

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEONARDO AUGUSTO CAVICHIA COTOVICZ

**USO DO MODELO INCREMENTAL PARA O DESENVOLVIMENTO DE
UM MÓDULO DE CAPTURA DE *LEADS* PARA O VTIGER CRM**

PONTA GROSSA

2021

LEONARDO AUGUSTO CAVICHIA COTOVICZ

**USO DO MODELO INCREMENTAL PARA O DESENVOLVIMENTO DE
UM MÓDULO DE CAPTURA DE *LEADS* PARA O VTIGER CRM**

**Use of incremental model for the development of a lead capture
module for Vtiger CRM**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. MSc. Vinícius Camargo Andrade

Coorientador: Prof. Dr. Richard Duarte Ribeiro

PONTA GROSSA

2021

LEONARDO AUGUSTO CAVICHIA COTOVICZ

**USO DO MODELO INCREMENTAL PARA O DESENVOLVIMENTO DE
UM MÓDULO DE CAPTURA DE *LEADS* PARA O VTIGER CRM**

Trabalho de Conclusão de curso apresentado como requisito para obtenção do título de Tecnólogo em Análise em Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 31/agosto/2021

Vinícius Camargo Andrade
Mestre
Universidade Tecnológica Federal do Paraná

Richard Duarte Ribeiro
Doutor
Universidade Tecnológica Federal do Paraná

Diego Roberto Antunes
Doutor
Universidade Tecnológica Federal do Paraná

Luiz Rafael Schmitke
Mestre
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2021

AGRADECIMENTOS

Agradeço a toda instituição por me proporcionar uma excelente educação superior, assim como todos os professores com que tive contato, cada um me apresentou um conhecimento novo.

Agradeço aos meus orientadores Prof. MSc. Vinícius Camargo Andrade e Prof. Dr. Richard Duarte Ribeiro pela oportunidade que tive de receber seus conhecimentos tanto pelas aulas do curso, bem como sendo seu orientando.

Agradeço a todos os colegas de curso pelo qual tive o prazer de estudar junto, seja por um simples trabalho em equipe, ou até mesmo durante toda a graduação.

Agradeço a minha Vó Nadir e minha Mãe Andreia por todos os dias em que elas me incentivavam a terminar a graduação, fazendo com que eu nunca desistisse dos meus propósitos.

Agradeço a minha Madrinha por me auxiliar desde a escolha do curso, bem como seu apoio para me manter em outra cidade, propiciando que eu pudesse concluir o curso com qualidade.

Agradeço aos meus pais, irmãos e familiares, por todo o apoio e carinho.

Agradeço a minha cachorra Loba por sempre me receber com alegria sempre que retornava a minha cidade natal depois um longo semestre.

Agradeço a minha Namorada pela companhia, paciência e auxílio no desenvolvimento deste trabalho.

RESUMO

As empresas têm buscado, cada vez mais, alternativas para se relacionar com seus clientes, sendo necessário traçar estratégias de marketing que sejam atrativas aos seus futuros contatos. Nesse sentido uma das ferramentas que pode auxiliar, é o Vtiger CRM, o qual armazena dados relevantes de seus consumidores, além de possibilitar a busca de novos possíveis clientes, conhecidos como *leads*. Com isso, utilizando-se das ferramentas de desenvolvimento disponíveis, o objetivo desse trabalho foi implementar um módulo de integração para o Vtiger CRM, de modo que auxilie na captura de novos *leads* gerados por meio de publicidades em redes sociais. As redes sociais escolhidas para o desenvolvimento da integração, foram o Facebook e Instagram, isto porque, além de possuírem diversos clientes em potencial, pertencem à mesma corporação, possibilitando a utilização da mesma base de código para ambos os sistemas. O desenvolvimento ocorreu por meio do modelo incremental e as fases desse modelo consistiram em: definir requisitos; atribuir requisitos aos incrementos; projetar a arquitetura do sistema; desenvolver incrementos do sistema; validar incrementos; realizar o projeto do sistema com reuso; validar o sistema; e, por fim, concluir o sistema final. Como resultado, o projeto possibilitou um novo meio de capturar leads para o Vtiger CRM, proporcionando oportunidades de negócios com novos clientes.

Palavras-chave: Serviços ao cliente; Mídias sociais; Modelos de desenvolvimento; Engenharia de software; API; integração web.

ABSTRACT

Companies have increasingly sought alternatives to be able to relate to their customers, making it necessary to devise marketing strategies that are attractive to their future contacts. One of the tools that can help in this regard is Vtiger CRM, which stores relevant data from its consumers, in addition to making it possible to search for new potential customers, known as leads. Thus, using the development tools available, the objective of this work was to implement an integration module for Vtiger CRM, in order to assist in capturing new leads generated through advertising on social networks. The social networks chosen for the development of the integration were Facebook and Instagram, because, in addition to having several potential customers, they belong to the same Corporation, enabling the use of the same code base for both systems. The development took place through the incremental model and the phases of this model consisted of: defining requirements; assign requirements to increments; design the system architecture; develop system increments; validate increments; carry out the system design with reuse; validate the system; and finally complete the final system. As a result, the project enabled a new way to capture leads for Vtiger CRM, providing business opportunities with new customers.

Keywords: Customer services; Social media; Development models; Software engineering; API; web integration.

LISTA DE FIGURAS

FIGURA 1 – MENSAGENS DE REQUISIÇÕES	21
FIGURA 2 – MENSAGENS DE RESPOSTAS	21
FIGURA 3 – FLUXO DE AUTORIZAÇÃO	24
FIGURA 4 – FLUXO DE EVENTOS	25
FIGURA 5 – COMPONENTES PRINCIPAIS DO AJAX E SEU FLUXO DE UTILIZAÇÃO	26
FIGURA 6 – EXEMPLO DIAGRAMA DE CASO DE USO	29
FIGURA 7 – EXEMPLO DIAGRAMA ATIVIDADES	31
FIGURA 8 – DIAGRAMA DE ATIVIDADES COM PARTIÇÕES.....	32
FIGURA 9 - MODELO INCREMENTAL	33
FIGURA 10 – MÓDULO DE LEADS VISÃO DE CRIAÇÃO DO REGISTRO	37
FIGURA 11 – PADRÃO MVC COM CAMADAS E RESPONSABILIDADES	42
FIGURA 12 – CASO DE USO DO MÓDULO	48
FIGURA 13 – DIAGRAMA DE ATIVIDADE GERENCIAR PÁGINAS	49
FIGURA 14 – DIAGRAMA DE ATIVIDADE VALIDAR NOVO LEAD.....	50
FIGURA 15 – MODELO ENTIDADE-RELACIONAMENTO	51
FIGURA 16 – ÁREA DE CONFIGURAÇÃO DO MÓDULO	53
FIGURA 17 – SOLICITAÇÃO DE AUTORIZAÇÃO A RECURSOS.	54
FIGURA 18 – ÁREA DE CONFIGURAÇÃO AUTENTICADO	55
FIGURA 19 – EXEMPLO DE PÁGINA COM UM FORMULARIO DE DADOS.....	57
FIGURA 20 – ÁREA DE CONFIGURAÇÃO DE MAPEAMENTOS DE CAMPOS...58	
FIGURA 21 – TESTE 1: ENVIO	59
FIGURA 22 – TESTE 1: RECEBIMENTO	60
FIGURA 23 – TESTE 2: MAPEAMENTO	60
FIGURA 24 – TESTE 2: RECEBIMENTO	61
FIGURA 25 – TESTE 4: ENVIO	62
FIGURA 26 – TESTE 4: RECEBIMENTO	63

LISTA DE QUADROS

QUADRO 1 - DESCRIÇÃO DOS CÓDIGOS DE RETORNO DO PROTOCOLO HTTP	23
QUADRO 2 – CASO DE USO REALIZAR LOGIN	69
QUADRO 3 – CASO DE USO REALIZAR LOGOUT	70
QUADRO 4 – CASO DE USO GERENCIAR CONFIGURAÇÕES ADICIONAIS	70
QUADRO 5 – CASO DE USO GERENCIAR PÁGINAS.....	71
QUADRO 6 – CASO DE USO LISTAR FORMULÁRIOS	72
QUADRO 7 – CASO DE USO MAPEAR CAMPOS	73
QUADRO 8 – CASO DE USO DISPARAR <i>WEBHOOK</i>	74
QUADRO 9 – CASO DE USO VALIDAR NOVO <i>LEAD</i>	75

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

AJAX	<i>Asynchronous JavaScript and XML</i>
API	<i>Application Programming Interface</i>
CRM	<i>Customer Relationship Management</i>
CSS	<i>Cascading Style Sheets</i>
CSV	<i>Comma-separated values</i>
ERP	<i>Enterprise Resource Planning</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model View Controller</i>
ODBC	<i>Open Database Connection</i>
PHP	<i>Hypertext Preprocessor</i>
REST	<i>Representation State Transfer</i>
SDK	<i>Software Development Kit</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	14
1.2	Objetivos Específico	14
1.3	Estrutura do trabalho	15
2	GESTÃO DE RELACIONAMENTO COM O CLIENTE	17
2.1	Tipos de sistemas CRM	17
2.2	Disponibilidade do CRM	18
3	TECNOLOGIAS DE INTERNET	20
3.1	HTTP	20
3.2	REST	22
3.3	OAuth 2.0	23
3.4	Webhooks	25
3.5	AJAX	26
4	UNIFIED MODELING LANGUAGE	28
4.1	Diagrama de casos de uso	29
4.2	Diagrama de atividade	30
5	DESENVOLVIMENTO INCREMENTAL	33
6	VTIGER CRM	36
6.1	Módulo de <i>leads</i>	37
6.2	Tecnologias utilizadas pelo Vtiger	38
6.2.1	Apache	39
6.2.2	PHP	39
6.2.3	MySQL	40
6.2.4	Smarty	40
7	PADRÃO DE SOFTWARE MVC	42
8	DESENVOLVIMENTO	44
8.1	Graph API	44
8.2	Definição dos requisitos genéricos	45
8.3	Atribuir requisitos aos incrementos	46
8.4	Arquitetura do sistema	47
8.4.1	Diagrama de caso de uso	47
8.4.2	Diagrama de atividade	49
8.4.3	Modelo entidade-relacionamento	51
8.5	Desenvolver incrementos	52
8.5.1	Área de configuração do módulo	53
8.5.2	Autenticação e autorização	53
8.5.3	Habilitar recebimento de leads	55
8.5.4	Mapeamento de campos	56

8.5.5	Recebimento de leads via webhook.....	58
8.6	Avaliar incrementos	59
8.6.1	Teste 1.....	59
8.6.2	Teste 2.....	60
8.6.3	Teste 3.....	61
8.6.4	Teste 4.....	62
8.7	Projeto do sistema com reuso	63
8.8	Validar sistema.....	63
9	CONCLUSÃO	65
9.1	Trabalhos futuros.....	65
	REFERÊNCIAS	66
	APÊNDICE A – DESCRIÇÕES DOS CASOS DE USO	68

1 INTRODUÇÃO

Em tempos de transformação digital, as empresas buscam, de modo crescente, maneiras de se organizar e se relacionar com seus clientes. Portanto, por muitas vezes, empresas precisam determinar estratégias de marketing para atrair e fidelizar novos contatos. Atualmente, existem diversas ferramentas, como por exemplo, o CRM (*Customer Relationship Management*), que auxilia empresas na gestão de relacionamentos com clientes, o que resulta em melhores vendas e negócios (GLUOCRM, 2021).

As plataformas de CRM estão entre as tecnologias mais relevantes no ramo empresarial, devido à possibilidade de agrupar contatos, marketing e oportunidades, em um único sistema. Um CRM armazena dados relevantes de clientes, como por exemplo, o nome, endereços e telefone. Além disso, mantêm outras informações sobre clientes, tais como visitas, reuniões, orçamentos, ligações, e-mails e acessos a campanhas publicitárias. Assim, estas informações armazenadas podem ser utilizadas pela equipe de vendas da empresa, que, com o auxílio das ferramentas, podem gerenciar um negócio futuro. Com mais informações sobre o relacionamento com o cliente é possível fideliza-lo de maneira simplificada (SALESFORCE, 2021).

Segundo Salesforce (2021), ao utilizar as funcionalidades de um CRM, a empresa pode buscar e encontrar possíveis clientes, denominado no CRM como **leads**, além de concretizar vendas e fidelizar clientes, o que culmina em um valor estimado de 37% das receitas. As informações sobre clientes são sempre atualizadas e, com isso, ocorre uma melhora no relacionamento entre a empresa e o comprador do serviço, o que pode gerar, como consequência, um aumento de até 45% na satisfação dos clientes.

Segundo Santos (2020), todo *lead* passa por uma jornada de compra e possui um ciclo em uma empresa, oferecendo grandes oportunidades para a mesma. Dessa forma, as empresas realizam o gerenciamento dessas oportunidades, utilizando estratégias e técnicas de gestão.

A primeira etapa desse ciclo é o aprendizado e descoberta, onde o possível cliente ainda não possui interesse em determinados produtos, ou está pesquisando soluções para problemas específicos. Nessa etapa, a empresa precisa atrair a atenção do cliente para a solução que procura.

Na segunda etapa, que consiste no reconhecimento do problema, o *lead* está ciente do problema e necessita da solução. Nessa fase o cliente pode buscar em redes sociais as empresas que podem solucionar seu problema. Além do mais, a empresa precisa estar preparada para atrair o cliente, utilizando-se de bons conteúdos digitais.

A terceira fase do ciclo envolve a solução, portanto, o cliente já possui algumas soluções em mãos e precisa avaliar qual empresa atende melhor seus requisitos. Visto isso, é nessa etapa que as empresas precisam demonstrar que possuem as soluções certas para determinado problema.

Após a análise de algumas soluções, o cliente fecha a sua compra, finalizando a última fase do ciclo, denominada "decisão de compra". Ainda nessa etapa, o cliente pode solicitar demonstrações e testes gratuitos do serviço oferecido.

As jornadas de compras podem ser apoiadas por ferramentas especializadas. Um exemplo a ser citado é o Vtiger CRM (2021a), que possui diversas funções que auxiliam no relacionamento com os clientes. Uma dessas soluções inclui a captura de dados por meio de formulários em sites, em que os clientes podem preencher campos de entrada com dados pessoais. Esses formulários podem ter como objetivo o primeiro contato do cliente com vendedores. Uma segunda maneira de se relacionar com possíveis clientes é a utilização da ferramenta de e-mail marketing. Esta, por sua vez, tem como o objetivo enviar e-mails com propostas e apresentações dos seus serviços a possíveis clientes (VTIGERCRM, 2021a).

Com a crescente expansão da Internet, e com mudanças significativas de hábitos digitais dos consumidores devido a pandemia do SARS-CoV-2, as redes sociais se tornaram uma das melhores ferramentas para o marketing digital. Um usuário passa em média 3 horas e 31 minutos em redes sociais diariamente. Redes sociais como o Facebook¹ e Instagram² possuem, mundialmente, cerca de 2,7 bilhões e 1,15 bilhão de usuários ativos respectivamente, sendo para a primeira 130 milhões somente no Brasil (VOLPATO, 2021).

Por meio de campanhas de publicidade em redes sociais, possíveis clientes podem cadastrar seus dados de contato ou solicitar demonstrações de serviços, sendo que plataformas como Facebook e Instagram possuem diversos clientes em

¹ Site Facebook: <http://www.facebook.com>

² Site Instagram: <http://www.instagram.com>

potencial. Neste contexto, este trabalho propõe a criação de um módulo de integração para a captura de *leads* por meio de redes sociais.

O módulo foi criado utilizando a versão de código aberto do Vtiger CRM³, isto porque, nesta versão já é possível usufruir de diversos módulos, tais como os módulos de *leads*, contatos, organizações, vendas, marketing, entre outros, além disso, o Vtiger possui ferramentas para o desenvolvimento de módulos e plugins. Com isso, o módulo de integração proposto no projeto visa auxiliar o cadastro de novos registros, atuando no módulo de *leads* do CRM. A integração desenvolvida permite realizar o registro de *leads* enviados pelas redes sociais e capturados pelo CRM.

A rede social escolhida para o projeto foi o Facebook e Instagram devido às suas altas taxas de usuários ativos, e também por pertencerem à mesma corporação. Devido a isso, ambas utilizam as mesmas ferramentas de consulta de dados que são utilizadas na captura de *leads*.

Os padrões de desenvolvimento e a linguagem de programação principal foram determinados conforme os utilizados pelo próprio Vtiger CRM. Com isso, a codificação do projeto foi realizada na linguagem PHP (*Hypertext Preprocessor*) com o padrão de arquitetura MVC (*Model View Controller*). O Facebook e Instagram disponibilizam ferramentas para o desenvolvimento na linguagem PHP e estas serão utilizadas para a comunicação entre as redes sociais e o CRM, permitindo a troca de dados entre os sistemas.

1.1 Objetivos

O objetivo geral deste trabalho é implementar um módulo de integração para o Vtiger CRM, que auxilie a captura de novos *leads* gerados por meio das campanhas de marketing publicadas em redes sociais.

1.2 Objetivos Específico

- Estudar a documentação do Vtiger CRM e Graph API.
- Realizar a modelagem do módulo de integração;

³ Site Vtiger CRM: <https://www.vtiger.com>

- Desenvolver um *Webhook* capaz de receber, tratar e salvar os *leads* enviados pelas redes sociais.
- Avaliar o módulo de integração proposto.

1.3 Estrutura do trabalho

O Capítulo 2 apresenta as ferramentas e estratégias que os sistemas CRM dispõem.

O Capítulo 3 exhibe as tecnologias de Internet utilizadas para o desenvolvimento do trabalho. A revisão é composta pelas seguintes tecnologias: HTTP (*Hypertext Transfer Protocol*), REST (*Representation State Transfer*), OAuth 2.0, Webhooks. AJAX (*Asynchronous JavaScript and XML*).

