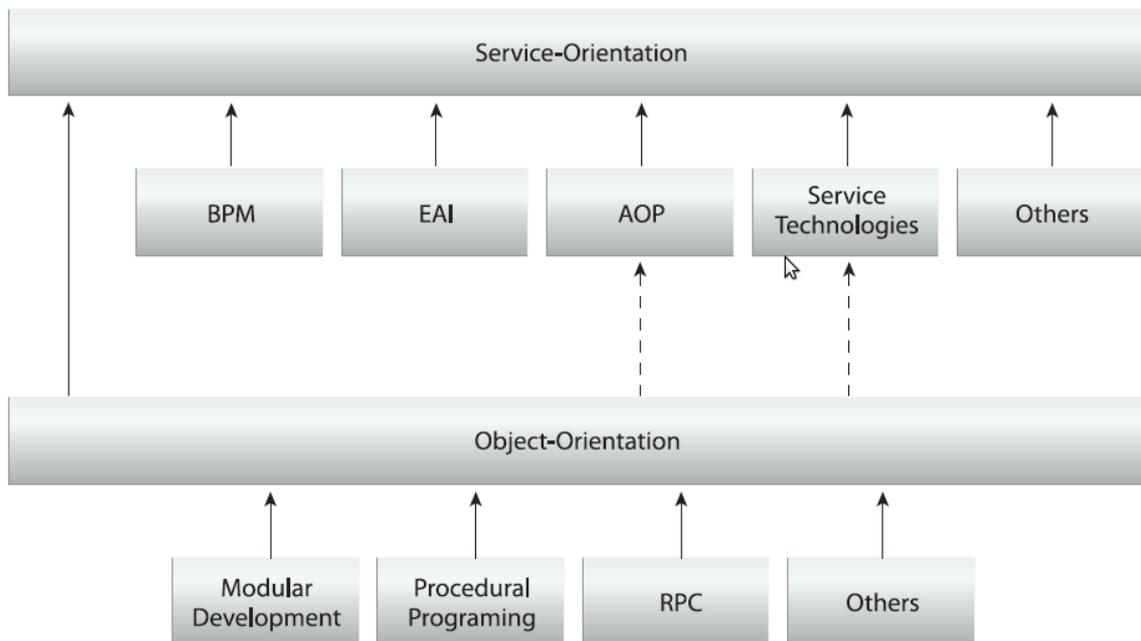
2.3.1 Serviço

O conceito de serviço é o conceito base para o paradigma de Orientação a Serviços. De acordo com o SOA Work Group(2011), um serviço é “a logical representation of a repeatable business activity that has a specified outcome”. Esta definição segue direção

semelhante a de Hurwitz et. al.(2007), que afirma que um serviço é “the logical encapsulation of business function”. Nestas definições, percebe-se o enfoque na abstração que fornece o conceito de serviço. Tal abstração é necessária ao se pensar em um modelo para representação de um mini-mundo. É inevitável fazer uma analogia de serviços em SOA, com a ideia de serviço disponível atualmente no dia a dia. Em uma cidade, pessoas e organizações disponibilizam serviços das mais variadas formas, formando uma grande cadeia. Em outras palavras, atividades repetitivas são efetuadas com o intuito de prover um resultado específico. Exemplos são inúmeros: serviço de correio, serviços básicos, como o de provimento de energia, de água, luz, saneamento e transporte, serviço de internet, serviço de contabilidade, serviço de supermercado, serviço de lavanderia, etc.

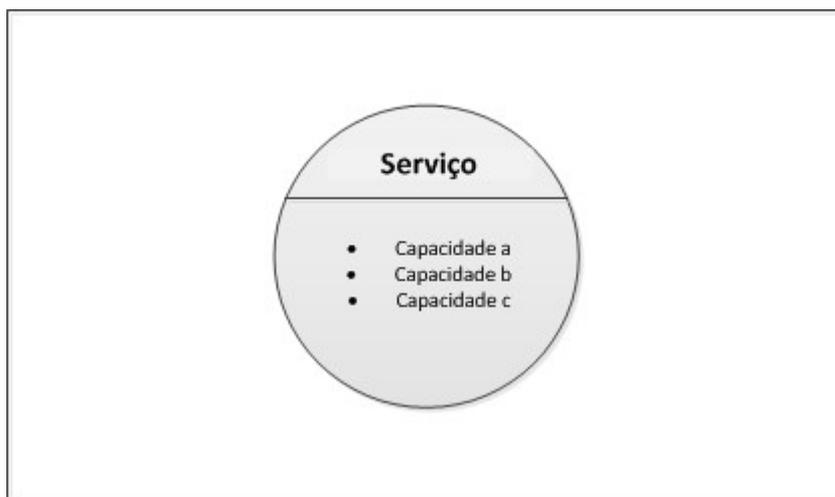
Figura 1 – A Orientação a Serviços e os paradigmas e tecnologia anteriores

Fonte: (ERL, 2008a)



Uma terceira definição de serviço encontrada na Literatura é a definição de Erl(2007), que afirma que serviços são coleções de capacidade. O termo capacidade, nesta definição, pode ser utilizado em seu sentido mais comum. Logo, definir serviços está diretamente associado a tarefa de relacionar capacidades em um contexto funcional. Por conta da simplicidade e objetividade desta definição, este será o conceito de serviço utilizado para o restante do texto, quando este termo estiver relacionado a SOA e ao paradigma de Orientação a Serviços. A Figura 2 representa bem esta definição. A seguir, serão descritas algumas das características mais importantes de serviços.

Figura 2 – Representação de Serviço



2.3.2. Serviços encapsulam lógica de negócio

Para manterem sua independência, serviços encapsulam lógica dentro de um contexto distinto. Tal lógica pode representar uma tarefa de negócio, uma entidade do mesmo, ou ter outro agrupamento (ERL, 2005). Ao representar tarefas de negócio, um serviço pode ser associado ao conceito de processo. Assim como um processo tem a capacidade de representar um conjunto variado de tarefas, sendo que uma parte destas tarefas podem ser vistas também como processos, o serviço tem a mesma característica, ou seja, serviços podem ser compostos por outros serviços. A Figura 3 representa essa flexibilidade inerente ao conceito de serviço.

Segundo o SOA Work Group (2011), uma outra propriedade dos serviços é que eles espelham atividades de negócio do mundo real. Essa associação entre processos e serviços é fundamental para SOA atingir um dos seus principais objetivos – alinhar a TI ao negócio. Estando os conceitos utilizados para a modelagem da solução tão bem relacionados e parecidos com os reais conceitos do negócio, torna-se possível a TI se adequar aos requisitos de tempo e agilidade existentes e as mudanças que ocorrem no negócio, sendo possível à ela, até mesmo, direcionar tais mudanças em busca de redução de custos, melhorias de processos organizacionais, maior qualidade nos produtos gerados, etc.

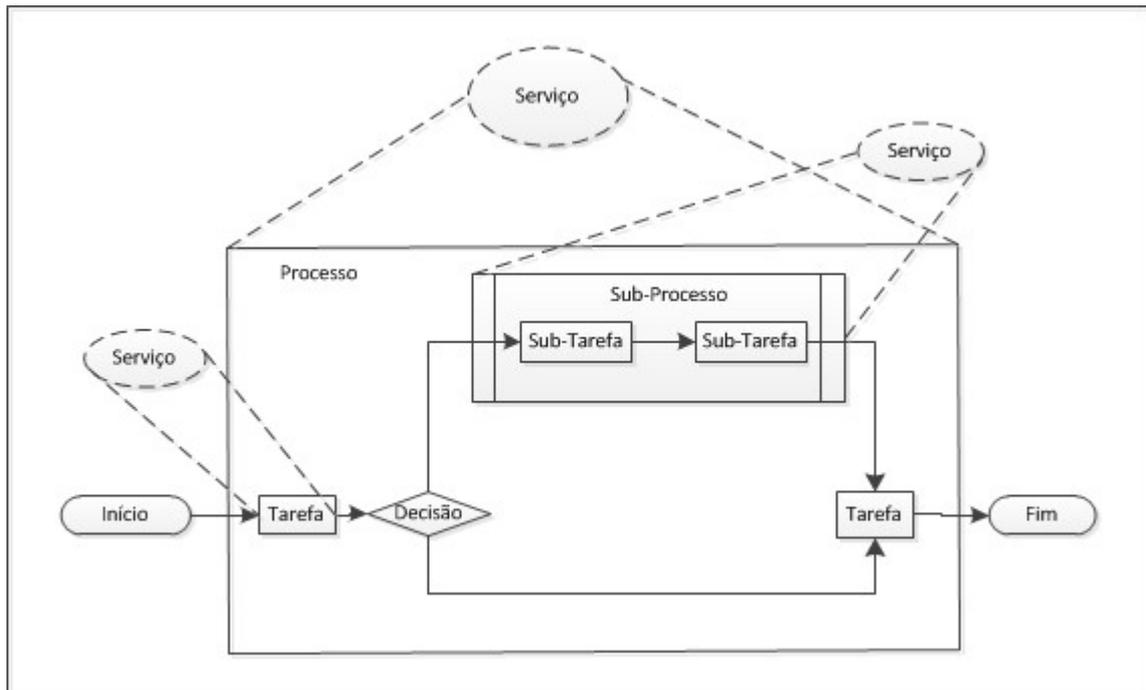
2.3.2. Serviços se relacionam

Segundo Erl (2005), serviços podem ser utilizados por outros serviços ou programas. Para que serviços interajam, é necessário que um serviço que necessite relacionar-se com outro conheça o que este outro serviço dispõe. O meio pelo qual torna-se possível a um

serviço expor suas capacidades e outros serviços tomarem conhecimento destas é a **descrição de serviço**. A partir do momento em que um cliente tem acesso a uma descrição de serviço, torna-se possível a ele se utilizar das capacidades do serviço, com algum objetivo.

Pode-se perceber que o acoplamento gerado no relacionamento é baixo. No caso da Figura 4, o serviço B não sabe que é o serviço A que lhe fornece uma operação, pois tem acesso somente a descrição do serviço.

Figura 3 – Associação entre Serviço e Processo



2.3.3. Serviços se comunicam

Para serviços interagirem e realizarem algo significativo, eles devem trocar informações (ERL, 2005). Para manterem um nível de acoplamento o mais baixo possível, as mensagens devem ter a capacidade de operarem independentemente do contexto que encontram-se. Essa necessidade de baixo acoplamento é representada na Figura 5.

Ao enviar uma mensagem a seu destino, um serviço perde o controle sobre o que acontece com ela. Uma analogia interessante a se fazer aqui é com o serviço de correio. Ao se enviar uma carta, o emissor perde o controle sobre o que ocorre com a mesma. Para que o correio possa entregar a carta a seu destino é necessário se preencher um envelope contendo somente a informação necessária para a entrega, sendo que o conteúdo da mensagem está encapsulado dentro deste.

