



TRABALHO DE GRADUAÇÃO

**A Systematic Mapping Study:  
Qualidade de Serviço  
em Computação Orientada a Serviços**

**Danilo Filgueira Mendonça**

Brasília, Março de 2012

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

## TRABALHO DE GRADUAÇÃO

# **A Systematic Mapping Study: Qualidade de Serviço em Computação Orientada a Serviços**

**Danilo Filgueira Mendonça**

Relatório submetido ao Departamento de Engenharia Elétrica  
como requisito parcial para obtenção do grau de  
Engenheiro de Redes de Comunicação

Banca Examinadora

Profa. Dra. Genáina Nunes Rodrigues , \_\_\_\_\_  
CIC/UnB  
(Orientadora)

## FICHA CATALOGRÁFICA

MENDONCA, D. F.. A Systematic Mapping Study:  
QOS em Computação Orientada a Serviços [Distrito Federal] 2012.  
v, 52p. (ENE/FT/UnB, Engenheiro de Redes de Comunicação, 2012)  
Monografia de Graduação - Universidade de Brasília. Faculdade de Tecnologia.  
Departamento de Engenharia Elétrica.

1. Computação Orientada a Serviços	2. SOC
3. Arquitetura Orientada a Serviços	4. SOA
5. Qualidade de Serviços	6. QOS
I. ENE/FT/UnB	II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

MENDONCA, D. F. e (2012). A Systematic Mapping Study: QOS em Computação Orientada a Serviços Monografia de Graduação, Publicação ENE 01/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 52p.

## CESSÃO DE DIREITOS

NOMES DOS AUTORES: Danilo Filgueira Mendonça e .

TÍTULO: A Systematic Mapping Study: QOS em Computação Orientada a Serviços

GRAU / ANO: Engenheiro de Redes de Comunicação / 2012.

É concedida à Universidade de Brasília permissão para reproduzir cópias desta monografia de graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte desta monografia de graduação pode ser reproduzida sem a autorização por escrito dos autores.

---

Danilo Filgueira Mendonça  
SQN 303 BL F APTO 605 - ASA NORTE  
CEP 70735-060 - Brasília - DF - Brasil.

---

---

## **Dedicatória**

*Às manhãs de meu avô.*

*Danilo Filgueira Mendonça*

## Agradecimentos

*Danilo Filgueira Mendonça*

---

## RESUMO

A Computação Orientada a Serviços ressurgiu com o intuito de prover maior eficiência à produção e consumo de recursos computacionais, especialmente os softwares, que passam a compor unidades coesas e granulares de lógica capazes de se intercomunicarem, aumentando o reuso, a agilidade e o retorno de investimento. Dada a incerteza sobre o estado da pesquisa relacionada ao tema, detectou-se a necessidade de um mapeamento de publicações, indicando quais tipos e quais subáreas receberam maior foco em detrimento daquelas que receberam menor foco, determinando também seus benefícios e limitações. A partir de um processo sistemático de mapeamento de estudos, ou Systematic Mapping Study, que envolve a busca por publicações em fóruns de interesse de modo a classificá-las, avaliamos artigos que tratam de QOS em SOC para enfim responder às questões de pesquisa que orientam e motivam o presente estudo. O resultado foi um mapeamento amplo que classifica XXX publicações e que permite uma melhor compreensão dos esforços de pesquisa dentro da área de interesse.

O trabalho é dividido em 5 capítulos: Introdução, Referência Teórica, Abordagem, Discussão, Conclusão e Trabalhos Futuros. O capítulo de Introdução trata da área de interesse e da metodologia utilizada para o mapeamento. Aborda-se também os objetivos que são divididos em objetivos principais, gerais, específicos e a longo prazo do trabalho. Neste capítulo também apresentam-se as motivações e é mostrada a devida importância do trabalho. Ainda nesse capítulo são descritos trabalhos relacionados. No capítulo de Referencial Teórico são apresentados os principais conceitos necessários para o entendimento do processo de mapeamento sistemático de estudos, além daqueles que envolvem os conceitos de QOS em SOC. No capítulo de Abordagem descreve-se o processo de mapeamento, desde a busca, o processo de inclusão e exclusão de publicações e a classificação dos mesmos. No penúltimo capítulo, Discussão, são colocadas considerações à respeito das questões de pesquisa que motivaram esse mapeamento. Na Conclusão é feito um balanço dos objetivos e a visão final sobre os resultados obtidos, assim como a expectativa para trabalhos futuros.

---

## ABSTRACT

### *English version*

The work here presented shows how to build applications that perform data collection and analysis of virtual social networks, focusing on Facebook. They are also summarized ways of carrying out the processes of collecting, analyzing and building applications and presented the concepts involved in understanding these processes.

The work is divided into five chapters: Introduction, Theoretical Background, Implementation, Case Studies and Conclusion. In the Introduction chapter, all chapters of this work are discussed superficially. It also addresses the goals are divided into primary objectives, general, specific and long-term work. This chapter also presents the motivations that are all justified and is shown the importance of proper work. Although this chapter describes related work. Theoretical Background In chapter presents the main concepts of work that are necessary for understanding the ensuing chapters. Implementation is presented in chapter created the application, what it does and how. In addition to showing the reader the aspects involved in software development. In the penultimate chapter, case study presents the analysis performed from the saved data categorization. In the last chapter, Conclusion, reflects on the work and the achievement of objectives, addressing some of the content described in all chapters of the work. Then we have four annexes relating to the Implementation Chapter.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	MOTIVAÇÃO	1
1.2	OBJETIVOS	2
1.2.1	OBJETIVOS GERAIS	2
1.2.2	OBJETIVOS ESPECÍFICOS	2
1.3	TRABALHOS RELACIONADOS	2
1.4	ORGANIZAÇÃO DO TRABALHO	2
<b>2</b>	<b>EMBASAMENTO TEÓRICO</b>	<b>3</b>
2.1	SERVICE ORIENTED COMPUTING	3
2.1.1	SERVICE ORIENTATION HISTORY	4
2.1.2	COMPONENT BASED ARCHITECTURE	4
2.1.3	OBJECT ORIENTATION	6
2.1.4	WEB SERVICES	8
2.1.5	ASPECT ORIENTED PROGRAMMING	9
2.1.6	BUSINESS PROCESS MODEL	9
2.1.7	SERVICE ORIENTED DESIGN	11
2.1.8	SERVICE ORIENTED ARCHITECTURE	11
2.2	SYSTEMATIC MAPPING STUDY	13
<b>3</b>	<b>IMPLEMENTAÇÃO</b>	<b>15</b>
3.1	DESCRIÇÃO GERAL DA IMPLEMENTAÇÃO	15
3.2	TECNOLOGIAS UTILIZADAS	15
3.2.1	TECNOLOGIAS DO FACEBOOK	16
3.2.2	TECNOLOGIAS GERAIS: LINGUAGENS DE PROGRAMAÇÃO, BIBLIOTECAS E IDES	16
3.2.3	SERVIDOR DE HOSPEDAGEM E DE DOMÍNIO	17
3.3	REGRAS DE NEGÓCIO, ANÁLISE DE REQUISITOS E METODOLOGIAS	17
3.3.1	ELABORAÇÃO DA REGRA DE NEGÓCIO E ANÁLISE DE REQUISITOS	18
3.3.2	REGRAS DE NEGÓCIO: COLETAR E ANALISAR O QUE?	19
3.3.3	PRINCIPAIS REQUISITO PARA COLETA E ANÁLISE	20
3.4	MINERAÇÃO DE DADOS	21
3.5	METODOLOGIAS, PADRÕES E TESTES DO PROJETO	22
3.6	ESTRATÉGIAS PARA ATINGIR PÚBLICO ALVO	22
<b>4</b>	<b>ESTUDO DE CASO</b>	<b>24</b>
4.1	DISCUSSÃO DE RESULTADOS	24
<b>5</b>	<b>CONCLUSÃO</b>	<b>27</b>
5.1	SUGESTÃO DE MELHORIAS	28
	<b>REFERÊNCIAS</b>	<b>29</b>
<b>6</b>	<b>ANEXOS</b>	<b>31</b>
6.1	ANEXO I - ESTRUTURA DE ARQUIVOS DO PROJETO IMPLEMENTADO	31
6.2	ANEXO II - FORMAS DE ACESSO E CONFIGURAÇÃO DO APLICATIVO NO FACEBOOK	31
6.3	ANEXO III - CD COM ARQUIVOS DO PROJETO IMPLEMENTADO	31

<b>ANEXOS.....</b>	<b>32</b>
--------------------	-----------



## LISTA DE FIGURAS

2.1	Princípios relacionados ao SOA e seus benefícios .....	3
2.2	Origens da Orientação a Serviço.....	4
2.3	Comparação entre atividades típicas de componentes e serviços .....	5
2.4	Comparação entre escopos da Orientação a Objetos a da Orientação a Serviços .....	7
2.5	Modelagem de um BP por meio de um diagrama BPMN .....	10
2.6	Processo para o Systematic Mapping Study .....	14

# NOTAÇÕES

## Siglas

RSO	<i>Redes Sociais Online</i>
OSN	<i>Online Social Networks</i>
WEB	<i>Sistema de Documentos de Hipertexto Interligados que podem ser executados via Internet.</i>

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

Services, SOA, Web Services, SOAP, REST, Service Orientation, Cloud, SaaS. Nos últimos anos esses e outros acrônimos tornaram-se frequentes na tecnologia da informação. O surgimento de um novo paradigma, impulsionado pelo amadurecimento da internet e pela proximidade crescente entre negócios e TI, criou novos caminhos e oportunidades para trabalhos de desenvolvimento e pesquisa. Nesse sentido, um grande número de estudos foram e vem sendo conduzidos com foco nos diversos aspectos da computação orientada a serviços, tais quais arquitetura, modelos, métodos, processos, ferramentas diversas, frameworks, métricas, problemas solucionados e ainda vigentes. Dessa forma, a intenção daqueles interessados em iniciar suas atividades na área fica comprometida pela dificuldade em obter informações claras sobre o atual estado da arte, os desafios, os temas mais abordados e aqueles com deficit em pesquisas. Esses dados são cruciais para que esforços sejam bem direcionados e para que a ciência caminhe com cooperação e eficiência.

