

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

**RELATÓRIO DE ESTÁGIO**

**DESENVOLVIMENTO E EVOLUÇÃO DAS APLICAÇÕES  
ONIBUZZ E TWEE.LI**

**ARTHUR DE SOUZA RIBEIRO**

**Estagiário**

**CLÁUDIO DE SOUZA BAPTISTA**

**Orientador Acadêmico**

**ALESSANDRO BARBOSA LIMA**

**Supervisor Técnico**

**Campina Grande – PB**

**Dezembro de 2010**

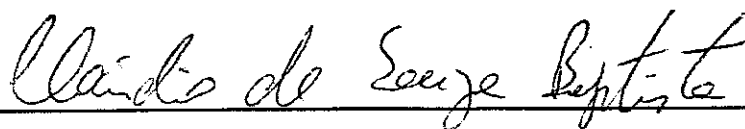


Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

**DESENVOLVIMENTO E EVOLUÇÃO DAS APLICAÇÕES ONIBUZZ E TWEE.LI**

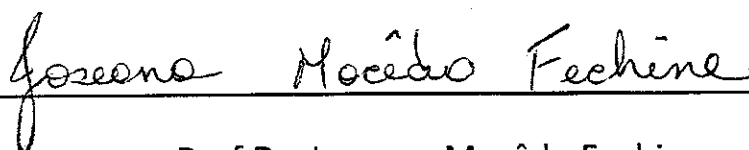
APROVADO EM \_\_\_\_\_

**BANCA EXAMINADORA**

---

Prof. Dr. Cláudio de Souza Baptista

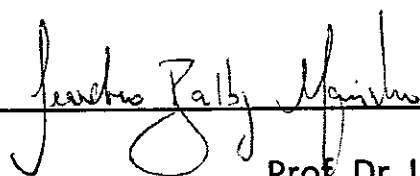
ORIENTADOR ACADÊMICO



---

Prof Dr. Joseana Macêdo Fachine

MEMBRO DA BANCA



---

Prof. Dr. Leandro Balby

MEMBRO DA BANCA

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente à empresa E-Life Monitor de Estudos de Mercado LTDA por ter me concedido a oportunidade de estagiar para a mesma, experiência que em muito me engrandeceu para tornar-me um profissional cada vez mais completo. Gostaria de estender meus agradecimentos também a Jairson Vitorino e Alessandro Barbosa, CTO e CEO da empresa, respectivamente, por nunca me deixar faltar nada e sempre conseguir me manter motivado e atualizado com as tarefas que a mim eram requisitadas.

Com relação à parte acadêmica do meu estágio, gostaria de agradecer à professora Joseana Fachine pela atenção dada aos alunos da disciplina, sempre cordial e procurando nos ajudar, e, como não poderia deixar de ser, ao meu orientador acadêmico, o professor Cláudio de Souza Baptista, por ter me orientado em como proceder no meu estágio me dando dicas e acompanhando de forma atuante meu estágio.

Além de tudo gostaria de agradecer à minha família, minha namorada e meus amigos, pelo apoio a mim concedido, nunca deixando que eu me abatesse nos momentos mais críticos.

## 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 do aluno Arthur de Souza Ribeiro cuja matrícula é: 20811005, com as atividades realizadas ao longo do período de 2010.2 sob orientação acadêmica do professor Cláudio de Souza Baptista e supervisão Técnica de Alessandro Barbosa Lima, perfazendo um total de 300 horas.

O estágio foi realizado na E-Life Monitor de Estudos de Mercado LTDA.

O conteúdo do relatório está distribuído conforme a descrição a seguir:

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

Durante o período letivo de 2010.2 o aluno Arthur de Souza Ribeiro desempenhou atividades como estagiário para a empresa E-Life Monitor de Estudos de Mercado LTDA. Durante o tempo em que estagiou, o estagiário trabalhou em duas aplicações que se relacionam com a plataforma do Twitter, o Onibuzz e o Twee.li. O Onibuzz é uma aplicação que utiliza um mecanismo P2P para criar uma rede interna de compartilhamento de links, que futuramente pode ser usado para verificar tendências na internet para empresas. O Twee.li é um comunicador instantâneo que visa melhorar a relação interpessoal entre empresas e seus consumidores. Neste relatório, são descritas as atividades realizadas pelo estagiário nestes dois projetos para a referida empresa.

## SUMÁRIO

AGRADECIMENTOS .....	3
APRESENTAÇÃO.....	4
RESUMO .....	5
SUMÁRIO .....	6
LISTA DE SIGLAS E ABREVIATURAS.....	7
LISTA DE FIGURAS.....	8
LISTA DE QUADROS .....	9
1. INTRODUÇÃO.....	11
2. AMBIENTE DE ESTÁGIO .....	13
2.1. SOBRE A EMPRESA .....	13
2.2. CARGA-HORÁRIA .....	13
2.3. INFRA-ESTRUTURA .....	13
2.4. SUPERVISÃO .....	14
2.5. ASPECTOS POSITIVOS.....	15
2.6. ASPECTOS NEGATIVOS.....	15
3. FUNDAMENTAÇÃO TEÓRICA .....	17
4. ATIVIDADES REALIZADAS .....	21
4.1. MUDANÇAS COM RELAÇÃO AO CRONOGRAMA INICIAL .....	21
4.2. PROJETO ONIBUZZ .....	24
4.2.1. O que é? .....	24
4.2.2. Atividades desenvolvidas no projeto.....	25
4.3. PROJETO TWEE.LI .....	33
4.3.1. O que é? .....	33
4.3.2. Atividades realizadas no projeto.....	35
5. CONSIDERAÇÕES FINAIS .....	44
APÊNDICES .....	48
APÊNDICE A: PLANO DE ESTÁGIO.....	49

## **LISTA DE SIGLAS E ABREVIATURAS**

- CEO – Chief Executive Officer
- CTO – Chief Technical Officer
- IDE – Integrated Development Enviroment
- API – Interface de Programação de Aplicações
- P2P – Entre pares, arquitetura de sistemas distribuídos caracterizada pela descentralização das funções na rede
- BD – Banco de Dados
- MVC – Model View Controller.
- OO – Orientação a Objetos
- DAO – Data Access Object
- SQL – Script Query Language



**LISTA DE FIGURAS**

Figura 1 - Arquitetura do Onibuzz .....	25
Figura 2 - Ilustração do MVC .....	26
Figura 3 - Esquema conceitual do BD do servidor do Onibuzz .....	27
Figura 4 - Exemplo do advertisement enviado quando se procura termo futebol.....	29
Figura 5 - Tela de login para administrador do Onibuzz.....	30
Figura 6 - Tela que mostra contas de Twitter cadastradas no Onibuzz.....	30
Figura 7 - Planilha com resultados gerados a partir do servidor .....	31
Figura 8 - Página para adição de propaganda no servidor.....	31
Figura 9 - Exemplo de tweet enviado por e-mail através do Onibuzz.....	32
Figura 10 - Home Page do Onibuzz .....	32
Figura 11 - Skin do Twee.li para a Hellmann's. ....	34
Figura 12 - Tela de chat em grupo para a empresa.....	35
Figura 13 - Telas antes e depois fazer uma busca por usuarios.....	36
Figura 14 - Telas antes e depois do refatoramento .....	37
Figura 15 - Tela de menções do Twee.li .....	38
Figura 16 - Aba <i>timeline</i> do Twee.li.....	39
Figura 17 - Log de conversas de usuário .....	40
Figura 18 - Telas de adicionar e aceitar contato, respectivamente .....	41

**LISTA DE QUADROS**

Quadro 1 - Disposição das atividades planejadas.....	22
Quadro 2 - Disposição das atividades executadas.....	22
Quadro 3 - Cronograma das atividades planejadas .....	23
Quadro 4 - Cronograma das atividades executadas.....	23

SEÇÃO I

# INTRODUÇÃO

## 1. INTRODUÇÃO

É imprescindível na vida de qualquer profissional, desde a formação acadêmica, alguma maneira de interagir com o mercado de trabalho, adquirindo experiência, descobrindo como se dão as relações e procurando sempre resolver problemas quando se estiver em situações adversas. Neste contexto, surge a disciplina de estágio integrado, que auxilia o estudante a praticar os conhecimentos adquiridos na graduação ao passo em que o estudante está vinculado a uma empresa do mundo real, sabendo assim, como se dão os problemas no mercado.

Com base nisso, o aluno Arthur de Souza Ribeiro participou de um estágio integrado durante o período letivo de 2010.2 na empresa E-Life Monitor de Estudos de Mercado LTDA, que é a líder em monitoração e análise da mídia gerada pelo consumidor e gestão de relacionamentos em mídias sociais na América Latina e Portugal.

O estágio tem por principal objetivo, utilizar os conhecimentos do estagiário no desenvolvimento e evolução de duas aplicações da empresa (Onibuzz e Twee.li, respectivamente). Aplicações estas, que serão/foram lançadas no mercado e que deram ao estagiário a oportunidade única de conseguir desenvolver software a ser lançado no mesmo, além de aprimorar ainda mais os conhecimentos adquiridos ao longo da graduação.

Como objetivos específicos do estágio, estão:

- a) Descobrir uma boa arquitetura P2P para o Onibuzz
- b) Desenvolvimento do módulo do servidor do Onibuzz
- c) Realizar estudos sobre Adobe AIR para fazer uso nas aplicações cliente
- d) Adicionar novas funcionalidades ao Twee.li que passará a ser vendido para empresas
- e) Manter e adicionar novas funcionalidades ao Onibuzz, para o seu lançamento.

Ao longo deste relatório, serão descritas como foram realizadas as atividades para cumprimento dos objetivos específicos do estágio.