Serviços que forneçam descrições e que se comunicam através de mensagens formam

uma arquitetura básica para SOA. No entanto, possuir tais componentes em uma arquitetura não significa possuir uma arquitetura deste tipo. O que define uma arquitetura como SOA é a forma pela qual estes componentes básicos vão ser projetados.(ERL, 2005)

Figura 4 – Relacionamento entre serviços

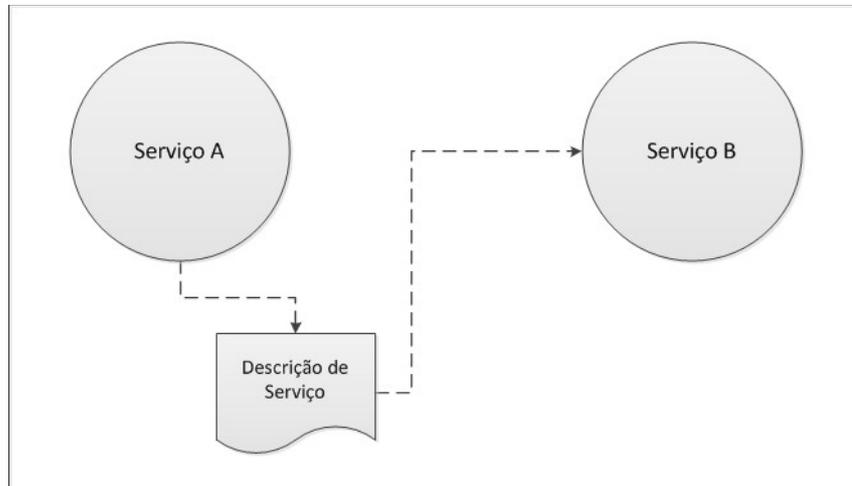
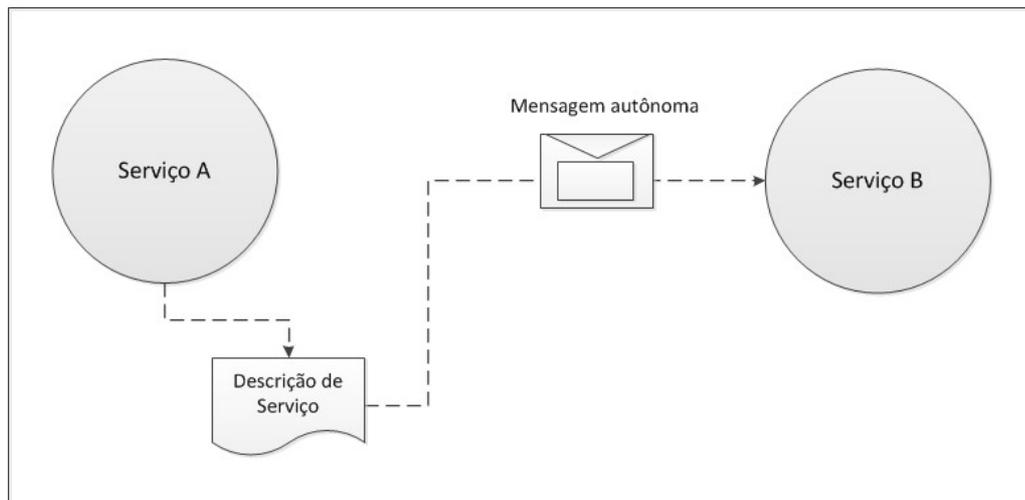


Figura 5 – Comunicação entre serviços



São os princípios da Orientação a Serviços que iram fornecer meios para se desenvolver SOAs.

2.4. Princípios da Orientação a Serviços:

Um princípio presente nas Engenharias é o “dividir para conquistar”. Ao se deparar com um problema grande e complexo é necessário dividi-lo em problemas menores para resolvê-lo. A Engenharia de Software se baseia demasiadamente neste princípio, sendo que

seus métodos focam em como dividir o problema e como reutilizar as soluções.

O princípio da divisão e conquista é utilizado em vários paradigmas de desenvolvimento. Erl(2005) afirma que a Orientação a Serviços é uma nova forma de se aplicar este princípio, afirmando também que ao se analisar individualmente as características de SOA, pode-se perceber direta ou indiretamente a aplicação deste princípio. Tais características, de acordo com Erl(2007), estão descritas nos oito princípios da Orientação a Serviços.

Estes princípios, que permitem à arquiteturas geradas em um projeto orientado a serviços possuírem as características de SOA, são descritos a seguir:

- **Serviços compartilham contratos padronizados:** Serviços dentro do mesmo inventário estão em conformidade com os mesmos padrões de projeto de contrato.

Os serviços, suas capacidades e suas mensagens são formalmente definidas através de contratos. De acordo com o autor, um inventário de serviços “is an independently standardized and governed collection of complementary services within a boundary that represents an enterprise or a meaningful segment of an enterprise”(ERL, 2007). Como consequência deste princípio, serviços podem inter-operar dentro de um determinado contexto, pois existe um contrato pelo qual fornecedores de serviço e consumidores de serviço baseiam-se.

- **Serviços são fracamente acoplados:** Contratos de serviço impõe requisitos de baixo acoplamento para os consumidores e eles mesmo são desacoplados do ambiente que os circundam.

Aos consumidores de determinado serviço somente é necessário conhecer o contrato de serviço. Como consequência, se o fornecedor for alterado, mas o contrato continuar o mesmo, não será necessário ocorrer nenhuma alteração no consumidor.

- **Informação de serviço não essencial é abstraída:** Contratos de serviços contêm apenas informações essenciais e a informação sobre os serviços é limitada ao que é publicado nos contratos de serviços.

É através deste princípio que um serviço pode se comportar como uma caixa-preta, escondendo do ambiente externo a lógica que implementa. Portanto, pode-se desenvolver um serviço para realizar uma tarefa extremamente complexa, de alta granularidade, como por exemplo a execução de um processo de autorização de uso de um cartão de crédito, assim como é possível desenvolver um serviço para um tarefa mais simples, como por exemplo, informar se a data de validade do mesmo cartão não se passou.

- **Serviços são reusáveis:** Serviços contém e expressam lógica agnóstica e podem ser classificados como recursos empresariais reutilizáveis.

Segundo o SOAGlossary(2011) uma lógica agnóstica é uma “logic that is sufficiently generic so that it is not specific to (has no knowledge of) a particular parent task”. Serviços tem requisitos de reusabilidade e, portanto, devem ser projetados para se tornar o máximo possível independentes de contexto.

- **Serviços são autônomos:** Serviços exercem um alto nível de controle sobre suas plataformas, em tempo de execução.

Este princípio garante que serviços possam executar sua lógica independentemente de fatores externos ou mesmo de outros serviços. Segundo Erl(2005), a autonomia é uma das considerações primárias ao se dividir a lógica de aplicação e as capacidades entre os serviços.

- **Serviços minimizam a manutenção de estado:** Serviços minimizam o consumo de recursos por retardarem o gerenciamento de informações de estado até que este seja necessário.

Minimizar informação referente ao estado permite aos serviços estarem mais disponíveis aos seus consumidores quando um serviço é solicitado. Isto é, em si, resultado de um aumento na escalabilidade no provimento do serviço. Além disso, ao minimizarem as informações de estado que armazenam, serviços tornam-se mais reutilizáveis.

- **Serviços são descobríveis:** Serviços são complementados com metadados de comunicação pelos quais eles podem ser efetivamente descobertos e interpretados.

Consumidores de serviço podem encontrar em um inventário algum serviço que forneça a capacidade que buscam. Para isso, serviços possuem informações complementares sobre o que fazem e onde estão.

- **Serviços são componíveis:** Serviços são efetivamente participantes de composições, independentemente do tamanho e da complexidade destas.

Este princípio permite a um serviço se utilizar de outros com o intuito de executar sua lógica. Como consequência, serviços podem ser projetados e implementar lógica complexa através de capacidades de outros serviços.

Estes são os oito princípios da Orientação a Serviços, como proposto por Erl(2007). Vale a pena ressaltar que aquilo que este autor descreve como princípios da Orientação a Serviços, Jossuttis(2007) descreve como atributos adicionais dos serviços. Erl quer destacar ao definir tais princípios que apesar de se usar tecnologias orientadas a serviço, como o caso

de Web Services, não necessariamente o produto de software desenvolvido pode ser considerado orientado a serviços. Outro ponto que este autor quer por em destaque é que para se obter, de fato, uma arquitetura orientada a serviços, obtendo completamente seus benefícios, é necessário que o projeto de serviços seja direcionado por estes princípios.

2.5. Comparação entre Orientação a Serviços e Orientação a Objetos

Após a apresentação dos princípios da Orientação a Serviços, torna-se possível uma comparação entre este paradigma e o paradigma mais difundido nos dias atuais, o da Orientação a Objetos. Como já visto anteriormente, alguns princípios e melhores práticas da Orientação a Objetos foram adotados pela Orientação a Serviços, até por se tratar de uma evolução de paradigmas. Portanto, uma boa comparação entre estes paradigmas tende a ser aquela que destaca onde a Orientação a Serviços foi além da Orientação a Objetos e desenvolveu algo de novo. Para tanto, logo a seguir, cada um dos princípios da Orientação a Serviços será posto em uma perspectiva da Orientação a Objetos.(ERL, 2005)

Serviços compartilham contratos padronizados: Este princípio da Orientação a Serviços é bastante análogo ao princípio de projeto da Orientação a objetos de programar para interfaces. Tendo se tornado uma melhor prática no contexto da Orientação a Objetos, por facilitar o desenvolvimento e manutenção de sistemas, esta abordagem foi reaproveitada na Orientação a Serviços.

Serviços são fracamente acoplados: A Orientação a Objetos também fornece à Orientação a Serviços a experiência de se desenvolver unidades de lógica que sejam fracamente acopladas entre si. No entanto, a Orientação a Serviços eleva este princípio a um patamar maior que o paradigma anterior o faz. Utilizar de interfaces em Orientação a Objetos reduz o acoplamento entre objetos em sistema, no entanto, outros princípios deste paradigma, como a herança, acabam por tornar esse relacionamento acoplado. A Orientação a Serviços desvia desta perspectiva e estabelece, como um objetivo inicial de serviços, serem o máximo possível desacoplados de outros serviços no ambiente em que existem.

Informação de serviço não essencial é abstraída: Assim como na Orientação a Objetos, onde o princípio do encapsulamento permite que um objeto esconda do ambiente a lógica que executa para operar e fornece a este ambiente, somente uma interface, pela qual outros objetos podem trocar mensagens, a Orientação a Serviços possibilita a um serviço abstrair a lógica que opera e fornece a outros serviços através do seu contrato de serviços.

Serviços são reusáveis: É observado na Orientação a Objetos que o desejo de se

desenvolver unidades de processamento de lógica altamente reusáveis é buscado. Isso é perceptível nos princípios da modularidade e do encapsulamento, encontrado neste paradigma. A Orientação a Serviços mantém este desejo, afirmando que serviços, desde o seu projeto, devem ser reusáveis.

Serviços são autônomos: A autonomia é mais enfatizada na Orientação a Serviços do que na Orientação a Objetos. O fraco acoplamento entre serviços permite alcançar um nível de independência maior entre os serviços, quando se comparado com objetos, uma vez que dependências relacionadas a herança, entre outras, diminuem a autonomia de determinado objeto em relação ao ambiente.

Serviços minimizam a manutenção de estado: Diferentemente de objetos, que são um agregado de operações e dados e como consequência quase sempre mantém o seu estado, serviços evitam ao máximo manter esse tipo de informação. Apesar de ser possível criar serviços que sempre armazenem o estado e objetos que nunca fazem isso, a Orientação a serviços apresenta neste princípio, um enfoque consideravelmente diferente do paradigma anterior.

Serviços são descobríveis: Projetar interfaces que sejam consistentes e auto-descritivas é uma outra melhor prática da Orientação a Objetos que a Orientação a Serviços estende. Contratos de serviços também são desenhados para serem facilmente compreendidos e encontrados, no entanto, esta preocupação não se restringe ao projeto, ocorrendo também durante a execução de serviços em um inventário.

Serviços são componíveis: A Orientação a Serviços, em contexto fracamente acoplado, suporta conceitos de associação, como a Orientação a Objetos faz através da agregação e da composição. Logo, é possível utilizar de vários serviços para se implementar determinada lógica, como é feito em Orientação a Objetos.