Um Mapping Study visa classificar de forma sistemática e ampla um conjunto de estudos. Dada a grande quantidade de publicações no escopo da orientação a serviços, sua metodologia ágil e que permite a análise de um maior número de estudos [1] justifica sua escolha em detrimento de outras metodologias, sendo a mais conhecida o Systematic Review. Essa última exige uma análise minuciosa e detalhada de cada publicação, o que requer um esforço considerável e inviabiliza a inclusão de um grande número de publicações num quadro de poucos pesquisadores. Assim, dados os fatos citados e o interesse em obter uma classificação ampla e significativa da ciência relacionada à orientação a serviços, este trabalho de conclusão de curso em Engenharia de Redes de Comunicação realiza um Mapping Study com foco em orientação a serviços. Mais especificamente e de forma a preservar um maior grau de coesão nos resultados, nosso mapeamento irá focar na questão de qualidade, sobretudo a qualidade de serviços, termo aqui empregado de forma literal e posterior ao termo QoS, uma vez que os principais agentes do paradigma em questão são, coincidentemente, denominados serviços.

Sobre a motivação pessoal desse trabalho é preciso destacar meu interesse na Engenharia de Software, principalmente em seu aspecto distribuído, uma vez que possibilita novos patamares de interoperabilidade, aumentando a gama de soluções da tecnologia da informação e tomando proveito do avanço nas tecnologias

de comunicação de dados, o que envolve questões próprias da Engenharia de Redes. Assim, vejo justificada a iniciativa desse trabalho, que mesmo em sua característica acadêmica *latu sensu*, irá trazer contribuições para ambas as áreas.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivos Gerais**

Os objetivos gerais de um Mapping Study incluem u

### **1.2.2 Objetivos Específicos**

## **1.3 TRABALHOS RELACIONADOS**

## **1.4 ORGANIZAÇÃO DO TRABALHO**

Neste trabalho descreveremos características de RSO em geral e da RSO Facebook em específico, apresentaremos conceitos de redes complexas, resumiremos as principais técnicas de coleta e análise de dados em RSOs entre outros no próximo capítulo, Embasamento Teórico. Logo após no capítulo Implementação, descreveremos o Aplicativo Desenvolvido, Tecnologias Utilizadas, Metodologia de Desenvolvimento e Gerenciamento, Estratégias para atingir o Público Alvo No aplicativo no capítulo Implementação. Seguindo pelo capítulo Resultados. Enfim, chegamos a uma conclusão e propomos algumas melhorias para este trabalho, no capítulo conclusão.

## 2 EMBASAMENTO TEÓRICO

### 2.1 SERVICE ORIENTED COMPUTING

A tecnologia da informação já se consolidou como ferramenta essencial para as instituições, afetando rotinas, ações, estruturas e modelos de negócios. Os alto custos associados ao consumo e provimento de recursos de TI e a necessidade de uma maior agilidade para responder às mudanças casam-se às propostas da computação orientada a serviços, ou SOC, cujo paradigma de design é composto por princípios que, quando absorvidos, criam condições para o cumprimento dos objetivos e benefícios almejados pela orientação a serviços, entre eles o aumento do retorno de investimento, da agilidade, da reusabilidade e da interoperabilidade, assim como um maior alinhamento entre TI e modelo de negócios.



Figura 2.1: Princípios relacionados ao SOA e seus benefícios

O paradigma de design da orientação a serviços norteia a configuração do suprimento e consumo dos recursos de TI, em especial as soluções lógicas ou softwares. Apesar do sucesso de outros paradigmas na computação, por exemplo a orientação a objetos, nota-se que problemas distintos, ou seja, da eficiência, agilidade e retorno de investimento na composição de aplicações, ainda deveriam ser atacados. Assim, as aplicações silos com pouca interoperabilidade e sem reuso passaram a ser vistas como um modelo ineficiente, visto que aumentam o custo de desenvolvimento de novas aplicações a longo prazo e reduzem

drasticamente a agilidade ou tempo de resposta às novas demandas, sobretudo quando os modelos de negócio exigem maior flexibilidade. Dessa forma, os benefícios idealizados pelo SOC fizeram desse paradigma um foco de estudos em diversos centros acadêmicos, sobretudo à partir do surgimento da arquitetura orientada a serviços, ou SOA. Também é notável a adesão de grandes vendedores da indústria de TI a essa tecnologia, fomentando tanto pesquisa quanto sua adoção em instituições públicas e privadas.

O termo SOC representa um aglomerado de conceitos relacionados à orientação a serviços [2]. Além de um paradigma de design, engloba princípios, padrões, governança e arquitetura, notavelmente o SOA. Nesse trabalho, o termo SOC poderá ser usado para se referir de forma genérica à orientação a objetos como um todo.

### 2.1.1 Service Orientation History

Historicamente, o paradigma de orientação a serviços tem como base diversos outros paradigmas e arquitetura. Por meio da adaptação e amadurecimento de conceitos já experimentados em TI e dada a conjectura de que alguns objetivos poderiam ser melhor conquistados, formou-se o que hoje define a orientação a serviços em termos de princípios de design, arquitetura e tecnologias capazes de implementá-los.

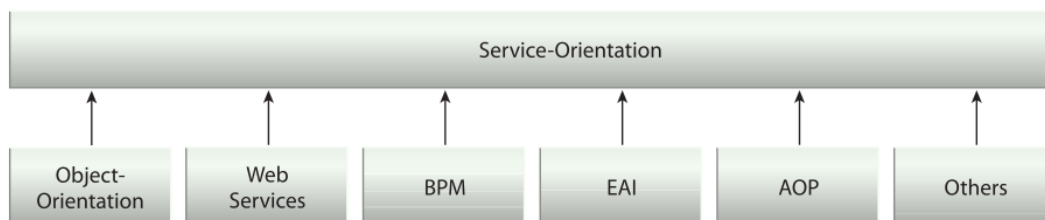


Figura 2.2: Origens da Orientação a Serviço

### 2.1.2 Component Based Architecture

Entre as mais importantes influências está a arquitetura baseada em componentes. Com ela, a orientação a serviços compartilha visões, uma vez que ambas se sustentam sobre os conceitos de unidades lógicas auto-contidas, auto-descritas, modulares, encapsuladas, que fazem uso de interfaces, contratos e especificações com possibilidade de composição com outras unidades.

No entanto, serviços diferenciam-se fundamentalmente de componentes. Enquanto componentes variam entre modelos *white box*, *gray box* ou *black box*, de acordo com o nível de customização realizável, os serviços são sempre hermeticamente encapsulados e distribuídos em forma de *black boxes*. Nos primeiros,

foca-se na especificação técnica do conjunto de funcionalidades que desempenham, permitindo o uso das mesmas nos códigos que as invocam uma vez que os componentes tenham sido devidamente importados ou ligados à aplicação. Sua especificação pode incluir também uma definição abstrata de sua estrutura interna [3]. Para os últimos, contratos são estabelecidos em conjunto às descrições das funcionalidades expostas pelas interfaces do serviço. Segundo [4], a interação dos serviços ocorre de forma desacoplada por meio do uso de parâmetros pré-estabelecidos em sua descrição.

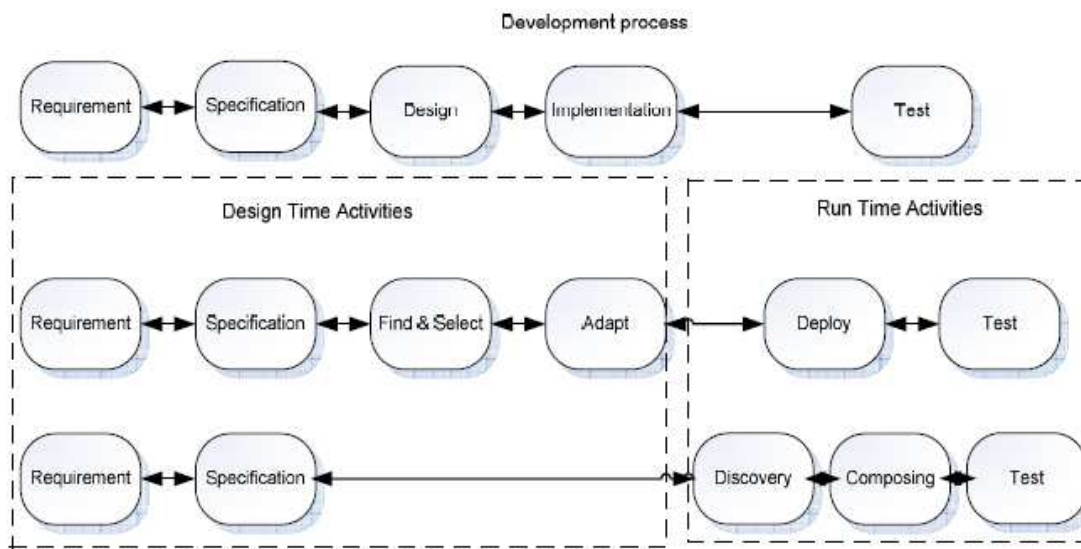


Figura 2.3: Comparação entre atividades típicas de componentes e serviços

A composição dos serviços é feita de forma dinâmica em tempo de execução, enquanto a composição de componentes é feita na fase de design ou em tempo de execução. Isso relaciona-se ao tipo de acoplamento entre provedor e consumidor, sendo essa a principal diferença entre ambos paradigmas. Nos componentes é frequente um maior acoplamento, de forma que um grau mais elevado de cumplicidade entre provedor e consumidor é exigida de acordo com o tipo de interação. Outro fator está na dependência quanto ao modelo de componentes usado, que deve ser compatível. Em contraste, nos serviços o baixo acoplamento entre provedor e consumidor é um princípio a ser seguido e um de maior importância no paradigma de orientação a serviços, havendo transparência no processo de troca de mensagens. Dessa forma, o provedor irá desfrutar de liberdade e flexibilidade para a escolha da tecnologia para implementação, assim como o contratante irá somente se ater, funcionalmente, à interface. Essa separação propicia a criação de serviços abstratos e reutilizáveis, o que segue princípios de design da orientação a serviços. Para componentes, é possível uma otimização da performance em detrimento da flexibilidade por meio da

composição na fase de design, mas dificilmente se concebe um cenário de integração de um componente sem uma documentação ou até mesmo a comunicação com a equipe responsável pelo seu provimento.