## SEÇÃO II

# AMBIENTE DE ESTÁGIO

## **2. AMBIENTE DE ESTÁGIO**

### **2.1. SOBRE A EMPRESA**

A E.life foi fundada em 2004 com foco na monitoração e análise de mídias sociais. A empresa extrai a partir de milhões de opiniões o núcleo das que são relevantes para fornecer insights de marketing únicos para seus clientes. São analisados todos os tipos de conteúdo publicado espontaneamente na internet pelos consumidores descrevendo sua experiência com produtos. A E.life possui mais de 60 projetos completados, e é ativa em seis países.

Dentre seus objetivos principais está entender as empresas a entender dinâmicas das redes sociais e a relação com sua marca e segmento de atuação, entregando projetos que serão utilizados por áreas como Marketing, Comunicação e RP, Atendimento ao Cliente e Inteligência de Mercado. Orientar os clientes com relação à percepção de marca e serviços, comportamento e hábitos de consumo, tendências e relacionamento com consumidor a partir das redes sociais. Em resumo, a E-Life trata-se de uma empresa de inteligência de mercado

O setor no qual o estágio foi desenvolvido foi o setor de desenvolvimento, que possui dois núcleos, o núcleo Recife, com escritório próprio e profissionais contratados, e o núcleo Campina Grande que conta com este estagiário mais alguns outros que foram contratados. Neste setor estão relacionadas as atividades de desenvolvimento de software da empresa, seja de soluções Web ou soluções Desktop.

### **2.2. CARGA-HORÁRIA**

O estágio ocorreu entre os dias 01/08/2010 e 01/12/2010, em que foi cumprida a carga-horária de 20 horas semanais totalizando 300 horas de trabalho, descontando dias em que foi feriado.

### **2.3. INFRA-ESTRUTURA**

Devido a empresa ter chegado a pouquíssimo tempo em Campina Grande (chegou em Junho de 2010), a empresa não possui ainda escritório próprio na cidade, portanto, o estágio era realizado em modelo *Home Office*, em que o estagiário tem a liberdade de realizar suas atividades de casa e reportar ao supervisor da empresa todas as semanas. Todos os atuais integrantes do núcleo Campina Grande utilizam seus computadores pessoais para realização das atividades, e todos atuam no regime de *Home Office*.

Para a realização das atividades desse estágio, foi utilizado um *Notebook Semp*

*Toshiba IS 1462* com processador *Intel Pentium Dual Core Mobile T2330 1.6 GHz*, memória RAM principal de *2 GB* e disco rígido de *160 GB*. Neste instrumento de Hardware, foi utilizado para o desenvolvimento das atividades, o sistemas operacional *Microsoft Windows XP*, e como IDEs de desenvolvimento *eclipse*, *Flex Builder 4*, *Flex Builder 3*, e *Adobe Flash Professional CS5*.

## 2.4. SUPERVISÃO

### Supervisão Técnica

- **Nome:**  
Alessandro Barbosa Lima
- **Função:**  
CEO
- **Endereço Profissional:**  
E-Life Monitor Estudos de Mercado Ltda  
Al. Ribeirão Preto, 130, Sala 41, Bela Vista.  
CEP: 01.331-000.  
São Paulo – SP.
- **E-mail:**  
[alwali@elife.com](mailto:alwali@elife.com)

### Supervisão Acadêmica

- **Nome:**  
Cláudio de Souza Baptista
- **Função:**  
Professor adjunto do departamento de sistemas e computação (DSC)
- **Endereço Profissional:**  
Rua Aprígio Veloso, 882, Bodocongó  
Bloco CN – Sala 210  
CEP: 58109-970  
Campina Grande – Paraíba, Brasil
- **E-mail:**  
[baptista@dsc.ufcg.edu.br](mailto:baptista@dsc.ufcg.edu.br)  
[claudiobaptista@yahoo.com.br](mailto:claudiobaptista@yahoo.com.br)

## **2.5. ASPECTOS POSITIVOS**

Como aspectos positivos de como a realização do estágio transcorreu, pode-se destacar a flexibilidade de horários por parte do estagiário, em que este poderia montar sua escala de horários, além de, devido a modalidade de trabalho ser *Home-Office*, o conforto para o estagiário é bem maior pois não é necessário sair de sua casa para realizar as tarefas relacionadas ao estágio.

## **2.6. ASPECTOS NEGATIVOS**

Como aspectos negativos do modelo de estágio realizado, gostaria de destacar a distância com relação a equipe de trabalho (boa parte encontra-se em Recife), algumas vezes o trabalho do estagiário, dependia de algum funcionário, e essa distância impossibilitava que um ajudasse o outro de forma mútua, deixando o trabalho mais eficiente.



SEÇÃO III

FUNDAMENTAÇÃO

TEÓRICA

### 3. FUNDAMENTAÇÃO TEÓRICA

Devido o estágio realizado ser basicamente a respeito de desenvolvimento de Software, muitas ferramentas foram utilizadas. Algumas dessas ferramentas não eram de conhecimento do estagiário, enquanto outras eram. Este estágio também foi uma ótima oportunidade para ser colocados em prática conceitos vistos durante a graduação, que em muito foram úteis no desenvolvimento pleno das tarefas requisitadas pela empresa.

Foram utilizados os seguintes conceitos e tecnologias a fim de realizar as tarefas solicitadas ao estagiário, bem como para apresentar resultados:

**Paradigma de programação orientado a objetos** – Paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação de diversas unidades de software chamadas objetos.

**Banco de dados relacional** – É um conceito abstrato que define maneiras de armazenar, manipular e recuperar dados estruturados unicamente na forma de tabelas, construindo um banco de dados.

**Model View Controller (MVC)** – Padrão de arquitetura de software que visa separar a lógica de negócio da lógica de apresentação, permitindo o desenvolvimento, teste e manutenção isolado de ambos

**Data Access Object (DAO)** – Padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados.

**Java** – Linguagem de programação orientada a objetos, desenvolvida pela Sun Microsystems. É uma linguagem de programação portátil, com recursos de segurança e de interação em rede.

**Apache Tomcat** – O Tomcat é um servidor web Java, que tem a capacidade de atuar como servidor Web, ou pode funcionar integrado a um servidor Web dedicado.

**MySQL** – É um sistema de gerenciamento de banco de dados (SGBD) que utiliza a linguagem SQL (Script Query Language) como interface. É um dos SGBDs mais utilizados no mundo.

**Java Database Connectivity (JDBC)** – É um conjunto de interfaces escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional.

**Java Server Pages (JSP)** – Tecnologia utilizada no desenvolvimento de aplicações Web. Que possui a vantagem do usuário poder integrar código Java a página que está rodando no servidor Web.

**JSP Standard Tag Library (JSTL)** – JSTL consiste em uma coleção de bibliotecas, tendo cada uma um propósito bem definido, que permitem escrever páginas JSPs sem código Java, aumentando assim a legibilidade do código e a interação entre desenvolvedores e web designers.

**Apache POI** – Biblioteca da Apache, que tem por objetivo facilitar a leitura e escrita de arquivos em formatos do *Microsoft Office* através de código Java.

**GSON** – É uma biblioteca Java para converter objetos Java em Strings JSON, e também converter strings JSON em objetos Java.

**Apache Commons FileUpload** – Biblioteca da Apache para conseguir, de forma simples, realizar upload de arquivos através de Java Servlets.

**JavaMail** – API de Java utilizada para enviar e-mails.

**Adobe AIR** – Framework da Adobe para criação de aplicações *Desktop* a partir de tecnologias de desenvolvimento para páginas Web.

**Tweetr** – Biblioteca para interação de aplicações feitas com Adobe AIR com o servidor do Twitter.

**Adobe Cirrus** – API da Adobe que habilita conexão entre pares numa rede, utilizando *Real Time Media Flow Protocol (RTMP)* através da plataforma *Adobe Flash*.

**Subversion** – É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer, é recomendado principalmente para desenvolvimento em equipe.

**Eclipse** – IDE desenvolvida em Java, com código aberto, para a construção de programas de computador.

**Adobe Flex Builder** – IDE utilizada para desenvolver aplicações em Adobe AIR.

**SSH Secure Shell** – Cliente de Secure Shell (SSH) para fornecer acesso seguro para administração em um servidor.

**brModelo** – Ferramenta para facilitar a criação de modelos relacionais de um banco de dados.

**Microsoft Word** – Processador de texto produzido pela Microsoft.

**Microsoft PowerPoint** – Programa utilizado para edição e exibição de apresentações gráficas produzido pela Microsoft.

Os conceitos e as ferramentas que acima foram citadas, relacionaram-se de alguma forma com as atividades que estagiário desempenhou, seja ela ligada a desenvolvimento de software, interação com equipe de desenvolvimento e para apresentação de resultados para orientadores.

## SEÇÃO IV

# ATIVIDADES REALIZADAS

#### **4. ATIVIDADES REALIZADAS**

No período de estágio, as atividades realizadas pelo estagiário, ficaram concentradas em duas aplicações, uma que já havia sido desenvolvida e precisava de evolução e melhorias (Twee.li) e outras que seria desenvolvida desde o início (Onibuzz).

Nas próximas seções serão descritas as mudanças que ocorreram com relação ao cronograma inicial que havia sido previsto para o estagiário, as atividades realizadas no projeto Onibuzz e as atividades realizadas no projeto Twee.li, explicando o que é cada projeto e as atividades realizadas de forma cronológica.

##### **4.1. MUDANÇAS COM RELAÇÃO AO CRONOGRAMA INICIAL**

O cronograma inicial de atividades planejadas está descrito no Apêndice A deste documento. No início do processo de estágio, estava previsto que o estagiário trabalharia apenas no desenvolvimento da aplicação Onibuzz, porém, devido a questões mercadológicas da empresa, o mesmo teve de trabalhar também na aplicação Twee.li, alterando assim as atividades que seriam realizadas, bem como seu cronograma.