Através da análise dos princípios da Orientação a Serviços e de sua comparação com a Orientação a Objetos, foi demonstrado como a Orientação a Serviços é, em suma, uma extensão deste outro paradigma. Logo, eles não são paradigmas que estão em competição, muito pelo contrário, as soluções orientadas a serviços desenvolvidas são, tipicamente, uma mistura de componentes orientados a objetos e serviços.(ERL, 2005)

2.6. Arquitetura Orientada a Serviços:

O termo Arquitetura Orientada a Serviços tem sido utilizado das mais variadas formas nos últimos anos. Existem muitas definições disponíveis para SOA, no entanto, muitas delas

são imprecisas ou até mesmo equivocadas. Erl et. al.(2008b) argumenta que o termo Arquitetura Orientada a Serviços, historicamente, foi tão difundido na literatura de marketing de fornecedores de tecnologia e na mídia que quase tornou um sinônimo de computação orientada a serviços. O mesmo autor define SOA da seguinte maneira:

Service-oriented architecture represents an architectural model that aims to enhance the agility and cost-effectiveness of an enterprise while reducing the overall burden of IT on an organization. It accomplishes this by positioning services as the primary means through which solution logic is represented. SOA supports service-orientation in the realization of the strategic goals associated with service-oriented computing. (ERL et. al., 2008b)

É do mesmo autor esta outra definição de SOA:

"Service-oriented architecture" is a term that represents a model in which automation logic is decomposed into smaller, distinct units of logic. Collectively, these units comprise a larger piece of business automation logic. Individually, these units can be distributed. (ERL, 2005)

O Open Group(2011) define SOA de forma bem simples, sendo um estilo arquitetural que suporta orientação a serviços. Uma outra definição, mais abstrata de SOA é dada por Josuttis:

SOA is not a concrete architecture: it is something that leads to a concrete architecture. You might call it a style, paradigm, concept, perspective, philosophy, or representation. That is, SOA is not a concrete tool or framework you can purchase. It is an approach, a way of thinking, a value system that leads to certain concrete decisions when designing a concrete software architecture. (JOSUTTIS, 2007, p 29)

Analisando-se as definições, pode-se notar como o conceito de SOA está vinculado a Orientação a Serviços. Serviços são os meios básicos pelos quais pode-se construir uma SOA. Sendo assim, uma arquitetura orientada a serviços é aquela desenvolvida tendo serviços como unidades que encapsulam lógica para automação de tarefas e tendo aplicados princípios próprios da Orientação a Serviços. Está é a concepção de SOA tomada para este trabalho.

Ao se utilizar estes princípios, são esperados benefícios por toda a arquitetura de software de uma organização.

2.6.1. Benefícios esperados ao se utilizar SOA

Os benefícios mais comumente obtidos através de SOA são listados a seguir. Estes benefícios são sensivelmente perceptíveis no momento em que a organização alcança um certo grau de maturidade no desenvolvimento e utilização desta arquitetura. No entanto, mesmo em níveis iniciais de implantação, alguns destes benefício já podem ser parcialmente observados.(ERL, 2008)

- **Melhor integração:** SOA pode resultar na criação de soluções que consistem naturalmente de serviços inter operáveis. Inter operabilidade vem a ser a capacidade que um conjunto de soluções de automação possui de se comunicar para a execução de suas operações. Como SOA permite a utilização de frameworks de mensagens independentes de fornecedor, existe potencial para as empresas implementarem descrições de serviço e estruturas de mensagens altamente padronizadas. O resultado geral desta atividade vem a ser, ao passar do tempo, uma diminuição considerável nos esforços para desenvolvimento de soluções de integração, tornando-se esta tarefa mais semelhante a uma tarefa de modelagem.
- **Reuso nativo:** A Orientação a Serviços promove o projeto de serviços que sejam nativamente reutilizáveis. Como serviços são projetados para serem reusáveis, existe grande possibilidade de se aproveitar soluções orientadas a serviços já existentes. Como consequência, a medida que a organização vai alcançando maturidade com o uso de SOA, os esforços e custos associados ao desenvolvimento de novas soluções tende a diminuir.
- **Arquiteturas e soluções racionalizadas:** O princípio da composabilidade da Orientação a Serviços, quando aplicada a um nível de infra estrutura, permite a criação de ambientes de automação altamente otimizados onde somente as tecnologias necessárias tornam-se parte da arquitetura.
- **Aproveitamento do investimento legado:** Como SOA promove o desenvolvimento de soluções fracamente acopladas, independentes de tecnologia de implementação e que são naturalmente inter operáveis, torna-se possível se reutilizar soluções legadas, fornecendo a lógica de negócio presente nestas, através de serviços. O custo e o esforço para se integrar a solução legada são consideravelmente diminuídos, assim como a necessidade

de se substituir software deste tipo.

- **Estabelecimento de uma representação de dados padronizada em XML:** Nos níveis mais fundamentais SOA é construído e dirigido por XML. Com isso, SOA leva a um grande aproveitamento da plataforma de representação de dados desta tecnologia. A medida que as representações de dados são construídas através de XML, o modelo vai sendo refinado, obtendo-se um padrão para toda a organização. Neste caso, o que é descrito como XML vai se transformando em ontologia referente ao domínio da empresa. Isto diminui os esforços com desenvolvimento, tanto em análise, como em projeto e documentação.
- **Investimento focado em infra estrutura de comunicação:** SOA permite, por meio de sua inter operabilidade, que as organizações invistam somente em uma tecnologia para fazer comunicação entre os sistemas que possui. O custo do escalamento da infra estrutura de comunicação é diminuído, pois somente um padrão tecnológico é necessário para o provimento de uma federação. Heil(2009) referindo-se a um artigo da Sun não mais disponível, define federação como “groups of devices and software components organized into a single, dynamic distributed system ”.
- **Melhores alternativas de desenvolvimento:** Como SOA se utiliza de um framework de comunicação independente de fornecedor, a TI torna-se independente de determinada plataforma de middleware ou de tecnologia proprietária. A inter operabilidade de SOA também promove a utilização da melhor alternativa em determinado contexto, podendo-se considerar a tecnologia em si, políticas da organização, questão de licenciamento, etc. SOA permite a TI ter maior flexibilidade e, conseqüentemente, capacidade para atender os requisitos organizacionais.
- **Agilidade Organizacional:** Talvez este seja o benefício mais esperado ao se implantar SOA. A inter operabilidade, a composabilidade, a reusabilidade e o baixo acoplamento promovidos pela Orientação a Serviços auxiliam bastante na agilidade que a organização precisa ter com sua TI. SOA promove o desenvolvimento de uma arquitetura em camadas, onde os dois principais domínios da organização, o de lógica de negócio e o de aplicação estão fracamente acoplados. Como consequência, desenvolver novas soluções tende

a ser um processo mais ágil, com reutilização de lógica de negócio encapsulada em serviços de negócio, ou mesmo, em serviços de aplicação já existentes.

Sendo SOA um paradigma, um conjunto de ideias, é necessário um meio concreto pelo qual ela pode ser implementada. Existem algumas tecnologias pelas quais isto é possível. Este trabalho irá abordar a mais utilizada, difundida e adequada delas, a tecnologia de Web Services conforme Erl(2005).

3. A tecnologia de Web Services

Como já referido anteriormente, o sucesso obtido pela Internet e pela Web trouxe a necessidade de um meio mais homogêneo, pelo qual aplicações disponibilizadas por diferentes pessoas e organizações pudessem se comunicar sem a necessidade de adaptação entre as tecnologias utilizadas pelas partes. Algumas tentativas de se resolver esse problema foram pensadas e desenvolvidas mas nenhuma alcançou tanto sucesso como a tecnologia de Web Services.

Esta tecnologia tem sido amplamente utilizada para permitir interoperabilidade entre plataformas de software desenvolvidas por diferentes fornecedores. Erl(2005) afirma que a tecnologia de Web Service é a mais adequada e bem sucedida em se aplicar SOA: “The term "service-oriented" and various abstract SOA models existed before the arrival of Web services. However, no one technology advancement has been so suitable and successful in manifesting SOA than Web services”. De fato, os conceitos, modelos e estruturas disponibilizadas pela tecnologia de Web Services podem ser facilmente adaptados para refletirem os princípios da Orientação a Serviços.

Este capítulo tem como objetivo apresentar os principais conceitos relacionados a esta tecnologia.

3.1. Visão Geral

A tecnologia de Web Service é desenvolvida e padronizada pelo W3C e disponibilizada como um framework contendo modelos, arquiteturas, conceitos e sub-frameworks(ERL 2005). Tidwell , Snell e Kulchenko(2001) afirmam que o framework de Web Services representa a evolução dos princípios que guiam a Internet há anos.

Este framework foi amplamente aceito pela indústria e tornou-se um padrão de fato. Inúmeras corporações se utilizam dele para desenvolvimento de soluções de automação tanto em ambiente interno como em ambiente externo. Ele tem sido utilizado em uma série de aplicações e muitas plataformas fornecem suporte nativo a ele. Como exemplo, pode-se citar o Java EE e o .NET, talvez as plataformas de desenvolvimento mais utilizadas no mercado corporativo.

O W3C(2004) define um web service como “a software system designed to support interoperable machine-to-machine interaction over a network”. No mesmo documento, a W3C afirma que um web service é uma noção abstrata que deve ser implementada por

determinado agente. Um agente vem a ser um software ou hardware que envia e recebe mensagens, enquanto que o conjunto de funcionalidades fornecidas vem a ser o web service. Essa consideração permite um fraco acoplamento entre a interface provida pelo web service e sua implementação, através do agente. Pode-se ver aqui uma aplicação do princípio mais básico da Orientação a Serviços.

A seguir, serão abordados os principais componentes que compõe esta tecnologia.

3.2. Componentes da tecnologia de Web Services

O framework de Web Services foi desenvolvido, basicamente, sobre duas tecnologias padronizadas para o ambiente Web: o HTTP, protocolo utilizado para transferência de Hiper Texto e o XML, uma meta linguagem de marcação muito utilizada, entre outras coisas, para a definição de metadados. Ao se desenvolver uma aplicação de Web Services irá se utilizar muito do XML para se definir os componentes da aplicação.

De um ponto de vista estrutural, a tecnologia de Web Services é semelhante ao modelo proposto pela Orientação a Serviços. Uma aplicação em Web Services é composta por **serviços** disponibilizados através de **descrições** e que se comunicam através de **mensagens**. Estes componentes serão descritos na sequência.

3.2.1. Serviço

O W3C(2004) define que um serviço “is an abstract resource that represents a capability of performing tasks that represents a coherent functionality from the point of view of provider entities and requester entities”. Logo, um serviço disponibiliza certa quantidade de lógica, que é capaz de operar, a seus consumidores.

Erl et. al.(2008b) define o componente de serviço para o framework de Web Service como “a body of solution logic that provides a physically decoupled technical contract consisting of a WSDL definition and one or more XML Schema and/or WS- Policy definitions”. Por ser um corpo de solução lógica e por fornecer um contrato fisicamente desacoplado, o conceito de serviço em Web Services é muito semelhante ao conceito de serviço na Orientação a Serviços, estando diretamente vinculado a este. Como consequência, **ao se utilizar Web Services como tecnologia de implementação para SOA**, pode-se intercambiar os conceitos de serviço sem maiores prejuízos. Assim, as características de serviço aplicadas a SOA são refletidas no framework de Web Services.