O Capítulo 4 aborda sobre a linguagem visual de modelagem de sistemas chamada *Unified Modeling Language* (UML). A revisão literária é composta pelo tópico principal que descreve as principais características e diagramas da linguagem, seguido pelos tópicos que abordam com maior ênfase os seguintes diagramas da linguagem: Diagrama de caso de uso e Diagrama de atividades.

O Capítulo 5 trata sobre o modelo de desenvolvimento incremental que foi utilizado no trabalho para delimitar as etapas de desenvolvimento e organizar a sequência de passos até a conclusão do projeto.

O Capítulo 6 apresenta o sistema Vtiger CRM que foi utilizado no projeto, juntamente com suas principais tecnologias. Além disso é abordado sobre o módulo de *leads* deste CRM, sendo este módulo foco do trabalho, visto que os *leads* gerados pelas redes sociais são mantidos aqui. O último tópico desse capítulo apresenta as seguintes tecnologias de desenvolvimento que o Vtiger CRM possui: Apache, PHP, Smarty, MySQL.

A estrutura de desenvolvimento MVC (*Model View Controller*) é apresentada no Capítulo 7. O MVC foi utilizado para separar as responsabilidades de lógicas de negócios da apresentação visual do sistema.

O Capítulo 8 descreve sobre o desenvolvimento do módulo de captura de *leads* para o Vtiger CRM. Neste capítulo é apresentada toda a sequência de desenvolvimento da aplicação, bem como os resultados e testes de validação. Após explicar sobre a Graph API, o capítulo utiliza as fases do modelo incremental para

enumerar os tópicos, ou seja, cada tópico é uma etapa do ciclo incremental. Os seguintes tópicos são abordados neste capítulo: Graph API, Definição dos requisitos genéricos, Atribuição dos requisitos aos incrementos, Definição da arquitetura do sistema, Desenvolvimento dos incrementos, Projeto de sistema com reuso de software, Validar o sistema, e por fim, o Sistema completo.

Por fim, o Capítulo 9 apresenta a conclusão do trabalho, bem como os trabalhos a serem desenvolvidos futuramente.

2 GESTÃO DE RELACIONAMENTO COM O CLIENTE

Segundo GluoCRM (2021), a gestão de relacionamento com o cliente, do inglês *Customer Relationship Management* (CRM), é uma mescla de técnicas e métodos para criar e aprimorar o relacionamento com clientes. Um sistema de CRM é comumente se referido ao contexto de utilização de ferramentas que possibilitam empresas operar as seguintes entidades (GLUOCRM, 2021):

- Gerenciamento de contatos;
- Gestão de funil de vendas;
- Campanhas de marketing;
- Atendimento ao consumidor;

Informações como nome, telefone e endereço são valiosas para sistemas CRM. As atividades, como ligações, e-mails e outras formas de contatos com a empresa podem ser trabalhadas para, em determinado momento, gerarem novos clientes e, por fim, fidelizados. O principal papel de um CRM é facilitar a gestão do funil de vendas, movendo os *leads* por diversas etapas até o fechamento da compra (GLUOCRM, 2021).

Sistemas ERP (*Enterprise Resource Planning*) são tecnologias para processos organizacionais, essencialmente financeiros, como emissão de notas fiscais, contas a pagar e receber, relatórios monetários. Empresas que possuem sistemas CRM em seus negócios geralmente solicitam integração entre sistemas ERP (SALESFORCE, 2021).

Segundo Salesforce (2021), o CRM melhora as ferramentas disponíveis para equipes de vendas, contribuindo na jornada de compra e a utilização de técnicas de vendas eficazes. Já o ERP possui módulos específicos para controles financeiros e burocráticos, como a contabilidade da empresa, sendo essa função não requisitada em sistemas CRM.

2.1 Tipos de sistemas CRM

Empresas que buscam utilizar sistemas CRM em suas tarefas, podem encontrar diversos tipos disponíveis no mercado. A escolha se dá pela avaliação do

orçamento disponível, além do planejamento de quais serão as funcionalidades necessárias para o negócio (BAFUTTO, 2019). Cada tipo de sistemas CRM é detalhado a seguir:

- CRM Operacional: utilizando-se de soluções para os processos de negócios, o objetivo do CRM desse modelo é facilitar os serviços de marketing, vendas e automatização de serviços. Empresas que tendem a possuir trabalhos repetitivos em atividades de gerenciamento de clientes, podem reduzir esses processos utilizando um CRM com essa finalidade.
- CRM Analítico: requisitado para tratamento massivo de dados. Utilizando de ferramentas como mineração de dados, inteligência artificial, bem como campanhas de marketing estratégicas, é possível se utilizar de todas as informações de clientes, tornando o relacionamento customizado e único.
- CRM Colaborativo: compartilhar informações entre entidades de negócios, melhorando o fluxo de troca de dados. Um exemplo é o setor responsável por atrair clientes, ele pode filtrar as melhores oportunidades e encaminhá-las para o setor de vendas que pode concretizar um novo negócio, e com isso a equipe de pós-venda pode fidelizar o cliente com novas oportunidades.

2.2 Disponibilidade do CRM

Companhias que disponibilizam sistemas CRM para empresas, podem oferecer dois tipos de disponibilidade para o sistema. A primeira, CRM Local, geralmente é um CRM com licença de uso gratuita e empresas pagam técnicos para implantação em seus próprios servidores. A segunda, CRM Nuvem é distribuído pela empresa mantenedora e o cliente não tem preocupação com instalações e treinamento, pois planos são fechados para qualificar todas as equipes para utilização do sistema (SALESFORCE, 2021). Os detalhes de cada tipo de disponibilidade são vistos a abaixo:

- CRM Local: o sistema CRM é instalado e hospedado em servidores próprios da empresa de negócios. Empresas que necessitam de CRM locais devem arcar com custos de implantação e licença de uso para os sistemas utilizados. As atualizações de softwares devem ser garantidas

com antecedência e normalmente são contratadas equipes de tecnologias específicas para prestação do suporte ao CRM.

- CRM Nuvem: o software é armazenado nos servidores da empresa responsável pelo fornecimento do CRM e, com isso, os usuários (empresas) não precisam se preocupar com manutenção ou atualização de software e hardware. Os dados armazenados em nuvem, podem ser acessados em tempo real e estão sempre disponíveis.

3 Tecnologias de Internet

Este capítulo aborda as tecnologias de Internet utilizadas no desenvolvimento do trabalho. A Seção 3.1 descreve o protocolo HTTP. A Seção 3.2 apresenta o padrão para troca de dados REST. A Seção 3.3 aborda sobre o protocolo de autorização e autenticação OAuth2. A Seção 3.4 detalha os retornos de chamadas HTTP denominados *webhooks*.

3.1 HTTP

HTTP (*Hypertext Transfer Protocol*) é um protocolo que possibilita obter diversos recursos na Internet, como documentos HTML (*HyperText Markup Language*), sendo a base de troca de dados na Internet. HTTP é um protocolo cliente-servidor, ou seja, cada troca de dados consiste em uma solicitação ou resposta. O protocolo fornece uma interface padronizada de como interagir com os dados. Clientes podem solicitar dados de servidores utilizando solicitações (*requests*), e servidores retornam os dados com as respostas (*responses*) (FIELDING, RESCHKE, 2021).

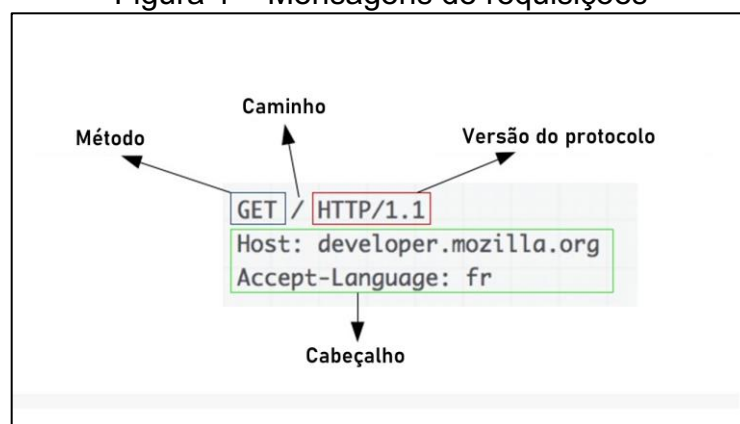
Segundo Fielding e Reschke (2021), as trocas de dados entre cliente e servidores são simples, além disso, o HTTP é extensível, e novas funcionalidades podem ser incluídas com padronizações de trocas de dados entre clientes e servidores. O destino da solicitação HTTP é definido como recurso. Quando o servidor recebe uma solicitação enviada pelo cliente, ele reconstrói o recurso e envia de volta a representação do mesmo.

O fluxo HTTP é composto das seguintes fases (FIELDING, RESCHKE, 2021):

- Uma conexão é aberta e será usada nas requisições e respostas. O usuário pode abrir novas conexões ou usar as existentes.
- Monta e envia uma mensagem HTTP para o servidor, e este retorna os dados solicitados.
- Processa a resposta do servidor.
- Novas requisições podem reutilizar a conexão existente.

As mensagens HTTP podem ser requisições ou respostas, cada uma possui seus próprios elementos (MOZILLA, 2021a). A Figura 1 ilustra os componentes das mensagens de requisições.

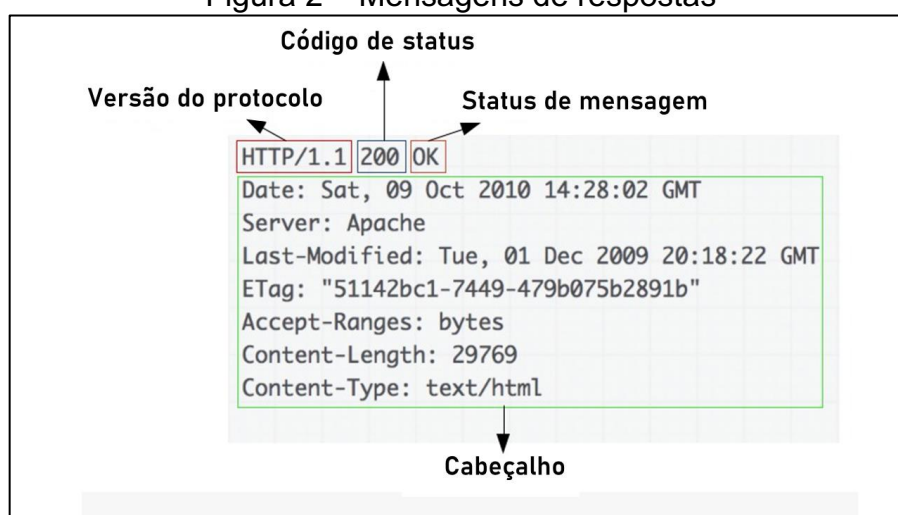
Figura 1 – Mensagens de requisições



Fonte: Adaptado de Mozilla (2021a)

Os componentes de uma requisição são: um método HTTP, que define qual solicitação o usuário deseja fazer; o caminho do recurso; a versão do protocolo HTTP; os cabeçalhos, que contêm informações a serem utilizadas pelos servidores e, por fim, o corpo da requisição, que são dados adicionais enviados pelo cliente. A Figura 2 ilustra os componentes das mensagens de respostas.

Figura 2 – Mensagens de respostas



Fonte: Adaptado de Mozilla (2021a)

Segundo Mozilla (2021a), os componentes de uma resposta são: a versão do protocolo HTTP; o código de status, que indica se a solicitação do usuário foi concluída com sucesso ou não, e informa o porquê; uma mensagem de status com uma breve

informação sobre o código de status; um cabeçalho, igual ao de uma requisição; e por fim, um corpo com recursos opcionais a serem enviados ao cliente.

3.2 REST

Segundo Lecheta (2015), *Representation State Transfer* (REST), é um padrão de arquitetura de sistemas, que auxilia na integração de projetos de softwares. O REST se beneficia do protocolo HTTP para a criação de utilitários que transferem arquivos nos formatos XML⁴ e JSON⁵.

Um sistema, para ser determinado como RESTful necessita seguir os princípios do padrão (LECHETA, 2015):

- A solicitação de recursos deve ser independente e não manter o estado.
- Possui as operações de leitura e escrita padronizadas, neste caso utilizando o protocolo HTTP.
- Identificação única de recurso, no REST é utilizado um identificador, ou seja, cada recurso deve possuir sua própria URI⁶, e é possível consultar informações específicas dele.

O REST determina a utilização explícita do protocolo HTTP, e para isso é utilizado o seguinte mapeamento (LECHETA, 2015):

- GET: utilizado para solicitar um recurso, é comumente utilizado para consultas. Por exemplo, para solicitarmos os dados do cliente de identificação 1: **GET /cliente/1**.
- POST: utilizado para criação de novos recursos. Uma requisição com o objeto em formato JSON ou XML, deve ser enviada no corpo da chamada.

⁴ A XML (*Extensible Markup Language*) é uma linguagem de marcação criada com o propósito de fornecer estruturas simples. O XML pode ser utilizado em diferentes contextos como: intercâmbio de dados via *web*, arquivos de configurações, documentos, dados bancários (W3C, 2021a).

⁵ O JSON (*JavaScript Object Notation*) é um formato para intercâmbio de dados entre sistemas. Embora possua *JavaScript* em seu nome, ele pode ser utilizado por qualquer linguagem que o suporte (BASSETT, 2015).

⁶ A URI (*Uniform Resource Identifier*) é um identificador utilizado para encontrar recursos únicos na internet (BERNERS-LEE, 2021).

- PUT: Utilizado para atualizações de dados. O objeto deve ser enviado no corpo da requisição. Ambos os métodos PUT e POST podem atualizar e criar dados, porém não é comumente empregado.
- DELETE: Utilizado para deletar ou desativar dados. Por exemplo, para deletarmos um cliente de identificação 1: **DELETE /cliente/1**.

Cada requisição retorna um código de resposta do protocolo HTTP. Este código é retornado no cabeçalho, e informa ao requisitante o que ocorreu com sua solicitação. O Quadro 1 mostra a descrição de cada código de retorno:

Quadro 1 - Descrição dos códigos de retorno do protocolo HTTP

Código de retorno	Descrição
100 - 199	Utilizado para informações
200 - 299	Sucesso nas requisições
300 - 399	Redirecionamentos
400 - 499	Erros de solicitações
500 - 599	Erros do servidor

Fonte: Mozilla (2021b)

3.3 OAUTH 2.0

Segundo Hardt (2021), a estrutura de autorização do OAuth 2.0 permite que aplicativos de terceiros solicitem recursos com acesso limitado a outros sistemas. Essa estrutura permite que usuários façam conexão em sites e aplicativos usando suas contas comumente utilizadas como, por exemplo, Facebook, Twitter e Microsoft. O padrão OAuth 2.0 simplifica a estrutura e padrões do OAuth 1.0, tornando este último obsoleto.

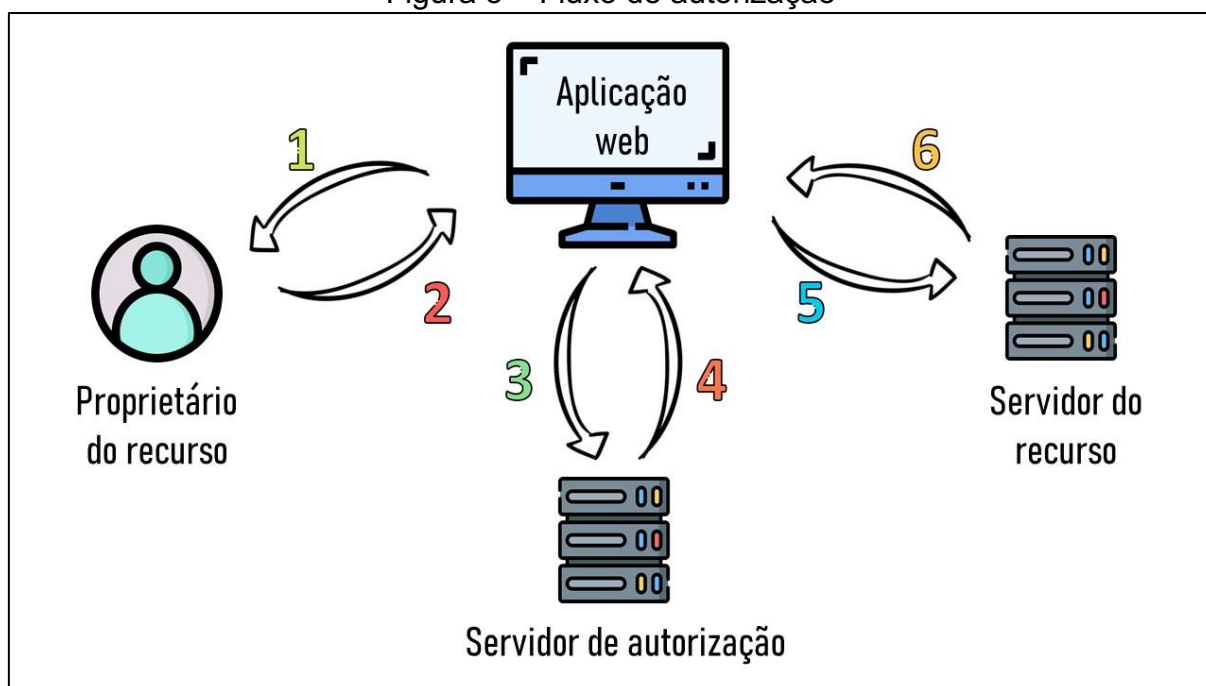
O processo de autorização do OAuth permite que usuários se conectem de maneira segura a recursos de servidores em seu nome, sem expor senhas. São definidos 4 papéis para o processo de autorização seja realizado (HARDT, 2021):

- Proprietário do recurso: concede acesso a recursos protegidos;
- Servidor do recurso: hospeda os recursos, envia e responde às solicitações de informações;

- Aplicação *web* (cliente): se utiliza de chaves de acesso emitidas pelo servidor de autorização para acessar recursos em nome de seu proprietário.
- Servidor de autorização: distribui a chave de acesso ao cliente, quando o proprietário se autenticar com sucesso;

O fluxo de autorização do padrão OAuth 2.0 é ilustrado na Figura 3 e exemplifica a troca de dados entre as 4 entidades descritas acima:

Figura 3 – Fluxo de autorização



Fonte: Autoria própria

A aplicação *web*, na primeira etapa do fluxo, solicita a autorização de uso ao proprietário do recurso e pode utilizar fluxos de autorização próprios ou utilizar o servidor de autorização como fornecedor.