No início do estágio estava planejado passar apenas 40 horas trabalhando nas funcionalidades que seriam acrescentadas ao servidor, porém, devido à demanda ter crescido bastante, esse tempo passou para 75 horas. A atividade relacionada à criação das classes de interação com o banco de dados também sofreu alterações no tocante ao número de horas que seriam gastas com a mesma, esse número passou de 20 horas para 10 horas. As atividades no Twee.li demandaram 80 horas, por serem relacionadas a adição de muitas funcionalidades e também a correção de outras que estavam com problemas. No tocante ao desenvolvimento na parte cliente da aplicação Onibuzz, houve uma mudança no enfoque desta atividade, ou seja, o estagiário teria apenas de verificar possíveis bugs e elencar novas funcionalidades a serem implementadas. E, por fim, as atividades de preparação do relatório e da defesa de estágio tomaram as últimas semanas do mesmo, pois, a partir daí tinha-se todas as atividades que foram realizadas, de forma a facilitar a preparação destes, no fim, estas atividades totalizaram 35 e 10 horas, cada.

A análise acima descrita é ilustrada nos quadros seguintes, onde é possível ver como era a organização das atividades no início e ao fim do estágio (quadros 1 e 2), da mesma forma que é possível visualizar como era o cronograma no início e como ficou ao fim do estágio (quadros 3 e 4).

Quadro 1 - Disposição das atividades planejadas

<b>ID</b>	<b>Atividade</b>	<b>Tempo</b>
T1	Estudo das tecnologias a serem utilizadas	40 horas
T2	Modelar tabelas do Banco de Dados	20 horas
T3	Criar classes de interação com Banco de Dados no servidor	20 horas
T4	Criação dos serviços a serem providos pelo servidor	40 horas
T5	Reuniões semanais com supervisor técnico e acadêmico	25 horas
T6	Desenvolvimentos a serem efetuados na aplicação cliente	50 horas
T7	Evolução do sistema e elicitação e implementação de novos requisitos	40 horas
T8	Escrita do relatório de estágio	45 horas
T9	Preparação e apresentação da defesa do estágio	20 horas
Total		300 horas

Quadro 2 - Disposição das atividades executadas

<b>ID</b>	<b>Atividade</b>	<b>Tempo</b>
T1	Estudo das tecnologias a serem utilizadas	40 horas
T2	Modelar Banco de Dados e suas respectivas tabelas no servidor do Onibuzz	20 horas
T3	Criar classes de interação com Banco de Dados no servidor	10 horas
T4	Criação dos serviços a serem providos pelo servidor	75 horas
T5	Reuniões semanais com supervisor técnico e acadêmico	20 horas
T6	Desenvolvimentos a serem realizados na aplicação Twee.li	80 horas
T7	Elicitação de requisitos para o Onibuzz e testes nas funcionalidades implementadas	10 horas
T8	Escrita do relatório de estágio	35 horas
T9	Preparação e apresentação da defesa do estágio	10 horas
Total		300 horas

Quadro 3 - Cronograma das atividades planejadas

	Agosto				Setembro				Outubro				Novembro				Dezembro		
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17		
T1	■																		
T2		■																	
T3			■																
T4				■	■	■													
T5	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
T6							■	■	■	■	■								
T7											■	■	■	■	■				
T8											■	■	■	■	■	■	■		
T9																■	■		

Quadro 4 - Cronograma das atividades executadas

	Agosto				Setembro				Outubro				Novembro				Dezembro		
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17		
T1	■							■											
T2		■																	
T3			■																
T4			■	■	■	■	■												
T5	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
T6									■	■	■	■	■						
T7													■	■					
T8															■	■	■		
T9																■	■		



## 4.2. PROJETO ONIBUZZ

Neste tópico, serão abordadas as atividades que o estagiário realizou no projeto Onibuzz, explicitando o que é este projeto, como ele se encaixa no contexto da empresa, e, por fim, as contribuições do estagiário para o mesmo.

Neste projeto, o estagiário ficou responsável por estudar uma arquitetura P2P para o restante da equipe do projeto, pelos desenvolvimentos no módulo de servidor da aplicação e também por verificar bugs e novas funcionalidades na aplicação cliente quando esta ficasse pronta.

### 4.2.1. O que é?

O Onibuzz é a primeira aplicação que usa protocolos P2P para espalhar e classificar links a partir da plataforma do Twitter. A idéia principal é mostrar todos os tweets de quem o usuário segue e que menciona uma URL. Esses tweets são filtrados e, a partir deles é mostrada uma pequena página com uma breve descrição do que o link contém. A ordem com que os tweets são mostrados para o usuário pode ser cronológica, por filtragem de tags, usuário ou número de pessoas que adicionaram o mesmo à sua lista de favoritos.

A aplicação possui dois tipos de usuários, os usuários *free* (gratuitos) que não pagam para usar a aplicação e tem acesso a algumas funcionalidades da aplicação, e os usuários *onibuzz*, que pagam para fazer uso de aplicação e possuem acesso a mais funcionalidades que os usuários *free*.

Essa é uma das formas de como a empresa pode fazer lucro com a aplicação, a outra é através de um mecanismo de propaganda, cadastrando *advertisements* (propagandas) para localidades específicas, e enviando para os clientes que sejam de localidades compatíveis.

No contexto da E.life, o Onibuzz é uma tentativa de *crowdsourcing* em uma comunidade ao redor do Twitter para identificar tendências via o mecanismo de adicionar aos favoritos e associar tags via P2P, a idéia é conseguir, a partir de uma amostra da rede, informações para conseguir fazer abordagens e estudos para empresas.

O desenvolvimento da aplicação foi realizado pela E.life em conjunto com a Comment Lab, empresa sediada em Recife e que ficou, juntamente com o estagiário, responsável pelo desenvolvimento da aplicação, sendo o mesmo o único da E.life a estar engajado no projeto.

#### 4.2.2. Atividades desenvolvidas no projeto

Nesta seção, serão mostradas as atividades que foram realizadas neste projeto seguindo ordem cronológica, e também a forma como se desenvolveu a atividade.

- Estudo de tecnologia para aplicação cliente

Para a viabilidade do Onibuzz, a primeira questão abordada foi como que seria feita a comunicação entre pares na aplicação. A primeira alternativa foi o uso do protocolo BitTorrent, porém, o mesmo se mostrou útil apenas para compartilhamento de arquivos, fugindo totalmente do escopo da aplicação.

A alternativa adotada foi fazer uso do *framework* Cirrus da Adobe. Usando esta ferramenta, cria-se uma rede interna em que os usuários podem se comunicar diretamente, evitando assim, sobrecarga e dependência de servidores. Ao fazer uma busca na rede Cirrus do Onibuzz, um usuário pergunta a todos da rede, quem possui ocorrência da busca envia o resultado direcionado a quem solicitou a mesma.

As requisições de busca são enviadas a alguns nós na rede e esse vão repassando aos demais, porém as respostas vêm diretamente para o nó fonte da rede que requisitou a busca. Na Figura 1, é ilustrada a arquitetura final dos estudos.

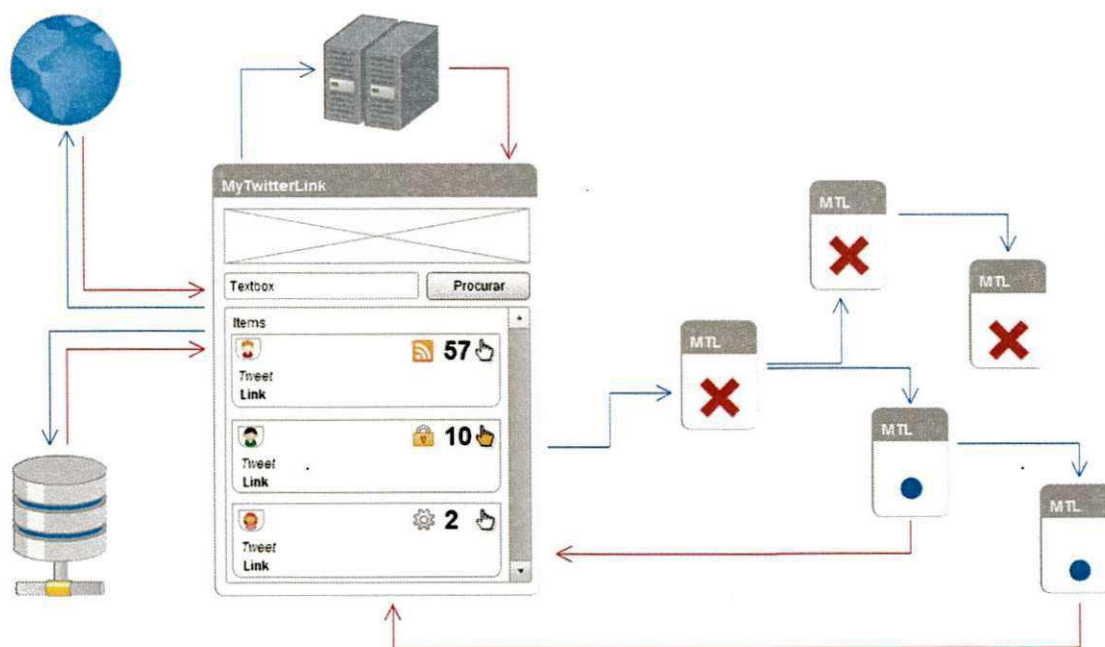


Figura 1 - Arquitetura do Onibuzz

- **Estudo de tecnologias para módulo do servidor**

Para o módulo do servidor da aplicação, que serve para a empresa administrá-la (gerenciar usuários, propagandas e pesquisas realizadas), foi adotada como tecnologia a linguagem de programação Java, por já ser de conhecimento do estagiário, diminuindo assim, a quantidade de horas para aprender uma nova tecnologia.

