



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ANTONIO HENRIQUE DA SILVA LAGES COSTA

**MARKETING ANALYTICS - UMA API PARA POTENCIALIZAR O MARKETING
DIGITAL**

TERESINA, PIAUÍ

2024

ANTONIO HENRIQUE DA SILVA LAGES COSTA

MARKETING ANALYTICS - UMA API PARA POTENCIALIZAR O MARKETING
DIGITAL

Projeto apresentado à Banca Examinadora como requisito para aprovação na disciplina de Trabalho de Conclusão de Curso II do Curso Superior de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí.

Orientador: Prof. Dr. Adalton de Sena Almeida

TERESINA, PIAUÍ

2024

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Costa, Antônio Henrique da Silva Lages

C837m Marketing analytics : uma API para potencializar o marketing digital /
Antônio Henrique da Silva Lages Costa. - 2024.
86 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e
Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e
Tecnologia do Piauí, Campus Teresina Central, 2024.

Orientador : Prof Dr. Adalton de Sena Almeida.

1. Marketing digital. 2. Ferramenta. 3. Clientes. 4. Empresas. I. Título.

CDD - 004.21

Elaborado por Maria Rosismar Farias CRB 3/631

ANTONIO HENRIQUE DA SILVA LAGES COSTA

MARKETING ANALYTICS - UMA API PARA POTENCIALIZAR O MARKETING
DIGITAL

Projeto apresentado à Banca Examinadora como requisito para aprovação na disciplina de Trabalho de Conclusão de Curso II do Curso Superior de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí.

Aprovado pela banca examinadora em _____.

BANCA EXAMINADORA:

Prof. Dr. Adalton de Sena Almeida

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

Prof. M^º. Rogério da Silva

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

Prof. Dr. Thiago Elias

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

TERESINA, PIAUÍ

2024

Dedico este trabalho a todos que acreditaram que ele sairia.

AGRADECIMENTOS

Agradeço primeiramente à minha mãe Bernardete que sempre fez o possível para que eu pudesse concluir este curso.

Agradeço a todos que acompanharam esta jornada desde o começo diretamente em especial a Isadora e o Nicolas e indiretamente - meu primo Alexandre, o professor Rogério da Silva, meus amigos do dia a dia e todos com quem já tive o prazer de trabalhar junto - através do compartilhamento de conhecimento e experiência.

Agradeço a banca examinadora que compartilhou informações que foram de suma importância para a escrita deste trabalho.

Por fim, agradeço ao meu orientador Adalton Sena, por ter tido a paciência em me ajudar neste trabalho, bem como por ter me incentivado a começar a ler com mais frequência.

“Posso falhar, mas não vou desistir.”
(Ryze, O Mago Rúnico)

RESUMO

No constante crescimento do mercado de *marketing* digital, empresas que buscam fornecer serviços de qualidade e com resultados significativos acabam tendo vantagens diante das concorrentes. Por consequência, o portfólio de clientes da organização acaba crescendo e as atividades que são realizadas em cima desses clientes também crescem tornando-se complexas e árduas. Diante dessa situação, as empresas de *marketing* digital acabam buscando ferramentas especializadas em *marketing* disponíveis no mercado, entretanto, deparam-se com outro problema. Estas ferramentas trabalham de forma segmentada, podendo trazer ainda mais complexidade na organização interna destas companhias.

Diante desse contexto, surgiu o 'Marketing Analytics', sendo uma ferramenta que permite a centralização de dados das plataformas do Facebook, Instagram e Google, em uma única plataforma. Ao ter estes dados centralizados, será possível a criação de relatórios customizados, um gerenciamento fácil e ágil dos clientes, bem como uma análise adequada desses dados. Além do mais, este produto permite que usuários possam desenvolver suas próprias telas, podendo incrementar ainda mais a proposta deste trabalho, através do consumo da API que é disponibilizada.

Palavras-chaves: marketing digital. ferramenta. clientes. empresas. dados

ABSTRACT

In the constant growth of the digital marketing market, companies that seek to provide quality services with significant results end up having an advantage over their competitors. competitors. As a result, the organization's client portfolio grows and the activities that are carried out on behalf of these clients also grows, becoming complex and arduous. Faced with this situation, digital marketing companies end up looking for specialized marketing tools available on the market. However, they run into another problem, These tools work in a segmented way, which can make the internal organization of these companies even more complex. organization of these companies.

In this context, 'Marketing Analytics' has emerged as a tool that allows for the the centralization of data from the Facebook, Instagram and Google platforms into a single platform. By having this data centralized, it will be possible to create customized reports, easily and easy and agile management of customers, as well as proper analysis of this data. What's more Furthermore, as a bonus, this product allows users to develop their own screens, and can further enhance the proposal of this work by using the API that is available.

Key-words: digital marketing. tool. clients. companies. data.

LISTA DE ILUSTRAÇÕES

Figura 1 – Captura de Tela da tela inicial do mLabs	23
Figura 2 – Captura de Tela da tela inicial do RD Station	24
Figura 3 – Captura de Tela da tela inicial do Make	25
Figura 4 – Exemplo de documentação gerada pelo framework FastAPI	28
Figura 5 – Exemplo de documentação gerada pelo framework FastAPI que detalha uma rota específica	28
Figura 6 – Exemplo de criação de pontos de extremidade com todos os métodos	29
Figura 7 – Exemplo de criação de pontos de extremidade somente com os métodos desejados	30
Figura 8 – Exemplo de criação de pontos de extremidade somente com os métodos desejados	30
Figura 9 – Exemplo de criação de pontos de extremidade somente com os métodos desejados	30
Figura 10 – Exemplo de criação de pontos de extremidade somente com os métodos desejados	31
Figura 11 – Estrutura da Marketing API	38
Figura 12 – Como funciona a Graph API	39
Figura 13 – Diagrama UML de Casos de Uso 01 do Marketing Analytics	41
Figura 14 – Diagrama UML de Casos de Uso 02 do Marketing Analytics	41
Figura 15 – Diagrama UML de Casos de Uso 03 do Marketing Analytics	42
Figura 16 – Diagrama UML de Casos de Uso 04 do Marketing Analytics	43
Figura 17 – Diagrama de Classes	44
Figura 18 – Arquitetura monolítica	45
Figura 19 – Arquitetura do Marketing Analytics	46
Figura 20 – Diagrama de Entidade e Relacionamento	47
Figura 21 – Captura de tela da tela inicial do painel administrativo do Marketing Analytics	48
Figura 22 – Captura de tela da tela que lista as tarefas periódicas do Marketing Analytics	49
Figura 23 – Captura de tela da tela que lista os usuários do Marketing Analytics	50
Figura 24 – Captura de Tela do Quadro do Trello	55
Figura 25 – Exemplo básico da utilização do Model Bakery	58
Figura 26 – Exemplo básico da utilização da Factory Boy	58
Figura 27 – Exemplo básico da utilização da Factory Boy especificando campos	58
Figura 28 – Relatório de cobertura de testes gerado pela biblioteca Coverage	59

