

INSTITUTO FEDERAL

Brasília

Instituto Federal de Brasília
Bacharelado em Ciência da Computação
Campus Taguatinga

MELHORIA DO SISTEMA INTEGRADO DE DIVULGAÇÃO DE INFORMAÇÕES DO IFB: MELHOR INTEGRAÇÃO AO FACEBOOK E APLICATIVO MOBILE

Por

MAXWELL BORGES BEZERRA

Trabalho de Graduação

BRASÍLIA/2018

Maxwell Borges Bezerra

**MELHORIA DO SISTEMA INTEGRADO DE DIVULGAÇÃO DE
INFORMAÇÕES DO IFB: MELHOR INTEGRAÇÃO AO FACEBOOK
E APlicativo MOBILE**

Trabalho apresentado ao Programa de Bacharelado em Ciência da Computação do Departamento de Computação da Instituto Federal de Brasília como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Daniel Saad Nogueira

BRASÍLIA
2018

Maxwell Borges Bezerra

Melhoria do Sistema Integrado de Divulgação de Informações do IFB: melhor integração ao Facebook e Aplicativo Mobile/ Maxwell Borges Bezerra. – BRASÍLIA, 2018-

108 p. : il. (algumas color.) ; 30 cm.

Orientador Prof. Me. Daniel Saad Nogueira

Trabalho de Graduação – Instituto Federal de Brasília, 2018.

1. Sistema Integrado. 2. Aplicativo Mobile. I. Prof. Me. Daniel Saad Nogueira. II. Instituto Federal de Educação, Ciência e Tecnologia de Brasília. III. Campus Taguatinga. IV. Melhoria do Sistema Integrado de Divulgação de Informações do IFB: melhor integração ao Facebook e Aplicativo Mobile

CDU 004

Trabalho de Graduação apresentada por **Maxwell Borges Bezerra** ao programa de Graduação em Ciência da Computação do Instituto Federal de Brasília, sob o título **Melhoria do Sistema Integrado de Divulgação de Informações do IFB: melhor integração ao Facebook e Aplicativo Mobile**, orientada pelo **Prof. Me. Daniel Saad Nogueira** e aprovada pela banca examinadora formada pelos professores:

Prof. Me. Daniel Saad Nogueira
Departamento de Computação/IFB
Orientador

Prof. Dr. Guilherme Novaes Ramos
Departamento de Computação/UnB
Convidado 1

Prof. Dr. Fabiano Cavalcanti Fernandes
Departamento de Computação/IFB
Convidado 2

BRASÍLIA
2018

Agradecimentos

Resumo

Este trabalho apresenta o Sistema Integrado de Divulgação de Informações do IFB Câmpus Taguatinga - SID que, por meio de uso dos conceitos de sinalização digital e marketing digital, visa veicular notícias em forma de publicações que são repassadas por meio de televisões no ambiente do *Campus* ou fora dele por meio de um aplicativo móvel instalado nos celulares.

Foram realizadas diversas alterações no sistema inicial chamado de SIDv2. As alterações vão desde modificações na arquitetura existente para implementação de novas funcionalidades, até a implementação de um aplicativo exclusivo para celulares. As alterações foram realizadas de modo a flexibilizar a implantação de novas funcionalidades e serviços, para que sistemas distintos pudessem fazer uso de um mesmo sistema.

O SIDv3 também passou a possuir a integração completa com a rede social Facebook, disponibilizando a possibilidade de realização de publicações em páginas do Facebook e apresentação dos conteúdos referentes a essas publicações. Com isto é realizado uma junção de meios que anteriormente eram distintos no IFB.

Além disso, o aplicativo *mobile* servirá de forma a repassar as divulgações criadas, além de realizar o consumo de uma API fictícia para interação entre alunos e professores com a troca de mensagens. Esse consumo de API possibilita uma futura integração com o Sistema de Gestão Acadêmica - SGA.

Abstract

Lista de Figuras

5.1	Casos de uso da ações do módulo administrador	53
5.2	Sequencia para efetivação de <i>login</i>	54
5.3	Modelo entidade e relacionamento	55
5.4	Diagrama de classe do módulo administrador	55
5.5	Diagrama de sequência para inserção	56
5.6	Página de inserção do módulo administrador.	56
5.7	Página de listagem do módulo administrador.	58
5.8	Diagrama de classe do módulo API	61
5.9	Casos de uso da ações do módulo cliente	61
5.10	Diagrama de classe do módulo cliente	62
5.11	Página do cliente	62
5.12	Casos de uso aplicativo móvel.	63
5.13	Modelo entidade e relacionamento aplicativo móvel.	64
5.14	Página de inserção no módulo administrador.	65
5.15	Página do Aluno.	65
5.16	Página do professor.	66
5.17	Divulgação enviada ao Facebook com auxílio do SID	68
A.1	Tela de login do Administrador	84
A.2	Tela de inserção de novas publicações	85
A.3	Tela de listagem das publicações	87
A.4	Tela de edição das publicações	87
A.5	Tela de detalhamento das publicações	88
A.6	Tela do cliente	89
A.7	Diagrama de casos de uso do módulo administrador	90
A.8	Diagrama de casos de uso do módulo cliente	90
A.9	Diagrama de casos de uso do aplicativo móvel	91
A.10	Diagrama de classes do módulo administrador	92
A.11	Diagrama de classes do submódulo API	92
A.12	Diagrama de classes do módulo Cliente	93
A.13	Diagrama entidade–relacionamento do aplicativo móvel (API fictícia)	93
A.14	Diagrama entidade–relacionamento do <i>web</i>	94
A.15	Diagrama de sequência para inserir	94
A.16	Diagrama de sequência para login	95
A.17	Diagrama de sequência para deletar	95
A.18	Diagrama de sequência para detalhar	96

A.19 Diagrama de sequência para editar	96
A.20 Diagrama de sequência para listar	97

Lista de Algoritmos

3.1	Exemplo de JSON	31
4.1	Obtendo Token de acesso a página	36
4.2	Obtendo <i>token</i> de acesso de aplicativos	37
4.3	Retorno <i>token</i> de acesso de aplicativo 4.2	37
4.4	Obtendo Token de acesso a página	37
4.5	Retorno <i>token</i> de acesso de página 4.4	37
4.6	Obtendo Token de uma única página	38
4.7	Retorno <i>token</i> de uma única página específica	38
4.8	Obtendo Token Infinito	39
4.9	Conexão entre aplicativo e SDK	40
4.10	Solicitação de Login	40
4.11	Requisição com o uso do ID	41
4.12	Requisição com o uso do ID+aresta	41
4.13	Requisição com o uso do ID+campo	41
4.14	Requisição POST	42
4.15	Requisição DELETE	42
4.16	Requisitar informações do próprio usuário	43
4.17	Requisitar informações de um usuário específico	43
4.18	Requisitar informações de uma página	44
4.19	Resposta do servidor as requisições 4.16, 4.17 e 4.18 (Usuário e Página)	44
4.20	Requisitar informações de uma postagem específica	44
4.21	Resposta do servidor a uma requisição 4.20 (Postagem)	44
4.22	Requisitar informações de um comentário específico	44
4.23	Resposta do servidor a uma requisição 4.22 (Comentário)	45
4.24	Requisitar uma evento específico	45
4.25	Resposta do servidor a requisição 4.24 (Evento)	45
4.26	Requisitar uma álbum específico	46
4.27	Requisitando todas as publicações de um usuário	46
4.28	Resposta da requisição 4.27 (Feed)	46
4.29	Requisitar todos os álbuns de uma página	47
4.30	Requisitar os vídeos publicados na página	47
4.31	Requisitar os eventos agendados pela página	47
4.32	Requisitar as fotos de perfil publicadas na página	48
4.33	Requisitar todos os comentários de uma postagem em uma página	48
4.34	Requisitar todas as curtidas de um comentário	48
4.35	Resposta das Listagens 4.29, 4.32 e 4.30 (Álbuns, Fotos e Videos)	48
4.36	Resposta da requisição 4.31 (Eventos)	49

4.37	Resposta da requisição 4.33 (Comentários)	49
4.38	Resposta da requisição 4.34 (Likes)	50
4.39	Requisitar os comentários de todas as publicações da página	50
4.40	Requisição de múltiplos campos	51
4.41	Retorno da requisição 4.40 (Campos)	51
5.1	Requisitando dados para divulgação	59
5.2	Retorno da requisição 5.1	59
5.3	Requisitar todos os dados	60
5.4	Retorno da requisição 5.3	60
5.5	Requisitar lista de dados específica	60
5.6	Retorno da requisição 5.1	60
5.7	Incluir novas mensagens	64
5.8	Permissões concedidas	67
5.9	Foto de usuário	68
5.10	Foto de usuário	69

Lista de Tabelas

2.1 Comparativo	25
4.1 Exemplos de nós e arestas	35
6.1 Comparativo	73
A.1 Interfaces de Usuário	83
A.2 Comandos da tela de logar	84
A.3 Campos da tela de inserção	86
A.4 Comandos da tela de inserção	86
A.5 Comandos da tela de listagem	87
A.6 Campos da tela de edição	88
A.7 Comandos da tela de edição	88
A.8 Comandos da tela de detalhamento	89
A.9 Requisitos de usuário	98

Lista de Acrônimos

Sumário

1	Introdução	16
1.1	Motivação	18
1.2	Proposta	19
1.3	Objetivos	20
1.3.1	Objetivos Específicos	20
1.4	Metodologia	20
1.5	Organização do documento	21
2	Trabalhos Relacionados	22
2.1	OOZO	22
2.2	MangoSigns	23
2.3	SID Formosa	24
2.4	Screenly	24
2.5	Xibo	24
2.6	Comparativo	25
3	Referencial Teórico	26
3.1	Marketing Digital	26
3.2	Sinalização digital	27
3.3	Arquitetura Cliente-Servidor	28
3.4	Linguagens de programação	28
3.4.1	PHP	29
3.4.2	JavaScript	29
3.5	Banco de Dados	30
3.6	Interface de Programação de Aplicativos	30
3.6.1	REST	30
3.7	Linguagem de modelagem unificada - UML	31
3.8	Metodologia de desenvolvimento - SCRUM	31
3.9	JSON	31
3.10	QR Code	32
3.11	Frameworks	32
3.11.1	Zend	32
3.11.2	Doctrine	32
3.11.3	Cordova	33
3.11.4	Framework 7	33

4 Graph Api	34
4.1 Visão Geral	34
4.2 Token de acesso	36
4.2.1 <i>Tokens</i> de usuário	36
4.2.2 <i>Tokens</i> de aplicativo	37
4.2.3 <i>Tokens</i> de página	37
4.2.4 <i>Tokens</i> infinitos	38
4.3 Autenticação	39
4.4 Permissões	40
4.5 Requisições	41
4.5.1 Tipos de Requisição	42
4.5.2 Nó	42
4.5.3 Aresta	46
4.5.4 Campo	50
5 SID	52
5.1 Visão Geral	52
5.2 Modulo Administrador	53
5.2.1 Módulo API	58
5.3 Módulo Cliente	61
5.4 Aplicativo móvel	63
5.5 Integração	66
5.5.1 <i>login, token</i> de acesso e permissões	66
5.5.2 Publicar	67
5.5.3 Requisição	68
5.6 Possível solução para Implantação - Raspberry	69
6 Resultados	70
6.1 Melhorias em relação ao SIDv2	73
6.1.1 Módulo Administrador	73
6.1.2 Módulo Cliente	74
6.1.3 Submódulo API e Aplicativo móvel	74
6.1.4 Banco de dados	74
6.2 Dificuldades encontradas	75
7 Considerações Finais	76
7.1 Trabalhos Futuros	76
7.1.1 Aplicativo <i>mobile</i>	76
7.1.2 Moderação dos comentários	77
7.1.3 Atraso da recuperação de dados	77

Referências	78
Apêndice	81
A Documentação	82
A.1 Prototipação	83
A.1.1 Visão Geral	83
A.1.2 Interface de usuário	83
A.1.2.1 Interface do administrador (Logar)	83
A.1.2.2 Interface do Administrador (Inserir)	84
A.1.2.3 Interface do Administrador (Listar)	86
A.1.2.4 Interface do Administrador (Editar)	87
A.1.2.5 Interface do Administrador (Detalhar)	88
A.1.2.6 Interface Cliente	89
A.2 Diagramas	90
A.2.1 Diagramas de casos de uso	90
A.2.2 Diagramas de classe	92
A.2.3 Diagramas entidade–relacionamento	93
A.2.4 Diagramas de sequência	94
A.3 Artefatos	98

1

Introdução

Apesar de muito utilizada, a definição exata de mídia é difícil de ser explicada. De acordo com GUAZINA (2007, p.51), o termo “mídia” é usado para representar os meios de comunicação em massa como jornais, televisão, rádio, cinema e *Internet*.

Canais de televisão, outdoors ou até mesmo panfletos, em ambientes públicos ou privados, podem introduzir diversas mudanças comportamentais e comerciais, por isso ESCOBAR (2007, p.3) explica que a mídia é capaz de redefinir “o modo como o homem se comunica e se relaciona com os semelhantes”, assim como HJARVARD (2012, p.54) coloca que ela pode modificar a relação e o comportamento humano, criando o conceito de midiatização, que caracteriza a influência que a mídia possui sobre as pessoas.

GUAZINA (2007, p.53) aponta que os meios de comunicação influenciam no processo de construção e formação das consciências. Além disso, SILVA (2007) indica que a falta de acesso ou interesse por parte da população em buscar outras fontes, induzem no questionamento da veracidade das informações que são repassadas, mostrando o poder que a mídia pode ter sobre as pessoas.

Com o passar dos anos, as divulgações de forma estática e tradicionais (revistas, jornais e canais de televisão) deixaram de ser os meios mais eficientes de se expor um conteúdo ou propaganda. Para MEDITSCH (2001), o advento e popularização da *Internet* trouxeram ameaças para esses meios, então foi necessário que novas formas de expor conteúdos fossem pensadas e desenvolvidas em conjunto com a *Internet*.

Para ESCOBAR (2007, p.4), os atuais meios de veiculação de notícias (televisão e rádio) oferecem a possibilidade da informação ser obtida em tempo real, mas quando há a inclusão da *Internet*, coloca-se a possibilidade de interação com as informações que são recebidas, onde os usuários atuam de forma simultânea, comentando ou opinando sobre aquele determinado assunto. Para DEUZE (2002), o surgimento da *Internet* traz a possibilidade do público “responder, interagir ou mesmo customizar certas histórias”.

Diante disso, surgiram novas formas e ferramentas de *marketing* ampliando os meios de disseminação de notícia. Visto que, esse é responsável por criar e melhorar essas publicações, posto que essas possuem o objetivo de influenciar pessoas a adquirirem ou aderirem determinados produtos ou serviços, com o uso de diversos aparelhos. Sendo assim foram necessários que a

publicidade e a tecnologia convergissem, definindo um novo conceito, o *marketing* digital, que englobam novas tecnologias de comunicação e incluem as redes sociais, como meio de exposição das propagandas.

A *Internet*, também resultou na popularização do uso das redes sociais para as diversas finalidades, incluindo a exposição de anúncios. SEO; PARK (2017) apontam que em uma pesquisa realizada pela marketingcloud.com, 90% das compras efetuadas, durante o período da pesquisa, foram influenciadas pelas redes sociais. O incremento destes meios, para ESCOBAR (2007, p.2), colocam a interatividade em evidencia, pois por muitas vezes são usadas para ter um melhor contato com o usuário, então, a utilização de novas técnicas mais interativas com seu receptor tornam a leitura menos monótona e com maior possibilidade de obter a atenção do usuário, de forma mais consistente.

Assim sendo, as mídias sociais obtiveram um importante papel para disseminação de informações ou conteúdos, tornando-se uma das ferramentas mais atraentes para divulgações. ROSA (2010) aborda que crescentemente as redes sociais estão no cotidiano das pessoas. Essas são interessantes não apenas por serem um dos meios mais acessados atualmente, bem como possuem a facilidade de interação dos usuários com as notícias.

A evolução do *marketing* trouxe consigo a exibição de propagandas não só nas mídias sociais, mas também em dispositivos móveis. Isso permite que se tenha receptores dos informes em varias localidades, além da possibilidade de interação com essas notícias que estão sendo transmitidas. Para SANTOS (2014), a utilização do *marketing* na *Internet*, favorece a criação desse em ambiente digital.

Para ter maior abrangência, apodera-se do conceito de sinalização digital, que para MACHADO (2010, p.37) dar-se com o uso de telas espalhadas por diversos pontos com diferentes informações que são repassadas via *Internet*. Com isso poderá haver receptores em diversos locais, independente de cidade, estado ou país. As telas ou televisores apresentam informações e propagandas de forma dinâmica, que podem ser gerenciadas remotamente de acordo com a necessidade.

O uso de dispositivos móveis, concomitante com as telas, podem atrair um maior público, pois os dispositivos móveis, com o passar dos anos, estão sendo cada vez mais utilizados pelas pessoas dados que podem ser confirmados por pesquisas feitas pela agência eMarketer (2017) no qual afirmam que até 2019 mais de 80% das pessoas que acessam a *Internet*, usarão o celular para acessa-la. Por isso, é crescente a utilização destes aparelhos como ferramenta para divulgação de informações, não somente pelo grande número de equipamentos, mas também pela integração com as redes sociais que esses oferecem.

Para divulgação de notícias e informações, o Instituto Federal de Educação Ciência e Tecnologia de Brasília – *Campus Taguatinga* (IFB), utiliza-se principalmente de suas páginas *web* e Facebook. Para o uso desses meios, é necessário que os administradores façam publicações independentes para cada uma das páginas, além disso, essas não são bem divulgadas, na maioria das vezes os estudantes e visitantes não ficam cientes das notícias que são publicadas.

Ademais, as únicas formas de contato entre professores e alunos é por meio do e-mail ou com uso de aplicativos externos a instituição, necessitando que cada estudante encaminhe o seu e-mail para que o professor possa entrar em contato, pois o atual sistema da instituição não dispõe de uma funcionalidade, na qual os professores possam obter o e-mail dos alunos automaticamente.

Para PINHEIRO (2010), a comunicação interna é de fato importante para o sucesso de uma organização, em uma instituição educacional não é diferente, é interessante que os alunos e os professores sejam informados de futuros eventos, palestras e notícias. Os atuais veículos institucionais do IFB não possuem integração e atuam de forma independente, o que acabam degradando a qualidade e a disseminação dos informes. Além disso é complicada a interação dos usuários com as notícias, pois essas são expostas de forma descentralizada, ocasionando a baixa interatividade com as informações do IFB. Visto que, para SANTOS (2014), a interatividade é significante pois tem como propósito “estreita o relacionamento com o público”.

O Sistema Integrado de Divulgação de Informações do IFB versão 3 (SIDv3) oferece uma maior visibilidade das notícias, sendo possível, por meio de painéis espalhados pelo *Campus*, uma melhor interação da comunidade com as notícias, apresentando nos painéis os comentários que foram publicados na mídia social, além de uma melhor forma de comunicação entre professor e aluno, oferecido pelo aplicativo móvel que simula um sistema de comunicação institucional, integrado ao Sistema de Gestão Acadêmica (SGA).

1.1 Motivação

BIANCHI; HATJE (2006) abordam que algo inédito e atual deve atrair a atenção do telespectador, sendo esse crucial para o sucesso das notícias que estão sendo exibidas, quanto melhor essas forem disseminadas, maior a chance de sucesso. Nesse intuito, a interatividade e o dinamismo podem ser considerados modernos e o uso de ferramentas do cotidiano das pessoas, despertam e mantém a atenção do usuário, fazendo-o ter interesse em acompanhar e participar das notícias ou matérias.

Atualmente, o IFB utiliza o seu perfil na rede social Facebook, sua página oficial e os murais de cada *Campus* para veicular notícias intrínsecas a instituição, sejam essas referentes a eventos ou institucionais. Para cada notícia é necessário que o administrador acesse as páginas separadamente e realize postagens independentes, além da necessidade de fixação de algumas dessas informações nos murais. Variados meios de divulgação e falta de uma ordem definida de onde cada publicação deve ser postada, pode ocasionar a confusão ou a não localização da mesma.

PINHEIRO (2010) cita a importância da comunicação interna, quando se tem um sistema de divulgação defasado, as informações podem não atingir o resultado esperado, então é preciso uma melhor forma de exposição e edição dessas no IFB, uma forma mais precisa e ágil de apresentação dessas para alunos, professores e visitantes, contando com um sistema mais

interativo, que seja capaz de ter uma participação mais adequada do público com a notícia, como SANTOS (2014) aborda ser importante para o sucesso da informação.

Pensando nisso, verifica-se a indispensabilidade de um sistema onde é possível expor de forma mais precisa, as notícias referentes à instituição. Além de facilitar a criação, edição e exclusão das informações, faça a integração dos meios de comunicação possibilitando a troca do conteúdo em todos os pontos em que está sendo exibido, utilizando um único sistema. Isso permite que os usuários tenham um acesso mais rápido as informações e uma melhor forma de interatividade do espectador com a notícia, disponibilizando o endereço para acesso completo a publicação no Facebook e exibindo os comentários feitos na mesma.

A escolha da rede social Facebook é baseada nas pesquisas realizadas por MUCHARDIE; YUDIANA; GUNAWAN (2016) na qual a apontam como uma das mais utilizadas, a maioria das pessoas que trabalham com *marketing* digital a empregam para a exposição de conteúdos. Não somente pela grande quantidade de usuários que essa detém, mas também por possuir funcionalidades como: compartilhamento fácil e suportar textos com mais de 63000 caracteres, visto que o limite é superior em relação a outras redes sociais como Instagram e Twitter, na qual a capacidade é de aproximadamente 2200 e 140 caracteres, respectivamente.

As pesquisas da eMarketer (2017) abordam que é crescente a utilização de celulares, portanto, um aplicativo para dispositivos móveis que exibe o mesmo conteúdo que são apresentados na rede social e nos televisores, disseminaria melhor as informações. Além disso, uma funcionalidade adicional estaria disponível apenas para estudantes e professores, essa ofereceria uma forma de contato entre os mesmos, observando que atualmente é realizada presencialmente, através de e-mails ou com uso de outros *softwares* complementares.

1.2 Proposta

Com uso da arquitetura cliente-servidor e tendo como base o Sistema Inteligente de Divulgações do IFB em sua versão 2 (SIDv2) implementado por SOBRINHO (2017), é proposto a elaboração da terceira versão. Com o uso dos conceitos de sinalização digital e marketing digital, a proposta é implementar as melhorias para que o sistema seja mais completo na apresentação de conteúdos referentes ao IFB.

O conteúdo criado, com auxílio do sistema, será vinculado a página do IFB no Facebook, sendo exibido nos painéis juntamente com os comentários realizados, devidamente moderados pelos administradores, em telas espalhadas em locais de maior movimento no *Campus* Taguatinga do Instituto Federal de Brasília e nos dispositivos móveis de cada pessoa que possua o aplicativo instalado.

O sistema visa proporcionar a integração dos meios usados atualmente para apresentação das informações referentes ao *Campus*, além da inclusão de outros meios. Portanto, a ideia do SID é prover um meio que a comunidade possa interagir mais com as publicações acadêmicas e o administrador tenha uma maior facilidade de criação e edição das notícias, integrando vários

serviços em um único.

Na versão para dispositivos móveis, além da apresentação das notícias, os professores e os alunos terão acesso a outra funcionalidade: o docente poderá enviar informações e avisos distintos para cada turma que ele leciona, enquanto os alunos poderão acessar cada mensagem enviada pelo professor para a turma em que ele está cadastrado.

A função de envio de mensagem é restrita e se dará através de um login, usando uma matrícula e senha fictícia cadastrados no bancos de dados que simula plataformas acadêmicas já existentes, onde não será possível o uso de dados reais por restrições de acesso as essas plataformas.

1.3 Objetivos

Levantar os principais requisitos de um sistema de sinalização e marketing digital integrado a rede social Facebook.

Estudar e detalhar a documentação da API do Facebook para integração de sistemas.

Definir e implementar o Sistema Integrado de Divulgações do IFB (SIDv3).

Implementar aplicativo móvel com funcionalidades adicionais, além das utilidades providas do SIDv3. Realizando o consumo de um API fictícia do Sistema de Gestão Acadêmica (SGA) visando uma futura integração à plataforma existente quando for disponibilizada.

1.3.1 Objetivos Específicos

- Estudo e utilização de frameworks necessários para implementação do sistema, como ZEND, Doctrine, Cordova e Framework7.
- Disponibilização de uma API REST para interoperabilidade do SIDv3 com outros sistemas.
- Sugestão de uma proposta de implantação viável no *Campus Taguatinga* utilizando computadores Raspberry Pi.

1.4 Metodologia

A revisão de bibliografia será feita como meio de direcionamento do trabalho, onde serão usadas comparações entre ferramentas que apresentam o conceito de sinalização e *marketing* digital, com o objetivo de avaliar as deficiências de cada uma delas, baseando-se nas necessidades do *Campus*.

A análise servirá de forma a definir o que será necessário desenvolver ou alterar, para melhorar a maneira com que as informações são disseminadas, seguindo os conceitos que são considerados primordiais em um sistema de sinalização digital.

O estudo da documentação da Graph API e de suas ferramentas, tais como a Graph API Explorer, viabilizará a melhor integração do sistema com a rede social Facebook. Esses instrumentos serão utilizados para realização de testes práticos das diversas funcionalidades que a API dispõe, selecionando quais serão necessários a implementação, para que seja possível a recuperação e envio de dados e a melhoria no processo de *login*.