3.2.2. Descrição de Serviços

De acordo com o W3C(2004) um descrição de serviços é “a set of documents that describe the interface to and semantics of a service”. As descrições de serviço permitem a um relacionamento entre fornecedor e consumidor de serviço ser de fraco acoplamento. O fornecedor publica uma descrição, ou contrato de serviço, na qual define um conjunto de metadados sobre o serviço. Estes metadados incluem(ERL 2008b):

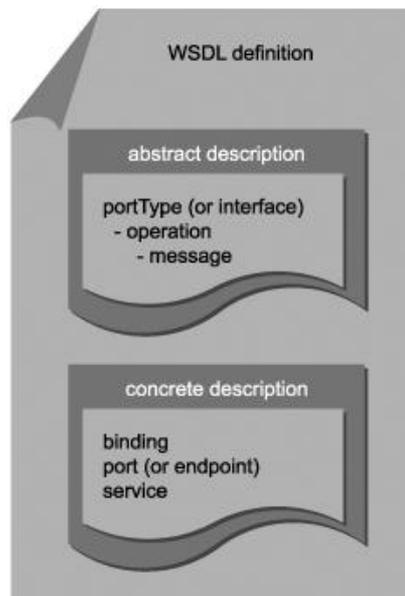
- A proposta e funções de suas operações.
- As mensagens que necessitam ser trocadas.
- Modelos de dados usados para definir a estrutura das mensagens.
- Um conjunto de condições sobre as quais as operações são fornecidas.
- Informação sobre como e onde o serviço pode ser acessado.

Estas informações são descritas através de documentos WSDL(Web Service Description Language), XML Schema e WS-Policy. O documento WSDL passa a ser o ponto de contato entre o provedor e o consumidor do serviço.

Em um documento WSDL existem dois tipos de descrição, uma abstrata, independente de implementação e outra concreta contendo detalhes específicos desta. Através desta separação obtêm-se baixo acoplamento entre a interface que o serviço disponibiliza e sua implementação em determinada plataforma. A Figura 6 contém um esboço de um documento WSDL.(ERL, 2005)

Figura 6 – Esquema de um documento WSDL

Fonte: (ERL, 2005)



A descrição abstrata é composta por elementos portType(ou interface), operation e message. O elemento portType agrupa as operações que são disponibilizadas pelo serviço. O elemento operation também é um encapsulamento, só que de mensagens que podem ser trocadas entre o fornecedor e o consumidor do serviço. Já o elemento message define uma forma pela qual é possível aos serviços se comunicarem. Em suma, mensagens são agrupadas em operações que são agrupadas em interfaces.

A descrição concreta é composta por elementos binding, port(ou endpoint) e service. O elemento binding detalha a tecnologia de comunicação que pode ser utilizada pelo consumidor para se conectar ao provedor do serviço. O elemento port está associado com o elemento binding, definindo o endereço físico pelo qual o serviço pode ser acessado em determinado protocolo. Já o elemento service torna possível se agrupar vários elementos port para o serviço.

O XML Schema é utilizado para formalizar a estrutura das informações trocadas pelo provedor e pelo consumidor dos serviços. Em outras palavras, é utilizado para definir como são estruturados os dados esperados como entrada e enviados como saída pelo fornecedor do serviço.

O WS-Policy permite definir políticas para o serviço. Políticas podem fornecer regras, preferências e processar detalhes que não podem ser efetuados através da descrição de serviço.

3.2.3. Mensagens

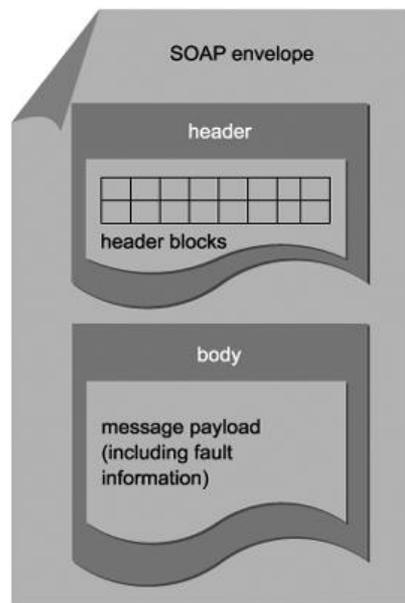
O W3C(2004) define mensagem como “the basic unit of data sent from one Web services agent to another in the context of Web services”. Serviços trocam mensagens com o intuito de se comunicarem. Assim como é esperado que as mensagens na Orientação a Serviços auxiliem no baixo acoplamento entre fornecedores e consumidores de serviço, este é um objetivo de mensagens no framework de Web Service. Para isto, as mensagens encapsulam certos metadados que, juntamente com um protocolo para esta comunicação, o SOAP(Simple Object Access Protocol), possibilitam-nas serem independentes de contexto. Tais mensagens, trocadas através deste protocolo, possuem a estrutura descrita na Figura 7.

Em SOAP, o conteúdo das mensagens é encapsulado por um envelope, no qual existem duas partes. A primeira, o cabeçalho, contém meta informação sobre a comunicação. Assim é possível se retirar dos serviços a responsabilidade de manter determinados estados da comunicação, assim como lógica de roteamento, de controle de acesso, de transação, etc. Esta

característica possibilita implementar, através de Web Services, princípios da Orientação a Serviços como baixo acoplamento e reusabilidade. A segunda parte do envelope, o corpo, contém os dados da comunicação, em si, que ocorre entre fornecedores e consumidores de serviço. No corpo também existe uma seção, chamada fault, que mantém informação de lógica de controle de exceção. SOAP é flexível ao ponto de possibilitar a inserção de tal lógica.

Figura 7 – Esquema de mensagem SOAP

Fonte: (ERL, 2005)



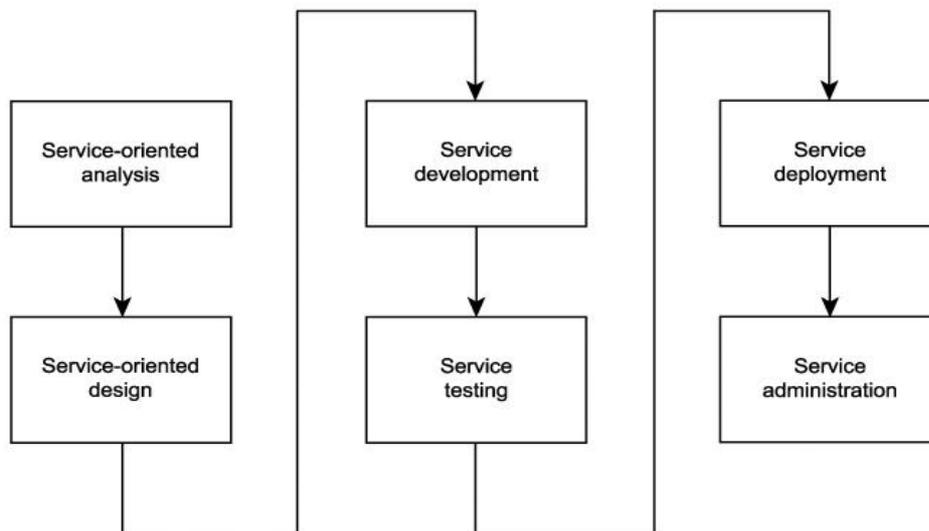
Após a explanação dos principais conceitos relacionados a Web Services mostra-se como esta tecnologia se apresenta como a mais adequada para se implementar SOA. Pode-se perceber como a associação entre serviços, descrição de serviços e mensagens em SOA é direta com estes mesmos conceitos em Web Services. Essa naturalidade com a qual é possível intercambiar os conceitos do paradigma e da tecnologia auxilia muito a compreensão e o projeto de serviços SOA através desta tecnologia.

4. Processo de entrega de SOA

O paradigma de Orientação a Serviços tem seus próprios conceitos e princípios. Por conta disto, é necessário que o processo de entrega de SOA considere estas características e seja dirigido por elas. Assim, os projetos pelo qual uma arquitetura orientada a serviços é desenvolvida utilizam uma versão orientada a serviços do processo normal de desenvolvimento de software. Erl(2005) aborda este processo de desenvolvimento de SOA, destacando as fases deste processo. A Figura 8 é um esboço destas fases.

Figura 8 – Ciclo de vida de um projeto de SOA

Fonte: (ERL, 2005)



Cada uma das fase do ciclo serão sucintamente apresentadas, de acordo com as descrições de Erl(2005).

Análise Orientada a Serviços: O objetivo da análise orientada a serviços é representar através da Orientação a Serviços os requisitos de automação que o negócio possui. Para isso, no processo de análise busca-se responder as duas seguintes questões:

- Quais os serviços que precisam ser construídos?
- Que lógica deve ser encapsulada por qual serviço?

Projeto Orientado a Serviços: O objetivo do projeto orientado a serviços vem a ser utilizar o que é descoberto na análise para determinar de que forma deve ser construída a solução orientada a serviços. No processo de projeto, muitos padrões que incorporam convenções da indústria e princípios da orientação a serviços são analisados e utilizados, gerando serviços concretos através de definições abstratas de serviço geradas na análise. Para

isso, algumas questões são respondidas, tais como:

- Como pode-se gerar definições de interfaces físicas de serviço através de candidatos de serviços modelados na fase de análise?
- Quais as características de SOA se quer implementar e suportar?
- Que padrões da indústria e extensões serão necessárias para que a SOA desenvolvida implemente o projeto de serviço planejado e as características de SOA desejadas?

Desenvolvimento de Serviços: Nesta fase ocorre a construção dos serviços. O que foi projetado na fase anterior é vinculado a uma específica plataforma de desenvolvimento.

Teste de Serviços: Nesta fase, o que foi construído é testado de acordo com os requisitos da análise e as considerações sobre quais características de SOA seriam suportadas, realizadas na fase de projeto.

Implantação de Serviços: Nesta fase, a solução desenvolvida é integrada ao restante da estrutura de computação já existente.

Administração de Serviços: Nesta fase, após implantado o serviço, busca-se mantê-lo em funcionamento de acordo com requisitos preestabelecidos na análise.

As fases de análise e projeto são as fases onde a Orientação a Serviços essencialmente ocorre. São nestas fases que métodos específicos deste paradigma são realizados com o intuito de se desenvolver SOA. O subcapítulo a seguir irá fornecer uma ideia básica sobre camadas de serviço para que no próximo sejam apresentados os processos de análise e projeto Orientados a Serviços propostos por Erl(2005).

4.1. Camadas de Serviços

É necessário, antes de se fazer análise e projeto orientado a serviços, ter uma ideia de como conseguir adicionar a uma determinada arquitetura de computação, as características de SOA. Erl(2005) afirma que, mesmo se utilizando os princípios da Orientação a Serviços e a tecnologia de Web Services, com o intuito de se construir SOA, ainda é necessário se efetuar trabalho para que todas as características de SOA sejam vistas na arquitetura.

Para tal, o mesmo autor relata a necessidade de existir um modelo de camadas, onde os serviços sejam agrupados de acordo a determinado critério. O modelo em camadas é utilizado em várias arquiteturas de plataformas computacionais, até mesmo a nível de hardware, com o intuito de se abstrair complexidade. O critério utilizado em SOA para

realizar esta abstração é, normalmente, uma classificação baseada nas características que os serviços devem possuir de um ponto de vista geral da arquitetura. Isto é conhecido, em SOA, como modelo de serviço.

O mesmo autor(2007) afirma, ainda, que os serviços podem ser classificados de acordo aos seguintes critérios:

- Tipo de lógica que eles encapsulam.
- A potencial quantidade de reuso da lógica que possui.
- Como essa lógica se relaciona a domínios existentes dentro da corporação.

Desta forma, é possível definir três modelos de serviço:

- Serviço de tarefa
- Serviço de entidade
- Serviço utilitário.

Estes modelos de serviços são conhecidos como modelos de serviços primários, por existirem, praticamente, em qualquer arquitetura orientada a serviços. A seguir as características de cada um serão melhor descritas.

