

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Departamento de Sistemas e Computação

Relatório de Estágio

Desenvolvimento e Evolução de Sistemas de  
Monitoramento da Informação em Redes Sociais

**Amaury Bartolomeu Carneiro de Medeiros**  
Estagiário

**Cláudio de Souza Baptista**  
Orientador Acadêmico

**Alessandro Barbosa Lima**  
Supervisor Técnico

**Campina Grande – PB**  
**Dezembro de 2010**



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

Desenvolvimento e Evolução de Sistemas de Monitoramento da  
Informação em Redes Sociais

APROVADO EM \_\_\_\_\_

**BANCA EXAMINADORA**

*Cláudio de Souza Baptista*

---

Prof. Dr. Cláudio de Souza Baptista

ORIENTADOR ACADÊMICO

*Joseana Macêdo Fachine*

---

Profa. Dra. Joseana Macêdo Fachine

MEMBRO DA BANCA

*Leandro Balby Marinho*

---

Prof. Dr. Leandro Balby Marinho

MEMBRO DA BANCA

## AGRADECIMENTOS

Gostaria de agradecer à E.life, pela oportunidade de realizar o estágio integrado na empresa. Assim, pude aplicar os conhecimentos adquiridos durante a graduação no mercado de trabalho. Agradeço ao Supervisor Técnico Alessandro Barbosa e ao também dono da empresa Jairson Vitorino, pelo apoio dentro da E.life. Agradeço, também, aos professores Cláudio Baptista e Joseana Fachine, pelo acompanhamento no decorrer do estágio. Gostaria de agradecer, ainda, aos desenvolvedores da empresa, do núcleo de Recife, pela ajuda tanto na explicação do funcionamento dos sistemas, quanto no desenvolvimento de atividades.

## SUMÁRIO

Apresentação .....	5
Resumo .....	6
Lista de Siglas e Abreviaturas .....	7
Lista de Figuras .....	8
Lista de Quadros.....	9
<b>1. INTRODUÇÃO .....</b>	<b>11</b>
<b>2. AMBIENTE DE ESTÁGIO.....</b>	<b>13</b>
2.1 A EMPRESA .....	13
2.2 LOCAL DE DESENVOLVIMENTO DO ESTÁGIO .....	14
2.2.1 Aspectos Positivos .....	14
2.2.2 Aspectos Negativos .....	14
2.3 CARGA HORÁRIA E PERÍODO DE ESTÁGIO.....	14
2.4 INFRAESTRUTURA DE HARDWARE E SOFTWARE .....	15
2.5 SUPERVISÃO E ORIENTAÇÃO .....	15
2.5.1 Orientador Acadêmico .....	15
2.5.2 Supervisor técnico .....	15
<b>3. FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS .....</b>	<b>17</b>
<b>4. ATIVIDADES REALIZADAS.....</b>	<b>21</b>
4.1 CONFIGURAÇÃO DO AMBIENTE E ENTENDIMENTO DO SISTEMA .....	22
4.2 LEVANTAMENTO DE EMPRESAS NO LINKEDIN .....	23
4.3 SOCIAL CRM .....	26
4.4 E.LIFE PEOPLE SEARCH E SERVLETS .....	29
4.5 E.LIFE SAC E JSP.....	33
4.6 CONEXÃO COM O FACEBOOK E OAUTH .....	36
4.7 VALIDAÇÃO.....	39
<b>5. CONSIDERAÇÕES FINAIS.....</b>	<b>41</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>43</b>
<b>APÊNDICE A – PLANO DE ESTÁGIO.....</b>	<b>46</b>

## APRESENTAÇÃO

Como parte das exigências do curso de Ciência da Computação, da Universidade Federal de Campina Grande, para cumprimento da disciplina de estágio integrado, apresenta-se o relatório de estágio, contendo as atividades realizadas pelo aluno Amaury Bartolomeu Carneiro de Medeiros, durante o período letivo 2010.2, sob orientação acadêmica do Prof. Dr. Cláudio de Souza Baptista e supervisão técnica de Alessandro Barbosa Lima. O objetivo do estágio foi desenvolver e evoluir sistemas de monitoramento da informação em redes sociais.

O estágio foi realizado na empresa E.life Monitor Estudos de Mercado LTDA (E.life), na modalidade *home-office*.

O conteúdo do relatório segue a distribuição abaixo:

Seção 1 – Introdução.

Seção 2 – Ambiente de Estágio.

Seção 3 – Fundamentação Teórica e Tecnologias Utilizadas.

Seção 4 – Atividades do Estágio.

Seção 5 – Considerações Finais.

Referências Bibliográficas.

Apêndices.

## RESUMO

Atualmente, com a disseminação das mídias sociais na internet, é bastante comum que quem tenha acesso à internet, possua um perfil em alguma rede social. Estas redes atuam como um motor de comunicação dos usuários na rede. Usualmente, ao se deparar com um problema num produto adquirido numa empresa, o consumidor prefere buscar uma solução na internet a entrar em contato com a empresa que forneceu aquele produto. A informação parece vir mais rápido, quando é proveniente de outras pessoas, que muitas vezes já passaram pelo mesmo problema do usuário que está buscando ajuda. O grande problema é que essa informação que circula na internet passa despercebida pelas empresas. A proposta da E.life é desenvolver sistemas capazes de monitorar essa informação e colocá-las sob supervisão da empresa, que passa a ganhar o poder da informação, podendo interagir com os clientes ou fazer mudanças nos seu produtos, de acordo com os comentários dos clientes, que se sentem mais confortáveis para criticar, quando não estão interagindo diretamente com a empresa. O escopo do estágio realizado – com atividades descritas nesse relatório – é, justamente, o desenvolvimento e evolução de sistemas no intuito de realizar esse monitoramento.

## LISTA DE SIGLAS E ABREVIATURAS

**API** – *Application Programming Interface* – Interface de Programação de Aplicações

**BOSS** – *Build your Own Search Service*

**BM2** – BuzzMonitor, versão 2.0.

**CEO** – *Chief Executive Officer* – Diretor Executivo

**CSS** – *Cascading Style Sheets*

**CTO** – *Chief Technology Officer* – Diretor de Tecnologia

**DAO** – Data Access Object

**E.life** – E.life Monitor Estudos de Mercado LTDA

**Embedded** – Laboratório de Sistemas Embarcados e Computação Pervasiva

**FTP** – *File Transfer Protocol* – Protocolo de Transferência de Arquivos

**HTML** – *HyperText Markup Language*

**HTTP** – *HyperText Transfer Protocol*

**HTTPS** – *HyperText Transfer Protocol Secure*

**IDE** – *Integrated Development Environment* – Ambiente de Desenvolvimento

**JSON** – *JavaScript Object Notation*

**JSP** – *JavaServer Pages*

**MVC** – *Model-View-Controller*

**OO** – Paradigma de Linguagem de Programação Orientado a Objetos

**SO** – Sistema Operacional

**SGBD** – Sistema Gerenciador de Banco de Dados

**SQL** – *Structured Query Language*

**UFCG** – Universidade Federal de Campina Grande

**URL** – *Uniform Resource Locator*

**XML** – *Extensible Markup Language*



## LISTA DE FIGURAS

Figura 1 - BUZZMONITOR 2.0 .....	23
Figura 2 - WIDGET TOP URLS .....	24
Figura 3 - FLUXO DE CONTROLE DA AUTENTICAÇÃO DO LINKEDIN .....	25
Figura 4 - PLANILHA COM DADOS DO LINKEDIN .....	26
Figura 5 - FUNCIONAMENTO DO E.LIFE SAC.....	27
Figura 6 - PAINEL DO USUARIO NO SACQUESTIONS .....	28
Figura 7 - FILTRO POR DATA NO SACQUESTIONS .....	28
Figura 8 - DIAGRAMA DE CLASSES DO E.LIFE PEOPLE SEARCH.....	30
Figura 9 - INTEGRAÇÃO DE SAC QUESTIONS E E.LIFE PEOPLE SEARCH .....	33
Figura 10 - INTERFACE ORIGINAL DO E.LIFE SAC.....	35
Figura 11 - ADIÇÃO E REMOÇÃO DE LABELS NO E.LIFE SAC.....	35
Figura 12 - FILTROS DO E.LIFE SAC .....	36
Figura 13 - LOGIN COM FACEBOOK.....	38
Figura 14 - REQUISIÇÃO DE PERMISSÃO DO ELIFESAC PARA USAR OS DADOS DO FACEBOOK .....	38

## LISTA DE QUADROS

Quadro 1 - TECNOLOGIAS UTILIZADAS.....	17
Quadro 2 - COMPARATIVO ENTRE OS TEMPOS ESTIMADOS E GASTOS DAS ATIVIDADES RELIZADAS .....	21

SEÇÃO I

INTRODUÇÃO

# 1. INTRODUÇÃO

Existe uma frase no Talmud (alicerce de leis e rituais judaicos) que diz: “A coisa principal da vida não é o conhecimento, mas o uso que dele se faz” (DJMAL, 2003). Apesar do contexto histórico e temporal da frase, a natureza do estágio não fica muito aquém do seu sentido. O estágio é a oportunidade de aplicar conhecimentos adquiridos no decorrer da graduação a um ambiente profissional. Quando ingressar no mercado de trabalho, o estagiário levará consigo uma carga de experiência maior, o que pode ser crucial, quando for comparado a alguém que vivenciou apenas a teoria.

Essa maneira de pensar motivou Amaury Medeiros a ingressar num estágio integrado, durante o segundo semestre letivo de 2010. O estágio contemplou um total de 300 horas, dispostas em 20 horas semanais. O estagiário trabalhou, majoritariamente, em casa, com exceção de algumas vezes, quando precisou usar as dependências da Universidade Federal de Campina Grande (UFCG), do qual ele é aluno.

O objetivo geral do estágio foi evoluir e desenvolver sistemas voltados para a análise de informações que circulam em mídias sociais. A empresa no qual o estágio foi desenvolvido faz uso desses sistemas para gerar relatórios periódicos para as empresas clientes, que desejam saber o que é falado sobre a sua marca na internet.

Objetivos específicos:

- a) Desenvolver ferramentas web para obter dados sobre usuários;
- b) Adquirir experiência no âmbito do desenvolvimento de sistemas *web*, visto a sua ascensão no mercado de trabalho;
- c) Desenvolvimento de soluções *web* para comunicação entre empresa-cliente e consumidor, por meio do sistema E.life SAC;
- d) Adaptação do estagiário ao mercado de trabalho, em um ambiente ágil, com produção feita sob demanda.



