

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE SOFTWARE

FABRICIO JHONATA SOUSA DE OLIVEIRA

**UM MODELO PARA A CRIAÇÃO, DIVULGAÇÃO E
MONETIZAÇÃO DE CONTEÚDOS DINÂMICOS COM ÊNFASE EM
VIRALIZAÇÕES**

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2017

FABRICIO JHONATA SOUSA DE OLIVEIRA

**UM MODELO PARA A CRIAÇÃO, DIVULGAÇÃO E
MONETIZAÇÃO DE CONTEÚDOS DINÂMICOS COM ÊNFASE EM
VIRALIZAÇÕES**

Trabalho de conclusão de curso de graduação apresentado à disciplina Trabalho de Conclusão de Curso 1, do Curso Superior de Engenharia de Software da Universidade Tecnológica Federal do Paraná, Câmpus Cornélio Procópio como avaliação para obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Me. Diogo Cezar Teixeira Batista

CORNÉLIO PROCÓPIO

2017

RESUMO

OLIVEIRA, Fabricio. Um modelo para a criação, divulgação e monetização de conteúdos dinâmicos com ênfase em viralizações. 44 f. Proposta de Trabalho de Conclusão de Curso – Engenharia De Software, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Palavras-chave: monetização, modelo, conteúdo dinâmico, facebook api, graph api, laravel, php, viral

ABSTRACT

OLIVEIRA, Fabricio. A model for the creation, disclosure and monetization of dynamic content with an emphasis on viralization. 44 f. Proposta de Trabalho de Conclusão de Curso – Engenharia De Software, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Keywords: monetization, model, dynamic content, facebook api, graph api, laravel, php, viral

LISTA DE FIGURAS

FIGURA 8	– Exemplo de um viral	10
FIGURA 9	– Exemplo de uma imagem gerada após a interação com o usuário	11
FIGURA 11	– Exemplo da autenticação com o Facebook Login	15
FIGURA 12	– Exemplo do explorador da <i>Graph API</i>	16
FIGURA 13	– Facebook Pages - Relatório - Resumo da Página	17
FIGURA 14	– Facebook Pages - Relatório - Pessoas	18
FIGURA 16	– Caso de uso geral do gerenciador de conteúdo	21
FIGURA 17	– Exemplo de um resultado de um aplicativo	26
FIGURA 18	– Protótipo de tela da inserção dos dados básicos	26
FIGURA 19	– Protótipo de tela da inserção dos fundos do aplicativo	27
FIGURA 20	– Protótipo de tela da inserção das fotos do usuário	28
FIGURA 21	– Protótipo de tela da lista de Grupos de Textos inseridos	28
FIGURA 22	– Protótipo de tela da inserção dos Grupos de Textos	29
FIGURA 23	– Caso de uso geral do site	31
FIGURA 24	– Protótipo de tela da visualização do aplicativo	32
FIGURA 25	– Protótipo de tela da visualização do resultado gerado	33
FIGURA 26	– Visão geral da arquitetura do sistema	38
FIGURA 27	– Arquitetura cliente-servidor	38
FIGURA 28	– Exemplo do funcionamento do ELB	39

LISTA DE TABELAS

TABELA 9	– Tabela comparativa dos trabalhos relacionados	20
TABELA 11	– Atributos do aplicativo	22
TABELA 12	– Componentes que podem ser inseridos em um aplicativo	23
TABELA 13	– Componente de Fundos.	23
TABELA 14	– Componente de Fotos.	23
TABELA 15	– Componente Grupo de Textos.	24
TABELA 16	– Parte do componente Grupo de Textos.	24
TABELA 17	– Componente de Fotos.	25
TABELA 18	– Parte do componente fundos e dos textos do grupo de textos.	25

LISTA DE SIGLAS

MVC	Model-View-Controller
ORM	Object Relational Mapping
PSR	PHP Standards Recommendations
SGBD	Sistema de Gerenciamento de Banco de Dados
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
ELB	Elastic Load Balancing
RDS	Relational Database Service
GB	Gigabyte
S3	Simple Storage Service
CDN	Content Delivery Network

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	12
1.2	ORGANIZAÇÃO DO TEXTO	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	WEB 2.0	13
2.2	FACEBOOK	13
2.2.1	FACEBOOK LOGIN	14
2.2.2	FACEBOOK GRAPH API	15
2.2.3	FACEBOOK PAGES	16
2.3	MARKETING VIRAL	18
2.4	TRABALHOS RELACIONADOS	19
3	PROPOSTA	21
3.1	SISTEMA WEB	21
3.1.1	GERENCIADOR DE CONTEÚDO	21
3.1.1.1	INSERÇÃO DE CONTEÚDO	25
3.1.2	SITE	31
3.1.2.1	CONSTRUÇÃO DA IMAGEM	31
3.2	TECNOLOGIAS E FERRAMENTAS	33
3.2.1	PHP	33
3.2.2	LARAVEL	34
3.2.3	MYSQL	35
3.2.4	REDIS	36
3.2.5	INTERFACE WEB	36
3.2.6	AMAZON WEB SERVICES	36
3.3	SCRUM SOLO	37
3.4	ARQUITETURA DO SISTEMA	37
4	CRONOGRAMA	42
	REFERÊNCIAS	43

1 INTRODUÇÃO

O termo *Web 2.0* é usado para descrever, além de outras coisas, uma *Internet* colaborativa. Na qual os usuários podem interagir a partir de serviços online. Esse modelo colaborou para o surgimento de diversos produtos e serviços digitais que contém funcionalidades para auxiliar essas interações (O'REILLY, 2007).

Como exemplos destes serviços online, pode-se destacar a utilização de aplicativos *online* que atualmente fazem parte do dia-a-dia da maioria das pessoas. Seria difícil imaginar passar um único dia sem a utilização do *Facebook*, *YouTube* ou *Instagram*.

O *Facebook* é uma rede social que permite que seus usuários tenham um perfil para compartilhar suas informações, como: *status* de relacionamento, cidade atual, emprego atual, formações, fotos, vídeos, textos, entre outros. Atualmente o *Facebook* conta com dois bilhões de usuários ativos (FACEBOOK, 2017a). Já o *Twitter* é uma rede social que funciona como um *microblog*, no qual os seus usuários podem criar postagens de até 140 caracteres. Atualmente o *Twitter* conta com 328 milhões de usuários em todo o mundo (TWITTER, 2017). O *Instagram* é uma rede social que permite que seus usuários compartilhem fotos e atualmente possui 400 milhões de usuários ativos (INSTAGRAM, 2017). O *YouTube* é uma rede social que fornece funcionalidades para que seus usuários compartilhem e assistam vídeos. Atualmente um bilhão de horas em vídeos são assistidos por dia no *YouTube* (YOUTUBE, 2017).

No *Facebook*, objeto de estudo deste trabalho, destaca-se que, quando um usuário acessa sua conta, ele poderia ter acesso a 1.500 publicações relevantes para o seu perfil, mas isso não acontece, pois o *Facebook* utiliza algoritmos para filtrar e classificar o que, de acordo com essa inteligência, é o mais relevante para sua *timeline*. Desta forma, nem tudo que é publicado nessas plataformas é mostrado para todos os seus seguidores e amigos (BOLAND, 2014).

Alguns usuários destas redes sociais tem em suas contas milhares ou até milhões de seguidores que acompanham suas publicações. Portanto o conteúdo publicado por esses usuários normalmente tem um grande **alcance**. Pode-se definir este alcance como: numero de pessoas

que viram uma publicação. Publicações com um grande alcance tem maior possibilidade de ter um efeito *viral*. (SOCIALBAKERS, 2017c, 2017b, 2017a)

Um *viral* funciona como uma epidemia, passando de pessoa para pessoa. Agindo da seguinte forma, um usuário publica um conteúdo em sua conta, seus amigos ou seguidores ao notar o conteúdo publicado, quando há um alto índice de identificação, desejam demonstrar a sua empatia com o conteúdo em questão, compartilhando novamente em suas contas, e assim por diante, fazendo com que o acesso ao conteúdo seja exponencial. Caso o conteúdo seja monetizado, será possível gerar uma grande receita. (CHU, 2011).



Figura 8: Exemplo do funcionamento da viralização de um conteúdo

Fonte: (Marc's Blog, 2013)

O *YouTube* oferece para os seus usuários, que atuam como criadores de conteúdo, uma forma simples e integrada de monetizar os seus vídeos. Para monetizar os conteúdos, o *YouTube* inclui anúncios nos vídeos, que gera uma receita sobre essa publicação e então repassa parte dos ganhos obtidos para o usuário que publicou o conteúdo. Para que o usuário habilite a monetização é necessário que o mesmo tenha um total de 10 mil visualizações em seus vídeos publicados. (BARDIN, 2017).

Algumas redes sociais como por exemplo o *Facebook*, ainda não oferecem uma forma integrada para que os criadores de conteúdo consigam monetizar suas publicações (FACEBOOK, 2016).

Uma das alternativas encontradas para monetizar contas em redes sociais, é enviar os seus seguidores para fora da rede social através de *links* (HABER, 2015). Porém, para que esse *link* seja criado é necessário a utilização de alguma ferramenta externa a rede social, que gere uma aplicação no formato *Web*. Nesse sentido, para usuários sem tanta experiência, a criação

de uma fonte externa pode ser mais complicada.