O SIDv2 será o sistema base e com auxílio das operacionalidades disponíveis na Graph API, será implementado no sistema as interações com o Facebook. As informações serão apresentadas em multiplataformas que podem ser televisores, painéis, páginas *web* ou celulares. Essas informações podem ser alteradas acessando ao servidor com o sistema instalado e conectado a *Internet*. Após serem criadas ou modificadas, as publicações poderão ser transmitidas e acessadas pelos clientes em distintas plataformas ao mesmo tempo.

A metodologia presente neste trabalho está direcionada ao desenvolvimento de ferramentas que possuem a finalidade de melhoria no processo de comunicação e veiculação de informações e notícias em diferentes plataformas.

Todo o sistema, incluindo o móvel, seguirá o padrão de desenvolvimento ágil, com metodologia SCRUM, sendo definido *sprints* semanais, para definição das funcionalidades a serem desenvolvidas ou melhoradas.

1.5 Organização do documento

O Capítulo 2, tem como objetivo situar o leitor sobre outras ferramentas que possuem conceitos que se assemelham com o SID, expondo como elas funcionam a fim de realizar um comparativo entre as funcionalidades que apresentadas e as que são requeridas.

O Capítulo 3, tem como propósito apresentar ao leitor de forma detalhada cada conceito e ferramenta abordada no decorrer do documento.

No Capítulo 4, Graph API, é apresentado um resumo detalhado das principais funcionalidades que a ferramenta Graph API oferece. Neste capítulo também são colocados alguns exemplos de aplicação destas diferentes funcionalidades.

As explicações detalhadas sobre cada detalhe do sistema, como a estrutura do SID, os detalhes de algumas interfaces e como ele realiza as suas ações em conjunto com a Graph API, apresentada no Capítulo 4, são expostas no Capítulo 5.

Os resultados obtidos estão descritos no Capítulo 6, que aborda de forma resumida todo o resultado final que foi obtido no desenvolvimento do sistema, uma comparação entre os sistemas testados e o SIDv3, além das dificuldades encontradas.

E no Capítulo 7, considerações finais, repassa ao leitor os benefícios de se usar o SID e possíveis melhorias futuras que poderão ser implementadas.

2

Trabalhos Relacionados

Este capítulo tem como propósito apresentar ao leitor algumas ferramentas que fazem uso dos conceitos de sinalização e *marketing* digital, exibindo em telas externas, divulgações criadas pelo administrador no servidor. Para cada ferramenta, serão apresentadas as suas funcionalidades e críticas, expondo suas limitações.

Todas elas seguem a ideia de uma arquitetura cliente-servidor, onde o servidor é responsável por gerenciar os conteúdos que serão exibidos em telas que seguem os conceitos de sinalização digital e possuem o cliente instalado.

2.1 OOZO

Uma solução encontrada na literatura que faz integração entre *marketing* digital e o conceito de sinalização digital é o software OOZO (2017), onde o módulo cliente é multiplataformas e pode ser executado em qualquer sistema que possua navegador de página *web* e instalado em sistemas operacionais Linux, Windows ou macOS para acesso local.

O módulo administrador usado para criação, edição e gerenciamento fica localizado na página oficial do aplicativo, necessitando de login para acesso. Para acessá-lo é possível criar uma nova conta, usar uma conta do Facebook ou Google.

Em sua versão profissional, a mais completa, pode-se sincronizar as redes sociais Twitter, Instagram e Facebook, sendo necessário efetuar login em cada rede social separadamente. Já em sua versão gratuita, alguns recursos não estão disponíveis como o filtro de *hashtag* e a opção de publicar imagens e vídeos, além de exibir nas telas públicas, propagandas de outras empresas juntamente com as suas publicações.

Após a sincronização, o sistema recupera automaticamente *hashtag* do Twitter que foram definidas pelo administrador e postagens do Facebook, Instagram e Twitter, sendo possível limitar o número de itens recuperados. Entretanto, há possibilidade de apenas deletar a publicação do sistema, não sendo possível realizar modificações nas postagens recuperadas, não oferece suporte a enviar uma nova publicação para as redes sociais e não se pode recuperar comentários ou curtidas das publicações.

Todo o conteúdo publicado na ferramenta fica armazenado em servidores proprietários,

onde o módulo administrador é acessado usando uma página *web*, necessitando de acesso para a manutenção e gerenciamento do sistema.

Na página inicial do módulo administrador se pode gerenciar as publicações que já foram postadas nas redes sociais. Selecionando uma dessas publicações, é possível ter o acesso direto a publicação ou excluí-la da ferramenta. Entretanto, ao excluir do sistema, a publicação não é excluída da rede social, além disso, na versão gratuita parte do tempo de exibição das informações é de outras propagandas que não foram criadas pelo usuário.

No módulo cliente, o OOZO não exibe os comentários feitos na publicação da rede social, a ferramenta somente captura de forma automática a foto, o texto publicado e o endereço da publicação e então os envia para exibição. O acesso a notícia completa pelo usuário é feita com a leitura de um *QR code*, que contém o endereço que foi recuperado da publicação no Facebook.

2.2 MangoSigns

A segunda solução encontrada é o MangoSigns (2017), em que o módulo cliente é multiplataformas e pode ser instalado localmente no navegador Chrome, sistema operacional Windows ou dispositivos móveis com sistema Android.

A versão paga possui integração com as principais redes sociais como Twitter, Instagram e Facebook, informações do tempo, entre outras funcionalidades. A versão gratuita não possui integração as redes sociais, oferecendo somente informações básicas como data, envio de imagens e estilos de fonte que serão apresentados no cliente, além de possuir limite de 3 imagens por publicação que será exibida. Além de no módulo cliente não possuir um *QR code* para que o telespectador possa acessar a notícia completa.

A página de administração do aplicativo é acessível através de navegadores, é possível criar uma nova conta, usar uma conta do Facebook ou uma conta do Google. A interface do módulo administrador é usado para criação, edição e gerenciamento de publicações, entretanto, é necessário um dispositivo Android, navegador Chrome ou um dispositivo próprio chamado MangoSing Box conectado a uma TV ou monitor para exibição do conteúdo criado.

As publicações criadas são imagens fixas com blocos adicionados manualmente que sobrepõem as imagens. Elas são exibidas e trocadas automaticamente de acordo com o tempo configurado, sendo possível determinar tempo de transição e até quando a publicação será exibida, além da localização (exclusivo da versão paga).

A integração com as redes sociais é somente para recuperação automática de publicações feitas, não sendo possível recuperar informações como comentários e curtidas feitas na publicação. Em nenhuma de suas versões há a possibilidade de enviar a divulgação criada para as redes sociais a partir da ferramenta. A página do MangoSigns oferece diversas interfaces prontas para criação dos imagens personalizadas, podendo também ser criada uma nova interface.

2.3 SID Formosa

O SID versão 2 (Sistema Inteligente de Divulgação de Informações do IFG-Formosa), é outra solução, dispondo de dois módulos, sendo o módulo administrador, usado para criação, edição e gerenciamento das publicações e o módulo cliente, usado para exibição das notícias criadas.

O acesso ao módulo administrador se dá quando utilizada a conta do Facebook, que esteja com o e-mail cadastrado no banco de dados do SID. Apesar de possuir integração com as redes sociais é limitada a criação de novas publicações no perfil do usuário do sistema, não é possível recuperar nenhuma informação de publicação, comentário ou curtida.

O SID, no módulo cliente, apresenta as informações que lhe são configuradas, recuperando informações previamente armazenadas no banco de dados. As publicações apresentam na tela somente a imagem, um *QR Code* usado pelo usuário caso tenha interesse em acessar a notícia completa e a legenda que foi configurada. SOBRINHO (2017)

Entre os problemas encontrados estão: o link que será apresentado no QRCode deve ser configurado manualmente na criação da publicação, não oferece suporte para configurar data de início da apresentação das publicações, falta de uma documentação detalhada, integração bastante limitada, além de não oferecer nenhum tipo de aplicação móvel.

2.4 Screenly

O Screenly (2017) usa o Raspberry Pi no módulo cliente, necessitando de um programa próprio que deve ser instalado no equipamento para seu funcionamento. Para acesso a área restrita, é possível apenas criar uma nova conta ou logar com uma existente, não sendo oferecido nenhum outro meio de automatização de *login*.

A exibição das publicações é feita por meio de um Raspberry Pi com o uso de um *software* proprietário da Screenly instalado e uma televisão conectada a ele. Todas as versões possuem limitação do número de equipamentos que podem exibir o conteúdo.

Em nenhuma das versões o Screenly possui integração com as redes sociais. Em sua versão gratuita, ele é limitado a criar no máximo duas publicações para serem exibidas de forma sequencial, essas publicações podem ser imagens, links ou vídeos.

2.5 Xibo

Outra solução encontrada é o *software* Xibo (2017), que se trata de um sistema baseado em arquitetura cliente-servidor que permite diversas customizações, no painel de administrador é apresentado diversos blocos, cada item que será apresentado fica dentro de um bloco escolhido. Além disso, ele suporta o envio de diferentes tipos de arquivos, tais como como vídeos, áudios e Flash.

Apesar de suportar diferentes formatos, oferecer *templates* prontos e ser gratuito, ele não possui integração com as redes sociais, não sendo possível recuperar ou enviar para o Facebook qualquer dado das divulgações criadas por ele.

2.6 Comparativo

A Tabela 1 faz um comparativo entre os sistemas citados acima, comparando algumas das funcionalidades consideradas importantes para sistemas que trabalham com a implantação de sinalização digital e *marketing* digital, na qual seus elementos comparativos são descritos a seguir:

- I) Comprometimento com o propósito: o sistema em questão possibilita a veiculação de informações através de mecanismos de sinalização digital?
- II) Criação simples de divulgações: o operador possui facilidade de incluir novas divulgações com aspecto atrativo?
- III) Portabilidade móvel: o módulo cliente está disponível para a plataforma móvel em forma de aplicativo?
- IV) Integração com redes sociais: o sistema realiza integração completa com as redes sociais, oferecendo recuperação e envio de dados.
- V) A versão gratuita explora toda a capacidade do programa?
- VI) É possível criar uma nova publicação e enviá-la para as redes sociais?

Tabela 2.1 Comparativo

Quesito/Sistema	OOZO	MangoSigns	SID v2	Screenly	XIBO
I	SIM	SIM	SIM	SIM	SIM
II	SIM	SIM	SIM	SIM	SIM
III	NÃO	SIM	NÃO	NÃO	NÃO
IV	NÃO	NÃO	NÃO	NÃO	NÃO
V	NÃO	NÃO	SIM	NÃO	SIM
VI	NÃO	NÃO	SIM	NÃO	NÃO

Com o comparativo pode se notar que nenhum dos sistemas descritos atendem as funcionalidades requeridas. Por isso, o sistema SIDv3 tem o objetivo de atender todas as funcionalidades descritas como necessárias na Tabela 2.1. Oferecendo um aplicativo móvel para exibição das publicações, melhor interação acadêmica entre professores e alunos, e uma integração completa com as redes sociais, possibilitando o envio e recuperação de dados das páginas/comunidades do Facebook.

3

Referencial Teórico

Este capítulo tem o propósito de apresentar ao leitor os principais conceitos e ferramentas utilizados para desenvolvimento do sistema. Apresentando de forma superficial o que é cada uma e sua finalidade.

3.1 Marketing Digital

O *marketing* já passou por diversas modificações e melhorias com o passar dos anos, começando com jornais e revistas, passou por panfletos, pôsteres e *outdoors*, teve-se também os rádios, televisões e telefones até atingir hoje o que é conhecido como *marketing* digital. Ele se aproveita da popularização das redes sociais e da crescente facilidade de acesso a *Internet* para aproximar produtores e vendedores as pessoas, que são consumidores, integrando ou expondo produtos ou serviços de maneira simplificada no cotidiano das pessoas.

O advento da *Internet*, obrigou o *marketing* a ser mais digital, inserindo-se em novos contextos, CANTO; CORSO (2017) afirma que o *marketing* digital é uma

“premissa básica para aqueles que desejam estreitar as relações com os clientes”

criando melhores estratégias de *marketing* e inserindo-as em um ambiente digital, integrando as redes sociais para se ter uma ampla divulgação.

RYAN (2016, p.2) relata que o *marketing* digital em 2016 se tornou uma grande fonte de negócios, quando, com a revolução online, trouxe uma nova onda de consumidores, aqueles que integram a tecnologia a sua vida cotidiana. Para ele, publicidade tem tudo a ver com influenciar pessoas e para isso propagandas e tecnologias vem se convergindo cada vez mais para se criar um panorama de *marketing*. A tecnologia e a publicidade se tornaram parte do cotidiano das pessoas.

SANTOS (2014) comenta que em um primeiro momento as estratégias de mercado eram voltadas para meios tradicionais como TVs, rádios, jornais, revistas e outros. Com a popularização da *Internet* foi necessário a busca por outros meios de para se realizar o *marketing*, chegando ao *marketing* digital que é o uso das mídias sociais ou de ferramentas cotidianas, como a *Internet*, para exposição dos produtos.

O *marketing* tem por objetivo, como cita RYAN (2016), persuadir as pessoas a tomar determinadas ações que são desejadas pela divulgação, por exemplo, influenciar alguém a escolher uma determinada marca ou objeto, apenas como aquela apresentação.

Na opinião de TORRERS; COZER (2000), as mídias sociais são páginas de *Internet* onde os usuários são, ao mesmo tempo, produtores e consumidores das informações que nelas são criadas, sendo possível a criação e compartilhamento dessas informações. Ainda para eles, essas mídias receberam esse nome por serem livres e terem a possibilidade de colaboração e interação de todos que nelas estão, além de ser um meio de transmissão das informações e conteúdos.

O *marketing* digital faz uso também das mídias sociais para uma melhor apresentação e disseminação do conteúdo desejado. Como apresenta TORRERS; COZER (2000), a facilidade de exposição e compartilhamento que as redes sociais oferecem, as tornam mais atrativas quando se quer expor algo.

3.2 Sinalização digital

Para MUNARI (2006), a sinalização pode ser feita das mais diversas formas, desde códigos, sinais, luzes, canetas, lápis ou até mesmo por imagem. Ela deve ser melhorada de modo a atingir o objetivo proposto pelo criador, seja ela para propaganda, informativo ou entretenimento, mas sempre com o mesmo objetivo, obter a atenção do telespectador.

Uma pesquisa com 91 telespectadores em 11 diferentes exibições realizada por MULLER et al. (2009), usando painéis públicos e o conceito de sinalização digital, revela que os usuários consideram os anúncios chatos e por muitas vezes o ignoram. Quando o intuito é atrair a atenção do usuário, a interatividade por ser uma grande ferramenta.

Com o uso de programas para gerenciamento do conteúdo, MACHADO (2010, p.31) explica que a sinalização digital consiste na transmissão de conteúdo, via *Internet*, à televisões ou painéis dos mais diversos tamanhos e nos mais diversos locais, usualmente instalados em pontos estratégicos onde as pessoas costumam ficar algum tempo como bancos, ônibus, elevadores. Ele ainda apresenta diversas vantagens e modos de uso para a sinalização digital, em que cada empresa pode optar ou não por ter determinada funcionalidade, mas em geral as vantagens apresentadas são opções como a de atualização instantânea de conteúdo, agendamento de programação, painel para controle, anúncios diferenciados e divisão de tela com diferentes conteúdo.

CINTRA (2010, p.11) relaciona a sinalização digital como alternativa aos *outdoors*, que poluem visualmente grandes cidades. Usualmente colocadas em pontos estratégicos e de grande movimento, tem-se a grande vantagem da economia de tempo e dinheiro. A era da comunicação digital como apresenta a autora, coloca como vantagem a agilidade e a facilidade de troca do conteúdo contido nos painéis, por ser de forma digital e geralmente remota, além disso, é possível a inserção de novos recursos se comparados com outdoors, tais como movimentos nas imagens,

efeitos digitais e som.

De acordo com MISHIMA; SAKURADA; HAGIWARA (2016), sistemas de sinalização digital que possuem algum dispositivo eletrônico são amplamente usados para exibir informações dinâmicas, ao contrários de anúncios estáticos, que ficam por determinado ou muitas vezes logos tempos na mesma informação, o que acaba por muitas vezes, não atraindo a atenção ou interesse de quem os observa.

Entretanto, o uso da sinalização digital não fica restrito a uso externo, para propagandas, ela também pode ser usada para fins internos, para apresentar informações apenas para dentro de uma empresa. Além disso é possível a troca ou atualização de conteúdo em todos os pontos conectados ao mesmo tempo, economizando tempo e mão de obra para reajustar cada ponto específico.

3.3 Arquitetura Cliente-Servidor

Cliente/Servidor é uma arquitetura computacional que envolve requisições de serviços de clientes para servidores. CECIN (2005) explica que nesse paradigma, o servidor é responsável pela computação sobre os dados, enquanto o cliente lida apenas na camada de apresentação.

Uma definição para o que seria a arquitetura Cliente/Servidor, seria a existência de uma plataforma base para que as aplicações, onde um ou mais Clientes e um ou mais servidores, juntamente com o sistema operacional executem um processamento distribuído.

Em resumo, o cliente é um processo que interage com o usuário através de uma interface, enquanto o servidor fornece um determinado serviço que fica disponível para todo cliente que o necessita.

3.4 Linguagens de programação

Para SEBESTA (2011), existe uma lista de potenciais vantagens no estudo dos conceitos de linguagem de programação, entre elas está a da capacidade aumentada para expressar ideias. Ainda para SEBESTA (2011), para as pessoas que não possuem esse conhecimento é mais complicado criar conceitos de estruturas que elas não podem descrever ou expressar.

Na lista feita por SEBESTA (2011) aponta também o embasamento para escolher linguagens adequadas e a habilidade aumentada para aprender novas linguagens como parte vantagem do estudo desses conceitos.

Na linguagem de programação, metodologias de projeto, ferramentas de desenvolvimento e a própria linguagem de programação estão ainda em evolução, por esse motivo SEBESTA (2011) considera o desenvolvimento de software uma profissão excitante, mas que obriga a se ter um aprendizado contínuo.

As linguagens de programação possuem vários conceitos, entre eles estão linguagens de programação estruturada, modular, orientada a objeto, orientada a aspecto e linear. Cada uma

possui sua particularidade e seu grau de abstração que podem ser baixo nível, médio nível e alto nível.

Sendo o padrão seguido por diversas linguagens, a orientação a objeto é usada por diversas linguagens conhecidas, tais como PHP, java e C#. Além do uso de classes e objetos, para RICARTE (2001), um dos grandes diferenciais da orientação a objeto está o conceito de herança que facilita a extensão e o polimorfismo que permite selecionar funcionalidades de forma dinâmica.

As linguagens de baixo nível, são linguagens mais complexas para compreensão humana, onde o código é uma representação direta do código de máquina. O maior exemplo de linguagem de baixo nível é o Assembly.

As linguagens de médio nível, são linguagens intermediárias, que são abstratas, mas com uma facilidade maior de compreensão. Entre os exemplos de linguagem de médio nível estão o C e o C++.

Já as linguagens de alto nível são bem mais inteligíveis pelo ser humano, possuindo baixo nível de abstração. Entre os exemplos desse tipo de linguagem estão o Java e o PHP.

3.4.1 PHP

O PHP é uma das linguagens de programação mais populares no mundo, para VASWANI (2010, p.2), é a ferramenta escolhida por milhões de pessoas no planeta para desenvolvimento de aplicações *Web*. Isso porque o PHP é desenvolvido e mantido mundialmente por uma comunidade voluntária.

Para VASWANI (2010, p.2), as razões pela popularização do PHP como linguagem de programação se dá pelo fato de não ser complicada de se aprender, escalável, de fácil obtenção e muito funcional com aplicações de terceiros. Além disso, desenvolvedores relatam grandes níveis de satisfação como a linguagem.

Além de ser uma linguagem que possui suporte a diversos *frameworks* que auxiliam os desenvolvedores, possuindo compatibilidade com diferentes sistemas de banco de dados, ela ainda é livre e não necessita de licença para uso. Uma pesquisa da W3TECHS (2018), mostra que grandes sites como o do Facebook e Wikipedia foram desenvolvidos nessa linguagem, isso faz com que a linguagem seja mais utilizada, por possuir exemplos de sucesso.

3.4.2 JavaScript

JavaScript - JS é uma linguagem de programação *Web*. BALDUÍNO (2014) comenta que JS se faz presente nas mais diversas aplicações, em seu *backend*, seja jogos, emulação de hardwares ou até como linguagem padrão de plataformas inteiras.

Uma aplicação com JS pode ser usada para invocações AJAX, onde é possível realizar diversos tipos de requisições de dados de um servidor.

3.5 Banco de Dados

Quanto maior é a quantidade de dados, maior é a necessidade do uso de um banco de dados. O uso de papéis para armazenar informações podem necessitar de uma grande quantidade de espaço físico e de um grande controle, para que nada se perda.

Com o uso de um banco de dados, tem-se a possibilidade de se armazenar as mais diversas informações e criar uma relação entre cada uma delas para uma melhor organização.

Para ELMASRI; NAVATHE (2005), é viável afirmar que os bancos de dados representam um papel crítico em quase todas as áreas que os computadores são utilizados. Considerando uma definição genérica usada por ELMASRI; NAVATHE (2005), o banco é uma coleção de dados relacionados, onde os dados podem ser gravados e possuem um significado.

3.6 Interface de Programação de Aplicativos

A API (Application Programming Interface) é um instrumento que visa facilitar o desenvolvimento e a vida dos programadores. Na visão de MAKICE (2009), é uma ferramenta que permite uma aplicação compartilhar dados, onde através de requisições URL, usando métodos HTTP, é possível receber dados estruturados de páginas Web, por exemplo.

Exemplo do uso delas, são redes sociais mais famosas, como Facebook, Twitter e Instagram que possuem suas próprias APIs e são usadas pelos programadores para se ter a integração mais facilmente realizada entre os serviços que as redes sociais oferecem e a sua aplicação.

As APIs das redes sociais por exemplo, oferecem recursos como recuperação e envio de informações, além de outros serviços, como os mecanismos de geolocalização, com uso do GPS.

3.6.1 REST

A *REpresentational State Transfer*(REST), é frequentemente usado em *Web services*, permitindo que os clientes acessem e manipulem os dados através de requisições. ZHOU; LI; LUO (2014) abordam que é possível que clientes requisitem diferentes dados de um mesmo recurso URI, ou seja, requisitar a partir do endereço da página, seguido do recurso que deseja solicitar.

MAKICE (2009) considera que a REST insere escalabilidade e flexibilidade nas aplicações fazendo com que os dados fiquem em camadas, sem estados e bem definidos, portanto, devem esconder detalhes de implementação e não depender de cookies ou sessão.

As requisições de dados podem ter como respostas arquivos do tipo XML, HTML, JSON, entre outras, podendo-se receber um único recurso ou uma coleção deles.

Na arquitetura cliente-servidor, a REST é usada para que os clientes possam realizar requisições de dados aos servidores, seguindo as regras e com uso dos métodos HTTP como GET, POST, DELETE e PUT.

3.7 Linguagem de modelagem unificada - UML

Para GUEDES (2009), a UML é uma linguagem visual para modelagem de softwares. Possuindo um propósito geral, ela é a linguagem-padrão adotada pela engenharia de software para modelagem.

A UML tem o objetivo de auxiliar os desenvolvedores de software a definir as características que o sistema terá, definindo os requisitos, comportamento, estrutura lógica e até mesmo as necessidades físicas como equipamentos e sistemas que serão usados.

Para as definições podem ser separadas em diversos diagramas, como o de classe, casos de uso, de objeto, entre outros.

3.8 Metodologia de desenvolvimento - SCRUM

Existem diversas metodologias de desenvolvimento que são consideradas ágil, onde a SCRUM está inserida. Para SILVA; SOUZA; CAMARGO (2013), o SCRUM se destaca onde os requisitos não estão claros ou mudam com frequência, sendo um processo de desenvolvimento de *software* incremental.

Os processos são divididos em *sprints*, que são as interações que ocorrem durante o desenvolvimento do *software*. Dentro de cada uma dessas *sprints* acontecem reuniões que são colocadas as atualizações do status do projeto.

3.9 JSON

Frequentemente utilizados em *Web services*, o JSON (JavaScript Object Notation) é uma formatação de fácil leitura e escrita por seres humanos, usado para troca de dados e baseado em subconjunto do javascript.

Podem ser estruturados de duas formas, podendo ser como uma coleção de pares relacionando nome e valor ou então uma lista ordenada de valores.

Podendo possuir os quatro tipos básicos de conteúdo: numérico, booleano, caractere e String, o JSON é um modelo para armazenamento e transmissão de informações no formato texto e com capacidade de transmitir um grande volume de dados.

Na Listagem 3.1, é criado uma variável contendo várias informações sobre um usuário, por exemplo.

```
1 $variavel = {  
2     "name": "Fulano",  
3     "age": 50,  
4     "city": "Brasilia",  
5     "inicialNome": "f",  
6     "PossuiAnimal": false  
7 };
```

Listagem 3.1 Exemplo de JSON

3.10 QR Code

Sendo o *QR Code* um código bidimensional de leitura rápida, SOUSA (2014) explica que o seu uso implica em duas etapas, que é a impressão e a leitura do código. A leitura está condicionada ao uso de um *scanner*, estando diretamente ligado ao uso dos dispositivos móveis e a *Internet* para acesso ao conteúdo.