4.1.1. Serviço de Tarefa

Para Erl(2007), um serviço de tarefa vem a ser um serviço com uma fronteira funcional diretamente associada a uma específica tarefa ou processo pai de negócio. Processo pai é aquele relacionado a várias entidades de negócio. Este tipo de serviço tende a ter um menor potencial de reuso e é geralmente posicionado como um controlador de composição, sendo assim responsável por agrupar outros serviços, serviços estes, mais independentes de uma determinada tarefa ou processo. Uma composição de serviços, para o mesmo autor, é um agregado coordenado de serviços. A Figura 9 a seguir dá um exemplo deste modelo. Neste exemplo, AnaliseDeMatricula é um serviço de tarefa.

4.1.2. Serviço de Entidade

Um serviço de entidade é um serviço centrado no negócio, que baseia sua fronteira funcional e seu contexto em uma ou mais entidades de negócio relacionadas. É considerado um modelo altamente reusável pois é agnóstico para a maioria dos processos de negócio. Sendo assim, um único serviço pode ser utilizado para automatizar vários processos

organizacionais. A seguir é apresentado um exemplo, através da Figura 10:

Figura 9 – Exemplo de serviço de tarefa

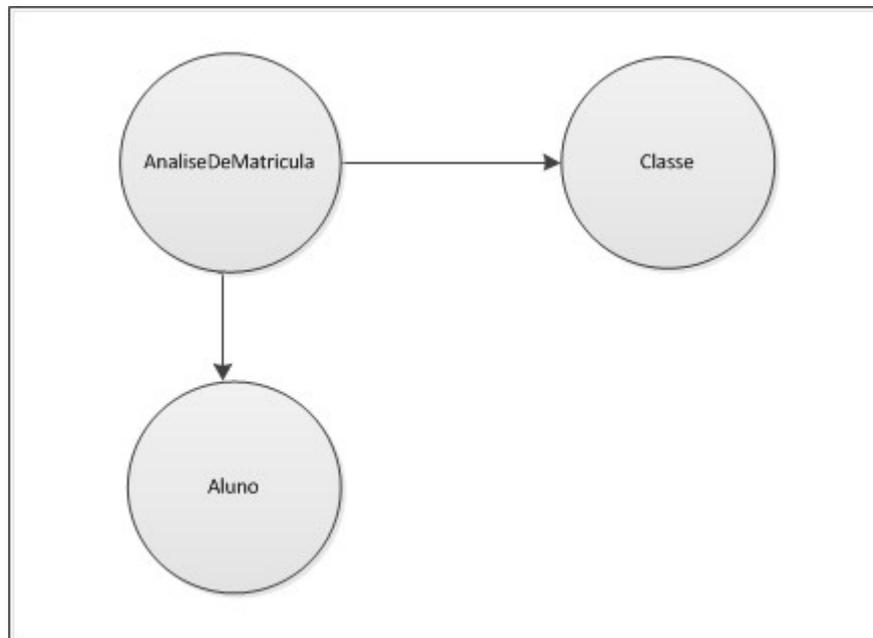
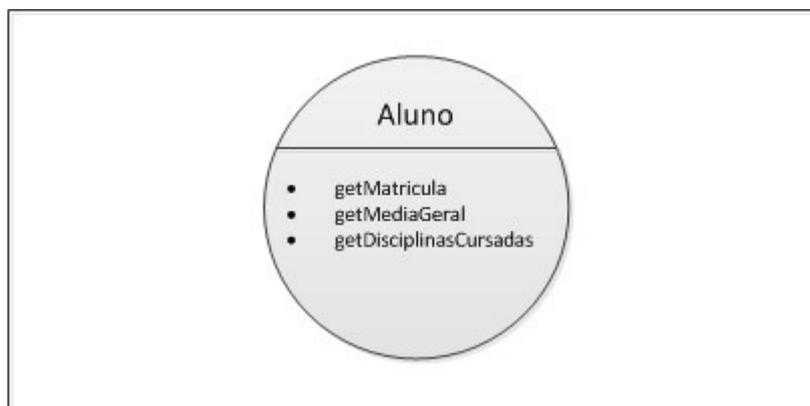


Figura 10 – Exemplo de serviço de entidade



4.1.3. Serviço Utilitário

Um serviço utilitário é aquele que fornece funcionalidades de utilidade transversal e reusável, tais como log de eventos, notificação e tratamento de exceção. Ele é idealmente agnóstico, pois não é específico de qualquer aplicação em si, consistindo de uma série de capacidades genéricas que vários sistemas necessitam. A Figura 11 traz um exemplo de um serviço utilitário.

Após definidos os principais modelos de serviço, que podem ser adaptados as necessidades organizacionais, é interessante se apresentar uma estrutura em camadas

fornecendo uma visão geral de como, normalmente, SOA pode ser implementado. Isto é realizado na Figura 12.

Figura 11 – Exemplo de serviço utilitário

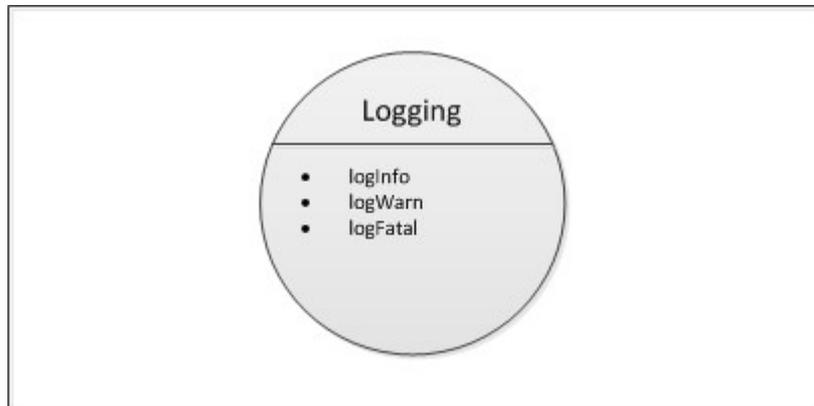
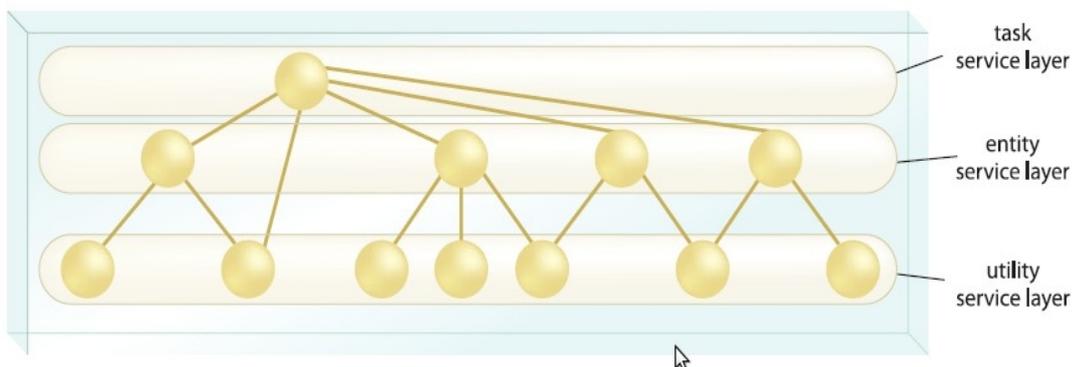


Figura 12 - Camadas de serviços

(Fonte: ERL, 2007)



A estrutura em camadas é característica de SOA. Dependendo da complexidade do negócio é possível se criar mais camadas e obter assim uma maior reusabilidade e abstração por toda a arquitetura.

4.3. Processos de Análise e Projeto Orientados a Serviços

Erl(2005) propõe um processo para análise e outro para projeto orientados a serviços. Como já mencionado anteriormente, estas são as fases do processo de entrega de SOA onde serão utilizados princípios próprios do paradigma de Orientação a Serviços. A seguir estes processos são apresentados. Analisar cada uma das etapas está além do escopo deste trabalho.

O processo de análise é composto pelas seguintes etapas:

1. Definir requisitos dos negócios.

2. Identificar sistemas de automação.
3. Modelar serviços candidatos.

Por sua vez, o processo de projeto possui as seguintes etapas:

1. Compor SOA.
2. Projetar serviços de negócio centrados em entidades.
3. Projetar serviços de aplicação.
4. Projetar serviços de negócio centrados em tarefas.
5. Projetar processos de negócio orientados a serviços.

5. Desenvolvimento da solução

Este capítulo tem como objetivo descrever o processo de desenvolvimento da solução baseada em SOA e desenvolvida através de Web Services com o intuito de prover a um novo sistema que está sendo desenvolvido no ambiente da UESB, o sistema para avaliação de professores, um meio de se utilizar lógica de negócio presente no sistema acadêmico(SAGRES). Este processo de desenvolvimento ocorreu de modo iterativo, como proposto por Erl(2005). Para isso, será descrita a contextualização do ambiente de sistemas da universidade e o que de mais importante ocorreu durante os processos de análise, projeto e construção da solução.

5.1. Contextualização

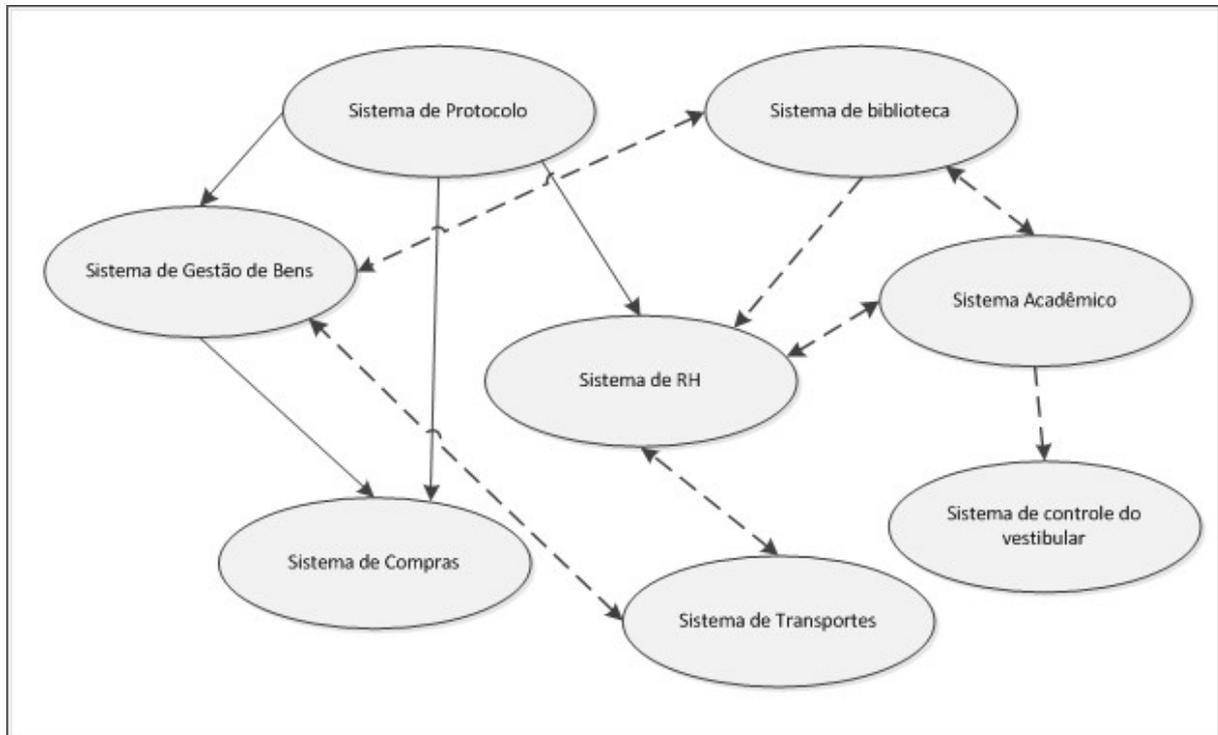
A UESB possui um conjunto de sistemas de informação pelos quais algumas partes de alguns dos inúmeros processos que ocorrem em seu ambiente interno e externo são automatizadas. Estes sistemas tem suas funcionalidades agrupadas de acordo a critérios administrativos e organizacionais desta instituição. A maioria destes sistemas são desenvolvidos e mantidos pela própria universidade, através da UINFOR, setor responsável, entre outras coisas, pela construção dos sistemas de informação necessários.