Para a arquitetura do mesmo, foi escolhido utilizar o padrão MVC (*Model View Controller*) que permite separar bem as entidades relacionadas aos dados das entidades relacionadas a serviços e da lógica de apresentação.

Na Figura 2, é apresentado o comportamento do MVC. O usuário faz uma requisição a uma controladora (*controller*) que usa dados do modelo (*model*) e responde através da visão (*view*) para o usuário.

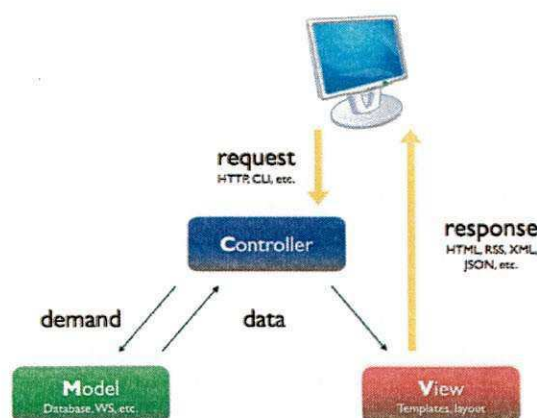


Figura 2 - Ilustração do MVC

Apesar do conhecimento da linguagem de programação por parte do estagiário, o mesmo teve de estudar como a tecnologia adotada se manifestava em programação para Web. Para isto algumas horas foram gastas estudando Servlets para a parte de controle do servidor, JSP e JSTL para a parte de visões, de mostrar para os usuários do servidor os dados que os mesmo requeressem, e também a tecnologia JDBC para interação do servidor com o banco de dados MySQL, que foi adotado por ser grátis, ter uma boa estabilidade e ser de conhecimento do estagiário.

Além disso, precisou-se estudar uma maneira para que o desenvolvimento do módulo se desse de acordo com o modelo arquitetural escolhido, para isso, utilizou como material de

estudo uma apostila da Caelum, que é uma empresa que ministra cursos em diversas áreas inclusive em programação para Web.

### • Modelar Banco de dados e suas tabelas do servidor do Onibuzz

Como atividade seguinte ao estudo das tecnologias, o estagiário modelou o banco de dados que fica no servidor do Onibuzz, para isto, primeiramente foi feito um modelo conceitual, e, a partir desse, foi feita a implementação das tabelas que seriam utilizadas.

Para o modelo conceitual do BD da aplicação, partiu-se do pressuposto de que uma conta *Onibuzz* pode possuir uma ou mais contas de Twitter, e que cada propaganda (*advertisement*) teria uma localização e os termos que apontariam para ela. Cada conta de Twitter pode fazer várias pesquisas que ficam cadastradas no banco de dados da aplicação e cada conta do tipo *Onibuzz* está associada a um tipo, que, dependendo de qual seja, pode conceder mais benefícios ao usuário.

Após entendimento do domínio da aplicação e de como suas respectivas informações seriam armazenadas no banco de dados, o esquema conceitual ficou como está descrito na Figura 3.

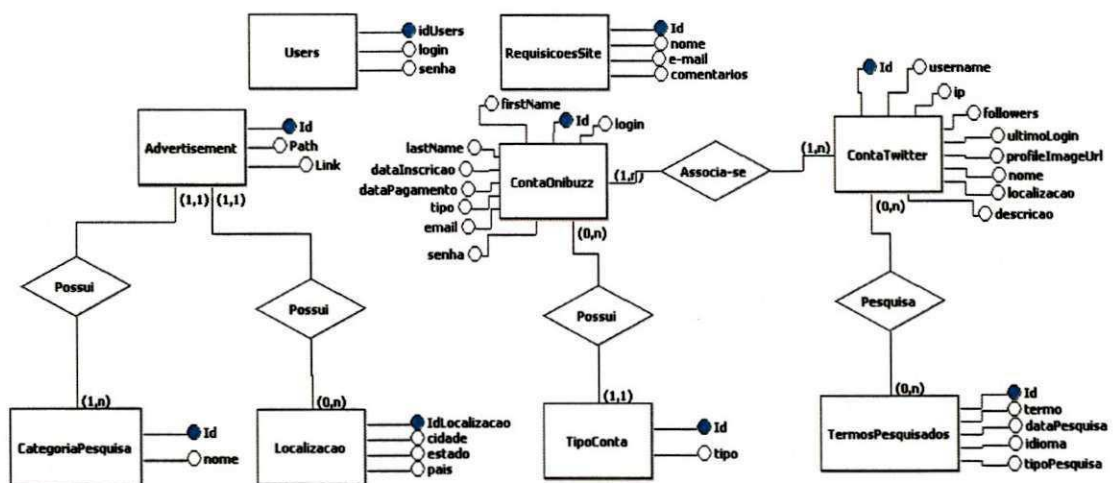


Figura 3 - Esquema conceitual do BD do servidor do Onibuzz

A partir do esquema conceitual ficou mais simples de entender como ficariam organizadas as tabelas no banco, assim, partiu-se para a fase de implementação das tabelas. Finalizada esta etapa foram obtidas 12 tabelas para o armazenamento eficiente dos dados.



- **Criação de classes de interação do servidor com o banco de dados**

Para a interação do servidor com o banco de dados, foi definido que o usuário utilizaria da tecnologia JDBC, esta foi escolhida pelo mesmo por se tratar de uma API que não requer muito tempo de configuração do sistema, é de mais fácil aprendizado e por já estar integrada com programação Java para Web no material de estudo que foi utilizado.

Para a construção das conexões com o banco de dados, foi utilizado o *design pattern* conhecido como *Factory* que prega o encapsulamento da construção de objetos complicados.

Para a construção das classes de interação com o banco de dados, também foi utilizado outro *design pattern*, este conhecido como *Data Access Object (DAO)*, através do mesmo se isola o acesso ao banco de dados em classes simples, cuja instância é um objeto responsável por acessar os dados.

Nesta etapa foram desenvolvidas as classes que representam entidades no banco de dados do servidor, bem como as respectivas classes de interação do servidor com o BD. Nessas classes são realizadas operações básicas, como, adicionar, editar e remover uma linha de uma tabela e algumas operações que são específicas das mesmas.

- **Criação dos serviços a serem providos pelo servidor**

Nesta etapa do desenvolvimento, foram implementadas as classes que representam serviços no servidor, esses serviços ficaram na forma de Servlets e realizam diferentes funções no mesmo.

O primeiro serviço feito para o servidor diz respeito à adição de contas de Twitter que se logaram na aplicação. Neste serviço, a aplicação faz uma requisição de que um usuário quer ter acesso, o servidor verifica a situação do mesmo e retorna se o usuário trata-se de algum do tipo *free* ou do tipo *onibuzz* e se está em atraso ou em dia. Caso o usuário não exista no banco de dados, o mesmo é adicionado para que o administrador da aplicação saiba quem já a usou. As respostas para a aplicação cliente são enviadas na forma de JSON que é um formato leve para intercâmbio de dados computacionais, formato este definido pelo usuário com o restante da equipe de desenvolvimento e que é gerado a partir da biblioteca GSON.

Após este serviço, foi feito o serviço para registrar os termos que eram pesquisados na aplicação, estes termos são obtidos a partir de requisições HTTP, são registrados de acordo com quem pesquisou, com a data, com o tipo da pesquisa e com o idioma. Após registrar o

termo no banco de dados, o servidor busca um *advertisement* que melhor se encaixa com a localidade do usuário e que tem mais relevância para o termo que ele pesquisou e o envia, tendo assim um mecanismo para mudanças de *advertisement* a partir da localidade e do interesse do usuário. Na Figura 4, é ilustrado o envio de um *advertisement* para um dado termo pesquisado.



Figura 4 - Exemplo do advertisement enviado quando se procura termo futebol

Após fazer esses dois serviços básicos de interação com a aplicação cliente, o estagiário direcionou suas atividades para a parte das páginas onde seria realizada a administração do Onibuzz, para esta tarefa foram utilizados conhecimentos de JSP e JSTL para a montagem da interface, foi solicitado que esta interface fosse bem simples, por se tratar de páginas que apenas o administrador do sistema pode acessar.

A primeira funcionalidade feita nessa parte de administração foi um mecanismo de login para restringir o uso a apenas quem possuir login e senha na aplicação. Esta atividade foi concluída com sucesso e a tela como é ilustrada na Figura 5.

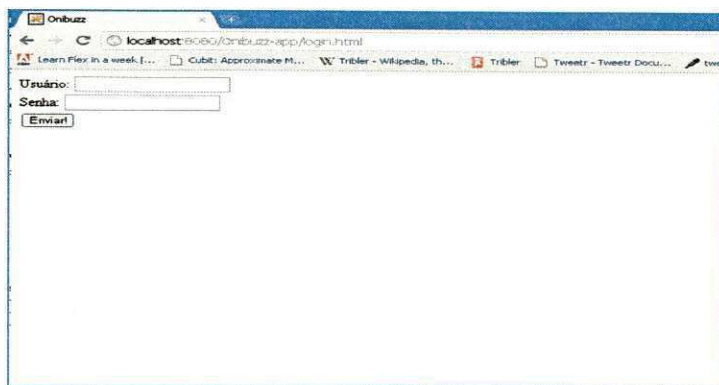


Figura 5 - Tela de login para administrador do Onibuzz

Feita esta funcionalidade, partiu-se para a parte de administração propriamente dita da aplicação. A primeira parte encontra-se no que diz respeito a listar as contas de Twitter que já acessaram a mesma. Para isto bastou acessar os dados que estavam gravados no banco de dados e montar a respectiva tela, que, é ilustrada na Figura 6.