Mesmo com modelos que definem a distribuição de componentes, i.e. J2EE e CORBA, os resultados obtidos são de naturezas distintas. O processo de integração remota previsto pela distribuição de componentes mantém o aspecto citado para a sua especificação, uma vez que irá, de forma refinada, distribuir modelos de objetos, seus estados e suas propriedades, enquanto os serviços estão num patamar menos elaborado de integração, fornecendo funcionalidades coesas em alto nível no padrão requisição e resposta. De fato, há casos em que um serviço irá fazer uso de componentes em sua implementação, que por conceito independe de sua descrição, desde que mantenha a funcionalidade por ela exposta. Portanto, componentes e serviços se situam em domínios diferentes, assim como as necessidades atendidas e os problemas por eles enfrentados [5]. Em suma, os maiores ganhos dos serviços em relação aos componentes estão na maior interoperabilidade, na maior gama de possibilidades para composições, uma vez que permite integração com com serviços de terceiros e por fim no menor acoplamento entre provedor e consumidor.

### **2.1.3 Object Orientation**

A orientação a objetos também teve grande influência na orientação a serviços. Muitos conceitos por ela reforçados tem precedência nesse paradigma, entre eles o alinhamento do modelo de negócio e da TI, visto que as classes são moldadas à partir de conceitos e objetos reais, muitas vezes representações do próprio negócio. O aumento da robustez, uma vez que a orientação a objetos mantém um forte processo de design da solução com uso de diagramas UML e possui um complexo modelo para exceções e rotinas de teste. O aumento da extensibilidade, haja vista a característica modular das classes e as diversas possibilidades de herança, polimorfismo e associações capazes de estender o escopo da solução. Da flexibilidade, dado o uso de encapsulamento e abstração, sendo possível a adaptação do funcionamento a novas realidades. Por fim, da reusabilidade e da produtividade, dada a existência de classes abstratas e de códigos genéricos e reutilizáveis [2].

Uma notável diferença entre os dois paradigmas está no escopo onde atuam. A orientação a objetos visa estruturar soluções localizadas e podem vir a ser aplicada diversas vezes, podendo englobar todo o cenário da instituição através de diferentes ciclos. A orientação a serviços desde sua aplicação inicial irá atuar em porções mais extensas, podendo ou não chegar a todo o contexto de soluções de tecnologia da organização. Isso se deve a alguns princípios do SOC, que evita a formação de aplicações silos e fomenta a formação de um inventário de serviços granulares e interoperáveis de modo a viabilizar a composição dos mesmos com



base na reusabilidade. Assim, é possível afirmar que a aplicação da orientação a objetos geralmente leva a soluções isoladas ou à criação de um nível limitado de reusabilidade e abrangência, enquanto a aplicação de orientação a serviços tende a transformar o cenário como um todo englobando unidades lógicas que, se obtiverem êxito em alcançar os objetivos e princípios do design da orientação a serviços e se vencerem as implicações técnicas consequentes, irão disponibilizar um conjunto de soluções flexíveis e que respondem a novas demandas de negócio com maior velocidade e menor custo.

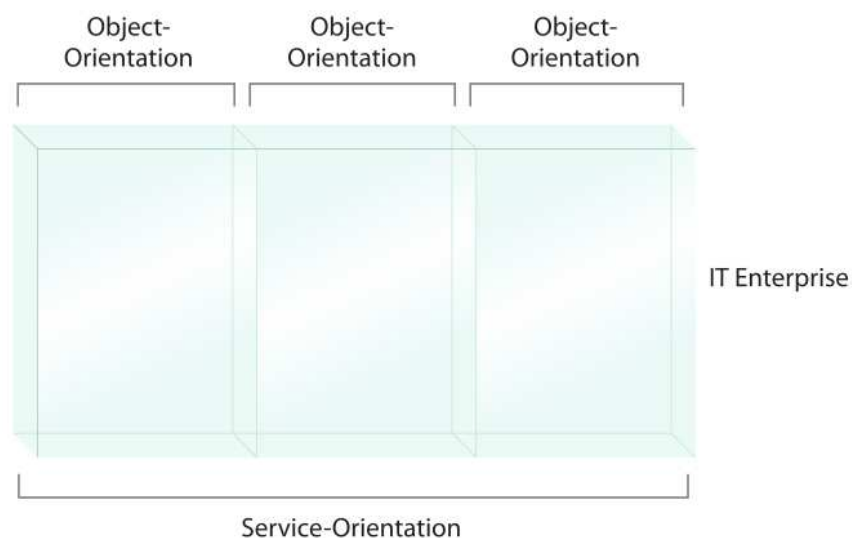


Figura 2.4: Comparação entre escopos da Orientação a Objetos a da Orientação a Serviços

Outro importante modelo arquitetural influente no SOC é a o EAI ou Enterprise Application Integration. Esse conjunto de princípios arquiteturais foca na integração de soluções heterogêneas de uma empresa. Seu principal componente é denominado *middleware*, ou aplicação intermediária, e tem a função de transportar mensagens entre as aplicações de maneira transparente e irrestrita. O motivo pelo qual essa arquitetura influenciou o surgimento da orientação a serviços está em seu caráter distribuído e integrador, possibilitando que unidades funcionais de lógica troquem mensagem e cooperem formando novas aplicações e dando vazão às demandas de negócios com maior agilidade. No entanto, é visível que mesmo havendo uma integração, as aplicações envolvidas ainda se situam num modelo de aplicações silos e com pouco potencial de reuso e de serem compostas de maneiras diversas e flexíveis. Assim, visionando as possibilidades trazidas por um inventário de soluções coesas, abstratas e de fácil composição, o SOC inovou com os princípios de design em que se baseia para criação e manutenção de serviços, forma como passaram a ser denominadas as unidades lógicas disponíveis e integradas pela camada intermediária.

#### 2.1.4 Web Services

Seria imprescindível, na tarefa de descrever as origens da computação orientada a serviços, deixar de citar a importante contribuição dos Web Services. Mais que uma influência, os Web Services estão na raiz do funcionamento da arquitetura SOA, mesmo sendo essa agnóstica à tecnologia usada na implementação de seus serviços.

Web Services surgiram anteriormente ao SOC, oferecendo uma integração transparente entre aplicações distribuídas por meio de um sistema de troca de mensagens que faz uso de protocolos padronizados e especificações. Em sua primeira geração, tentativas foram feitas com o uso direto de Remote Procedure Calls, ou RPC. Apesar desse modelo já ser conhecido e usado, seu uso em Web Services não obteve apoio devido ao acoplamento resultante de configurações específicas para cada linguagem e logo deixou de ser praticado de forma direta !!CITE. Também foram desenvolvidas as especificações centrais, tais quais o WSDL, linguagem responsável pela descrição das funcionalidades de um serviço, o UDDI, que consiste num registro de negócios e serviços, onde é possível encontrar (descobrir) Web Services registrados por provedores, além das especificações para o aumento da interoperabilidade, conhecidas por WS-I profiles. O SOAP, protocolo de troca de mensagens em formato XML e que possibilita o uso protocolos de transporte conhecidos da internet, i.e. HTTP, SMTP, ou outros independentes, passou a ser usado em Web Services mais simples e logo em Web Services descritos com WSDL e associados a repositórios UDDI, chamados de *Big Web Services* !!CITE. Além de promover a integração de mensagens, o SOAP também possibilita a interoperabilidade de chamadas RPC por meio do encapsulamento dessas chamadas.

A segunda geração de Web Services trouxe avanços em segurança, transações entre serviços e garantias na troca de mensagens, entre outros [2]. Essas tecnologias e especificações relacionadas seguem a nomenclatura WS-\* e tem a característica de serem passíveis de composição entre si e fornecerem um rico conjunto de ferramentas para o ambiente de Web Services. Entretanto, críticas quanto à complexidade e enrijecimento da arquitetura como consequência ao uso destes padrões levaram ao fortalecimento de outros modelos. O REST, ou Representational State Transfer, trata exatamente da transferência de representações de recursos usando um conjunto restrito de ações, denominadas verbos, usualmente por meio do protocolo HTTP e permite a criação de RESTfull Web Services. Nesses há uma liberdade de diversas decisões arquiteturais, ou *freedom-from-choice*, mantendo-se a premissa de requisições *stateless*, um vez que toda informação necessária ao processamento da requisição está nela contido. Em REST, cada recurso irá possuir uma única identificação URI, por meio da qual operações, i.e. GET, POST, PUT e DELETE, serão realizadas com os recursos. Entre os problemas encontrados com o uso dessa arquitetura está a ausência de

suporte a vários atributos QoS. A segurança, por exemplo, fica a cargo do protocolo de transporte, que na prática é limitado ao HTTPS. Não há garantias para as mensagens, salvo em implementações próprias, e a composição é feita por meio de *mashups*, termo proveniente da Web 2.0 e que define uma combinação de funcionalidades entre fontes diversas e heterogêneas de maneira livre e sem especificações, ou seja, caberá ao usuário dos serviços elaborar seu modelo de composição ou simplesmente não utilizar nenhum modelo, algo mais propício ao ambiente de aplicações de sítios da internet que ao ambiente controlado de empresas.