Dado este cenário, a proposta desse trabalho é desenvolver um sistema, que ofereça funcionalidades para a criação e divulgação de conteúdos através de *links*. O conteúdo a ser criado, no escopo deste trabalho, limita-se a uma imagem que interage com o usuário que a acessa. Para que essa interação seja atingida, o usuário deverá se autenticar com seu *Facebook* no sistema proposto, com objetivo de obter-se dados como: nome, idade, fotos, lista de amigos e outros. Com estes dados será possível gerar uma imagem personalizada como mostra a Figura 9, que pode-se observar que o nome do usuário e sua foto foram utilizadas para criar uma imagem atrativa e lúdica, com o objetivo de instigar o compartilhamento desta imagem entre seus amigos.



Figura 9: Exemplo de uma imagem construída após a interação com o usuário

Fonte: Autoria própria

Espera-se que com a utilização da ferramenta proposta, haja uma maior identificação por parte dos usuários, aumentando a rede de compartilhamentos deste conteúdo, e podendo chegar ao efeito *viral*, descrito em detalhes no capítulo de fundamentação teórica (2.3). O sistema deverá oferecer funcionalidades para que o criador de conteúdo consiga gerar *links* dos conteúdos que o mesmo criou, para realizar a divulgação em suas redes sociais. Todo conteúdo criado deverá incluir anúncios, para que os acessos aos conteúdos gerem receita, garantindo que haja monetização dos conteúdos criados dentro do sistema.

1.1 OBJETIVOS

Desenvolver um sistema que ofereça funcionalidades para criação e divulgação de conteúdos dinâmicos integrados com anúncios. Para alcançar o objetivo do trabalho, será necessário realizar os seguintes objetivos específicos:

- Desenvolver uma plataforma para a criação e publicação de conteúdos;
- Desenvolver uma ferramenta para geração de *links* para a divulgação;
- Permitir que os usuários realizem autenticação e integração com o *Facebook* para que o sistema consiga acessar os seus dados;
- Disponibilizar espaços no *layout* do sistema para a inclusão de anúncios para que seja possível monetizar os conteúdos;
- Desenvolver uma arquitetura que possibilite o sistema ser escalável, e assim suportar diversos usuários online simultaneamente.

1.2 ORGANIZAÇÃO DO TEXTO

Este trabalho está estruturada da seguinte forma: no Capítulo 2 são definidos os principais conceitos que serão utilizados para o desenvolvimento do sistema, e uma comparação dos trabalhos relacionados. No Capítulo 3 descreve-se a proposta deste trabalho. No Capítulo 4 estão descritas as tecnologias, ferramentas e metodologia que serão utilizadas para o desenvolvimento desta proposta. Por fim, o Capítulo 5 que descreve o cronograma a ser seguido para o desenvolvimento desta proposta.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 WEB 2.0

Uma das definições de *Web 2.0* é descrita por (O'REILLY, 2007), que define que a partir da *Web 2.0* a *Internet* passou ser considerada uma plataforma social. Pois a partir da *Web 2.0* os serviços começaram a permitir que os usuários não apenas o consumam informações, mas também criem o seus próprios conteúdos, dando origem a esta plataforma social. (SHUEN, 2008).

A *Web 2.0* remete a um novo formato de ferramentas e serviços, que permitem aos usuários publicar informações na *web*, já que anteriormente essa prática era feita apenas por empresas, profissionais da área e entusiastas. Portanto, a principal característica das funcionalidades que surgiram com a *Web 2.0*, foi a possibilidade de serem utilizadas para compartilhar informações na *web*, como por exemplo, comentar em uma postagem, compartilhar sua opinião em uma rede social, entre outros. Esses conceitos foram a base para o surgimento de *blogs*, redes sociais, entre outros. As principais características da *Web 2.0* são a identidade, presença, comunicação, relacionamento e troca de informações, na qual, a finalidade é aumentar e estimular a participação dos usuários na *Internet*, fazendo com que os serviços *web* se tornem cada vez melhores quanto mais usados por seus usuários. (COOKE; BUCKLEY, 2008)

2.2 FACEBOOK

Com surgimento da *Web 2.0*, os avanços tecnológicos, o surgimento dos dispositivos móveis, e preços acessíveis a *Internet* de alta velocidade, as redes sociais se tornaram parte do dia-a-dia de muitos indivíduos. As redes sociais são *sites* que contém ferramentas para que seus usuários se comuniquem, compartilhem conhecimentos, ideias, interesses e informações pessoais como: fotos, vídeos, relacionamentos, e outros. As redes sociais são uma das grandes mídias de influencia da atualidade, já que os seus usuários além de leitores, são também os produtores de conteúdo (ABEDNIYA; MAHMOUEI, 2010; TRUSOV et al., 2009).

O crescimento acelerado das redes sociais, resultou na facilidade para a obtenção de dados dos usuários, fazendo com que seja possível a criação de produtos e serviços personalizados, como por exemplo, verificar se um usuário cursa o ensino superior e oferecer ou não uma conta-corrente universitária para o mesmo (HARTLINE et al., 2008).

O *Facebook* é uma rede social que foi lançada em Fevereiro de 2004 e atualmente conta com 2 bilhões de usuários. No início foi disponibilizada apenas para alunos da Universidade de Harvard, localizada em Massachusetts nos Estados Unidos, mas em 2006 tornou-se disponível para todo o mundo, sendo necessário apenas um e-mail válido e idade superior a 13 anos. O *Facebook* permite que seus usuários compartilhem suas informações pessoais como por exemplo: idade, cidade atual, fotos, textos, crenças e outros, também disponibiliza um recurso para troca de mensagens em tempo real, para que seus usuários possam se comunicar de forma instantânea com os seus amigos. Além disso, o *Facebook* possui muitos outros recursos, que não são relevantes no escopo deste trabalho. (FACEBOOK, 2017b).

2.2.1 FACEBOOK LOGIN

O *Facebook Login*³ facilita o processo de autenticação em diversos tipos de aplicações, como por exemplo, aplicativos nativos *Android*, *IOS* e também para aplicações *web*. Uma de suas funcionalidades mais atrativas, além da praticidade do *login* através de um clique, são seus diversos recursos de segurança, que auxiliam os usuários a controlar quais dados eles desejam compartilhar. Na Figura 11 contém um exemplo de como é feito esse auxílio, na qual, ele permite que os usuários selecionem o que eles desejam ou não compartilhar com o serviço que se esta autenticando.

O *Facebook Login* é um serviço que utiliza o *Auth 2.0* para autenticação e autorização com o uso do *Login Dialog* (Janela de diálogo que permite ao usuário se autenticar e autorizar o que a aplicação pode acessar ou não acessar, como na Figura 11). Após o usuário realizar a autenticação, o *Facebook Login* irá retornar as permissões que foram aprovadas pelo usuário e um *token* de acesso. Com esse *token* será possível consultar e interagir com os dados desse usuário, através da *Graph API*⁴, que será detalhada no próximo tópico.

³<https://developers.facebook.com/docs/facebook-login/>

⁴*Auth 2.0* é um protocolo aberto para permitir a autorização segura, num método simples e padronizado, a aplicações *web*, móveis e *desktop*

⁴<https://developers.facebook.com/docs/graph-api/overview/>



Figura 11: Exemplo de autenticação com o uso do *Facebook Login*³. No lado esquerdo é a tela inicial, já do lado direito é após acessar a opção "editar isso", onde é possível selecionar quais dados ou permissões é permitido que a aplicação acesse.

Fonte: Autoria própria

2.2.2 FACEBOOK GRAPH API

A *Graph API*⁴ é a principal forma de incluir e excluir dados na plataforma do *Facebook*. A *Graph API* é uma API Restful (baseada em HTTP), com isso é possível funcionar em qualquer linguagem que tenha a possibilidade de enviar requisições HTTP, ou até mesmo a partir de um navegador web. Com a *Graph API* é possível consultar dados, realizar publicações, gerenciar anúncios, publicar fotos entre outros. Com a ferramenta "Explorador da *Graph API*", é possível inserir ou gerar um token, informar quais dados que sejam retornados e visualizá-los em JSON, como mostra a figura 12. Essa API será utilizada para o desenvolvimento dessa proposta, para que seja possível acessar os dados dos usuários que estão interagindo com a aplicação web.