Na segunda etapa, o proprietário do recurso informa quais os recursos que a aplicação *web* terá acesso, e esta solicita uma chave de acesso a estes recursos na terceira fase.

O servidor de autorização, na quarta fase, autentica o cliente e valida suas permissões. Se estiverem válidas, encaminha a chave gerada para a aplicação *web*.

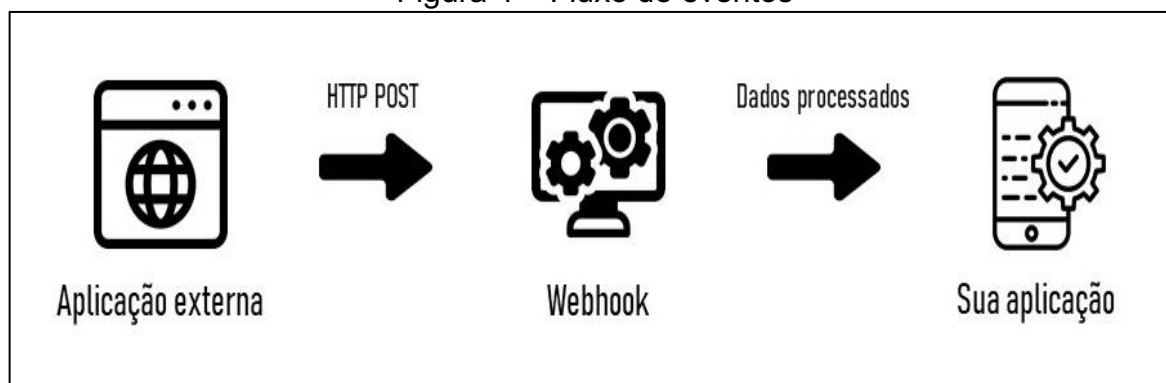
Na quinta fase do fluxo, a aplicação *web* solicita o recurso em nome do usuário, utilizando-se da chave de acesso. O servidor do recurso valida as

informações enviadas pela aplicação *web* e retorna os recursos correspondentes (HARDT, 2012).

3.4 Webhooks

Os *webhooks* são retornos de chamadas HTTP via POST criados para expandir comportamentos de um aplicativo ou página *web*. Aplicações que suportam *webhooks* permitem que usuários especifiquem uma URL (*Uniform Resource Locator*), ou seja, o caminho para o *webhook*, cujo as informações serão enviadas quando determinados eventos forem disparados. Em geral, o caminho especificado pelo usuário é um aplicativo de terceiro, ou uma URL do seu próprio servidor *web* ou aplicativo com a função de manipular dados capturados pelo retorno (SENDGRID, 2014). A Figura 4 exemplifica o fluxo de eventos utilizado pelos *webhooks*.

Figura 4 – Fluxo de eventos



Fonte: Autoria própria

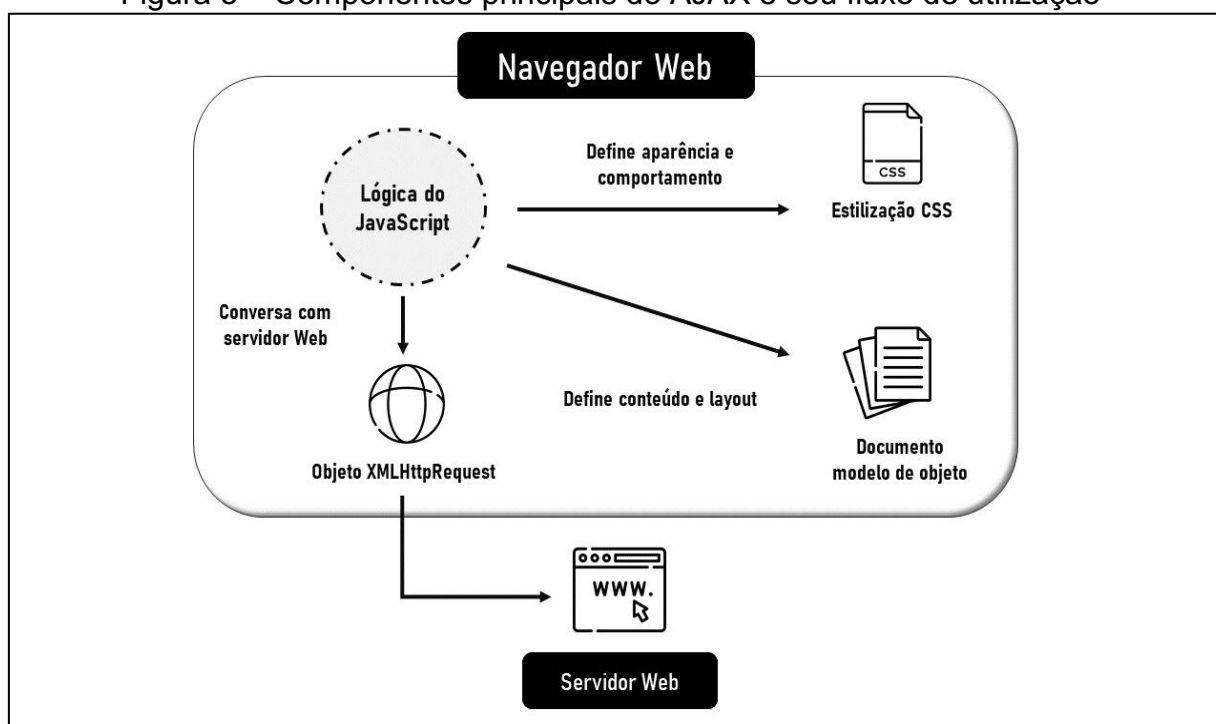
Usuários especificam uma URL na aplicação externa e informam para quais eventos desejam receber informações, como por exemplo, comentários em publicações de blogs, novos pagamentos, e-mails recebidos, usuários conectados e atualizações de informações. Os eventos são disparados assim que determinadas ações ocorrem, e são enviados com o recurso para a URL informada (SENDGRID, 2014).

Segundo Sendgrid (2014), as URL's configuradas em *webhooks* são abertas ao acesso externo e, com isso, podem ser disparadas por qualquer aplicativo que as possuem. As aplicações podem configurar chaves de acesso para evitar problemas de disparos desconhecidos.

3.5 AJAX

O acrônimo AJAX significa Asynchronous JavaScript and XML, e é um padrão de tecnologia que inclui HTML, JavaScript e um objeto XMLHttpRequest. Com o AJAX é possível realizar mudanças na interface web sem atualizar a página do navegador. O X em AJAX vem do XML (*Extensible Markup Language*), padrão de troca de dados muito utilizado na época da criação do AJAX. Embora hoje em dia ainda se faça uso do XML, o uso do JSON (*JavaScript Object Notation*) se tornou a principal forma de empacotar os dados utilizados em requisições AJAX, isto porque o modelo JSON é leve e parecido com objetos JavaScript (MOZILLA, 2021c). A Figura 5 ilustra os componentes principais do AJAX e seu fluxo de utilização.

Figura 5 – Componentes principais do AJAX e seu fluxo de utilização



Fonte: Adaptado de Crane *et al.* (2007)

A Figura 6 ilustra o fluxo de interações entre o cliente (navegador do usuário) e o servidor web. A lógica do aplicativo é formada pelo JavaScript que conecta as diferentes partes da interação. Na primeira etapa, o código JavaScript solicita ao objeto XMLHttpRequest novos dados e solicitações dos usuários ao servidor. O servidor busca os dados utilizando lógica de negócios e acesso ao banco de dados e retorna os registros requisitados. A lógica do JavaScript altera o modelo de dados

apresentado aos usuários, e com as novas informações recebidas pelo servidor é possível alterar o visual da tela, como tabelas, textos, imagens, além de definir comportamentos e estilizações com o CSS⁷ (CRANE *et al.*, 2007).

Todas as alterações de interface podem ser realizadas recarregando a página inteira, contudo, isso pode ser custoso para o servidor, pois é necessário receber todos os arquivos novamente. Com AJAX não é necessário realizar alterações em conteúdos estáticos em sites, como por exemplo, menus e rodapés. Ao clicar em acessar uma nova entidade, somente o conteúdo específico desse recurso precisa ser alterado, mantendo o restante da página inalterada (CRANE *et al.*, 2007).

⁷ O CSS (*Cascading Style Sheets*) é um formato simples para adicionar estilos a páginas *web*, sendo possível acrescentar fontes, cores, alinhamentos, margens (W3C, 2021b).

4 UNIFIED MODELING LANGUAGE

De acordo com Guedes (2014), a Linguagem de Modelagem Unificada, do inglês *Unified Modeling Language* (UML), é uma linguagem de modelagem de sistemas. Os diagramas e fluxos visuais da UML, podem apoiar engenheiros de softwares para estabelecer propriedades do sistema, como por exemplo, requisitos, qualidades, organizações, processos, exigências físicas como equipamentos, e necessidades de uso de outros sistemas como banco de dados.

A UML é sustentada por diversos diagramas, cada um com suas perspectivas. Conforme a análise e modelagem do sistema é possível descobrir a complexidade da aplicação, além disso, os diagramas se complementam. Determinados diagramas destacam o sistema como um todo, outros são usados para objetivos específicos (GUEDES, 2014).

Segundo Guedes (2014), a diagramação do sistema permite observar falhas em diversos pontos do sistema, o que resulta em diminuição de problemas encontrados na fase de desenvolvimento do projeto. Cada diagrama possui sua finalidade, porém não é necessário modelar todos, pois alguns servem para objetivos específicos, muitas vezes não requisitado no sistema projetado.

A seguir são listados alguns exemplos de diagramas da UML com suas descrições (GUEDES, 2014):

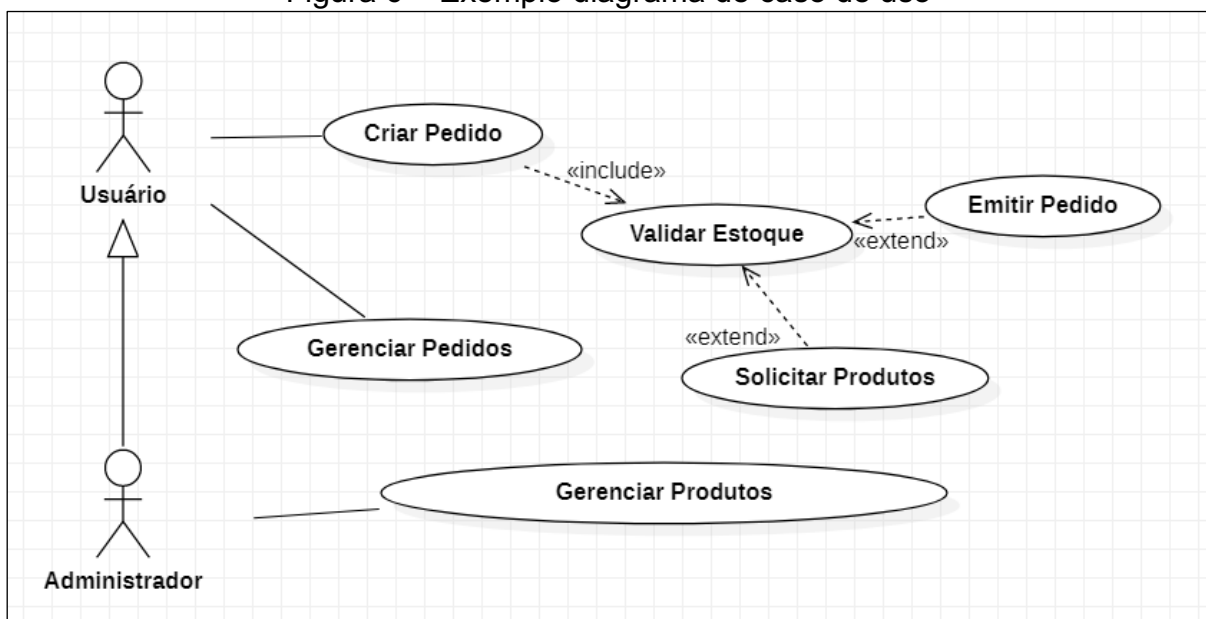
- Diagrama de classes: utilizado para estabelecer a organização de classes do sistema, com seus atributos e métodos.
- Diagrama de caso de uso: adequado para o levantamento de requisitos iniciais do projeto e ilustra as funcionalidades que o usuário necessita.
- Diagrama de objetos: geralmente empregado como complemento ao diagrama de classes, sendo utilizado para mostrar em tempo de execução os valores dos atributos dos objetos de uma classe.
- Diagrama de sequência: usado para ilustrar a sequência de troca de mensagens entre os objetos do processo em questão.
- Diagrama de atividades: adequado para ilustrar o fluxo de atividades de uma determinada ação no sistema, todas as etapas para a sua conclusão são abordadas neste diagrama.

Para este projeto optou-se por utilizar os diagramas de casos de uso e atividades, que serão abordados com maior ênfase nas Seções 4.1 e 4.2, respectivamente.

4.1 Diagrama de casos de uso

O diagrama de casos de uso é uma maneira de compreender os requisitos funcionais do projeto e demonstrar interações entre sistemas e usuários. Para simbolizar um caso de uso é utilizada a figura de uma elipse. Processos e usuários que interagem de alguma forma com o sistema têm o seu papel definido como Ator. Os atores podem ser usuários, clientes, funcionários, diretores, gerentes, administradores, ou até mesmo outros sistemas. Os papéis em um caso de uso podem ser realizados por diversos atores, e um ator pode realizar diversas funcionalidades (GUEDES, 2014). A Figura 6 apresenta um exemplo de diagrama de caso de uso.

Figura 6 – Exemplo diagrama de caso de uso



Fonte: Adaptado de Guedes (2014)

Os atores são exemplificados no diagrama (Figura 6), como usuário e administrador, possuem um relacionamento de generalização, ou seja, o administrador possui todas as características que o usuário dispõe, já o usuário, não

pode acessar os casos de uso do administrador, no exemplo, o caso de uso Gerenciar Produtos (GUEDES, 2014).

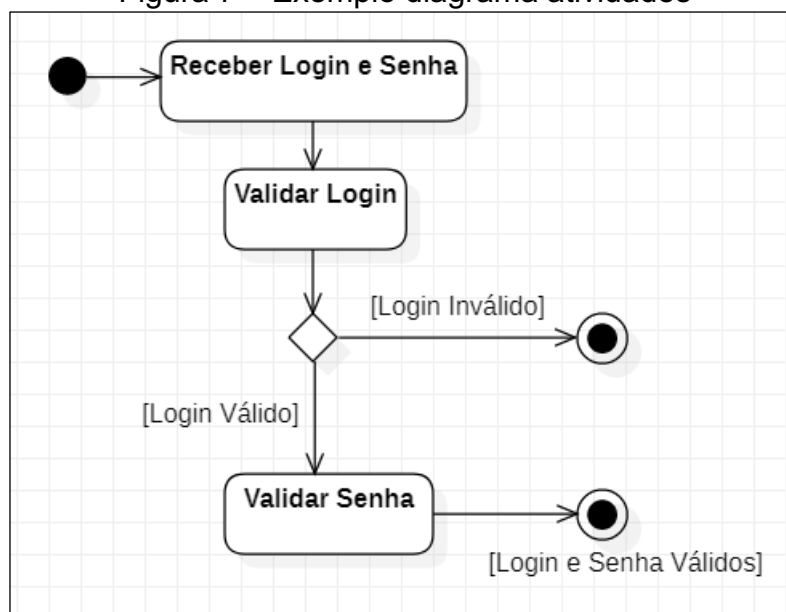
O usuário inicia o caso de uso “Criar Pedido”, este possui o relacionamento de inclusão (*include*) com o caso de uso Validar Estoque. De acordo com Guedes (2014), isso indica que essa sequência é obrigatória, ou seja, ao “Criar um Pedido”, obrigatoriamente o segundo caso de uso será executado, validando se possui estoque dos produtos do pedido. O caso de uso “Validar Estoque” possui o relacionamento de extensão (*extend*) com os casos de usos “Emitir Pedido” e “Solicitar Produtos”. Segundo Guedes (2014), essa sequência é opcional, ou seja, o caso de uso “Emitir Pedido” será executado somente se o produto estiver com estoque disponível. Já o caso de uso “Solicitar Produtos” é executado quando o estoque estiver em falta, solicitando ao responsável a reposição do produto.

4.2 Diagrama de atividade

Segundo Guedes (2014), o diagrama de atividade é utilizado para demonstrar detalhes a nível de operação, além disso, possui similitudes com fluxogramas de controle algoritmo.

O nó inicial do diagrama é representado por um círculo preenchido, sendo a partir dele que o fluxo de atividades se inicia. Os nós de ações são os elementos principais do diagrama e são exemplificados com um retângulo arredondado, e demonstram ações como fase, etapas e escolhas que precisam ser realizadas. O controle de fluxo do nó é realizado por uma linha com uma seta direcionada ao nó que irá prosseguir o caminho. Para determinar etapas alternativas entre as atividades é utilizado um nó de decisão com as condições de escolha. Ao final do diagrama é colocado o nó final, este é representado por um círculo preenchido em seu interior com um círculo vazio (GUEDES, 2014). A Figura 7 ilustra um exemplo de diagrama de atividades.