Username	Ip	Nome	Tweets	Seguidores	Localizações	Perfil	Imagem	URL	Último Login	Data de Cadastro
arturibuzz	187.78.136.187	Arthur Ribeiro	aaa	87	Campos Gerais	aaa			10/09/2010	09/09/2010
ip	187.78.136.187		0	0					20/08/2010	20/08/2010
ip1	200.17.134.70		0	0					20/08/2010	20/08/2010
ip2	200.17.134.70		0	0					20/08/2010	20/08/2010
ip3	200.17.134.70		0	0					20/08/2010	20/08/2010
ip4	200.17.134.70		0	0					20/08/2010	20/08/2010
ip5	187.78.136.187		1	1					09/09/2010	09/09/2010
teste	187.78.111.61		0	0					20/08/2010	20/08/2010
testa	187.78.111.61		0	0					20/08/2010	20/08/2010
teste2	187.78.111.61		0	0					20/08/2010	20/08/2010
teste3	187.78.111.61		0	0					20/08/2010	20/08/2010
teste4	200.17.134.70		0	0					20/08/2010	20/08/2010
teste5	187.78.111.61		0	0					20/08/2010	20/08/2010
teste6	187.78.111.61		0	0					20/08/2010	20/08/2010
teste7	187.78.111.61		0	0					20/08/2010	20/08/2010
teste8	200.17.134.70		0	0					20/08/2010	20/08/2010
teste9	200.17.134.70		0	0					20/08/2010	20/08/2010
teste10	200.17.134.70		0	0					20/08/2010	20/08/2010
teste11	187.78.136.187		0	0					20/08/2010	20/08/2010
teste12	187.78.136.187		0	0					20/08/2010	20/08/2010
teste13	187.78.136.187		0	0					20/08/2010	20/08/2010
teste14	187.78.136.187		0	0					20/08/2010	20/08/2010
teste15	187.78.136.187		0	0					20/08/2010	20/08/2010
teste16	187.78.136.187		0	0					20/08/2010	20/08/2010
teste17	187.78.136.187		0	0					20/08/2010	20/08/2010
teste18	187.78.136.187		0	0					20/08/2010	20/08/2010
teste19	187.78.136.187		0	0					20/08/2010	20/08/2010
teste20	187.78.136.187		0	0					20/08/2010	20/08/2010

Figura 6 - Tela que mostra contas de Twitter cadastradas no Onibuzz

Da mesma forma que foi feita a tela para mostrar as contas de Twitter, também foram feitas telas para mostrar as contas *Onibuzz* e os termos pesquisados que estão cadastrados na aplicação.

Feita esta parte das páginas de relatório do Onibuzz, foi requisitado que fosse gerado um arquivo do tipo XLS (formato para *Microsoft Excel*) no mesmo formato do conteúdo das páginas JSP. Para isto, foi usada a biblioteca Apache POI que pode ser usadas para gerar arquivos do tipo solicitado. Os arquivos são gerados com sucesso e um exemplo pode ser visto através da Figura 7, que ilustra um documento gerado através do servidor do Onibuzz aberto no *Microsoft Excel*.



Termo Pesquisado	Username	Localização
1. automovel	arthursilveira	Brasil
2. automovel	arthursilveira	arthursilveira
3. automovel	arthursilveira	Loitões Inglaterra
4. automovel	arthursilveira	logiteira
5. automovel	arthursilveira	Campina Grande
6. automovel	arthursilveira	Campina Grande
7. automovel	arthursilveira	Campina Grande
8. automovel	arthursilveira	Campina Grande
9. automovel	arthursilveira	Campina Grande
10. automovel	arthursilveira	Campina Grande
11. automovel	arthursilveira	Campina Grande
12. automovel	arthursilveira	Campina Grande
13. automovel	arthursilveira	Campina Grande
14. automovel	arthursilveira	Campina Grande
15. automovel	arthursilveira	Campina Grande
16. automovel	arthursilveira	Campina Grande
17. automovel	arthursilveira	Campina Grande
18. automovel	arthursilveira	Campina Grande
19. automovel	arthursilveira	Campina Grande
20. automovel	arthursilveira	Campina Grande
21. automovel	arthursilveira	Campina Grande
22. automovel	arthursilveira	Campina Grande

Figura 7 - Planilha com resultados gerados a partir do servidor

Após a etapa de mostrar dados gravados no banco de dados foi feita a parte de gerência de propagandas, onde a parte que mais deu trabalho para o estagiário foi no que diz respeito ao *upload* de arquivos (imagens a serem usadas nas propagandas). Para esta parte foi utilizada a biblioteca Apache Commons FileUpload, que foi de fácil entendimento para o usuário e possibilitou a conclusão desta funcionalidade de forma satisfatória.

Na Figura 8, é apresentada a tela de adição de um *advertisement* à aplicação. Ao clicar no botão 'selecionar arquivo' uma tela é aberta onde se escolhe o arquivo para ser feito o *upload*, bem como é associado o respectivo nome e link a propaganda.



Figura 8 - Página para adição de propaganda no servidor

Feita a parte de gerência de *advertisements*, a última parte restante no tocante aos serviços a serem implantados no servidor da aplicação foi a parte de envio de tweets por e-mail a partir de requisições HTTP. Para esta etapa, foi utilizada a API JavaMail, que possibilita que sejam enviados e-mails a partir de código Java. Também foi necessária uma configuração no servidor da empresa, para que fosse habilitada a funcionalidade de SMTP, que é o protocolo para envio de e-mails. Esta task também foi concluída de forma satisfatória e fechou as atividades de serviços a serem providos pelo servidor, restando apenas fazer programar o layout da Home Page da aplicação.



Na Figura 9, é ilustrado um e-mail que foi enviado a partir do Onibuzz. O endereço do remetente é [no-reply@onibuzz.com](mailto:no-reply@onibuzz.com) que é o endereço que foi habilitado no servidor SMTP da aplicação.



Figura 9 - Exemplo de tweet enviado por e-mail através do Onibuzz

Por fim, foi realizada a programação da Home Page da aplicação, informando quantas pessoas a mencionaram no Twitter ou no Facebook e a adição de um formulário para o usuário fazer comentários ou tirar dúvidas a respeito da aplicação além de uma badge em Flash para instalar a aplicação diretamente do browser. Feito isso, ficou concluída toda a parte do módulo do servidor do Onibuzz.

Na Figura 10, é ilustrado o layout da Home Page da aplicação. No canto inferior estão localizadas as menções no Twitter e no Facebook. O botão laranja localizado à direita na página é o botão utilizado para se fazer o download da aplicação.



Figura 10 - Home Page do Onibuzz

- **Elicitação de requisitos e testes no Onibuzz**

Nesta última atividade relacionada ao Onibuzz, foi solicitado ao estagiário que elencasse novos requisitos a serem implementados e bugs a serem consertados.

Após reuniões e um período utilizando a aplicação para elencar bugs e novas funcionalidades, foi elencada uma nova funcionalidade para a aplicação funcionar também como um leitor de links completo, não só mostrando agora uma breve descrição, mas sim todo o seu conteúdo, separando por tag. Também foram elencados alguns bugs na busca descentralizada e na interface da aplicação, esses bugs foram repassados à outra equipe que compõe o projeto para que pudessem ser consertados.

### **4.3. PROJETO TWEELI**

Nesta seção serão abordadas quais as atividades realizadas pelo estagiário no projeto Twee.li.

Primeiramente será dada uma breve explicação do que é o Twee.li e como ele se encaixa no contexto da E.life para abordar empresas. Após isso serão mostradas as atividades que o estagiário realizou.

#### **4.3.1. O que é?**

O Twee.li é o primeiro comunicador instantâneo mundial baseado no Twitter. Seu desenvolvimento foi todo realizado pela E.life e já houve uma versão lançada no Brasil, com previsão agora para lançamento da mesma em outros países.

No contexto da E.life, a idéia é oferecer o Twee.li para empresas, a fim de melhorar o relacionamento entre a marca e os consumidores, pois isto poderia ser feito de forma privada e em tempo real.

Seria possível também ter uma sala de bate-papos coletiva para conversas coletivas sobre determinado tema sugerido pela marca. Por exemplo, no contexto da Hellmann's (marca de maionese) poderia convidar chefes de cozinha para um dia de chat com as consumidoras interessadas em receitas.

Além de funcionalidades como as já citadas, também seria possível disponibilizar uma aba com vídeos institucionais sobre a marca e uma aba para realizar pesquisas com os consumidores da marca para realizar estudos com as respostas obtidas. Tudo isso numa só

aplicação que ainda é possível realizar operações básicas de qualquer cliente de Twitter e conversar diretamente com seus amigos.

Nas Figuras 11 e 12, são apresentadas, respectivamente, o Twee.li com interface adaptado para uma empresa (neste caso a Hellmann's) e uma tela mostrando como se dá o chat em grupo (também na versão para a Hellmann's).



Figura 11 - Skin do Twee.li para a Hellmann's.



Figura 12 - Tela de chat em grupo para a empresa

O estagiário foi inserido nesse projeto, para acelerar a adição de novas funcionalidades e correção de alguns problemas que afetavam a eficiência da aplicação, a fim de acelerar a entrega de uma versão do Twee.li para empresas, principalmente para a ZON, empresa de Tv à cabo portuguesa, que já solicitou uma versão do Twee.li para melhorar a interação da mesma com seus consumidores.

#### 4.3.2. Atividades realizadas no projeto

Assim como foi feito no tópico 4.2.2, neste tópico serão mostradas as atividades realizadas pelo usuário na aplicação Twee.li, seguindo a ordem cronológica de atividades.

- **Estudo de tecnologias utilizadas na aplicação**

Para conseguir realizar atividades no projeto Twee.li, o estagiário teve que estudar a tecnologia Adobe AIR, que usa da linguagem ActionScript, e é a ferramenta que é utilizada para fazer desenvolvimentos no Twee.li. Nesta etapa foram realizados estudos sobre como criar interfaces gráficas para usuários, como fazer requisições Web dos tipos HTTP e WebServices, como criar componentes e fazer controle sobre eles e como é a sintaxe da linguagem ActionScript.