Estes sistemas encontram-se implementados em diferentes arquiteturas e em diferentes plataformas de desenvolvimento. Alguns possuem arquitetura desktop(stand alone) utilizando-se a plataforma de desenvolvimento Delphi, outros possuem arquitetura Web, utilizando-se Java EE. Existem alguns sistemas que podem ser considerados híbridos, como o caso do sistema acadêmico, que não é desenvolvido na universidade, sendo fornecido como produto por uma organização externa. Este sistema tem módulos desktop e módulos web.

A estrutura de software corporativo da UESB tem sido desenvolvida desde o final da década de 90 até hoje. Conseqüentemente, muito se mudou na maneira pela qual os primeiros sistemas foram desenvolvidos e são desenvolvidos hoje. As atuais necessidades da universidade também são bem distintas daquelas iniciais, com relação a TI. Como consequência desta conjuntura, praticamente todos os sistemas desenvolvidos do início ao meio deste período devem ser classificados como sistemas legados. Uma outra característica que fortalece essa classificação é que a maioria destes foi desenvolvida em plataforma Delphi, sendo que o plano da UINFOR é descontinuar, aos poucos, o uso desta plataforma. Encontra-se aqui uma boa justificativa para a adoção de SOA.

A Figura 13 exibe uma visão geral do ambiente de sistemas encontrado atualmente na universidade. Destaca-se que, apenas os principais sistemas aparecem no esboço:

Figura 13 – Visão Geral do ambiente de sistemas da UESB



Nesta figura, as elipses representam os sistemas e os conectores representam dependência de lógica de negócio. O conector sai do sistema que necessita da lógica e vai àquele que o fornece. Conectores contínuos tem essa dependência de alguma forma já implementada e os tracejados necessitam de implementação. No caso dos sistemas que necessitam de lógica de negócio de outro e estão com os conectores tracejados, é provável que exista redundância na implementação desta lógica, ou seja, a lógica é implementada nos dois sistemas e é possível também que ela esteja gerando resultados diferentes para entradas iguais. Esse tipo de redundância é maléfica a estrutura de sistemas, pois torna mais onerosa qualquer alteração feita em um sistema que tenha lógica redundante. Uma outra consequência neste cenário é a possibilidade de um dos sistemas, ou mais, estarem gerando resultados errados.

Uma outra característica deste ambiente é a necessidade de desenvolvimento de novas soluções de software. Existe grande demanda de sistemas para atender a diversos clientes na UESB. Vale a pena ressaltar o dinamismo que pode ser visto na universidade. Existem inúmeros relacionamentos internamente na UESB e desta com o ambiente externo. Isto torna a realidade da UESB muito complexa e, conseqüentemente, de difícil análise. Assim, a UESB

tem grande necessidade de ter seu aparato de TI alinhado a suas atividades meio e atividades fim, podendo ela dessa forma melhor aproveitar as oportunidades e sobressair nas dificuldades. Essa é outra justificativa para se adotar SOA.

Com o intuito de complementar as informações já fornecidas, a tabela apresentada a seguir exibe informações referentes a características importantes associadas com os principais sistemas utilizados na UESB.

Tabela 1 – Caracterização dos principais sistemas da UESB

Sistema	Plataforma	Arquitetura	Cliente
Sistema Acadêmico (Sagres)		Híbrida	Pró-Reitoria da Graduação
Sistema de RH (Populus)	Delphi + Java EE	Híbrida	Pró-Reitoria de Administração e Recursos Humanos
Sistema de Protocolo (Lupus)	Delphi + Java EE	Híbrida	Pró-Reitoria de Administração e Recursos Humanos
Sistema de Gestão de Bens (Ajax)	Delphi	Desktop	Gerência Administrativa
Sistema de Compras	Delphi	Desktop	Gerência Administrativa
Sistema de Biblioteca(Pergamun)		Híbrida	Setor de Biblioteca
Sistema de Transportes	Delphi	Desktop	Setor de Transportes
Sistema de controle do processo de seleção de vestibular	Delphi + Java EE	Híbrida	Comissão permanente para o vestibular

É necessário observar que as células na coluna plataforma que estão em branco referem-se a sistemas fornecidos por terceiros e que, portanto, estão fora do controle da universidade.

Na sequência, serão apresentados os pontos mais importantes associados ao desenvolvimento da solução baseada em SOA e implementada com Web Services, já referenciada no trabalho. O termo solução, na sequência do trabalho, fará referência ao conjunto de web services desenvolvido.

5.2. Análise da solução

Assim como recomenda Erl(2005), o processo de análise seguido para se desenvolver a solução possui 3 etapas. Na primeira etapa foram analisados os requisitos do negócio, que de certa forma estão representados também na contextualização efetuada no subcapítulo anterior. Esta contextualização trouxe uma visão mais ampla e genérica das necessidades da organização, no caso a UESB, assim como deu base para se definir as camadas de serviço,

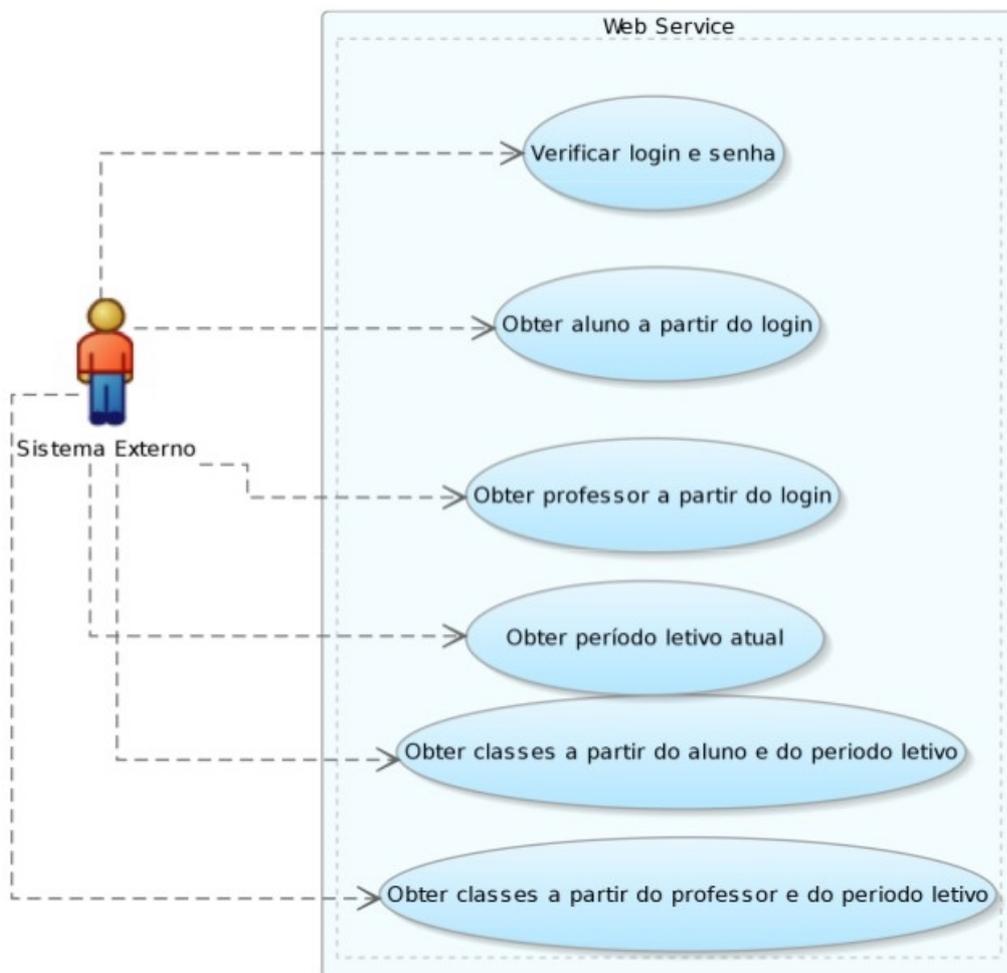
que servirão de estrutura base para o desenvolvimento desse projeto e de outros, caso desejado.

Ainda na primeira etapa foram desenvolvidos os casos de uso que permitiram a modelagem da solução, tanto no restante da análise quanto no projeto. Estes casos de uso serão apresentados a seguir.

5.2.1. Requisitos

Os casos de uso da solução desenvolvida estão listados no diagrama de casos de uso, apresentado na Figura 14.

Figura 14 – Diagrama de casos de uso da solução



Estes são os seis casos de uso que compõe a solução a ser desenvolvida. O ator Sistema Externo, inicialmente, vem a ser o sistema que está sendo desenvolvido para avaliação de professores.

Uma outra tarefa realizada na primeira etapa da análise é a definição do escopo do

projeto de SOA desenvolvido. Sendo este o **primeiro projeto onde se aplica tais conceitos na UINFOR**, foi escolhido um **escopo bem pequeno e definido** e que deve atender as necessidades do sistema de **avaliação de professores**. Após estas necessidades serem atendidas, é possível se projetar os serviços, sempre que possível, buscando as características de SOA, desde que não torne a solução tecnicamente muito complexa.

5.2.2. Lógica de automação já existente

A segunda etapa do processo de análise (identificar lógica de automação já existente) foi de certa forma comprometida, uma vez que a propriedade do código do sistema acadêmico não pertence a UESB. No entanto, a universidade tem um contrato de manutenção com a empresa que fornece o sistema e através deste, é possível se ter acesso pelo menos aos dados que são referentes ao SAGRES. A forma pelo qual a UINFOR tem acesso a estes dados é através de visões no banco. Sendo assim, não existe a possibilidade de reutilização de lógica de aplicação, restando somente as visões como meio de acesso aos dados já cadastrados na base. Essas visões serão utilizadas para que os web services que constituem a solução forneçam os dados, inicialmente ao sistema de avaliação de professores e depois aos outros que necessitarem.