⁴<https://developers.facebook.com/docs/graph-api/overview/>

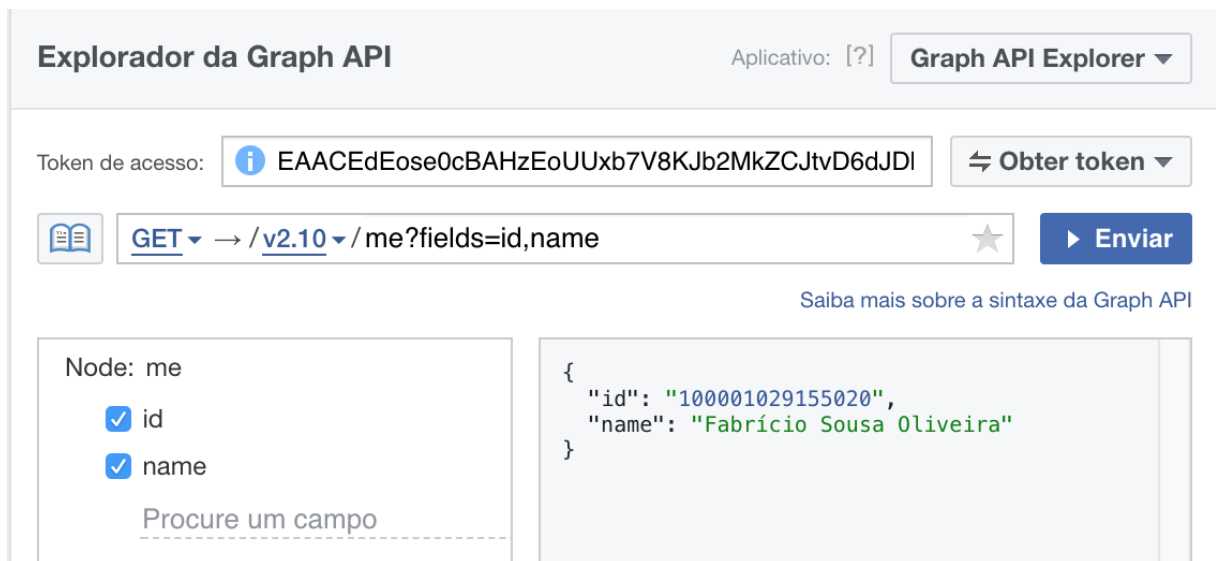


Figura 12: Exemplo do explorador da *Graph API*

Fonte: Autoria própria

2.2.3 FACEBOOK PAGES

Um recurso disponibilizado pelo *Facebook*, e de grande importância no contexto deste trabalho é a ferramenta *Facebook Pages*, com ela é possível que um usuário ou empresa desenvolva páginas personalizadas dentro do *Facebook*. Imagine que todos os usuários que estão utilizando a plataforma, ganham de brinde a possibilidade de criarem seu próprio "espaço" dentro da rede social. Com uma interface intuitiva, e simples, na qual qualquer pessoa consegue rapidamente criar páginas personalizadas que podem ser classificadas por diversas categorias como por exemplo, negócio ou local, empresa, organizações ou instituições, marca ou produto, artista, banda, entre outros. Essa ferramenta permite a inserção de imagens, vídeos, textos, *links* e outros.

Também é possível que os usuários do *Facebook* sigam a página, para que eles possam acompanhar suas publicações. Além disso, o *Facebook Pages* permite ser gerenciado por diversos usuários, fazendo com que a administração de uma página seja feita por uma equipe (FACEBOOK, 2017b). O *Facebook Pages* permite acessar as estatísticas dos usuários que seguem sua página, com esses dados é possível analisar a faixa-etária, gênero, país, cidade e idioma como mostra a Figura 13. Também é possível ter uma visão geral da sua página como visualizações, curtidas, alcance, envolvimento e outros como mostra a Figura 14.



Figura 13: Relatório gerado na ferramenta Facebook Pages: Visão geral

Fonte: Autoria própria (2017), a partir da análise da página Otariano

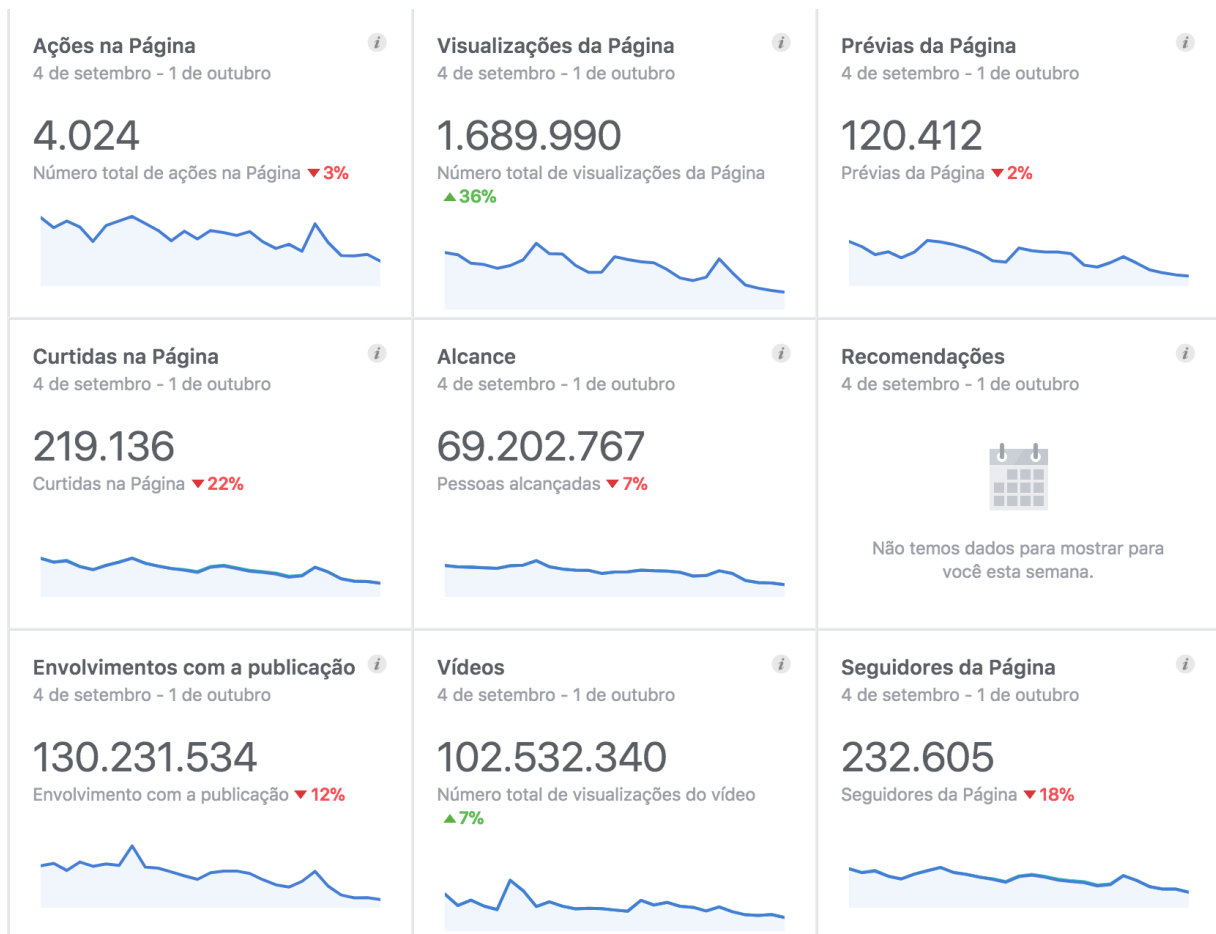


Figura 14: Relatório gerado na ferramenta Facebook Pages: Seguidores da Página

Fonte: Autoria própria (2017), a partir da análise da página Otariano²

Páginas com milhares seguidores conseguem alcançar milhões de pessoas mensalmente. A Figura 14 mostra o alcance da página Otariano² de 69 milhões entre 4 de setembro a 1 de outubro. Com todo esse alcance, identificou-se a possibilidade da criação de diversos conteúdos virais (que serão explicados na seção 2.3) ou enviar um grande número de usuários para um *site*, blog ou qualquer serviço *web* que seja acessível a partir de uma conexão com a internet e um acesso por um *link* externo ao Facebook.

2.3 MARKETING VIRAL

O termo *Marketing Viral* é usado para descrever estratégias para encorajar usuários de redes sociais a disseminar um conteúdo, fazendo com que a exposição e influência do mesmo seja exponencial (CAMARERO; San José, 2011).

²<https://www.facebook.com/humotariano/>

O marketing viral vem sendo utilizado nas redes sociais, por empresas e organizações que tem como objetivo alcançar e atrair usuários para suas peças publicitárias. A palavra *marketing* remete a ideia de divulgar uma mensagem ou produto. Já a palavra *viral* faz uma analogia com a transmissão de um vírus, que é passada de pessoa pra pessoa de forma rápida de forma exponencial (BARICHELO; OLIVEIRA, 2010). Com o uso do Marketing Viral é possível transformar a divulgação de um conteúdo em uma epidemia, pois alguns conteúdos são como vírus, possuem a capacidade de contagiar os usuários. Com o uso das ferramentas de publicação e compartilhamento de conteúdo que as redes sociais disponibilizam, essa epidemia pode se espalhar de forma mais simples e rápida (PEROTTONI et al., 2010; CASTRO, 2012).

Porém a viralização é um fenômeno imprevisível, mesmo com a criação de conteúdos focados para se tornar um viral, e com o uso de ferramentas que auxiliam sua criação, não é possível garanti-la. Criar uma campanha que tome grandes proporções não é algo simples, pois o sucesso do viral depende exclusivamente do público, "Quanto mais envolvente e interessante for a mensagem, maiores são as chances de o anúncio se espalhar com sucesso e maior será a rede de contatos criada por ele para reenvio da mensagem". Uma das formas de aumentar as chances de um conteúdo viralizar é facilitar o processo de compartilhamento. Caso o conteúdo não esteja publicado em uma rede social, como é o caso de um link de um blog, pode ser necessária a utilização de alguma ferramenta que divulguem o conteúdo em questão, como por exemplo, incluir o botão de compartilhar no *Facebook* junto ao conteúdo. *Marketing Viral* é uma forma de divulgação com um grande alcance e de baixo custo, com a entrega rápida da mensagem desejada para milhares de usuários. (BARICHELO; OLIVEIRA, 2010).