SEÇÃO II

AMBIENTE DE ESTÁGIO

## 2. AMBIENTE DE ESTÁGIO

Nesta seção, serão abordadas informações sobre a empresa na qual foi realizado o estágio, apresentando um breve histórico sobre essa, bem como suas áreas de atuação. Informações sobre a infraestrutura utilizada para o desenvolvimento do estágio também estão no escopo da seção.

### 2.1 A EMPRESA

No ano de 2004, dois amigos tiveram a idéia de montar uma empresa juntos. A idéia surgiu depois que Alessandro, um dos amigos, se deparou com alguns problemas no seu carro – um Chevrolet. Ele tentou, então, entrar em contato com o serviço de atendimento da Chevrolet, via email, mas a resposta não foi de grande ajuda.

Numa última tentativa, Alessandro tentou realizar uma busca na internet, por via do motor de busca da Google. A pesquisa lhe encaminhou para um blog, no qual diversas pessoas relatavam o mesmo problema que ele. A solução para o problema custou-lhe R\$0,50. Este fato, segundo o Diretor de Tecnologia (CTO) da empresa, Jairson Vitorino, é emblemático no contexto da fundação da E.life.

Aquela situação fez os amigos Jairson e Alessandro – atual Diretor Executivo (CEO) da empresa – que o consumidor de um produto poderia conhecer mais sobre o mesmo do que o próprio serviço de atendimento da empresa que o produziu. Então, os sócios montaram a E.life, no intuito de monitorar a informação que circula entre os usuários na internet e fornecer os dados para as empresas que estiverem interessadas em contratar tal serviço.

Atualmente, a E.life é a líder em monitoração, análise da mídia gerada pelo consumidor e gestão de relacionamento em mídias sociais na América Latina e Portugal. Sua sede localiza-se em São Paulo – SP, mas os funcionários estão distribuídos remotamente em cidades como Recife – PE, Campina Grande – PB ou até mesmo em outros países, como Alemanha e Portugal.

Os serviços da E.life focam na análise de mídia gerada pelos consumidores na *web*. A empresa fornece os resultados encontrados aos clientes por meio de relatórios fiéis ao que foi encontrado na pesquisa, mesmo que estes não sejam favoráveis ao cliente. Isso que diferencia a E.life – uma empresa focada na inteligência de mercado – de empresas de marketing puro, que muitas vezes amenizam os fatos, quando a situação da empresa não vai bem.

## 2.2 LOCAL DE DESENVOLVIMENTO DO ESTÁGIO

O estágio foi desenvolvido na modalidade *home-office*, ou seja, o estagiário realizou suas atividades, quase que inteiramente, no seu domicílio. Em alguns momentos, foi necessário utilizar a internet do Laboratório de Sistemas Embarcados de Computação Pervasiva (Embedded), da UFCG. As reuniões com o supervisor técnico e alguns outros funcionários da empresa foram realizadas pelo Skype, via conferência de voz.

Além do supervisor técnico, também participavam das reuniões o CTO Jairson Vitorino e os desenvolvedores da cidade de Recife. Esses desenvolvedores atuavam como mediadores entre o CTO e o estagiário.

No decorrer do estágio, foi realizada uma reunião presencial, na cidade de Recife, na qual buscou-se a integração dos funcionários da cidade com os estagiários de Campina Grande. Estes, ainda, tiveram a oportunidade de conhecer o CTO da empresa, que atualmente reside na cidade de Mannheim, na Alemanha.

### 2.2.1 ASPECTOS POSITIVOS

Trabalhar no conforto do lar possui grandes vantagens. Não existem problemas com transporte, que podem inclusive ocasionar perda de tempo além do previsto. Também existe uma grande tranquilidade com relação ao horário, que pode ser distribuído da forma que o estagiário bem entender.

### 2.2.2 ASPECTOS NEGATIVOS

A modalidade *home-office*, por trazer muita comodidade ao estagiário, pode acabar acostumando-o a trabalhar em horários bastante diversificados, causando uma possível falha de comunicação com os outros funcionários da empresa, que trabalham no mesmo escopo e precisam manter contato com ele. Além disso, o estagiário mescla o local de trabalho com o ambiente familiar, o que pode acarretar uma perigosa mistura, a qual ele pode não conseguir discernir qual ambiente ele deve estar em determinado momento.

## 2.3 CARGA HORÁRIA E PERÍODO DE ESTÁGIO

O estágio pela empresa E.life iniciou-se no dia 11 de agosto de 2010, tendo fim no dia 24 de novembro de 2010. O estagiário cumpriu um total de 300 horas, divididas em 20 horas semanais. A distribuição das horas no decorrer da semana, entretanto, variava de acordo com as demais tarefas do estagiário e o tempo diário disponível.

## 2.4 INFRAESTRUTURA DE HARDWARE E SOFTWARE

As atividades de estágio foram desenvolvidas em um notebook Dell, modelo Vostro 3500, com processador Intel® Core™ i5 e 3GB de memória RAM. A conexão com a internet foi feita pelas redes wireless do domicílio do estagiário e do Embedded.

As ferramentas de software utilizadas no decorrer do estágio encontram-se listadas, além de acompanhadas de uma breve descrição, na seção 3.

## 2.5 SUPERVISÃO E ORIENTAÇÃO

Esta seção provê informações básicas sobre o orientador acadêmico e o supervisor técnico.

### 2.5.1 ORIENTADOR ACADÊMICO

**Nome:** Prof. Dr. Cláudio de Souza Baptista

**Endereço Profissional:** Universidade Federal de Campina Grande – Departamento de sistemas e Computação - Rua Aprígio Veloso, s/n, Bodocongó. CEP: 58429-900 Campina Grande - PB, Brasil.

**Endereço de email:** [baptista@dsc.ufcg.edu.br](mailto:baptista@dsc.ufcg.edu.br)

### 2.5.2 SUPERVISOR TÉCNICO

**Nome:** Alessandro Barbosa Lima

**Endereço Profissional:** E-Life Monitor Estudos de Mercado Ltda – Al. Ribeirão Preto, 130, Sala 41, Bela Vista. C.E.P: 01.331-000. São Paulo – SP

**Endereço de email:** [albali@elife.com.br](mailto:albali@elife.com.br)



SEÇÃO III

---

FUNDAMENTAÇÃO

TEÓRICA

E

TECNOLOGIAS

UTILIZADAS

### 3. FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS UTILIZADAS

No decorrer do estágio, foram aplicados diversos conceitos aprendidos ao longo do curso de Ciência da Computação. Podemos citar, por exemplo, o paradigma de linguagem de programação orientado a objetos (OO) (WATT, 2004), dado que as linguagens de programação utilizadas durante o estágio se encaixavam, predominantemente, neste paradigma.

Outro ponto que foi de fundamental importância foi no âmbito dos padrões de projeto de software, com destaque para os padrões Façade e Model-View-Controller (MVC) (GAMMA et al., 1995), intensamente utilizados nos sistemas que foram evoluídos pelo estagiário.

Com relação à persistência de dados, utilizou-se alguns conceitos vistos na disciplina Banco de Dados, como realização e otimização de consultas, utilizando SQL. Como padrão de persistência de dados, foi utilizado o *Data Access Object* (DAO).

Alguns conceitos ainda não abordados na graduação do estagiário também foram utilizados, tais como o uso do protocolo Hypertext Transfer Protocol (HTTP) e aplicações Cliente-Servidor, no escopo de redes de computadores. Esses conteúdos, não vistos anteriormente, surgiram como aprendizado, pois obteve-se, além da experiência do estágio, conhecimento teórico, durante a sua execução.

Abaixo, encontra-se uma lista das principais tecnologias utilizadas no decorrer do estágio, com uma breve descrição de cada uma:

QUADRO 1 - TECNOLOGIAS UTILIZADAS

Tecnologia	Descrição	Mais Informações
Microsoft Windows 7	Sistema Operacional (SO) recente da Microsoft. Escolhido para ser usado durante o estágio, principalmente pela compatibilidade com os demais softwares utilizados.	<a href="http://www.microsoft.com/windows/windows-7/">http://www.microsoft.com/windows/windows-7/</a>
Linguagem de Programação Java	Linguagem padrão da plataforma Java, amplamente utilizada no decorrer do estágio.	<a href="http://www.oracle.com/technetwork/java/index.html">http://www.oracle.com/technetwork/java/index.html</a>
HyperText Markup Language (HTML)	Linguagem de marcação de hipertextos utilizada para produzir páginas na web. Amplamente utilizada no estágio, para o desenvolvimento do conteúdo das aplicações <i>web</i> a ser exibido no navegador.	<a href="http://www.w3.org/MarkUp/">http://www.w3.org/MarkUp/</a>
JavaScript	Linguagem de script criada para validar formulários no lado cliente, em aplicações cliente-servidor, e interagir com páginas dinâmicas. Também foi bastante utilizada	<a href="https://developer.mozilla.org/en/About_JavaScript">https://developer.mozilla.org/en/About_JavaScript</a>