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
REST	Representational State Transfer
HTML5	Hypertext Markup Language, versão 5
HTTPS	Hyper Text Transfer Protocol Secure
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
XML	Extensible Markup Language
MVC	Model View Controller
MTV	Model Template View
ORM	Object/Relational Mapping
SGBD	Sistema de Gerenciamento de Banco de Dados
NIST	National Institute of Standards and Technology
DRF	Django REST framework
ROI	Return on investments
SEO	Search Engine Optimization
CPC	Custo Por Click
T.I	Tecnologia e Informação
QA	Quality Assurance
E2E	End-to-End
PIP	Package Installer for Python
GA4	Google Analytics 4
UA	Universal Analytics
AMQP	Advanced Message Queuing Protocol
SQL	Structured Query Language

NOSQL	Banco de Dados Não Relacional
AWS	Amazon Web Service
CI/CD	Continuos Integration and Continuos Delivery
GB	Gigabyte

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Contextualização	14
1.2	Justificativa	15
1.3	Objetivos	16
1.3.1	Objetivo Geral	16
1.3.2	Objetivos Específicos	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	O que é o Marketing Digital	18
2.2	Diferenças entre Marketing Digital e Tradicional	20
2.3	Métricas no marketing digital: <i>Insights</i>	21
2.4	O que é um <i>Marketing</i> de Sucesso	21
2.5	Trabalhos Relacionados	22
2.5.1	mLabs	22
2.5.2	RD Station Marketing	24
2.5.3	Make	24
3	TECNOLOGIAS UTILIZADAS	26
3.1	Python	26
3.2	Django	26
3.3	Django REST framework	29
3.4	Celery	31
3.4.1	Flower	32
3.4.2	RabbitMQ	32
3.5	PostgreSQL	32
3.6	Redis	33
3.7	Vultr Object Storage	33
3.8	Git	33
3.9	Docker	34
4	MODELAGEM DO PROJETO	35
4.1	Requisitos	35
4.2	Entendendo a Estrutura Base	36
4.2.1	Facebook Pages API	37
4.2.2	Marketing API	37
4.2.3	Instagram API	38

4.2.4	Representação da relação da Graph API com as demais APIs	39
4.2.5	Webhooks	39
4.3	Diagramas de casos de uso	40
4.4	Diagramas de classe	43
4.5	Arquitetura do Sistema	44
4.6	Diagrama de Entidades-Relacionamentos	47
4.7	Interface	48
4.8	Padrões de Projeto	50
4.8.1	Adapter	51
4.8.2	Factory Method	51
5	SOFTWARE	52
5.1	O que são metodologias de desenvolvimento de software	52
5.1.1	Metodologias tradicionais	53
5.1.1.1	Modelo Cascata	53
5.1.2	Métodos ágeis	53
5.1.2.1	SCRUM	53
5.1.3	Aplicação neste projeto	54
5.2	Teste de software	55
6	CONCLUSÃO	60
6.1	Trabalhos Futuros	60
	REFERÊNCIAS	62
	ANEXO A – ESPECIFICAÇÃO DE CASOS DE USO	65
	ANEXO B – DETALHAMENTO DOS REQUISITOS	73
B.1	Introdução	73
B.2	Requisitos Funcionais	73
B.3	Requisitos Não-Funcionais	85
B.3.1	Requisitos de Produtos	85
B.3.1.1	Usabilidade	85
B.3.1.2	Confiabilidade	85
B.3.1.3	Segurança	85
B.3.2	Requisitos de Processo	86
B.3.2.1	Padrões	86
B.3.2.2	Organização	86

1 INTRODUÇÃO

Neste capítulo, iremos contextualizar a respeito das dificuldades que empresas do ramo do *marketing* digital enfrentam no seu dia a dia. Em seguida, são apresentadas as justificativas utilizadas para este *software*.

1.1 CONTEXTUALIZAÇÃO

As primeiras formas de vendas eram caracterizadas por serem apenas uma simples troca direta de produtos, tal prática era conhecida como escambo. Durante o período do Brasil Colônia esta atividade foi bastante utilizada pelos Portugueses que ofereciam utensílios domésticos aos indígenas em troca da extração do Pau-brasil. Porém, a medida que os primeiros mercados iriam surgindo e o comércio iria se expandido, era necessário atrair cada vez mais os clientes. Por volta do século XX, o *marketing* teve sua origem, focando em impulsionar as vendas e a distribuição de produtos, no qual foi inicialmente, associado a economia e tendo um grande impacto nas relações da mesma com a sociedade[1].

Ainda por volta do século XX ocorreu a Terceira Revolução Industrial, sendo esta marcada por diversos avanços tecnológicos em diversas áreas, como a informática, robótica, telecomunicações e também pela substituição da mão-de-obra por máquinas em indústrias ocasionando um crescente desemprego[2]. Um desses avanços é a Internet, sendo esta uma rede de computadores que inicialmente era restrita, porém, a sua evolução proporcionou que a mesma saísse do mundo acadêmico e de certos grupos sociais e fosse colocada a disposição do público em geral[3]. Este marco foi de grande importância para o mercado do *marketing*, pois uma vez que os usuários estavam conectados nas mídias sociais era possível utilizar dos canais *on-lines* para direcionar propagandas de produtos ou serviços para estes usuários. Dessa forma, surgiram também novas formas de vender, tendo destaque o *e-commerce* caracterizado por permitir a venda de produtos e/ou serviços através do meio digital. Diante desse cenário, surgiu-se o *marketing* digital em que o seu grande foco era a utilização de mídias sociais, como por exemplo, *sites*, redes sociais, *blogs*, para impulsionar vendas ou até mesmo criar presenças no meio digital, por consequência, surgiu a profissão dos especialistas na área.