2.4 TRABALHOS RELACIONADOS

Nessa seção será apresentado os trabalhos relacionados com essa proposta. Para isso vamos apresentar uma tabela de comparações entre as ferramentas encontradas. Será comparado se há possibilidade de um criador de conteúdo criar conteúdos monetizados e os tipos de conteúdos que podem ser publicados nas ferramentas relacionadas. As ferramentas são o Vonvon¹, Namtests², Zip network³, Blasting News⁴, como mostra a tabela abaixo:

¹<http://br.vonvon.me/>

²<https://pt.nametests.com/>

³<http://zip.network/>

⁴<http://www.blastingnews.com/>

Tabela 9: Tabela comparativa dos trabalhos relacionados

	Possibilidade de criar conteúdos monetizados?	Tipo de Conteúdos
Vonvon	Não	Conteúdos Dinâmicos
Nametests	Não	Conteúdos Dinâmicos
Zip.network	Sim	Artigos e Notícias
Blasting News	Sim	Artigos e Notícias

3 PROPOSTA

Este capítulo apresenta a proposta de desenvolvimento de um sistema *web*, capaz de auxiliar na criação e divulgação de conteúdos que permitem a interação com o usuário que o acessa, resultando em uma imagem personalizada, a partir dos dados do Facebook do usuário. Para facilitar a compreensão utilizaremos o nome de **aplicativo** para definir este tipo de conteúdo.

3.1 SISTEMA WEB

O sistema proposto é dividido em duas partes, que são o site e gerenciador de conteúdo.

3.1.1 GERENCIADOR DE CONTEÚDO

O gerenciador de conteúdo será a área onde os criadores de conteúdo iram inserir os aplicativos. O diagrama de caso de uso geral abaixo, mostra as funcionalidades que devem conter no gerenciador de conteúdo.

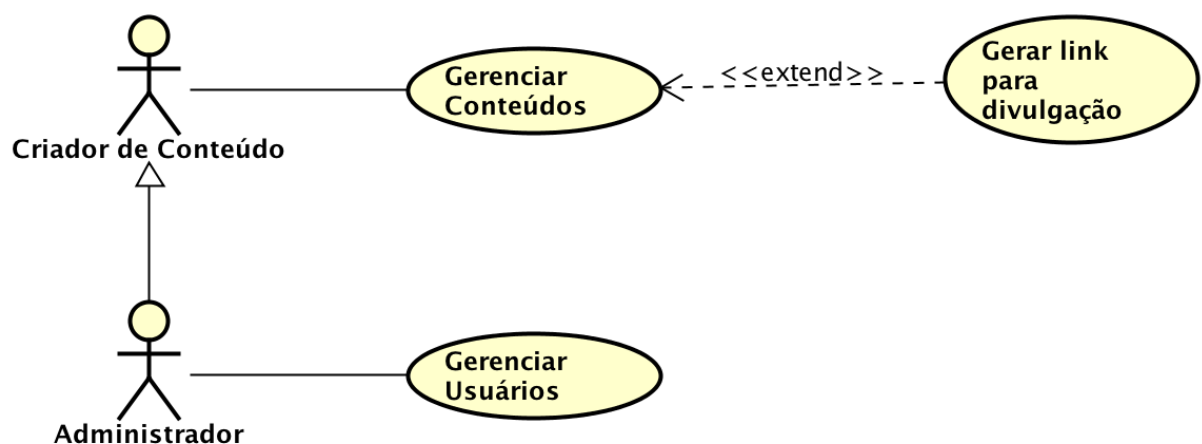


Figura 16: Caso de uso geral do gerenciador de conteúdo.

Fonte: Autoria própria

Para isso os sistema deve conter os seguintes requisitos funcionais:

- **RF1** O sistema deve permitir níveis de acesso, conforme o papel especificado. Cada usuário terá a opção de ter um papel entre dois, que serão dois: administradores ou criadores de conteúdo.
- **RF2** Os administradores iram realizar todas as atividades que um criador de conteúdo possa realizar;
- **RF3** O sistema deve permitir que criadores de conteúdos insiram aplicativos;
- **RF4** O sistema deve permitir que os administradores possam gerenciar todos os usuários;
- **RF5** O sistema deve permitir que os criadores de conteúdo consigam gerar um link para a divulgação.

As Tabelas abaixo mostram quais campos devem conter no gerenciador de conteúdo para auxiliar na criação de uma aplicativo.

Atributos do Aplicativo

Atributo	Tipo	Descrição
Titulo	Texto	Esse atributo será usado na visualização e listagem do aplicativo
Texto	Texto	Esse atributo será usado na visualização e listagem do aplicativo
Imagem de Listagem	Imagem	Essa atributo será usada na listagem dos aplicativos
Imagem de Chamada	Imagem	Esse atributo será usado na visualização do aplicativo
Componentes	Componentes do Aplicativo	Existem diversos componentes que são descritos na tabela “Componentes do Aplicativo”

Tabela 11: Atributos do aplicativo

A tabela 11 mostra os campos básicos para a inserção de um aplicativo, a tabela 12 irá mostrar quais componentes poderão ser inseridos no aplicativos.

Componentes do Aplicativo

Nome do Componente	Descrição
Fundos	Fundos do aplicativo que será selecionado de forma randômica
Fotos	Fotos do usuário que irá interagir com o aplicativo
Grupo de Textos	Textos que será incluso no aplicativo de forma randômica

Tabela 12: Componentes que podem ser inseridos em um aplicativo

Para mostrar em detalhes os atributos de cada componentes foi criado foi criado uma tabela para cada componente, como mostra as figuras abaixo.

Fundos

Atributo	Tipo
Imagem	Imagem
Filtro	Filtro

Tabela 13: Componente de Fundos.

O componente de fundo permite a inserção de diversos fundos, sendo necessário pelo menos um fundo por aplicativo, a tabela 13 mostra os campos necessários para a inserção de um fundo. Os fundos devem variar de forma randômica como será mostrado posteriormente.

Fotos

Atributo	Tipo
Altura	Imagem
Largura	Filtro
Posição Vertical	Inteiro
Posição Horizontal	Inteiro
Índice	Inteiro
Opacidade	Inteiro (0 - 100)
Fundo?	booleano
Escala Cinza?	booleano
Arredondar Foto?	booleano

Tabela 14: Componente de Fotos.

O componente de fotos permite a inserção de diversas fotos do usuário, a tabela 14 mostra os campos necessários para a inserção de uma foto. Todas as fotos inseridas devem ser inclusas no aplicativo, como será mostrado posteriormente.

Grupo de Texto

Atributo	Tipo
Altura	Imagem
Largura	Filtro
Posição Vertical	Inteiro
Posição Horizontal	Inteiro
Fonte	Tipografia
Tamanho da Fonte	Inteiro
Cor da Fonte	hexadecimal de cor
Estilo da Fonte	Maiscula, minuscule, Primeira maiuscula
Alinhamento Vertical do Texto	Topo, Centro, Rodapé
Alinhamento Horizontal do Texto	Esquerda, Centro, Direita
Textos	Textos

Tabela 15: Componente Grupo de Textos.

Textos

Atributo	Tipo
Texto	Texto
Filtro	Filtro

Tabela 16: Parte do componente Grupo de Textos.

O Grupo de Textos permite a inserção de diversos grupos de texto em um aplicativo, cada grupo de texto podem conter diversos textos, sendo necessário ao menos um texto por grupo. A tabela 15 e 17 mostra os campos necessários para a inserção de uma Grupo de Texto. Todas os Grupos de Textos inseridos devem ser inclusos no aplicativo, como será mostrado posteriormente.

Fotos

Atributo	Tipo
Altura	Imagem
Largura	Filtro
Posição Vertical	Inteiro
Posição Horizontal	Inteiro
Índice	Inteiro
Opacidade	Inteiro (0 - 100)
Fundo?	booleano
Escala Cinza?	booleano
Arredondar Foto?	booleano

Tabela 17: Componente de Fotos.

Os componentes Fundos e os Textos do Grupo de Textos tem a opção de incluir um filtro por item inserido, a tabela 18 mostra os atributos que podem ser inseridos em um filtro.

Filtro

Atributo	Tipo
Sexo	Masculino, Feminino
Signo	Lista de Signos do zodíaco

Tabela 18: Parte do componente fundos e dos textos do grupo de textos.

3.1.1.1 INSERÇÃO DE CONTEÚDO

Nessa subseção será mostrado os protótipos de tela do gerenciador de conteúdo, na área de inserção de conteúdo. O protótipo será apresentado simulando a inserção de um aplicativo idêntico ao mostrado na Figura 17.



Figura 17: Exemplo de um resultado de um aplicativo

Fonte: Autoria própria

A Figura 18, mostra o protótipo de tela para a inserção dos dados básicos do aplicativo, que são o título, texto, imagem de listagem e imagem de chamada. No exemplo abaixo foi simulado a inserção do aplicativo chamado "6 características suas que são marcantes!", esse aplicativo será utilizado em toda essa subseção para exemplificar a inserção de conteúdo.