	ao longo do estágio.	
Extensible Markup Language (XML)	Tem a função de gerar linguagens de marcação para necessidades especiais. Foi utilizada no projeto para organizar dados coletados sobre usuários e empresas de uma determinada rede social.	<a href="http://www.w3.org/XML">http://www.w3.org/XML</a>
JavaServer Pages (JSP)	Tecnologia utilizada para criar páginas HTML com a possibilidade de inserir código Java, adicionando um certo dinamismo às páginas produzidas. Bastante utilizado em uma das aplicações evoluídas durante o estágio, cujo código-fonte era baseado fortemente em JSP.	<a href="http://java.sun.com/products/jsp/">http://java.sun.com/products/jsp/</a>
Structured Query Language (SQL)	Linguagem de pesquisa declarativa para banco de dados. Utilizada para fazer consultas sobre a base de dados das aplicações desenvolvidas e/ou evoluídas no estágio.	<a href="http://www.acm.org/sigmod/record/issues/0403/index.html#standards">http://www.acm.org/sigmod/record/issues/0403/index.html#standards</a>
Apache Tomcat	Servidor web Java, sendo considerado uma espécie de <i>container</i> de servlets. Foi utilizado para a implantação das aplicações <i>web</i> desenvolvidas.	<a href="http://tomcat.apache.org">http://tomcat.apache.org</a>
MySQL	Um dos Sistemas Gerenciadores de Banco de Dados (SGBD) mais utilizados no mundo, mantido pela empresa Oracle e utiliza a linguagem SQL. Utilizado na manipulação do banco de dados de todas as aplicações desenvolvidas.	<a href="http://www.mysql.com/">http://www.mysql.com/</a>
MySQL Administrator	Ferramenta gráfica para auxiliar no gerenciamento de tabelas e consultas do MySQL. Utilizado no estágio para alteração de tabelas e realização de backup/restauração de esquemas.	<a href="http://dev.mysql.com/doc/administrator/pt/index.html">http://dev.mysql.com/doc/administrator/pt/index.html</a>
Skype	Software que provê comunicações pela internet, via conexões de voz. Utilizado para reuniões com o CEO da empresa e os desenvolvedores da cidade de Recife – PE.	<a href="http://www.skype.com/">http://www.skype.com/</a>
Interface de Programação de Aplicações (API) do LinkedIn	O LinkedIn é uma rede social voltada ao âmbito profissional. Este site provê uma API própria, utilizada no estágio para obter dados de empresas e usuários.	<a href="http://developer.linkedin.com/community/apis">http://developer.linkedin.com/community/apis</a>
JDOM	API de Java utilizada para converter elementos de XML para objetos Java. Utilizado para transformar dados obtidos do LinkedIn (em XML) para objetos Java.	<a href="http://www.jdom.org/">http://www.jdom.org/</a>
API Apache POI	API utilizada para escrever objetos Java em arquivos de formato XLS, que podem ser visualizados em softwares baseados em planilha, como o Microsoft Excel.	<a href="http://poi.apache.org/">http://poi.apache.org/</a>
Microsoft Excel	Software bastante conhecido na manipulação de planilhas. Foi utilizado para observar dados exportados para planilhas nas aplicações <i>web</i> , com o intuito de organizá-los em arquivos.	<a href="http://office.microsoft.com/pt-br/excel/">http://office.microsoft.com/pt-br/excel/</a>
jQuery	Biblioteca de JavaScript, utilizada para simplificar o lado cliente de aplicações <i>web</i> . Foi utilizada no desenvolvimento das aplicações do estágio, para fazer acesso direto à elementos HTML a partir do código JavaScript da aplicação.	<a href="http://jquery.com/">http://jquery.com/</a>
API Twitter4j	API utilizada para integrar aplicações Java com o Twitter, obtendo informações de perfis de usuários da	<a href="http://twitter4j.org/">http://twitter4j.org/</a>

	rede social.	
API Facebook Graph	API disponibilizada pela rede social Facebook, que permite que seus dados sejam exibidos na forma de objetos JavaScript (JSON's). Foi utilizada para comunicação das aplicações desenvolvidas com o Facebook.	<a href="http://developers.facebook.com/docs/api">http://developers.facebook.com/docs/api</a>
API Gson	API da Google que tem a função de fazer a conversão entre objetos Java e objetos JavaScript e vice-versa.	<a href="http://code.google.com/p/google-gson/">http://code.google.com/p/google-gson/</a>
API Yahoo! BOSS	Tem o intuito de prover os métodos necessários para um sistema de busca. Comunica uma aplicação web com o sistema de busca do Yahoo! Utilizado para desenvolver uma das aplicações do estágio.	<a href="http://developer.yahoo.com/search/booss/">http://developer.yahoo.com/search/booss/</a>
FileZilla	Cliente de Protocolo de Transferência de Arquivos (FTP) de código livre. Utilizado para subir as aplicações para o servidor.	<a href="http://filezilla-project.org/">http://filezilla-project.org/</a>
Eclipse Galileo	Ambiente de Desenvolvimento cujo projeto inicial foi desenvolvido na IBM, que mais tarde doou-o como software livre. Alguns plugins para o eclipse, com propósitos citados no início da sessão 4.1, também foram utilizados no decorrer do estágio.	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>
Apache Ant	Ferramenta criada para automatizar a construção de software. Utilizada para gerar arquivos e fazer a implantação (deploy) das aplicações de forma automática.	<a href="http://ant.apache.org/">http://ant.apache.org/</a>
Microsoft Word	Processador de Texto da Microsoft, foi utilizado para produzir o relatório de estágio e outros documentos da disciplina, como o plano de estágio.	<a href="http://office.microsoft.com/pt-br/word/default.aspx">http://office.microsoft.com/pt-br/word/default.aspx</a>



SEÇÃO IV

ATIVIDADES

REALIZADAS

## 4. ATIVIDADES REALIZADAS

Nesta seção, são descritas as atividades realizadas no decorrer do estágio. É importante comentar que houve pequenas mudanças em relação ao planejamento inicial, que pode ser encontrado na seção 8 do Apêndice A – Plano de Estágio. No Quadro 2, é apresentado um comparativo entre o tempo planejado e o tempo gasto em cada atividade.

QUADRO 2 - COMPARATIVO ENTRE OS TEMPOS ESTIMADOS E GASTOS DAS ATIVIDADES REALIZADAS

<b>ID</b>	<b>Atividade Realizada</b>	<b>Tempo Estimado</b>	<b>Tempo Gasto</b>
T1	Familiarização com o sistema	20h	20h
T2	Estudo de tecnologias e API's a serem usadas no sistema	20h	30h
T3	Produção de relatórios semanais	10h	10h
T4	Escrita do relatório final	40h	30h
T5	Desenvolvimento de sistemas de monitoramento de redes sociais	90h	130h
T6	Integração do subsistema desenvolvido	80h	40h
T7	Reuniões semanais com o supervisor técnico	25h	25h
T8	Reuniões com o supervisor acadêmico	15h	8h
T9	Reuniões com outros desenvolvedores da empresa	-	7h
Total		300h	300h

Ainda no plano de estágio, na Seção 5, é informado que o objetivo do estágio é trabalhar na evolução do sistema BuzzMonitor 2.0 e no desenvolvimento de seus subsistemas. Todavia, a política de relacionamento da empresa aproxima muito os funcionários. Isso fez com que ocorressem reuniões – não planejadas inicialmente – com outros desenvolvedores, que alocavam novas tarefas para o estagiário.

As tarefas alocadas ao estagiário, por parte dos outros desenvolvedores da empresa, comumente fugiam ao escopo do BM2. Este fator contribuiu para o destoamento entre a execução do estágio e seu planejamento. Sendo assim, pode ser considerado que o fluir do estágio se deu em uma maneira *ad hoc*.

As subseções a seguir possuem o propósito de detalhar cada atividade realizada pelo estagiário. A melhor forma encontrada para distribuí-las foi cronologicamente, para facilitar a compreensão e manter uma certa ordem.

#### 4.1 CONFIGURAÇÃO DO AMBIENTE E ENTENDIMENTO DO SISTEMA

A primeira atividade do estágio foi configurar o ambiente para executar o sistema BuzzMonitor 2.0 (BM2). Para isso, o estagiário instalou a IDE Eclipse Galileo. Na intenção de acelerar o desenvolvimento e auxiliar a codificação, foram instalados alguns *plugins* nesta IDE. Os *plugins* instalados proviam auxílio no desenvolvimento em JavaScript, Cascading Style Sheets(CSS), HTML, XML e JSP. Além destes, foi instalado um *plugin* com o objetivo de executar o Apache Tomcat pela própria IDE.

Uma ferramenta gráfica de suporte ao SGBD MySQL, utilizado para persistências de dados no BM2, também foi instalada. O MySQL Administrator permite a edição de tabelas, consultas no banco e *backup* de esquemas de forma simples e intuitiva.

Depois de configurado o ambiente, o BM2 foi utilizado pelo estagiário, com paralela análise do código. Durante esse período, os desenvolvedores da cidade de Recife – PE, por possuírem mais experiência com o código e com a aplicação, eram frequentemente consultados, via Skype.

O BM2 consiste em uma plataforma que agrega alguns subsistemas. Sua interface consiste em um conjunto de abas, cada qual representando um subsistema. Cada subsistema possui um conjunto de *widgets*, as quais apresentam

os dados coletados. Na Figura 1, é apresentada uma visão do BM2 em funcionamento.



FIGURA 1 - BUZZMONITOR 2.0.

Como o objetivo do estágio era, a princípio, evoluir o BM2 e o mesmo baseava-se na disponibilização de *widgets* nos subsistemas, o estagiário começou a desenvolver um *widget* – o Top URLs – dentro do subsistema Facebook Meter. Este *widget* tinha a função de expor as 10 URLs mais comentadas por usuários do facebook, bem como mostrar a quantidade de comentários sobre aquelas URLs.

O propósito desta primeira atividade de desenvolvimento foi, apenas, se ambientar com o desenvolvimento de *widgets*. O Facebook Meter não foi estudado profundamente e o Top URLs foi, mais tardiamente, modificado por outro estagiário, de forma que expusesse apenas URLs relacionadas com a marca cliente.

O desenvolvimento de *widgets*, dentro da E.life, segue um padrão. Este padrão pode ser encontrado dentro de um documento de propriedade da empresa. Tal documento possui as diretrizes para montar os *widgets*. O Top URLs, na sua primeira versão, pode ser visualizado na Figura 2.

## 4.2 LEVANTAMENTO DE EMPRESAS NO LINKEDIN

Depois de adquirir o conhecimento para a criação de *widgets* no BuzzMonitor 2.0, houve a primeira mudança no escopo do estágio, através de uma atividade solicitada pelo CEO e supervisor técnico Alessandro Barbosa.





FIGURA 2 - WIDGET TOP URLS.

Nesse contexto, a atividade alocada ao estagiário foi realizar uma coleta de dados sobre as empresas, bem como de seus empregados, e gerar uma planilha atualizada, a ser usada pelos funcionários do setor de marketing da empresa.

Para iniciar essa atividade, foi necessário um estudo acerca da API do LinkedIn, que é disponibilizada online. O LinkedIn limita o número de requisições feitas diariamente, portanto procurou-se obter o máximo de conhecimento da API possível, a fim de se evitar testes desnecessários. Além disso, algumas informações dos usuários podem ser omitidas, o que causava uma baixa qualidade na coleta dos dados. Para contornar isso, usou-se a conta paga do CEO da empresa. Contas pagas podem visualizar o conteúdo do site numa amplitude maior, o que garantiu a boa formação dos dados.

A princípio, foi solicitada uma chave, chamada de API Key, para ser utilizada pela aplicação desenvolvida – que fazia a coleta dos dados. A aplicação fazia, então, uma requisição ao LinkedIn, passando a API Key obtida pelo usuário.

Diferentemente do padrão OAuth, descrito posteriormente na seção 4.6, a resposta do LinkedIn continha não só um *token* de requisição, mas também uma URL. A cada requisição feita, era necessário uma visita à esta URL, que informava um número verificador.