Em meados de março de 2020, foi declarada pandemia mundial e devido a isto, medidas de quarentena foram tomadas pelos representantes governamentais. Dessa forma, o *e-commerce* ficou em destaque, uma vez que as empresas tiveram que mudar sua forma de vender, deixando o ambiente físico e migrando para o digital[4]. Nesta modalidade, as empresas conseguiram alguns benefícios, como a redução de custo ao realizar pedidos, uma vez que o usuário pode navegar pela interface e a diminuição de erros durante o processo dentre outros benefícios[5]. Dessa forma, era possível também obter métricas e *Feedback* de clientes de forma rápida e fácil.

Diante dessa situação, empresas de *marketing* começaram a utilizar ainda mais plataformas como Facebook, Instagram e Google, sendo para oferecer serviços como *Social Media*¹ ou para impulsionar a venda de produtos e/ou serviços através de campanhas e anúncios. Estas atividades permitem que sejam obtidos métricas importantes para que seja realizado um *marketing* de sucesso. Ademais, para garantir um relacionamento amigável e duradouro com seus clientes, bem como oferecer um serviço de máxima qualidade, além de utilizar as ferramentas que as próprias plataformas disponibilizavam, a empresa buscou outras ferramentas voltadas para o ramo do *marketing* digital disponíveis no mercado para que pudesse aprimorar e melhorar os serviços que eram oferecidos .

1.2 JUSTIFICATIVA

A evolução da internet proporcionou novos meios de comunicação e interação entre pessoas e grupos. Um desses meios é as redes sociais, espaços no mundo digital onde o usuário poderia expressar opiniões e até mesmo obter informações[6]. Porém, esta evolução trouxe consigo novas formas de venda através também do meio digital. Dessa forma, plataformas como Instagram, Facebook e Google ganharam forças no mercado de *marketing* digital, uma vez que dominando e entendendo o funcionamento desses aplicativos era possível estabelecer um vínculo e posteriormente uma relação duradora e saudável com seus clientes[6], assim como estabelecer uma presença no meio digital servindo como base para poder alavancar as vendas tornando esses recursos essenciais para os profissionais da área.

Para atingir estes objetivos, é necessário entender as dificuldades que o cliente está enfrentando, podendo então definir estratégias que podem alcançar os resultados desejados e por fim sanar os problemas expostos. Todavia, a medida que a carteira de clientes vai se estendendo, os problemas ficam cada vez mais visíveis: a dificuldade para gerenciar esses clientes de forma rápida e prática de modo que não prejudique a produtividade do profissional, tal qual uma análise adequada dos seus dados com o intuito de criar estratégias de *marketing* em que seja possível resolver os problemas do cliente.

Nesse contexto, o gerenciamento de redes sociais ou popularmente conhecido como *Social Media*, excede o simples agendamento de conteúdo nas plataformas para os clientes ou até mesmo o monitoramento de novos seguidores e curtidas em postagens. Refere-se a um agrupamento abrangente de estratégias com o objetivo de atrair e engajar o público alvo. Profissionais da área, além de realizar estas atividades, devem definir estratégias diárias de conteúdos para os clientes. Estas estratégias devem ser ajustadas com horários ideais para cada cliente[7]. Esta atividade tem por objetivo não apenas criar e manter uma presença firme no meio digital mas também visa otimizar e aperfeiçoar a visibilidade e interação com o público-alvo, ou seja, é uma visão ampla que tem por objetivo incrementar a presença online e fortalecer a relação com o público-alvo de cada cliente, tornando a relação entre organização e cliente amigável.

¹ Serviço caracterizado pela gestão de redes sociais.

Independentemente da área de atuação, os dados possuem uma importância significativa para o sucesso de um negócio, podendo ter diversas utilidades dentro de uma empresa, como por exemplo, utilização para calcular a previsão de faturamento, o ticket médio de um serviço e a ajuda na tomada de decisões estratégicas. No contexto do *marketing* digital esses dados são chamados de *insights* ou métricas e podem ser coletados e analisados por meio das plataformas específicas da área. Entretanto, estas atividades não são simples de serem realizadas, sendo necessário que o profissional possua habilidades que permitam que esses dados sejam localizados, extraídos e tratados de maneira ágil e eficiente.

Neste contexto, as plataformas do Facebook, Google e Instagram podem ser úteis até certo ponto, após isso tornam-se inviáveis tanto para o gerenciamento de clientes quanto para análise de dados. Levando a necessidade de buscar por ferramentas voltadas exclusivamente para o *marketing* digital disponíveis no mercado. Estas, por sua vez, são ferramentas que trabalham de forma isolada em alguma área do *marketing*, ou seja, os capacitados na área acabam tendo que contratar várias plataformas distintas, tornando o custo do serviço alto e complicando ainda mais a organização interna do trabalho.

Diante dessa situação de dificuldade de gerenciamento das redes sociais e da análise adequada dos dados dos clientes, torna-se necessário desenvolver uma alternativa de fácil acesso e uso para enfrentar esses desafios, tendo em vista que uma gestão fragmentada pode prejudicar a organização interna e por consequência pode atrapalhar no relacionamento com os clientes. Dessa forma, este trabalho se justifica em expor e resolver tais problemas, propondo um *software* que visa permitir uma melhor organização da empresa, melhorando a produtividade dos profissionais do mercado de *marketing* digital, tais como agências, *freelancers* e até mesmo influenciadores digitais.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma API que irá coletar e armazenar os dados das plataformas Facebook, Instagram e Google com o objetivo de disponibiliza-los para que os usuários possam fazer uso deles de acordo com as suas necessidades.

1.3.2 Objetivos Específicos

- Entender sobre o funcionamento e a aplicação de estratégias de *marketing* digital;
- Aprender sobre a estrutura da Graph Application Interface Programming (API) do Facebook, bem como as suas demais APIs;
- Aprimorar conhecimentos sobre engenharia de *software*;

- Permitir que profissionais de *marketing* ou de outra área, com o conhecimento necessário, possam criar suas próprias telas e relatórios através das chamadas na API;

RESUMO

Neste capítulo foi apresentada uma contextualização sobre o problema tratado neste trabalho e a justificativa de tal assunto, que pode ser resumida como sendo necessidade de uma solução de fácil acesso que permita o gerenciamento de clientes de forma eficiente e a análise de dados de forma rápida. Ao final, foram detalhados os objetivos gerais e específicos do trabalho.