SOUSA (2014) comenta que a ferramenta é cada vez mais usada na estratégia de marketing digital, concebendo a comunicação e a interação entre organizações e público-alvo.

3.11 Frameworks

Frameworks são usado para diversas funções, entre elas estão a de automatizar tarefas repetitivas e a de melhorar a organização e estruturação do código.

No geral, para MINETTO (2007) um *framework* é uma coleção de códigos-fontes, classes, funções, técnicas e metodologias que visam facilitar o desenvolvimento de novos softwares. A seguir, estão descritos alguns dos *frameworks* utilizados.

3.11.1 Zend

Zend Framework é uma coleção de pacotes profissionais para PHP, podendo ser usado na versão 5.6 ou superior do PHP ZEND (2018), ele oferece completa orientação a objeto, boas práticas, reusabilidade, internacionalização, código aberto, suporte da comunidade, boa documentação, entre várias outras vantagens VASWANI (2010)[p.4-5].

VASWANI (2010, p.3) explica que o Zend oferece uma completa implementação do *Model-View-Controller* (MVC), servindo para aplicações de médio e grande complexidade e é geralmente usada no desenvolvimento de aplicações *Web*.

Usado em conjunto com o PHP para estruturação do projeto SID, o Zend oferece a facilidade de separação de módulos, essa funcionalidade foi usada para divisão do projeto em três diferentes módulos, são eles, administrador, cliente e API, onde cada um dos módulos oferece diferentes funcionalidades.

3.11.2 Doctrine

Nas aplicações que armazenam informações no banco de dados é feito uma camada de abstração dos dados do banco, com o gerenciamento de conexões e execução de consultas SQL. O Zend cria um *Object-Relational Mapping*(ORM) usando o doctrine.

O doctrine é usado para obter dados de um banco, usando o ORM ele cria um mapeamento do dados para simplificar as tarefas de consulta VASWANI (2010, p.102).

O doctrine é usado na aplicação para fazer a comunicação entre a aplicação criada e o banco de dados. Com ele é possível fazer a inserção, listagem, edição e exclusão de dados.

3.11.3 Cordova

Cordova é um *framework* mantido pela Apache, disponível para diversos sistemas e é usado para criação de aplicativos multiplataforma, utilizando-se de linguagens *Web* como HTML, CSS e javascript é possível desenvolver para diversas plataformas distintas, seja ela *browser*, Android, iOS e outras PREZOTTO; BONIATI (2017).

No intuito da criação de um aplicativo multiplataforma, o SID para dispositivo móvel foi desenvolvido usando o *framework* Cordova. Com ele é possível criar o executável e posteriormente a instalação do aplicativo desenvolvido nos mais diversos *smartphones*.

3.11.4 Framework 7

Framework 7 - F7 é um *framework* gratuito e de código aberto para desenvolver aplicações híbridas que funcionam em navegadores, Android e iOS com visões e estéticas nativas FRAMEWORK7 (2018).

A criação de uma aplicação híbrida com o F7 é feita usando uma estrutura com HTML, CSS e JS. O HTML é usado para criação dos itens que serão mostrados para o usuário, as telas por exemplo. O CSS é usado para organizar e estilizar o conteúdo criado no HTML. Já o JS é usado para realizar as requisições e transições de tela.

4

Graph Api

Para integração entre o Facebook e outros aplicativos externos é necessário o uso de uma API, sendo que a disponibilizada pela rede social em questão é a Graph. Ela é usada para que aplicativos externos possam realizar as requisições e envio de dados, realizando extração e inserção de informações na rede social, além de possibilitar consulta e gerência dessas informações.

A plataforma do Facebook realiza essas ações através de requisições HTTP, com isso é possível realizar as mais diversas tarefas, entre elas estão a de publicar novas histórias, gerenciar anúncios e carregar fotos FACEBOOK (2018a).

4.1 Visão Geral

A Graph segue um padrão de estrutura que se assemelha a um grafo. Para MELO (2014), um grafo pode ser explicado como um conjunto finito de vértices de modo que cada par de vértices relacionado é chamado de aresta. A estrutura da Graph, segue esse mesmo conceito, possuindo nós, arestas e, adicionalmente, campos.

Os nós representam elementos únicos, cada um deles é considerados um vértice. Na rede social, eles podem ser os mais diversos elementos, como: páginas, usuários, comentários, fotos, entre outros. Cada um desses elementos são considerados um nó distinto FACEBOOK (2018a) e possuem identificadores únicos (ID).

O relacionamentos entre dois vértices são chamadas de arestas e representam a ligação entre os nós. As arestas são conexões entre um objeto único e outro objeto ou coleção de objetos, tais como as fotos de uma página ou o conjunto de comentários em uma foto.

Os campos são os atributos mais específicos dos nós, tais como: o link de uma publicação; as informações de quem realizou o comentário; o tamanho de uma imagem; entre outros.

A obtenção dos elementos é feita através de requisições HTTP e podem ser feitas diretamente do navegador ou usando outras aplicações que usem bibliotecas HTTP. Entretanto, para o seu funcionamento, elas devem seguir a sintaxe correta para requisição e obtenção da resposta do servidor.

As requisições devem ser feitas a conteúdos existentes na rede social e usando os

conceitos de nós, vértices e campos juntamente com as requisições HTTP com métodos GET, POST ou DELETE e um *token* de acesso para que se possa obter os elementos desejados, onde a resposta será um JSON contendo os dados solicitados.

Todas as requisições de dados ao Facebook devem ser feitas a uma URL que serão redirecionadas para um servidor que estarão hospedados todos os dados que podem ser requisitados: `graph-video.facebook.com`, usada para requisições de vídeos e `graph.facebook.com`, usadas para todos os outros tipos de requisições.

A API trabalha de forma a auxiliar o desenvolvedor a realizar requisições, oferecendo diversas classes e métodos, para as mais diversas funcionalidades. Essas classes podem ser usadas para recuperar um vértice ou uma aresta, por exemplo.

Nesse capítulo, serão demonstrados exemplos de requisições e de retorno, esses exemplos foram escritos em código PHP, que atualmente está entre as linguagens suportadas pela SDK do Facebook. A variável `$graphNode` será responsável por receber maioria das respostas das requisições. Enquanto o objeto `$fb` será a instância criada através da SDK para chamada das classes e métodos presentes nele. A Tabela 4.1 apresenta alguns exemplos de nós, a descrição deles e alguns dos nós que cada um pode alcançar.

Tabela 4.1: Exemplos de nós e arestas

Nó	Descrição	Exemplos de Arestas
Usuário	Representa o perfil de um usuário.	Álbuns de fotos, Fotos específicas, Linha do tempo, Amigos
Página	Representa uma página.	Álbuns de fotos, Vídeos publicados, Linha do tempo, Eventos
Postagem	As postagens em um perfil ou página.	Comentários, Curtidas, Reações
Comentário	Comentários feitos em uma postagem.	Comentários, Curtidas, Reações
Álbum	Álbuns criados em um perfil ou página.	Fotos específicas, Capa do álbum, Curtidas

Tabela: 4.1 continuação da página anterior.

Foto	Fotos postadas em um perfil ou página.	Comentários, Curtidas, Reações
Evento	Eventos criados pelo perfil ou página.	Administradores, Comentários, Fotos específicas
Vídeo	Vídeos criados pelo perfil ou página.	Comentários, Curtidas, Reações

4.2 Token de acesso

O *token* de acesso é uma cadeia de caracteres usada para realizar chamadas da Graph API. O tempo de duração deles podem ser curtos ou longos, variando de cerca de uma hora a duração infinita.

Para todas as requisições feitas para a rede social é necessário o uso do *token* de acesso, que serve de maneira a autenticar o usuário sem a necessidade que um novo *login* seja feito a cada requisição, além de identificar o aplicativo, o usuário que executará a ação e quais os dados serão possíveis acessar usando a Graph de acordo com as permissões solicitadas.

Os *tokens* são separados em quatro tipos: *token* de acesso do usuário, usado para alterações de uma conta de usuário específica; os *token* de aplicativo, usado para que as requisições sejam feitas em nome do aplicativo; o *token* de página, usado para realizar modificações em páginas; *token* de cliente, usado em aplicativos moveis e aplicações de computador.

4.2.1 Tokens de usuário

Os *tokens* de usuário são requisitados através da Graph, que por sua vez verifica e provê as devidas permissões que foram solicitadas. A requisição desse *token* é feita através do método “getRedirectLoginHelper()” que retorna um objeto capaz de invocar outros métodos, inclusive o método “getAccessToken()”, usado para solicitar o *token*. Neste método é necessário a passagem de uma string contendo a URL da página inicial do aplicativo, que foi representada como “Pagina de retorno” na Listagem 4.1, nela é realizada uma requisição do *token*, onde a variável \$accessToken conterá um JSON com string com a cadeia de caracteres.

```

1 $helper = $fb->getRedirectLoginHelper ();
2 $accessToken = $helper->getAccessToken (Pagina de retorno);

```

Listagem 4.1 Obtendo Token de acesso a página

4.2.2 *Tokens de aplicativo*

Os *tokens* de aplicativos possuem limitações, restringindo o acesso a conteúdos públicos, não exibindo alguns dados de usuários como as curtidas. A obtenção dele pode ser feito conforme a Listagem 4.2, devendo ser passado o ID único e o ID secreto do aplicativo como parâmetros. O retorno deste tipo de requisição está representado na Listagem 4.3, contendo o *token* e o tipo.

```
1 $fb->get (' /oauth/access_token?client_id={app-id}&client_secret={  
    app-secret}&grant_type=client_credentials');
```

Listagem 4.2 Obtendo *token* de acesso de aplicativos

```
1 {  
2     "access_token": {token},  
3     "token_type": "bearer"  
4 }
```

Listagem 4.3 Retorno *token* de acesso de aplicativo 4.2

4.2.3 *Tokens de página*

Os *tokens* de página dependem de uma permissão específica chamada *manage pages*, além da necessidade do usuário ser administrador da página. Para se obter o *token* é necessário realizar uma requisição de todas as páginas que o usuário é administrador, além do *token* de usuário, como mostra a Listagem 4.4.

```
1 $response = $fb->get(  
2     '/me/accounts',  
3     '{access-token}'  
4 );  
5 $graphNode = $response->getGraphNode();
```

Listagem 4.4 Obtendo Token de acesso a página

O retorno da Listagem 4.4 será um JSON como na Listagem 4.5, contendo uma lista com todas as páginas e as informações referentes a cada uma delas, como o *token* da página, a categoria, o nome, o id e as permissões que o usuário tem sobre página.

```
1 {  
2     "data": [  
3         {  
4             "access_token": {token},  
5             "category": "Education",  
6             "category_list": [  
7                 {  
8                     "id": "2250",
```

```

9         "name": "Education"
10        }
11      ],
12      "name": "IFB",
13      "id": "415358248866659",
14      "perms": [
15        "ADMINISTER",
16        "EDIT_PROFILE",
17        "CREATE_CONTENT",
18        "MODERATE_CONTENT"
19      ]
20    }
21  ]
22 }
```

Listagem 4.5 Retorno *token de acesso de página 4.4*

É possível também obter de uma página específica, ao invés de todas as páginas que o usuário é administrador. Para isso é necessário seguir a Listagem 4.6 que terá como resposta o *access token* e o ID da página específica. onde o retorno será o JSON da Listagem 4.7.

```

1 $response = $fb->get(
2   '/415358248866659',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.6 Obtendo Token de uma única página

```

1 {
2   "access_token": {token},
3   "id": "415358248866659"
4 }
```

Listagem 4.7 Retorno *token de uma única página específica*

4.2.4 *Tokens infinitos*

Os *token* com duração infinita só podem ser solicitados para página ou usuário, para isso é necessário uma requisição com passagem dos parâmetros ID único, ID secreto, e o uso do *token* de usuário de curta duração, obtido na Listagem 4.4. Este tipo de requisição é feita conforme ilustrada na Listagem 4.8. Obtém-se como retorno um JSON contendo o *token* de acesso vitalício.

```
1 $response = $fb->get(            
2     '/oauth/access_token?grant_type=fb_exchange_token&client_id={ ID  
3         Unico}&client_secret={ ID secreto}&fb_exchange_token={Token} ',  
4     '{access-token}'  
5 );  
$graphNode = $response->getGraphNode();
```

Listagem 4.8 Obtendo Token Infinito

Cada linguagem possui a sua forma específica de obter o *token* por meio do uso da SDK específica de cada uma. Além da chamada usando o SDK, para gerar o token, com exceção do de aplicativo, é necessário um usuário autenticado.

4.3 Autenticação

O Facebook disponibiliza diversas ferramentas, dentre elas está a que viabiliza o login com a rede social, chamada “login com Facebook”. Ela segue os padrões de conformidade do protocolo OAuth 2.0, que provê um fluxo de autorização específica para aplicações *Web*, telefones móveis, entre outras OAUTH (2018).

A ferramenta é disponibilizada pela rede social e oferece um sistema de autenticação multiplataforma e controle de acesso, através da análise de permissões, definindo as operações que o usuário poderá usar FACEBOOK (2018b). Ela funciona de forma a autenticar o usuário do aplicativo por meio de uma conta vinculada a rede social.

Para utilizar esta ferramenta é necessário o envio de alguns parâmetros de identificação do aplicativo, tais como, o *app_id*, *app_secret*, *default_graph_version*, *fileUpload*, entre outros.

- O *app_id* e o *app_secret* são obrigatórios, os dois são IDs únicos para cada aplicativo vinculado a rede social, esses IDs são gerados pelo Facebook no momento da criação do aplicativo, entretanto, o *app_id* é publico, enquanto o *app_secret* é secreto.
- O parâmetro *default_graph_version*, não é obrigatório. Ele irá identificar qual versão da Graph o seu programa irá usar. Caso não seja passado como parâmetro, o Facebook irá usar a última versão da API lançada.
- O *fileUpload* não é obrigatório. Ele é o parâmetro necessário para informar se será enviado arquivo de imagem ou não. Essa imagem será exibida na publicação no Facebook.

Na Listagem 4.9, os parâmetros necessários para validação são passados em um *array* e armazenados na variável *\$newFacebook*. Os parâmetros passados na variável são enviados para a API que fará a conexão com o Facebook. Após a conexão com o SDK é possível a realização de

requisições, elas podem ser do tipo *GET*, *POST* ou *DELETE* e serão feitas a partir de chamadas da SDK.

```

1 $newFacebook = array (
2     'app_id' => {ID},
3     'app_secret' => {ID},
4     'default_graph_version' => v2.10,
5     'fileUpload' => true;
6 )
7 $fb = new \Facebook\Facebook ( $newFacebook );

```

Listagem 4.9 Conexão entre aplicativo e SDK

Para realização de *login* é necessário também o uso do método *getRedirectLoginHelper()*, para que seja possível invocar o método “*getReRequestUrl()*”, usado para solicitar do usuário as permissões que o aplicativo requisita. Nesse segundo método é necessário a passagem da URL que a página do Facebook irá retornar após o *login* e as permissões necessárias. O *login* é possível usando a Listagem 4.10, onde a variável *\$permissions* conterá as permissões que serão solicitadas.

```

1 $helper = $fb->getRedirectLoginHelper ();
2 $permissions = [
3     'email',
4     'manage_pages',
5     'publish_pages'
6 ];
7 $loginUrl = $helper->getLoginUrl ( 'Pagina de retorno',
8                                     $permissions );

```

Listagem 4.10 Solicitação de Login

4.4 Permissões

Um sistema de permissões é utilizado na Graph API para controle de acesso e manipulação de conteúdo, sendo possível o controle de publicação e de edição de informações, por exemplo. Assim, para que alguma modificação na publicação ou solicitação de dados por parte do modulo administrador possa ser efetivada, é necessário possuir as permissões adequadas. As permissões funcionam de forma a autorizar o acesso a um determinado conteúdo, onde o sistema poderá solicitar ao usuário uma nova permissão somente quando for necessário o uso dela.

As permissões descrevem quais as possíveis ações podem ser feitas em cooperação com a Graph API, elas determinam quais tipos de dados pode-se gerenciar e quais as possíveis respostas a API pode retornar. A forma de solicitar permissões está descrita na Listagem 4.10,

onde a variável \$permissions possui diversas permissões que serão solicitadas no momento que o usuário for realizar o login.

O Facebook oferece diversas permissões, elas podem ser de leitura ou de escrita, cada uma poderá ser usada para se obter um determinado acesso a um determinado dado. As permissões podem ser usadas para que a Graph acessar dados específicos quando solicitado, tais como as permissões de email, *user_birthday*, *user_friends*, usadas para recuperar email, data de aniversário e amigos, respectivamente de um determinado usuário.

Não somente os usuários, as permissões também abrangem as páginas do Facebook, por exemplo, pode-se usar a *manage_pages*, a *publish_pages*, usadas para gerenciar e criar novos conteúdos para as páginas, respectivamente.

Algumas permissões, como as de gerenciamento de páginas, necessitam que pré-requisitos sejam atendidos para que a solicitação da permissão seja possível. Por exemplo, a *manage_pages* necessita que o usuário que está efetuando o *login* seja administrador da página.

4.5 Requisições

As requisições seguem um padrão, onde em todas elas serão necessárias a passagem do tipo, do ID e do *token* de acesso. Entretanto, é possível obter dados mais específicos, para isso é inserido novos parâmetros como as arestas ou os campos.

Em uma requisição é possível recuperar diferentes dados um nó específico, seguindo a Listagem 4.11. A partir de um nó, é possível encontrar outros nós, para isso são utilizadas as arestas, como na Listagem 4.12. Mas existe a possibilidade de solicitar apenas dados específicos, definindo os campos como na Listagem 4.13.

```
1 $fb->get ( 
2   '{ ID }',
3   '{ access-token }' );
```

Listagem 4.11 Requisição com o uso do ID

```
1 $fb->get ( 
2   '{ ID }/{aresta}',
3   '{ access-token }' );
```

Listagem 4.12 Requisição com o uso do ID+aresta

```
1 $fb->get ( 
2   '{ ID }?field={campo}',
3   '{ access-token }' );
```

Listagem 4.13 Requisição com o uso do ID+campo

4.5.1 Tipos de Requisição

Os métodos utilizados na GRAPH API estão em conformidade com uma arquitetura REST, de modo que GET é utilizado para recuperação de informações, POST para inserção de novas informações e DELETE para remoção de informações.

Para envio ou edição de dados é necessário o uso do POST, ele pode ser usado para realizar uma nova publicação, envio de um comentário, entre outros. Na Listagem 4.14 é realizada uma nova publicação na linha de tempo de um perfil ou página, com a passagem do parâmetro “message”, contendo o texto que será apresentado. O retorno será um JSON contendo somente o ID da publicação criada.

```

1 $response = $fb->post (
2   '{ID}/feed',
3   array (
4     'message' => {mensagem}
5   ),
6   '{access-token}'
7 );
8 $graphNode = $response->getGraphNode();

```

Listagem 4.14 Requisição POST

Para deletar uma publicação ou comentário é necessário o uso do DELETE, com a passagem dos parâmetros ID e *token* de acesso, como na Listagem 4.15, onde o retorno será um JSON, contendo uma variável “success” do tipo booleano, contendo verdadeiro ou falso.

```

1 $response = $fb->delete (
2   {ID},
3   array (),
4   '{access-token}'
5 );
6 $graphNode = $response->getGraphNode();

```

Listagem 4.15 Requisição DELETE

4.5.2 Nô

Os nós representam os vértices de um grafo, eles podem ser os mais diversos elementos. Cada nó possui um identificador próprio (ID) único. Nos exemplos apresentados abaixo, são demonstradas algumas possibilidades de requisições de um nó.

Para requisições feitas usando somente o nó, geralmente necessita-se somente que dois parâmetros sejam repassados, que é o ID e o *token*. Esses parâmetros devem ser repassados para a Graph, através da chamada dos métodos do objeto \$fb, com esse objeto é possível instanciar as diferentes classes e métodos disponibilizadas pela API. Além disso, a variável \$response

receberá o retorno desses requisições e a variável \$graphNode receberá o conteúdo estruturado de acordo com o método requisitado, que podem ser “getGraphNode()” ou “getGraphEdge()”.

O método GraphEdge() é usado para retornar conteúdos referentes a aresta, enquanto a classe GraphNode() é usada para retornar conteúdo referentes aos nós, podendo referenciar os métodos asArray, asJson e getField que são usados para representar respectivamente um retorno em formato de ARRAY, em formato JSON ou um campo específico em formato ARRAY.

Os dados que são retornados são os campos solicitados durante a requisição e podem variar de acordo com os parâmetros que são colocados. Entre os principais retornos estão os campos *name*, ID, *created_time*, entre diversos outros.

Alguns parâmetros de retorno podem ter definições diferentes para cada tipo de requisição. Um exemplo disto é o parâmetro “name”, em que na requisição de informações de usuário ele contém o nome de usuário ou página, enquanto na requisição de postagem, esse mesmo parâmetro é a descrição ou texto da postagem. Outros já são padrões, com o ID que é o identificador único do nó e o “*created_time*”, que é a data da criação da publicação, comentário ou evento.

O próprio utilizador da rede social, que está autenticado e está usando a API, também é considerado um nó. Na Listagem 4.16, é requisitado informações do próprio utilizador. O parâmetro /me se refere ao usuário, nesta requisição é obtido o mesmo retorno que a requisição da Listagem 4.17.

```
1 $response = $fb->get (
2   '/me',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.16 Requisitar informações do próprio usuário

Com exceção da requisição anterior, elas devem seguir uma sintaxe, contendo ao menos um valor numérico que representa o ID e um conjunto de caracteres que representa o *token* de acesso. Tanto o ID, quanto o *token* de acesso, são variáveis do tipo string e são representados respectivamente nas linhas 2 e 3 nos exemplos de requisições das Listagens 4.17 a 4.26.

É possível requisitar diversas informações de um usuário ou página quando se possui o ID deles, nos exemplos das Listagens 4.17 e 4.18 são apresentados requisições de informações básicas de um usuário e de uma página, respectivamente. O conteúdo de retorno será estruturado da forma apresentada na Listagem 4.19

```
1 $response = $fb->get (
2   '/1371436046298678/',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.17 Requisitar informações de um usuário específico

```

1 $response = $fb->get(
2   '/415358248866659/',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.18 Requisitar informações de uma página

```

1 {
2   "name": "Maxwell Borges",
3   "id": "1371436046298678"
4 }

```

Listagem 4.19 Resposta do servidor as requisições 4.16, 4.17 e 4.18 (Usuário e Página)

Para que dados básicas de uma postagem sejam obtidos, é necessário a mesma sintaxe da requisição feita para dados de usuários, como é apresentado na Listagem 4.20. O retorno será estruturado conforme a Listagem 4.21.

```

1 $response = $fb->get(
2   '/511846152551201/',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.20 Requisitar informações de uma postagem específica

```

1 {
2   "created_time": "2018-05-21T20:15:54+0000",
3   "name": "Bem vindo! Esse é o SID, integracao, facilidade e
4     interatividade em um unico sistema.",
5   "id": "511846152551201"
}

```

Listagem 4.21 Resposta do servidor a uma requisição 4.20 (Postagem)

Sendo mais específico ainda é possível requisitar informações como o comentário feito em uma postagem de uma página. Na requisição da Listagem 4.22 é solicitado dados básicos de um comentário. O retorno será estruturado da forma da Listagem 4.23.

```

1 $response = $fb->get(
2   '/511846152551201_515868852148931/',
3   '{access-token}');

```

```
4 $graphNode = $response->getGraphNode();
```

Listagem 4.22 Requisitar informações de um comentário específico

```
1 {
2     "created_time": "2018-05-30T14:00:17+0000",
3     "from": {
4         "name": "Maxwell Borges",
5         "id": "1371436046298678"
6     },
7     "message": "Ola",
8     "id": "511846152551201_515868852148931"
9 }
```

Listagem 4.23 Resposta do servidor a uma requisição 4.22 (Comentário)

É possível obter dados de eventos e de álbuns criados por uma página. Uma forma de requisitar esses dados é apresentado nas Listagens 4.24 e 4.26. Onde o retorno será como apresentado na Listagem 4.25

```
1 $response = $fb->get(
2     '/383399765490475',
3     '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.24 Requisitar uma evento específico

```
1 {
2     "description": "Sera um evento bem legal",
3     "end_time": "2018-06-02T19:00:00-0300",
4     "name": "Aula",
5     "place": {
6         "name": "por ai"
7     },
8     "start_time": "2018-05-30T16:00:00-0300",
9     "event_times": [
10         {
11             "id": "383399778823807",
12             "start_time": "2018-05-30T16:00:00-0300",
13             "end_time": "2018-05-30T19:00:00-0300"
14         }
15     ],
16     "id": "383399765490475"
17 }
```

Listagem 4.25 Resposta do servidor a requisição 4.24 (Evento)

```

1 $response = $fb->get(
2   '/420148828387601',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.26 Requisitar uma álbum específico

É possível chegar a cada um desses vértices usando apenas o ID do nó desejado, um *token* de acesso e permissões específicas para acesso a cada um dos nós, essas permissões foram explicados na Seções 4.2 e 4.4, respectivamente.

4.5.3 Aresta

As arestas são as ligações entre os vértices, elas representam a conexões entre um objeto único e uma coleção de objetos. Na rede social, como quase tudo é considerado um vértice, é possível buscar as arestas de cada um deles. Os vértices podem ser o conteúdo de uma página, postagem, comentário, entre outros.

As arestas são usadas para alcançar outros nós. Por exemplo, uma postagem é um vértice e um comentário da postagem é outro vértice. Para se obter o ID do comentário, é necessário uma requisição contendo o vértice que é o ID da postagem e o parâmetro comentários que é a aresta que permite alcançar o vértice correto, seguindo a sintaxe da Listagem 4.12.

Através de uma aresta de um vértice, é possível alcançar uma coleção de outros vértices.

Para obtenção de arestas, assim como na de vértices, é necessário o uso de no mínimo um ID e de um *token* de acesso, com incremento de um novo parâmetro junto ao ID. Esse novo parâmetro é o elemento mais específico que se deseja obter partindo de um nó. Nas Listagens a seguir, ele está representado na linha 2, após o ID.

Para obtenção das publicações de um usuário ou página, é necessário seguir a Listagem 4.27, nesta requisição o retorno será uma lista estruturada conforme a Listagem 4.28 apresenta.