Figura 7 – Exemplo diagrama atividades

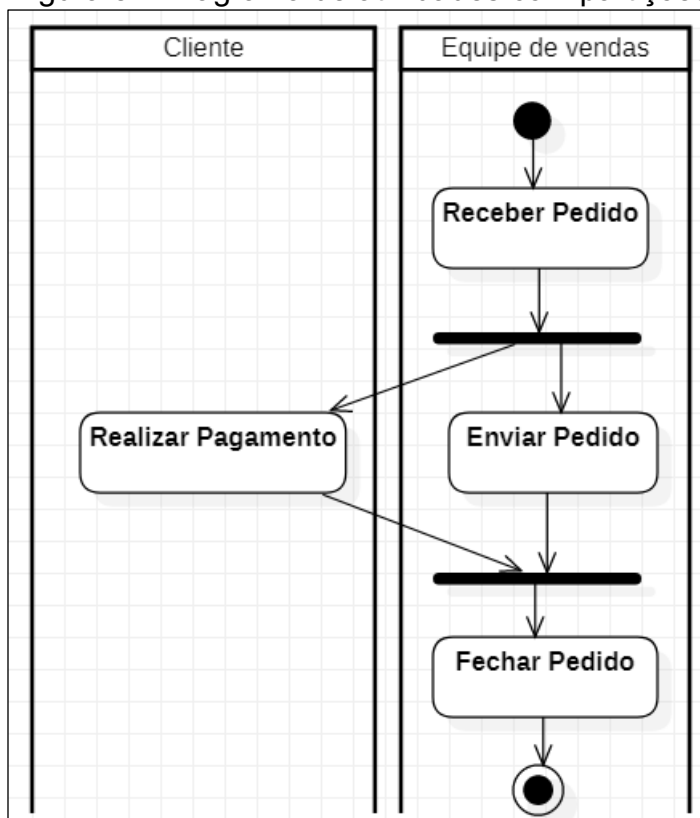


Fonte: Adaptado de Guedes (2014)

Na Figura 7, o fluxo se inicia com o círculo preenchido, e a primeira atividade a ser executada é a “Receber Login e Senha”. Após a execução desta atividade é preciso “Validar Login”, com isso é apresentado a estrutura condicional, e se for inválido o usuário é levado ao fim do fluxo, representado pelo círculo semipreenchido, ou se for válido o fluxo continua com a atividade de Validar Senha (GUEDES, 2014).

Quando é preciso mesclar fluxos dos nós de atividades ou até mesmo separar em dois ou mais caminhos distintos, pode se usar um nó de bifurcação ou união, ambos representados por uma barra. Além disso, para separar os fluxos de atividades que passam por diferentes domínios do sistema, é utilizado um retângulo com sua identificação e este abrange todas as atividades pertinentes ao domínio abordado (GUEDES, 2014). A Figura 8 representa o nó de bifurcação e união, junto a partição de atividades.

Figura 8 – Diagrama de atividades com partições



Fonte: Adaptado de Guedes (2014)

A equipe de vendas inicia o fluxo de atividades recebendo o pedido. O cliente precisa realizar o pagamento do pedido, e em paralelo o pedido é enviado ao fechamento, aguardando o pagamento do pedido, e após a conclusão das duas atividades em concorrência, o pedido é fechado (GUEDES, 2014).

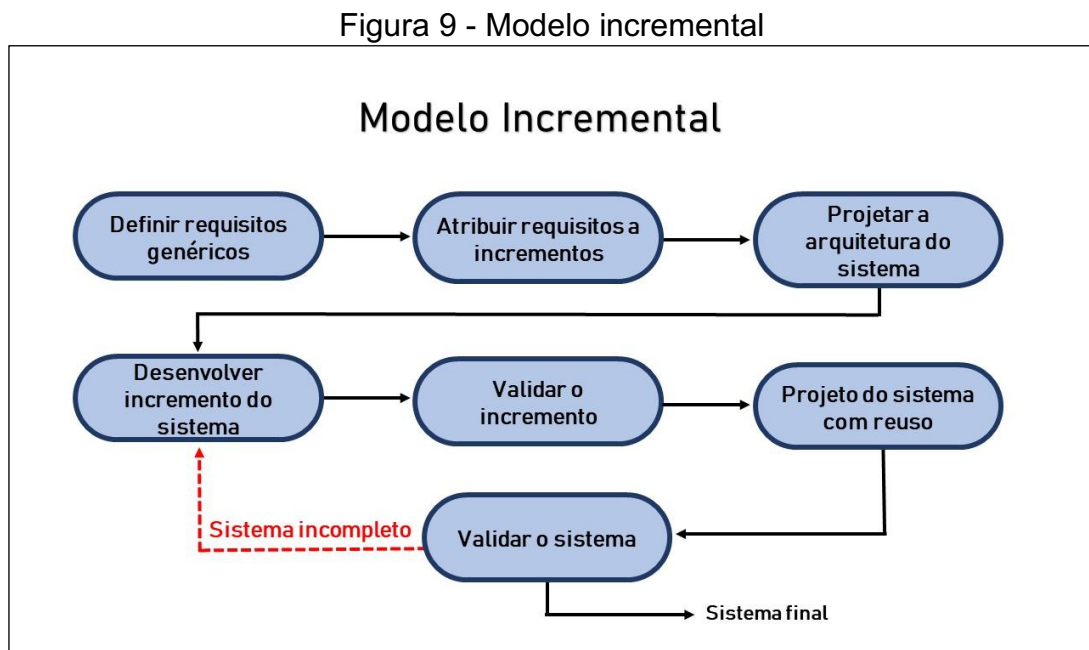
5 DESENVOLVIMENTO INCREMENTAL

Quando não se possui os requisitos principais e iniciais do software a ser desenvolvido, a utilização de um processo produtivo totalmente linear não é recomendada, sendo assim, é necessário o rápido fornecimento do conjunto inicial de funcionalidades aos usuários e, após os primeiros usos, utilizando-se do *feedback* dos mesmos, pode-se realizar melhorias em versões posteriores (PRESSMAN, 2011).

Segundo Pressman (2011), o modelo incremental utiliza-se de fluxos lineares e paralelos, isto porque, ao estar desenvolvendo determinado incremento, pode-se iniciar outro. Com isso, o usuário poderá testar as novas funcionalidades em um tempo reduzido.

No modelo incremental a primeira entrega é um software funcional, o qual será o modelo para as próximas entregas. Após os testes do incremento pelo cliente, os requisitos para o próximo incremento são definidos, iniciando assim o próximo ciclo de desenvolvimento.

Cada incremento deve possuir os requisitos para entregar uma versão funcional ao usuário, e além disso, os incrementos são planejados considerando as modificações e eventuais problemas com o sistema desenvolvido (PRESSMAN, 2011). A figura 9 ilustra o desenvolvimento incremental.



Fonte: Adaptado de Sommerville (2011)

Segundo Sommerville (2011), o modelo incremental possui 8 fases, como ilustrado na Figura 9. Na definição de requisitos, fase 1, são estabelecidos os requisitos iniciais e finais do projeto. O projetista do sistema consulta o cliente e define o escopo da aplicação e sua viabilidade. Esta fase é considerada parte essencial do sistema, pois erros na definição dos requisitos podem gerar problemas na construção e implementação do software.

Na segunda fase, atribuir requisitos aos incrementos, os requisitos genéricos determinados na fase 1 são separados. Cada incremento possui seus próprios requisitos, e nesta etapa são definidas as funcionalidades que cada entrega irá fornecer, além de determinar as prioridades do desenvolvimento.

A terceira fase, projetar a arquitetura do sistema, é utilizada para definir as tecnologias, plataformas e o funcionamento do sistema. É preciso determinar os seguintes requisitos críticos: (i) desempenho; (ii) proteção; (iii) segurança; (iv) disponibilidade; e (v) manutenção.

A definição da arquitetura do sistema é complexa, pois é preciso conciliar com coesão os requisitos funcionais e não funcionais⁸ do projeto.

A quarta fase, desenvolver incrementos do sistema, é a fase intermediária do projeto e nela são desenvolvidos tanto a primeira versão do sistema quanto as demais. A primeira entrega do sistema ao cliente precisa conter um sistema funcional, mesmo com poucos recursos, e ele precisa ser utilizável.

Após a equipe de desenvolvimento concluir a quarta fase, as próximas etapas, 5, 6 e 7, validar incrementos, projeto do sistema com reuso e validar o sistema, respectivamente, são executadas sequencialmente. O incremento é validado conforme os requisitos e suas funcionalidades e passa a integrar o sistema principal, qualquer divergência fará o incremento retornar à quarta fase. A validação do sistema é feita por um grupo de testes e usuários a fim de encontrar possíveis problemas com o sistema após a atualização.

Se o sistema estiver concluído, a iteração passa para a última fase, chamada sistema final, aqui é finalizado o contrato de desenvolvimento, e todas as necessidades estão satisfeitas pelo cliente. A partir desta etapa o cliente pode

⁸ Segundo Pressman (2011), requisitos funcionais descrevem a forma que o sistema deve interagir e reagir para concluir os objetivos propostos. Os requisitos não funcionais descrevem a qualidade e as características não essenciais para o funcionamento do sistema.

requisitar novas funcionalidades e, com isso, a quarta fase é iniciada e um novo contrato de desenvolvimento é elaborado.

A utilização da entrega incremental possui as seguintes vantagens (SOMMERVILLE, 2011; PRESSMAN, 2011):

- Os clientes utilizam a primeira entrega para avaliar o sistema e determinar pontos-chave para as próximas versões.
- A primeira entrega possui todos os requisitos necessários para deixar o sistema funcional de maneira imediata;
- Como o sistema é desenvolvido em partes, e cada incremento não depende necessariamente do seu antecessor, as chances de clientes encontrarem falhas significativas são reduzidas.
- O sistema inicial não depende necessariamente de um grande grupo de desenvolvimento e, após as primeiras avaliações do sistema, mais colaboradores podem ser adicionados ao projeto.

Por sua vez, as desvantagens na utilização de um modelo incremental são (SOMMERVILLE, 2011; PRESSMAN, 2011):

- A primeira entrega pode depender que vários módulos sejam elaborados, mesmo que pequenos recursos sejam desenvolvidos, a arquitetura inicial pode ser complexa.
- O modelo incremental não é recomendado para criação de sistemas a partir de softwares legados, isto porque o cliente pode esperar que todas as funcionalidades do sistema antigo estejam implementadas na primeira entrega.
- Alguns clientes podem determinar datas difíceis de serem cumpridas.
- Não se pode especificar totalmente o contrato de desenvolvimento do sistema, isto porque mudanças podem ocorrer a cada entrega, e com isso, empresas burocráticas podem não aderir a esse tipo de negócio.

6 VTIGER CRM

O Vtiger CRM surgiu de uma ramificação de outro projeto de CRM (*Customer Relationship Management*) chamado Sugar CRM. O Sugar foi lançado sob licença modificada do Mozilla Public License, o que tornava seu código fonte livre. Sridhar Vembu, Diretor executivo da AdventNet, em 2004 criou o Vtiger. O código fonte desenvolvido teve como premissa ser também de código aberto. A versão 5 do Vtiger se desfez da maior parte do código desenvolvido para o Sugar CRM, tornando-se um projeto independente (ROSSI, 2011).

O Vtiger CRM possui diversas funcionalidades e entidades que são disponibilizadas por padrão e gratuitamente no modelo de código aberto. A seguir são listadas algumas funcionalidades: (VTIGER, 2021a, VTIGER, 2021b):

- Gerenciamento de contatos: armazenamento de dados de clientes, como por exemplo nome, e-mail, telefone, registros, endereço, além de ser possível a criação de novos campos para uso específico.
- Gerenciamento de oportunidades: histórico completo das oportunidades de negócios da empresa, como a lista de todos os eventos ocorridos com elas como ligações, reuniões, orçamentos.
- Gestão de chamados e suporte aos usuários: controle o status do chamado, como tempo para resolução do problema, a severidade do caso, qual é sua prioridade e por fim associar a um contato.
- Gestão de inventário: organização de seus produtos e serviços com preços e impostos.
- Calendário: gestão de sua lista de reuniões, sendo possível criar encontros com clientes em potencial.
- Gestão de cotações: criação de orçamentos com informações de preços, descrições, formas de pagamento, produtos e serviços. O módulo de cotações permite criar e personalizar informações sobre a cotação, além de ser possível exportar em tabelas CSV (*Comma-separated values*) e enviá-los por e-mail aos clientes interessados.
- Sequências de e-mails: programação e disparo de e-mails predefinidos para clientes.

- Gestão de usuários: hierarquia de permissões, informações de perfil, configurações de hora e data, além do gerenciamento de senhas.

6.1 Módulo de *leads*

O gerenciamento de *leads* é o primeiro passo de vendas no CRM. O *lead* possui dados e informações sobre pessoas e organizações a qual ela representa. Um *lead* pode ser capturado por folhetos de marketing, sites, cartões de visita, e-mails, indicações de outros *leads*. A equipe de vendas pode ordenar, selecionar e classificar as melhores oportunidades. Existem dois modelos de *lead*: o primeiro, indica uma organização, e o segundo, representa uma pessoa (VTIGER, 2021b).

O Vtiger CRM possibilita diversas formas para a criação de *leads*, uma delas é a importação de registros via tabelas CSV. Outro método, é a captura por meio do site da empresa, através dos formulários de dados, ou seja, ao cadastrar suas informações no formulário o registro é armazenado no CRM como um *lead*. Além disso, é possível criar *leads* manualmente com todas as informações que julgar necessário, embora, apenas o sobrenome tem o preenchimento obrigatório (VTIGER, 2021b). A Figura 10 ilustra o módulo de *leads* na visão de criação do registro.

Figura 10 – Módulo de *leads* visão de criação do registro

vtiger

LEADS > All > Adicionando novo

Criando Novo(a) Lead

Detalhes Lead

Nome	Nada	Sobrenome *	
Telefone		Empresa	
Celular		Título	
Fax		Fonte Lead	Selecionar uma Opção
E-mail		Atividade	Selecionar uma Opção
Website		Receita Anual	R\$
Status Lead	Selecionar uma Opção	No. Empregados	
Avaliação	Selecionar uma Opção	E-mail Alternativo	
Responsável *	Leonardo Administrator	Recusa Email	<input type="checkbox"/>

Fonte: Autoria própria (2021)

Os campos obrigatórios são sobrenome e responsável. O responsável por padrão, é relacionado com o primeiro usuário (Administrador) do sistema. Já o

sobrenome necessita ser preenchido para a criação do registro. A seguir é abordado sobre alguns campos do módulo de *leads*.

- O campo **Fonte Lead** é usado para determinar qual método foi utilizado para a captura do registro, como exemplo, boca a boca, relações públicas, conferências, parceiros e diversos outros, além de ser possível customizar essas opções.
- O campo **Atividade** é utilizado para manter a área de atuação do *lead*, sendo possível escolher diversas opções, como por exemplo, saúde, governo, banco, transportes, engenharia, alimentação, entre outras.
- O campo **Status Lead** é aplicado para determinar qual é o status do *lead*, as opções pode ser tentativa de contato, contatar no futuro, não contatado, descartado, se ele está quente, morno ou frio. Todas as opções determinam o quão perto ou longe a equipe de vendas está de fechar o negócio com este *lead*.
- O campo **Avaliação** determina qual é a relação do *lead* diante do CRM, ou seja, se ele está ativo, foi perdido, teve seu projeto cancelado ou adquirido, teve o relacionamento encerrado, essas opções são usadas pela equipe de vendas para filtrar os *leads* que estão disponíveis no CRM.

6.2 Tecnologias utilizadas pelo Vtiger

Segundo Rossi (2011), as tecnologias que o Vtiger CRM possui em sua construção são:

- Servidor *web* Apache.
- Linguagem de programação PHP (Hypertext Preprocessor).
- Banco de dados MySQL.
- Mecanismo de modelos *Smarty*.

As tecnologias são abordadas de forma detalhada nas subseções a seguir. A subseção 6.2.1 descreve sobre o servidor *web* Apache. A subseção 6.2.2 aborda sobre a linguagem de programação PHP. A subseção 6.2.3 detalha o banco de dados MySQL. E por fim, a subseção 6.2.4 discorre sobre o mecanismo de modelos *Smarty*.

6.2.1 Apache

Apache HTTP (*Hypertext Transfer Protocol*) *Server Project* é um projeto de servidor *web* gratuito desenvolvido com o apoio de desenvolvedores de código aberto, com a premissa de implementar uma aplicação de servidor poderosa, com diversos recursos a nível de servidores comerciais. Os desenvolvedores que apoiam o projeto, localizados por todo o mundo, utilizando-se da internet podem trocar ideias, organizar e desenvolver o servidor e sua documentação associada (APACHE, 2021).

Segundo Rossi (2011), o Apache permite que o desenvolvedor hospede um site na Internet, como por exemplo, o próprio Vtiger CRM. Apesar do Vtiger CRM utilizar o servidor Apache, é possível configurar outros servidores.

6.2.2 PHP

A linguagem de programação PHP foi criada no outono de 1994 por Rasmus Lerdorf e utiliza scripts escritos em linguagem C para criar páginas dinâmicas na internet. Em 1995, o código-fonte foi disponibilizado a desenvolvedores que gostariam de manter o projeto. Em 1998, o PHP passou por mudanças, e uma delas foi a introdução ao suporte a orientação a objetos. Ainda em 1998, a linguagem possibilitou a criação de novos módulos, atraindo novos desenvolvedores, sendo que, no final do mesmo ano, já estava sendo utilizada em 10% dos domínios da internet (DALL'OGGIO, 2017).

A versão 5 do PHP necessitou de um longo período para ser concluída, pois era preciso implementar sólidos conceitos de orientação a objetos, e teve sua versão oficial lançada em julho de 2004 e pode ser utilizada em três grandes áreas de desenvolvimento (PHP, 2021):

1. Em aplicações de Internet, sendo utilizado como servidor web: esta é a área que torna o PHP uma das linguagens mais relevante no desenvolvimento de aplicações cliente-servidor;
2. *Scripts* de comandos para Windows e Linux: utilizado na criação de rotinas para processamento de diversas tarefas de propósito geral;
3. Criação de aplicações *desktop*: por meio da linguagem PHP é possível escrever interfaces gráficas para aplicativos *desktops*.