Os próximos capítulos estão organizados da seguinte forma:

- **Fundamentação Teórica:** Neste capítulo são apresentados todos os conceitos teóricos utilizados no desenvolvimento da solução proposta no presente trabalho;
- **Tecnologias Utilizadas:** Neste capítulo será falado sobre todas as tecnologias que foram utilizadas para o desenvolvimento do software.
- **Modelagem do Projeto:** Capítulo dedicado a explicar sobre as características, comportamentos do software e como foi desenvolvido.
- **Software:** Capítulo dedicado a apresentação da solução implementada.
- **Conclusão:** Neste capítulo é feita a conclusão do trabalho dado seus objetivos propostos e são listados os trabalhos futuros para melhorar e/ou expandir a utilização da solução.

2 FUNDAMENTAÇÃO TEÓRICA

Para melhor entender sobre a solução que foi proposta por este trabalho, é necessário entender alguns conceitos envolvendo *marketing* que estão diretamente relacionados.

2.1 O QUE É O MARKETING DIGITAL

Pode-se dizer que o *marketing* digital é uma "evolução" do *marketing* tradicional, uma vez que mantém o objetivo principal que é o impulsionamento de vendas de serviços e/ou produtos, porém adiciona novos conceitos, dentre eles, o Custo Por Clique (CPC), *Search Engine Optimization* (SEO) e novas estratégias para obter o sucesso, sendo algumas delas o tráfego pago e o tráfego orgânico, que serão explicados detalhadamente. Dessa forma, o *marketing* digital é imprescindível para as empresas nos dias de hoje, uma vez que é responsável por estabelecer uma ponte entre a instituição e os seus clientes, permitindo então entender as necessidades e desejos do mercado e do público alvo. Entretanto, engana-se que esta atividade é caracterizada apenas por uma única estratégia que todos os profissionais utilizam, cada cliente possui estratégias ajustadas aos seus interesses. Desta forma, para compreender de maneira aprofundada sobre esta área que está em constante crescimento, é essencial conhecer sobre *marketing* tradicional e suas evoluções até os dias atuais.

Neste contexto, a origem desta área nos remete ao século XX, período em que ocorria a Segunda Revolução Industrial que era caracterizada por ser um processo de industrialização. O *marketing* era uma atividade unicamente voltada para o mercado e com o foco em impulsionar a venda e distribuição de produtos. Estava diretamente ligada a economia sendo a primeira fase de um processo evolutivo marcado por 4 fases.

Por volta dos anos 50, o *marketing mix* surgiu marcando assim a segunda fase. Esta evolução foi caracterizada pelo surgimento de um conjunto de ferramentas no qual as empresas utilizavam para atingir os objetivos desejados[8]. Dentre essas ferramentas, estão os 4P's ou também chamados de *marketing* composto. Os 4P's surgiram com o objetivo para posicionar a marca no mercado de acordo com as necessidades dos clientes em que cada "P" fala sobre uma determinada área.

A seguir será explicado sobre os 4P's do *Marketing mix*:

- **P de Preço:** É a estratégia que é responsável pela precificação do produto. Sendo essencial para o sucesso do produto, uma vez que posiciona a marca no mercado;
- **P de Produto:** Refere-se ao produto que satisfaz as necessidades dos clientes envolvendo a qualidade, design, características e os seus diferenciais. Vale ressaltar que este P também faz referências a serviços;

- **P de Promoção:** Diferente do que o nome sugere, este P é mais do que divulgar o produto. É focado em comunicar aos clientes sobre a marca despertando interesse e expectativa por parte do consumidor. Neste P, alguns conceitos já conhecidos estão associados, sendo eles a propaganda, blogs e publicidade;
- **P de Praça:** Está associado aos meios como o produto chega ao cliente podendo ser por meios físicos ou virtuais, questões como frete e logística, canais de distribuição dentre outros estão envolvidos a este P.

Posteriormente, em sua terceira fase o *marketing* passou a envolver, não somente o lucro, mas também melhorias na sociedade. Sendo entendido como uma disciplina que estudava as relações entre diversas entidades. Por fim, a última fase em que o foco era o relacionamento com os clientes com o objetivo de fidelização. Um exemplo de *marketing* tradicional nos dias de hoje, é a propaganda em revistas e televisões.

Após essas evoluções do *marketing* somadas a evolução da internet nos anos 90 que deixou ambientes isolados e foi disponibilizada para o público geral, surgiu-se uma nova forma de vender serviços ou produtos. O surgimento dessa nova área de conhecimento trouxe consigo novos conceitos, sendo 2 deles o tráfego orgânico e o pago. Esse é caracterizado por atrair clientes de forma natural, em que a empresa não paga por anúncios para captar esses clientes[9]. Ao contrário deste que é descrito por ser uma estratégia focada em atrair e prospectar clientes, utilizando meios financeiros para atingir esse objetivo[10]. Dessa forma, ao navegar por mídias sociais os usuários podem acabar indo de encontro a algo que lhes interessa, podendo ser de forma orgânica ou paga, ambas são estratégias de *marketing* digital, o que as difere é apenas a maneira como cliente chega, desse modo, esta área é definida por ser estrategicamente *on-line* para vender um produto ou serviço[11]

Ao adentrar na área do *marketing* digital, tem-se a exposição de novos conceitos, dessa forma a junção de estratégias visando atrair e prospectar novos clientes, assim como criar e estruturar uma presença nos meios *on-lines* mostra que esta área está além de vender produtos ou serviços. Para iniciar esse processo é essencial compreender o público-alvo, não tratando-se apenas de conhecer as características demográficas mas, de se aprofundar nos interesses e comportamento de cada usuário ou grupo. Uma vez entendido sobre isso, é possível criar estratégias eficazes e estabelecer uma presença nos meios *on-lines*, sendo este a expressão de identidade da marca. Trata-se de uma abordagem que envolve estratégias com o intuito de interagir com o público-alvo e assim então construir um relacionamento e consolidar a marca.

Dessa forma, para garantir uma conexão com o público é necessário ter um investimento em uma produção de conteúdo de qualidade. Este por sua vez, é focado na criação de materiais que podem ir desde postagens simples até vídeos mais elaborados com o intuito de atender as necessidades não só do público-alvo principal, mas também de atrair a atenção de novos públicos. Blogs informativos são caracterizados por ter uma grande quantidade de informações, porém

para o blog está bem posicionado nos motores de busca é necessário ter conhecimentos em SEO, que consiste em estratégias para otimizar o conteúdo para os motores com o objetivo da marca ser encontrada de forma orgânica por aqueles que buscam o que a marca oferece. Este por sua vez, possui uma vasta gama de conhecimentos técnicos que devem ser estudados para serem aplicados com máxima eficácia, como por exemplo, conhecimento sobre Meta Tags. O *marketing* digital possui uma série de abordagens que podem ser utilizadas em conjunto para alcançar os objetivos de forma ágil e eficaz.