```

1 $response = $fb->get(
2   '/1371436046298678/feed',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.27 Requisitando todas as publicações de um usuário

```

1 {
2   "data": [

```

```

3   {
4     "created_time": "2018-05-30T18:03:41+0000",
5     "message": "Sera um evento bem legal",
6     "story": "IFB added an event.",
7     "id": "415358248866659_383399765490475"
8   },
9   {
10    "created_time": "2018-05-21T20:15:54+0000",
11    "message": "Bem vindo! Esse e o SID, integracao, facilidade e
12      interatividade em um unico sistema.",
13    "id": "415358248866659_511846152551201"
14  }
15 }
```

Listagem 4.28 Resposta da requisição 4.27 (Feed)

Com as arestas é possível obter também a lista de álbuns (4.29), vídeos (4.30), eventos (4.31) e fotos de perfil (4.32) que foram criados pelo usuário ou página. Para isso, é necessário o uso de um parâmetro específico para cada um, sendo o ID com o incremento de respectivamente “/albums”, “/videos”, “/events” ou “/photos”.

```

1 $response = $fb->get(
2   '/415358248866659/albums',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.29 Requisitar todos os álbuns de uma página

```

1 $response = $fb->get(
2   '/415358248866659/videos',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.30 Requisitar os vídeos publicados na página

```

1 $response = $fb->get(
2   '/415358248866659/events',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.31 Requisitar os eventos agendados pela página

```

1 $response = $fb->get(
2   '/415358248866659/photos',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.32 Requisitar as fotos de perfil publicadas na página

Sendo mais específico na busca de uma nó, é possível também obter a lista com dados de todos os comentários de uma postagem (4.33) ou todos as curtidas de um comentário (4.34). Isso é possível com o uso do ID, com o incremento dos parâmetros “/comments” ou “/likes”, respectivamente.

```

1 $response = $fb->get(
2   '511846152551201/comments',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.33 Requisitar todos os comentários de uma postagem em uma página

```

1 $response = $fb->get(
2   '511846152551201_515868852148931/likes',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();

```

Listagem 4.34 Requisitar todas as curtidas de um comentário

Assim como nas requisições de nós, as requisições de arestas também retornam em formato estruturado. Com a diferença de que é uma coleção de objetos, podendo conter nenhuma ou diversas posições. Cada item desta coleção fica armazenado em uma posição do vetor “*data*” conforme a Listagem 4.35 que apresenta a resposta a requisição da Listagem 4.30. As resposta das Listagens 4.29 e 4.32 é similar a da Listagem 4.30, com conteúdo específico de cada requisição.

```

1 {
2   "data": [
3     {
4       "created_time": "2017-10-24T01:22:51+0000",
5       "name": "Timeline Photos",
6       "id": "420148828387601"
7     },
8     {

```

```
9     "created_time": "2017-10-10T22:45:42+0000",
10    "name": "Profile Pictures",
11    "id": "415358282199989"
12  }
13 ]
14 }
```

Listagem 4.35 Resposta das Listagens 4.29, 4.32 e 4.30 (Álbuns, Fotos e Videos)

```
1 {
2   "data": [
3     {
4       "created_time": "2018-05-30T18:03:41+0000",
5       "message": "Sera um evento bem legal",
6       "story": "IFB added an event.",
7       "id": "41535824886659_383399765490475"
8     },
9   ],
10 }
```

Listagem 4.36 Resposta da requisição 4.31 (Eventos)

```
1 {
2   "data": [
3     {
4       "created_time": "2018-04-09T17:27:04+0000",
5       "from": {
6         "name": "Maxwell Borges",
7         "id": "1371436046298678"
8       },
9       "message": "Esse e um comentario",
10      "id": "487816184954198_492696814466135"
11    },
12    {
13      "created_time": "2018-04-13T12:03:17+0000",
14      "from": {
15        "name": "Maxwell Borges",
16        "id": "1371436046298678"
17      },
18      "message": "Comentario 2",
19      "id": "487816184954198_494448427624307"
20    }
21  ]
```

22 }

Listagem 4.37 Resposta da requisição 4.33 (Comentários)

```

1 {
2   "data": [
3     {
4       "id": "1305648676232580",
5       "name": "Fernando Nascimento"
6     },
7     {
8       "id": "1371436046298678",
9       "name": "Maxwell Borges"
10    }
11  ]
12 }
```

Listagem 4.38 Resposta da requisição 4.34 (Likes)**4.5.4 Campo**

Já os campos, são usados para representar os dados de um objeto específico, dados que serão incluídos na resposta. Como exemplos de campos, temos: data de aniversário de um usuário ou nome de uma página. Os atributos de campo, podem ser os mais diversos, tais como: *comments,likes,link*, entre outros.

Esses parâmetros são colocados juntamente com o ID do objeto que se deseja obter os dados, eles são colocados referenciados com o o incremento do texto “?fields=”, onde cada elemento seguinte representará o dado específico que se deseja obter. Nas Listagens 4.39 e 4.40, esses parâmetros são passados na linha 2.

Na Listagem 4.39, é feito a requisição dos comentários presentes nas publicações feitas na *timeline* de uma página.

```

1 $response = $fb->get(
2   '/415358248866659/feed?fields=comments',
3   '{access-token}'
4 );
5 $graphNode = $response->getGraphNode();
```

Listagem 4.39 Requisitar os comentários de todas as publicações da página

Em posse de um ID e com a mesma requisição, é possível obter múltiplos dados, como: *link* e o as curtidas de uma publicação específica. Na Listagem 4.40, é solicitado mais de um parâmetro para a mesma publicação.

```
1 $response = $fb->get(  
2   '/415358248866659_511846152551201?fields=comments,likes,link',  
3   'access-token'  
4 );  
5 $graphNode = $response->getGraphNode();
```

Listagem 4.40 Requisição de múltiplos campos

O retorno das requisições com campo segue o mesmo padrão que para requisições de nós ou de arestas, com o diferencial do campo solicitado.

```
1 {  
2   "comments": {  
3     "data": [  
4       {LISTA COM COMENTARIOS}  
5     ],  
6   },  
7   "likes": {  
8     "data": [  
9       {LISTA COM CURTIDAS}  
10    ],  
11  },  
12  "link": "LINK DA PUBLICACAO",  
13  "id": "ID DA PUBLICACAO"  
14 }
```

Listagem 4.41 Retorno da requisição 4.40 (Campos)

5

SID

O Sistema Integrado de Divulgação de informações do IFB *Campus Taguatinga* - SID, tem como principal objetivo fornecer informações através de sinalização em sua forma digital, oferecendo de forma fácil o gerenciamento dessas informações que serão repassadas.

Neste Capítulo, será feito o detalhamento do SID, primeiramente apresentando os detalhes de forma superficial, macroscópica. Posteriormente é apresentado de maneira microscópica, apresentando cada detalhe de como e onde são realizadas as demandas de cada elemento presente no sistema.

5.1 Visão Geral

O SID está estruturado em duas vertentes, o *Web* e o móvel, ele foi desenvolvido com o objetivo de oferecer um sistema de divulgação que realiza integração com as redes sociais de forma ágil, intuitiva, dinâmica e amigável para os administradores e para os usuários comuns, além de melhorar a efetividade do processo de disseminação das informações referentes ao IFB *Campus Taguatinga*. Atendendo ao objetivo principal, que é divulgação de informações através de uma plataforma *Web* e móvel.

Necessitando sempre do uso da rede para realizar atualizações, o SID está dividido em dois módulos e um aplicativo. O primeiro módulo é o administrador, onde é possível fazer o gerenciamento completo do conteúdo que será apresentado no segundo módulo chamado cliente. No segundo módulo será apresentado as informações que foram cadastradas no módulo administrador, onde então serão propagadas por monitores ou celulares. O aplicativo é um sistema móvel, onde o usuário poderá ter acesso a todas as divulgações assim como no módulo cliente, mas com o incremento de troca de mensagens entre professores e alunos através do consumo de uma API fictícia que simula a interface a ser oferecida pelo Sistema de Gestão Acadêmica do IFB (SGA).

A divisão de módulos foi feita para que se possa atender a arquitetura cliente-servidor, ela se mostrou necessária para minimizar o processamento do módulo cliente e do aplicativo que se utiliza do módulo cliente, centralizando o processamento das informações em um sistema mais robusto, ficando a cargo dos clientes exibirem as informações recebidas e realizar pequenos

processamentos.

O sistema WEB foi desenvolvido na linguagem PHP, usada para estruturação do projeto, a linguagem JavaScript para realização das requisições de trocas de informações entre os módulos e as linguagens HTML e CSS para desenvolvimento das telas que serão apresentadas no sistema. Usa-se também o Banco de dados PostgreSQL para armazenamento e persistência das informações.

Baseado em desenvolvimento ágil, com metodologia SCRUM, foram definidos sprints semanais, para definição das funcionalidades a serem desenvolvidas ou melhoradas. Esta metodologia se torna importante para que o progresso seja acompanhado em cada parte do seu desenvolvimento.

No desenvolvimento, foram usados alguns *frameworks*. O primeiro deles, usado em conjunto com o PHP, foi o ZEND que tem a finalidade de estruturar o código em modelo, visão e controle. Outro *framework* foi o Doctrine, usado para efetivação da comunicação entre o banco de dados e a orientação a objeto. O terceiro foi o Framework7 (F7), em conjunto com o Cordova, para que fosse possível o desenvolvimento do aplicativo móvel.

Se utilizando dos conceitos de UML abordados na Seção 3.7, é feito a representação dos elementos, demonstrando cada artefato do SIDv3, como o diagrama de classe, de sequência e seus casos de uso.

Uma apresentação mais detalhada do funcionamento interno de cada tela disponível do SID é apresentada no Apêndice A.

5.2 Modulo Administrador

Com o uso da arquitetura cliente-servidor, esse módulo fica situado no servidor e é responsável por conceder ao usuário administrador a funcionalidade de gerenciar todas as informações do sistema podendo inserir, alterar, e retirar as divulgações, fazendo todo o gerenciamento das informações que são repassadas ao módulo cliente e móvel.

O modelo de casos de uso que representa o módulo é apresentado na Figura 5.1. Ele tem a finalidade de demonstrar os principais processos que o administrador pode executar no módulo.

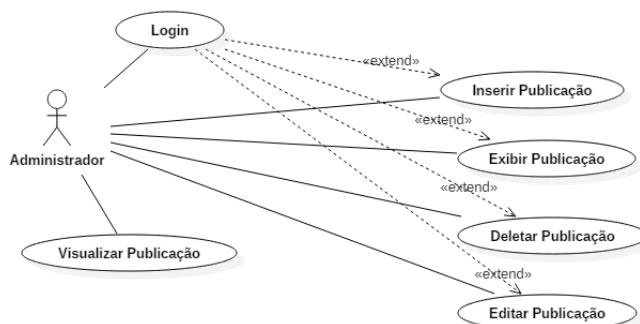


Figura 5.1 Casos de uso da ações do módulo administrador

O acesso ao módulo é restrito, sendo necessário autenticação e para esse fim foi escolhido a ferramenta “*login com o Facebook*”, que a própria rede social disponibiliza como explicado na Seção 4.3. Optou-se pelo uso dela, pois provê maior segurança, sendo baseada em sistemas criptográficos e também por facilitar o processo de recuperação de alguns dados básicos que são necessários para o processo de autenticação com o sistema. O diagrama de sequência que representa o processo de autenticação está representado na Figura 5.2.

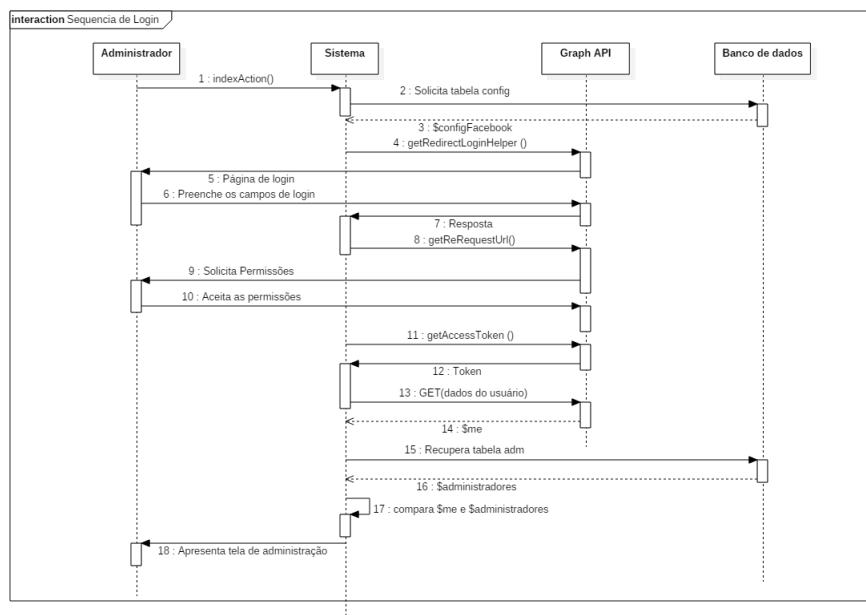
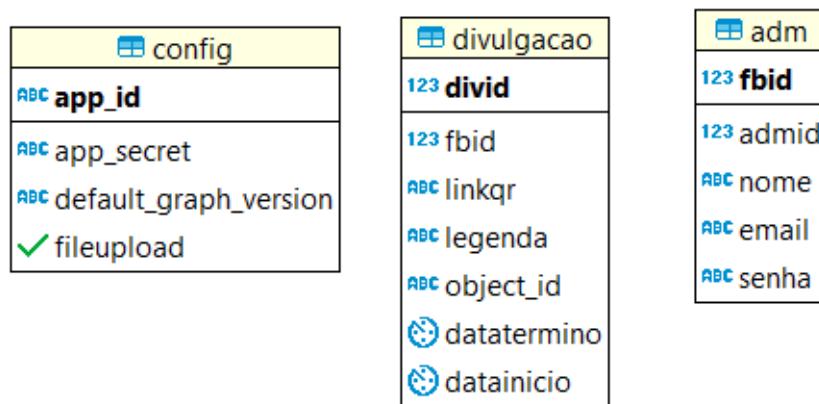
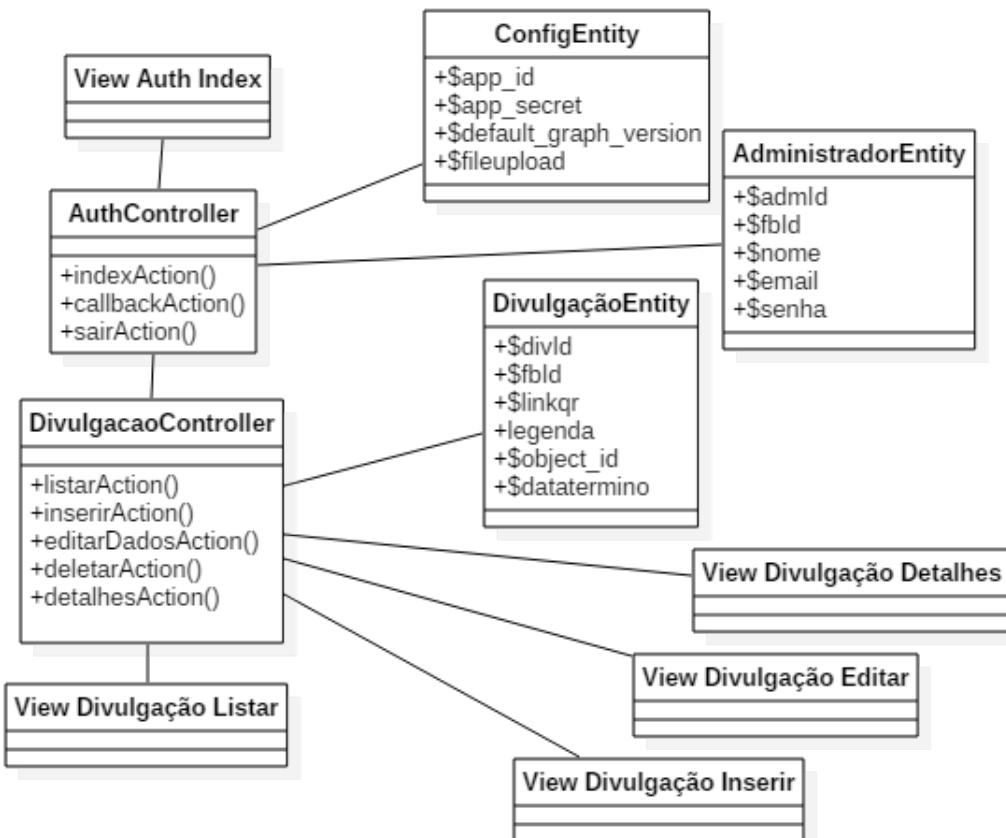


Figura 5.2 Sequencia para efetivação de *login*

A estruturação do banco de dados é composta por três tabelas, sendo mapeadas com as entidades “config”, “divulgacao” e “adm”. A função da tabela “divulgacao” é armazenar todos os objetos relacionados as divulgações que serão exibidas no cliente e no aplicativo. A tabela “adm” é usada para armazenar as informações referentes aos administradores do sistema. A tabela “config” é destinada a armazenar informações únicas para efetivação da comunicação entre aplicativo e Graph API.

**Figura 5.3** Modelo entidade e relacionamento

A Figura 5.4, mostra a representação e o relacionamento entre as classes disponíveis no módulo. No diagrama de sequência 5.5 é exposta a sequência de uso das classes ao inserir uma nova publicação no sistema.

**Figura 5.4** Diagrama de classe do módulo administrador

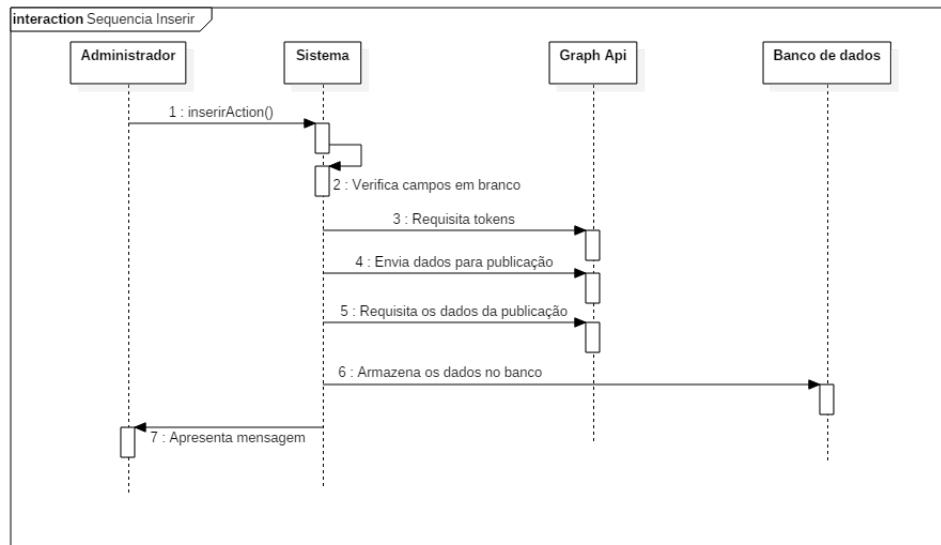


Figura 5.5 Diagrama de sequência para inserção

Após a efetivação da autenticação, o usuário poderá ter acesso a todas as funcionalidades disponíveis no sistema, divididas entre as telas de “inserir” e “Home”. A tela de inserir, é usada para criação de uma nova publicação, enquanto a tela Home oferece a opção de listar, detalhar, deletar e editar essa publicações que foram criadas.

A Figura 5.6, representa a visão do operador da tela de inserção, onde cada elemento da página está enumerado.

A interface de usuário para a inserção de uma nova divulgação. Ela contém os seguintes elementos, todos numerados:

- Legenda: Um campo de texto com placeholder "Texto para exibida na publicação".
- Text area: Um grande campo de texto para inserir o conteúdo da publicação.
- Data de inicio da divulgação: Um campo com três dropdowns para selecionar Dia, Mês e Ano.
- Data de término da divulgação: Um campo com três dropdowns para selecionar Dia, Mês e Ano.
- Imagem: Um campo com placeholder "Escolher arquivo" e uma sublegendinha "Nenhum arquivo selecionado".
- Botões: Um botão "Cancelar" e um botão verde "Inserir".

Figura 5.6 Página de inserção do módulo administrador.

Os elementos presentes na tela de inserir são listados abaixo, juntamente com suas funcionalidades:

1. Legenda: É um campo de texto destinado a informação a ser repassada de forma sucinta e clara ao usuário final por intermédio cliente ou móvel.
2. Texto: É todo o texto que será enviado ao Facebook, contendo informações como descrição da publicação, data e horário da realização e um possível *link* para acesso a mais informações.
3. Data de Início: É a data inicial em que a publicação começará a ser exibida no cliente e no móvel.
4. Data de Término: É a data final em que a publicação deixará de ser exibida no cliente e no móvel.
5. Imagem: Será a imagem enviada para o Facebook, para o cliente e para o móvel. Essa imagem pode ser um *banner* de apresentação de um evento, por exemplo.

Todos os campos são de preenchimento obrigatório. O campo 5 é repassado para a API como parâmetro *source*, já o campo 2 é repassado como parâmetro *messagem*, os outros campos são enviados somente para o banco de dados.

Se durante o envio alguma das transações não forem efetivadas, será retornado a mensagem de erro na tela, assim como retornará a mensagem de sucesso, caso não ocorra nenhum erro.

O texto do campo 1 poderá conter letras ou números, entretanto possui a limitação máxima de 80 caracteres, isso é necessário para facilitar a leitura do usuário ao texto que será inserido, pois ele será armazenado no banco de dados e recuperado no módulo cliente para exibição.

Os outros campos não possuem limitação de caracteres, entretanto os campos 3, 4 e 5 possuem restrição de tipo de entrada, onde os campos 3 e 4 obrigatoriamente será um número inteiro e o campo 5, será um arquivo de imagem do tipo JPEG, JPG, GIF ou PNG.

Os campos 3 e 4 são botões do tipo dropdown, onde é preciso selecionar um dos valores que aparecem em cada um dos campos. Os possíveis valores para o campo “dia” vão de 1 a 31, os valores do campo “mês” vão de 1 a 12, enquanto os valores do campo “ano” irão do ano atual a dois anos seguintes.

A página “Home” é responsável por listar todas as publicações criadas, a visão do usuário será a mesma que representada na Figura 5.7. As funcionalidades de cada elemento da página são descritas logo abaixo. Essa é a página inicial do sistema, nela é possível ter acesso a página de inserção, listagem, detalhamento e ao botão excluir.

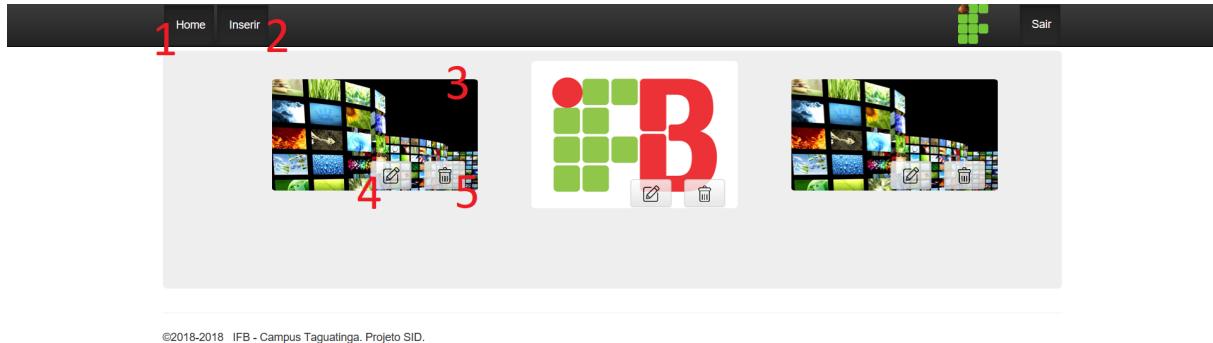


Figura 5.7 Página de listagem do módulo administrador.

1. Home: Botão para acesso a página inicial, onde são listadas todas as publicações criadas.
2. Inserir: Botão para acesso a página de inserção.
3. Detalhes: Ao clicar sobre a foto é possível obter dados específicos da publicação selecionada, tais como: Legenda, ID do autor, link de acesso a publicação, data de início, data de término, ID da publicação.
4. Editar: Ao clicar no botão editar, será apresentado uma nova página edição dos dados específicos da publicação selecionada.
5. Excluir: Exclui a publicação selecionada.

5.2.1 Módulo API

O módulo API faz parte do módulo administrador e possui a finalidade de recuperar os dados das publicações realizadas pelo administrador no Facebook, estruturá-los de forma que atendam a arquitetura REST e realizar o controle de alguns dados. Isso é crucial para que o módulo cliente e o aplicativo possam realizar requisições de solicitação ou envio de dados.

O controle de dados realizado pelo módulo API é indispensável para que conteúdos como comentários inadequados não sejam exibidos. Para esse controle é efetuado uma varredura de todos os usuários que curtiram cada comentários da publicação, caso seja encontrado a curtida de um dos administradores cadastrados no banco de dados o comentário é aprovado para ser exibido.

Além disso, as requisições para esse módulo devem seguir uma sintaxe de acordo com os dados desejados. Como mostra a Listagem 5.1, onde é requisitado as informações das publicações presentes no banco.