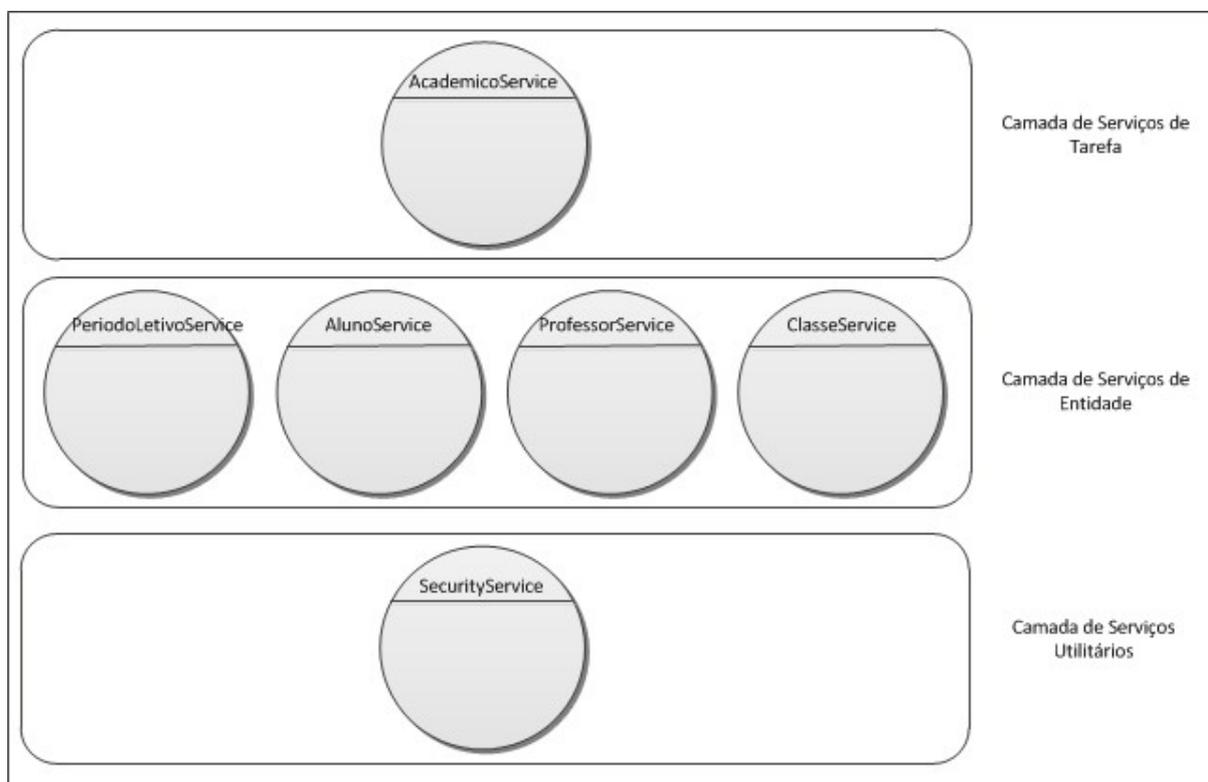
5.2.3. Camadas de serviço e serviços candidatos

Inicialmente existiria somente uma camada de serviços, que iria conter os modelos de serviços de tarefa. Esta camada faria uma fronteira entre o sistema acadêmico e os outros sistemas da universidade. No entanto, após refinamento, tornou-se interessante a existência de um modelo com mais camadas, surgindo então a camada de serviços de entidade e a camada de serviços utilitários. Dessa forma aumenta-se a reusabilidade do solução, pois quem necessitar de serviços de entidade ou serviços utilitários relacionados ao sistema acadêmico, pode utilizar os serviços já existentes. Como já afirmado no capítulo 4, serviços de entidade e serviços utilitários tendem a ser amplamente reutilizáveis no contexto de SOA.

A Figura 15 apresenta o modelo de camadas de serviços desenvolvido e os serviços candidatos. Percebe-se também pela figura que existem 6 candidatos de serviços. O `AcademicoService` é um serviço de tarefa que fará uma fronteira entre o consumidor do serviço (normalmente uma aplicação da UESB) e o restante das camadas. Os serviços de entidade permitirão que operações relacionadas a cada um dos conceitos de domínio, no caso,

Período Letivo, Aluno, Professor e Classe, estejam acessíveis aos consumidores. Por fim, localizado na camada de serviços utilitários, estará o SecurityService, que encapsulará lógica agnóstica referente a controle de permissões.

Figura 15 – Camadas de serviço e serviços candidatos



5.3. Projeto da solução

O processo de projeto utilizado para desenvolvimento da solução é uma adaptação do processo proposto por Erl(2005). Este adaptação contém três etapas, a saber:

- Compor SOA.
- Projetar camadas de serviço.
- Projetar os serviços em cada camada.

Para o escopo deste trabalho, compor SOA vem a ser definir a tecnologia que dará suporte aos princípios de SOA. Os critérios utilizados na escolha da tecnologia para desenvolvimento da solução são informados na sequência.

5.3.1. Escolha da tecnologia

SOA, conforme listado no capítulo 2, suporta diversidade de fornecedor. Isso é

essencial no ambiente da UESB e ao desenvolvimento do sistema de avaliação de professores, uma vez que este sistema está sendo implementado fora da UINFOR. No entanto, para o desenvolvimento desta solução em específico, a plataforma Java EE é a única escolha aceitável. Os motivos pelos quais isto ocorre são os seguintes:

- A UINFOR tem como padrão desenvolver para essa plataforma.
- Essa plataforma possui um conjunto de benefícios que auxiliam no desenvolvimento da solução, tais como: Suporte de várias IDEs, diminuição considerável do trabalho de se criar arquivos XML, frameworks disponíveis, etc.
- Já existe, na UINFOR, um servidor web em ambiente de produção no qual a solução será implantada.

Ainda argumentando sobre escolhas tecnológicas, apesar de se escolher Java EE, existem várias escolhas mais específicas a se fazer pois Java EE também suporta diversidade de fornecedores. A mais importante destas vem a ser a escolha da versão desta plataforma. A versão escolhida foi o Java EE 5, quando comparado ao Java EE 6. O motivo principal desta escolha é a necessidade de se utilizar o framework Spring, um outro padrão tecnológico da UINFOR. Além disso, o Spring possui um módulo para desenvolvimento de web services muito flexível e que deixa o desenvolvimento destes mais adequado a plataforma Java. A seguir, são listados os principais componentes tecnológicos utilizados no desenvolvimento do projeto:

- Java EE 5
- Spring Framework
- Spring Web Services
- Hibernate
- Netbeans IDE

Depois de mostrar os critérios pelos quais a tecnologia foi escolhida, é momento de apresentar as considerações realizadas durante as duas próximas etapas do processo de projeto.

5.3.2. Projeto de camadas e de serviços

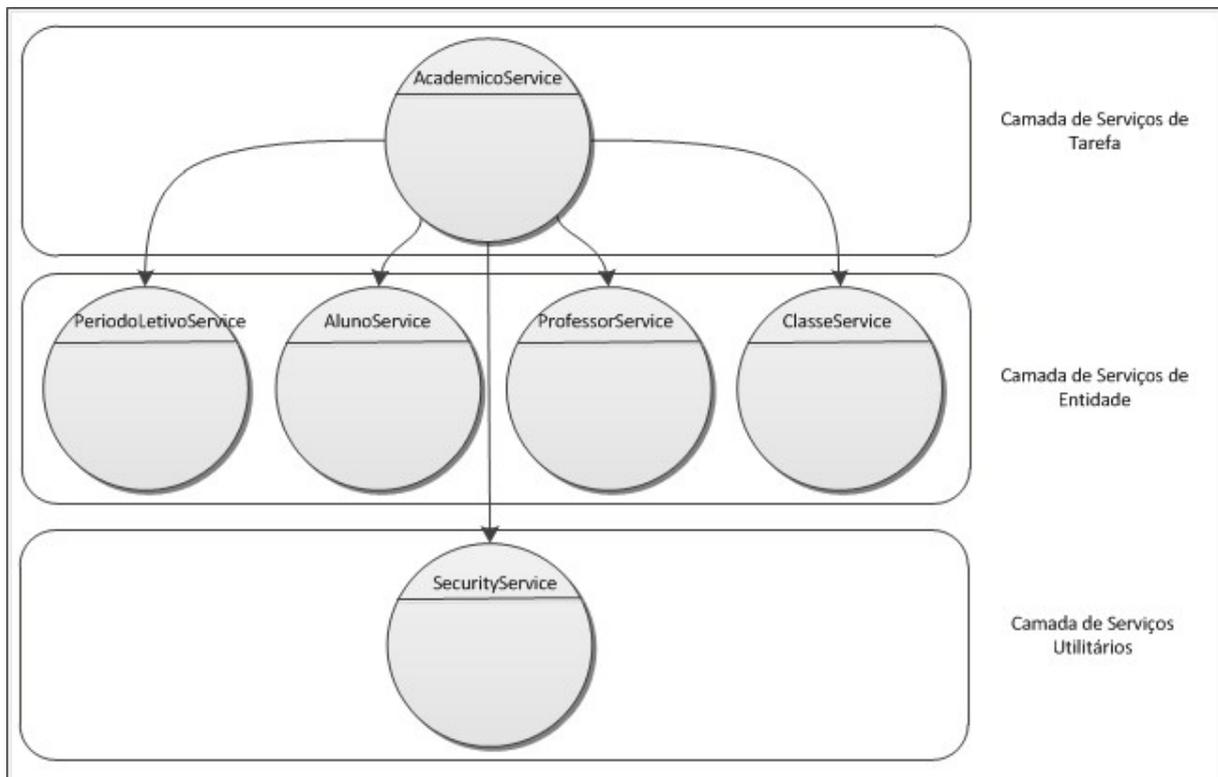
Como saída da análise, foi gerado um modelo de camadas e um conjunto de serviços candidatos. Este modelo de camadas, assim como os serviços candidatos pode sofrer alterações da análise para o projeto. Dos requisitos mostrados no diagrama de casos de uso,

foram retiradas seis capacidades que devem estar no inventário de serviços que constitui a solução. Estas capacidades são as listadas a seguir:

- checkLogin
- getAlunoFromLogin
- getProfessorFromLogin
- getPeriodoLetivoAtual
- getClassesFromAlunoAndPeriodoLetivo
- getClassesFromProfessorAndPeriodoLetivo

As camadas desenvolvidas na análise continuaram as mesmas no projeto, uma vez que estes três níveis de camadas trazem um alto grau de flexibilidade para a arquitetura e não foi necessário nenhuma mudança nesta estrutura. As capacidades foram atribuídas aos serviços, no entanto por questão de reusabilidade, em alguns casos a mesma capacidade foi adicionada a mais de um serviço. O resultado desse processo é apresentado a seguir:

Figura 16 – Camadas de Serviços após o termino da fase de projeto



Observando-se as diferenças nas apresentações das camadas, na fase de análise e de projeto, percebe-se que na fase de projeto os relacionamentos entre os serviços são formados. A Figura 17 exibe cada um dos serviços com suas respectivas capacidades. Vale a pena

ressaltar que o serviço acadêmico(AcademicoService) engloba todas as capacidades listadas anteriormente. Essa foi uma escolha de projeto e tem como objetivo abstrair em um só serviço as capacidades de determinado agrupamento lógico de serviços. Dessa forma, será muito mais fácil para o sistema de avaliação de professores utilizar os serviços que necessita. Este também é um claro exemplo de uma composição de serviços.

5.4. Construção da solução

Depois de concluído o projeto orientado a serviços, o próximo passo para a conclusão da aplicação vem a ser a construção, em si, do agente do web service. Para tanto, foi utilizado o ambiente de desenvolvimento integrado NetBeans, que fornece algumas ferramentas que auxiliam o desenvolvimento de web services com Java EE. O primeiro passo para a construção, que em si é uma tarefa de projeto, só que orientado a objetos, vem a ser a criação de um modelo de classes, abstraído na Figura 18 por meio de um diagrama do mesmo tipo. Este modelo de classes possui as entidades aluno, professor, período letivo, classe e disciplina.

Estas classes formam a estrutura básica que o sistema de avaliação de professores necessita para operar. Além disso, são as entidades chave para todo o sistema acadêmico, logo, a possibilidade de que a estrutura delas seja reutilizada em uma outra necessidade é muito grande.

É necessário destacar que este modelo de classes passa a ser válido em todo o ambiente de sistemas da UESB. Isto mostra como o benefício prometido por SOA, que diz respeito a um direcionamento em sentido a uma representação padronizada de dados, realmente é alcançável.