2.2 DIFERENÇAS ENTRE MARKETING DIGITAL E TRADICIONAL

O advento do *marketing* digital não necessariamente irá fazer o tradicional desaparecer ou parar de ser utilizado e apesar de ambos possuírem os mesmo objetivos, que é impulsionar as vendas de serviços ou produtos, eles ainda possuem suas diferenças. A principal diferença entre ambos é que o *marketing* digital se utiliza de canais *on-lines* como redes sociais e blogs para alcançar seus objetivos, sendo o *e-commerce* um grande exemplo de *marketing* digital, por sua vez o tradicional faz uso de meios físicos como televisões, rádios e revistas.

Devido a esta segmentação de meios, o *marketing* digital permite que seja possível a análise de dados em tempo real, por exemplo, é possível acompanhar as métricas de uma campanha no Facebook ou Google, isso permite que seja possível ajustar as estratégias de *marketing* em tempo real[11]. Esta prática, permite que os objetivos dos clientes possam ser alcançados de maneira mais ágil, garantindo também um maior controle sobre os custos[12]. Além do mais, é possível ter uma interação mais dinâmica com os seus clientes, melhorando assim o *feedback*. Por sua vez, o *marketing* tradicional não permite isso, uma vez que faz uso do *Outbound Marketing*, este será explicado posteriormente.

Outra grande diferença é que na área do *marketing* digital é possível escolher o público que se deseja alcançar, por exemplo, é possível escolher promover um produto apenas para um grupo selecionado de pessoas, ou seja, a organização não vai atrás dos clientes, a mesma atrai o cliente por meio de presença nas redes sociais, com postagens, interações, dentre outras atividades, esta estratégia é chamada de *Inbound Marketing*. Já o *marketing* tradicional, como citado anteriormente, utiliza o *Outbound Marketing* estratégia pela qual a empresa vai atrás do cliente utilizando meios "tradicionais", como televisões e revistas, ou seja, esta estratégia é caracterizada por ser mais generalizada[12].

Apesar dessas diferenças e principalmente do *marketing* digital se mostrar superior ao tradicional, os resultados são definidos de acordo com as estratégias definidas pelos profissionais, ou seja, ambas as abordagens possuem funções e objetivos específicos. Dessa forma, fica explícito que a existência de uma abordagem não necessariamente exclui a outra, ambas podem ser trabalhadas de forma isolada mas podem ser utilizadas de forma conjunta, basta apenas o profissional ter o conhecimento e a experiência necessária para poder utilizá-las de maneira eficaz.

2.3 MÉTRICAS NO MARKETING DIGITAL: *INSIGHTS*

Para medir a lucratividade do negócio algumas medidas e cálculos são utilizados para determinar estes resultados, no *marketing* tradicional e digital não é diferente. O *Return on investments* (ROI) é uma forma de determinar o lucro em relação ao custo de investimento em *marketing*, esta métrica pode ser utilizada tanto para o digital quanto para o tradicional. Entretanto, o *marketing* digital trouxe consigo métricas exclusivas.

Dessa forma, o lançamento de campanhas para venda de serviços e/ou produtos na internet ou até mesmo o gerenciamento de redes sociais vai muito além do que acompanhar a quantidade de *leads* cadastrados na campanha ou até mesmo analisar quantidade de seguidores em um perfil ou curtidas em uma postagem, é necessário entender sobre as métricas e como elas podem ser úteis para o crescimento do negócio.

As métricas são indicadores utilizados para medir os resultados obtidos, sendo cruciais para direcionar estratégias digitais de maneira mais eficaz, entender o comportamento do público-alvo e otimizar o desempenho das campanhas *on-line*. Essa nova forma de analisar dados, permitiu uma análise mais precisa. Dessa forma, era possível saber exatamente quantos usuários acessaram um site ou aplicativo e quais foram os seus comportamentos, podendo assim melhorar a experiência do usuário[13]. Ademais, permite ainda a análise dos dados dos usuários, tais como informações demográficas, interesses, com o objetivo de criar estratégias específicas para esses usuários.

O Facebook e o Instagram disponibilizam esses *Insights* tanto através de suas APIs quanto através de suas interfaces gráficas, sendo de fácil acesso e entendimento. O Google por sua vez, também disponibiliza das mesmas formas como estes, entretanto, é necessário ter alguns entendimentos prévios para utilizar da plataforma, tais como, entendimento sobre contas, propriedades, vistas de propriedades e diferenças entre o Google Analytics 4 (GA4) e o Universal Analytics (UA), das quais a principal distinção é que o GA4 baseia seus dados em eventos enquanto o UA faz uma abordagem utilizando sessões.

2.4 O QUE É UM *MARKETING* DE SUCESSO

Segundo Aurélio, o significado da palavra sucesso é *éxito*, dessa forma, ao aplicarmos no contexto do *marketing* digital pode-se dizer que esta associado ao fato de alcançar os objetivos propostos. No entanto, como já falado anteriormente, para trabalhar nesta área é necessário ter conhecimento e experiência para saber lidar com diversas situações. Partindo de um pressuposto, um cliente possui uma pequena empresa no ramo do varejo e deseja alavancar suas vendas. Neste cenário, o cliente pode optar tanto pelo tradicional quanto pelo digital ou então por ambos.

Entendendo a importância de cada abordagem, o cliente pode optar pela utilização do *marketing* tradicional, no qual utilizaria de revistas, Tvs e rádios para impulsionar suas vendas. Entretanto, como já mostrado anteriormente, as deficiências do *marketing* tradicional em relação

ao digital pode fazer com que os resultados acabem demorando um pouco para serem notados pelo cliente, sendo esta uma abordagem a longo prazo.

Partindo para o *marketing* digital, é possível dividir em dois cenários, sendo o primeiro quando o cliente decide investir em campanhas e anúncios *on-lines*, a eficácia da estratégia definida será medida pelo ROI. Este cenário é um pouco mais desafiador, uma vez que envolve investimento, dessa forma os profissionais não podem levar muito tempo para obter os resultados. Dessa maneira, o monitoramento constante das métricas é essencial, uma vez que permite ajustar as campanhas, tornando-as mais otimizadas. Neste contexto, o sucesso será atribuído a capacidade de atrair cliente, convertê-los em vendas com o mínimo de valor gasto.