Em posse do verificador e do *token* de requisição, a aplicação estava apta a realizar o pedido do *token* de acesso e assim o fazia. Este *token* de acesso, obtido

como resposta do LinkedIn, permitia a chamada direta à API, para a obtenção dos dados procurados. Na Figura 3, é apresentado o diagrama de sequência UML (MILES; HAMILTON, 2006) que descreve o fluxo do processo descrito.

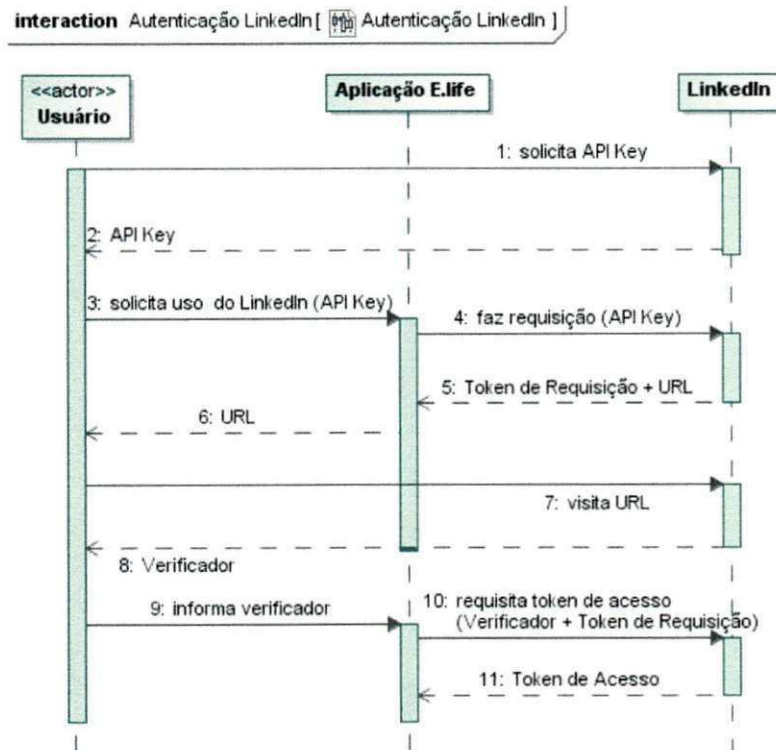


FIGURA 3 - FLUXO DE CONTROLE DA AUTENTICAÇÃO DO LINKEDIN.

O script desenvolvido apresentou uma grande flexibilidade, inclusive, para futuros levantamentos desse tipo. Para fazer o levantamento sobre uma empresa, basta substituir o nome padrão, no código, e executar o programa.

Os dados coletados sobre as empresas solicitadas eram saivos no formato XML, pois era a forma de resposta do LinkedIn. Para a transformação dos dados de XML para objetos Java, utilizou-se a API JDOM. Depois de convertidos para objetos Java, os dados foram escritos em um arquivo do formato XLS, utilizando a API Apache POI. Por questões de organização, foi utilizado o Microsoft Excel para um acabamento final nos resultados. Na Figura 4, ilustra-se como os dados foram entregues à equipe de marketing.

A entrega da planilha foi mediada por um dos desenvolvedores de Recife – PE, que introduziu o estagiário aos funcionários do setor de marketing da empresa, mostrando que não há uma fronteira muito grande entre os diversos setores da

empresa, não havendo interação apenas entre os membros de um determinado setor.

UNIVERSIDADE MACKENZIE								
NAME	LAST NAME	HEADLINE	LOCATION	SECTOR	CURRENT POSITION	COMPANY	YEAR	MONTH
1401	Almeida	Associação Marketing and After Sales Consultant at IPC Development Associaç	São Paulo, São Paulo, Brazil	Management Consulting	Professor	Universidade Mackenzie	2009	5
1402	David	Head of Digital Channel at Expansão de Negócios	São Paulo, São Paulo, Brazil	Digital Media	Head of Digital Channel	Expansão de Negócios	2010	7
1403	Carvalho	Director of Finance, professional services provider	São Paulo, São Paulo, Brazil	Professional Services	Professor MBA	Universidade de São Paulo	2009	5
1404	Bluh	Professor Assistente at Universidade Presbiteriana Mackenzie and Research C	São Paulo, São Paulo, Brazil	Research	Professor Adjunto	Universidade Mackenzie	2009	5
1405	ROBERTO	Sales Channel Manager at CIPRILO de São Paulo and Associated Professors at Univer	São Paulo, São Paulo, Brazil	Computer Hardware	Quem de Deus - São Ti	EPICOM de São Paulo	2010	7
1406	Wassan	Professor at Universidade Mackenzie and Higher Education Consultant	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	Não informado	?
1407	Paulo Roberto	CONSULTING SERVICES MANAGER, SENIOR FINANCIAL SERVICES CONSULTANT	São Paulo, São Paulo, Brazil	Management Consulting	Executive	Henry Consultoria Turística Ltda	2008	10
1408	Paulo	Senior Business Professional	São Paulo, São Paulo, Brazil	Business Services	Sen. de Consultoria	Universidade Mackenzie	2009	10
1409	Paulo Enzo	Analista de Universidade Mackenzie	São Paulo, São Paulo, Brazil	Education	Analista	Universidade Mackenzie	Não informado	?
1410	Marysaura	Senior Business Professional	São Paulo, São Paulo, Brazil	Business Services	Professor	Universidade Mackenzie	2009	10
1411	Marta	Senior Lecturer	São Paulo, São Paulo, Brazil	Education Management	Professor	Universidade Mackenzie	Não informado	?
1412	Thales Nelson	Manager Enterprise Affairs	São Paulo, São Paulo, Brazil	Professional Training & Consulting	Post-Graduation Professor	Universidade Mackenzie	2009	10
1413	Bluh	Professor - Psicopedagogia at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Education Management	Professor - Psicopedagogia	Universidade Mackenzie	2009	7
1414	Luciano	Professor at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Education Management	Professor	Universidade Mackenzie	2009	7
1415	Julia Virginia	Professor at Universidade Mackenzie and Civil Engineering Consultant	São Paulo, São Paulo, Brazil	Civil Engineering	Professor	Universidade Mackenzie	Não informado	?
1416	Paulo	Director at BEMCO and Ship Fabricator	São Paulo, São Paulo, Brazil	Manufacturing	Vice-President	Parsons Corp. Universidade Mackenzie	2010	7
1417	Marcelo José	Professor at Universidade Mackenzie and Higher Education Consultant	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	Não informado	?
1418	Diego Rogério	Associate at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Associate	Universidade Mackenzie	Não informado	?
1419	Roberto	Professor at Universidade Mackenzie and Student at Universidade de São Paulo	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	1994	7
1420	Paulo	Analista de Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	Não informado	?
1421	Renzo	Professor at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor / Coordinator	Universidade Mackenzie	Não informado	?
1422	Roberto	Professor at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	2009	7
1423	Diego	Education Management Consultant and Professional	São Paulo, São Paulo, Brazil	Education Management	Professor / Coordinator	Universidade Mackenzie	2009	7
1424	Paulo	Professor at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	2009	7
1425	Paulo	Professor at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	2009	7
1426	Almeida	Professor at Universidade Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	Não informado	?
1427	Thales	Professor at Universidade Presbiteriana Mackenzie	São Paulo, São Paulo, Brazil	Higher Education	Professor	Universidade Mackenzie	2009	7

VARIG LINHAS AÉREAS								
NAME	LAST NAME	HEADLINE	LOCATION	SECTOR	CURRENT POSITION	COMPANY	YEAR	MONTH
1428	Diego	IF Projeto Financeiro Analista at Volkswagen do Brasil	São Paulo, São Paulo, Brazil	Automotive	IF Projeto Financeiro Adm/Financeira	Volkswagen	Não informado	?
1429	Paulo	Coord. Qualificação Comercial at Volkswagen	São Paulo, São Paulo, Brazil	Automotive	Coord. Qualificação Comercial - São Paulo	Volkswagen	2009	7
1430	Paulo	Analista de vendas at Volkswagen	São Paulo, São Paulo, Brazil	Automotive	Analista de vendas	Volkswagen	Não informado	?
1431	Paulo	Vendedor at Volkswagen Financial Services	São Paulo, São Paulo, Brazil	Financial Services	Elenco Volkswagen	Volkswagen Serviços Financeiros	Não informado	?

VOLKSWAGEN								
NAME	LAST NAME	HEADLINE	LOCATION	SECTOR	CURRENT POSITION	COMPANY	YEAR	MONTH
1432	Diego	IF Projeto Financeiro Analista at Volkswagen do Brasil	São Paulo, São Paulo, Brazil	Automotive	IF Projeto Financeiro Adm/Financeira	Volkswagen	Não informado	?
1433	Paulo	Coord. Qualificação Comercial at Volkswagen	São Paulo, São Paulo, Brazil	Automotive	Coord. Qualificação Comercial - São Paulo	Volkswagen	2009	7
1434	Paulo	Analista de vendas at Volkswagen	São Paulo, São Paulo, Brazil	Automotive	Analista de vendas	Volkswagen	Não informado	?
1435	Paulo	Vendedor at Volkswagen Financial Services	São Paulo, São Paulo, Brazil	Financial Services	Elenco Volkswagen	Volkswagen Serviços Financeiros	Não informado	?

FIGURA 4 - PLANILHA COM DADOS DO LINKEDIN.

### 4.3 SOCIAL CRM

Um dos subsistemas do BuzzMonitor 2.0 é o Social CRM. Este subsistema tem o intuito de agrupar tudo relacionado à interação entre a empresa e seu consumidor, como o diálogo entre eles, publicações da empresa em redes sociais, etc. Durante o estágio, focou-se na evolução de um *widget* do Social CRM: o SAC Questions. Este *widget* tem a função de prover o diálogo entre empresa e consumidor, utilizando as funcionalidades da aplicação E.life SAC, ilustrada na figura 5 e brevemente descrita logo em seguida.

O E.life SAC é um sistema criado para as empresas clientes da E.life utilizarem um canal de comunicação com seus próprios clientes. Os clientes fazem as perguntas através de aplicativos, existentes nas redes sociais Orkut e Facebook. Essas perguntas são enviadas para o banco de dados da E.life.

As perguntas são respondidas por funcionários da empresa cliente, encarregados de fazer o contato com os clientes. Quando o funcionário se autentica no E.life SAC, ele é redirecionado para uma página, na qual se encontram as perguntas realizadas pelos seus clientes. A aplicação também se comunica com o Banco de Dados, buscando as perguntas e armazenando as respostas.