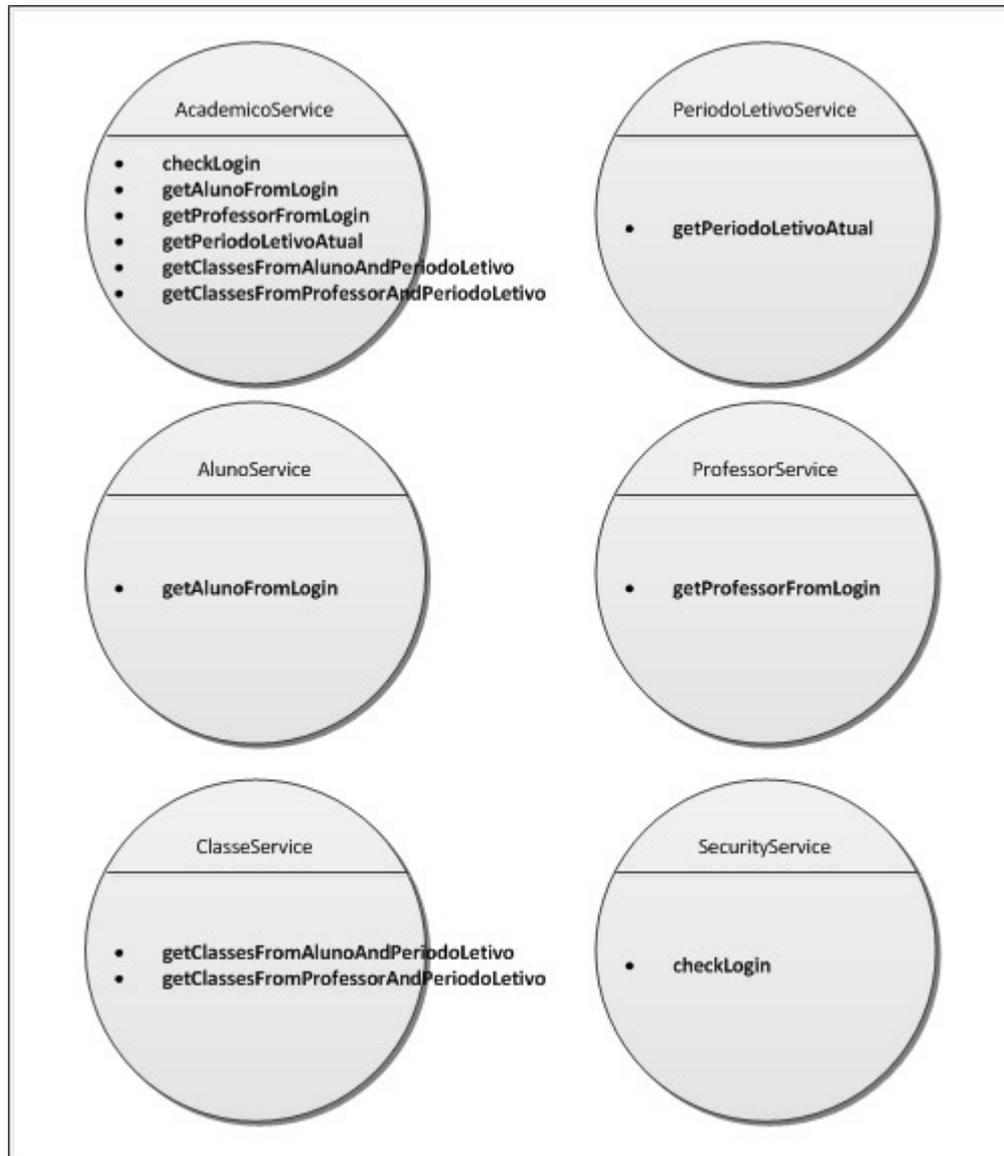
5.4.1. Arquitetura

A arquitetura utilizada na solução é composta por duas camadas, uma de negócios e outra de persistência. A camada de negócio contém as entidades e os componentes relacionados com lógica de negócio no contexto da aplicação. Aqui entram os componentes de infraestrutura do Spring Web Services, que abstraem considerável parte do trabalho em se desenvolver estes artefatos.

Já a camada de persistência é implementada, em sua maior parte através do framework Hibernate. Os objetos que se encontram nesta camada tem responsabilidades associadas a

manipulação de dados no banco do SAGRES.

Figura 17 – Serviços e suas respectivas capacidades

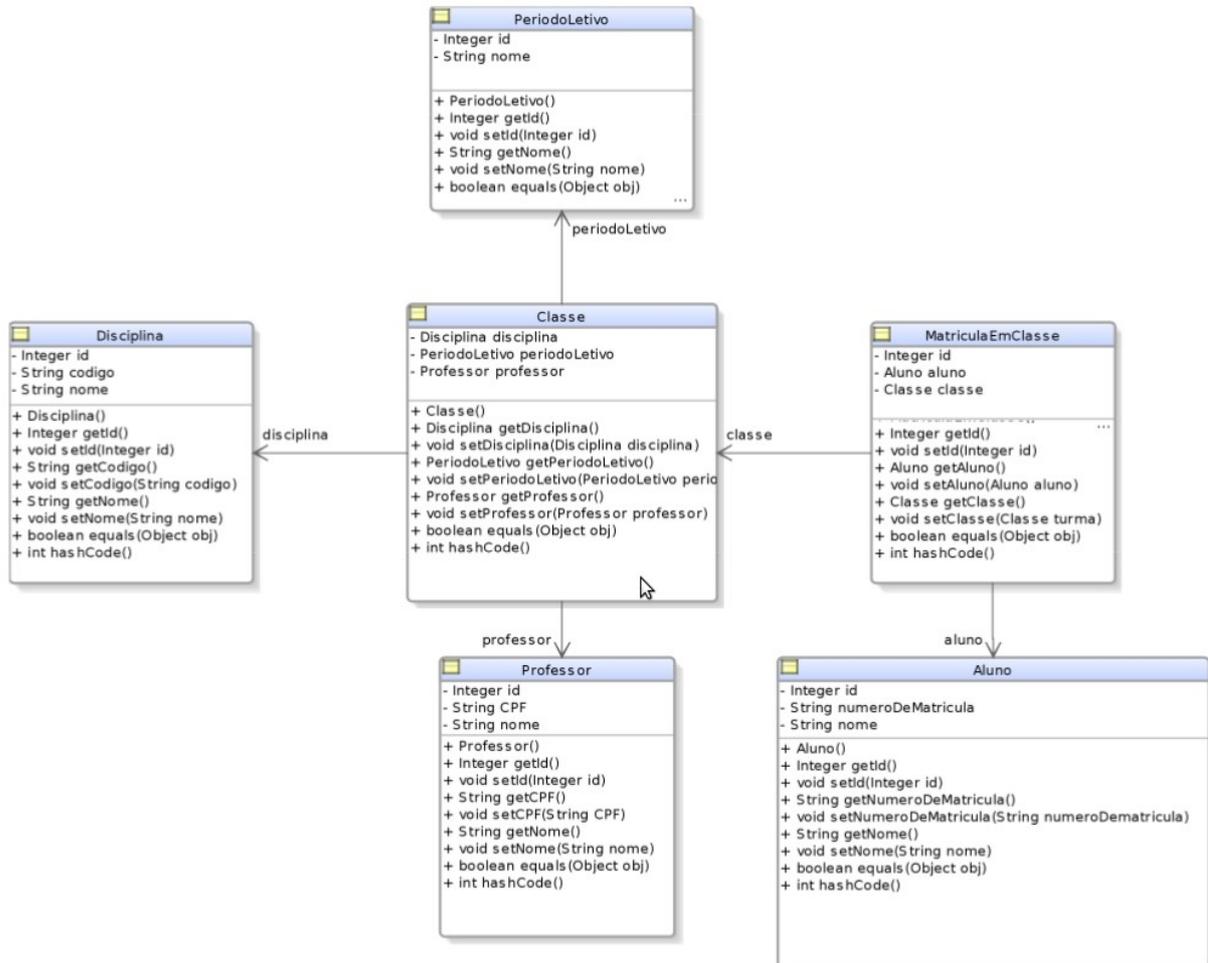


5.4.2. Testes e resultado do processo de construção

Durante o processo de construção foram efetuados testes dos serviços, com o intuito de garantir uma certa confiabilidade aos dados obtidos. Como o sistema de avaliação ainda está em desenvolvimento, não pode ocorrer, de fato a integração entre a solução desenvolvida. No entanto, por conta do escopo bem delimitado e dos resultados dos testes, realmente não é esperado nenhum problema na integração entre a solução desenvolvida e o novo sistema. Os serviços encontram-se implantados e acessíveis através do ambiente de sistemas da

universidade.

Figura 18 – Diagrama de classes de entidade



6. Conclusão

Através da análise realizada durante todo o trabalho aliado ao desenvolvimento do estudo de caso e da experiência com outros paradigmas de desenvolvimento de sistemas distribuídos é possível se afirmar que SOA alcança seus principais objetivos. Pôde-se perceber que os conceitos associados ao paradigma de Orientação a Serviços realmente permitem o desenvolvimento de um ambiente de sistemas onde é mais fácil se reutilizar e alterar o que já foi desenvolvido. Um outro benefício que pode ser observado através do trabalho foi a agilidade obtida através da arquitetura, que permite alterações em processos e regras de negócio serem facilmente refletidos nos sistemas.

O nível de abstração fornecido por camadas de serviço também pode ser observado, sobretudo, no estudo de caso, onde foi visível a flexibilidade ganha ao se adicionar camadas de serviços a arquitetura desenvolvida. A tecnologia de Web Services também mostrou-se muito adequada ao se trabalhar com SOA. Muitas das características de SOA, como baixo acoplamento, neutralidade de fornecedor e federação são suportadas nativamente por esta tecnologia.

Aumentar o escopo do trabalho, abordando também cada uma das características de SOA e descrever em maior grau os processos de análise e projeto orientados a serviços poderiam ser algumas melhorias propostas a revisão bibliográfica realizada no trabalho. Algumas melhorias a serem feitas no estudo de caso estão relacionadas a aumentar o escopo de SOA que foi utilizado no projeto, inicialmente a uma maior parte do sistema acadêmico, com o intuito de se verificar a arquitetura desenvolvida em um contexto maior.

Espera-se que este trabalho dê uma boa introdução a SOA e o estudo de caso sirva de experiência para a UINFOR, caso venha a adotar este paradigma no seu ambiente de sistemas.

7. Referências:

- BASS, L. **Software Architecture in Practice**. 2 ed. [S.l.]: Addison Wesley, 2003
- BEAN, J. **SOA and Web Services Interface Design: Principles, Techniques and Standards**. [S.l.]: Elsevier, 2010
- ERL, T. **SOA: Principals of Service Design**. Crawfordsville : Prentice Hall , 2007.
- ERL, T. **Service-Oriented Architecture: Concepts, Technology, and Design**. Crawfordsville: Prentice Hall , 2005.
- ERL, T., 2008a. **SOA Design Patterns** . Crawfordsville: Prentice Hall , 2008a.
- ERL, T. Et al., 2008b. **Web Service Contract Design and Versioning for SOA**. Crawfordsville: Prentice Hall , 2008.
- HEIL , Maiara. **Uma Proposta De Guia De Referência Para Provedores De Software Como Um Serviço** . 2009. 134p. Dissertação(Mestre em Engenharia de Automação e Sistemas) - Universidade Federal de Santa Catarina, Florianópolis. 2009
- HURWITZ, J. Et al. **Service Oriented Architecture For Dummies**. Indianapolis: Willey, 2007
- SILVA, E. L. D.; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. Editora da UFSC, 2005.
- SOA CONSORTIUM, 2008a. Disponível em <http://www.soa-consortium.org>. Acesso em: 03/09/2011
- SOA WORK GROUP, **The SOA Source Book**. 4 ed. Disponível em <http://www.opengroup.org/soa/source-book/soa/index.htm>. Acesso em 03/09/2011
- SOA GLOSSARY, **Definitions for Service-Oriented Computing Terms**. 2011. Disponível em <http://www.w3.org/TR/ws-arch/#service>, Acesso em 06/09/2011.
- TIDWELL , D.; SNELL , J.; KULCHENKO , P. **Programming Web Services with SOAP** .

[S.l.]: O'Reilly 2001.

WORLD WIDE WEB CONSORTIUM, **Web services architecture. Technical report.** 2004.
Disponível em <http://www.w3.org/TR/ws-arch/#service>, Acesso em 26/08/2011.