Por fim, o segundo cenário é caracterizado pelo cliente buscar impulsionar suas vendas de forma orgânica. Neste contexto, é de suma importância que o cliente possua uma presença nas redes sociais. Essa presença deve ser maximizada utilizando estratégias de SEO, *marketing* de conteúdo e principalmente o gerenciamento das redes sociais. Geralmente, este cenário não é tão complicado, uma vez que o cliente faz investimento apenas na realização dos serviços. Dessa forma, o sucesso deste cenário será atribuído ao quanto a marca conquistou de visibilidade e engajamento sem depender de investimentos financeiros.

Logo, pode-se concluir que um *marketing* de sucesso não depende exclusivamente da abordagem que é utilizada, qual meio será utilizado para alcançar esses resultados e nem qual o profissional que está realizando o trabalho, mas sim da ação em conjunto. Os cenários citados demandam de análises aprofundadas e detalhadas para que os resultados almejados pelos clientes sejam obtidos de forma rápida e econômica. Contudo, caracterizam um *marketing* de sucesso, uma vez que alcançaram os objetivos definidos em conjunto com o cliente.

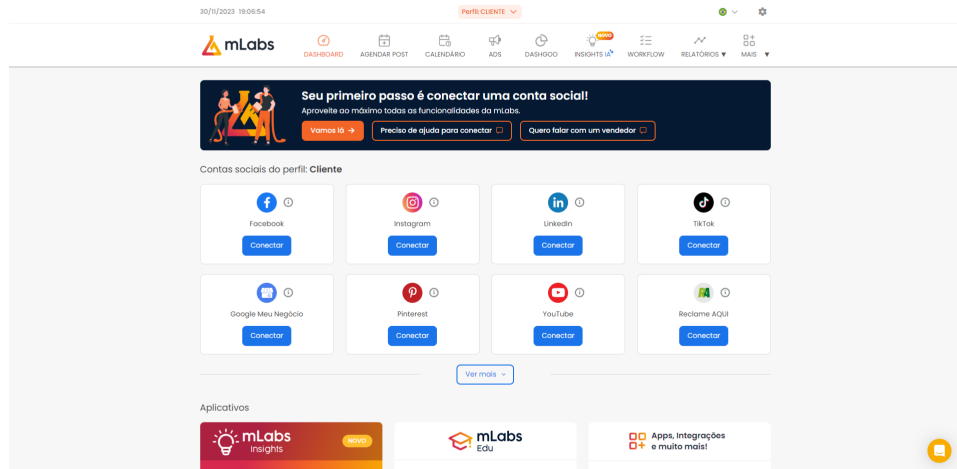
2.5 TRABALHOS RELACIONADOS

Nesta seção, falaremos sobre algumas ferramentas que estão disponíveis no mercado para profissionais de *marketing* digital. O objetivo é entender sobre o que cada plataforma faz e disponibiliza, bem como seus pontos fortes e fracos.

2.5.1 mLabs

mLabs é uma plataforma com o foco no gerenciamento de redes sociais, tais quais o Google Meu Negócio, LinkedIn, Instagram, Facebook, Twitter e Pinterest[14], centralizando-os em um único lugar. A plataforma disponibiliza diversas funcionalidades, dentre elas o agendamento de postagens, relatórios personalizados, comparação entre concorrentes no Instagram. Todas essas funcionalidades centralizadas em uma única plataforma visam melhorar a organização e produtividade dos profissionais de *marketing* digital.

Figura 1 – Captura de Tela da tela inicial do mLabs



Fonte: Elaborado pelo Autor

Principais pontos positivos:

- Permite agendamento de postagens em redes sociais;
- Permite capturar as métricas das postagens;
- Permite controlar o fluxo de criação e aprovação de postagens;
- Possui uma interface amigável e de fácil uso;
- Permite o monitoramento de concorrentes;

Deficiências:

- **Indisponibilidade de algumas métricas:** A ferramenta não permite obter métricas de campanhas e anúncios;
- **Plano com muitas limitações:** O plano mais básico da plataforma oferece uma quantidade limitada de funcionalidades onde não é possível ter uma análise completa de métricas enquanto o mais caro disponibiliza todas as funcionalidades podendo até realizar análise em cima dos concorrentes.

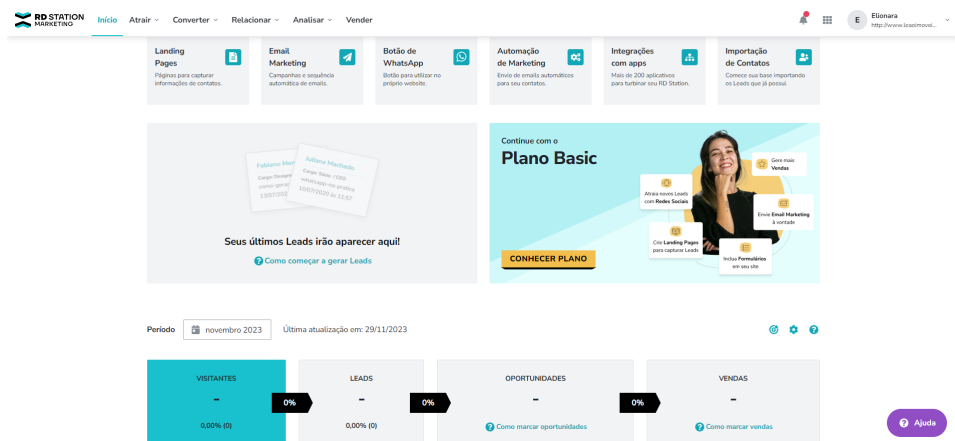
A diferença de funcionalidades disponíveis entre o plano mais básico da plataforma para o mais caro é discrepante;

- **Voltado apenas para o gerenciamento de redes sociais:** A ferramenta não permite que seja trabalhado com *leads* que são cliente em potencial, ou seja, são clientes que podem se tornar uma possível venda;

2.5.2 RD Station Marketing

RD Station Marketing é uma ferramenta em que seu principal objetivo é a automação de *marketing* digital e vendas, permitindo que o usuário possa gerenciar seus *leads*, planejar estratégias de SEO e até mesmo criar automações de e-mails.