Atualmente o debate entre ambas tecnologias para implementação de Web Services aponta para a contextualização de seu uso: cenários mais simples e ad-hoc tendem a obter melhor eficiência com RESTFull Web Services. Em casos de maior complexidade acompanhadas de transações, segurança, garantias e outros aspectos de QoS oferecidos pelos WS-\* com manutenção da interoperabilidade, sobretudo em ambientes corporativos, o uso de Big Services mostra-se mais viável, especialmente a longo prazo [6].

### **2.1.5 Aspect Oriented Programming**

O AOP define como principal objetivo a separação de tópicos, ou *separation of concerns*, de forma a identificar interesses comuns entre aplicações que poderão ser modularizados e reutilizados. Em geral a modularização é feita em tópicos que se espalham por diversas camadas de abstração, ou *crosscutting concern*.

A principal correlação da AOP com a orientação a serviços está no princípio da reutilização de unidades lógicas que são abstratas e agnósticas à lógica de negócio e ao restante da aplicação, proposta que se encontra definida em princípios de design dos serviços.

### **2.1.6 Business Process Model**

Por BPM entende-se a disciplina composta por técnicas e métodos voltados ao design, gerenciamento, aperfeiçoamento e controle de processos de negócio, ou Business Process, com intuito de atingir os objetivos de uma organização [7]. Um BP é caracterizado pelo conjunto de práticas e atividades encarregadas de produzirem um produto ou serviço. É comum representá-los em gráficos com sequências de atividades, que se iniciam no objetivo e finalizam em seu cumprimento. O grande foco do BPM está na otimização dos resultados obtidos pelos BPs e no alinhamento da TI com os objetivos estratégicos da organização. É, em essência, uma extensão ou amadurecimento de práticas já conhecidas pela administração desde que surgiram os conceitos relacionados ao processo produtivo, mas o grande salto qualitativo se deu com o

uso da TI para a automação da gestão de BPs, sendo a primeira experiência conhecida desenvolvida para General Electric em 1954 [8].

Uma das atividades centrais do BPM é o modelamento, ou Business Process Modeling, que realiza o descobrimento e modelagem de BPs numa organização, além de melhorias a processos já existentes. A representação é feita pelo padrão BPMN com o uso de diagramas gráficos que se assemelham aos diagramas de atividade da linguagem UML [9].

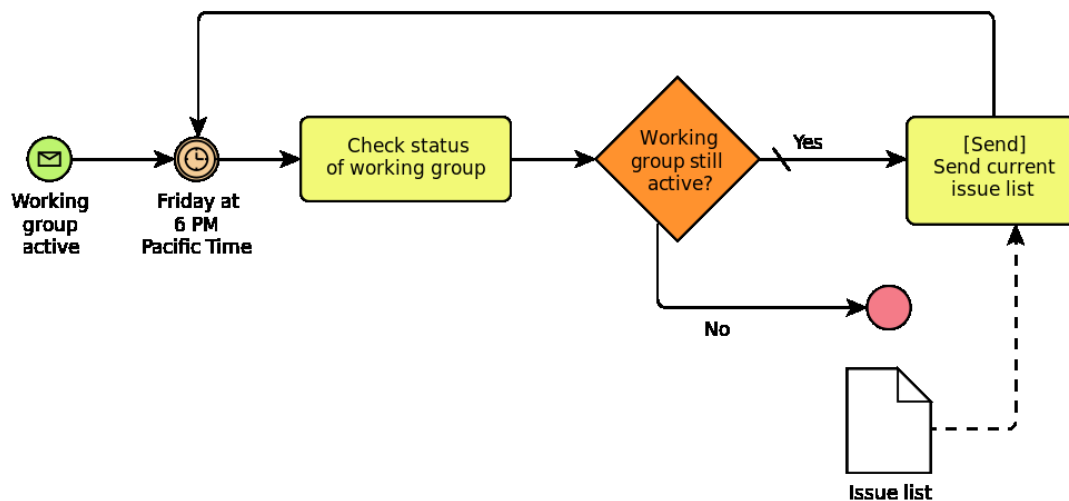


Figura 2.5: Modelagem de um BP por meio de um diagrama BPMN

Outra importante atividade está no monitoramento de processos, ou BAM, em geral mais completa e detalhada que o monitoramento obtido com ferramentas do Business Process Management Suit, o BPMS. Assim, o rastreamento de informações acerca do serviço oferece estatísticas sobre sua performance e estado.

Um dos princípios da orientação a serviços está no alinhamento entre modelo de negócios e tecnologia da informação. Como propósito, o BPM possibilita que processos desse modelo estejam mais visíveis, adaptáveis e extensíveis, o que aumenta o tempo de resposta e oferece ferramentas para o mapeamento entre requisitos de negócios e recursos da TI, o que provê alinhamento entre os dois mundos e um maior valor agregado às soluções. O BPM ocupa o papel de organizar uma abstração de alto nível dos processos de negócio, enquanto o SOA irá se adequar no suporte a esses processos com características de agilidade, flexibilidade e eficiência que serão passadas ao negócio e consequentemente à organização. Dessa forma, a orientação a serviços por meio do SOA e de seus princípios de design surgiu num contexto que já se mostrava apto a recebe-la e não como uma revolução daquilo que já existia.

### 2.1.7 Service Oriented Design

O principal resultado da aplicação do design estabelecido pelo SOC está na criação de unidades lógicas denominadas serviços. Assim, ao invés de aplicações isoladas que trazem consigo todas as funcionalidades, são criadas unidades coesas, granulares, abstratas e de baixo acoplamento e alta interoperabilidade cujo acesso é feito por uma interface que encapsula sua lógica e cujo funcionamento é bem descrito e garantido por um contrato. Com isso, é possível realizar uma composição ágil que irá servir uma ou, idealmente, várias aplicações por meio do reuso e da orquestração de serviços.

Vários princípios de design regem a confecção dos serviços e de seus inventários.

### 2.1.8 Service Oriented Architecture

Os mesmos agentes responsáveis pela comercialização de SOA, também são responsáveis pela vulgarização do termo, ao ponto do mesmo ter se tornado ambíguo e com significados diversos.

Com base nos princípios de SOC, criou-se uma arquitetura responsável por guiar o processo de adoção desse paradigma. O SOA, ou *Service Oriented Architecture*, define uma série de processos, papéis, regras, modelos e camadas, entre outros componentes da arquitetura, propondo transformações importantes no modus operandi no setor de tecnologia da informação. Dada sua abrangência e complexidade de fundamentos, o SOA é protagonista no avanço do SOC, tendo seu conceito se difundido de forma mais ampla que o primeiro, haja vista que recebeu grande atenção do meio acadêmico, como ilustra nosso mapeamento de estudos, além de ter sido um produto de destaque para grandes vendedores da indústria de TI. Entre as prioridades do SOA apontada pelo *SOA Manifesto* estão: o valor de negócio sobre a estratégia técnica adotada, os objetivos estratégicos sobre os benefícios específicos de projeto, a interoperabilidade intrínseca sobre as integrações customizadas, os serviços compartilhados sobre implementações com propósito específico, a flexibilidade sobre otimização e o refinamento evolutivo sobre a perfeição inicial [10].

Por qualidade de serviço ou QOS de um sistema entendem-se seus atributos ou requisitos não funcionais. Ou seja, são aspectos qualitativos de estado e não definições de funcionamento, podem ser citados a performance, a segurança, a disponibilidade, a modificabilidade, a escalabilidade, etc. Com o amadurecimento e evolução da engenharia de software, tais atributos tornaram-se cruciais para a gerência de riscos e também de custos das aplicações, participando ativamente na definição da arquitetura do sistema com peso maior que os próprios requisitos funcionais.

A performance está diretamente relacionada ao tempo, seja pelo tempo de resposta, pela quantidade de

requisições por unidade de tempo ou pela capacidade de atender à requisição numa quantidade de tempo determinada[11]. Nesse sentido, o uso de serviços distribuídos acarreta prejuízos à performance inerentes à rede utilizada, além de processamento extra devido ao tratamento e encaminhamento das mensagens entre serviços, especialmente quando trata-se de formatos baseados em texto, como o XML, que possuem tamanho maior que sua representação binária. Também irá prejudicar a performance o processo de descobrimento e alocação de serviços na situações em que esse processo não é pré-estabelecido.

A interoperabilidade relaciona-se à capacidade dos componentes de compartilharem informação e operarem de forma acordada[11]. Dado que o ambiente SOC pressupõe a criação de serviços e, especificamente em SOA, serviços implementados com diversas tecnologias sem prejuízo à intercomunicação, verifica-se que um dos principais benefícios trazidos por essa arquitetura é para a interoperabilidade.

A segurança, como um atributo de QOS, possui alguns princípios. A autenticidade trata da capacidade do sistema em identificar a autoria da informação enviada. A integridade irá resguardar a informação tal qual foi enviada, sem alterações. A confidencialidade garante que somente atores ou entidades autorizadas tenham acesso às informações. O paradigma SOC tem desafios nesses critérios de qualidade à medida em que os serviços são distribuídos em ambientes internos restritos ou na própria internet. É imediata a percepção de que a troca de informações entre as unidades lógicas do sistema pode acarretar sérios riscos à segurança, estando os esforços disponíveis para garanti-la limitados pela necessidade de se manter a interoperabilidade e a performance no ambiente SOC, já que requerem a adoção adicional de especificações, protocolos e consequentemente uma maior redundância de informação e processamento.