Título <input type="text" value="6 características suas que são marcantes!"/>	
Texto* <input type="text" value="Nossas características nos descrevem, e essas são as minhas características!"/>	
Imagem destaque * Formato: 600x400 	Imagem chamada * Formato: 1200x630

Figura 18: Protótipo de tela da inserção dos dados básicos do aplicativo.

Fonte: Autoria própria

A Figura 19, mostra o protótipo de tela para a inserção dos fundos do aplicativo. No exemplo mostrado foi inserido duas imagens e seus filtros são: A imagem em tons de verde será para o sexo masculino, já a em tons de vermelho será para o sexo feminino.

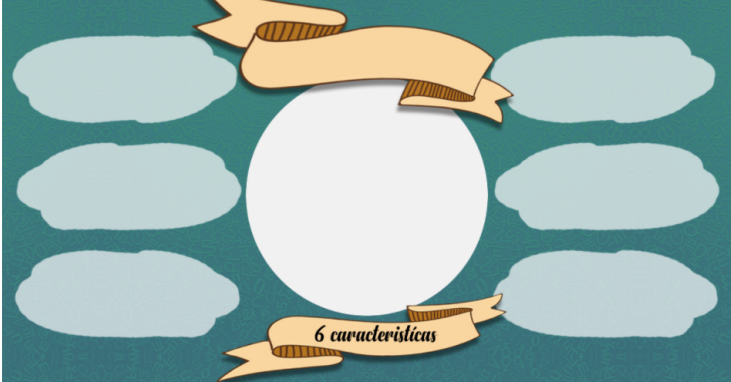
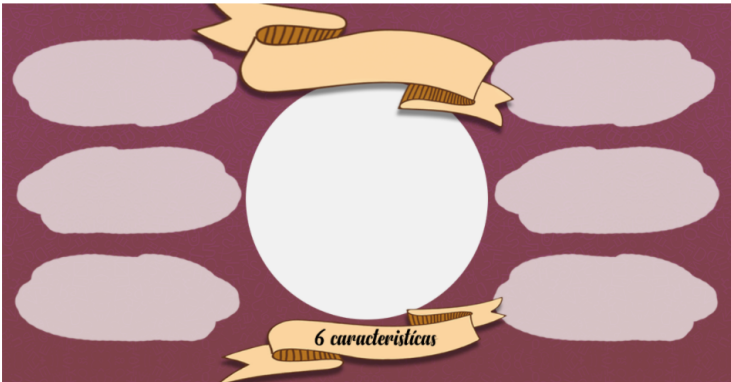
Fundos do Aplicativo					
1	<div> <p>Imagem *</p> <p>Formato: 1200x630</p>  </div> <div> <p>Filtrar imagem</p> <table border="1"> <thead> <tr> <th>Sexo</th> <th>Signo</th> </tr> </thead> <tbody> <tr> <td>Masculino ▾</td> <td>- Selecciona ▾</td> </tr> </tbody> </table> </div>	Sexo	Signo	Masculino ▾	- Selecciona ▾
Sexo	Signo				
Masculino ▾	- Selecciona ▾				
2	<div> <p>Imagem *</p> <p>Formato: 1200x630</p>  </div> <div> <p>Filtrar imagem</p> <table border="1"> <thead> <tr> <th>Sexo</th> <th>Signo</th> </tr> </thead> <tbody> <tr> <td>Feminino ▾</td> <td>- Selecciona ▾</td> </tr> </tbody> </table> </div>	Sexo	Signo	Feminino ▾	- Selecciona ▾
Sexo	Signo				
Feminino ▾	- Selecciona ▾				

Figura 19: Protótipo de tela da inserção dos fundos do aplicativo.

Fonte: Autoria própria

A Figura 20, mostra o protótipo de tela para a inserção dos fundos do aplicativo. No exemplo mostrado foi inserido duas imagens e seus filtros são: A imagem em tons de verde será para o sexo masculino, já a em tons de vermelho será para o sexo feminino. Ao selecionar o campo "Imagem de Fundo", como mostrado na Figura 20, a foto do usuário irá ficar em baixo dos fundos, resultando em uma imagem arredondada, como foi mostrado na Figura 17.

Lista de Fotos				
1	Configuração do Box		Configuração da Foto	
	W *	H *	X *	Y *
	411	424	395	112

Lista de Fotos				
1	Configuração do Box		Configuração da Foto	
	Indice *	Opacidade *	Imagem de Fundo	Imagem Cinza
	0	100	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 20: Protótipo de tela da inserção das fotos do usuário no aplicativo.

Fonte: Autoria própria

A Figura 21 e 22, mostra o protótipo de tela para a inserção dos fundos do aplicativo. No exemplo mostrado foi inserido duas imagens e seus filtros são: A imagem em tons de verdade será para o sexo masculino, já a em tons de vermelho será para o sexo feminino.

Grupos de Textos	
1	Grupo de Texto ▼
2	Grupo de Texto ▼
3	Grupo de Texto ▼
4	Grupo de Texto ▼
5	Grupo de Texto ▼
6	Grupo de Texto ▼
7	Grupo de Texto ▼

Figura 21: Protótipo de tela da lista de Grupos de Textos inseridos.

Fonte: Autoria própria

1 Grupo de Texto

Configuração do box

Configuração dos Textos

Textos

W *

H *

X *

Y *

303

119

44

72

1 Grupo de Texto

Configuração do box

Configuração dos Textos

Textos

Family *

Size *

Color *

Case *

Align X *

Align Y *

Chesnan Regular

40

Selecione cor

Normal

Centro

Centro

1 Grupo de Texto

Configuração do box

Configuração dos Textos

Textos

Textos

1 Texto

Texto *

Adora um bom papo

Filtros

Sexo

- Selecionar

Signo

- Selecionar

2 Texto

Texto *

Exagero em pessoa

Filtros

Sexo

- Selecionar

Signo

- Selecionar

Figura 22: Protótipo de tela da inserção dos Grupos de Textos.

Fonte: Autoria própria

A Figura 22, mostra os campos de um grupo de textos preenchidos. Para o exemplo será usado 7 grupos de textos, como mostrado na Figura 21, sendo um grupo para cada balão das imagens de fundo, e um para inserir o nome do usuário. Como mostrado na figura é possível inserir diversos textos em um grupo de texto, porém será selecionado apenas um texto por grupo ao gerar o resultado.

3.1.2 SITE

O site é o sistema web que irá interagir com o usuário. Para que essa interação aconteça ele irá ver a publicação do link em sua rede social, ao acessar ele irá poder interagir com o sistema. O diagrama de caso de uso geral abaixo mostra uma visão ampla do funcionamento do site.

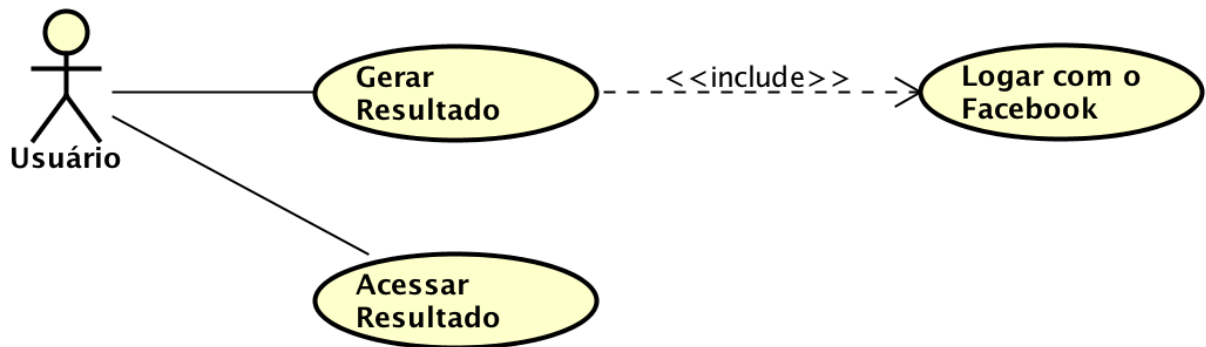


Figura 23: Caso de uso geral do site.

Fonte: Autoria própria

Para isso o sistema deve conter os seguintes requisitos funcionais:

- **RF6** O sistema deve permitir que os usuários realize a autenticação a partir do Facebook Login;
- **RF7** O sistema deve permitir que o usuário interaja com o aplicativo resultando em um imagem;
- **RF8** O sistema deve permitir que o usuário compartilhe o seu resultado.

3.1.2.1 CONSTRUÇÃO DA IMAGEM

A construção da imagem é a fase onde o sistema irá integrar os dados do usuário com a personalização definida na inserção do aplicativo, na subseção 3.1.1.1. A Figura 24 mostra o protótipo da tela de visualização do aplicativo, nele é possível identificar os atributos de visualização (título, texto e imagem de chamada) e um espaço para a inclusão de anúncios. O botão "Ver resultado" será usado para iniciar o processamento da imagem, iniciando com a autenticação com o *Facebook Login* para ter acesso aos dados como nome, data de aniversário e fotos, para o sistema realizar a construção da imagem.