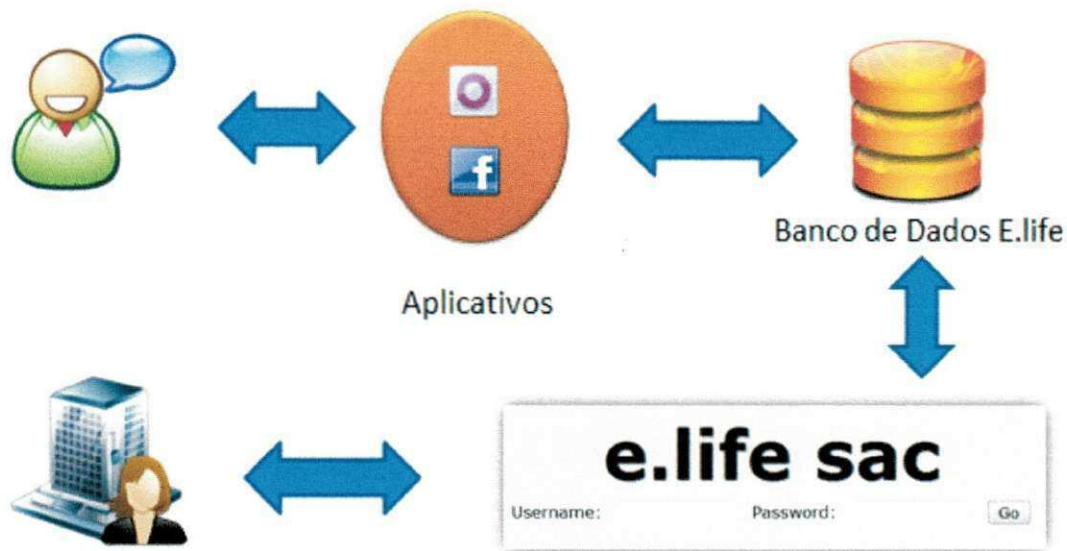


FIGURA 5 - FUNCIONAMENTO DO E.LIFE SAC.

Os clientes que realizaram as perguntas podem visualizar as respectivas respostas através dos aplicativos do Orkut e do Facebook. Quando iniciado, o aplicativo busca as respostas respectivas às perguntas do usuário na base de dados da E.life e as exibe na tela.

Como foi falado, o Social CRM foi apenas a adaptação do E.life SAC para um *widget* do BuzzMonitor. As atividades realizadas neste contexto incluem o desenvolvimento de um painel deslizante, no qual se exibe as informações do autor da pergunta, adição de novos filtros, mudanças na interface e exportação de resultados.

Ao responder uma pergunta, a empresa só possuía a informação do nome do seu autor. Entretanto, a base de dados continha algumas informações também relevantes, tais como email, localidade e sexo. Na intenção de prover estes dados para a empresa cliente, foi criado um painel com as informações do usuário. Este painel, bem como uma imagem do SAC Questions, pode ser visto na Figura 6.

Os analistas da E.life acharam que era importante para a empresa ter um controle sobre perguntas feitas em determinado intervalo de tempo. Isso gerou uma necessidade de um filtro de perguntas por data. Sendo assim, foi feito um filtro que exibe apenas perguntas realizadas em um certo período de tempo. Na Figura 7, é ilustrado seu funcionamento.

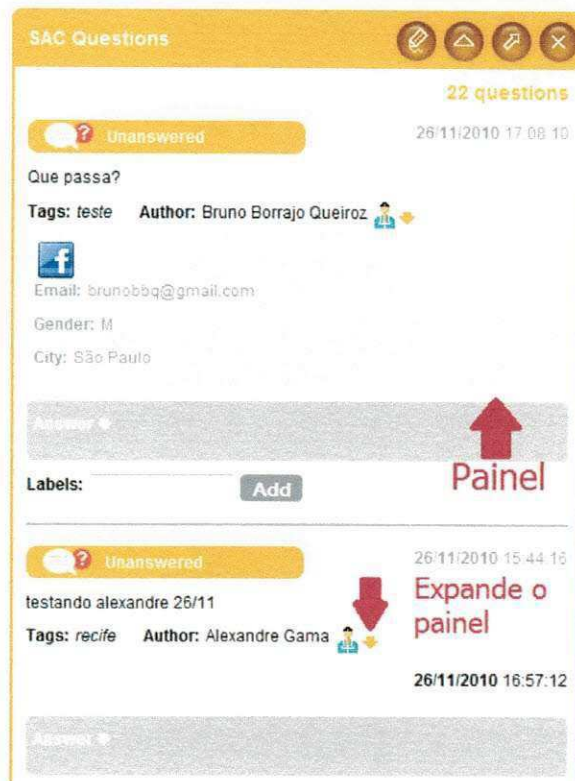


FIGURA 6 - PAINEL DO USUARIO NO SACQUESTIONS.



FIGURA 7 - FILTRO POR DATA NO SACQUESTIONS.

Os dados das perguntas e respostas eram gravados no banco de dados da E.life, mas só eram exibidos para os analistas da empresa através da interface da *widget*. Foi desenvolvida, então, uma funcionalidade de exportar planilhas no formato XLS contendo todos os dados das questões, como autor, resposta e data.

Frequentemente, o CEO ou o CTO da empresa ficavam insatisfeitos com algum elemento da interface. Sendo assim, o estagiário precisou, constantemente, alterar os elementos da mesma. Para isso, foi utilizado código HTML e JavaScript mesclados. A biblioteca de JavaScript JQuery foi extremamente útil durante esse processo.

#### 4.4 E.LIFE PEOPLE SEARCH E SERVLETS

Durante uma reunião, via Skype, com o CTO da empresa – Jairson Vitorino – e outros dois desenvolvedores da E.life, do núcleo de Recife – PE, decidiu-se que a empresa cliente poderia querer obter mais informações sobre os seus consumidores. Dessa forma, houve a gênese do E.life People Search, um sistema que busca redes sociais de um indivíduo, a partir do seu nome e/ou username.

A idéia inicial era integrar o E.life People Search ao SAC Questions, para obter outras redes sociais dos autores das perguntas, o que podia ser útil para a empresa, que iria ter mais opções de caminho para responder os autores das perguntas, bem como poderia acompanhá-los por outros meios.

Antes de iniciar o desenvolvimento, foi realizada uma pesquisa em sistemas *web*, para saber se era possível reaproveitar alguma ferramenta de código aberto ou fazer uso de alguma API, para fazer as buscas. Assim, foram realizados muitos testes em ferramentas do gênero, como KGB People, 123People e Wink.

Algumas das ferramentas quase não exibiam resultados. Outras encontravam algumas redes sociais do usuário, mas cobravam, para utilização de sua API. Algumas ferramentas, ainda, sequer forneciam acesso à API. Os resultados dos testes e da pesquisa deixaram claro que a melhor solução seria mesmo desenvolver uma aplicação do zero.

As vantagens de se desenvolver a própria aplicação estavam claras. Primeiro, o código seria de propriedade E.life, podendo ser utilizado em outros sistemas da

empresa, quando fosse necessário. Além disso, o E.life People Search não buscaria apenas a URL das redes sociais. Ele tentaria obter todos os dados possíveis do usuário na rede pesquisada.

Os conceitos de UML(MILES; HAMILTON, 2006) adquiridos na disciplina Sistemas de Informação II foram fundamentais para a modelagem da aplicação. Entretanto, era sabido que, para a implementação do sistema – um webservice – seria necessário a noção do conceito de servlets.

Inicialmente, pensou-se em fazer um pacote, o qual realizava as pesquisas, de fato, e um que continha o servlet, cuja implementação ainda ia ser estudada, que retornava o objeto a ser pesquisado. O Diagrama de Classes final do E.life People Search encontra-se ilustrado na Figura 8.

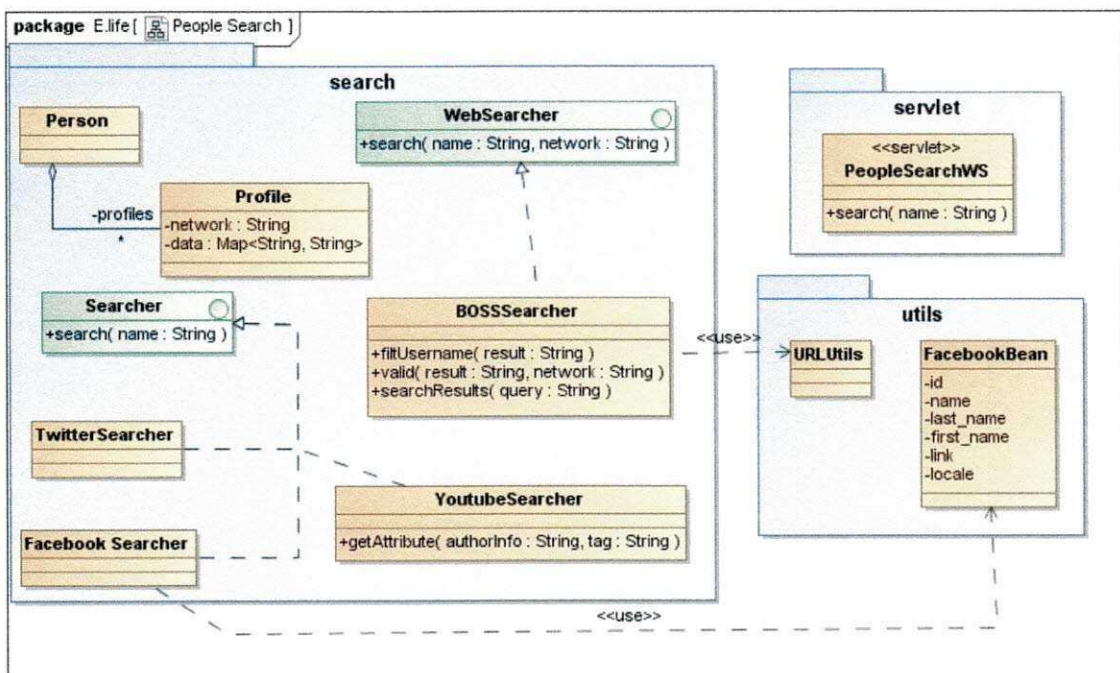


FIGURA 8 - DIAGRAMA DE CLASSES DO E.LIFE PEOPLE SEARCH.

O sistema recebe um atributo, que é o nome a ser pesquisado na rede. Este nome é, então, pesquisado em três redes sociais diferentes – Youtube, Facebook e Twitter. Cada pesquisa se comporta de forma diferente e é realizada em uma thread separada.