A disponibilidade define o tempo em que o sistema estará operacional e portanto respondendo a requisições. Tratando-se de serviços, cada um irá possuir uma disponibilidade que irá influenciar diretamente na disponibilidade dos sistemas que dependem desses serviços, portanto o ambiente SOC possui o desafio de manter-se operacional diante de suas unidades com particularidades próprias. Como possíveis estratégias para mitigar os problemas causados por serviços inoperantes estão a replicação e o balanceamento de carga, além da troca do provedor do serviço. Nesse sentido, o monitoramento, a gerência e a seleção de serviços são processos importantes em SOC e sua automação um dos maiores desafios.

A confiabilidade é um atributo relacionado à disponibilidade e trata da capacidade do sistema em operar sem erros. Mais uma vez, no ambiente distribuído essa capacidade irá depender da confiabilidade dos serviços que compõe o sistema, mas também das mensagens enviadas e recebidas. Dessa forma, um sistema SOC confiável deverá utilizar especificações e protocolos que garantam a confiabilidade das mensagens ou fazer uso de ferramentas e implementações que cuidem desse processo.

A modificabilidade mede a capacidade do sistema de sofrer modificações com menores custos e maior velocidade [12].

O ambiente proposto pelas arquiteturas orientadas a serviço está sujeito a condições particulares diferentes daquelas já estudadas e conhecidas em outros paradigmas, havendo variáveis que elevam a complexidade da análise de parâmetros de qualidade, tanto na fase de planejamento quanto em fase de execução por meio do monitoramento e da gerência dos serviços, sendo esse um obstáculo sólido à adoção de arquiteturas como o SOA. Nesse sentido, o presente estudo visa mapear publicações cujo foco está na qualidade de serviços em SOC, contemplando cenários com ou sem o uso de SOA, proporcionando uma redução da incerteza quanto ao atual estado de desenvolvimento da ciência contribuinte ao tema abordado e quanto aos desafios e avanços já conquistados.

## 2.2 SYSTEMATIC MAPPING STUDY

O *Systematic Mapping Study* propõe uma abordagem que preza pela amplitude em detrimento da profundidade de análise. Com isso, é possível incluir um número mais significativo de publicações e um escopo de estudos mais amplo que aquele obtido pelo *Systematic Literature Review*, tendo esse último uma abordagem que difere-se pela maior precisão e riqueza de detalhes em sua classificação, podendo ser aplicado num possível trabalho futuro, cujo enfoque seriam subáreas de interesse identificadas pelo primeiro processo de mapeamento aqui apresentado.

Um dos maiores revezes dos SRs está no esforço considerável por ele requerido [1]. Dada a vastidão de tópicos abordados pela computação orientada a objetos, é factível supor que um trabalho de SR se tornaria inviável devido ao elevado número de estudos publicados e um número reduzido de pessoal responsável pela suas análises, uma vez que deseja-se obter um panorama geral ou especificando tópicos de grande amplitude.

Um Mapping Study é designado para prover uma ampla visão de determinada área de pesquisa, para verificar se indícios de pesquisa existem em determinados tópicos e prover indicações quantitativas das evidências. [13].

O MS segue um processo sistêmico que visa garantir um nível de qualidade adequado aos resultados obtidos. Algumas passos são diretamente herdados dos SRs, enquanto outros são adaptados para se garantir a viabilidade de seus objetivos. Cada passo do processo de mapeamento irá gerar um resultado, sendo o

resultado final o mapeamento propriamente dito [1].

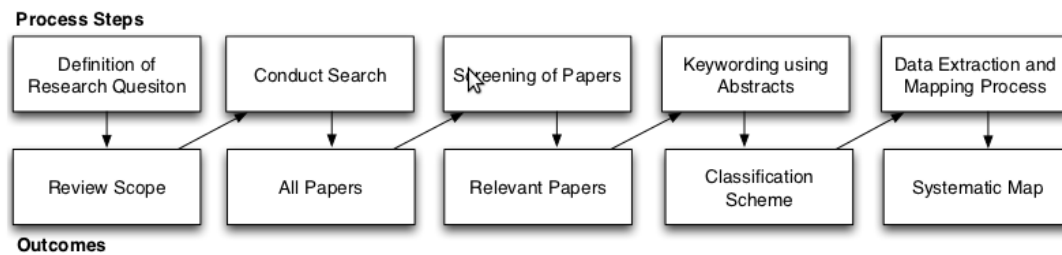


Figura 2.6: Processo para o Systematic Mapping Study

Para o presente trabalho, definiu-se que o foco irá tratar de aspectos de qualidade na orientação a serviços, sobretudo no que se refere aos resultados obtidos. Com essa decisão entende-se que o SOC já atingiu patamares elevados de funcionalidade por meio de padrões e especificações, arquiteturas e processos. Segundo [14], devido ao crescente acordo na implementação e gerência de aspectos funcionais de serviços, tal qual a adoção de WSDL para a descrição, SOAP para troca de mensagens, ou WS-BPEL para a composição, os interesses de pesquisadores estão se voltando aos aspectos não funcionais de aplicações orientadas a serviços.



## 3 IMPLEMENTAÇÃO

Na implementação deste trabalho, põe-se em prática os conceitos apresentados no capítulo Embasamento Teórico, principalmente no que é relativo ao Facebook. Este aplicativo, como dito no capítulo Motivação, é de forma geral um coletor e analisador de dados da rede social online Facebook. As vantagens de se coletar dados utilizando uma API e os motivos da escolha da rede social online Facebook foram abordados no capítulo Motivação. Neste capítulo apresentar-se-á o aplicativo desenvolvido e serão abordados os principais temas relacionados ao aplicativo. Este capítulo está dividido em seis seções, abordadas na seguinte ordem: Descrição Geral da Implementação; Tecnologias Utilizadas; Regras de Negócio, Análise de Requisitos e Metodologias; Metodologias e Testes do Projeto; Estratégias para Atingir Público Alvo. Neste capítulo são também referenciados todos os anexos deste trabalho.

### 3.1 DESCRIÇÃO GERAL DA IMPLEMENTAÇÃO

A implementação deste trabalho é um software coletor e analisador de informações de estudantes da UnB presentes na RSO Facebook. Trata-se de um aplicativo da Facebook que se enquadra na categoria de aplicativos de utilidade social. Entende-se por "estudantes da Universidade de Brasília presentes na rede social online Facebook" usuários da rede social online Facebook que tem a página da Universidade de Brasília adicionada na parte de educação do seu perfil. Por conveniência, chamaremos de agora em diante este grupo de Rede UnB Facebook. O aplicativo funciona da seguinte maneira. As informações da Rede UnB Facebook são coletadas e salvas em um banco de dados. As informações salvas são analisadas de forma a identificar padrões através de um método de categorização (mineração de dados). Estas informações são coletadas do perfil do DCE-UnB no Facebook. As informações são coletadas por meio de um aplicativo do Facebook após devida autorização do usuário. Por final, as informações analisadas são apresentadas para o usuário final.

### 3.2 TECNOLOGIAS UTILIZADAS

O software foi desenvolvido com tecnologias atuais e agéis que eram dominadas pelo desenvolvedor do software, autor deste trabalho. Dessa maneira, buscou-se obter a melhor performance na coleta e análise

das informações, aumentar o tempo de vida e a escalabilidade do software. As tecnologias abordadas a seguir são divididas em Tecnologias do Facebook e Tecnologias Gerais.

### **3.2.1 Tecnologias do Facebook**

No capítulo Embasamento Teórico, mostrou-se que o Facebook está em processo de mudança de tecnologias antigas, mas ainda funcionais, para tecnologias atuais e compatíveis com o resto da web. As tecnologias escolhidas para o desenvolvimento da implementação deste trabalho foram todas atuais: GRAPH API, FQL, Javascript SDK com OAuth 2.0, PHP SDK v3.0.1. O FQL foi implementado utilizando as funções "FB.Data.query" e "query.wait", providas pela Javascript SDK e que fazem consultas via Graph API. Estas funções juntamente com a função FB.api, também provida pela Javascript SDK, foram as responsáveis pela coleta dos dados.

Durante o desenvolvimento do aplicativo; foram realizados alguns testes utilizando o Console de Teste FQL.query e a ferramenta Graph API Explorer; ambos presentes no portal de desenvolvedores do Facebook. Na subseção Testes, apresentar-se-á mais detalhes sobre a fase de testes do aplicativo.

### **3.2.2 Tecnologias Gerais: Linguagens de Programação, Bibliotecas e IDEs**

O aplicativo do Facebook foi desenvolvido utilizando um misto de três linguagens de programação diferentes. As linguagens foram escolhidas levando em consideração três pré-requisitos estabelecidos pelo desenvolvedor do aplicativo. Os pré-requisitos foram o domínio das linguagens a serem escolhidas pelo desenvolvedor do software (autor deste trabalho) e a clareza e a quantidade de informações úteis disponíveis na documentação oficial de desenvolvedores do Facebook e na web relacionadas as linguagens a serem escolhidas. Além da existência de um SDK, relacionado as linguagens a serem escolhidas, que facilitasse a construção do aplicativo. Após serem analisados os requisitos, foram escolhidas as seguintes linguagens de programação: PHP, Javascript e Actionscript. A escolha das duas primeiras linguagens citadas são justificados pelo fato do desenvolvedor ter domínio sobre estas, a documentação das mesmas para a construção de aplicativos no Facebook é clara e a maior existente. Além disso, estas linguagens tem SDKs que agilizam a construção de aplicativos, tornam a aplicação mais escalável e fácil de ser configurada. A escolha do ActionScript, linguagem proprietária da empresa Adobe, é justificada principalmente pela capacidade de se fazer interfaces gráficas de usuários ricas de forma rápida. Esta linguagem foi utilizada junto ao IDE Flash Professional CS5.5 Tiral para desenvolver os infográficos. Nota-se que nenhuma das linguagens sozinhas

atende plenamente as necessidades de desenvolvimento do aplicativo sozinhas, por isso a necessidade das três.