```
1 http://{IP DO SERVIDOR}:{PORTA}/bd
```

Listagem 5.1 Requisitando dados para divulgação

A resposta para a solicitação da Listagem 5.1 é um arquivo do tipo JSON contendo o conteúdo apresentado no Listagem 5.2. O uso de cada um dos itens será explicado na Seção 5.3.

```
1 {
2     0: [
3         "bd": [
4             "linkqr": {LINK DA DIVULGACAO}
5             "legenda": {LEGENDA QUE SERA EXIBIDA}
6         ],
7         "comentarios": null,
8         "imagem": []
9     1: [
10        "bd": [
11            "linkqr": {LINK DA DIVULGACAO}
12            "legenda": {LEGENDA QUE SERA EXIBIDA}
13        ],
14        "comentarios": [
15            0: [
16                "created_time": {DATA DA CIRACAO},
17                "from": [
18                    "nome": {NOME DO USUARIO},
19                    "id": {ID DO USUARIO}
20                ],
21                "message": {MENSAGEM APRESENTADA},
22                "id": {ID DA PUBLICACAO},
23                "urlFoto": {URL DA FOTO DE PERFIL DO USUARIO}
24            ],
25        ],
26        "imagem": {IMAGEM DE EXIBICAO, CODIFICADA}
27    ]
28 }
```

Listagem 5.2 Retorno da requisição 5.1

A propriedade “bd”, contém uma *array* de duas posições, contendo 2 *strings*, sendo elas legenda e link. Esses dados foram armazenados pelo servidor no banco de dados, na criação da publicação. A propriedade “comentarios” é um array de *strings* que pode possuir zero ou mais

itens. Cada item representa um comentário da publicação, cada qual contendo data de criação, autor, foto de perfil e identificador do comentário. Já a “imagem” é uma *string* que representa uma imagem codificada em base64 a ser decodificada pelo módulo cliente.

O módulo API tem a função de, em conjunto com o banco de dados, realizar a simulação de um sistema de gestão acadêmico, sendo possível consultar alguns dados através de requisições. Os dados que podem ser recuperados, variam de acordo com a sintaxe utilizada.

É possível requisitar diversos dados distintos ou todos eles juntos. No Listagem 5.3 é feito a requisição de todos os dados armazenados, que retornará um JSON de diversas posições, onde cada posição representa um mensagem enviada por um professor.

```
1 http://{IP DO SERVIDOR}/mobile
```

Listagem 5.3 Requisitar todos os dados

```
1 0 : [
2     "id_mensagem": {NUMERO DA MENSAGEM NO SERVIDOR},
3     "matricula_professor": {MATRICULA DO PROFESSOR},
4     "nome_professor": {NOME DO PROFESSOR QUE ENVIOU},
5     "id_turma": {IDENTIFICACAO DA TURMA},
6     "mensagem": {MENSAGEM ENCAMINHADA PELO PROFESSOR}
7 ],
8 1 : [
9     "id_mensagem": {NUMERO DA MENSAGEM NO SERVIDOR},
10    "matricula_professor": {MATRICULA DO PROFESSOR},
11    "nome_professor": {NOME DO PROFESSOR QUE ENVIOU},
12    "id_turma": {IDENTIFICACAO DA TURMA},
13    "mensagem": {MENSAGEM ENCAMINHADA PELO PROFESSOR}
14 ]
```

Listagem 5.4 Retorno da requisição 5.3

A inclusão de um novo parâmetro na requisição da Listagem 5.3 torna a busca mais específica. Por exemplo, pode ser usado “/professores”, “/alunos/”, “/turmas/” ou “/mensagens/”, isso retornará um JSON contendo uma lista somente com os dados solicitados. Na Listagem 5.5, o retorno será uma lista com os dados de todos os professores cadastrados como mostra a Listagem 5.6.

```
1 http://{IP DO SERVIDOR}/mobile/professores/
```

Listagem 5.5 Requisitar lista de dados específica

```
1 0 : [
2     "matricula": {MATRICULA DO PROFESSOR},
3     "nome": {NOME DO PROFESSOR QUE ENVIOU},
4     "senha": {SENHA CRIPTOGRAFADA}
```

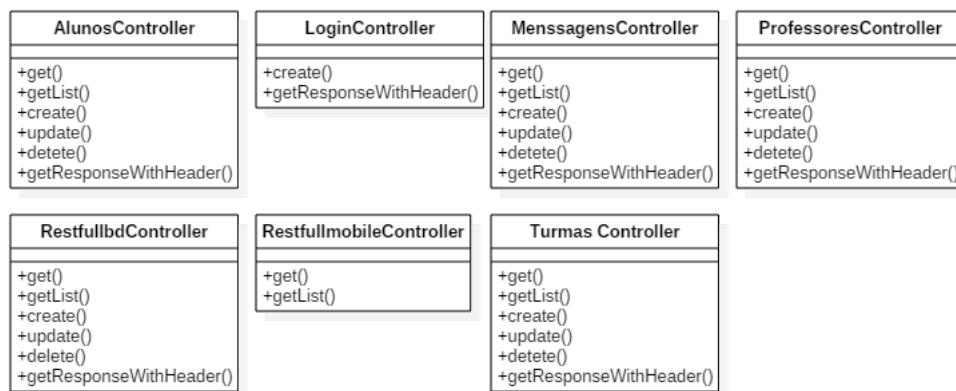
```

5   ],
6   1 : [
7     "matricula": {MATRICULA DO PROFESSOR},
8     "nome": {NOME DO PROFESSOR QUE ENVIOU},
9     "senha": {SENHA CRIPTOGRAFADA}
10 ]

```

Listagem 5.6 Retorno da requisição 5.1

O diagrama de classe que representa esse módulo é descrito na Figura 5.8. Cada uma das classes é usada para uma determinada requisição.

**Figura 5.8** Diagrama de classe do módulo API

5.3 Módulo Cliente

O modulo cliente tem a função de apresentar as divulgações de maneira atrativa, intuitiva e dinâmica para o usuário. O seu caso de uso é apresentado na Figura 5.9. A estrutura e organização do que será apresentado, está disposto na Figura 5.11 e o seu diagrama de classe é apresentado na Figura 5.10.

**Figura 5.9** Casos de uso da ações do módulo cliente



Figura 5.10 Diagrama de classe do módulo cliente

Os elementos que compõem a página que será apresentada são divididos em cinco itens, que estão descritos abaixo.

1. Imagem: O conteúdo poderá ser apresentado como uma imagem estática ou como GIF (imagem com animação), não oferecendo suporte a vídeos, até o momento, mas que poderá ser implementado em versões futuras.
2. Legenda: A legenda é apresentada em movimento linear da direita para a esquerda, possibilitando a leitura de forma dinâmica de toda a frase.
3. QR Code: Imagem, que por meio de um aplicativo celular, possibilita a leitura que contém o *link* de acesso a publicação publicada na rede social Facebook.
4. Horário: Relógio que apresentará a data e hora atual.
5. Comentários: Espaço que será apresentado comentários publicados e devidamente moderados.



Figura 5.11 Página do cliente

Consumindo uma API REST com requisições do tipo GET feitas a URL da Listagem 5.1, o módulo cliente receberá como resposta um JSON, contendo o número de posições equivalentes ao número de publicações válidas. Cada posição desse JSON terá três variáveis, são elas: "bd", "comentarios" e "imagem", como apresentado na Listagem 5.2.

O conteúdo recuperado da variável "bd", será apresentado nos itens 2 e 3, respectivamente. Já o conteúdo recuperado da variável "comentarios" será apresentado no item 5. Enquanto os da variável "imagem", será descriptografado e apresentado em forma de imagem no item 1.

O campo 5 irá apresentar 5 elementos recuperados na variável “comentarios”. Esses elementos serão: a imagem de perfil, o nome, o comentário, a data e a hora da publicação do comentário feito pelo usuário. Antes da apresentação das informações de comentários, é escolhido um número randômico que representará a posição do array do comentário que será apresentado.

5.4 Aplicativo móvel

Desenvolvido com auxílio dos *Frameworks* Cordova, Framework7 e Dom. O aplicativo móvel é responsável por apresentar de forma pública as divulgações, assim como o módulo cliente, com algumas funcionalidades a mais que são restritas.

As funcionalidades adicionais do aplicativo são as de disponibilizar uma melhor forma de comunicação entre professores e alunos, necessitando apenas de autenticação. Para autenticação é preciso que o aluno ou o professor tenha uma matrícula e senha cadastradas no banco.

Os casos de uso do aplicativo são demonstrados na Figura 5.12, enquanto o modelo entidade e relacionamento é apresentado na Figura 5.13.

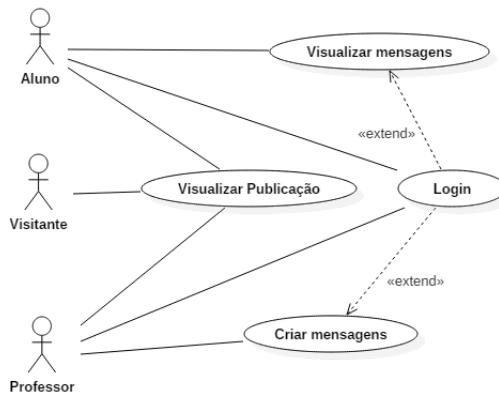


Figura 5.12 Casos de uso aplicativo móvel.

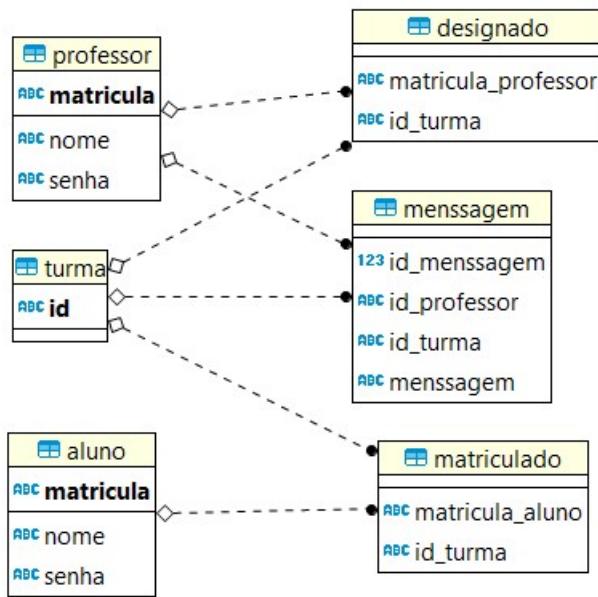


Figura 5.13 Modelo entidade e relacionamento aplicativo móvel.

O aplicativo contará com três telas distintas que serão a tela inicial, a tela do aluno e a tela do professor. Em cada tela o sistema faz requisição de um serviço diferente para o módulo API. Na tela inicial é feito a requisição GET a Listagem 5.1, na tela do aluno é feito a requisição GET a Listagem 5.3 e na tela do professor é feito um POST a Listagem 5.7,

1 `http://{IP DO SERVIDOR}/mensagens/`

Listagem 5.7 Incluir novas mensagens

A tela inicial será apresentada como na Figura 5.14, exibindo o nome do SID e o botão de login na parte superior e as informações das publicações válidas, que foram requisitadas do módulo API, no restante da tela.

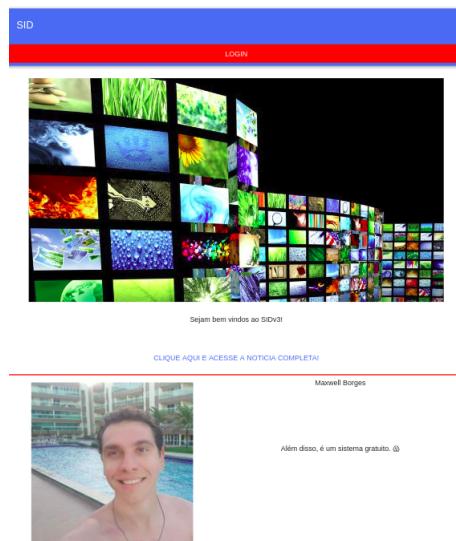


Figura 5.14 Página de inserção no módulo administrador.

A funcionalidade de login é indispensável para que alunos e professores efetuem a autenticação e possam acessar as páginas restritas destinadas a cada um.

A tela do aluno será apresentada como na Figura 5.15, após o login, será apresentado ao aluno todas as mensagens que o aluno possui em ordem cronológica da mais nova para a mais antiga. Os campos apresentados são os de professor, turma e mensagem, separando cada item por blocos.

Mensagens	
Professor: professor2	Mensagem
Turma: turma2014	Esta mensagem é do professor 2
Professor: professor2	Mensagem
Turma: turma2014	Outra nova mensagem para a turma
Professor: professor2	Mensagem
Turma: turma2014	mensagem do professor
Professor: professor2	Mensagem
Turma: turma2014	nova mensagem
Professor: professor2	Mensagem
Turma: turma2014	Essa é uma mensagem

Figura 5.15 Página do Aluno.

Já a tela do professor será apresentada conforme a Figura 5.16, após o login, será apresentado ao professor a matrícula dele, a lista de turmas que ele leciona e uma caixa de texto que o professor usará para digitar a mensagem.

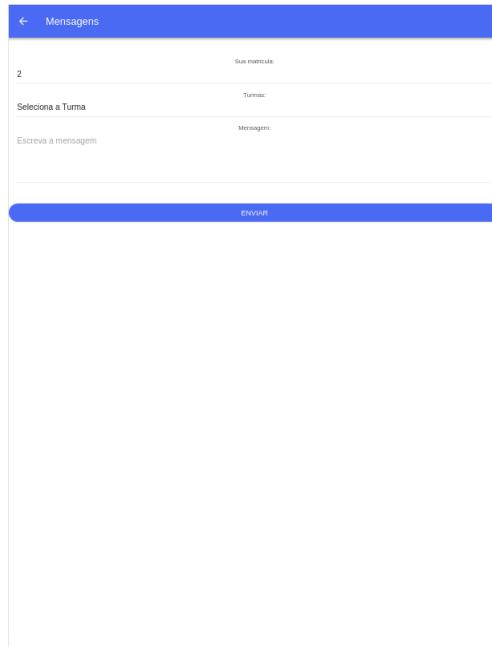


Figura 5.16 Página do professor.

Todos os campos são de preenchimento obrigatório e caso algum erro ocorra no processo de envio, o sistema apresentará um notificação para o professor com o erro ocorrido.

5.5 Integração

A integração entre o sistema e o Facebook se dá com o uso da Graph API. Ferramenta disponibilizada pela rede social para que feita a integração de sistemas externos a rede social Facebook.

Com o uso dela, pode-se recuperar os mais diversos dados de acordo com as permissões que são concedidas.

5.5.1 *login, token* de acesso e permissões

Para autenticação e efetivação de todas as requisições feitas pelo módulo administrador para o Facebook é necessário que um *login* seja feito. Para isso, é utilizado a ferramenta de “*login* do Facebook” que é disponibilizada pela rede social.

A ferramenta de “*login* do Facebook” que é disponibilizada pelo mesmo, oferece um botão que pode ser colocado na página que possui a finalidade de iniciar o processo de login. Após ser clicado é efetuado o processo de login conforme segue padrão estabelecido, solicitando as credenciais cadastradas na rede social, com isso é feito a requisição com as permissões necessárias conforme Listagem 4.10.

No processo de *login* são solicitados as permissões necessárias para que o sistema funcione, essas permissões são a “email”, “manage_pages” e “publish_pages”, sendo obrigatório o aceite para acesso ao sistema. Usadas respectivamente para que seja possível a identificação

do usuário do SID, para recuperar as permissões de acesso a página e a última para permitir que aplicativos publiquem na página FACEBOOK (2018c).

Após a efetivação do login o sistema realiza a requisição da Listagem 5.8 para verificar as permissões que o usuário concedeu, onde o retorno será um JSON contendo todas as permissões concedidas.

```
1 $response = $fb->get(            
2     '/me/permissions',             
3     '{access-token}'             
4 );                             
5                                 
6 $graphNode = $response->getGraphNode();
```

Listagem 5.8 Permissões concedidas

Então o sistema compara as permissões aceitas com as permissões necessárias e caso alguma não esteja presente, o acesso é negado, retornando para a página de login e apresentando a mensagem de erro para o usuário.

A existência de um usuário vinculado a rede social é irrefutável, pois existe a necessidade da página apresentar qual o perfil está realizando a ação, além da necessidade de moderação dos comentários que serão exibidos. Além disso, para efetivação de todas as requisições feitas pelo módulo administrador para o Facebook, seja ela para requisitar informações das publicações ou realizar uma nova publicação, é usando o *token* de acesso de usuário, que é obtido no processo de login, conforme Listagem 4.1.

Entretanto, para o acesso é indispensável também que o um usuário esteja cadastrado no banco de dados do SID. Essa ação é necessária para que somente usuários autorizados possam realizar alteração na página usando o sistema.

Após o login, algumas informações usadas são guardadas na sessão do usuário, para usos posteriores. Porém, todo o processo de armazenamento de informações, envios ao Facebook e retornos de solicitações não devem ser visíveis ao usuário.

5.5.2 Publicar

Todas as publicações que são realizadas no Facebook usando o SID são feitas ao nó “/photo”, onde a cada nova publicação será gerado um ID para ela e terá a mesma estrutura de uma imagem postada por um usuário convencional desta rede social.

Na criação de uma nova publicação, usando o SID, são passados para a Graph API dois parâmetros para inserção através do método POST, conforme Listagem 4.14. O primeiro deles é “message”, onde será passado o texto que será exibido na publicação e o outro é “source”, onde será passado a imagem para ser exibida juntamente com o texto. Para envio de imagem para a rede social, é necessário passar na imagem como parâmetro o método “fileToUpload” conforme Listagem 4.9.

Alguns dos elementos que são solicitados pela aplicação na criação de uma nova publicação são omitidos no envio para o Facebook, pois esses dados serão usados apenas para serem armazenados no banco para uso do módulo cliente. Os elementos omitidos são os campos data de início, a data de término e a legenda.

A Figura 5.17 apresenta como fica uma publicação no Facebook após o uso do SID para criação da mesma, enquanto a Figura 5.11 apresenta como será exibido no módulo cliente. Nela são apresentadas informações como quem realizou a publicação, o texto e a imagem que foram informados durante a criação da divulgação pelo SID.



Figura 5.17 Divulgação enviada ao Facebook com auxílio do SID

5.5.3 Requisição

Usando o ID da foto é possível recuperar informações das arestas, que podem ser a URL, os comentários e as curtidas conforme Listagem 4.40.

Esse recurso é utilizado para recuperar os mais diversos elementos que as permissões concedidas autorizam, entre os elementos recuperados estão o endereço da publicação, os comentários, curtidas e a imagem de perfil que serão apresentados respectivamente nos campos destinados ao QR Code e na coluna de comentários do módulo cliente.

A URL da publicação que é criada, é armazenada no banco de dados no momento da criação da mesma pelo modulo administrador. A requisição segue a Listagem 5.9.

```
1 $response = $fb->get(  
2     '/{ID PUBLICACAO}?fields=link',  
3     '{access-token}'
```

```

4   );
5
6   $graphNode = $response->getGraphNode();

```

Listagem 5.9 Foto de usuário

A recuperação dos outros dados necessários é feito pelo módulo API, onde ele realiza requisições a Graph. As requisições feitas pelo módulo são para recuperação de comentários da Listagem 4.33, de curtidas desses comentários, conforme Listagem 4.34 e foto de perfil do usuário que realizou o comentário conforme Listagem 5.10.