Para a busca no Twitter, utilizou-se a API twitter4j. Depois de verificar se existe o username pesquisado no Twitter, o E.life People Search usa o twitter4j para

coletar dados sobre o mesmo. Atualmente, os dados obtidos incluem nome do usuário, URL, número de seguidores, número de seguidos, foto e localidade.

No caso da busca do Youtube, foi necessário fazer uma verificação no HTML puro da página encontrada. O Youtube usa uma marcação específica quando um usuário não é encontrado. Bastou então realizar uma checagem. Se o conteúdo da página incluir essa marcação, a busca não foi bem sucedida. Caso contrário, os dados são pesquisados.

Como não existia API para o Youtube, foi preciso criar um *parser* de HTML para Java, para obter os dados que atualmente são coletados. Os dados obtidos no perfil dos usuários no Youtube são URL, idade, visualizações do canal, país, número de inscritos no seu canal, total de material enviado, entre outros.

Já no Facebook, foi utilizada uma abordagem diferente. O Facebook mantém uma API, chamada Graph. Esta API permite o acesso a recursos do Facebook via código.

O E.life People Search utilizou o recurso da visualização do perfil dos usuários na forma de JavaScript Object Notation (JSON). Como a aplicação estava codificada em Java, o resultado na forma de objeto JavaScript não era tão interessante. Sendo assim, foi criada uma classe especialmente para tratar isso.

A classe FacebookBean tem o intuito de serializar JSON's em objetos Java. Isto é feito através do uso da API Gson, da Google. A classe possui apenas atributos e métodos de acesso aos mesmos, seguindo o padrão JavaBeans. Dessa maneira, a API Gson transforma o perfil obtido em JSON para um objeto Java.

Nos três casos acima, caso não seja encontrado o perfil do usuário, é feita uma pesquisa no Yahoo, através de sua API BOSS (Build your Own Search Service), na rede social correspondente. A classe BOSSSearcher é responsável por esta pesquisa.

Depois de obter os dados de cada rede, é criado um objeto do tipo Profile, contendo o nome da mesma como atributo, bem como os dados correspondente ao perfil do usuário na rede social.

Depois de obter os três objetos do tipo Profile, é criado um objeto Person, que será retornado pela aplicação. Ainda não estava claro a forma com que este objeto



seria retornado. Era necessário um conhecimento de servlets, para entender como seria o funcionamento do E.life People Search.

O estudo de servlets se deu através da apostila do curso da Caelum: Java para Desenvolvimento Web. Após o estudo e a realização de vários testes simples, ficou um pouco mais claro como as requisições e respostas iriam funcionar no E.life People Search. Além disso, obteve-se o conhecimento de como realizar o mapeamento de servlets no arquivo web.xml do projeto, envio e captura de parâmetros, entre outros conceitos.

Dentro do pacote servlet, criou-se a classe PeopleSearchWS. Essa classe é um servlet e é responsável por receber a requisição da pesquisa do usuário nas redes sociais. Essa requisição envia um parâmetro, contendo o nome ou apelido do usuário a ser pesquisado. O servlet pega este parâmetro e chama os métodos de busca da aplicação. Quando encontra, envia o resultado encontrado na resposta, no formato JSON.

A aplicação que tiver de utilizar o E.life People Search fará a chamada ao servlet, passando o nome a ser pesquisado como parâmetro e receberá um objeto do tipo String (se a aplicação for em Java) ou um JSON (caso esteja sendo utilizado JavaScript). Em seguida, a aplicação pode manipular os dados obtidos como bem entender.

Depois da implementação do E.life People Search, visou-se sua integração com o SAC Questions. Sendo assim, em uma discussão bastante democrática com o CTO da empresa e alguns desenvolvedores, decidiu-se que os resultados seriam exibidos no painel que continha as informações dos autores das perguntas, juntamente com as informações que lá existiam.

Após a configuração e mapeamento do servlet dentro do escopo do BuzzMonitor 2.0, iniciou-se a integração com a interface. A integração foi realizada em um tempo inferior ao esperado.

Quando a integração foi realizada, foi pedido que alguns funcionários de diversos setores da E.life utilizassem e testasse o SAC Questions para analisar o funcionamento do E.life People Search no mesmo. Isto porque as aplicações da E.life não são utilizadas/analizadas apenas pelos desenvolvedores de software.

Depois da avaliação, chegou-se à conclusão que uma grande desvantagem da ferramenta era a falta de acurácia. Não se pode garantir que a rede social encontrada, ao se pesquisar por um autor, pertence realmente a ele. Além disso, nem sempre a rede social é encontrada.

Como vantagem da aplicação, existe o fato de que a velocidade da obtenção dos resultados parece ser maior do que as ferramentas do gênero existentes, além da quantidade maior de informação garimpada na pesquisa.

Na Figura 9, é apresentada a integração do E.life People Search ao *widget* Social CRM. Nota-se que foi encontrado o Twitter do autor da pergunta, neste caso.

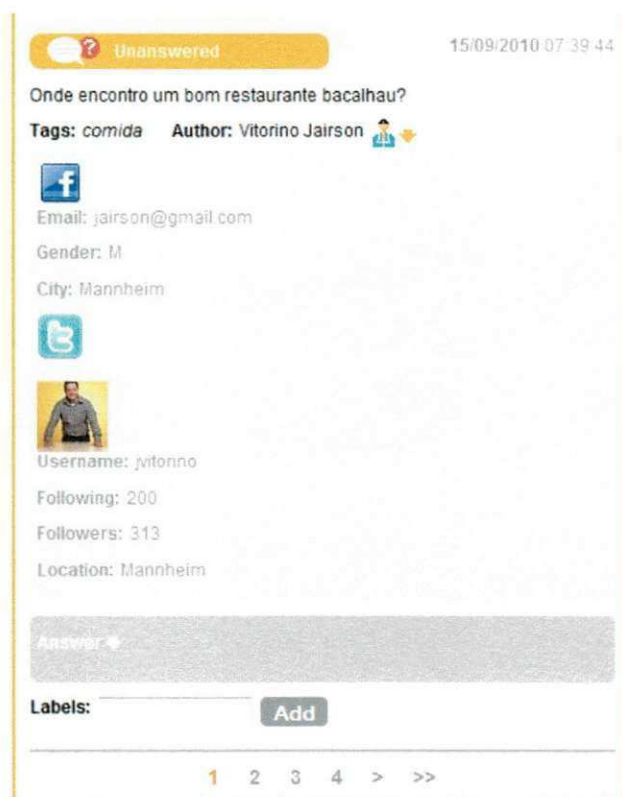


FIGURA 9 - INTEGRAÇÃO DE SAC QUESTIONS E E.LIFE PEOPLE SEARCH.

## 4.5 E.LIFE SAC E JSP

O foco da E.life, atualmente, é o BuzzMonitor 2.0. Sendo assim, novos filtros, mudanças na interface e integrações com aplicações menores, como o E.life People Search, estavam sendo feitos apenas no *widget* do SAC Questions. Isso fez com que a aplicação original do E.life SAC fosse deixada de lado.

A intenção da E.life é que, no futuro, as empresas clientes que utilizam subsistemas do BM2 passem a utilizar apenas a plataforma do BM2, deixando as aplicações independentes de lado. No entanto, empresas como a Whiskas – empresa que produz rações para gatos – ainda utilizam a interface original do E.life SAC, o que causou uma necessidade na implementação de filtros existentes no *widget* do SAC Questions que ainda não estavam disponíveis no E.life Sac.

Antes de começar a implementação, imaginou-se que seria um trabalho bem pequeno implementar os filtros no E.life SAC, já que os mesmos já haviam sido implementados no SAC Questions. Porém, o código do E.life SAC está todo baseado em JSP, diferentemente do SAC Questions, que é implementado em JavaScript puro, seguindo o padrão do BM2.

Sendo assim, fez-se necessário o estudo de JSP. Para realização do estudo, contou-se com o suporte do mesmo livro utilizado para o estudo de servlets: o exemplar da Caelum – Java para Desenvolvimento Web.

Servlets nos dão a opção de desenvolver um código dinâmico. Entretanto, criar servlets sempre que quisermos integrar código HTML com Java acarretaria num grande trabalho para manutenção das páginas do projeto web. Assim, uma tecnologia que permita usar código HTML de forma direta, mas que também possibilite a integração de código Java seria ideal.

Essa é a característica principal do JSP. Esta tecnologia permite que se crie uma página com código HTML. No entanto, permite a integração direta de código Java puro. Assim, é possível integrar comportamento dinâmico para as páginas HTML, utilizando comandos condicionais, comandos iterativos, declaração de variáveis, entre outros.

Apesar do tempo gasto no estudo de JSP, a adaptação com o código do E.life SAC se deu mais rápida do que a adaptação com o código do BM2, escrito em Java/JavaScript.

A interface do E.life SAC é bem mais simples do que o BuzzMonitor 2.0. Durante a implementação dos filtros, o estagiário também precisou editar os arquivos de estilo (CSS), para modificar a maneira como os campos eram expostos na tela. Na Figura 10, pode ser vista a interface do E.life SAC.

Query: \_\_\_\_\_

Status: Any

22 results

Question:  
Que passa?  
Tags: teste  
Author: undefined  
[author details](#)

26/11/2010 17:08:10

Answer:

[answer](#)

Question:  
testando alexandre 26/11

Tags: recife  
Author: undefined

26/11/2010 15:44:16

[author details](#)

Answer:

[answer](#)

26/11/2010 16:57:12

Question:

FIGURA 10 - INTERFACE ORIGINAL DO E.LIFE SAC.

A primeira funcionalidade implementada foi a adição e remoção de labels para as perguntas. Assim, cada pergunta poderia ter uma ou mais labels para um possível agrupamento de perguntas. Na Figura 11, encontra-se uma imagem da aplicação, depois da implementação desta funcionalidade.

Question:  
a testar  
Labels: [lab3] [label 2]  
Author: Rui Boaventura  
[author details](#)  
Answer:  
[answer](#)

20/10/2010 11:09:19

FIGURA 11 - ADIÇÃO E REMOÇÃO DE LABELS NO E.LIFE SAC.

Em seguida, foi implementado o filtro por labels. Dessa maneira, o usuário poderá selecionar apenas questões que possuem determinada label. As labels são armazenadas na base de dados e são recuperadas e salvas junto com as perguntas respectivas.

O filtro que foi implementado, em sequência, foi o de datas. A implementação se deu de forma análoga ao que foi feito no *widget* do SAC Questions. Houve modificação no servlet, acrescentando a passagem do parâmetro que representa o período que limita as questões a serem exibidas. A classe controladora e DAO, que faz a comunicação com o Banco de Dados, não foram modificadas, pois eram as mesmas utilizadas pelo SAC Questions.