A linguagem de programação PHP foi utilizada na estruturação do aplicativo, gravação dos dados coletados no banco de dados, para prover autenticação do aplicativo, cobrar permissões do usuário, para acessar as informações de *signed request* do usuário entre outras utilidades.

A linguagem de programação Javascript foi utilizada em diversas frentes do software: Interface Gráfica do Usuário, Chamada de Requisições POST e algumas funções intrínsecas da API do Facebook. A Interface Gráfica do Usuário relativa as duas abas onde serão apresentados os infográficos foi criada utilizando-se a biblioteca javascript JQuery conhecida como jquery-ui.min.js (v1.8). As chamadas de requisições POST foram feitas utilizando a biblioteca javascript JQuery conhecida como jquery.min.js (v1.4). As funções intrínsecas da API do Facebook se referem a coleta de dados dos usuários, utilização de cookies e autenticação OAuth 2.0.

Durante o processo de codificação do aplicativo foi utilizado o IDE Zend Studio Trial (v8.0.1). Este IDE foi escolhido, pois foi premiado pela Infoworld 2011 como Melhor IDE PHP.

As tecnologias utilizadas pouparam tempo de desenvolvimento do software. Mais detalhes das linguagens de programação PHP e JavaScript estão no capítulo Embasamento Teórico. A estrutura de arquivos do projeto implementado e a descrição das formas de acesso e configuração do aplicativo no Facebook está disponível no anexo I e II, respectivamente.

### **3.2.3 Servidor de Hospedagem e de Domínio**

O aplicativo está rodando em servidor de hospedagem configurado com Apache Web Server (v2.2.+); PHP (v5.3.6+) e PostgreSQL (v9.0.4+). Trata-se de um servidor de hospedagem alugado pela empresa Locaweb. O domínio do aplicativo é [www.adsum.com.br/facebook/pedro3/](http://www.adsum.com.br/facebook/pedro3/) está registrado junto a FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) por intermédio da empresa RegistroBR, que realiza registros de domínios para Internet no Brasil.

## **3.3 REGRAS DE NEGÓCIO, ANÁLISE DE REQUISITOS E METODOLOGIAS**

Após a realização de uma pesquisa sobre os principais métodos de coleta e análise de dados em redes sociais abordados no capítulo Embasamento Teórico, optou-se dentre várias opções pela construção de

um aplicativo para a rede social online Facebook. A tomada desta decisão levou a realização de mais uma pesquisa sobre como desenvolver aplicativos para o Facebook que resultaram nas seguintes decisões: utilização das Tecnologias do Facebook, Linguagens de Programação, Bibliotecas, IDEs e Servidor de Hospedagem com as características citados na seção passada deste capítulo.

### **3.3.1 Elaboração da Regra de Negócio e Análise de Requisitos**

Após a realização da primeira pesquisa citada na subseção anterior foi definida uma parte da regra de negócio do aplicativo que está descrita na seção Descrição Geral da Implementação. Logo após, foi realizada uma segunda pesquisa, também descrita na subseção anterior, que definiu alguns requisitos do projeto. Esta pesquisa foi orientada aos pré-requisitos definidos pelo desenvolvedor. A análise de requisitos foi realizada de forma bastante intuitiva. Sabia-se que seriam usadas as *APIs* do Facebook. Portanto, decidiu-se fazer uso das Tecnologias do Facebook mais atuais. Sabia-se que o aplicativo coletaria e analisaria alguns dados de usuários do Facebook pertencentes ao grupo Rede UnB Facebook, porém não se sabia de forma detalhada quais dados seriam estes e quais seriam as análises feitas com base nos mesmos. Neste momento, o processo de análise de requisitos parou e o processo de desenvolvimento das regras de negócio recomeçou. Os dados e as análises foram definidos para complementar as regras de negócio. Esta parte das regras de negócio se encontram na subseção seguinte. Logo após a complementação das regras de negócio recomeçou-se a análise de requisitos da mesma maneira. Sabia-se que o ambiente de funcionamento do aplicativo desenvolvido é a web; portanto foram escolhidas Linguagens de Programação, Bibliotecas e IDEs para este ambiente de acordo com os pré-requisitos do desenvolvedor. Sabia-se que para coletar dados de usuários do Facebook são necessárias permissões, descritas no capítulo Embasamento Teórico, portanto foram listados as permissões necessárias para a coleta dos dados necessários. E assim por diante todo o escopo do processo de análise de requisitos e das regras de negócio foi definido. Foram demandadas cerca de 30 horas para a análise de requisitos; 100 horas para a elaboração das regras de negócio e 250 horas de codificação, num total de 380 horas. Neste relatório continham as datas e os horários que as atividades relativas ao desenvolvimento do software começavam e terminavam. Estas eram separadas em três sessões: análise de requisitos, regras de negócio e codificação. Uma observação importante é que o processo de análise de requisitos demandou apenas 8% do total de horas dedicadas ao projeto. Acredita-se que isto tenha acontecido, porque as regras de negócio do aplicativo e os pré-requisitos do desenvolvedor foram bastante específicos, deixando o espaço amostral de possíveis requisitos pequeno. Outra observação importante é que no processo de desenvolvimento deste software as análises de requisitos não dependeram

do fato de as regras de negócio estarem totalmente concluídas. Notou-se assim um desenvolvimento de software iterativo.

### 3.3.2 Regras de Negócio: Coletar e Analisar o que?

Como dito anteriormente; a implementação deste trabalho é um aplicativo que coleta e analisa alguns dados de usuários da Rede UnB Facebook. Nesta subseção são descritos de forma detalhada quais os dados que o aplicativo coleta e analisa e assim as regras de negócio do aplicativo são totalmente abordadas.

Nesta primeira versão do aplicativo, versão beta, coletar-se-á somente os dados relativos as informações do perfil dos amigos dos usuários. Estes dados não incluem o conteúdo gerado por estes usuários nem os dados do próprio perfil dos amigos destes usuários. É importante saber que somente os dados disponíveis serão coletados, ou seja, aqueles existentes no perfil do usuário. Os dados que serão coletados são descritos abaixo e estão classificados em quatro camadas.

- Principal

- nome; uid; quantidade de amigos (grau); gênero sexual; data de nascimento; naturalidade; localidade da moradia atual; email; visão política; religião; estado civil; website;

- Trabalho

- nome(s) da(s) instituição(ões) que trabalha ou que trabalhou; localidade(s) da(s) respectiva(s) instituição(ões); cargo(s) que trabalha ou que trabalhou na(s) respectiva(s) instituição(ões); data de entrada e data de saída na(s) respectiva(s) instituição(ões); instituição que está trabalhando atualmente e desde quando

- Educação

- nome(s) da(s) instituição(ões) que estuda ou que estudou; tipo da(s) instituição(ões), por exemplo, ensino superior; Ano de entrada na(s) respectiva(s) instituição(ões) área de concentração na(s) respectiva(s) instituição(ões)

- Páginas "curtidas"

- nome(s) da(s) seguinte(s) categoria(s) página(s) curtidas: filmes, músicas, livros e show televisivos.

Como abordado no capítulo Embasamento Teórico, as páginas representam os gostos do usuário e as categorias de páginas mais populares são relacionadas filmes e livros. É importante notar nesta subseção como os dados do perfil de um usuário no Facebook são bem organizados, justificando mais uma vez a escolha desta rede social online para a coleta de dados.

Na análise de dados, descrita abaixo nesta subseção, são mostrados os tratamentos que serão feitos com os dados coletados. As conclusões alcançadas a partir destes tratamentos serão abordadas no capítulo seguinte chamado Resultados Obtidos.

Serão feitos os seguintes tratamentos de dados: Descobrir-se-á padrões de usuários da Rede UnB Facebook categorizando estes padrões segundo o método de categorização explicado no capítulo Embasamento Teórico. Assim, a partir das informações salvas descobriremos várias "subredes" de acordo com os dados do perfil do usuário. Logo, existiram várias "subredes", como por exemplo, a "subrede" de organização educacional de ensino médio, a "subrede" de organização de trabalho, "subrede" de gênero sexual, "subrede" idade e assim por diante. É importante abordar aqui que também haverá a subrede de dados não preenchidos pelo perfil da pessoa, ou seja, dados que foram salvos como uma *string* sem caracteres ou nulos.