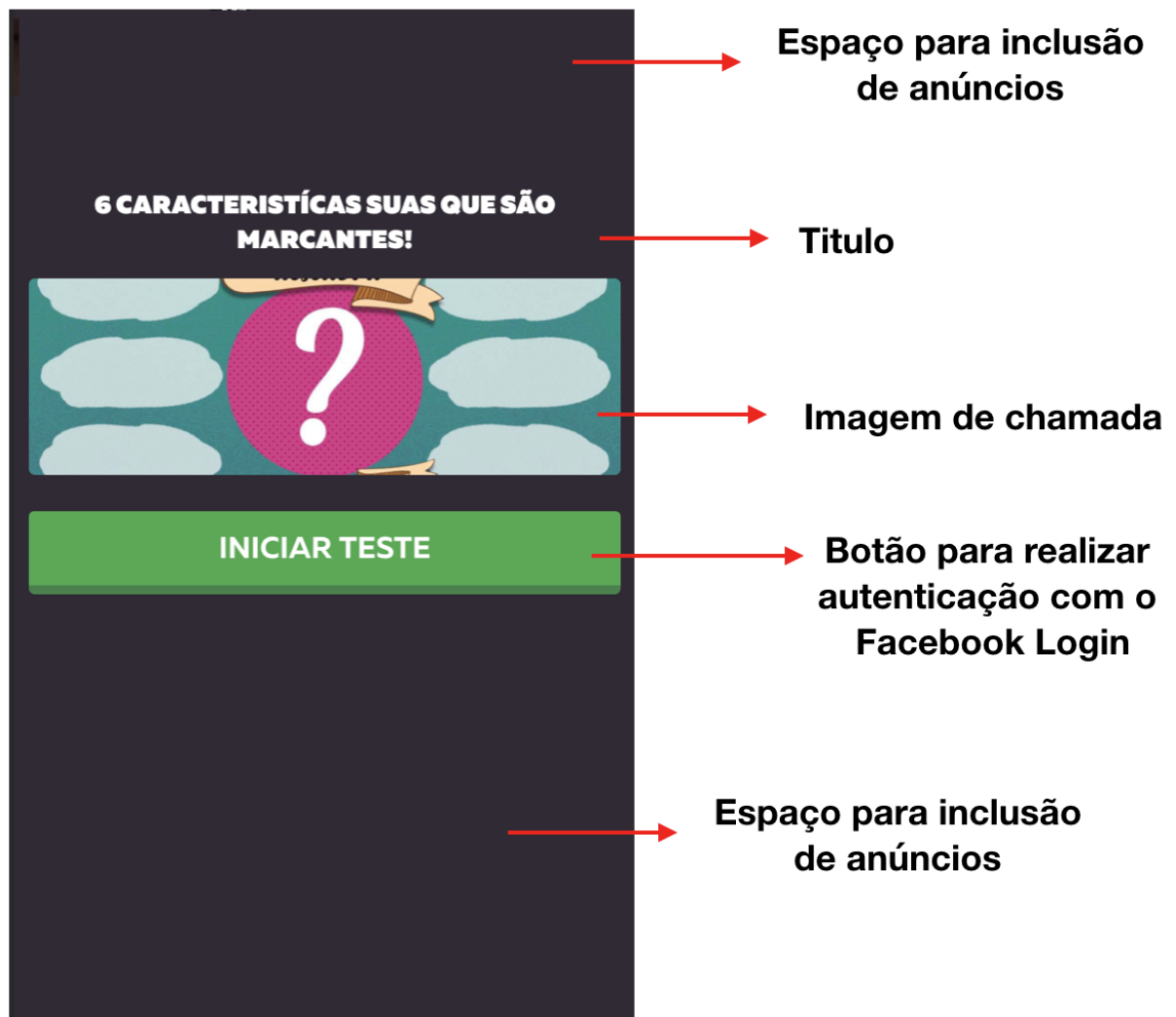


Figura 24: Protótipo de tela da visualização do aplicativo.

Fonte: Autoria própria

Depois de construir a imagem o sistema irá redirecionar o usuário para visualizar a mesma. Nessa tela irá conter a imagem gerada, um botão para gerar novamente e outro para que o usuário possa compartilhar em seu Facebook, como mostra na Figura 25.



Figura 25: Protótipo de tela da visualização do resultado gerado.

Fonte: Autoria própria

3.2 TECNOLOGIAS E FERRAMENTAS

A seguir, destacam-se as principais ferramentas e tecnologias a serem utilizadas neste trabalho.

3.2.1 PHP

O sistema *web* será desenvolvido utilizando a linguagem PHP, por ser uma linguagem rica em recursos e amplamente consolidada no desenvolvimento *web*. O PHP 7 é a nova versão da linguagem e que será utilizada para desenvolver essa proposta, trouxe um salto de *perfor-*

mance entre 10 a 25% em comparação com sua versão anterior. Também trouxe melhorias como uso otimizado de memória, classes anônimas, possibilidade do tratamento de erros fatais, entre outros. Com esse avanço a linguagem aumentou sua confiabilidade para o desenvolvimento de softwares de grande porte (OTEMUYIWA, 2015; SAGGIO, 2016).

3.2.2 LARAVEL

Em conjunto com o PHP será utilizado o Laravel, um *framework* PHP voltado para o desenvolvimento web. O *framework* utiliza a arquitetura MVC (*Model-View-Controller*), que auxilia na separação de responsabilidades da aplicação, onde o *Model* é a camada que realiza as interações com banco de dados e regras de negócio, a camada *View* é a parte que é visualizada pelo cliente, como por exemplo, em aplicações *web*, é o que será renderizado no navegador do usuário. Por final, a camada *Controller* é que realiza a ponte entre o *Model* e o *View*.

Para a criação da camada *View*, o Laravel utiliza uma *Engine de template* chamada *Blade*, que traz uma gama de ferramentas que ajudam a criar interfaces de forma rápida e evitar a duplicação de código. Para se comunicar com um Banco de Dados o Laravel utiliza uma implementação da interface *ActiveRecord* chamada de *Eloquent ORM (Object Relational Mapping)*, que é uma ferramenta que traz várias funcionalidades para o gerenciamento do banco de dados.

O gerenciamento de rotas do Laravel é bem simples e direto. As rotas são uma definição do que será executado a partir de uma URL que foi acessada, como por exemplo, caso o usuário acesse a URL `"/produtos/lista"`, o sistema deve mostrar a lista de produtos. O Laravel permite que uma rota seja associada a um *Controller*, garantindo assim que o *Controller* cuide do processamento do que foi requisitado pelo usuário.

O Laravel trás também dois recursos significativos para a gestão do banco de dados da aplicação, que são as migrações e os *seeders*. As migrações são como um controle de versão do banco de dados, permitindo que seja possível o compartilhamento e modificação do esquema de banco de dados de forma rápida e prática, pois todo banco de dados é criado via programação, criando um arquivo para cada tabela ou alteração no banco de dados. O código abaixo contém um exemplo de uma migração para a criação da tabela de pessoa.

Listing 3.1: Python example

```
<?php
class CriarTabelaDePessoa extends Migration{
public function up(){
Schema::create('pessoa', function (Blueprint $table) {
```

```

$table->increments('id');
$table->string('nome');
$table->string('email');
});
}
public function down(){
Schema::drop('pessoa');
}
}
?>

```

O outro recurso que foi citado acima, são as *seeders*, que auxiliam na sementeação de dados de teste na aplicação. Com ela é possível inserir dados aleatório nas tabelas de forma automatiza.

O Laravel fornece uma interface de linha de comando, chamada *Artisan*. Com ela é possível gerar as classes que serão necessárias para sua aplicação, como, um *Controller*, uma *Model*, *Seeders* e outros. Também é possível simular a execução de uma classe do projeto e interagir com a mesma de forma dinâmica.

O gerenciamento de dependências do Laravel é feito utilizando o *Composer*, que é uma ferramenta para o gerenciamento de dependências de projetos em PHP. Ele permite que você declare as bibliotecas que seu projeto depende, e oferece uma interface via linha de comanda para que você possa atualizar, remover e instalar dependências de forma rápida.

O Laravel incentiva o uso de boas práticas de programação e utiliza as PSR (*PHP Standards Recommendations*) como guia para estilo de escrita do código, o seu intuito principal é ajudar a desenvolver aplicações com alto desempenho de forma rápida.

3.2.3 MYSQL

O MySQL é um gerenciador de banco de dados, que utiliza como padrão a linguagem SQL. Atualmente o mesmo é amplamente usado em aplicações para *web*, empresas como Google, NASA, Netflix e Suzuki, utilizam MySQL em suas aplicações. Algumas das características que o torna uma boa opção de SGBD é sua velocidade, escalabilidade e confiabilidade. Além disso o MySQL tem a capacidade de manipular tabelas com mais de 50 milhões de registros, uma alta velocidade na execução de comandos e um fácil e eficiente controle de privilégios de usuários. Dessa forma o MySQL junto do PHP formam uma excelente escolha para o desenvolvimento de aplicações web robusta.

3.2.4 REDIS

O Redis é uma ferramenta que auxilia o armazenamento de estrutura de dados no formato de chave-valor na memória. Como seus dados sempre ficarem na memória o seu acesso é extremamente rápido, o que faz com que o Redis seja uma ferramenta ideal para trabalhar com *cache*. Portanto para garantir a escalabilidade do sistema, evitando que o MySQL se torne um gargalo na aplicação, será utilizado o Redis como *cache* do banco de dados, como será mostrado no capítulo da arquitetura do sistema.