Depois da implementação dos filtros, houve necessidade de uma pequena mudança no estilo da interface, para que os campos de preenchimento dos filtros fossem dispostos na tela. A versão final da interface do E.life SAC pode ser vista na Figura 12.

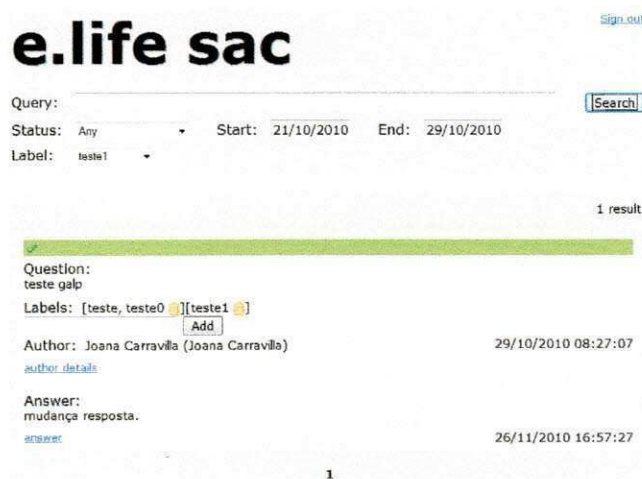


FIGURA 12 - FILTROS DO E.LIFE SAC.

## 4.6 CONEXÃO COM O FACEBOOK E OAUTH

Alguns projetos da E.life estavam com uma situação em comum. Por motivos diversos, as aplicações deveriam se comunicar com o Facebook, afim de obter dados ou fazer publicações. Sendo assim, buscou-se uma forma de se realizar uma autenticação no Facebook através das aplicações da E.life. Sendo assim, iniciou-se um vasto estudo na API Graph, do Facebook.

Já no início do estudo, percebeu-se que era preciso uma certa fundamentação teórica acerca do padrão de autenticação OAuth, utilizado para autenticação no Facebook.

O OAuth permite que o usuário acesse recursos privados de uma aplicação (neste caso, o Facebook) através de uma aplicação externa (que seriam as aplicações da E.life) sem a necessidade de compartilhar sua senha.

O fluxo seguido pelo OAuth se dá da maneira que segue:

- A partir da aplicação externa, o usuário clica em um link para utilizar a aplicação-alvo e recebe um **token de requisição**.
- Em seguida, há um redirecionamento para uma tela de login na aplicação alvo e o usuário entra com seus dados.
- Depois, o usuário é perguntado se permite que a aplicação externa se comunique com a aplicação alvo. Ao permitir, o servidor marca o *token* de requisição como autorizado e redireciona o usuário para a aplicação externa.
- Ao retornar para a aplicação externa, há uma troca entre o *token* de requisição pelo **token de acesso**. Este *token* tem como finalidade acessar os recursos privados da aplicação.
- Com o token de acesso, o usuário pode fazer chamadas diretas à aplicação alvo e utilizar seus recursos privados.

O Facebook utiliza o OAuth 2.0, o que permite a conexão com servlets Java. Além desta vantagem, o OAuth 2.0 permite autenticação sem criptografia, utilizando HTTPS.

O primeiro passo para a autenticação com o Facebook foi criar uma aplicação. Uma aplicação criada no Facebook recebe três informações: *Application ID*, *API Key* e *Application Secret*. Estes dados são utilizados ao fazer as requisições para a Graph API.

Em seguida, foi criado um botão de autenticação com o Facebook. Esse botão faz uma requisição pelo token de acesso e redireciona o usuário para uma tela de login com o Facebook.

Os testes foram realizados no *widget* SAC Questions, do BuzzMonitor 2.0. Na Figura 13, apresenta-se o botão citado acima, na aplicação.

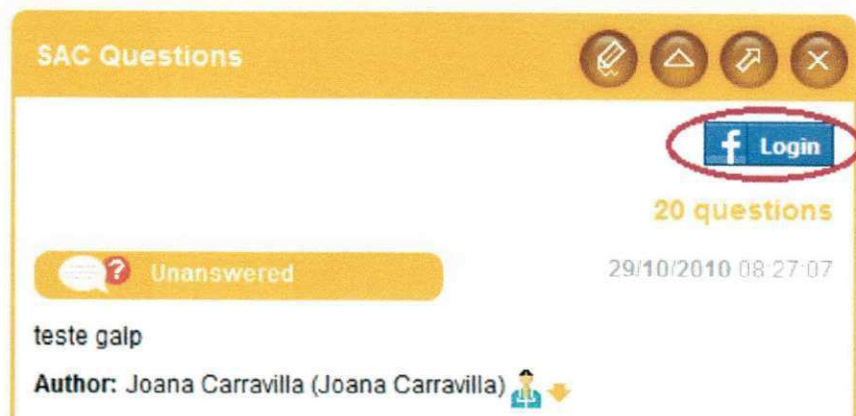


FIGURA 13 - LOGIN COM FACEBOOK.

Após o login no Facebook, uma mensagem, perguntando se o usuário permite que a aplicação E.life SAC tenha acesso aos seus dados do facebook, é exibida. Caso o usuário permita, a aplicação chama um servlet, previamente especificado na chamada da requisição do *token* de acesso. Na Figura 14, é ilustrada a situação na qual o usuário é indagado se permite que a aplicação tenha controle sobre parte dos seus dados.

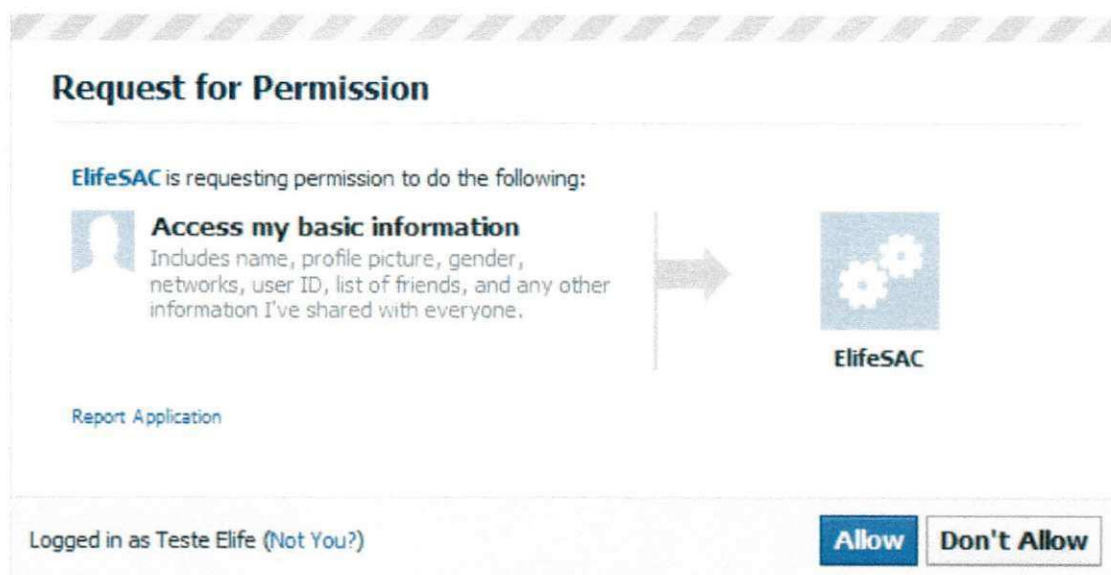


FIGURA 14 - REQUISIÇÃO DE PERMISSÃO DO ELIFESAC PARA USAR OS DADOS DO FACEBOOK.

Neste servlet, é capturado um parâmetro *code*. Este parâmetro é uma espécie de código de segurança e deve ser trocado pelo *token* de acesso. Essa troca foi feita através do próprio servlet, que faz uma nova chamada à API, fornecendo o código de segurança e recebendo o *token* de acesso como resposta. Depois da obtenção do *token* de acesso, o servlet redireciona o fluxo da aplicação para a aplicação original (no caso dos testes, para o *widget* do SAC Questions).

Depois de obtido, o *token* de acesso é gravado na sessão. Dessa maneira, ao iniciar o SAC Questions, uma checagem na sessão é realizada. Caso a sessão contenha o *token* de acesso, o botão de login com o facebook não aparece mais na tela.

Em posse do *token* de acesso, as chamadas na Graph API podem ser realizadas de forma direta. No caso do SAC Questions, estas chamadas envolverão funcionalidades de postar na wall (mural) do usuário do Facebook. No entanto, esta funcionalidade não se encontra no escopo do estágio realizado.

#### 4.7 VALIDAÇÃO

As aplicações desenvolvidas e/ou evoluídas eram utilizadas por outros desenvolvedores da empresa, além de outros funcionários de outros setores de E.life. Dessa forma, elas passavam por um certo período de valiação, constituído de testes funcionais. Nesse período, eram solicitadas mudanças nas aplicações, seja de interface ou de funcionalidade.



SEÇÃO V

CONSIDERAÇÕES

FINAIS

## 5. CONSIDERAÇÕES FINAIS

O estágio realizado foi de fundamental importância para a formação profissional do estagiário, que foi capaz de vivenciar um ambiente real do mercado de trabalho. Os sistemas desenvolvidos e evoluídos para a E.life eram disponibilizados para os clientes e para funcionários da empresa. Estes, por sua vez, davam o *feedback* sobre a aplicação, indicando mudanças – em elementos da interface ou em alguma funcionalidade – ou apenas assegurando o sucesso do desenvolvimento.

É importante destacar o amadurecimento do aluno acerca de habilidades corriqueiras em processos de desenvolvimento de software, tais como mudanças de requisitos e superação de novos riscos detectados. Estas lições não foram apenas para o estágio, mas sim para toda a vida profissional.

No escopo acadêmico, foi possível aplicar conceitos vistos ao longo da graduação. Vale ressaltar que alguns desses conceitos tiveram que ser aprofundados ao longo do estágio, para seu melhor uso. Além disso, conceitos completamente novos para o estagiário foram abordados, permitindo uma boa base para quando o mesmo for cursar as disciplinas respectivas. Finalmente, o estágio se mostrou de grande valia, não só profissionalmente, mas também em termos acadêmicos.

Um ponto negativo do estágio, foi o problema encontrado ao se realizá-lo na modalidade *home-office*. A família, e em alguns momentos o próprio estagiário, não soube distinguir o ambiente familiar do ambiente de trabalho, o que ocasionou uma leve perda de desempenho em alguns pontos do estágio.