### 3.3.3 Principais Requisito para Coleta e Análise

A partir das regras de negócio descritas na subseção passada foram pesquisados os requisitos para continuar a construção do aplicativo. Abordar-se-á nesta subseção estes requisitos: a escolha das permissões para a coleta dos dados e alguns detalhes de tratamento destes dados. Para a escolha das permissões, fez-se uma simples associação de quais permissões seriam necessárias para coletar todas informações. Esta associação foi feita com base na tabela de permissões presente no portal de desenvolvedores do Facebook. Abaixo são apresentadas as 17 permissões necessárias para a coleta de dados que pretendida pelo aplicativo:

- user\_about\_me, friends\_about\_me, friends\_likes, friends\_online\_presence, friends\_relationships, friends\_work\_history, friends\_activities, friends\_education\_history, friends\_hometown, friends\_location, friends\_photo\_video\_tags, friends\_videos, friends\_birthday, friends\_events, friends\_interests, friends\_photos, friends\_religion\_politics, friends\_website, email

Todas as permissões citadas acima referem-se a dados exclusivos do usuário. Contudo, o aplicativo desenvolvido para este trabalho pede ao usuário mais permissões. Isto é feito para que seja possível em

versões futuras do aplicativo a coleta de outros dados do usuário, sem que haja necessidade de pedir estas informações para os usuários novamente. O resto das permissões pedidas são mostradas abaixo de forma categorizada:

- Permissões para Coleta do próprio usuário:
  - user\_about\_me, user\_likes, user\_online\_presence, user\_relationships, user\_work\_history, user\_activities, user\_education\_history, user\_hometown, user\_location, user\_photo\_video\_tags, user\_videos, user\_birthday, user\_events, user\_interests, user\_photos, user\_religion\_politics, user\_website
- Permissões extendidas
  - read\_stream, off-line\_access, online\_presence, publish\_stream

Estas permissões garantem a coleta dos seguintes dados do usuário: todos os dados citados na subseção anterior, mudando apenas o fato de que agora estes são relacionados aos amigos do usuário analisado; o conteúdo gerado pelo usuário (CGU) em tempo real; a presença online. Além disso, a permissão off-line\_access permite que o aplicativo acesse os dados a qualquer hora do dia e a permissão publish\_stream garante que o aplicativo gere qualquer tipo conteúdo como se fosse o usuário.

Os detalhes relativos ao tratamento dos dados são abordados nas seguintes situações. Deve-se extrair do dado coletado email apenas o domínio deste email, já que será apenas este o objeto a ser analisado. Outro caso que se enquadra neste requisito é quando um dado coletado não está preenchido no perfil de um usuário, ou seja, não existe. Este dado tem valor nulo ou indefinido. Se um dado com valor nulo ou indefinido for acessado pelo aplicativo ocorre um erro e o aplicativo para de funcionar. Dessa maneira, não é continuada a coleta e os demais dados já coletados não são gravados no banco de dados. Daí, a necessidade de verificação em todos os dados colhidos se estes existem.

### **3.4 MINERAÇÃO DE DADOS**

O processo de Mineração de Dados, em inglês *Data Mining*, consiste em explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, detectando assim novos subconjuntos de dados. Dessa forma, obter a partir dos dados operativos brutos, informação útil para subsidiar a tomada de decisão nos escalões médios e altos da empresa. O método de categorização consiste em agrupar perfis, dentre todos

já salvos, de acordo com uma ou mais informações colhidas. Por exemplo: onde a maioria dos alunos da Rede UnB Facebook estudou no ensino médio. Os dados categorizadas e o conhecimento gerado a partir destes dados que serão apresentados no próximo capítulo, Resultados.

### 3.5 METODOLOGIAS, PADRÕES E TESTES DO PROJETO

O aplicativo desenvolvido não seguiu nenhum *Design Pattern*(Padrões de Projetos) como o modelo *MVC*(*Model View Controller*) e nenhuma Metodologia de Desenvolvimento de Software como *RUP* (*Rational Unified Process*) ou utilização de Diagramas *UML*, (*Unified Modeling Language*). Além disso, não foram utilizadas Metodologias de Gerenciamento de Projetos de Software como *SCRUM* ou *XP* ou Metodologias de Testes. Dessa maneira, a construção deste aplicativo não faz uso de metodologias comuns em projetos de Software atuais e que são consideradas boas práticas da Engenharia de Software. O projeto foi feito dessa maneira; pois a parte de codificação do software não é complexa nem extensa. Além disso, o software foi desenvolvido por apenas uma pessoa. Dessa maneira, não houve a necessidade de seguir metodologias comuns do desenvolvimento de software e nem houveram dificuldades de desenvolvimento resultantes desta decisão. Os testes foram feitos em paralelo ao desenvolvimento do projeto e realizados pelo próprio desenvolvedor. Como o software não tinha muitos pontos de função e realizava apenas duas tarefas complexas que são a análise e a coleta de dados. Nesta fase de desenvolvimento do projeto, não houveram grandes dificuldades nem necessidade de grandes correções ou modificações de fluxos a partir da detecção de erros.

### 3.6 ESTRATÉGIAS PARA ATINGIR PÚBLICO ALVO

O público alvo do projeto (alunos e ex-alunos da UnB que tem perfil na RSO Facebook) foi atingido por meio da coleta de dados de amigos do perfil do DCE-UNB, mostrado na figura 1.8, que conta com 3490 "amigos", sendo que a maioria destes "amigos"alunos e ex-alunos da UnB como será mostrado no próximo capítulo, Resultados. Foi assinado um termo de contrato entre o autor deste trabalho e o DCE-UNB afirmando que os dados não seriam vendidos ou cedidas a outros projetos que não fossem também aprovados pela instituição. Outra cláusula acorda que os dados serviriam somente para identificar padrões no público alvo do projeto e nada além disso. Os dados de todo público alvo do projeto seria viável através do acesso a conta da página da Universidade de Brasília, mostrada na figura 1.7. Isto porque toda



vez que alguém adiciona a Universidade de Brasília na parte de educação do seu perfil, esta marcação (/textitmetadados) é notificada no perfil desta página. Assim, através do número de notificações recebidas sabe-se exatamente quantas pessoas fizeram a ação descrita anteriormente. Estima-se que existam mais de 36.000 alunos da UnB representados por perfis no Facebook. Esta estimativa é baseada na quantidade de membros da comunidade da "UnB - Universidade de Brasília" no Orkut, mostrada na figura 1.4, em que a maioria dos membros era aluno ou ex-aluno da UnB. A partir dessa base, leva-se em conta que como dito no capítulo Introdução hoje o número de usuários e acessos ao Facebook são maiores do que no Orkut portanto notou-se uma forte migração de usuários da primeira RSO citada para a segunda, ainda mais considerando os alunos da Universidade de Brasília que são mais "elitizados digitalmente".

## 4 ESTUDO DE CASO

Nesta seção, discutiremos os dados categorizadas que foram salvos no banco de dados e descreveremos o conhecimento que pode ser gerado a partir destes dados. Estima-se que o aplicativo tenha coletado cerca de 11% do espaço amostral universal da rede UnB Facebook, segundo estimativas mostradas na sessão "Estratégias para atingir público alvo" do capítulo anterior. Esta porcentagem é bastante significativa e serve para ser aproveitada como caso de estudo deste trabalho. Em pesquisas sociológicas comuns a porcentagem de dados colhidos raras vezes atingi esta porcentagem. O que revela maior acurácia destas pesquisas e portanto possibilidades de análises mais verossimilhanes. Um exemplo disso é relativo as pesquisas eleitorais que atingi somente 0,00158421% do espaço amostral universal. Estatísticas: coletou-se dados de 3.960 pessoas do total de 36.000. No total temos 40.000 informações dos perfis dos usuários classificados entre informações pessoais e páginas "curtidas", sendo 10.000 são informações não preenchidas. A partir dos dados coletados fizemos diversos tipos de análises obtivemos resultados interessantes. Nas seções deste capítulo detalhar-se-á quais foram estes resultados e análises.

### 4.1 DISCUSSÃO DE RESULTADOS

A seguir serão apresentados os dados categorizadas em "subredes" conforme descrito no capítulo anterior. A partir destes dados teremos subsídios para fazermos algumas análises que geraram conhecimento útil. São apresentados somente os dois dados melhor "rankeados" na "subrede" quando existe a possibilidade de se cadastrar mais de dois dados. quantidade de amigos (grau); gênero sexual; data de nascimento; naturalidade; localidade da moradia atual; email; visão política; religião; estado civil; website;

- média da quantidade de amigos: 260
- Genero sexual: Masculino(53%) e Feminino(45%); Não-preenchido: (2%)
- Idade: 17-18(20%); 19-21(30%) ; 21-23(24,5%) ; Outros (25,5%) ;
- Trabalho: Universidade de Brasília (2%); Outros: (40%) ; Não-preenchido(54%)
- Educação: Alub (6%); Galois (3,5%); Outros(30%); Não-preenchido(60%)
- Naturalidade: Brasília (60,1%) , Nordeste (15,3%) e Outros(8,7%) ; Não-preenchido(4%)

- Visão Política: Partido dos Trabalhadores (2,7%); Outros: (2,3%); Não-preenchido(95%)
- Religião: Católico(2,1%); Outros(2,9%); Não-preenchido(92,4%)
- Moradia Atual: Brasília(90,4%); Outros(4,1%); Não-preenchido(4,5%)
- Estado civil: Solteiro (40%<sup>1</sup>); Algum tipo de relacionamento (2%) ; Não-preenchido(56%)
- Email: gmail (50%) ; hotmail (30%); Outros(10%) ;
- Website: preenchido (1%); Não-preenchido(97%)
- Filmes: The GodFather (6%); Tropa de Elite (4%); Outros (15%); Não-preenchido(75%)
- livros: O Pequeno Príncipe (3%) ; Outros (30%); Não-preenchido(77%)
- Músicas: Black Eyed Peas (7%); Jorge Ben Jor (5%); Outros (30%) ; Não-preenchido(77%)
- Shows Televisivos: Pânico da TV (4,1%) ; Outros (6,7%); Não-preenchido(56%)

O único dado disponível na página de estatísticas oficiais do Facebook, mas que não tem correlação nenhuma com os dados coletados é a média da quantidade de amigos que na Rede UnB Facebook é de 260 e nas estatísticas oficiais do Facebook é de 130 usuários. Isto pode mostrar que o usuários desta rede fazem mais uso do Facebook que a média global de usuários. Pode-se se base dessa análise consiste que para o usuário aceitar a solicitação de amizade de uma pessoa e esta adicionar no número de amigos do seu perfil esta necessita de pelo menos fazer *login* no Facebook. Outra possível análise é que pessoas dentro da Universidade de Brasília se relacionam com mais pessoas do que a média global, o que talvez possa ser generalizado às pessoas que estudam em grandes organizações educacionais. Isto pode representar maior troca de experiências entre pessoas vivenciados durante a fase da Universidade. Pode se ainda tentar generalizar que os (ex-)alunos da UnB são mais incluídos digitalmente do que a média global. Todavia, estas suposições só poderiam ser comprovadas através de análises de CGU e modelagem do comportamento do usuário.