```

1   $response = $fb->get(
2       '/{ID_USUARIO}?fields=picture',
3       '{access-token}'
4   );
5
6   $graphNode = $response->getGraphNode();

```

Listagem 5.10 Foto de usuário

A recuperação dos comentários e da foto do perfil do usuário que publicou cada comentário é crucial para que essa informação seja exibida no módulo cliente, no elemento 5, conforme a Figura 5.11.

A recuperação de curtidas é necessária para que possa ser feito a moderação dos comentários a serem exibidos. A moderação é feita de forma a verificar o ID de todos os usuários que curtiram cada comentário a fim de verificar se algum dos IDs encontrados é o ID do administrador que está cadastrado no banco.

Os comentários que possuem a curtida do administrador é registrado para exibição no elemento 5 da Figura 5.11.

5.6 Possível solução para Implantação - Raspberry

Para viabilizar uma implementação barata ao *Campus Taguatinga*, computadores Raspberry Pi podem ser utilizados.

Teste do ARISTOTELOUS (2016) apresenta a possibilidade de se ter um servidor completamente funcional com sistema operacional Linux por um equipamento de 35\$, possibilitando a criação de um servidor, por exemplo de um repositório na nuvem com um baixo custo, flexibilidade e eficiência energética.

Para CUSICK et al. (2014), placas de circuito oferecem vantagens como: Uso de pouco espaço; Desempenho significante com baixo custo e consumo; Suporte a diversas soluções de software, oferecendo múltiplas opções de interface com uma variante do Linux.

Portanto, este tipo de dispositivo provê diversos benefícios que podem ser de grande importância na escolha de um sistema para implementação do módulo cliente do SID.

6

Resultados

A ideia anterior do SIDv2, desenvolvido por SOBRINHO (2017), era a de um sistema inteligente de divulgação, entretanto, no presente trabalho a ideia foi alterada para um sistema integrado de divulgação. Isso se deu pelo fato de o sistema não possuir nenhum tipo de inteligência própria e sim realizar a integração entre redes sociais e usuário.

O SID passou por diversas reformulações de suas concepções iniciais. Em sua versão anterior, ele possuía integração bastante limitada, nenhum tipo de aplicativo mobile para melhor interação acadêmica e pouca interatividade do sistema com o usuário e vice-versa. Então, no presente trabalho foi implementado toda a integração com o Facebook, melhorado a interação com o usuário, além da criação de um protótipo *mobile*.

O presente trabalho apresenta a terceira versão do SID, com melhorias significantes como as apresentadas no Capítulo 5, estando pronto para implantação de modo a servir de aparato ao campus no processo de veiculação de informações.

Por falta de uma documentação anterior, foi necessário a criação de uma extensa documentação do software de modo a possibilitar futuras modificações e melhorias de maneira fácil.

No módulo administrador, foi possível alcançar todos os objetivos desejados, ficando responsável por todo o processamento e gerenciamento das informações no banco de dados. A partir dele é possível inserir, listar, deletar e editar as informações que servirão para que módulo API forneça os dados conforme a estrutura de uma arquitetura REST, permitindo que outras aplicações futuramente venham a consumir esta API.

Além disso, foi implementado uma verificação de segurança no processo de login que não existia no SIDv2, onde qualquer usuário cadastrado poderia acessar a aplicação. Entretanto, algumas funcionalidades oferecidas pelo módulo só estão disponíveis se forem concedidas algumas permissões específicas. Portanto, o login no módulo só é realizado caso todas as permissões solicitadas sejam aceitas.

O submódulo API, parte do módulo administrador, é estruturado de acordo com a arquitetura REST, com a capacidade de realizar um CRUD nas publicações e nas mensagens entre professor aluno.

Este submódulo foi implementado com duas funcionalidades distintas. A primeira é

obter os dados do armazenados no banco e realizar as chamadas a Graph para obter os dados dos comentários das publicações que serão exibidas e organizar esses dados para que sejam enviados. A segunda, é servir de aparato ao aplicativo, para que ele possa simular o consumo de uma API externa.

O módulo cliente do SID ficou estruturado e pronto para ser implantado e executado em diversos dispositivos, bastando o mesmo possuir um navegador de páginas WEB e conexão constante à internet para que sejam realizadas as consultas HTTP. Essa facilidade é possível graças à implementação em arquitetura cliente-servidor, esse arquitetura foi proposta para que o módulo consumisse poucos recursos, permitindo o uso de dispositivos que dispõem de uma grande eficiência energética, como: *Raspberry PI*, *BoxTVs* ou até mesmo em sistemas integrados às próprias *smartTvs*.

No protótipo do aplicativo mobile, foi implementado a funcionalidade de consumo de uma REST API fictícia, simulando o Sistema de Gestão Acadêmica do IFB, para exibição das divulgações e um sistema de comunicação interna entre professores e alunos, onde é necessário um login criado de maneira fictícia para simulação de um sistema acadêmico. necessitando apenas da liberação dos dados do SGA para autenticação do sistema e a adaptação das consultas feitas a esta API.

Na busca de soluções que seguem a ideia de criação de publicação para apresentação em dispositivos externos, foram encontradas diversas ferramentas com essa finalidade, todas elas usando o conceito de arquitetura cliente-servidor, as que apresentaram melhores funcionalidades foram a OOZO, a MangoSigns, o SID versão 2, a Screenly e a XIBO.

O OOZO, apresentado na sessão 2.1, é um sistema pago, porém possui uma versão gratuita com acesso limitado aos recursos oferecidos. Ele oferece suporte a multiplataformas, controle WEB, integração a redes sociais e acesso a publicações via QRCode. Apresentando duas limitações, uma é no que tange a integração com as redes sociais, onde é possível apenas recuperar e deletar publicações, a outra limitação está no acesso às divulgações por dispositivo móvel, onde é necessário o uso de navegadores WEB.

A MangoSigns, apresentado na sessão 2.2, é um sistema pago que possui uma versão gratuita com acesso limitado. A ferramenta possui acesso multiplataformas, inclusive móvel, diversos temas para apresentação e integração com as redes sociais. Possuindo a limitação da integração está disponível apenas na versão paga, além de ser possível apenas recuperar as publicações.

O SID versão 2, apresentado na sessão 2.3, é um sistema gratuito, que possui acesso somente WEB, acesso a publicação via QRCode e integração às redes sociais. Nele é possível apenas realizar publicação da página de perfil do usuário, não sendo possível recuperar nenhum dado.

O Screenly, além de usar um aplicativo proprietário, necessitando de um *raspberry*, ele não possui aplicativo mobile e nenhuma forma de integração com as redes sociais.

Já o Xibo, apesar de ser uma ferramenta gratuita, não possui aplicativo mobile e nenhuma

integração com as redes sociais.

Comparando cada uma das ferramentas encontradas com o SIDv3 é possível notar diversas funcionalidades que não estão presentes, são limitadas ou não são gratuitas, portanto diversos pontos que são considerados como essenciais e foram implementados no SIDv3, as ferramentas apresentadas não suportam.

Entre esses pontos, está a visualização das publicações por meio de um aplicativo móvel, onde com exceção do MangoSings, nenhum dos sistemas ofereciam um aplicativo para *smartphones*, necessitando de um dispositivo que possuísse um navegador WEB para acesso.

Outro ponto é o de integração com as redes sociais, nenhum das solução analisadas realizavam uma integração tão completa quanto ao do SIDv3, como a possibilidade de criação de novas publicações, recuperação de comentários, curtidas e fotos de perfil, além do filtro de comentários a serem exibidos.

Além disso, nenhum deles possuía um aplicativo que realiza a integração com um sistema acadêmico, para integração entre professores e alunos.

A tabela 6 tem o intuito de fazer um comparativo entre os sistemas apresentados, agora com inclusão do SID em sua terceira versão, comparando algumas das funcionalidades consideradas importantes para sistemas que trabalham com a implantação de sinalização digital e *marketing* digital, na qual seus elementos comparativos são descritos a seguir:

- I) Comprometimento com o propósito: o sistema em questão possibilita a veiculação de informações através de mecanismos de sinalização digital?
- II) Criação simples de divulgações: o operador possui facilidade de incluir novas divulgações com aspecto atrativo?
- III) Portabilidade *mobile*: o módulo cliente está disponível para a plataforma *mobile* em forma de aplicativo?
- IV) Integração com redes sociais: o sistema realiza integração completa com as redes sociais, oferecendo recuperação e envio de dados.
- V) A versão gratuita explora toda a capacidade do programa?
- VI) É possível criar uma nova publicação e envia-la para as redes sociais?

Tabela 6.1 Comparativo

Quesito/Sistema	OOZO	MangoSigns	SID v2	Screenly	XIBO	SIDv3
I	SIM	SIM	SIM	SIM	SIM	SIM
II	SIM	SIM	SIM	SIM	SIM	SIM
III	NÃO	SIM	NÃO	NÃO	NÃO	SIM
IV	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
V	NÃO	NÃO	SIM	NÃO	SIM	SIM
VI	NÃO	NÃO	SIM	NÃO	NÃO	SIM

Portanto, a tabela mostra que o SIDv3 atende a todos os requisitos desejados para o sistema.

6.1 Melhorias em relação ao SIDv2

6.1.1 Módulo Administrador

- Múltiplos administradores: No SIDv2, as informações dos administradores eram armazenadas em um arquivo texto no código fonte do sistema, na v3 os dados são armazenados em um banco de dados, onde é possível registrar inúmeras contas, diferente do arquivo texto, onde era só se recuperava um único usuário.
- Mensagens de erro ao logar: Na segunda versão, quando se entrava com um usuário ou senha que não estava registrado no arquivo texto, o sistema não retornava nenhuma informação de erro, apenas retornava para a tela de *login*. Na nova versão, o sistema apresenta na tela de *login* qual o problema foi encontrado pelo sistema para a não autenticação.
- Restrição de acesso: Na v2, bastava o administrador possuir os seus dados cadastrados, que a autenticação no sistema era efetivada, na nova versão o sistema realiza a verificação das permissões que são necessárias para o funcionamento do sistema e se essas permissões foram concedidas pelo usuário.
- Recuperação automática de endereços: No SIDv2, na página de inserção, o SID usava o campo de texto "Link para mais informações" e o *combobox* "Prioridade", entretanto no SIDv3, os dois campos foram retirados, o campo "link para mais informações" foi retirado para que o endereço fosse obtido de forma automática pelo sistema, recuperando o endereço da publicação no Facebook e o campo "Prioridade" foi retirado pois não estava implementado em nenhuma versão.
- Data de início: Na v2, na página de inserção, o administrador somente tinha a funcionalidade de definir a data em que a publicação deixaria de ser exibida no

módulo cliente. Na v3, foi inserido a nova funcionalidade de se definir também a data em que a publicação começará a ser exibida.

- Acesso direto a publicação: Na tela de detalhes, na v2 do SID, era mostrado os campos de *link* e prioridade. Esses campos foram substituídos para que fosse colocado o endereço para acesso direto a postagem no Facebook.
- Exclusão: Na segunda versão, o SID somente excluía os dados localmente do banco de dados e não era apresentado nenhuma mensagem para o usuário. Em sua terceira versão, foi implementado a funcionalidade de excluir não somente localmente, mas também a publicação do Facebook e para que a ação seja efetivada, é necessário o administrador confirmar em uma janela *popup* que é apresentada.

6.1.2 Módulo Cliente

- Atualização automática: Em sua segunda versão, o módulo cliente realizava a requisição ao servidor, que recuperava as informações do banco de dados e as enviava para serem exibidas no cliente, entretanto, essa requisição era realizada uma única vez, necessitando de uma atualização da página todas as vezes que um conteúdo fosse alterado. Na terceira versão, é realizado uma nova requisição de tempos em tempos de forma automática, de forma que quando uma nova publicação ou comentário é feito, o sistema automaticamente irá exibi-lo.
- Exibição de comentários: A v2 do SID, não apresentava nenhuma informação de comentário. Foi implementado todo o sistema de requisição, atualização e apresentação dessas informações.

6.1.3 Submódulo API e Aplicativo móvel

Em sua v2, o SID não possuía um submódulo API e nem um aplicativo móvel. Para a v3, foi realizado toda a criação da documentação e implementação do módulo e do sistema.

6.1.4 Banco de dados

- Armazenamento das imagens publicadas: Na segunda versão, ao se criar uma nova publicação, eram armazenados todas as informações no banco de dados em formato texto. Na nova versão, as imagens que eram armazenadas em formato bytea, agora são armazenadas localmente no servidor. A necessidade dessa alteração se deu para que se possa diminuir a latência na busca de informações no banco de dados.
- Tabela divulgacao: Os atributos "prioridade" e "imagem" foram retiradas pois não seriam mais usadas, no lugar, foram criadas os atributos "datainicio" e "object_id".

- Tabela config: A tabela foi criada para armazenar as informações necessárias para realizar requisições ao Facebook. Na versão 2, esses dados eram recuperados de um arquivo texto, não possibilitando a inclusão de novos aplicativos e vincular os a sistema.
- Tabelas para módulo mobile: Foram criadas as tabelas: professor, designado, turma, mensagem, aluno e matriculado. Esta ação foi necessária para que fosse possível realizar a criação de um sistema fictício que simula parte das funcionalidades do SGA.

6.2 Dificuldades encontradas

Durante o desenvolvimento do sistema, foram enfrentados diversos problemas, entre eles estão a falta da documentação detalhada da versão anterior, problemas com a aprovação de permissões solicitadas para o Facebook, a limitação do *token* de aplicativo e o não disponibilização da API do SGA.

A falta de uma documentação detalhada do SIDv2 acabou influenciando no entendimento de como as classes, métodos e o banco de dados se interagiam um com os outros. Sendo necessário um maior tempo de pesquisa e entendimento do sistema, além de ter sido necessário a criação de uma documentação totalmente nova, não sendo possível o aproveitamento de nenhuma anterior.

Com exceção da permissão “email” e “*public profile*”, todas as outras permissões necessitam passar pelo processo de análise do Facebook, sendo necessário o envio de um vídeo demonstrando o uso da permissão. O processo de análise não foi concluído até a finalização deste documento, portanto, o uso das permissões estão concedidas por tempo limitado.