Figura 2 – Captura de Tela da tela inicial do RD Station



Fonte: Elaborado pelo Autor

Principais pontos positivos:

- Permite criar automações, como por exemplo, o envio de e-mails;
- A criação de *landing pages* e formulários utilizando modelos já existentes;
- Permite a captura de *leads* e a organização dos mesmos da forma como desejar;

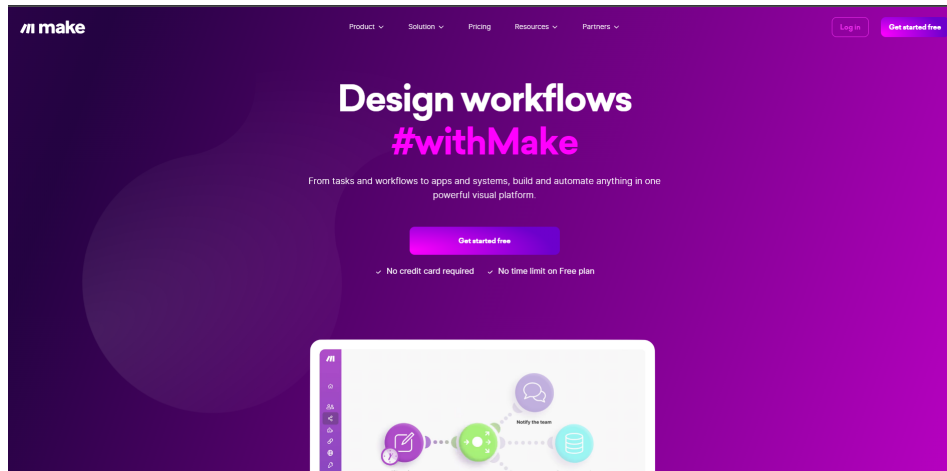
Deficiências:

- **Não trabalha com a gestão de redes sociais:** A ferramenta não permite que seja possível gerenciar as redes sociais;
- **Relatórios simples:** A ferramenta disponibiliza de relatórios porém não são tão completos quanto os relatórios disponibilizados pelas próprias plataformas como o Facebook e Instagram;

2.5.3 Make

Ferramenta que permite que seja realizada a automação de serviços, tais como, funções de *marketing* como a captura de *leads* provenientes de anúncios do Facebook, automação de equipes de Tecnologia e Informação (T.I), através do monitoramento de sistemas, documentação de softwares, dentre outros.

Figura 3 – Captura de Tela da tela inicial do Make



Fonte: Elaborado pelo Autor

Principais pontos positivos:

- Ferramenta *low code*¹, ou seja, pessoas sem conhecimentos técnicos na área do desenvolvimento de *softwares* podem realizar automações e conexões entre as redes sociais;
- A ferramenta disponibiliza várias opções para integrações, dentre eles, o Facebook e Instagram;

Deficiências:

- **Custo:** Empresas que possuem um grande volume de operações podem ter altos custos a medida que utilizam da plataforma;

RESUMO

Neste capítulo foram apresentados conceitos importantes que serviram de base para o desenvolvimento do *software* proposto, tal como foram apresentados os trabalhos relacionados.

¹ Plataforma que permite o desenvolvimento através de uma interface gráfica.

3 TECNOLOGIAS UTILIZADAS

3.1 PYTHON

Python é uma linguagem interpretada, orientada a objetos, de alto nível, que pode ser utilizada para o desenvolvimento *Web*, *Desktop*, *Mobile* e para a análise dados. Possui uma tipagem forte e dinâmica, este permite que não seja realizada a operação entre os dados sem que ocorra uma conversão explícita e esse permite que possa ser possível alterar o tipo do dado durante o tempo de execução do programa que não ocorrerá erros. Foi criado por Guido van Rossum e teve sua primeira *release*¹ em fevereiro de 1991. Atualmente encontra-se na versão 3.12.3 e é mantida pela Python Software Foundation[15].

A linguagem Python permite a criação de códigos limpos e fáceis de serem analisados e entendidos por outros profissionais. Devido a esta simplicidade em se desenvolver *softwares* o Python foi escolhido como a principal linguagem do projeto. Ademais, o Python possui uma comunidade muito vasta permitindo a interação constante entre profissionais da computação, uma documentação abundante que facilita o aprendizado e a resolução de problemas e por fim, possui bibliotecas de fácil acesso, por meio do seu gerenciador de pacotes (PIP), que aceleram o desenvolvimento de *softwares* e por consequência os potencializam. Todo o *Back-End* da aplicação, desde rotas da API até funções de utilidades genéricas foram desenvolvidas utilizando a linguagem Python.

Entretanto, existem alternativas a esta linguagem, dentre elas, o Java SE e o PHP. O Java é uma linguagem que teve seu lançamento na década de 90 e atualmente encontra-se na versão 22, sendo caracterizada por ser um linguagem orientada a objetos, fortemente tipada e estática. O PHP por sua vez, foi uma linguagem bastante utilizada no desenvolvimento *web* dos anos 2000, sendo lançada na década de 90 e atualmente na versão 8.3[16], é uma linguagem orientada a objetos, sendo fracamente tipada e dinâmica.

3.2 DJANGO

Django é um *framework*² *web* de alto nível escrito em Python que permite o desenvolvimento rápido[17]. Inicialmente, foi desenvolvido apenas para atender as necessidade de um site jornalístico tendo o seu lançamento no ano de 2005. Este *framework* adota a filosofia *Don't repeat yourself*, em que redundância é ruim para o desenvolvimento. Atualmente, encontra-se na versão 5.0.4 e é mantido pela Django Software Foundation. Django utiliza o padrão *model-template-view* (MTV), sendo este padrão semelhante ao padrão *model-view-controller* (MVC) que é bastante utilizado por *frameworks* PHP.

¹ Lançamento de uma nova versão oficial de um software.

² Conjunto de bibliotecas que agilizam o desenvolvimento do software.

A seguir, teremos uma analogia entre estes dois padrões:

- **M (MTV) = M (MVC):** São responsáveis pelo mapeamento do banco de dados, como por exemplo, criação de modelos de dados.
- **T (MTV) = V (MVC):** São responsáveis por renderizar páginas Hypertext Markup Language, versão 5 (HTML5) para os navegadores com o intuito de exibir dados para os usuários.
- **V (MTV) = C (MVC):** São responsáveis por implementar a regra de negócio do projeto, servindo como uma ponte entre as duas camadas já citadas.