Além de possuir liberdade de escolha entre os diversos sistemas operacionais disponíveis, uma das características principais da linguagem PHP é sua ampla gama de suporte a banco de dados. É possível se conectar a um banco de dados utilizando as extensões criadas pelos desenvolvedores para bancos específicos, ou ainda utilizar um banco com suporte a interface padronizada de conexão ODBC (*Open Database Connection*) (PHP, 2021).

6.2.3 MySQL

O MySQL, de acordo com MySQL (2021), é um sistema gerenciador de banco de dados SQL (*Structured Query Language*). O MySQL é de código aberto e um dos mais populares gerenciadores do mercado.

Um banco de dados é uma estrutura de dados, que pode representar uma simples lista de produtos, ou até mesmo grandes quantidades de dados. Para gerenciar as informações armazenadas nessas estruturas, como por exemplo, ler, cadastrar e tratar os dados armazenados, é necessário um gerenciador de banco de dados, como o próprio MySQL (MYSQL, 2021).

6.2.4 Smarty

Embora a linguagem PHP tivesse obtido grande aceitação por parte dos desenvolvedores, a separação de seus códigos HTML (*HyperText Markup Language*), ou seja, o isolamento do conteúdo de apresentação, ainda era uma questão a ser resolvida (SMARTY, 2021).

Segundo Smarty (2021), o mecanismo *Smarty* surgiu para solucionar problemas de separação de código. Os responsáveis pela interface do sistema traçam o conteúdo de cada página, construindo botões, campos, formulários, imagens e textos. Os programadores do sistema realizam a codificação dos registros, com foco na lógica de negócios. O conteúdo de uma página é passado via modelos de dados, e esses, por sua vez, são especificados com acordos entre programadores e *designers*.

Os arquivos criados utilizando *Smarty* podem conter lógicas, desde que essas funcionalidades utilizem códigos para melhorar a apresentação de conteúdo. Um exemplo, é a exibição de informações oriundas do banco de dados. Pode-se realizar a listagem das categorias de maneira dinâmica, utilizando estruturas de repetição (SMARTY, 2021).

As variáveis PHP utilizadas no HTML são encapsuladas por chaves. Ao mostrar a página *web* as variáveis são alteradas para corresponder aos seus valores reais. Os dados são fornecidos pelo programador, e se o *designer* julgar necessário mais opções de dados, apenas informa ao programador para criar ou atualizar novas variáveis (SMARTY, 2021).

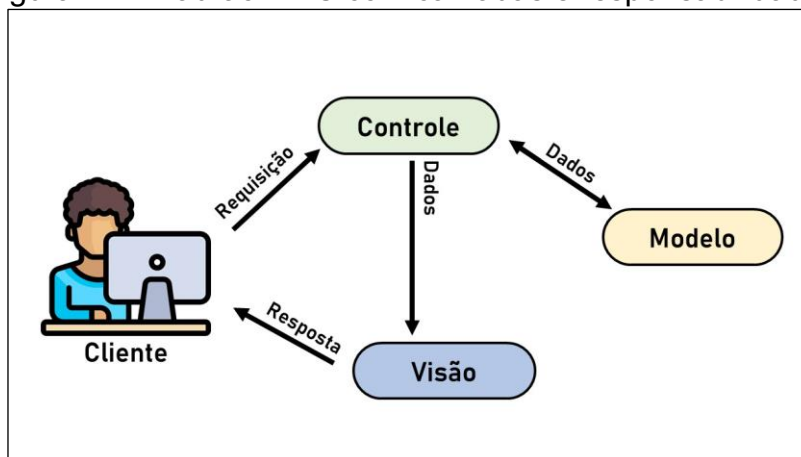
7 PADRÃO DE SOFTWARE MVC

Segundo Dall'Oglio (2014), o MVC, do inglês *Model View Controller*, é um padrão de software amplamente conhecido e utilizado no desenvolvimento de sistemas. O objetivo do MVC é separar o sistema em camadas, em que cada componente tem sua própria responsabilidade. O padrão possui três tipos de responsabilidade:

- Modelo (*model*): A camada de modelo é utilizada para representar métodos de negócio do sistema. Consultas aos dados armazenados, realizações de cálculos e validações de informações, são feitas nesta camada.
- Visão (*view*): A camada de visão tem como responsabilidade a entrada e saída de dados, ou seja, mostrar informações em tela e obtém dados do usuário. Outro objetivo desta classe é organizar os dados na tela. A visão não deve conter lógica de programação.
- Controle (*controller*): Esta camada é utilizada para controlar ações do usuário. O controle é responsável por chamar as funções correspondentes às solicitações de uma visão. Como exemplo de responsabilidade, tem-se uma entrada de dados de um formulário recebidas pela visão, estas precisam ser encaminhadas ao modelo de negócio que irá realizar toda a lógica com o registro.

A Figura 11 ilustra o padrão de software MVC com suas camadas e responsabilidades.

Figura 11 – Padrão MVC com camadas e responsabilidades



Fonte: Adaptado de Dall'Oglio (2014)

O cliente envia uma solicitação por meio da visão, como por exemplo, solicitar uma página do sistema, o controle recebe o pedido e encaminha para o modelo de negócio que irá buscar os dados referente a página. Aqui o modelo pode consultar no banco de dados, ou realizar outros tipos de lógica de negócio, como validações e tratamento de dados. O modelo retorna todos os dados necessários à camada de controle que encaminha para a visão correspondente. No fim, a visão mostra a página solicitada ao usuário (DALL'OGGIO, 2014).

8 DESENVOLVIMENTO

Este capítulo aborda o desenvolvimento do projeto do Trabalho de Conclusão de Curso. A Seção 8.1 descreve a Graph API. A Seção 8.2 define os requisitos genéricos. A Seção 8.3 atribui os requisitos ao projeto. A Seção 8.4 detalha a arquitetura do sistema. A Seção 8.5 aborda o desenvolvimento do projeto. A Seção 8.6 realiza os testes e a validação dos incrementos desenvolvidos. Na Seção 8.7 é realizado a integração dos arquivos desenvolvidos com o sistema. A Seção 8.8 descreve sobre a conclusão do desenvolvimento do projeto.

8.1 Graph API

O acesso à leitura e escrita de dados no Facebook e Instagram são feitos pelo Graph API (2021). Informações como amigos, fotos, comentários, curtidas em publicações, entre outros, podem ser retornadas em requisições na API (*Application Programming Interface*).

Todos os registros disponíveis no Graph API são acessíveis por meio de sua URI (*Uniform Resource Identifier*), além disso, é necessário realizar a autenticação e autorização OAuth2 para realizar as consultas.

Para facilitar as consultas e escritas em registros, o Graph API na versão 9.0 disponibiliza, para as diversas linguagens de programação, um SDK⁹ correspondente. O SDK de suporte à linguagem PHP (*Hypertext Preprocessor*), utilizada neste projeto, é disponibilizado na plataforma do GITHUB¹⁰ e pode ser acessado por qualquer desenvolvedor gratuitamente.

Para o propósito do projeto, é necessário consultar diversos registros na API. A seguir são apresentados exemplos dos métodos HTTP (*Hypertext Transfer Protocol*), URI, e a descrição das informações, respectivamente:

- **GET/me/accounts**: retorna todas as páginas administradas pelo usuário.
- **POST/123/subscribed_apps**: inscreve a página de identificador 123 para receber atualizações em campos informados no corpo da requisição, por

⁹ O SDK (*Software Development Kit*) é um conjunto de ferramentas, geralmente disponibilizado pelo fornecedor da plataforma, que auxilie o desenvolvimento de aplicações e recursos (REDHAT, 2021).

¹⁰ Site do SDK PHP: <https://github.com/facebookarchive/php-graph-sdk>

exemplo o *leadgen*, que a cada novo *lead* gerado pelo Facebook e Instagram eles são encaminhados para o *webhook*.

- **DELETE/123/subscribed_apps**: encerra o recebimento de novos *leads* gerados pela página.
- **GET/123/leadgen_forms**: retorna todos os formulários disponíveis na página correspondente.
- **GET/321/?fields=questions,tracking_parameters**: todos os campos do formulário de identificador 321 são retornados, isto é útil, pois é preciso mapear os campos correlacionados no CRM (*Customer Relationship Management*) individualmente.
- **GET/111/?fields=form_id,created_time,field_data**: quando um *webhook* é disparado pela Graph API, em muitos casos, somente o identificador único do registro é enviado, com isso, é preciso consultar todos os campos restantes do *lead* utilizando seu identificador, no exemplo 111.

8.2 Definição dos requisitos genéricos

O modelo incremental não utiliza requisitos fixos para criação de sistemas, e a cada incremento é possível colocar novas funcionalidades. Após analisar a documentação do Graph API e do Vtiger CRM, definiu-se os seguintes requisitos genéricos do sistema:

- Área de configuração do módulo: tela para agrupar todas as configurações do projeto.
- Autenticação e Autorização: Criar um projeto no Facebook que disponibilize as credenciais para usar o padrão OAuth2 para autenticar e autorizar o CRM a consultar dados no Graph API.
- Habilitar o recebimento de *leads*: permitir habilitar páginas do Facebook. Como o Instagram não possui páginas em sua plataforma é preciso que uma conta esteja vinculada a uma página do Facebook, dessa maneira ambas as plataformas encaminharam os *leads* para o CRM.
- Mapeamento de campos: realizar a associação entre campos dos formulários das campanhas de publicidade com o CRM;

- Recebimento de *leads* via *webhook*: criar uma URL capaz de processar os disparos enviados pelo Facebook e Instagram com os dados do *lead* gerado.

8.3 Atribuir requisitos aos incrementos

A segunda fase do modelo incremental sugere dividir os requisitos iniciais em incrementos menores por meio dos seus próprios ciclos de desenvolvimento. O módulo desenvolvido no projeto possui poucos requisitos iniciais e, por isso, optou-se por realizar um único ciclo do desenvolvimento incremental.

Nesta fase do modelo são definidas quais são as prioridades de entrega e quais as funcionalidades que cada incremento deve fornecer. As funcionalidades e a descrição de cada requisito são:

- Área de configuração do módulo: o Vtiger CRM possui diversos módulos por padrão, a maioria deles possui uma área de configuração. Nesta área é possível fazer ajustes, tais como alterações de recursos, textos, interface e funcionalidades do módulo. O módulo de integração do projeto irá se adequar ao padrão do CRM e, com isso, uma área de configuração será criada.
- Autenticação e Autorização: para realizar a conexão com o Graph API é preciso entrar com uma conta Facebook e este acesso será disponibilizado pela área de configuração.
- Habilitar o recebimento de *leads*: será disponibilizada a opção de ativar e desativar o recebimento de *leads* provenientes das páginas do usuário.
- Mapeamento de campos: uma nova tela na área de configuração deve listar todos os formulários disponíveis por página selecionada. Além disso, ao selecionar um formulário, o sistema lista todos os campos que o mesmo possui, sendo possível mapeá-los no CRM.
- Recebimento de *leads* via *webhook*: um gatilho HTTP será criado para o recebimento dos *leads*, este foi projetado para realizar o tratamento adequado de cada registro recebido, e encaminhado para os métodos correspondentes.

8.4 Arquitetura do sistema

Na terceira fase do modelo incremental definiu-se a arquitetura do sistema, neste caso, foi utilizada a própria arquitetura do Vtiger CRM.

A arquitetura do sistema precisa possuir os seguintes atributos: segurança, disponibilidade, desempenho, proteção e manutenção. Esses atributos são disponibilizados pelos próprios recursos do CRM. Os requisitos genéricos do sistema foram modelados pensando na otimização e coesão com os requisitos não funcionais que a arquitetura do Vtiger CRM possui. Além disso, nesta fase foi determinado o uso do padrão de software MVC (*Model View Controller*).

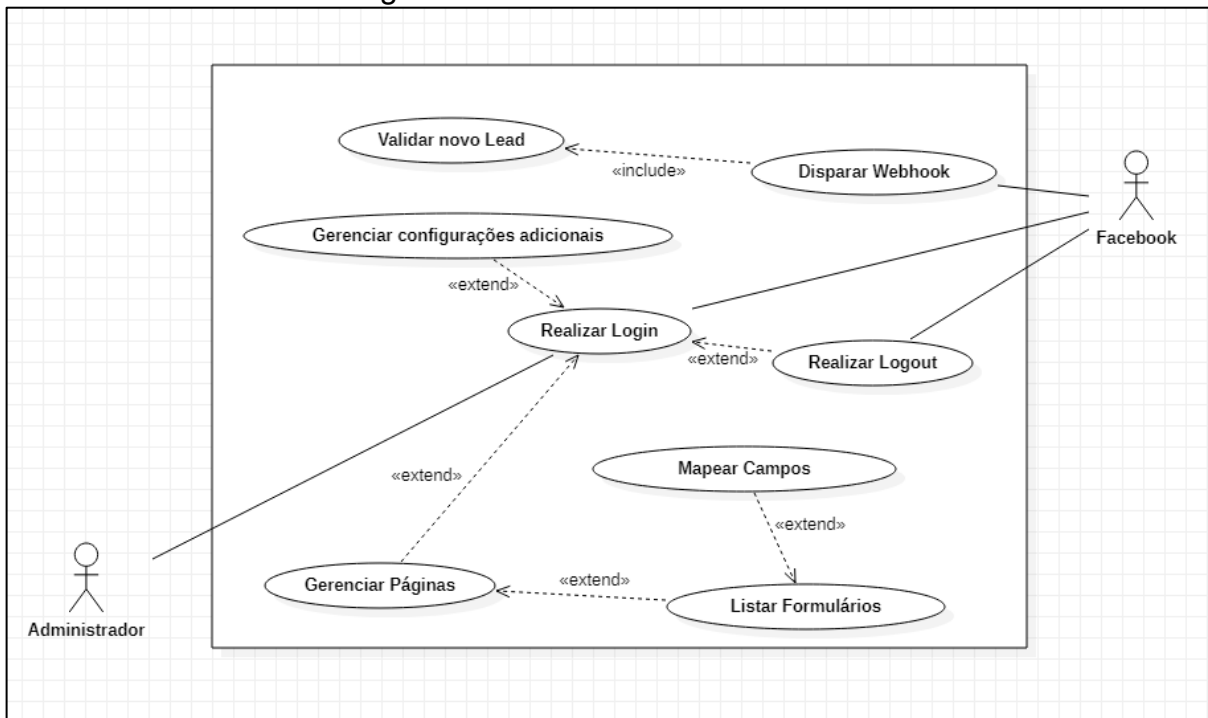
8.4.1 Diagrama de caso de uso

As abstrações do projeto são construídas usando os requisitos iniciais e para representar as interações do usuário com o sistema. Neste caso, utilizou-se um diagrama de caso de uso para ilustrar as funcionalidades e suas associações.

Neste projeto os atores foram denominados como Administrador, Facebook. Figura 12 ilustra o diagrama de caso de uso do módulo, utilizando a ferramenta StarUML¹¹.

¹¹ Site da Ferramenta StarUML: <https://staruml.io/>

Figura 12 – Caso de uso do módulo



Fonte: Autoria própria (2021)

Os casos de uso do projeto são listados e descritos a seguir:

- 1) *Realizar Login*: o administrador precisa efetuar a integração entre o CRM e o *Facebook*, por meio da autenticação da conta.
- 2) *Realizar Logout*: funcionalidade opcional para remover o acesso a sua conta e apagar dados de configurações.
- 3) *Gerenciar Páginas*: realizar o gerenciamento das páginas oriundas do *Facebook*, permitindo a inscrição e a remoção no CRM.
- 4) *Gerenciar configurações adicionais*: permite configurar opções sobre a duplicidade dos dados do *lead* recebido.
- 5) *Listar Formulários*: listar formulários pertencentes as páginas do *Facebook*, com seus respectivos campos de entrada de dados.
- 6) *Mapear Campos*: realizar o mapeamento entre os campos do CRM com os campos de um Formulário.
- 7) *Disparar webhook*: quando um novo *lead* é recebido pela rede social ele é encaminhado para a URL (*Uniform Resource Locator*) do CRM que realiza o primeiro tratamento, verificando a chave do disparo, e se o *lead* possui seu identificador único.

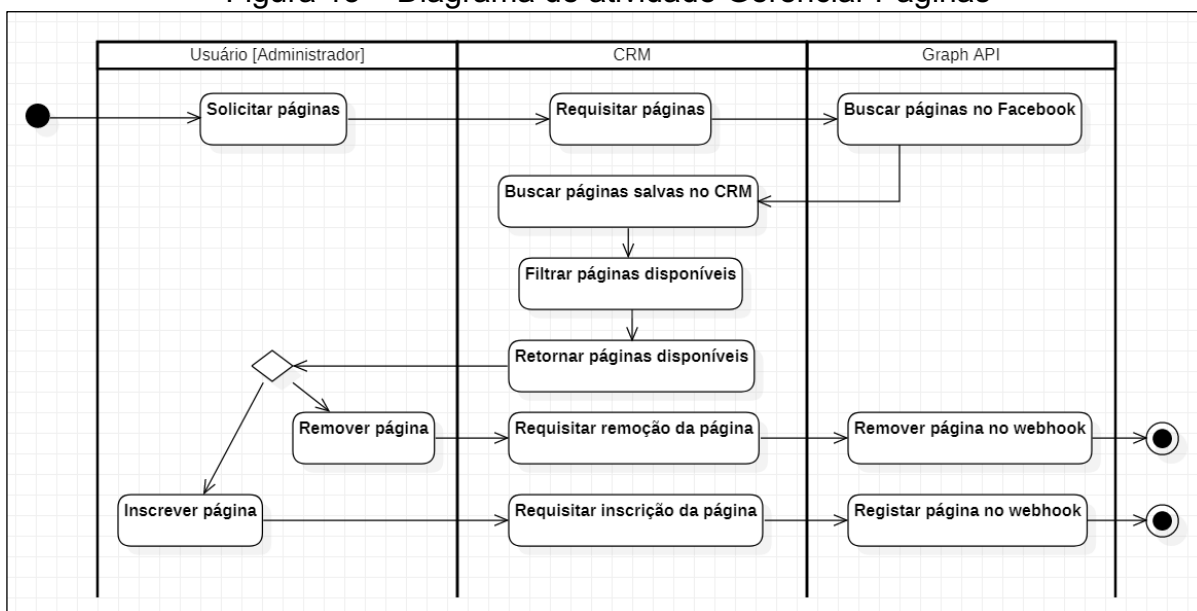
- 8) Validar novo *Lead*: o *lead* recebido pela rede social passa por uma validação de duplicidade antes de ser salvo, se o seu identificador já possuir cadastro no CRM ele é ignorado. Os campos são mapeados entre sistemas, e após tratamento dos dados ele é salvo no sistema.