A disparidade de 8% a mais dos perfis pertencentes ao gênero sexual masculino pode querer dizer que usuários são mais influentes a novas tendências. Outra análise baseada no fato de a porcentagem de homens solteiros na rede da UnB ser maior que é a de mulheres solteiras em 15% pode significar que eles estão utilizando o Facebook para procurar um relacionamento.

Percebe-se também que o colégio de ensino médio mais bem rankeado na coleta é o ALUB que dessa maneira pode comprovar o slogan da rede de colégios e cursinho que é "A que mais aprova na UnB". Em

segundo lugar tem-se o colégio Galois que é mais elitizado. Supõe-se que poucas pessoas que estudaram na UnB teriam acesso, mas está bem colocado porque a maioria dos estudantes que estudaram neste colégio e fazem UnB preencheram esta informação na parte de educação do seu perfil.

A parte de trabalho foi muito "disputada" no ranking uma vez que a diferenças eram pequenas de cerca de 0.1%. Contudo notou-se que a maioria dos estudantes que fazem UnB trabalham também na própria universidade. Talvez por causa no movimento estudantil, ganhando bolsas da universidade, pbics entre outros.

A idade da maioria dos estudantes da Rede UnB Facebook é na faixa 19 a 21 anos.

Percebe-se também a presença de pessoas de outras cidades do país principalmente do nordeste e centro-oeste para UnB.

A visão política e religiosa está entre as três menos preenchidas dos perfis da Rede UnB Facebook. Supõe-se que esta rede não se importe muito ou não quer divulgar para os amigos a religião, já que na universidade de Brasília há muito diversidade e a pessoa com esta informação pode ser alvo críticas.

A informação menos preenchida da rede é websites. Supõe-se que usuários não tenham o próprio website para compartilhar suas informações ou coisas que julgam interessantes com seus amigos porque tudo está "embutido" dentro do Facebook.

O livro mais famoso é o pequeno príncipe e representado por apenas 3% quando é comparado com show televisivo a diferença 1,1%, o que pode mostrar que apesar do ambiente acadêmicos os (ex-)alunos se interessam mais por programas televisivos do que por livros.

Vale lembrar ainda que todas as análises aqui feitas representam as informações capturadas do perfil do DCE-UNB, ou seja, talvez a captura plena das informações através de outros perfil como por exemplo o perfil da Universidade de Brasília, mostrado na figura 1.6, possa mudar o *ranking* de alguns dados, principalmente aqueles que a porcentagem está mais próxima como educação de ensino médio.

## 5 CONCLUSÃO

Conclui-se assim as atividades necessárias para o desempenho deste trabalho. Ao fim de 1 ano de pesquisa na área macro Redes Sociais e 3 meses intensos de pesquisas e desenvolvimento da implementação, incluindo levantamentos e elaboração de texto, entende-se que os objetivos foram em boa parte alcançados.

Apesar do trabalho não coletar na sua totalidade todo o universo amostral de (ex-)alunos da Rede UnB Facebook, por não ter acesso ao perfil da página da UnB no Facebook como descrito no capítulo Embasamento Teórico, foi coletado uma porcentagem relevante para validar a pesquisa sociológica que mostrou ser muito maior que pesquisa sociológicas comuns.

O capítulo Embasamento Teórico serve de guia geral de Redes Sociais Online introduzindo os principais conceitos e a importância destas cativando o leitor de forma que este continue a leitura do trabalho. No capítulo Embasamento Teórico faz-se uma conceituação dos principais conceitos por trás das técnicas de coleta e análise de dados em redes sociais. Logo em seguida neste mesmo capítulo é feita uma sumariação das técnicas de coleta e análise de dados que serve para motivar e basear pesquisadores para futuros estudos na área.

Na parte de implementação temos uma descrição detalhada do software feito, que pode servir como base para replicação de trabalhos similares em outras organizações educacionais. Como abordou-se neste mesmo capítulo o software está disponibilizado no sistema de versionamento social GIT (Anexo III) de forma gratuita, podendo ser baixado e utilizado de forma gratuita.

No capítulo Resultados mostram-se as vantagens e as utilidades das técnicas apresentadas, motivando ainda mais futuros pesquisadores. Nota-se aqui ainda que trata-se de um trabalho original com resultados originais e únicos no Brasil.

Entende-se ainda que o trabalho aqui desenvolvido conseguiu alcançar sua motivação de ser útil para a UnB, principal do autor deste trabalho. Dado estas considerações, espera-se que este trabalho seja bem utilizado pela UnB e seja dada continuidade deste dentro da UnB.

O trabalho aqui apresentado apresenta O capítulo de resultados nos leva a crer que Conclui-se assim que os métodos de coleta e análise de dados em redes sociais virtuais tem a utilidade e a vantagens. As utilidades comprovadas foram as seguintes... As vantagens comprovadas foram as seguintes...

As atividades práticas e os resultados necessários para a comprovação desta conclusão foram demons-

tradas com detalhes. Assim, acentuamos o resultado demonstrado. Além disso, foram sugeridas melhorias para o aplicativo e formas de atingir na prática estas melhorias. para que esta comprovação fosse mais notada inicialo deste trabalho foram .

Apesar das informações coletadas serem virtuais, o impacto dos resultados destes métodos são visíveis no mundo real.

## **5.1 SUGESTÃO DE MELHORIAS**

Com dito no capítulo Implementação a versão implementada deste projeto é a versão beta, coletando somente dados do perfil do usuário. A seguir, descrever-se-á algumas funcionalidades a serem implementadas em versões futuras:

- OLAP
- Análises de CGU e Redes Complexas
- Modelagem do comportamento do usuário

## REFERÊNCIAS

- [1] PETERSEN, K. et al. Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering*, v. 17, n. 1, p. 1–10, 2007.
- [2] ERL, T. *SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2007. ISBN 0132344823.
- [3] BREIVOLD, H. P.; LARSSON, M. Component-based and service-oriented software engineering: Key concepts and principles. In: *Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications*. Washington, DC, USA: IEEE Computer Society, 2007. p. 13–20. ISBN 0-7695-2977-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=1302497.1302990>>.
- [4] CRNKOVIC, I.; CHAUDRON, M.; LARSSON, S. Component-based development process and component lifecycle. In: *Proceedings of the International Conference on Software Engineering Advances*. Washington, DC, USA: IEEE Computer Society, 2006. p. 44–. ISBN 0-7695-2703-5. Disponível em: <<http://dl.acm.org/citation.cfm?id=1193212.1193814>>.
- [5] PATIL, S. *Integration Approaches: Web Services vs Distributed Component Models PART II*. 2003. [Online; accessed 20-February-2012]. Disponível em: <<http://soa.sys-con.com/node/39754>>.
- [6] PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. "big" web services: making the right architectural decision. In: *Proceedings of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008. (WWW '08), p. 805–814. ISBN 978-1-60558-085-2. Disponível em: <<http://doi.acm.org/10.1145/1367497.1367606>>.
- [7] JESTON, J.; NELIS, J. *Business Process Management: Practical Guidelines to Successful Implementations*. Butterworth-Heinemann, 2008. ISBN 9780750686563. Disponível em: <<http://books.google.com.br/books?id=QI9aaKRIPsC>>.
- [8] POSSEL, J. *Accenture's History: The beginnings in the early 1950's*. January 2011. [Online; accessed 29-February-2012]. Disponível em: <<http://www.accenture-blogpodium.nl/about-accenture/accentures-history-the-beginnings-in-the-early-1950s/>>.
- [9] WHITE, S. A.; CORP, I. B. M. Process modeling notations and workflow patterns. *Business, Future Strategies Inc.*, v. 21, n. 1999, p. 1–25, 1999.

- [10] VARIOS. *Soa Manifesto*. 2012. [Online; accessed 12-December-2011]. Disponível em: <<http://www.soa-manifesto.org/>>.
- [11] O'BRIEN, L.; MERSON, P.; BASS, L. Quality attributes for service-oriented architectures. In: *Proceedings of the International Workshop on Systems Development in SOA Environments*. Washington, DC, USA: IEEE Computer Society, 2007. (SDSOA '07), p. 3–. ISBN 0-7695-2960-7. Disponível em: <<http://dx.doi.org/10.1109/SDSOA.2007.10>>.
- [12] CLEMENTS, P.; KAZMAN, R.; KLEIN, M. *Evaluating Software Architectures: Methods and Case Studies*. [S.l.]: Addison-Wesley, 2001. ISBN 978-0-201-70482-2.
- [13] KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. *Version*, v. 2, n. EBSE 2007-001, p. 2007?01, 2007.
- [14] PAPAZOGLU, M. P.; HEUVEL, W.-J. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 16, p. 389–415, July 2007. ISSN 1066-8888. Disponível em: <<http://dx.doi.org/10.1007/s00778-007-0044-3>>.



## **6 ANEXOS**

**6.1 ANEXO I - ESTRUTURA DE ARQUIVOS DO PROJETO IMPLEMENTADO**

**6.2 ANEXO II - FORMAS DE ACESSO E CONFIGURAÇÃO DO APLICATIVO NO FACEBOOK**

**6.3 ANEXO III - CD COM ARQUIVOS DO PROJETO IMPLEMENTADO**