Os *tokens* de aplicativo apresentam a limitação de não autorizar que sejam recuperados dados considerados privados pela rede social, como as curtidas de um comentário, que são usadas para moderação dos comentários que serão exibidos. Para contornar o problema, foi necessário o uso de um *token* de usuário para realizar as requisições de solicitação de dados.

Uma dos objetivos do aplicativo *mobile*, é o uso de informações de alunos e professores presentes no SGA para efetivação da autenticação do usuário. Entretanto, o IFB não disponibilizou em tempo hábil a base de dados para que fosse possível a sua implementação. O problema foi contornado com o uso de uma base de dados fictícia, para posterior migração.

7

Considerações Finais

Com objetivo de aprimorar os meios de comunicação do IFB - Campus Taguatinga, foi usado os conceitos de marketing digital e sinalização digital para implementação de um sistema fácil e claro. Inserindo interação com o usuário que visualiza as divulgações, além de um sistema que simula o consumo da API do SGA, para possibilitar uma posterior inclusão quando esta interface do SGA for disponibilizada.

O SID está dividido em módulo administrador, módulo cliente e aplicativo. O administrador é responsável por desempenhar todo o processamento de recuperação, armazenamento e estruturação dos dados, enquanto o módulo cliente e o aplicativo realiza a função de exibição desses dados de forma acessível para o usuário.

Em suma, o SID apresenta benefícios consideravelmente melhores em relação às outras soluções que são utilizadas atualmente, além da divisão entre módulos possibilitar a exibição das divulgações em diversos dispositivos distintos, incluindo o aplicativo *mobile* que foi desenvolvido.

Entre os benefícios apresentados estão: integração completa com as redes sociais, possibilitando recuperação de comentários, curtidas e postagens em tempo real; divulgação de eventos e notícias através de telas espalhadas pelo campus; unificação de dois meios de comunicação distintos, facilitando a vida do operador do sistema; consumo de uma API Fictícia visando a simulação com o SGA para posterior integração com o mesmo, quando esta API do SGA estiver disponível.

7.1 Trabalhos Futuros

Apesar de atingido todos os objetivos, a implementação de uma REST API abre diversas possibilidades de ampliação do sistema. Sendo possível a inserção de novos módulos e funcionalidades.

7.1.1 Aplicativo *mobile*

A melhoria e finalização do aplicativo *mobile* é uma das propostas para melhoria, onde pode ser feita uma melhor integração com o SGA após a liberação da interface para acesso ao

SGA.

7.1.2 Moderação dos comentários

Além disso, é necessário uma nova forma de moderação dos comentários, pois existe as limitações descritas para os *tokens* de aplicativo.

7.1.3 Atraso da recuperação de dados

Alguns dados que são recuperados pelo módulo API, são repassados para o cliente em forma de URL, sendo fundamental a busca *online* desse dado pelo cliente, o que pode acabar gerando um atraso da entrega da informação. Implementar uma nova forma de recuperação desses dados como a utilização de uma cachê, poderia diminuir o atraso.

Referências

- ARISTOTELOUS, A. **Single Board Computer for Standardized Onboard Vehicle Network.** [S.l.: s.n.], 2016. 63p. Trabalho de Conclusão de Curso.
- BALDUÍNO, P. **Dominando JavaScript com jQuery.** [S.l.: s.n.], 2014. Livro.
- BIANCHI, P.; HATJE, M. **Mídia e esporte:** os valores-notícia e suas repercussões na sociedade contemporânea. [S.l.: s.n.], 2006. 13p. Revista Motrivivência.
- CANTO, L. C. D.; CORSO, K. B. **Marketing na Era Digital.** [S.l.: s.n.], 2017. 14p. Revista International Journal of Business Marketing.
- CECIN, F. R. **FreeMMG:** uma arquitetura cliente-servidor e par-a-par de suporte a jogos maciçamente distribuídos. [S.l.: s.n.], 2005. 101p. Tese de Mestrado.
- CINTRA, F. C. **Marketing Digital:** a era da tecnologia on-line. [S.l.: s.n.], 2010. 7p. Artigo UniFran.
- CUSICK, J. et al. **Design, Construction, and Use of a Single Board Computer Beowulf Cluster:** application of the small-footprint, low-cost, insignal 5420 octa board. [S.l.: s.n.], 2014. Artigo.
- DEUZE, M. **The Internet and Its Journalisms—Part I:** a typology of online journalism. 2002.
- ELMASRI, R.; NAVATHE, S. B. **Sistema de Banco de Dados. Revisor técnico Luiz Ricardo de Figueiredo.** [S.l.: s.n.], 2005. Livro.
- eMarketer. URL: <https://www.emarketer.com/Report/Worldwide-Internet-Mobile-Users-eMarketers-Updated-Estimates-Forecast-20172021/2002147>, Acessado em: 28-06-2018.
- ESCOBAR, J. L. **Blogs e interação mútua:** uma visão contextualizada. [S.l.: s.n.], 2007. 17p. XXX Congresso Brasileiro de Ciências da Comunicação.
- FACEBOOK. **Visão geral.** [S.l.: s.n.], 2018.
<https://developers.facebook.com/docs/graph-api/overview>, acessado em: 2018-04-10.
- FACEBOOK. **Login no Facebook para Aplicativos—Visão geral.** [S.l.: s.n.], 2018.
<https://developers.facebook.com/docs/facebook-login/overview/>, acessado em: 2018-05-23.
- FACEBOOK. **Referência a permissões - Login do Facebook.** [S.l.: s.n.], 2018.
<https://developers.facebook.com/docs/facebook-login/permissions>, acessado em: 2018-04-10.
- FRAMEWORK7. **Framework7 - Full featured HTML framework for building iOS & Android apps.** [S.l.: s.n.], 2018. <https://framework7.io/>, acessado em: 2018-05-08.
- GUAZINA, L. **O conceito de mídia na comunicação e na ciência política:** desafios interdisciplinares. [S.l.: s.n.], 2007. 49p. Revista Debates.
- GUEDES, G. T. **UML 2-Uma Abordagem Prática.** [S.l.: s.n.], 2009. Livro.
- HJARVARD, S. **Midiatização:** teorizando a mídia como agente de mudança social e cultural. [S.l.: s.n.], 2012. 38p. Revista Matrizes.

- MACHADO, J. D. A. **A sinalização digital como ferramenta de relações públicas: o caso light sa.** [S.l.: s.n.], 2010. 67p. Trabalho de Conclusão de Curso.
- MAKICE, K. **Twitter API: up and running: learn how to build applications with the twitter api.** [S.l.: s.n.], 2009. Livro.
- MangoSigns. **Free Digital Signage Software.** [S.l.: s.n.], 2017. <https://mangosigns.com/>, acessado em: 2017-09-11.
- MEDITSCH, E. **O ensino do radiojornalismo em tempos de internet.** [S.l.: s.n.], 2001. 10p. XXIV Congresso Brasileiro da Comunicação.
- MELO, G. S. de. **Introdução a Teoria dos Grafos.** [S.l.: s.n.], 2014. 35p. Dissertação de Mestrado.
- MINETTO, E. L. **Frameworks para Desenvolvimento em PHP.** [S.l.: s.n.], 2007. Livro.
- MISHIMA, K.; SAKURADA, T.; HAGIWARA, Y. **Cost effective digital signage system using low cost information device.** [S.l.: s.n.], 2016. 2p. Conferencia IEEE.
- MUCHARDIE, B. G.; YUDIANA, N. H.; GUNAWAN, A. **Effect of Social Media Marketing on Customer Engagement and its Impact on Brand Loyalty in Caring Colours Cosmetics, Martha Tilaar.** [S.l.: s.n.], 2016. 6p. Revista Binus.
- MULLER, J. et al. **Display blindness:** the effect of expectations on attention towards digital signage. [S.l.: s.n.], 2009. 2p. International Conference on Pervasive Computing.
- MUNARI, B. **Design e comunicação visual:** contribuição para uma metodologia didática. [S.l.: s.n.], 2006. Livro.
- OAUTH. **OAuth 2.0.** [S.l.: s.n.], 2018. <https://oauth.net/2/>, acessado em: 2018-05-23.
- OOZO. **Digital Signage platform for Raspberry Pi.** [S.l.: s.n.], 2017. <https://pt.oozo.tv/>, acessado em: 2017-09-09.
- PINHEIRO, R. C. E. **A importância da comunicação interna para o sucesso organizacional.** [S.l.: s.n.], 2010. 61p. Monografia para especialização.
- PREZOTTO, E. D.; BONIATI, B. B. **Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas.** [S.l.: s.n.], 2017. 8p. Anais do EATI.
- RICARTE, I. L. M. **Programação Orientada a Objetos:** uma abordagem com java. [S.l.: s.n.], 2001. 118p. Trabalho de Conclusão de Curso.
- ROSA, N. G. H. D. **O impacto das redes sociais no marketing:** perspectiva portuguesa. [S.l.: s.n.], 2010. 87p. Tese de Mestrado.
- RYAN, D. **Understanding digital marketing:** marketing strategies for engaging the digital generation. [S.l.: s.n.], 2016. Livro.
- SANTOS, F. A. D. **Marketing na Era Digital:** análise da marca chico rei. [S.l.: s.n.], 2014. 59p. Trabalho de Conclusão de Curso.
- Screenly. **Digital Signage for Raspberry Pi.** [S.l.: s.n.], 2017. <https://www.screenly.io/>, acessado em: 2017-09-10.

- SEBESTA, R. W. **Conceitos de linguagens de programação.** [S.l.: s.n.], 2011. Livro.
- SEO, E.-J.; PARK, J.-W. **A study on the effects of social media marketing activities on brand equity and customer response in the airline industry.** [S.l.: s.n.], 2017. 6p. Revista Elsevier.
- SILVA, D. E. D. S.; SOUZA, I. T. de; CAMARGO, T. **Metodologias Ágeis Para O Desenvolvimento De Software:** aplicação e o uso da metodologia scrum em contraste ao modelo tradicional de gerenciamento de projetos. [S.l.: s.n.], 2013. 7p. Revista Computação Aplicada-UNG.
- SILVA, R. M. D. **Educação, TV e Hegemonia.** [S.l.: s.n.], 2007. 56p. Espéculo: Revista De Estudios Literarios.
- SOBRINHO, M. d. C. **Sistema Inteligente de Divulgação de Informações do IFG-Formosa.** [S.l.: s.n.], 2017. 64p. Trabalho de Conclusão de Curso.
- SOUZA, A. S. B. T. d. **Uso do QR no marketing digital:** a perspetiva do utilizador português. [S.l.: s.n.], 2014. 70p. Tese de Mestrado.
- TORRERS, G.; COZER, A. **Alavancando negócios na internet.** [S.l.: s.n.], 2000. Livro.
- VASWANI, V. **Zend Framework, A Beginner's Guide.** [S.l.: s.n.], 2010. Livro.
- W3TECHS. **Usage statistics and market share of PHP for websites.** [S.l.: s.n.], 2018. <https://w3techs.com/technologies/details/pl-php/all/all>, acessado em: 2018-06-11.
- Xibo. **Digital Signage for Everyone!** [S.l.: s.n.], 2017. <http://xibo.org.uk/>, acessado em: 2017-09-10.
- ZEND. **About - Zend Framework.** [S.l.: s.n.], 2018. <https://framework.zend.com/about>, acessado em: 2017-10-10.
- ZHOU, W.; LI, L.; LUO, M. **REST API design patterns for SDN northbound API.** [S.l.: s.n.], 2014. 7p. Conferência IEEE.

Apêndice

A

Documentação

A.1 Prototipação

A.1.1 Visão Geral

O Sistema possuirá duas interfaces, sendo elas a de administrador e a do cliente. A interface do administrador é usada pelo administrador do sistema para criar, editar, listar ou visualizar as publicações. Enquanto a interface de cliente apresenta ao usuário uma tela com as informações criadas, comentários feitos nas redes sociais e um link em formato QR para que acessar a página *web* da publicação.

A.1.2 Interface de usuário

Tabela A.1 Interfaces de Usuário

Nome	Descrição
Interface de administrador (logar)	Tela inicial apresentada quando se quer acessar as funcionalidades de administrador.
Interface do administrador (Inserir)	Interface para criação das novas publicações que serão exibidas para o cliente.
Interface do administrador (Listar)	Interface que lista todas as publicações criadas até o momento na tela de criação.
Interface do administrador (Editar)	Interface que possibilita a edição de alguns dados que foram inseridos.
Interface do administrador (Detalhar)	Interface que apresenta os detalhes de cada publicação criada.
Interface do Cliente	Interface onde é possível o cliente visualizar e acessar a publicação criada.

A.1.2.1 Interface do administrador (Logar)

- Leiaute

200.130.152.84:6670



Figura A.1 Tela de login do Administrador

- Campos
Não se aplica.
- Comandos

Tabela A.2 Comandos da tela de logar

Nome	Descrição	Grupo	Requisitos de validade	Requisitos Diversos
Logar com Facebook	Usa a conta do Facebook do usuário para fazer login.	Todos	Autoriza se a conta estiver registrada no banco.	Deve possuir uma conta no Facebook.

A.1.2.2 Interface do Administrador (Inserir)

- Leiaute

The screenshot shows a web-based application interface for publishing. At the top, there is a dark header bar with the text "Home" and "Inserir". On the right side of the header is a green logo consisting of a stylized 'F' shape made of small squares, followed by the word "Sair". Below the header, the main content area has a light gray background. It contains several input fields and labels:

- A label "Nova Divulgação:" above a text area labeled "Legenda". The text area contains placeholder text: "Texto para exibida na publicação".
- A label "Data de inicio da divulgação" above three dropdown menus labeled "Dia", "Mês", and "Ano".
- A label "Data de término da divulgação" above three dropdown menus labeled "Dia", "Mês", and "Ano".
- A label "Imagen:" above a file input field. The field shows a "Browse..." button and the message "No file selected."
- At the bottom right of the form area are two buttons: "Cancelar" (Cancel) and "Inserir" (Insert).

©2018-2018 IFB - Campus Taguatinga Projeto SID

Figura A.2 Tela de inserção de novas publicações

■ Campos

Tabela A.3 Campos da tela de inserção

Nome	Descrição	Grupo	Requisitos de conteúdo	Requisitos de edição	Requisitos Diversos
Legenda	Legenda que irá ser apresentada no cliente.	Admin	Texto	–	–
Texto	Texto da publicação que será postada no Facebook.	Admin	Texto	–	–
Data de início	Data em que a publicação começará a ser exibida no cliente.	Admin	Inteiro selecionável entre as opções apresentadas	–	–
Data de Término	Data em que a publicação deixará de ser exibida no cliente.	Admin	Inteiro selecionável entre as opções apresentadas	–	–
Imagen	Imagen que irá ser apresentada na publicação do Facebook e no Cliente.	Admin	Arquivo do tipo imagem.	–	A imagem deve ter o formato suportado pelo Facebook

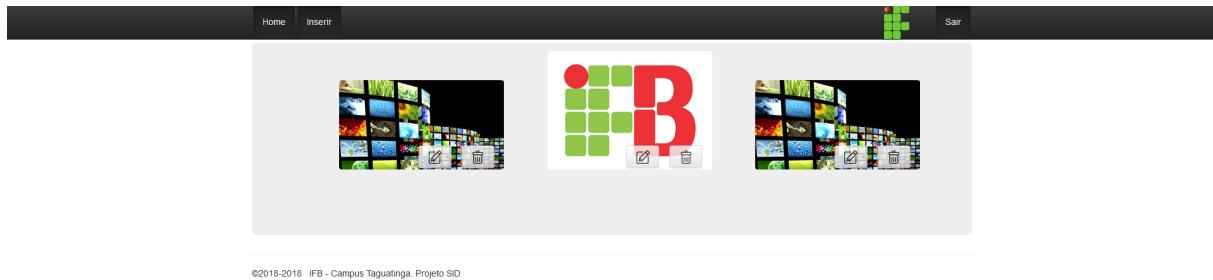
■ Comandos

Tabela A.4 Comandos da tela de inserção

Nome	Descrição	Grupo	Requisitos de validade	Requisitos Diversos
Inserir	Insere no banco e no Facebook as informações descritas nos campos.	admin	Todos os campos devem estar preenchidos.	
Cancelar	Cancela a inserção dos dados e retorna para a lista de publicações criadas.	admin		

A.1.2.3 Interface do Administrador (Listar)

■ Leiaute

**Figura A.3** Tela de listagem das publicações

- Campos
Não se aplica.
- Comandos

Tabela A.5 Comandos da tela de listagem

Nome	Descrição	Grupo	Requisitos de validade	Requisitos Diversos
Detalhar	Clicar sobre a imagem retorna a página com detalhes da publicação.	admin	—	—
Editar	O ícone de editar, retorna pagina de edição.	admin	—	—
Deletar	O ícone de excluir, exclui a publicação do banco e do Cliente.	admin	—	—

A.1.2.4 Interface do Administrador (Editar)

- Leiaute

The screenshot shows a modal dialog titled 'Editar Dados da Divulgação:' (Edit Disclosure Data). It contains a 'Legenda' section with a text input field containing 'conheça o SID'. Below it are two date pickers: 'Data de inicio da divulgação' (Start Date) set to '01/01/2018' and 'Data de término da divulgação' (End Date) set to '01/01/2030'. At the bottom are 'Cancelar' (Cancel) and 'Atualizar' (Update) buttons. At the very bottom left, a copyright notice reads '©2018-2019 IFB - Campus Taguatinga. Projeto SID.'

Figura A.4 Tela de edição das publicações

- Campos

Tabela A.6 Campos da tela de edição

Nome	Descrição	Grupo	Requisitos de conteúdo	Requisitos de edição	Requisitos Diversos
Legenda	Legenda que irá ser apresentada no cliente.	Admin	Texto	—	—