A documentação detalhada de cada caso de uso encontra-se em sua totalidade no Apêndice A.

8.4.2 Diagrama de atividade

Neste trabalho, o diagrama de atividades foi utilizado para modelar os algoritmos dos processos do projeto. Este trabalho possui diversos casos de uso, porém devido a similitude e a simplicidade de alguns, optou-se por exemplificar em diagramas de atividades apenas os casos de uso “Gerenciar Páginas” e “Validar Novo Lead”. A Figura 13 demonstra o fluxo de atividades para o caso de uso “Gerenciar Páginas”.

Figura 13 – Diagrama de atividade Gerenciar Páginas



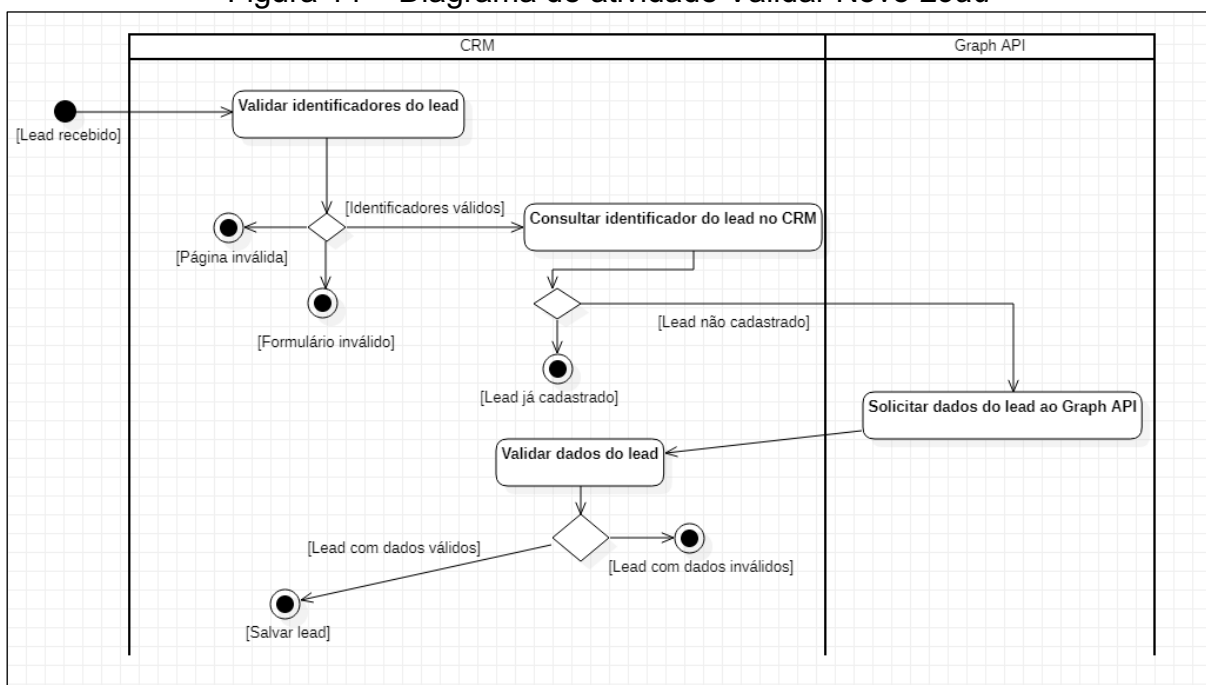
Fonte: Autoria própria (2021)

Ao iniciar o fluxo de atividades, o usuário solicita suas páginas disponíveis. O CRM realiza a requisição das páginas ao Graph API. A API retorna as páginas administradas pelo usuário. Após o retorno, o CRM consulta todas as páginas salvas no sistema, ou seja, as páginas já inscritas anteriormente. Com a lista de páginas de

ambos os sistemas, o CRM compara e filtra as disponíveis, ou seja, se determinada página for excluída no Facebook e ainda estiver salva no CRM ela se torna inválida e é removida do sistema. Em seguida, com a lista de páginas disponíveis, o CRM as retorna ao usuário. Na próxima etapa, o usuário pode inscrever uma página para receber *lead* no *webhook* ou remover uma página previamente inscrita para os recebimentos, ambas as solicitações são encaminhadas ao CRM que requisita ao Graph API a conclusão da tarefa. Por fim, o fluxo de atividades é encerrado.

A Figura 14 ilustra o fluxo de atividades do caso de uso “Validar Novo Lead”.

Figura 14 – Diagrama de atividade Validar Novo Lead



Fonte: Autoria própria

O fluxo de atividades se inicia com o lead recebido. A primeira atividade a ser executada são validações nos identificadores do registro, ou seja, se a página ou o formulário forem inválidos, em outras palavras, não pertencerem ao usuário ou não estiverem configurados no CRM, o fluxo se encerra. Ao continuar, caso os identificadores estejam válidos, é preciso consultar o identificador do lead no sistema. É necessário verificar se já existe cadastro no sistema, e se houver duplicidade, o fluxo termina. Com o identificador do registro é preciso solicitar o restante dos dados ao Graph API, isto porque a API envia somente as informações de identificação do lead ao *webhook* do CRM.

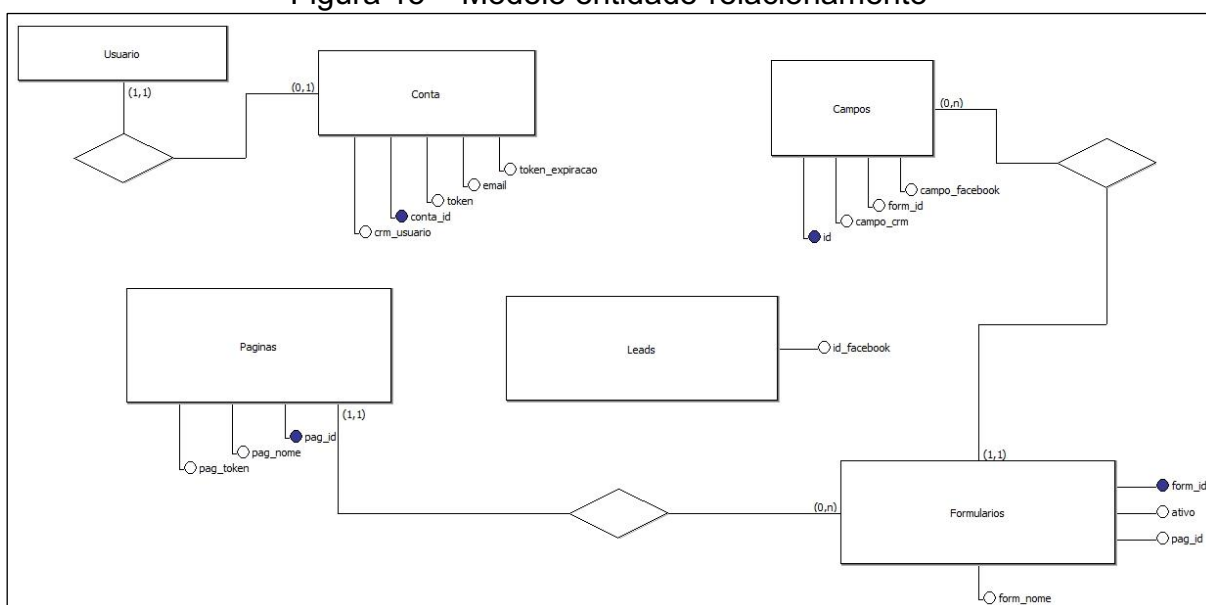
Após ter todos os dados disponíveis é preciso validar e tratar campos tais como datas, em que muitas vezes, o fuso horário pode ser diferente entre os sistemas. Além disso, campos com formatações inválidas tais como e-mails e telefones são ajustados conforme o padrão do CRM e se não for possível realizar tais ajustes o *lead* é ignorado. E, por fim, o *lead* é encaminhado para ser armazenado no sistema.

8.4.3 Modelo entidade-relacionamento

As informações sobre as páginas do usuário e formulários de campanhas de marketing necessitam ser armazenadas corretamente. Para isso desenvolveu-se o modelo relacional do banco de dados.

A Figura 15 ilustra a abstração dos requisitos iniciais em um modelo relacional de banco de dados do módulo do projeto. A ferramenta BrModelo¹² foi utilizada para gerar o modelo do projeto.

Figura 15 – Modelo entidade-relacionamento



Fonte: Autoria própria (2021)

As entidades de “Usuário” e “Leads” são fornecidas pelo Vtiger CRM e reutilizadas na ilustração do modelo. As configurações para utilizar a Graph API são armazenadas na tabela “Conta”, sendo necessário salvar a chave de acesso, sua data

¹² Site da Ferramenta BrModelo: <http://www.sis4.com/brModelo/>

de expiração, e-mail, e o identificador do usuário no CRM. A integração permite somente uma conta Facebook, além disso, o usuário que irá realizar as configurações do módulo deve possuir uma conta administrador em ambos os sistemas.

As páginas utilizadas na integração possuem suas próprias chaves. A tabela “Páginas” é utilizada para armazenamento dos dados de acesso, além do nome e seu respectivo identificador. Páginas do Facebook podem possuir inúmeros formulários de cadastros e, com isso, a tabela de Formulários armazena informações como nome, identificador da página e se o formulário está ativo no sistema.

Os formulários podem possuir diversos campos de entrada de dados, como nome, sobrenome, data de nascimento, e-mail, telefone, entre outras opções customizadas. Com isso, a tabela “Campos” possui a função de correlacionar os campos pertencentes aos formulários com os campos do CRM, ou seja, ao receber um dado no campo nome do formulário, e o CRM irá armazenar esse registro no seu campo corresponde. Com isso, a tabela “Campos” possui a função de correlacionar os campos pertencentes aos formulários com os do CRM, por exemplo, ao receber um dado no campo nome do formulário, o CRM irá armazenar esse registro no seu campo corresponde.

A tabela de “Leads” possui diversos campos, contudo apenas o sobrenome é obrigatório para a criação de um novo registro no sistema, além disso definiu-se o atributo *id_facebook* nesta entidade com o objetivo de armazenar o identificador do *lead* na plataforma do Facebook.

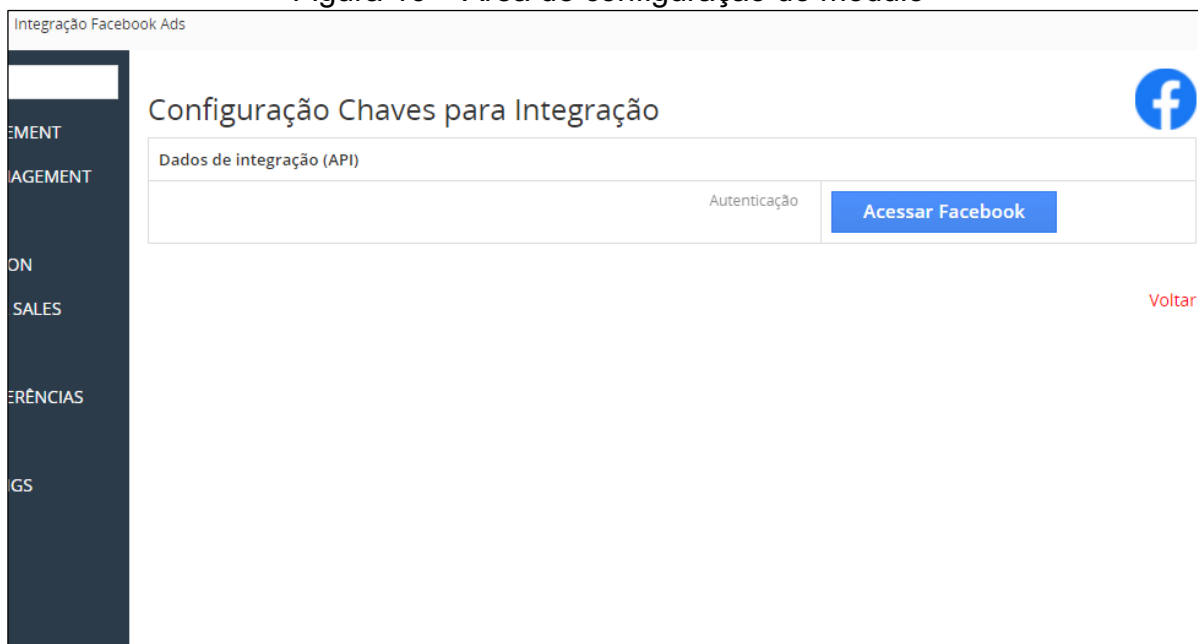
8.5 Desenvolver incrementos

Na quarta fase do modelo incremental é preciso desenvolver as funcionalidades propostas no incremento. O módulo construído para receber os *leads* provenientes do Facebook e Instagram possuem diversas funcionalidades. As funcionalidades desenvolvidas para o projeto são apresentadas de forma detalhada a seguir.

8.5.1 Área de configuração do módulo

A Figura 16 corresponde a tela de configuração da integração. Esta é a página inicial do módulo. Se o usuário não estiver autenticado anteriormente em sua conta Facebook, é mostrado o botão que permite acessar a mesma.

Figura 16 – Área de configuração do módulo



Fonte: Autoria Própria (2021)

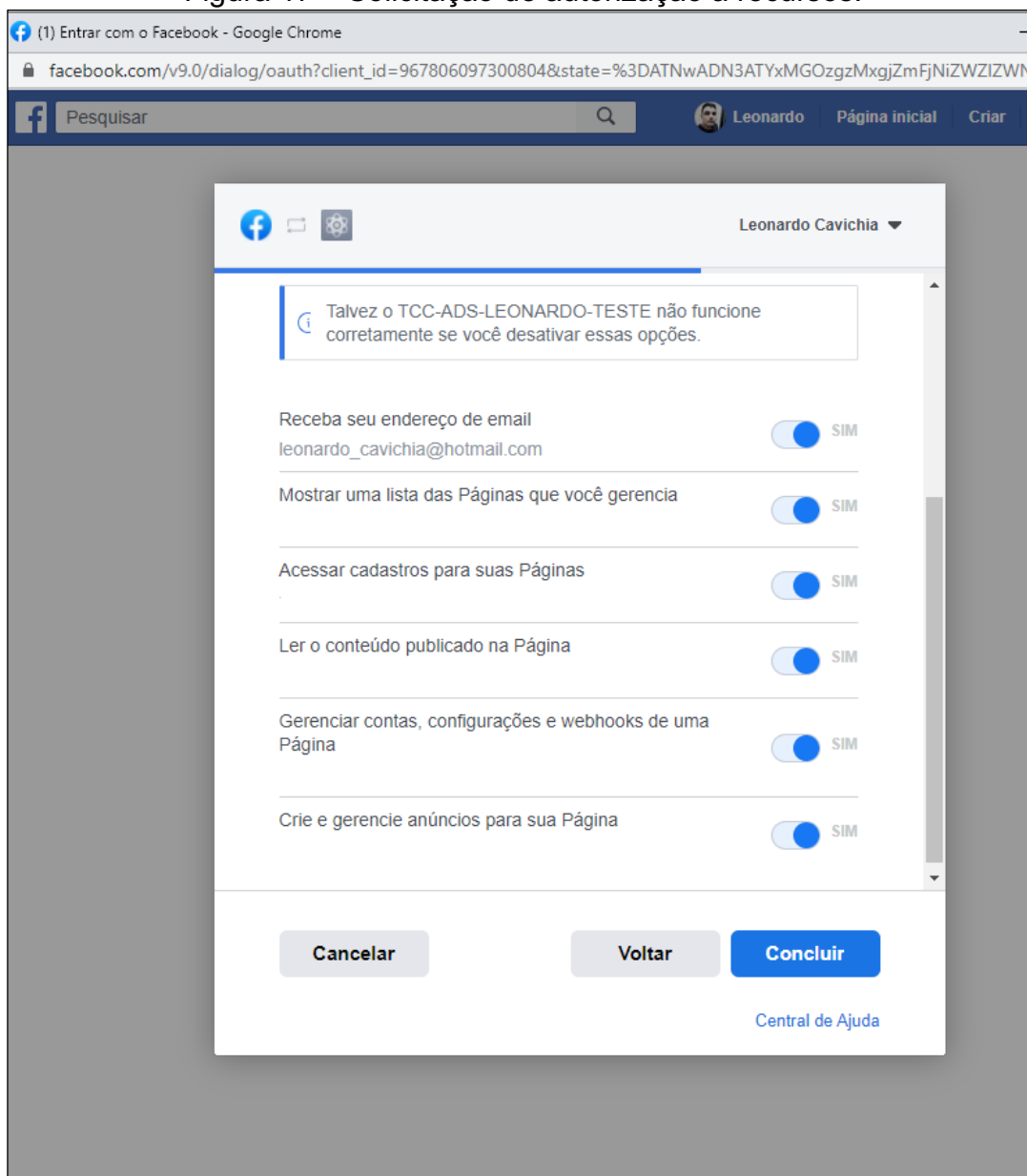
Esta tela contém inicialmente apenas o botão de acesso ao Facebook. Apesar do trabalho contemplar a rede social Instagram, apenas a conta Facebook é necessária devido à integração entre ambas as plataformas. Ao clicar no botão de acesso ao Facebook, é iniciado o processo de Autenticação e Autorização, que é abordado no tópico a seguir.