Seria de bom grado a oferta de uma disciplina optativa focada no desenvolvimento de sistemas web. Com a vivência do estágio, o estagiário percebeu que há uma ampla demanda no mercado para desenvolvedores *web*. Caso exista uma cadeira com esse escopo, seria interessante que ela fosse mais ofertada, pois certamente existiria uma grande demanda de alunos para cursá-la.



REFERÊNCIAS

BIBLIOGRÁFICAS

## REFERÊNCIAS BIBLIOGRÁFICAS

- CAELUM. **Java para Desenvolvimento Web**. Disponível em <<http://www.caelum.com.br/curso/fj-21-java-web/>>. Último acesso em: 30 de nov. 2010.
- DJMAL, T. **O que é o Talmud?** Morasha. Disponível em: <<http://www.morasha.com.br/edicoes/ed43/talmud.asp>>. Último acesso em: 22 de nov. 2010.
- E.LIFE. **E.life BuzzMonitor**. Disponível em: <<http://elife.com.br/tecnologia/>>. Último acesso em: 30 de nov. 2010.
- FACEBOOK. **Graph API Reference**. Disponível em: <<http://developers.facebook.com/docs/reference/api/>>. Último acesso em: 30 nov. 2010.
- GAMMA, E.; HELM, R.; JOHNSON, R.; Vlissides, J. **Design patterns: elements of reusable object-oriented software**. Addison-Wesley, Reading, MA, 1995.
- JDOM. **JDOM**. Disponível em: <<http://www.jdom.org/>>. Último acesso em: 30 nov. 2010.
- JQUERY. **jQuery: write less, do more**. Disponível em: <<http://jquery.com/>>. Último acesso em: 30 nov. 2010.
- LINKEDIN. **LinkedIn Developer Networks**. Disponível em: <<http://developer.linkedin.com/index.jspa>>. Último acesso em: 30 de nov. 2010.
- MILES, R.; HAMILTON, K. **Learning UML 2.0**. Cambridge: O'Reilly Media, 2006.
- MYSQL. **Introdução ao MySQL Administrator**. Disponível em: <<http://dev.mysql.com/doc/administrator/pt/mysql-administrator-introduction.html>>. Último acesso em: 30 de nov. 2010.
- ORACLE. **JavaBeans Technology**. Disponível em: <<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138795.html>>. Último acesso em 30 de nov. 2010.
- SUN MICROSYSTEMS. **JavaServer Pages Technology**. Disponível em: <<http://java.sun.com/products/jsp/>>. Último acesso em: 30 de nov. 2010.
- SUN MICROSYSTEMS. **Core J2EE Patterns – Data Access Object**. Disponível em: <<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>>. Último acesso em: 02 de dez. 2010.

THE APACHE SOFTWARE FOUNDATION. **Apache Tomcat**. Disponível em: <<http://tomcat.apache.org/>>. Último acesso em: 30 de nov. 2010.

W3C. **Extensible Markup Language (XML)**. Disponível em: <<http://www.w3.org/XML>>. Último acesso em: 30 de nov. 2010.

W3C. **HTML 4.01 Specification**. Disponível em: <<http://www.w3.org/TR/html4/>>. Último acesso em: 30 de nov. 2010.

Watt, D. A. **Programming Language Design Concepts**. Chichester: John Wiley & Sons, 2004.

YAHOO! **Yahoo! Search BOSS**. Disponível em: <<http://developer.yahoo.com/search/boss/>>. Último acesso em: 30 de nov. 2010.



# APÊNDICES

# APÊNDICE A – PLANO DE ESTÁGIO



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**  
**DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO**



## Plano de Estágio Integrado

**Amaury Bartolomeu Carneiro de Medeiros**

amaurymedeiros@gmail.com

Estagiário – UFCG

Agosto 2010

## **1. Informações Pessoais**

**Nome:** Amaury Bartolomeu Carneiro de Medeiros

**Matrícula:** 20811003

**Endereço residencial:** Rua Estelita Cruz, 1183 – Alto Branco

CEP: 58401-470, Campina Grande – PB.

**E-mail:** amaurymedeiros@gmail.com

**Telefones:** (83) 3341-7482

(83) 8889-2350

## **2. Ambiente de Estágio**

A E.life é uma empresa que tem como função ajudar as empresas a entender as dinâmicas das redes sociais e a relação com sua marca e segmento de atuação, entregando projetos que serão utilizados por áreas como Marketing, Comunicação, Relações Públicas, Atendimento ao Cliente e Inteligência de Mercado.

Através de monitorações quantitativas e qualitativas, estudo ad hoc e gestão de relacionamento, a E.life propicia resultados imparciais e independentes para os clientes. Os resultados são baseados nos comentários de usuários, sobre o cliente, nas redes sociais.

A sede da empresa localiza-se em São Paulo e existe uma filial em Portugal. A empresa emprega funcionários distribuídos de forma direta (em São Paulo) ou remota (Campina Grande, Recife, entre outras). O estágio será realizado em casa (home-office) e haverá constante contato com o supervisor via Skype.



### 3. Supervisão

#### ***Orientador Acadêmico***

**Nome:** Cláudio de Souza Baptista

**Endereço:** Laboratório de Sistemas de Informação, Departamento de Sistemas e Computação, Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande - Av. Aprígio Veloso, 882, Bodocongó, 58109-970, Campina Grande – PB, Brasil.

**Email:** [baptista@dsc.ufcg.edu.br](mailto:baptista@dsc.ufcg.edu.br)

#### ***Supervisor Técnico***

**Nome:** Alessandro Barbosa Lima

**Endereço:** E-Life Monitor Estudos de Mercado LTDA - Av. Paulista, 2300, Bela Vista, 01310-300, São Paulo – SP, Brasil.

**Email:** [albali@elife.com](mailto:albali@elife.com)

### 4. Resumo do problema

O poder das redes sociais na internet é bastante evidente nos dias de hoje. Através de sistemas desse tipo, milhões de usuários trocam mensagens diariamente. Muitas vezes, os usuários preferem comentar sobre determinada marca, produto ou serviço nas redes sociais, ao invés de recorrer ao serviço de atendimento da empresa responsável pelo mesmo.

Um dos fatores para que o usuário prefira buscar ajuda em redes sociais, ao invés de entrar em contato com a empresa, é que quase sempre existe alguém que vivenciou a mesma situação que ele e, muitas vezes, encontrou a solução para o problema. As empresas, no entanto, perdem o controle sobre estas informações que circulam diariamente na rede.

O BuzzMonitor é a solução da E.life para procurar sanar esse problema. Esse sistema é acessado via web e, através dele, é possível monitorar, analisar e agir sobre as conversações dos consumidores. Através do BuzzMonitor, os analistas da E.life produzem relatórios periódicos de como está a imagem da empresa em questão com base na monitoração do volume e na natureza de discussões de consumidores on-line em blogs, forums, sites pessoais e redes sociais.

## 5. Objetivos

O principal objetivo do trabalho é trabalhar em cima do desenvolvimento do sistema web BuzzMonitor, que se encontra na sua segunda versão. O BuzzMonitor divide-se em dois subsistemas: BuzzMonitor e BuzzMonitor Platform.

**Buzzmonitor:** Sistema de informação cuja finalidade é a gestão de testimonials (posts e comentários) na Web sobre os clientes cadastrados. Cada cliente possui um ou mais projetos, que provêm termos (ou menções) que devem ser encontrados nos testimonials. Cada testimonial está associado a um autor e um site, além de estar também ligado a um conjunto de tags.

**Buzzmonitor Platform:** Surgiu da necessidade de integração de outros sistemas da E.life ao Buzzmonitor. Provê a infra necessária para isso por meio do uso de widgets, pelos quais a interação com o usuário é feita, e tabs, que permite ao usuário organizar os seus widgets em diferentes abas. Esse subsistema também agrega informação dos atuais clientes, incluindo seus respectivos usuários. Com essa informação realiza mapeamento entre usuários e os esquemas nos quais estão localizados os dados do respectivo cliente. Cada cliente também possui informações a respeito dos demais sistemas (que não buzzmonitor) que possuem acesso.

A idéia inicial é trabalhar no desenvolvimento de dois subsistemas do BuzzMonitor e integrá-los com o BuzzMonitor Platform: Facebook Meter e E.life SAC.

**Facebook Meter:** Tem como objetivo monitorar e analisar o volume de buzz e criar ações de relacionamento com os consumidores no facebook. Através dele, é possível filtrar o que se fala da empresa através de classificações de termos e sentimentos.

**E.life SAC:** O E.life SAC coloca a empresa como participante da conversação com o cliente. Através desse sistema, a empresa pode categorizar comentários dos clientes e optar por responder, ou não, cada um deles.

Serão utilizadas as tecnologias MySQL[1], JavaScript[2], JavaServer Pages[3], Google Chart API[4], JavaScript Object Notation (JSON)[5] e Java[6] para o desenvolvimento dos sistemas.

## 6. Resultados Esperados

Espera-se evoluir o sistema web BuzzMonitor, trabalhando principalmente em cima dos subsistemas citados anteriormente. A meta é produzir os widgets requisitados para cada um dos subsistemas, fornecendo todos os dados que se deseja obter das redes sociais e demais websites que serão usados como fonte de pesquisa.

## 7. Metodologia

O intuito é que sejam dedicadas 4 horas diárias ao projeto, totalizando em 20 horas semanais, realizando o desenvolvimento de forma incremental e iterativa. Em intervalos menores ou iguais a uma semana, será feito um acompanhamento via Skype, no qual será apresentado os últimos avanços e novos requisitos para o sistema serão solicitados.

## 8. Atividades Planejadas

<b>ID</b>	<b>Atividade Realizada</b>	<b>Tempo Estimado</b>
T1	Familiarização com o sistema	20h
T2	Estudo de tecnologias e API's a serem usadas no sistema	20h
T3	Produção de relatórios semanais	10h
T4	Escrita do relatório final	40h
T5	Desenvolvimento de sistemas de monitoramento de redes sociais	90h
T6	Integração do subsistema desenvolvido	80h
T7	Reuniões semanais com o supervisor técnico	25h
T8	Reuniões com o supervisor acadêmico	15h
Total		300h

## **11. Aprovação**

Declaro para os devidos fins que aprovo o planejamento de atividades descritas neste documento como plano de estágio do aluno Amaury Bartolomeu Carneiro de Medeiros, matrícula 20811003.

---

**Cláudio de Souza Baptista**

Supervisor Acadêmico

---

**Alessandro Barbosa Lima**

Supervisor Técnico

---

**Joseana Macêdo Fchine**

Coordenador da Disciplina Estágio Integrado