Outros pontos que tiveram de ser estudados, foram no que diz respeito à interação de Adobe AIR com a API do Twitter e com SQLite para a persistência de dados da aplicação.

- **Correção de bugs**

No Twee.li cada usuário tem sua lista de contatos, que são ordenados pelo status (online e offline) e também por ordem alfabética. Ao usuário ficar online, a lista de contatos estava ficando inconsistente, não respeitando o critério de ordenação desejado, nesse contexto, o estagiário fez as devidas correções para extinção desse bug.

Outro problema era no que diz respeito à paginação de contatos ao se fazer uma busca. Essa paginação deveria ser recalculada, pois as buscas mostram apenas os 20 primeiros resultados. O estagiário trabalhou nessa tarefa e resolveu também esse problema na paginação.

Na Figura 13, é apresentada a correção em ação, a primeira tela ilustra a tela de contatos normal, antes de se fazer uma pesquisa mostra quatro páginas. Quando o usuário busca por algum termo esse número é reduzido para apenas uma.



Figura 13 - Telas antes de depois fazer uma busca por usuários

- **Melhorias na interface gráfica da aplicação**

Alguns ajustes se fizeram necessários para tornar a aplicação mais atrativa aos olhos de quem a utiliza. Neste ponto, foi requisitado ao estagiário que fizesse o refatoramento na tela de mostrar o perfil de um usuário do Twee.li. Para isso o estagiário usou conceitos de como criar componentes em Adobe AIR e também conseguiu concluir a tarefa de forma satisfatória.

Na Figura 14, é apresentada a tela antes (a) e como ficou após o refatoramento (b), respectivamente. Na nova tela, foi adicionado um botão para deixar de seguir ou passar a seguir determinado usuário, também foram acrescentados ícones para deixar a tela mais amigável e um novo design nas cores de fontes e caixas envolvendo os textos.



Figura 14 - Telas antes e depois do refatoramento

- **Mostrar menções a determinado usuário**

Outra tarefa atribuída ao estagiário foi que o mesmo desenvolvesse uma tela onde fosse possível visualizar as menções feitas a ele através do Twitter com possibilidade de respondê-las utilizando a mesma tela.

Para que esta atividade fosse concluída com êxito foi necessário conhecimentos na biblioteca Tweetr, que é a biblioteca mais utilizada para interação entre Adobe AIR e Twitter.

Para pegar as menções do usuário, foi feito um procedimento para interagir com a API Tweetr, a mesma retorna as menções ao usuário, que, posteriormente, são passadas para outro procedimento que monta a tela.

Na Figura 15, é ilustrado o resultado do trabalho nessa funcionalidade, cada menção ao usuário é mostrada nessa tela, e ao clicar no botão *reply* é possível responder a menção que foi feita.



Figura 15 - Tela de menções do Twee.li

- **Novas abas para a aplicação**

Outra funcionalidade solicitada para a aplicação, é que a mesma além de possuir a aba com os contatos do usuário, também possuísse uma aba mostrando a *Timeline* do mesmo e outra mostrando tweets das empresas cadastradas na aplicação.

A aba da *timeline* deve mostrar os tweets mais recentes de quem o usuário autenticado segue, para a criação da mesma foi utilizado um componente que é mostrado no momento do clique na aba desejada. Para a aba *companies* o procedimento se dá da mesma maneira da *timeline*, porém mostrando tweets das empresas que estão cadastradas na aplicação.



Na Figura 16, é apresentada uma tela do Twee.li com a aba *my timeline* selecionada. Nessa aba é possível que os usuários consigam ver os últimos tweets que chegaram à sua *timeline*.



Figura 16 - Aba *timeline* do Twee.li

- **Tela de log (histórico) de conversas**

Outra funcionalidade solicitada para o Twee.li era que ficasse armazenado localmente as conversas dos usuários, e que pudessem ser mostradas a partir de períodos de tempo (último mês, última semana e dia atual). Para a conclusão desta atividade, foram necessários conhecimento de SQLite, para que fosse possível armazenar as conversas localmente na máquina do usuário da aplicação.

Outro ponto relevante para a conclusão da tarefa foi fazer a interação do banco de dados SQLite com o código da aplicação. Para isso, as consultas certas tiveram que ser feitas para conseguir pegar as conversas de acordo com o período de tempo.

Além disso, para a montagem da tela, novamente foram exigidos conhecimentos de componentes em Adobe AIR, para, a partir daí conseguir mostrar o log de conversas de forma



satisfatória para o usuário. Esta tarefa foi também concluída com sucesso e já está em vigor nas últimas versões lançadas do Twee.li.

Na Figura 17, é ilustrado como ficou a organização da tela de histórico de conversas, que mostra qual o contato, informações sobre o mesmo, que foram adquiridas via Twitter, e as últimas frases que ambos trocaram.



Figura 17 - Log de conversas de usuário

- **Adicionar contatos apenas no contexto da aplicação**

No Twaali uma pessoa é considerada um contato, se no Twitter o usuário segue e também é seguido pela mesma. Foi solicitado que agora pudesse ser adicionado qualquer usuário do Twaali como contato.

Esta foi uma tarefa que também foi solucionada pelo estagiário. Para isso, o mesmo fez alterações tanto na camada cliente, como na camada de servidor da aplicação. Na camada cliente a alteração realizada foi implementar uma requisição ao servidor, para adicionar um determinado contato e ao aceitar determinado contato avisar também ao servidor. Na camada do servidor, foi feito um modo de o usuário não aceitar todas as requisições que chegassem (como era feito), mas sim repassasse esse controle para a aplicação cliente.

Para adicionar um novo contato ao Twee.li, o usuário deve apenas digitar o login de Twitter do contato desejado e pronto, já estará feito. Para aceitar, aparece uma tela perguntando ao usuário se deseja aceitar ou não a solicitação de amizade, sendo necessário apenas um clique num botão, as respectivas ações são realizadas nas telas ilustradas na Figura 18.

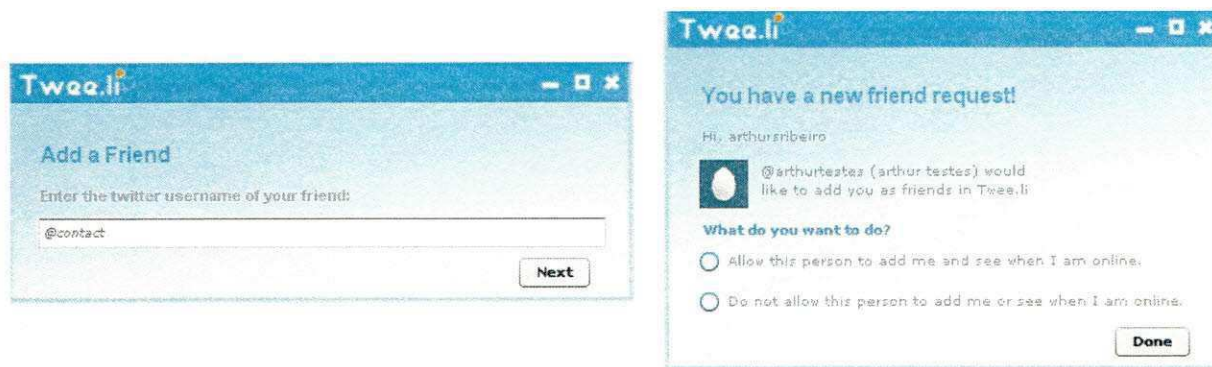


Figura 18 - Telas de adicionar e aceitar contato, respectivamente

- **Verificação e correção de memory leaks**

Um memory leak é um problema computacional em que uma porção de memória, alocada para determinada operação, não é liberada quando não é mais necessária.

No Twee.li existiam dois vazamentos de memória críticos, um se dava nas janelas de chat que eram criadas, porém, não liberava memória ao serem fechadas e a outra se dava na notificação de tweet, que também não liberava memória ao ser fechado.

Esses problemas de alocação de memória precisavam ser resolvidos, principalmente porque os atendentes das empresas necessitarão abrir e fechar várias telas de Chat durante o dia, bem como receber várias notificações de tweet, fazendo com que o consumo de memória apenas crescesse por parte da aplicação.

Para resolução deste problema, o estagiário teve de estudar como o *garbage collector* do Adobe AIR trabalhava. Feito esta etapa, foi necessário aprender como rodar a aplicação em modo Profile, para que fossem encontrados os maiores problemas que desencadeavam em memory leak.

Após executar a aplicação em modo Profile foi descoberto pelo estagiário que os

problemas maiores se davam na adição de imagens as janelas e a comunicação de eventos na aplicação.

Para resolver este problema, o estagiário fez uso de referências fracas para a parte de comunicação de eventos, essas referências fracas podem ser realizadas através da adição de alguns atributos no método de conexão. Para resolução do problema da adição de imagens as janelas, foi feita uma limpeza das mesmas manualmente pelo código da aplicação a cada vez que uma janela é fechada, fazendo assim com que a memória pudesse ser liberada e o consumo da mesma que estava crescendo de forma exponencial ficasse quase que constante.

Essa foi a última atividade do projeto Twee.li delegada ao estagiário, restando assim apenas a preparação do relatório de estágio, bem como de sua defesa.

SEÇÃO V

CONSIDERAÇÕES

FINAIS

## 5. CONSIDERAÇÕES FINAIS

O estágio se mostrou uma oportunidade única para que o estagiário pudesse aprofundar seus conhecimentos adquiridos ao longo da graduação além de que várias tecnologias pudessem ser aprendidas.

Com o estágio, tem-se a oportunidade única de ser inserido diretamente no mercado de trabalho, aprender como se dão as relações, como se dá a vida numa empresa e como aparecem cobranças no dia-a-dia de um funcionário.