8.5.2 Autenticação e autorização

Para utilizar todas as funcionalidades fornecidas pela Graph API é necessário realizar a autenticação com seus dados de usuário e senha. O administrador do CRM deve utilizar uma conta do Facebook e autorizar todas as permissões de recursos solicitados. Ao clicar no botão para acessar sua conta Facebook, é exibido um pop-up solicitando as credenciais de acesso do usuário. Após a realização da autenticação

é preciso escolher à quais páginas o CRM terá acesso. Por fim, os recursos necessários são ilustrados na Figura 17.

Figura 17 – Solicitação de autorização a recursos.



Fonte: Autoria Própria (2021)

Para que o CRM possa realizar a captura de *leads* corretamente, o usuário deve conceder as seguintes permissões de acesso a recursos:

- Mostrar a lista de páginas gerenciadas: com essa permissão o sistema tem acesso as informações das páginas do usuário;
- Gerenciar anúncios em sua página: essa permissão permite o CRM gerenciar anúncios publicados por uma página;

- Gerenciar contas, configurações e *webhooks*: permite o sistema inscrever páginas em *webhooks* para receber alterações em registros;
- Acessar cadastros (*leads*): autoriza o sistema a recuperar e ler informações capturadas por formulários de anúncios em páginas do usuário;
- Perfil público e e-mail: obrigatório e ativado por padrão, possibilitando ler informações básicas do perfil do usuário, como e-mail, nome e foto de perfil.

Após realizar o acesso a sua conta, o CRM disponibiliza novas funções, que são abordadas nos tópicos a seguir.

8.5.3 Habilitar recebimento de leads

Na área de configuração são listadas todas as páginas que o administrador gerencia junto ao Facebook e por padrão, elas não estão habilitadas a enviar *leads* para o CRM. O usuário pode selecionar as páginas nas quais deseja habilitar o recebimento clicando em cada uma delas. Ao selecionar uma página, é enviada uma solicitação para a Graph API cadastrando a página selecionada no *webhook* do CRM.

Todas as páginas cadastradas são mostradas no campo de páginas inscritas, sendo possível removê-las. A Figura 18 ilustra a área de configuração com o usuário autenticado em sua conta.

Figura 18 – Área de configuração autenticado

Configuração Chaves para Integração

Dados de integração (API)

Autenticação [Remover Integração](#)

Páginas disponíveis (FACEBOOK > CRM)

Páginas [Registrar Página](#)

Páginas Inscritas (FACEBOOK > WEBHOOK)

Páginas [Remover Página](#)

Configurações adicionais (LEAD > CRM)

Não salvar lead sem a chave de validação:

Chave de validação:

Definir como os registros duplicados devem ser manipulados:

[Acessar Mapeamentos](#) [Voltar](#)

Fonte: Autoria Própria (2021)

Além das opções de inscrição das páginas no CRM, é possível realizar as configurações adicionais que permitem determinar como será realizado o tratamento do *lead* quando recebido pelo *webhook*. A opção de não salvar o *lead* sem a chave de validação, se selecionada, irá ignorar o *lead* recebido e não realizará o armazenamento dos dados no CRM. O campo chave de validação é utilizado para escolher qual será o campo obrigatório para o recebimento, por exemplo, se a chave for o e-mail, os *leads* recebidos do Facebook sem um e-mail válido serão ignorados. As opções de chaves são todos os campos disponíveis no módulo de *leads*.

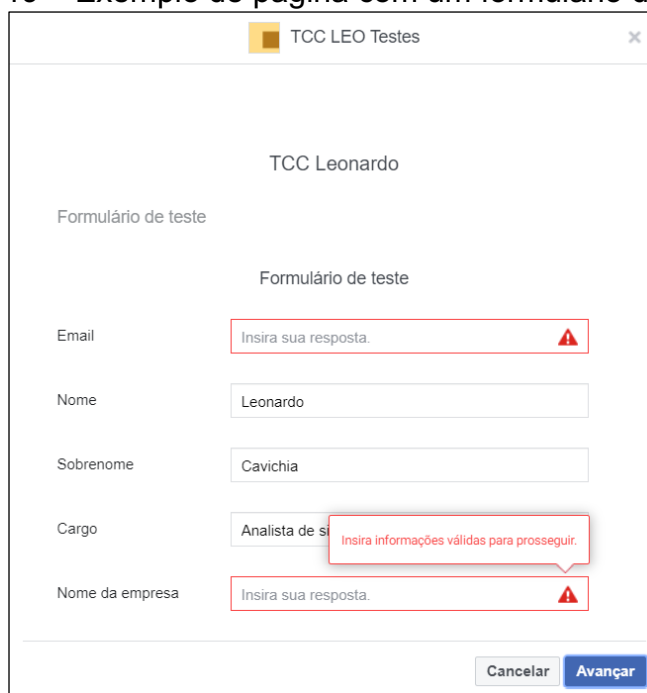
Por fim, na opção de definir como os registros duplicados serão manipulados é possível selecionar qual será a tratativa se forem encontrados *leads* já cadastrados no CRM com a mesma chave de validação. As opções de manipulação são:

- Nenhum: o *lead* será cadastrado no sistema ignorando a duplicidade das chaves.
- Sobrescrever: ao receber o *lead* do Facebook será feito a combinação dos dados com o registro já cadastrado no CRM.
- Pular: ignora o *lead* recebido.

8.5.4 Mapeamento de campos

Ao selecionar quais páginas o usuário gostaria de receber *leads*, é preciso realizar o mapeamento de campos entre os formulários de campanhas publicitárias e o CRM. Uma página em rede social pode possuir inúmeros formulários, que possuem campos obrigatórios, além de ser possível customizá-los. A Figura 19 ilustra um exemplo de página com um formulario e seus campos de entrada de dados.

Figura 19 – Exemplo de página com um formulário de dados



The image shows a browser window titled "TCC LEO Testes". Inside the window, the text "TCC Leonardo" is centered at the top. Below it, the text "Formulário de teste" appears twice. The form contains five input fields: "Email" (with a red border and a warning triangle icon), "Nome" (containing "Leonardo"), "Sobrenome" (containing "Cavichia"), "Cargo" (containing "Analista de si" and a red tooltip that says "Insira informações válidas para prosseguir."), and "Nome da empresa" (with a red border and a warning triangle icon). At the bottom right, there are two buttons: "Cancelar" and "Avançar".

Fonte: Autoria Própria (2021)

Na Figura 19, “TCC Leo Testes” é exemplificado como uma página no Facebook. Nesta mesma figura é mostrado um formulário de testes, utilizado para entrada de dados por parte dos usuários, e com isso e-mail, nome, sobrenome, cargo e por fim nome da empresa são representados como campos.

Para que o sistema possa armazenar corretamente os dados recebidos pela Graph API, é preciso habilitar o recebimento de *leads* por formulário mapeando todos os campos necessários. Para realizar o armazenamento, o módulo de *leads* do CRM necessita apenas do sobrenome, porém o Facebook possibilita a captura de inúmeras informações, como por exemplo, nome, e-mail, telefone, comentários, e estas necessitam ser correlacionadas corretamente.

A Figura 20 ilustra a área de configuração de mapeamentos de campos.

Figura 20 – Área de configuração de mapeamentos de campos

Mapeamento campos customizados
OBS: Antes de configurar o mapeamento, você deve configurar as chaves de integração

Páginas e Formulários disponíveis

Páginas: CRM TESTE

Formulários: Formulário de teste

Campos Facebook	Campos CRM
Email (E-mail)	Selecione uma opção
Full name (Nome Completo)	Seleccione uma opção Seleccione uma opção Receita Anual Bairro Buscar CEP Cadencia Prospecção Cidade Company Complemento Pais

Salvar

Fonte: Aatoria Própria (2021)

O usuário pode escolher entre as páginas habilitadas anteriormente e listar todos os formulários correspondentes. A tela irá mostrar todos os campos que o formulário possui. Aqui é necessário selecionar em qual campo do CRM, o campo do formulário (Facebook) será vinculado e cadastrado.

8.5.5 Recebimento de *leads* via *webhook*

Toda vez que é gerado um novo *lead* no Facebook ou Instagram ele é encaminhado para o CRM, chamando a URL projetada para o recebimento dos dados. Os dados enviados no *webhook* são apenas os identificadores do registro, ou seja, é recebido apenas o identificador da página, do formulário e do *lead*. Após verificar se os identificadores de página e formulário estão cadastrados no CRM, o identificador do *lead* é encaminhado para próxima etapa. Com o identificador do *lead* é preciso solicitar a Graph API todos os dados referentes ao registro, e após o retorno é possível seguir com a criação do *lead* no CRM.

8.6 Avaliar incrementos

Os testes do sistema são realizados na quinta fase do modelo incremental e podem ser realizados pelos desenvolvedores do projeto. O Facebook disponibiliza uma ferramenta¹³ online para simular o envio de *leads* a *webhooks* cadastrados. Nesta ferramenta é possível informar dados em todos os campos do formulário, ou apenas enviar com informações que o próprio Facebook disponibiliza, ou seja, dados fictícios. Os tópicos a seguir detalham e ilustram a realização de testes com a ferramenta de envio de *leads* e o recebimento dos mesmos no CRM.

8.6.1 Teste 1

O primeiro teste consistiu no envio do formulário com dados de nome e e-mail do *lead*. A Figura 21 ilustra o formulário de envio preenchido com dados do *lead*.

Figura 21 – Teste 1: Envio

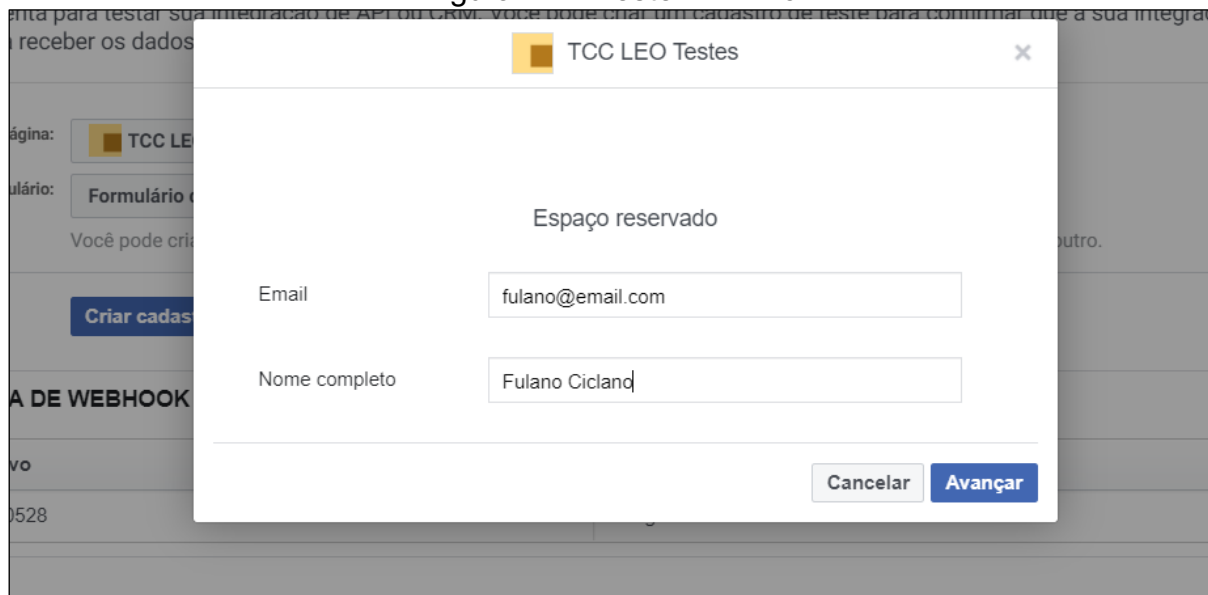


Figura 21 mostra uma janela de teste intitulada "TCC LEO Testes". O formulário dentro da janela contém os seguintes campos e valores:

Etiqueta	Valor
Email	fulano@email.com
Nome completo	Fulano Cicland

Na base da janela, há dois botões: "Cancelar" e "Avançar".

Fonte: Autoria Própria (2021)

Apesar de ser uma simulação, o formulário é apresentado igualmente ao que seria em uma entrada de dados verdadeira, por isso é possível digitar quaisquer informações nos campos de e-mail e nome completo. Como a página de teste não

¹³ Site da ferramenta: <https://developers.facebook.com/tools/lead-ads-testing>

possui uma logo cadastrada neste formulário, o formulário nos indicou que há um espaço reservado para a mesma. A Figura 22 ilustra o *lead* cadastrado no CRM.

Figura 22 – Teste 1: Recebimento

The screenshot shows a CRM interface for a lead named 'Fulano Ciclano'. The lead's email is 'fulano@email.com' and the source is 'FACEBOOK ADS'. The lead is assigned to 'Fulano Ciclano' with the code 'LEA-18918'. The interface includes a search bar, navigation tabs (Resumo, Detalhes, Atualizações, Comentários), and sections for 'Campos Chave', 'Atividades', and 'Comentários'. The 'Campos Chave' section shows the lead's name, code, and surname. The 'Atividades' section is empty. The 'Comentários' section has a text input field and a 'Postar' button.

Fonte: Autoria Própria (2021)

Observa-se que os dados estão em seus locais corretos, o campo “Nome” recebeu *Fulano Ciclano* e o campo “E-mail” possui *fulano@email.com*. Abaixo do campo e-mail possui a fonte do *lead*, neste caso, informa que foi recebido pelo *Facebook Ads*. Além disso, como o sobrenome é um campo obrigatório e sem ele não é possível armazenar o *lead* no sistema, atribuiu-se a ele o segundo dado do campo “Nome Completo”, ou seja, *Ciclano*.

8.6.2 Teste 2

Neste teste são enviados os mesmos dados do teste anterior, entretanto, o segundo teste consiste em mapear inadequadamente os campos de dados entre os sistemas. A Figura 23 ilustra o mapeamento de campos do Teste 2.

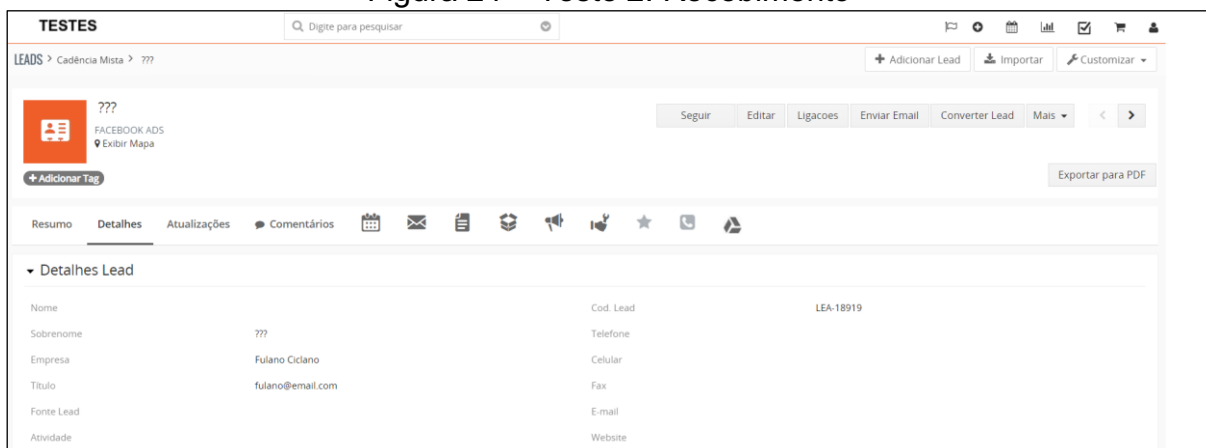
Figura 23 – Teste 2: Mapeamento

The screenshot shows a field mapping interface with two columns: 'Campos Facebook' and 'Campos CRM'. The 'Campos Facebook' column contains 'Email (E-mail)' and 'Full name (Nome Completo)'. The 'Campos CRM' column contains 'Título' and 'Empresa'. A green 'Salvar' button is located at the bottom center.

Fonte: Autoria Própria (2021)

Neste mapeamento foi indicado que o campo “E-mail” do Facebook correlaciona ao campo “Título no CRM” e o campo “Nome Completo” está vinculado ao campo Empresa. A Figura 24 ilustra como foi criado o *lead* com os campos mapeados erroneamente.

Figura 24 – Teste 2: Recebimento



Fonte: Autoria Própria (2021)

Neste recebimento observou-se que o campo “Empresa” foi preenchido com *Fulano Ciclano* e o campo “Título” foi cadastrado como *fulano@email.com*.

Como o sistema não reconheceu nenhuma entrada de dado no campo do “Sobrenome”, foi preenchido com “???”. Isto é útil, pois mesmo sem o sobrenome, é possível usar o Telefone ou E-mail para conseguir as informações não retornadas no formulário.

8.6.3 Teste 3

O terceiro teste consistiu em enviar um formulário com e-mail previamente cadastrado no CRM. A escolha de duplicidade foi de ignorar o novo registro e com isso o *lead* não foi armazenado no sistema.

8.6.4 Teste 4

No quarto teste foi enviado um formulário com campos diversos, como o endereço, nome da empresa, cargo e telefone. A Figura 25 apresenta o formulário de envio preenchidos com os dados diversos.

Figura 25 – Teste 4: Envio

The image shows a web browser window with a modal form titled "TCC LEO Testes". The form is titled "Formulário de testes" and "Espaço reservado". It contains the following fields and values:

Nome completo	Ciclano Fulano
Email	ciclano@email.com
Endereço	Rua joao da silva
Cargo	Analista de sistemas
Nome da empresa	Programadores PG
Telefone	BR +55 (41) 0000-0000

At the bottom of the form, there are two buttons: "Cancelar" and "Avançar". The background shows a blurred view of a website with various text elements like "Inteligencia artificial", "Developer Circles", "F8 Refresh", "Casos de sucesso", "Notícias", "Blog", "AR/VR", "400528", "cativo", "URA DE WEBHOOK", "Você pode cri", "ainda não tem um ca", "para receber os dados", "a sua integração está", "Página:", "TCC LE", "Formulário:", "TCC LEONA", "Criar cadas".