Data de início	Data em que a publicação começará a ser exibida no cliente.	Admin	Inteiro selecionável entre as opções apresentadas	—	—
Data de Término	Data em que a publicação deixará de ser exibida no cliente.	Admin	Inteiro selecionável entre as opções apresentadas	—	—

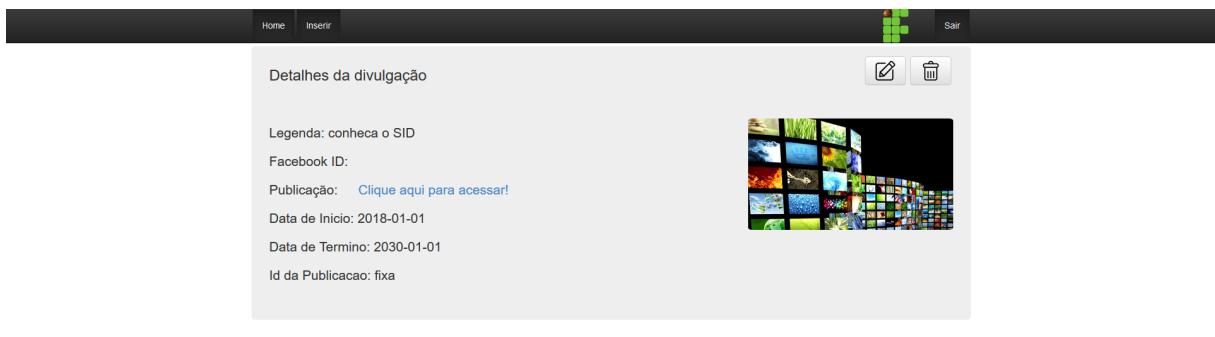
- Comandos

Tabela A.7 Comandos da tela de edição

Nome	Descrição	Grupo	Requisitos de validade	Requisitos Diversos
Atualizar	Realiza a atualização no banco com os novos dados	admin	Todos os campos devem estar preenchidos	—
Cancelar	Cancela as alterações e retorna a página de listagem	admin	—	—

A.1.2.5 Interface do Administrador (Detalhar)

- Leiaute

**Figura A.5** Tela de detalhamento das publicações

- Campos
Não se aplica.
- Comandos

Tabela A.8 Comandos da tela de detalhamento

Nome	Descrição	Grupo	Requisitos de validade	Requisitos Diversos
Clique Aqui	Acessa aolink da publicação no Facebook.	admin	–	–
Editar	O ícone de editar, retorna pagina de edição.	admin	–	–
Deletar	O ícone de excluir, exclui a publicação do banco e do Cliente.	admin	–	–

A.1.2.6 Interface Cliente

- Leiaute

**Figura A.6** Tela do cliente

- Campos
Não se aplica.
- Comandos
Não se aplica.

A.2 Diagramas

A.2.1 Diagramas de casos de uso

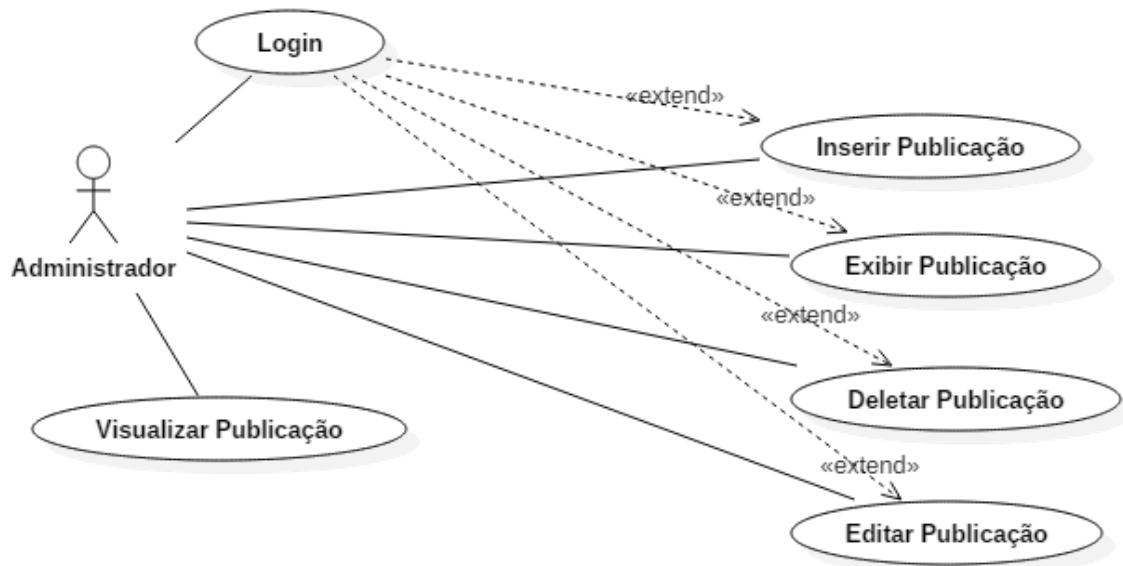


Figura A.7 Diagrama de casos de uso do módulo administrador



Figura A.8 Diagrama de casos de uso do módulo cliente

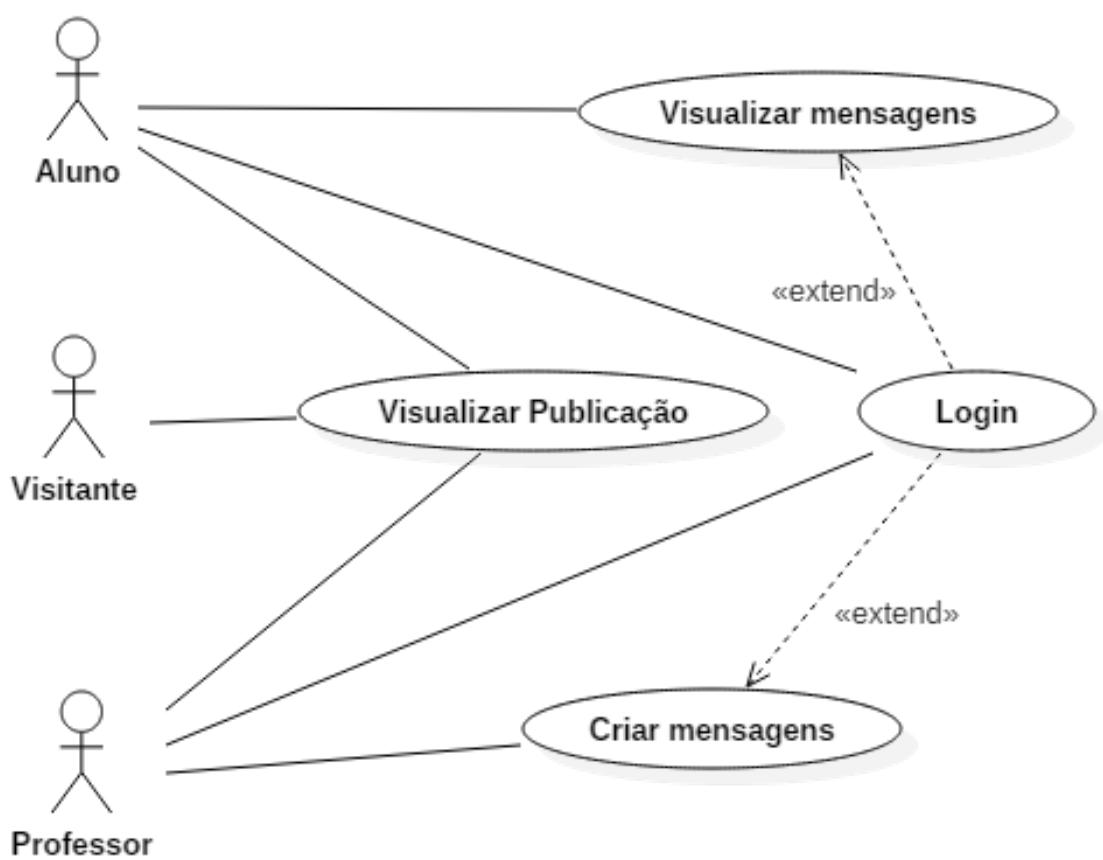


Figura A.9 Diagrama de casos de uso do aplicativo móvel

A.2.2 Diagramas de classe

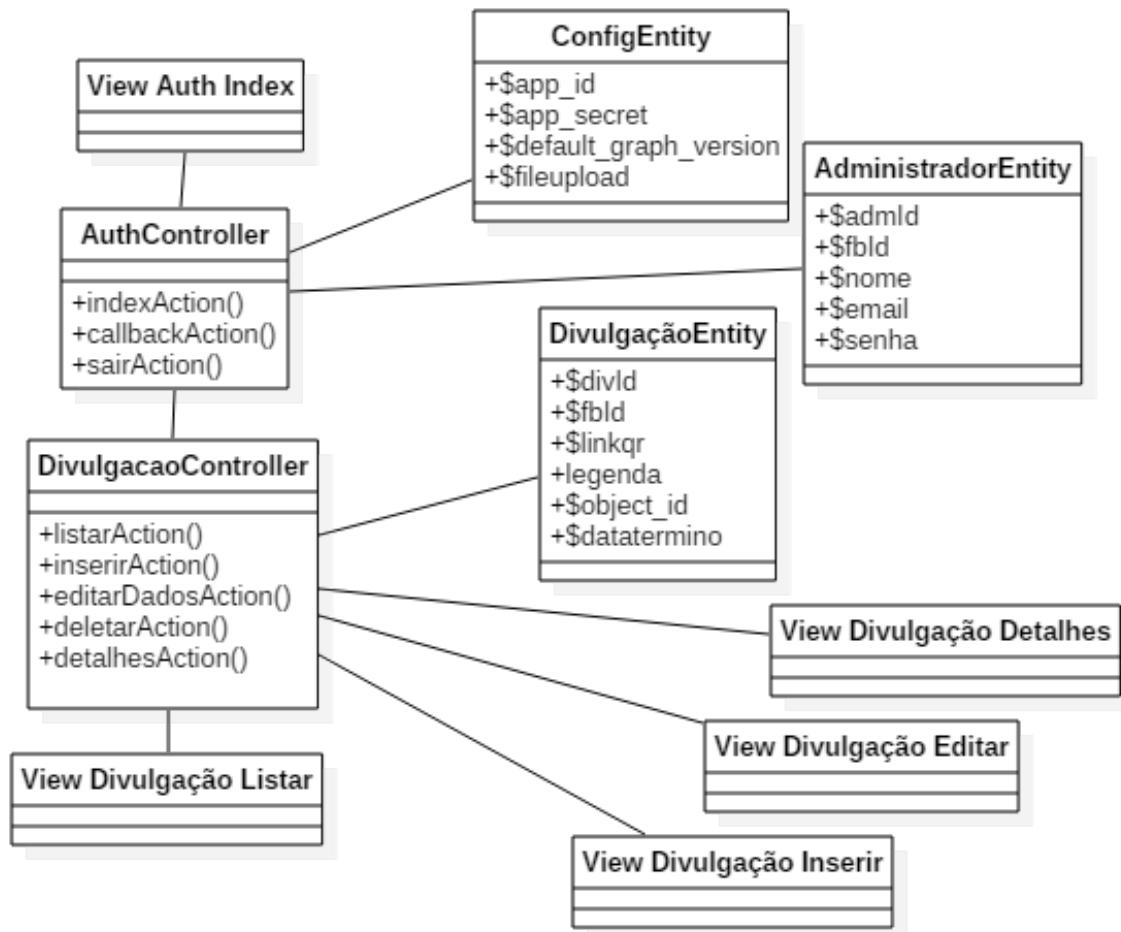


Figura A.10 Diagrama de classes do módulo administrador

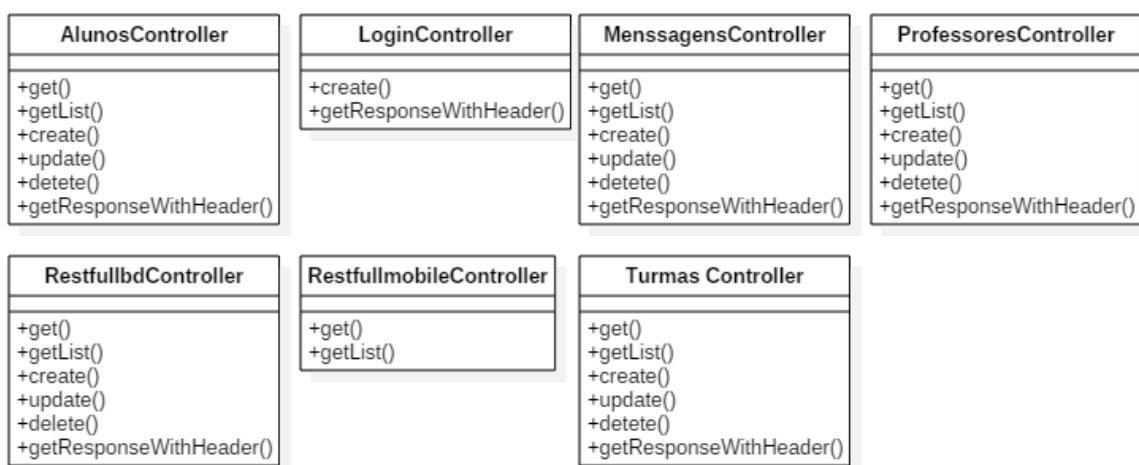


Figura A.11 Diagrama de classes do submódulo API

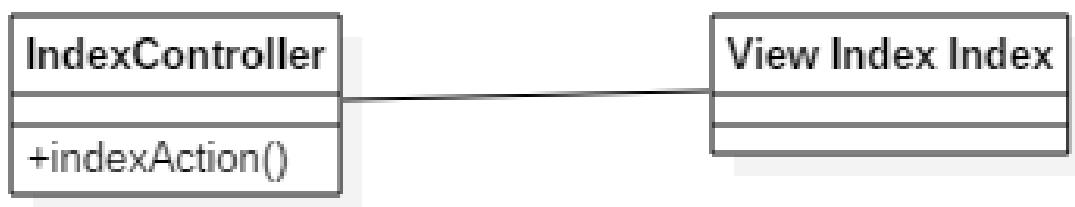


Figura A.12 Diagrama de classes do módulo Cliente

A.2.3 Diagramas entidade–relacionamento

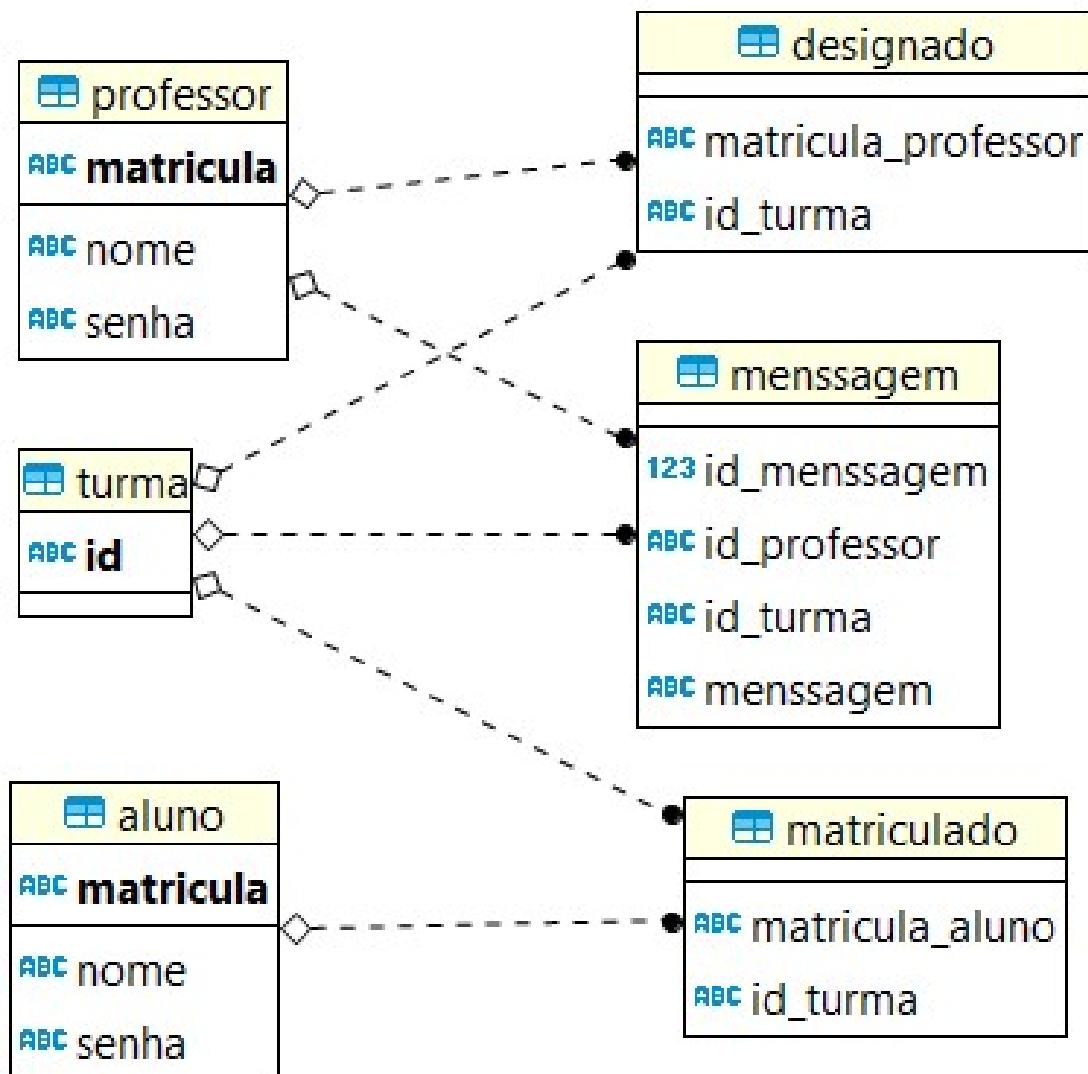


Figura A.13 Diagrama entidade–relacionamento do aplicativo móvel (API fictícia)

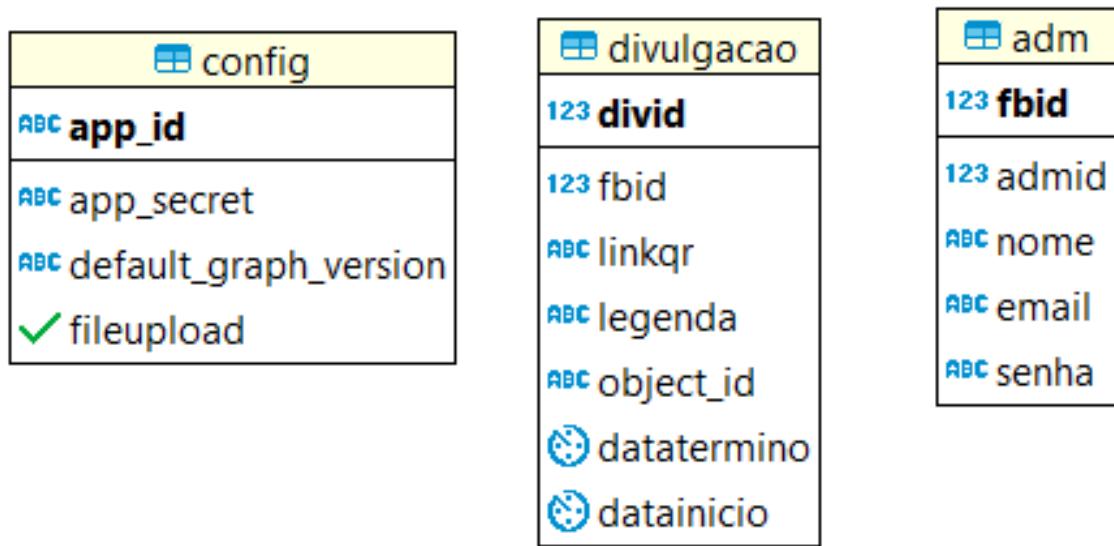


Figura A.14 Diagrama entidade–relacionamento do *web*

A.2.4 Diagramas de sequência

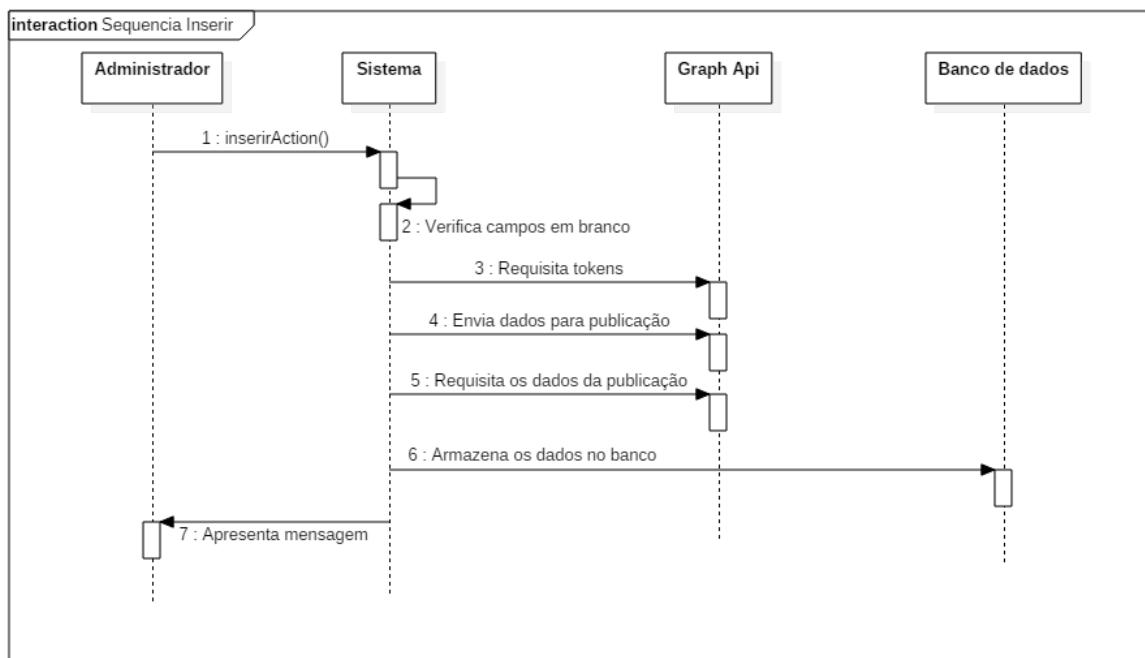
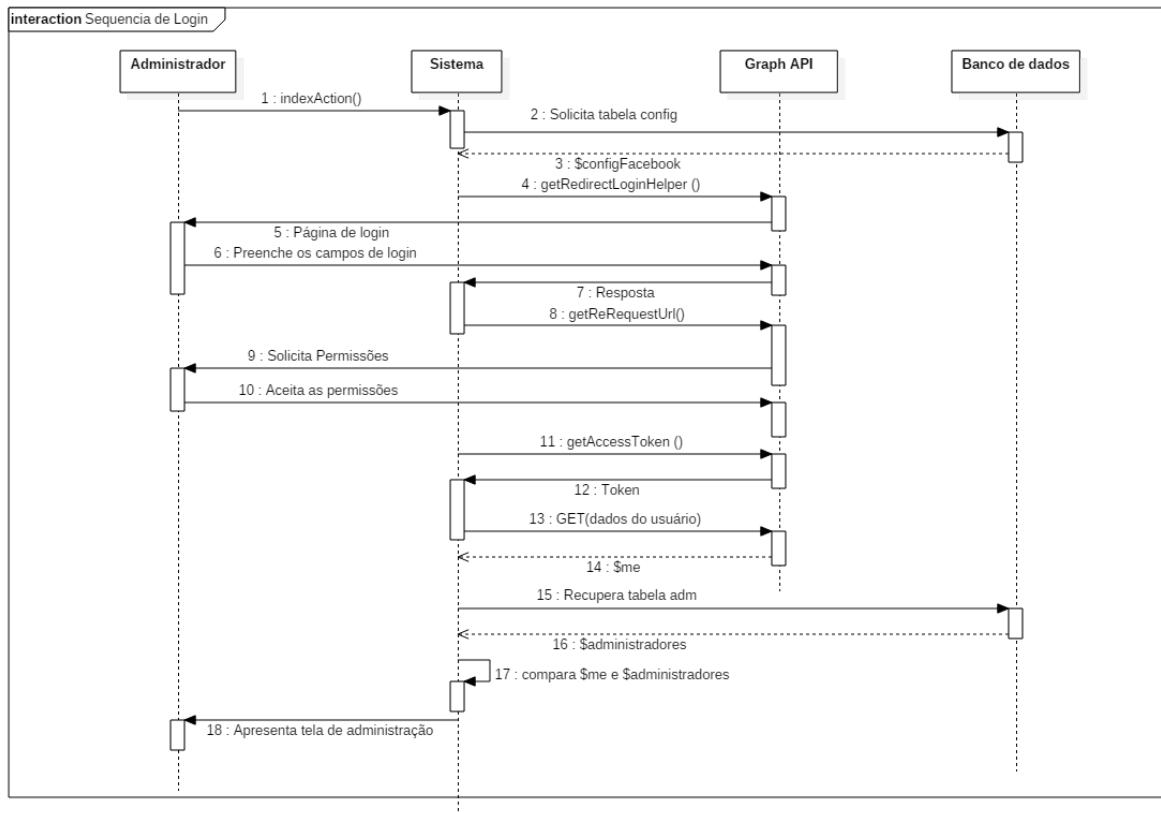
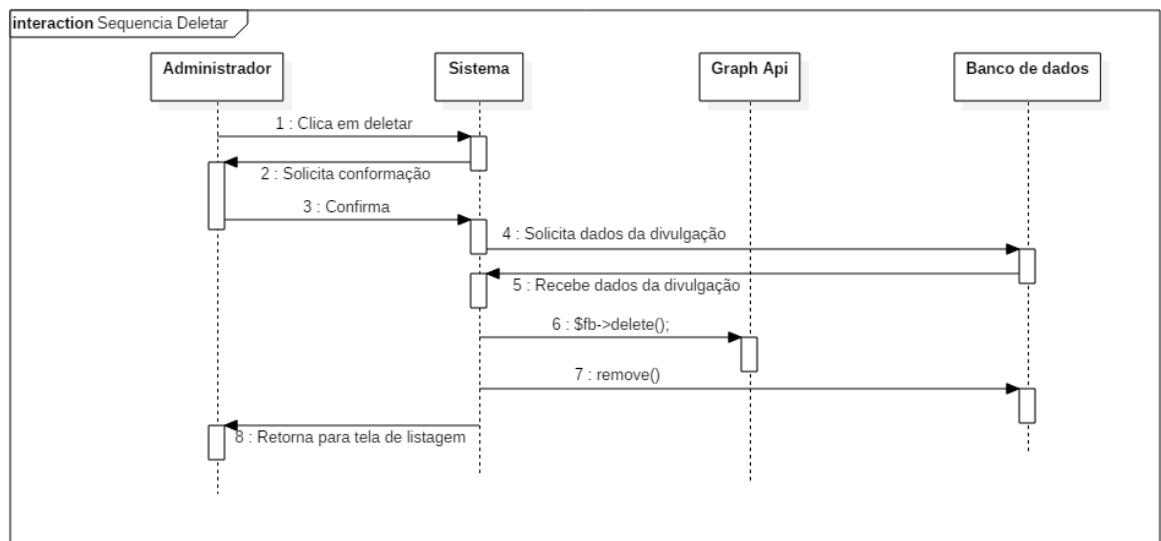


Figura A.15 Diagrama de sequência para inserir

**Figura A.16** Diagrama de sequência para login**Figura A.17** Diagrama de sequência para deletar

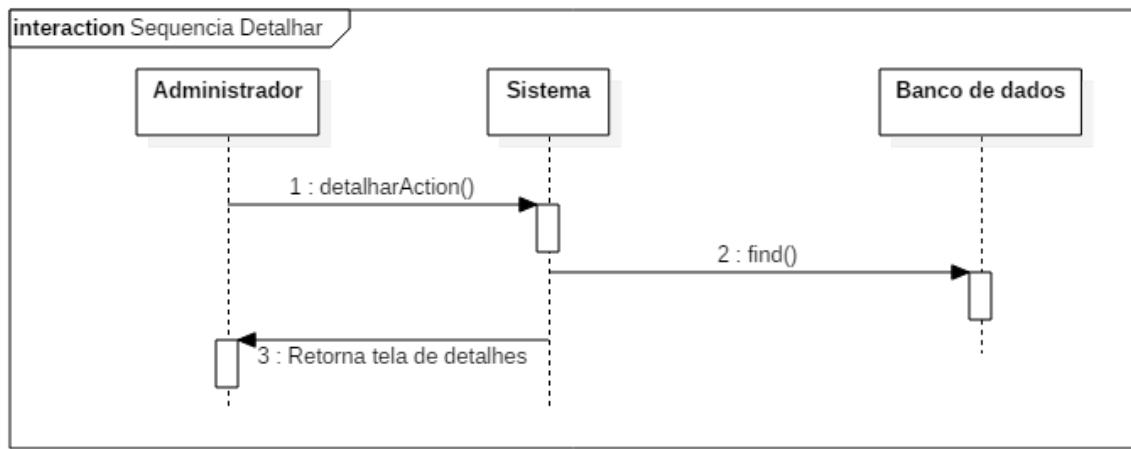


Figura A.18 Diagrama de sequência para detalhar

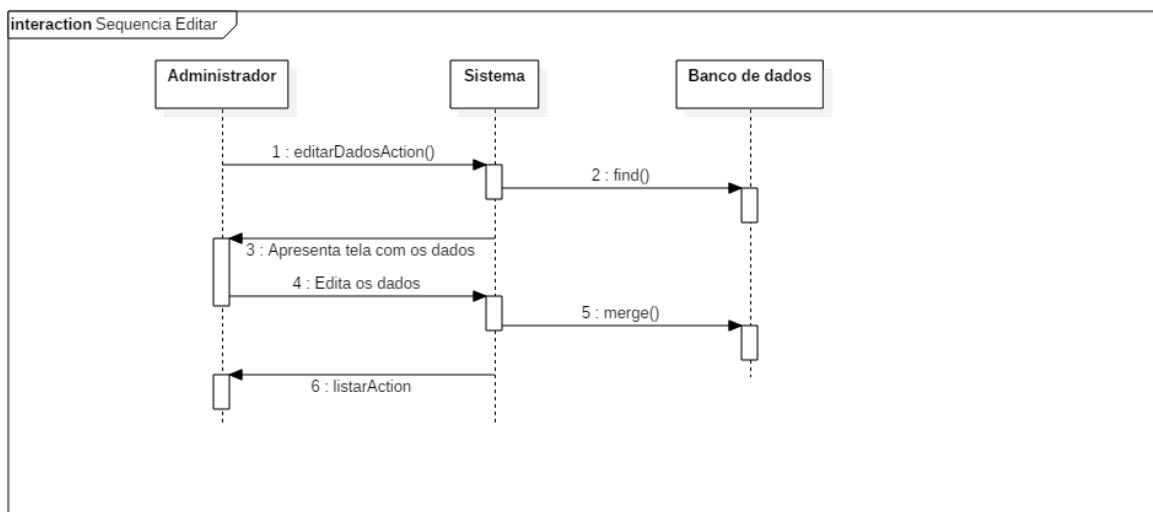


Figura A.19 Diagrama de sequência para editar

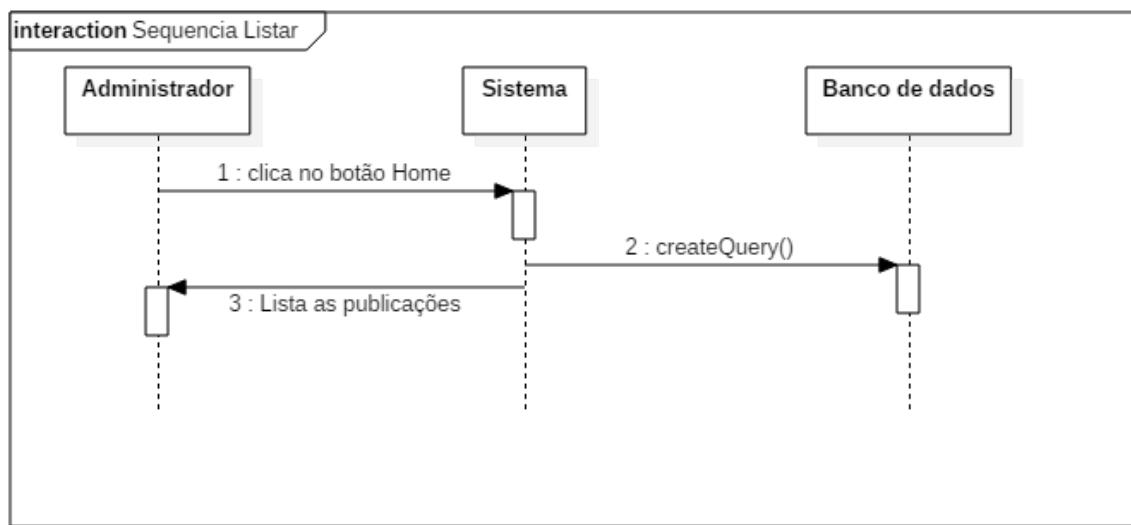


Figura A.20 Diagrama de sequência para listar

A.3 Artefatos

Tabela A.9 Requisitos de usuário

- [RU01] O sistema deverá permitir login dos administradores através do Facebook ou usando e-mail e senha do usuário cadastrados, autorizando que acessem ao sistema e nele realize as funções permitidas.
- [RU02] O sistema deverá permitir que o usuário devidamente cadastrado consiga criar publicações.
- [RU03] O sistema deverá permitir que o usuário devidamente cadastrado consiga editar as publicações que foram criadas.
- [RU04] O sistema deverá permitir que o usuário devidamente cadastrado consiga excluir uma ou mais publicações criadas.
- [RU05] O sistema deverá confirmar se o usuário deseja excluir uma publicação.
- [RU06] O sistema deverá listar todas as publicações criadas.
- [RU07] O sistema deverá armazenar todas as informações que forem inseridas para que possam serem feitas as edições, listagens e exclusões.
- [RU08] O sistema deverá possuir uma página de acesso público.
- [RU09] O sistema deverá apresentar apenas as publicações e comentários autorizados.
- [RU10] O sistema deverá possuir internet conectada.
- [RU11] O sistema deverá funcionar nos navegadores de internet dos sistemas operacionais Linux, Windows.

ESPECIFICAÇÃO DE REGRAS DE NEGÓCIO:

[RN1] Formato logar com Facebook

Nome	Formato
E-mail / Telefone	No mínimo 1 no máximo 50 caracteres.
Senha	No mínimo 1 no máximo 255 caracteres.

O usuário deve ter uma conta com um email/telefone e uma senha vinculadas ao Facebook para poder realizar o login.

[RN2] Formato editar publicação

Nome	Formato
Legenda	No mínimo 1 no máximo 50 caracteres.
Data de início [DIA]	Deve possuir apenas números, limitados ao valor 1 a 30.
Data de início [MES]	Deve possuir apenas números, limitados ao valor 1 a 12.
Data de início [ANO]	Deve possuir apenas números, limitados ao ano atual e mais 3 anos subsequentes.
Data de término [DIA]	Deve possuir apenas números, limitados ao valor 1 a 30.
Data de término [MES]	Deve possuir apenas números, limitados ao valor 1 a 12.
Data de término [ANO]	Deve possuir apenas números, limitados ao ano atual e mais 3 anos subsequentes.

Todos os campos apresentados na [RN2] são de preenchimento obrigatório para efetivação da operação.

[RN3] Formato inserir publicação

Nome	Formato
Legenda	No mínimo 1 no máximo 50 caracteres.
Texto	No mínimo 1 no máximo 255 caracteres.
Data de início [DIA]	Deve possuir apenas números, limitados ao valor 1 a 30.
Data de início [MES]	Deve possuir apenas números, limitados ao valor 1 a 12.
Data de início [ANO]	Deve possuir apenas números, limitados ao ano atual e mais 3 anos subsequentes.
Data de término [DIA]	Deve possuir apenas números, limitados ao valor 1 a 30.
Data de término [MES]	Deve possuir apenas números, limitados ao valor 1 a 12.
Data de término [ANO]	Deve possuir apenas números, limitados ao ano atual e mais 3 anos subsequentes.
Imagen	Imagen do tipo JPEG, JPG ou GIF.

Todos os campos apresentados na [RN3] são de preenchimento obrigatório para efetivação da operação.

REQUISITOS DO SISTEMA:

Funcional

[RF01] Incluir Publicações	
DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá permitir incluir novas publicações no sistema. Os dados necessários para uma nova publicação são: legenda, texto, data de início, data de término e uma imagem. Todos os campos são obrigatórios. Caso encontre erro ao inserir, o sistema apresentará uma mensagem com o erro.	[RU02]

Prioridade:	<input checked="" type="checkbox"/> Essencial
	<input type="checkbox"/> Importante
	<input type="checkbox"/> Desejável

[RF02] Listar Publicações	
DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá permitir listar as publicações que já estão cadastradas no sistema.	[RU06] [RU07]

Prioridade:	<input checked="" type="checkbox"/> Essencial
	<input type="checkbox"/> Importante
	<input type="checkbox"/> Desejável

[RF03] Editar Publicações	
DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá permitir editar as publicações que já estão cadastradas no sistema. Os campos que podem ser alterados são: legenda, data de início e data de término. Todos os campos são obrigatórios.	[RU03] [RU07]

Prioridade:	<input checked="" type="checkbox"/> Essencial
	<input type="checkbox"/> Importante
	<input type="checkbox"/> Desejável

[RF04] Deletar Publicações

DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá permitir deletar as publicações que estão cadastradas no sistema. Será exibido uma notificação pedindo a confirmação do administrador para deletar. Ao confirmar, será feito a exclusão dos dados no banco de dados e no Facebook.	[RU04] [RU05] [RU07]

Prioridade:	X	Essencial
		Importante
		Desejável

[RF05] Autorização

DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá solicitar que o usuário faça login pelo Facebook para que tenha acesso a áreas restritas do sistema.	[RU01]

Prioridade:	X	Essencial
		Importante
		Desejável

Não -funcional

[RNF01] Portabilidade

DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá poder ser executado em diversas plataformas, devido à diversificação de sistemas operacionais existente no ambiente de operação.	[RU11]

Prioridade:

X	Essencial
	Importante
	Desejável

[RNF02] Mantenabilidade

DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá ter a capacidade de não inserir dados incompletos, ou em branco.	

Prioridade:

X	Essencial
	Importante
	Desejável

[RNF02] Disponibilidade

DESCRIÇÃO REQUISITO	REQUISITOS USUÁRIO ATENDIDOS
O sistema deverá ter a capacidade fornecer dados para o usuário quando solicitado.	[RU10]

Prioridade:

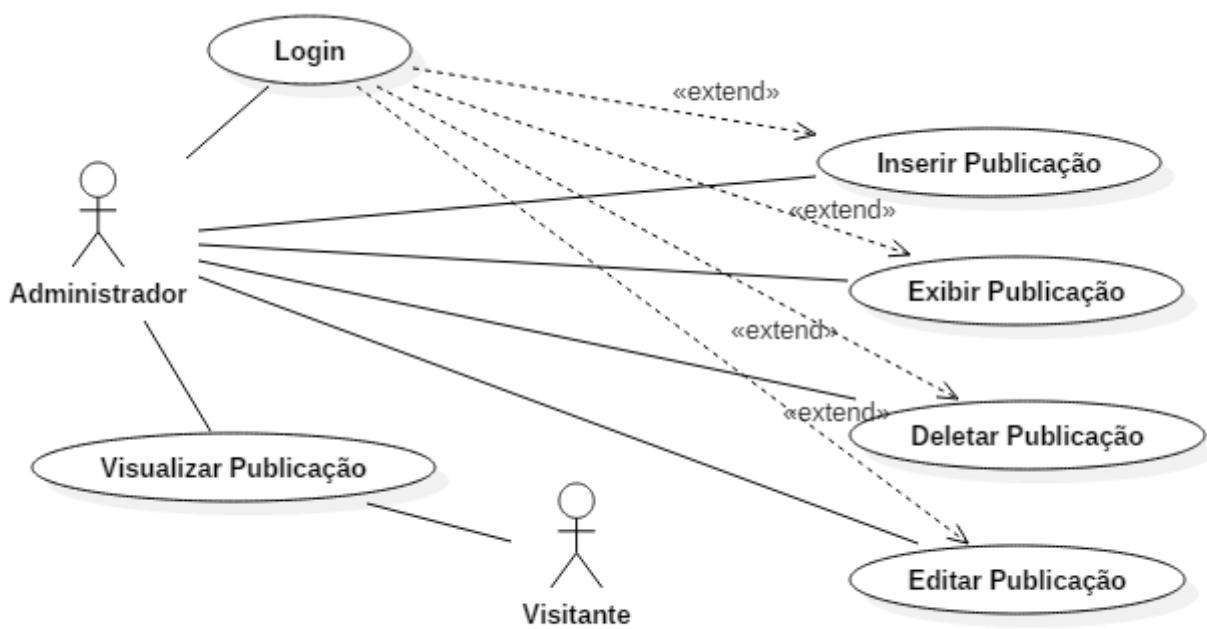
X	Essencial
	Importante
	Desejável

IDENTIFICAÇÃO DOS ATORES DO SISTEMA

Descrição	(Administrador)
Papel	Manutenção, criação, exibição e exclusão de publicações.

Descrição	(Visitante)
Papel	Leitura do QRCode.

VISÃO GERAL SISTEMA - MODELO CASO DE USO



ESPECIFICAÇÃO DE CASO DE USO

[ECU01] Login			
Atores:	Administrador		
Descrição	Logar com os dados cadastrados no banco de dados.		
Pré-condições	Estar conectado a internet ser cadastrado como usuário administrador do sistema.		
Pós-condições:			
Fluxo Principal (FP)			
ID	Ação do ator	ID	Ação / resposta do sistema
[P1]	Clicar no botão “Logar com Facebook”	[P2]	Realizar operação de acordo com [RN01];
[P3]	Aceitar as permissões solicitadas	[P4]	Retornar para a página inicial do sistema.
Fluxo Alternativo (FA)			
ID	Ação do ator	ID	Ação / resposta do sistema
[A1]	Em [P2] Usuário clicou em cancelar	[P5]	Retorna a página de login.
Fluxo de Exceção (FE)			
ID	Ação do ator	ID	Ação / resposta do sistema
[E1]	Em [P2] Usuário não possui conta no Facebook	[P5]	Retorna a página de login e informa o erro.
[E2]	Em [P2] Usuário não aceitou as permissões	[P6]	Retorna a página de login e informa o erro.
[E3]	Em [P2] Usuário não cadastrado no banco	[P7]	Retorna a página de login e informa o erro.

[ECU02] Inserir publicação			
Atores:	Administrador		
Descrição	Insere as publicações usando o sistema, as publicações são enviadas para o Facebook e para o banco de dados.		
Pré-condições	Estar conectado a internet ser cadastrado como usuário administrador do sistema.		
Pós-condições:	O sistema armazenará as informações no banco de dados e no Facebook.		
Fluxo Principal (FP)			
ID	Ação do ator	ID	Ação / resposta do sistema

		[P1]	Realizar operação de acordo com [RN01];
[P2]	O ator clicará em inserir;	[P3]	O sistema apresentará a página para criação de uma nova publicação.
		[P4]	Será apresentado a tela que deve seguir de acordo com [RN03].
[P5]	O ator clicará em publicar;	[P6]	O sistema apresentará a mensagem de inserido com sucesso.
[P7]	O ator clicará em sair	[P8]	O sistema fará o encerramento da sessão do usuário.

Fluxo Alternativo (FA)

ID	Ação do ator	ID	Ação / resposta do sistema
[A1]	Em [P5] o usuário clicar em cancelar.	[P9]	O sistema irá limpar os campos preenchidos pelo usuário e será exibido o [ECU05].

Fluxo de Exceção (FE)

ID	Ação do ator	ID	Ação / resposta do sistema
[E1]	Em [P5] o ator não preencheu todos os campos e tentou salvar;	[P10]	O sistema irá apresentar mensagem informando erro de preenchimento dos dados.

[ECU03] Excluir publicação

Atores:	Administrador
Descrição	A partir do sistema, será possível excluir as publicações feitas no Facebook e no banco de dados.
Pré-condições	Ter o conteúdo publicado no Facebook e cadastrado no banco de dados.
Pós-condições:	Excluirá os dados.

Fluxo Principal (FP)

ID	Ação do ator	ID	Ação / resposta do sistema
		[P1]	Realizar operação de acordo com [RN01];
		[P2]	O sistema seguirá de acordo com a [ECU05].
[P3]	O ator clicará em excluir uma publicação específica;	[P4]	O sistema irá apresentar uma notificação perguntando ao ator se deseja ou não excluir;
[P5]	O ator clicará em Ok;	[P6]	O sistema seguirá de acordo com a [ECU05], não apresentando mais a publicação excluída.

[P7]	O ator clicará em sair	[P8]	O sistema fará o encerramento da sessão do usuário.
------	------------------------	------	-----------------------------------------------------

Fluxo Alternativo FA

ID	Ação do ator	ID	Ação / resposta do sistema
[A1]	Em [P5] se o usuário optou por não excluir e clicou em cancelar;	[P9]	O sistema seguirá de acordo com a [ECU05], sem nenhuma alteração.

Fluxo de Exceção

ID	Ação do ator	ID	Ação / resposta do sistema
[E1]	Em [P6] o sistema não conseguiu excluir a publicação;	[P10]	O sistema apresentará a mensagem com o erro;

[ECU04] Editar publicação

Atores:	Administrador
Descrição	Editar as publicações armazenadas no banco de dados.
Pré-condições	Ter o conteúdo cadastrado no banco de dados.
Pós-condições:	Editará os dados que forem modificados.

Fluxo Principal (FP)

ID	Ação do ator	ID	Ação / resposta do sistema
		[P1]	Realizar operação de acordo com [RN01];
		[P2]	O sistema seguirá de acordo com a [ECU05].
[P3]	O ator clicará no botão editar;	[P4]	O sistema apresentará a tela de acordo com a [RN02];
		[P5]	O ator editará algum dado;
[P6]	O ator clicará em atualizar;	[P7]	O sistema seguirá de acordo com a [ECU05], com as devidas modificações.
[P8]	O ator clicará em sair	[P9]	O sistema fará o encerramento da sessão do usuário.

Fluxo Alternativo (FA)

ID	Ação do ator	ID	Ação / resposta do sistema
[A1]	Em [P6] o ator clicou em cancelar.	[P10]	O sistema seguirá de acordo com a [ECU05], sem nenhuma alteração.

Fluxo de Exceção (FE)

ID	Ação do ator	ID	Ação / resposta do sistema
----	--------------	----	----------------------------

[E1]	Em [P7] o ator deixou algum campo em branco.	[P11]	O sistema retornar uma mensagem com o erro.
------	----------------------------------------------	-------	---------------------------------------------

[ECU05] Listar publicação			
Atores:		Administrador	
Descrição		Listar as publicações cadastradas no banco de dados.	
Pré-condições		Ter o conteúdo cadastrado no banco de dados.	
Pós-condições:		Listará os dados que foram inseridos anteriormente.	
Fluxo Principal (FP)			
ID	Ação do ator	ID	Ação / resposta do sistema
		[P1]	Realizar operação de acordo com [RN01];
[P2]	O ator clicará em Home	[P3]	O sistema carregará a página com todas as publicações que estão armazenadas no banco;
[P4]	O ator clicará em sair	[P5]	O sistema fará o encerramento da sessão do usuário.
Fluxo Alternativo FA			
ID	Ação do ator	ID	Ação / resposta do sistema
[A1]	Em [P4] o ator clicará em editar;	[P6]	Seguirá conforme [ECU04];
[A2]	Em [P4] o ator clicará em excluir;	[P7]	Seguirá conforme [ECU03];
[A3]	Em [P4] o ator clicará em inserir;	[P8]	Seguirá conforme [ECU02];
[A4]	Em [P4] o ator clicará sobre a imagem;	[P10]	O sistema apresentará a pagina com todos os detalhes da publicação que foi selecionada.
Fluxo de Exceção			
ID	Ação do ator	ID	Ação / resposta do sistema
[E1]	Não possui nenhuma publicação ou houve erro com o banco de dados;	[P11]	Será apresentado a informação de que não possui nenhuma publicação;

[ECU06] Visualizar publicação	
Atores:	
Visitante, Administrador	

Descrição	Visualizar as publicações que estão sendo apresentadas		
Pré-condições			
Pós-condições:			
Fluxo Principal (FP)			
ID	Ação do ator	ID	Ação / resposta do sistema
		[P1]	Sistema exibe as publicações
[P2]	Autor visualiza a publicação		
[P3]	Autor acessa ao link QR code		
		[P4]	Sistema altera a publicação
Fluxo Alternativo FA			
ID	Ação do ator	ID	Ação / resposta do sistema
[A1]	Em [P3] autor não acessa ao QR code	[P4]	Sistema altera a publicação
Fluxo de Exceção			
ID	Ação do ator	ID	Ação / resposta do sistema