Como pontos negativos que puderam ser elencados pelo modelo o qual o estagiário adotou (*Home Office*), estão as dificuldades com relação à infra-estrutura. Algumas vezes o estagiário precisou usar uma internet de velocidade melhor e isso não era disponível em casa. Alguns recursos de *hardware* às vezes eram limitados para conferências via Skype e limitava o andamento ideal de algumas atividades.

Para o curso de ciência da computação da UFCG ter uma melhor eficácia em disciplinas como estágio integrado, deveriam existir disciplinas obrigatórias relacionadas à programação para Web. Na E.life, por exemplo, foram necessários conhecimentos nesta área, que cada vez mais se expande e mais profissionais são solicitados para preencher vagas relacionadas a mesma. Disciplinas como estágio integrado deveriam ser mais incentivadas por parte da coordenação da graduação a fim de tentar preparar melhor os egressos que se interessam por entrar no mercado de trabalho.

Cursar esta disciplina é, no entendimento do estagiário, algo essencial na formação acadêmica de um aluno. Os ganhos adquiridos vão além de conhecimentos sobre tecnologia e mercado de trabalho, na verdade, a disciplina de estágio integrado juntamente com o estágio, preparam o aluno para muitas situações que ele pode viver, podendo assim afirmar que trata-se de uma preparação para toda a vida.



REFERÊNCIAS

BIBLIOGRÁFICAS



MICROSOFT. Windows Xp Home. Disponível em: <<http://www.microsoft.com/Windowsxp/>>. Acesso em: 01 de dezembro de 2010

THE ECLIPSE FOUNDATION. Eclipse IDE. Disponível em: <<http://www.eclipse.org/>>. Acesso em: 01 de dezembro de 2010

ORACLE. MySQL – The most popular open source database. Disponível em: <<http://www.mysql.com>>. Acesso em: 01 de dezembro de 2010

ADOBE SYSTEMS INCORPORATED. Adobe AIR. Disponível em: <<http://www.adobe.com/products/air>>. Acesso em: 01 de dezembro de 2010

ADOBE SYSTEMS INCORPORATED. Adobe Cirrus. Disponível em: <http://labs.adobe.com/technologies/cirrus/>. Acesso em: 01 de dezembro de 2010

ORACLE. JavaServer Pages Technology. Disponível em: <<http://java.sun.com/products/jsp/>>. Acesso em: 01 de dezembro de 2010

SAUVÉ; Jacques Philippe. JSTL – JSP Standard Tag Library. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/jstl/jstl.htm>>. Acesso em: 01 de dezembro de 2010

GSON. A Java library to convert JSON to Java objects and vice-versa. Disponível em: <<http://code.google.com/p/google-gson>>. Acesso em: 01 de dezembro de 2010

THE APACHE SOFTWARE FOUNDATION. Apache Tomcat. Disponível em: <<http://tomcat.apache.org>>. Acesso em: 01 de dezembro de 2010

THE APACHE SOFTWARE FOUNDATION. Commons FileUpload. Disponível em: <<http://commons.apache.org/fileupload/>>. Acesso em: 01 de dezembro de 2010

TWEETR. Tweetr – Na AS3 Twitter API Implementation. Disponível em: <<http://wiki.swfjunkie.com/tweetr>>. Acesso em: 01 de dezembro de 2010.

ORACLE. Core J2EE Patterns – Data Access Object. Disponível em: <<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>>. Acesso em: 02 de dezembro de 2010.

MARCORATTI; José Carlos. Padrões de Projeto: O modelo MVC – Model View Controller. Disponível em: <[http://www.macoratti.net/vbn\\_mvc.htm](http://www.macoratti.net/vbn_mvc.htm)>. Acesso em: 02 de dezembro de 2010.

THE APACHE SOFTWARE FOUNDATION. Apache POI – The Java API for Microsoft Documents. Disponível em: <<http://poi.apache.org/>>. Acesso em: 02 de dezembro de 2010.

MICROSOFT. Word 2010. Disponível em: <<http://office.microsoft.com/pt-br/word/microsoft-word-2010-FX010048798.aspx>>. Acesso em: 02 de dezembro de 2010.

ADOBE SYSTEM INCORPORATED. Flex in a week vídeo training. Disponível em: <<http://www.adobe.com/devnet/flex/videotraining.html>>. Acesso em: 01 de dezembro de 2010.

CAELUM | ENSINO E INOVAÇÃO. Java para desenvolvimento Web. Disponível em: <<http://www.caelum.com.br/curso/fj-21-java-web/>>. Acesso em: 02 de dezembro de 2010.

HARUI; Alex. Using the Flex Builder 3.x Profiler. Disponível em: <[http://blogs.adobe.com/aharui/2008/09/using\\_the\\_flex\\_builder\\_3x\\_prof.html](http://blogs.adobe.com/aharui/2008/09/using_the_flex_builder_3x_prof.html)>. Acesso em: 01 de dezembro de 2010.

LABRIOLA; Michael. Training from the source: Profiling Flex Applications. Disponível em: <<http://insideria.com/2008/06/profiling-flex-applications-sa.html>>. Acesso em 02 de dezembro de 2010.





# APÊNDICES

**APÊNDICE A: PLANO DE ESTÁGIO**

Universidade Federal de Campina Grande - UFCG

Centro de Engenharia Elétrica e Informática - CEEI

Departamento de Sistemas e Computação – DSC

Plano de Estágio

Onibuzz

Arthur de Souza Ribeiro

Curso de Bacharelado em Ciência da Computação

Campina Grande, Agosto de 2010

## **1. Informações Pessoais**

Nome: Arthur de Souza Ribeiro

Matrícula: 20811005

Endereço Residencial: Rua Severino Pimentel, 1731. Bairro: Jardim Paulistano - Campina Grande - PB

Fone: +55 (83) 3331 3478, (83) 8828 5557

E-mail: arthurdesribeiro [at] gmail.com

## **2. Ambiente de Estágio**

O estágio será realizado para a empresa E-LIFE monitor de estudos de mercado LTDA. A E-life trata-se de uma empresa que é líder em monitoração, análise da mídia gerada pelo consumidor e gestão de relacionamento em mídias sociais na América Latina e Portugal.

Dentre seus objetivos principais está entender as empresas a entender dinâmicas das redes sociais e a relação com sua marca e segmento de atuação, entregando projetos que serão utilizados por áreas como Marketing, Comunicação e RP, Atendimento ao Cliente e Inteligência de Mercado. Orientar os clientes com relação à percepção de marca e serviços, comportamento e hábitos de consumo, tendências e relacionamento com consumidor a partir das redes sociais. Em resumo, a E-Life trata-se de uma empresa de inteligência de mercado.

Devido à empresa não possuir sede em Campina Grande as atividades do estágio serão realizadas na minha residência com a supervisão das mesmas sendo realizadas através de reuniões via Skype.

## **3. Supervisão (Técnica e Acadêmica)**

A supervisão acadêmica será realizada pelo professor Cláudio de Souza Baptista que possui experiência na área de Ciência da Computação, com ênfase em Sistemas de Informações e Banco de Dados, atuando principalmente nos seguintes temas: web based systems, geoprocessamento, biblioteca digital, multimedia databases e metadata e é associado a Universidade Federal de Campina Grande sendo professor do departamento de sistemas e computação.

A supervisão técnica será realizada por Alessandro Barbosa de Lima, CEO da empresa.

**Dados do supervisor acadêmico:**

**Nome:** Cláudio de Souza Baptista

**Endereço profissional:** Rua Aprígio Veloso, 882, Bodocongó Bloco CN - Sala 210  
58109970 Campina Grande - PB – Brasil

**Emails:** baptista at dsc.ufcg.edu.br, claudiobaptista at yahoo.com

**Telefone:** 83 3310 1119

**Fax:** 83 3310 1124

**Dados do supervisor técnico**

**Nome:** Alessandro Barbosa de Lima

**Endereço profissional:** E-Life Monitor Estudos de Mercado Ltda Al. Ribeirão Preto, 130, sala 41, Bela Vista. CEP: 01.331-000. São Paulo – SP.

**Email:** albali at elife.com.br

## 4. Resumo do problema

É notório que as redes sociais na internet têm crescido de forma bastante considerável e um exemplo é o Twitter, a ferramenta que aparentemente é bem simples hoje possui um universo de milhões de usuários que trocam mensagens, comentam assuntos que acham interessantes e compartilham links, por exemplo.

Com o Twitter surgiram também aplicações para facilitar o seu uso, aplicações essas denominadas “clientes de Twitter” que auxiliam no uso da ferramenta, pois a partir destas não é necessário ter que acessar a *homepage* da ferramenta para que se possa interagir de forma efetiva com a mesma.

O uso do Twitter tem crescido bastante principalmente no que diz respeito a compartilhamento e leitura de notícias, porém para esta atividade não existe uma aplicação (cliente de Twitter) que a manipule, ou seja, ordenar links por preferencia do usuário, número de acesso, associar a tags e etc.

## 5. Objetivos

Tem-se por objetivo, elicitar requisitos, fazer pesquisas com relação a tecnologias P2P e desenvolver software que resultará na aplicação Onibuzz, que se trata de um cliente de Twitter em que o usuário pode compartilhar links. A princípio os links compartilhados seriam de notícias e a partir deles poderia criar um ranking mediante popularidade, filtrar por tag e adicionar um link a sua lista de links favoritos. Seu grande diferencial é a existência de um

buscador descentralizado, ou seja, a conexão é feita diretamente entre os clientes, tornando assim, a existência do servidor obsoleta para esta finalidade.