Fonte: Autoria Própria (2021)

Neste exemplo observa-se que foram enviados campos além do padrão “E-mail” e “Nome completo”. O campo “Endereço” pode ser somente a rua ou o endereço completo com rua, bairro, número, estado e país, isto, devido ao fato que o formulário não possui a opção de preencher esses campos individualmente. Além disso, foram preenchidos os campos de “Nome da empresa” e “Cargo” com dados genéricos. Por outro lado, o campo “Telefone” foi preenchido com zeros para representar um número qualquer.

A Figura 26 ilustra o *lead* criado no CRM para o teste 4.

Figura 26 – Teste 4: Recebimento

The screenshot shows a CRM interface for a lead named Ciclano Fulano. The lead's contact information is as follows:

Detalhes Lead			
Nome	Ciclano Fulano	Cod. Lead	LEA-18920
Sobrenome	Fulano	Telefone	
Empresa	Programadores PG	Celular	+55410000000
Título	Analista de sistemas	Fax	
Fonte Lead		E-mail	ciclano@email.com

Fonte: Autoria Própria (2021)

Os campos “Empresa” e “Título” receberam os dados *Programadores PG* e *Analista de sistema*, respectivamente. A rua do endereço foi preenchida com *Rua João da Silva*, além disso, se o *lead* tivesse colocado dados como número, bairro e cidade, a equipe de vendas poderia separar os dados e atualizar o *lead* manualmente no CRM. O campo “Celular” está em azul pois é possível clicar e iniciar uma ligação para este contato. Ao clicar no E-mail é aberto a tela de envio de e-mails com os dados deste *lead* previamente preenchidos.

8.7 Projeto do sistema com reuso

A sexta fase do modelo incremental é utilizada para reaproveitar partes do sistema já desenvolvidas. Além disso, nesta etapa é acoplado o incremento com os componentes já desenvolvidos em versões anteriores. Como este projeto utilizou somente um ciclo do modelo incremental, ele foi desenvolvido integrado ao sistema do Vtiger CRM.

8.8 Validar sistema

Na sétima fase do modelo incremental é preciso validar o sistema com o usuário final, ou seja, o sistema está pronto para uso, porém ainda não recebeu o aval final do cliente para o encerramento do contrato.

Uma campanha de publicidade em redes sociais pode durar de dias a semanas, além disso é preciso considerar um bom investimento financeiro que possa atrair o público correto ao seu negócio. Como este trabalho não abordou um negócio em específico, como por exemplo, venda de produtos e/ou oferecimento de serviços, mas sim campanhas para qualquer tipo de negócio, não foi possível realizar os testes finais, visto que uma campanha financeiramente criada sem um público alvo delimitado corretamente, não atingiria quase nenhum alcance.

9 CONCLUSÃO

Este trabalho apresentou a criação de um módulo de integração para o Vtiger CRM, que auxilia a captura de *leads* por meio das campanhas de publicidade em redes sociais. O objetivo geral foi alcançado com êxito.

O modelo incremental foi utilizado como metodologia de desenvolvimento, isto porque ele é recomendado para sistemas que não possuem todos os requisitos previamente definidos. Caso houvessem novos requisitos no meio do desenvolvimento deste trabalho, como por exemplo, incluir outras redes sociais na captura dos *leads*, seria necessário um novo ciclo do modelo incremental, além disso seria possível utilizar a base de código já criada anteriormente.

Como não foi possível validar o sistema com usuários e campanhas de marketing reais, não foi possível determinar novos requisitos para esta integração. Embora o projeto aborde duas grandes redes sociais do mercado, muitas delas não foram utilizadas e possuem clientes em potencial.

9.1 Trabalhos futuros

Visto a importância da captura de *leads* por meio das campanhas de publicidade, sugere-se como trabalhos futuros a integração com novas redes sociais, como por exemplo, o LinkedIn¹⁴ e Tiktok¹⁵. Além disso, seria interessante a possibilidade de criar formulários de publicidades via CRM, pois isso facilitaria o mapeamento de campos entre os sistemas.

Como requisitos futuros para o módulo de integração criado neste trabalho, será necessário manter as bibliotecas de código sempre atualizadas, visto que o Facebook e Instagram estão sempre incluindo novas funcionalidades e até mesmo realizando mudanças em pontos-chaves do seu projeto.

¹⁴ Site LinkedIn: <https://br.linkedin.com/>

¹⁵ Site Tiktok: <https://www.tiktok.com/pt-BR/>

REFERÊNCIAS

- APACHE. **What is the Apache HTTP Server Project**. Disponível em: https://httpd.apache.org/ABOUT_APACHE.html. Acesso em: 25 jun. 2021.
- BAFUTTO, M (2019). **CRM: O que é CRM e quais os benefícios? Definição e conceitos do CRM**. Disponível em: <https://blog.nectarcrm.com.br/o-que-e-crm>. Acesso em: 08 mar. 2021.
- BASSET, L. **Introdução ao JSON**. 1. ed. São Paulo: Novatec Editora Ltda, 2015.
- BERNERS-LEE, T. **Uniform Resource Identifier (URI): Generic Syntax. RFC 3986**. Jan. 2005. Disponível em: <https://tools.ietf.org/html/rfc3986>. Acesso em: 25 abr. 2021.
- CRANE, D.; PASCARELLO, E.; JAMES, D. **Ajax em ação**. 1. ed. Person Prentice Hall, 2007.
- DALL'OGGIO, PABLO. **PHP Programando com orientação a objetos**. 3. ed. Novatec Editora Ltda, 2017.
- FIELDING, R. RESCHKE, J. **Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. RFC 7231**. Jun. 2014. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7231>. Acesso em: 20 maio 2021.
- GLUOCRM (2021). **Porque usar CRM: 7 motivos fundamentais para usar um CRM**. Disponível em: <https://www.gluocrm.com.br/blog/7-motivos-fundamentais-para-voce-utilizar-um-crm>. Acesso em: 20 mar. 2021.
- GUEDES, T. A. G. **UML 2 Guia Prático**. 2. ed. São Paulo, SP: Novatec Editora Ltda, 2014.
- HAMMER, L.E. **The OAuth 1.0 Protocol. RFC 5849**. Abr. 2010. Disponível em: <https://tools.ietf.org/html/rfc5849>. Acesso em: 28 mar. 2021.
- HARDT, D. **The OAuth 2.0 Authorization Framework. RFC 6749**. Out. 2012. Disponível em: <https://tools.ietf.org/html/rfc6749>. Acesso em: 28 mar. 2021.
- LECHETA, R. R. **Web Services RESTful**. 1. ed. São Paulo, Novatec Editora Ltda, 2015.
- MOZILLA. **Códigos de status de respostas HTTP**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status>. Acesso em: 05 abr. 2021b.
- MOZILLA. **Primeiros passos AJAX**. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/Guide/AJAX/Getting_Started. Acesso em: 05 abr. 2021c.
- MOZILLA. **Uma visão geral do HTTP**. 6 abr. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>. Acesso em: 05 abr. 2021a.
- MYSQL. **What is MySQL**. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. Acesso em: 25 jun. 2021.
- PHP. **O que o PHP pode fazer**. Disponível em: https://www.php.net/manual/pt_BR/intro-whatcando.php. Acesso em: 10 abr. 2021.
- PRESSMAN, R.S. **Engenharia de software: uma abordagem profissional**. 7. ed. McGraw Hill, 2011.
- REDHAT. **O que é um SDK**. Disponível em: <https://www.redhat.com/pt-br/topics/cloud-native-apps/what-is-SDK>. Acesso em: 25 jun. 2021.

ROSSI, D, IAN. **Beginner's Guide Record and consolidate all your customer information with vtiger CRM**. Ed: Packt Publishing, 2011.

SALESFORCE. **CRM**. Disponível em: <https://www.salesforce.com/br/crm>. Acesso em: 05 mar. 2021.

SANTOS, E. (2020). **O que é Lead e para que serve a gestão de Leads**. Disponível em: <https://resultadosdigitais.com.br/blog/leads>. Acesso em: 05 mar. 2021.

SENDGRID. **What's a Webhook?**. 20 jun. 2014. Disponível em: <https://sendgrid.com/blog/whats-webhook>. Acesso em: 12 mar. 2021.

SMARTY. **What is Smarty**. Disponível em: https://www.smarty.net/about_smarty. Acesso em: 05 mar. 2021.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo, SP: Pearson Prentice Hall, 2011.

VOLPATO, B. (2021). **Ranking das redes sociais 2020**. Disponível em: <https://resultadosdigitais.com.br/blog/redes-sociais-mais-usadas-no-brasil>. Acesso em: 05 mar. 2021.

VTIGER. **Docs**. Disponível em: <https://www.vtiger.com/docs>. Acesso em: 24 maio. 2021b.

VTIGER. **O verdadeiro CRM de código aberto**. Disponível em: <https://www.vtiger.com/pt/open-source-crm/>. Acesso em: 24 maio. 2021a.

W3C. **O que é CSS**. Disponível em: <https://www.w3.org/Style/CSS>. Acesso em: 22 abr. 2021b.

W3C. **XML ESSENTIALS**. Disponível em: <https://www.w3.org/standards/xml/core>. Acesso em: 22 abr. 2021a.

APÊNDICE A – Descrições dos casos de uso

Quadro 2 – Caso de Uso Realizar Login

Nome do caso de uso	Realizar Login
Ator principal	Administrador
Atores secundários	
Resumo	Descreve quais são as etapas para realizar a autenticação e autorização com sua conta
Pré-condições	O ator precisa ser administrador de suas páginas nas redes sociais e também do CRM
Pós-condições	O sistema atualiza a tela para mostrar novas funcionalidades e informa que o Login foi realizado com êxito
Fluxo principal	
Ações do ator	Ações do sistema
1. Solicita acesso a sua conta Facebook	
	2. Mostrar tela de autenticação e autorização ao administrador
3. Informar Login e Senha de sua conta Facebook	
	4. Validar informações de acesso
	5. Mostrar permissões disponíveis para escolha
5. Escolher quais páginas e permissões de dados o sistema CRM terá acesso em seu nome	
	6. Validar permissões escolhidas
	7. Salvar chave de acesso no sistema CRM
Fluxo alternativo 1 - Acesso inválido	
Ações do ator	Ações do sistema
	Informar ao usuário que seus dados de acesso são inválidos, e que não foi possível realizar o Login
Fluxo alternativo 2 - Permissões inválidas	
	Informar ao usuário que as permissões escolhidas são insuficientes para que a integração funcione corretamente, sendo

	necessário refazer o processo de autenticação e autorização
--	---

Fonte: Aatoria Própria (2021)

Quadro 3 – Caso de Uso Realizar Logout

Nome do caso de uso	Realizar Logout
Ator principal	Administrador
Atores secundários	
Resumo	Descreve o fluxo de etapas para realizar o logout de sua conta Facebook
Pré-condições	O administrador precisa ter realizado o Login anteriormente.
Pós-condições	A tela da integração é atualizada e o botão para realizar o Login é mostrado
Fluxo principal	
Ações do ator	Ações do sistema
1. Clicar no botão para realizar o Logout de sua conta Facebook	
	2. Remover todas as inscrições de páginas no <i>Webhook</i>
	3. Remover todas as chaves de acesso do banco de dados
	4. Informar ao administrador a conclusão com sucesso do Logout de conta
Fluxo alternativo 1 – Não foi possível remover páginas no <i>Webhook</i>	
Ações do ator	Ações do sistema
	Informar ao administrador que é necessário remover as páginas manualmente pela plataforma do Facebook

Fonte: Aatoria Própria (2021)

Quadro 4 – Caso de Uso Gerenciar configurações adicionais

Nome do caso de uso	Gerenciar configurações adicionais
Ator principal	Administrador
Atores secundários	
Resumo	Descreve as etapas para realizar o salvamento das configurações adicionais do módulo

Pré-condições	O administrador precisa ter realizado o Login com sua conta Facebook
Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema
1. Escolher se quer receber os <i>leads</i> sem uma chave de validação	
	2. Salvar opção escolhida no CRM
3. Escolher qual será a chave de validação	
	4. Salvar opção escolhida no CRM
5. Definir como os registros duplicados devem ser manipulados	
	6. Salvar opção escolhida no CRM
Fluxo alternativo – Não há	

Fonte: Autoria Própria (2021)

Quadro 5 – Caso de Uso Gerenciar Páginas

Nome do caso de uso	Gerenciar Páginas
Caso de uso geral	
Ator principal	Administrador
Atores secundários	
Resumo	Descreve as etapas para realizar a inscrição de uma página para o recebimento de <i>leads</i>
Pré-condições	O administrador precisa estar logado em sua conta Facebook com páginas disponíveis
Pós-condições	O sistema remove a página inscrita das opções disponíveis para inscrição
Fluxo principal	
Ações do ator	Ações do sistema
1. Solicitar as páginas disponíveis para inscrição	
	2. Validar quais são as páginas que o usuário possui em sua conta Facebook e apresentar ao usuário
3. Inscrever uma página no recebimento de <i>leads</i>	

	4. Enviar a solicitação de inscrição ao Graph API
Fluxo alternativo 1 – Remover Página Inscrita	
Ações do ator	Ações do sistema
1. Solicitar as páginas disponíveis para a remoção	
	2. Validar quais são as páginas que o usuário possui em integração com o CRM e apresentar ao usuário
3. Remover uma página no recebimento de <i>leads</i>	
	4. Enviar a solicitação de remoção ao Graph API
Fluxo alternativo 1 – Inscrição não permitida	
Ações do ator	Ações do sistema
	1. Informar ao usuário que a página não pode ser inscrita por algum motivo.
Fluxo alternativo 1 – Remoção não permitida	
Ações do ator	Ações do sistema
	1. Informar ao usuário que a página não pode ser removida por algum motivo. Sendo necessário sua remoção manual pela plataforma do Facebook

Fonte: Autoria Própria (2021)

Quadro 6 – Caso de Uso Listar Formulários

Nome do caso de uso	Listar Formulários
Ator principal	Administrador
Atores secundários	
Resumo	Descreve as etapas para a listagem dos formulários pertencentes as páginas do Facebook
Pré-condições	Ter cadastrado ao menos uma página para o recebimento de <i>leads</i>
Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema
1. Solicitar as páginas inscritas no <i>Webhook</i>	

	2. Retornar todas as páginas que o usuário se inscreveu para o recebimento de <i>leads</i>
3. Selecionar uma página para listagem dos seus formulários	
	4. Listar todos os formulários pertencentes à página selecionada
5. Selecionar um formulário para listagem dos seus campos de dados	
	6. Listar todos os campos de dados pertencentes ao formulário selecionado
Fluxo alternativo 1 – Não possuir formulários disponíveis	
Ações do ator	Ações do sistema
	Informar ao usuário que a página selecionada não possui formulários disponíveis

Fonte: Autoria Própria (2021)

Quadro 7 – Caso de Uso Mapear Campos

Nome do caso de uso	Mapear Campos
Ator principal	Administrador
Atores secundários	
Resumo	Relata sobre as etapas para realizar o mapeamento de campos entre o CRM e o formulário escolhido
Pré-condições	Usuário ter selecionado um formulário no caso de uso Listar Formulários
Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema
1. Relacionar os campos do formulário com os campos do CRM.	
2. Clicar no botão para salvar as configurações de mapeamento	
	3. Salvar mapeamento no CRM
Fluxo alternativo 1 – Não foi possível salvar mapeamento	
Ações do ator	Ações do sistema

	Informar ao usuário que foram relacionados todos os campos do formulário
--	--

Fonte: Autoria Própria (2021)

Quadro 8 – Caso de Uso Disparar *Webhook*

Nome do caso de uso	Disparar <i>Webhook</i>
Ator principal	Facebook
Atores secundários	
Resumo	Descreve as etapas que o <i>Webhook</i> realiza ao receber um novo <i>lead</i> da rede social
Pré-condições	Ter ao menos uma página e um formulário cadastrados corretamente no CRM
Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema
1. Encaminhar o novo <i>lead</i> ao <i>Webhook</i> do CRM	
	2. Validar se o identificador da página possui cadastro no sistema
	3. Validar se o identificador do formulário possui cadastro no sistema
	4. Validar se o <i>lead</i> possui seu identificador único
	5. Requisitar o restante dos dados do <i>lead</i> a Graph API
	6. Encaminhar <i>lead</i> para validar os dados recebidos
Fluxo alternativo 1 – Página inválida	
Ações do ator	Ações do sistema
	Encerrar o fluxo do <i>webhook</i>
Fluxo alternativo 2 – Formulário inválido	
Ações do ator	Ações do sistema
	Encerrar o fluxo do <i>webhook</i>
Fluxo alternativo 3 – Lead Invalido	
Ações do ator	Ações do sistema

	Encerrar o fluxo do <i>webhook</i>
--	------------------------------------

Fonte: Autoria Própria (2021)

Quadro 9 – Caso de Uso Validar Novo *Lead*

Nome do caso de uso	Validar Novo <i>Lead</i>
Ator principal	Facebook
Atores secundários	
Resumo	Descreve todas as etapas que o novo <i>lead</i> precisa passar antes de ser salvo no sistema
Pré-condições	Ter concluído o caso de uso Disparar <i>Webhook</i>
Pós-condições	
Fluxo principal	
Ações do ator	Ações do sistema
	1. Correlacionar os dados do <i>lead</i> com os campos do CRM
	2. Com base nas configurações adicionais, validar se o <i>lead</i> precisa da chave de validação preenchida
	3. Validar qual é a tratativa de duplicidade escolhida pelo usuário anteriormente
	4. Encaminhar <i>lead</i> para criação no CRM
Fluxo alternativo 1 – Chave de validação não preenchida	
Ações do ator	Ações do sistema
	Ignorar <i>lead</i> recebido
Fluxo alternativo 2 – <i>Lead</i> duplicado e com tratativa de ignorar	
Ações do ator	Ações do sistema
	Ignorar <i>lead</i> recebido

Fonte: Autoria Própria (2021)