3.2.5 INTERFACE WEB

Para o desenvolvimento da interface *web* do projeto, será utilizado as seguintes tecnologias:

- **HTML5** (*HyperText Markup Language*): É uma Linguagem de marcação, usada para a criação dos elementos da página *web*;
- **CSS3** (*Cascading Style Sheets*): É uma linguagem utilizada para personalizar os elementos criados com o HTML, como definir cores, formatos e outras formatações;
- **JavaScript**: é uma linguagem de programação *client-side*, utilizada para controlar o HTML e o CSS e manipular comportamentos na página;
- **jQuery**: é uma biblioteca JavaScript que auxilia a manipulação de elementos HTML;
- **Vue.js**: é uma biblioteca JavaScript que auxilia na criação de elementos em forma de componentes reativos.

3.2.6 AMAZON WEB SERVICES

A computação em nuvem é a entrega sob demanda de poder computacional, armazenamento, banco de dados, aplicações e outros recursos. Como todos serviços são sob demanda, a definição de preço será conforme o uso. As principais vantagens do seu uso é:

- Os custos e capacidade variam conforme o uso, com isso é possível aumentar ou diminuir sempre que houver demanda;
- Usar configurações de infra-estrutura pré-definidas, evitando retrabalho;
- Alta garantia de escalabilidade e disponibilidade.

Para o desenvolvimento desta proposta será utilizado a nuvem da AWS (*Amazon Web Services*), como será mostrado no capítulo da arquitetura do sistema.

3.3 SCRUM SOLO

O processo de desenvolvimento escolhido para o desenvolvimento dessa proposta será o *Scrum Solo*. O *Scrum Solo* é uma adaptação do *Scrum* original, essa adaptação foi criada para que seja possível utilizar o *Scrum* individualmente, que se divide nos seguintes artefatos (PAGOTTO et al.,):

- *Product Backlog*: Será o conjunto de itens que devem ser desenvolvidos para concluir esta proposta;
- *Sprint Backlog*: Lista de tarefas necessárias para a conclusão de partes do *Product Backlog*;
- *Sprint*: Tempo necessário para a conclusão de um conjunto de itens do *Sprint Backlog*, que nessa proposta deverá ser de 1 semana;

Para adequar o scrum solo a esta proposta, não irá haver reuniões diárias, mas reuniões de orientação ao fim de cada *Sprint*.

3.4 ARQUITETURA DO SISTEMA

Neste capítulo será descrito o funcionamento dos principais componentes que formam a arquitetura do sistema. Toda arquitetura foi desenvolvida pensando em dois tópicos principais, redução de custos e escalabilidade. Para isso foi usado diversas tecnologias descritas na sessão de ferramentas e tecnologias. A Figura abaixo mostra uma visão geral da arquitetura.

O modelo da arquitetura é o de cliente-servidor, no qual o cliente através da internet envia uma requisição para o servidor, o mesmo recebe, processa a solicitação e responde, como mostra a Figura 27.

Nesse modelo o servidor é a infra-estrutura que hospeda o sistema. Para que o sistema proposto funcione, será necessário que o servidor contenha instalado e configurado o Apache 2.4 e o PHP 7.

Já o para a criação do servidor será utilizado o serviço da AWS conhecido como EC2 (*Elastic Compute Cloud*). Esse serviço disponibiliza de forma rápida um servidor com o sistema

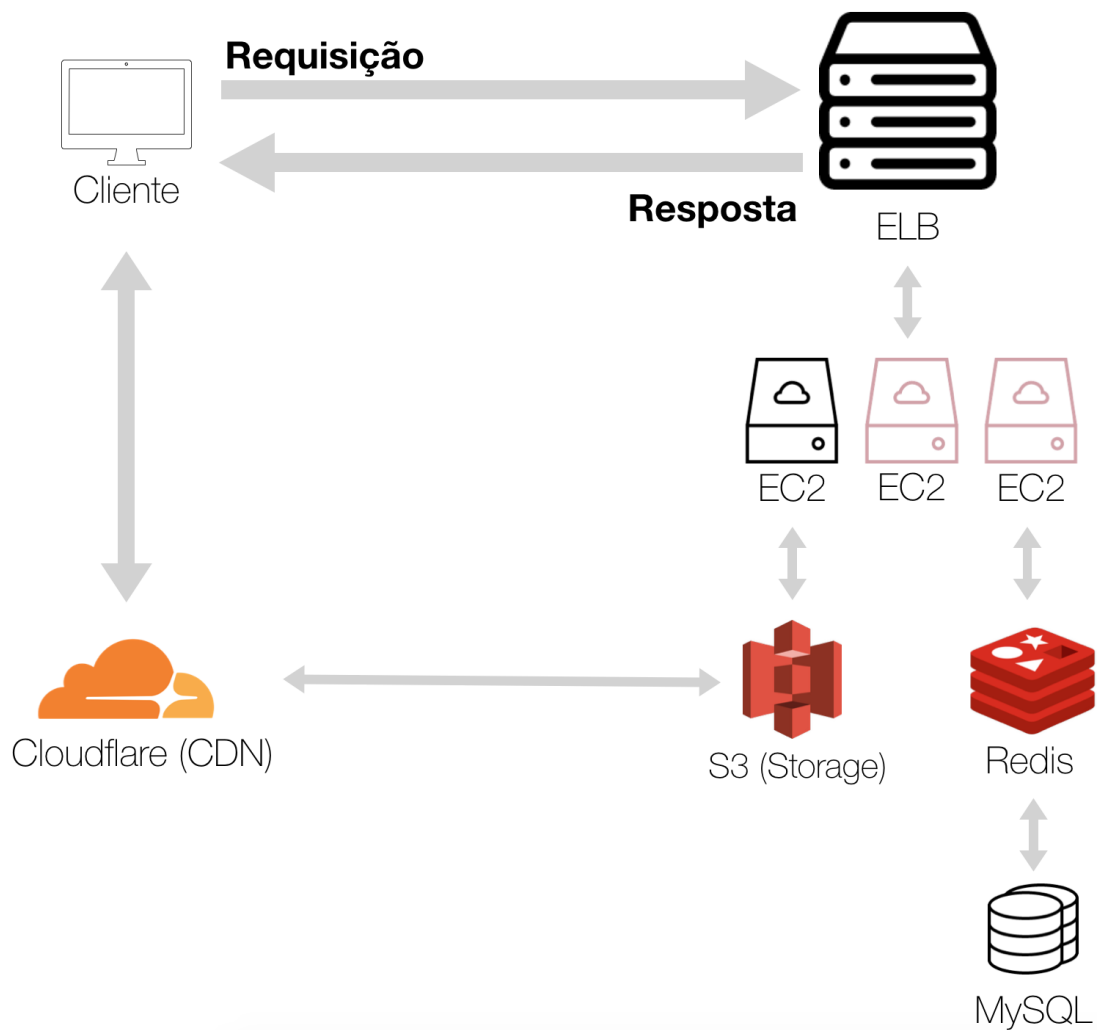


Figura 26: Visão geral da arquitetura do sistema

Fonte: Autoria própria

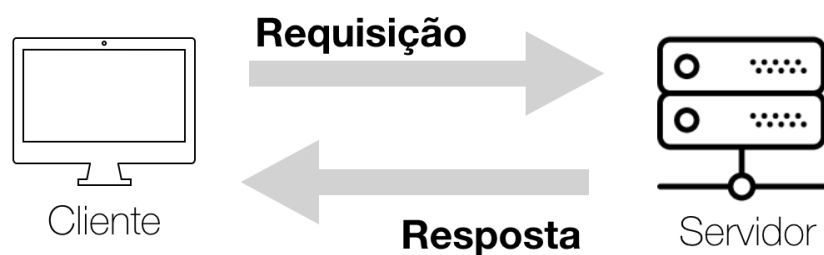


Figura 27: Exemplo do funcionamento da arquitetura cliente-servidor

Fonte: Autoria própria

operacional escolhido, cada servidor criado com o EC2 é conhecido como instância. Um das vantagens da utilização do EC2 é sua fácil integração com o balanceador de carga da AWS, chamado de ELB (*Elastic Load Balancing*).

Para garantir a escalabilidade foi incluído um balanceador de carga na frente dos servidores. Sua função é destinar as requisições para os servidores que estiverem com mais recursos disponíveis no momento. Caso todos os servidores estejam quase sem recursos, o balanceador de carga irá incluir um novo servidor. Caso a média dos recursos de todas as instâncias ativas estiverem baixas, o balanceador irá remover servidores. Portanto o balanceador de carga irá gerenciar a escalabilidade da aplicação, aumentando ou diminuindo os recursos. Será utilizado o serviço de balanceamento de carga da AWS o ELB.

Além disso é possível configurar um EC2 e ao integra-lo com o ELB, para que ele replique quando necessário uma instância idêntica a pré-definida inicialmente, como mostra a Figura 28.