Como objetivos específicos para a aplicação têm-se:

- Filtrar notícias de quem o usuário segue e qualquer profundidade de rede (limitado na versão *free*)
- O usuário acessa a aplicação através de seu login e senha do Twitter e tem um resumo de todas as URL's "twittadas" dado um período de tempo.
- O banco de dados de notícias é distribuído através de um protocolo *peer-to-peer*, poupando assim, altos custos com servidores.
- Haverá um servidor para registrar o nome do usuário/localização no Twitter e enviar um anúncio quando requisitado. Porém, mesmo que o servidor não responda, a aplicação deve continuar trabalhando.
- As notícias devem aparecer ranqueadas por popularidade
- Podem-se realizar operações básicas do Twitter entre usuários.
- O usuário pode "retwittar" notícias que sempre mencionarão Via @Onibuzz
- O usuário pode associar tags a notícias e também adicioná-las a sua lista de notícias favoritas. Esses dados serão utilizados no mecanismo de busca e na elaboração do ranking.
- Utilizar Adobe Flex (<http://www.adobe.com/br/products/flex/>) para desenvolvimento da aplicação.
- O conjunto de tweets com links que são alterados pelo usuário via o Onibuzz, são armazenados na máquina do cliente com SQL Lite.

## 6. Metodologia

A meta é trabalhar 4 horas diárias (totalizando 20 horas semanais) a fim de cumprir pequenas iterações, ao fim destas um artefato (parte do software) é gerado para análise. Semanalmente haverá acompanhamento via Skype para definir metas e acompanhar o trabalho que está sendo realizado. O software é desenvolvido em equipe, ficando cada integrante responsável por finalizar uma parte do sistema, partes essas definidas em reunião.

## 7. Atividades Planejadas

ID	Atividade	Tempo
T1	Estudo das tecnologias a serem utilizadas	40 horas
T2	Modelar tabelas do Banco de Dados	20 horas
T3	Criar classes de interação com Banco de Dados no servidor	20 horas
T4	Criação dos serviços a serem providos pelo servidor	40 horas
T5	Reuniões semanais com supervisor técnico e acadêmico	35 horas
T6	Desenvolvimentos a serem efetuados na aplicação cliente	50 horas
T7	Evolução do sistema e elicitação e implementação de novos requisitos	50 horas
T8	Escrita do relatório de estágio	50 horas
T9	Preparação e apresentação da defesa do estágio	20 horas
Total		325 horas

## 8. Resultados esperados

Espera-se produzir uma aplicação que cumpra com os objetivos traçados inclusive podendo aceitar um considerável número de usuários sendo esses tanto *free* como *corporate*.

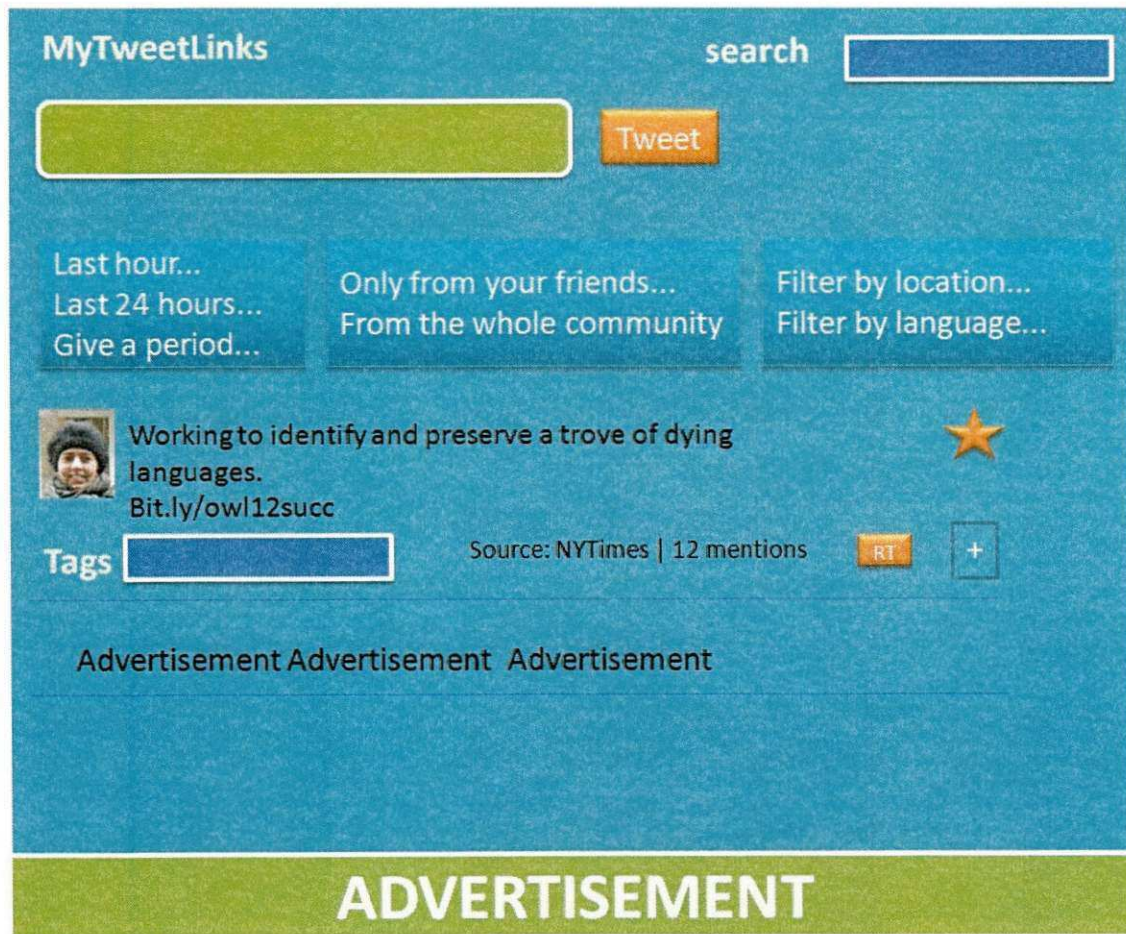
## 9. Cronograma

O cronograma a ser seguido para o desenvolvimento das atividades descritas no tópico 7 é o seguinte:

	Agosto				Setembro				Outubro				Novembro				Dezembro				
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17				
T1	■																				
T2		■																			
T3			■																		
T4				■	■	■	■														
T5	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
T6								■	■	■	■	■	■	■	■	■	■				
T7												■	■	■	■	■	■				
T8																	■	■	■	■	
T9																		■	■	■	

## 10. Anexos

- Prototipação de tela da aplicação:

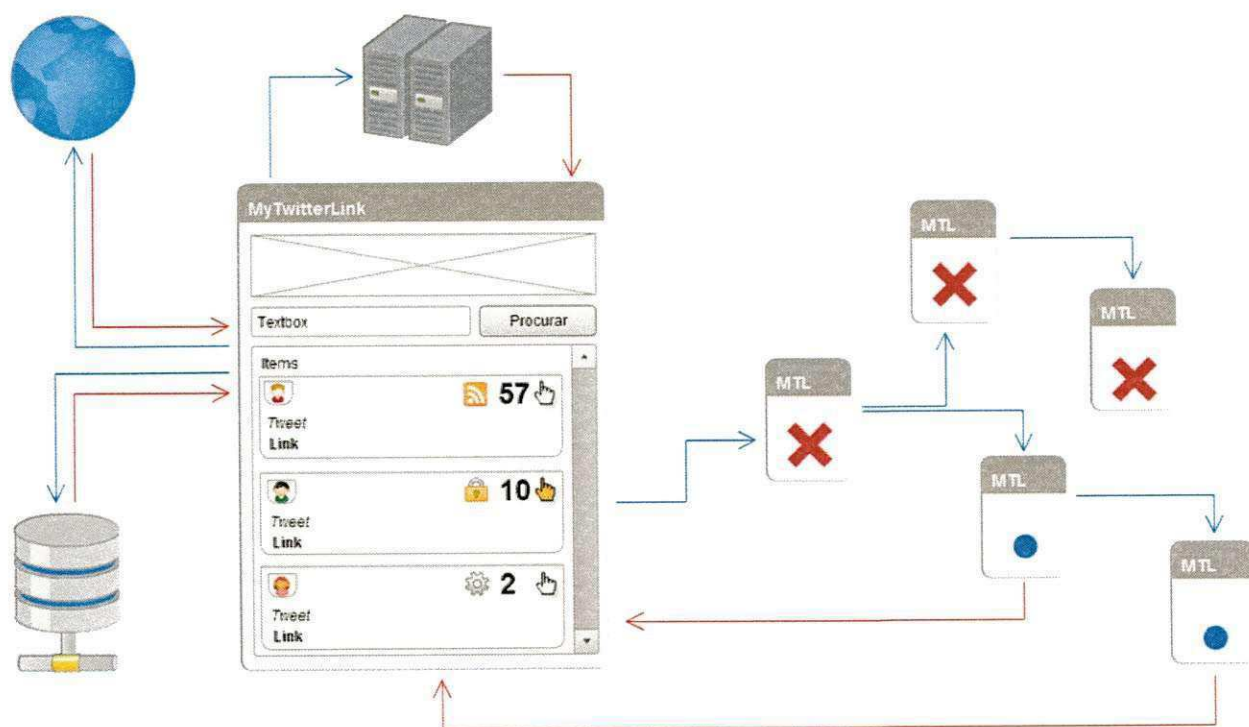




- Mockup da aplicação



- **Arquitetura do Sistema**



## 11. Aprovação

Declaro para os devidos fins que aprovo o planejamento das atividades descritas neste documento como plano de estágio do aluno Arthur de Souza Ribeiro, matrícula 20811005.

---

**Cláudio de Souza Baptista**

Supervisor Acadêmico

---

**Alessandro Barbosa Lima**

Supervisor Técnico

---

**Joseana Macêdo Fchine**

Coordenador da Disciplina Estágio Integrado