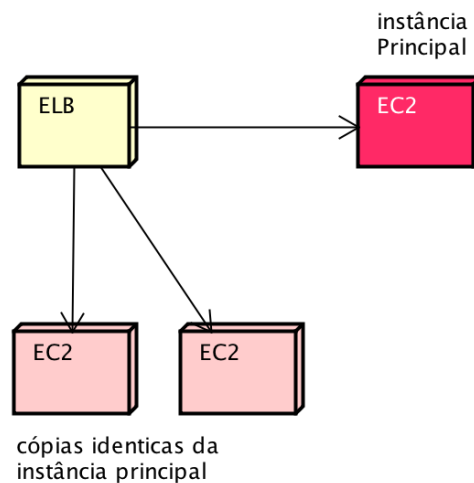


Figura 28: Exemplo do funcionamento do ELB

Fonte: Autoria própria

Para armazenamento de dados será utilizado um SGBD, o escolhido para a aplicação será o MySQL. Para provisionar o MySQL vamos utilizar o serviço da AWS chamado RDS (*Relational Database Service*), com ele é possível fazer a criação e gerenciamento do banco de dados de forma simples e com alta confiabilidade. Pois o RDS fornece duas coisas muito importantes, o backup de forma automatizada e espalhamento geográfico do banco de dados, garantindo que os dados estejam sempre seguros e disponíveis. Porém a cobrança do uso do serviço RDS da AWS é feito a partir de vários fatores, alguns deles são:

- Tipo de instancia que será utilizada no banco de dados (A AWS, fornece diversos tipos de instancias, com mais ou menos memoria, CPU ou rede), onde a AWS cobra por hora;
- Quantidade de dados armazenados em GB;
- Quantidade de entradas e saídas feitas no banco de dados.

Para reduzir os custos com o banco de dados e aumentar sua velocidade de resposta, cada consulta enviada ao MySQL, deverá ter seu resultado salvo como *cache* no Redis, para que as próximas consultas não sejam enviadas para o MySQL, mas venha direto do Redis, como mostra a figura 26. Outra vantagem com a utilização do Redis, é a possibilidade utilizar uma instancia de banco de dados que não tenha muitos recursos, já que as consultas estarão em *cache*, gerando assim poucas saídas do banco de dados e com isso não consumindo muitos recursos do MySQL.

As imagens geradas pelos usuários devem ser salvas. Porém se as imagens forem salvas dentro do EC2, quando o ELB remover um EC2 as imagens que estarão neste servidor serão perdidas. Para solucionar esse problema utilizaremos o serviço de *Storage* da AWS, chamado S3 (*Simple Storage Service*), com ele é possível salvar arquivos de diversos tamanhos e formatos na nuvem, e acessa-los via URL, API ou através da linha de comando. O S3 fornece diversos recursos. Um importante para a aplicação é a possibilidade de definir uma data de exclusão para um conjunto de arquivos. Será definido que os arquivos de imagens criados pelos usuários devem ser apagados após 30 dias da sua criação. Fazendo com que o custo de armazenamento de imagens seja reduzido.

A cobrança do S3 é feita por diversos fatores como, quantidade de GB armazenados, quantidade de dados transferidos. Para reduzir o custo de transferência de dados será utilizado um CDN (*Content Delivery Network*). O CDN "é uma rede de servidores que armazenam réplicas de arquivos ou conteúdos em memória (*cache*) e depois os entrega aos visitantes, baseando-se na localização geográfica para conectá-los ao servidor mais próximo e mais rápido,

reduzindo o tempo de transferência dos dados (latência). O CDN também se comunicará com o servidor de origem para entregar qualquer conteúdo ou arquivo que não tenha sido previamente armazenado em *cache*. Além disso seu uso permite que o site suporte melhor ataques DDoS e mantenha alta disponibilidade, por estar presente simultaneamente em diferentes *datacenters*”. Com isso a transferência de dados com o S3 será feita apenas uma vez por arquivo, reduzindo o custo de transferência de dados. Para essa proposta será utilizado o CDN da empresa Cloudflare¹, pois eles oferecem esse serviço de forma gratuita.

¹www.cloudflare.com

4 CRONOGRAMA

REFERÊNCIAS

ABEDNIYA, A.; MAHMOUEI, S. S. The Impact of Social Networking Websites to Facilitate the Effectiveness of Viral Marketing. **International Journal of Advanced Computer Science and Applications**, v. 1, n. 6, p. 139–146, 2010.

BARDIN, A. **Introducing Expanded YouTube Partner Program Safeguards to Protect Creators**. 2017. Disponível em: <<https://youtube-creators.googleblog.com/2017/04/introducing-expanded-youtube-partner.html>>. Acesso em: 19 de setembro de 2017.

BARICHELO, E. d. R.; OLIVEIRA, C. de. O marketing viral como estratégia publicitária nas novas ambiências midiáticas. **Em Questão**, p. 29–44, 2010. ISSN 1808-5245.

BOLAND, B. **Alcance orgânico no Facebook: suas dúvidas respondidas**. 2014. Disponível em: <<https://www.facebook.com/business/news/BR-Alcance-organico-no-Facebook-suas-duvidas-respondidas>>. Acesso em: 21 de setembro de 2017.

CAMARERO, C.; San José, R. Social and attitudinal determinants of viral marketing dynamics. **Computers in Human Behavior**, v. 27, n. 6, p. 2292–2300, 2011. ISSN 07475632.

CASTRO, G. G. d. S. Cheia de Charme: a classe trabalhadora no paraíso da cibercultura. **Revista Ciberlegenda**, p. 59–69, 2012.

CHU, S.-C. Viral advertising in social media. **Journal of Interactive Advertising**, Routledge, v. 12, n. 1, p. 30–43, 2011. Disponível em: <<http://dx.doi.org/10.1080/15252019.2011.10722189>>.

COOKE, M.; BUCKLEY, N. Web 2.0, social networks and the future of market research. **International Journal of Market Research**, v. 50, n. 2, p. 267–292, 2008. ISSN 14707853.

FACEBOOK. **Existe monetização no facebook? Um video com cinco milhoes de visualizações?** 2016. Disponível em: <<https://facebook.com/business/help/community/question/?id=1305668349497166>>. Acesso em: 19 de setembro de 2017.

FACEBOOK. **Agora somos 2 bilhões de pessoas: obrigado por estar em nossa comunidade!** 2017. Disponível em: <<https://br.newsroom.fb.com/news/2017/06/agora-somos-2-bilhoes-de-pessoas-obrigado-por-estar-em-nossa-comunidade/>>. Acesso em: 18 de setembro de 2017.

FACEBOOK. **Facebook - About**. 2017. Disponível em: <<https://newsroom.fb.com/products/>>. Acesso em: 3 de Outubro de 2017.

HABER, R. **I have a Facebook page with over 75,000 likes. How do I monetize the Facebook page?** 2015. Disponível em: <<https://www.quora.com/I-have-a-Facebook-page-with-over-75-000-likes-How-do-I-monetize-the-Facebook-page/answer/Rion-Haber>>.

HARTLINE, J.; MIRROKNI, V.; SUNDARARAJAN, M. Optimal marketing strategies over social networks. **Proceeding of the 17th international conference on World Wide Web WWW 08**, v. 5, n. 2, p. 189, 2008. ISSN 01655728.

INSTAGRAM. **Our Story**. 2017. Disponível em: <<https://instagram-press.com/our-story/>>. Acesso em: 18 de setembro de 2017.

Marc's Blog. **Virality VS Velocity**. 2013. Disponível em: <<http://marcuberti.com/2013/09/virality-velocity/>>.

O'REILLY, T. What is Web 2.0: Design patterns and business models for the next generation of software. **Communications & strategies**, v. 65, n. 4578, p. 17–37, 2007.

OTEMUYIWA, P. **Brace Up, PHP 7 is Changing The Game!!!** 2015. Disponível em: <<https://medium.com/@unicodeveloper/brace-up-php-7-is-changing-the-game-1ad2343cd67f>>.

PAGOTTO, T.; FABRI, J. A.; ERARIO, A. L. Scrum Solo.

PEROTTONI, A.; NETO, L. R.; BEHLING, H. P. A Viralização da Campanha The Responses na Internet. **Intercom**, n. IX, p. 1–15, 2010.

SAGGIO, R. **Novidades do PHP 7**. 2016. Disponível em: <<http://blog.caelum.com.br/novidades-do-php-7/>>.

SHUEN, A. **Web 2.0: A Strategy Guide**. O'Reilly, 2008. ISBN 978-0-596-52996-3. Disponível em: <<https://web20damas.wikispaces.com/file/view/Web+2.0,+A+Strategy+Guide.pdf>>.

SOCIALBAKERS. **Brazil Facebook page statistics**. 2017. Disponível em: <<https://www.socialbakers.com/statistics/facebook/pages/total/brazil/>>. Acesso em: 18 de setembro de 2017.

SOCIALBAKERS. **Twitter statistics for Brazil**. 2017. Disponível em: <<https://www.socialbakers.com/statistics/twitter/profiles/brazil/>>. Acesso em: 18 de setembro de 2017.

SOCIALBAKERS. **YouTube Channels Stats in Brazil**. 2017. Disponível em: <<https://www.socialbakers.com/statistics/youtube/channels/brazil/>>. Acesso em: 18 de setembro de 2017.

TRUSOV, M.; BUCKLIN, R. E.; PAUWELS, K. Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site. **Journal of Marketing**, v. 73, n. 5, p. 90–102, 2009. ISSN 0022-2429.

TWITTER. **About Twitter**. 2017. Disponível em: <<https://about.twitter.com/pt/company>>. Acesso em: 18 de setembro de 2017.

YOUTUBE. **Um bilhão de horas assistidas por dia**. 2017. Disponível em: <<https://www.youtube.com/intl/pt-BR/yt/about/press/>>. Acesso em: 19 de setembro de 2017.