Entretanto, vale resaltar que apesar de possuir este padrão, ao desenvolver com Django, não é obrigatório segui-lo a risca, podendo ficar a critério do usuário de utilizar outros padrões. Este *framework* possui uma documentação bem estruturada e completa, com tutoriais que facilitam o aprendizado e a sua utilização. Somados a uma comunidade bastante ativa, várias bibliotecas disponíveis e uma segurança fornecendo um sistema de autenticação e permissões bem robusto, tudo visando um desenvolvimento acelerado de aplicações fizeram com que este *framework* fosse escolhido para o desenvolvimento do produto proposto.

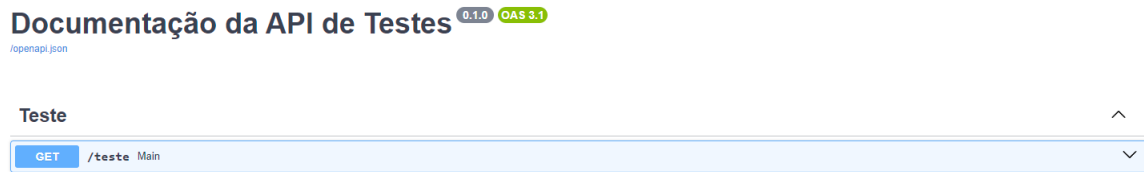
Entretanto, para se desenvolver APIs existem diversas tecnologias que podem ser utilizadas cada uma com suas características, benefícios. Dentre as alternativas existentes, três delas foram estudadas e analisadas para serem utilizadas no desenvolvimento deste projeto, sendo elas, o *framework web* **FastAPI**, o *microframework*³ **Flask** e o *framework full-stack* **Codeigniter**.

O **FastAPI** é escrito em Python e é focado no desenvolvimento de APIs de maneira rápida, disponibilizando para o desenvolvedor apenas o necessário para esta atividade, diferentemente do Django que disponibiliza uma interface administrativa através de classes Python que geram HTML5, Cascading Style Sheets (CSS) e Javascript, o FastAPI disponibiliza apenas uma interface de documentação do projeto, seguindo o padrão de especificação da OpenAPI chamado de Swagger[18], ficando a critério do desenvolvedor de customizar esta tela.

Seguem imagens de exemplos da documentação gerada pelo FastAPI:

³ Diferente do *framework full-stack*, os *microframework* são estruturas *web* totalmente simples e minimalistas, disponibilizam inicialmente apenas o necessário, como um sistema de rotas.

Figura 4 – Exemplo de documentação gerada pelo framework FastAPI



Fonte: Elaborado pelo Autor

Figura 5 – Exemplo de documentação gerada pelo framework FastAPI que detalha uma rota específica



Fonte: Elaborado pelo Autor

De forma detalhada, a primeira imagem mostra que ao gerar a documentação é possível ter acesso as rotas da API, no qual é exibido o caminho da rota e o seu método Hypertext Transfer Protocol (HTTP). A segunda imagem expõe de forma detalhada a rota que deseja visualizar, podendo observar os retornos que esta rota pode ter, seus *status code*, parâmetros dentre outras características. Vale resaltar que o desenvolvedor pode customizar as rotas, podendo adicionar novos retornos aos exemplos, novos argumentos dentre outras funcionalidades, tudo através do código sem precisar utilizar tecnologias *front-end*.

O **Flask** por sua vez, também é escrito em Python sendo bastante utilizado para se desenvolver microsserviços, ou seja, aplicações pequenas e específicas. Diferente do *framework* FastAPI, o Flask não gera a documentação que fica a critério do desenvolvedor de utilizar uma biblioteca externa para ter acesso a esta funcionalidade.

Por fim, o *framework full-stack* escrito em PHP, **Codeigniter** que trabalha de forma semelhante ao Django, porém utilizando o padrão MVC. Entretanto, o Django possui algumas vantagens em comparação a este *framework*, e os principais pontos são a interface administrativa e o Object-Relational Mapping (ORM) que o Django fornece ambos por padrão e o Codeigniter não. Em suma, para o desenvolvimento deste projeto, o Django se mostrou ser superior aos *frameworks* que foram considerados como alternativas.

3.3 DJANGO REST FRAMEWORK

Diferente do Django, o Django REST framework (DRF) não é um *framework* e sim uma biblioteca Python que estende algumas funcionalidades padrões do Django e adiciona novas que permite construir APIs de forma rápida e prática[19], por exemplo, o DRF adiciona novos conceitos ao desenvolvimento Django, conceitos conhecidos como *Viewsets* que permitem a implementação dos métodos HTTP com poucas linhas de código. Os principais e mais utilizados métodos HTTP são o *GET* para se buscar informações no servidor, o *POST* para enviar dados para o servidor com o intuito de criar um novo recurso, o *PUT* para criar um novo recurso ou substituir um que já existe e o *DELETE* para se deletar algum recurso do servidor. Estes métodos visam facilitar a comunicação entre os sistemas.

As imagens a seguir, mostram exemplos de como o DRF permite criar pontos de extremidades com facilidades:

Figura 6 – Exemplo de criação de pontos de extremidade com todos os métodos

```
class TesteViewSet(ModelViewSet):  
    queryset = Teste.objects.all()  
    serializer = TesteSerializer
```

Fonte: Elaborado pelo Autor

De forma detalhada, a imagem expõe uma classe chamada *TesteViewSet* que herda de uma outra classe do DRF chamada de *ModelViewSet*, esta classe permite implementar todos os métodos HTTP de maneira prática. Na segunda linha, é feito uma busca por todos os objetos do modelo de dados *Teste* para se utilizar em outros métodos HTTP, como o *PUT*, *DELETE* e *GET*. Enfim, é definido a classe que irá ser utilizada como *serializer*⁴ para os dados.

⁴ Processo de tradução de dados, para que os mesmos possam ser recuperados e utilizados por outros sistemas.

Figura 7 – Exemplo de criação de pontos de extremidade somente com os métodos desejados

```
class TesteViewSet(ModelViewSet):  
    queryset = Teste.objects.all()  
    serializer = TesteSerializer  
    http_method_names = ['get', 'post', 'put']
```

Fonte: Elaborado pelo Autor

Já a imagem acima, mostra que se o usuário não deseja implementar algum método HTTP, basta utilizar o atributo `http_method_names` e inserir os métodos que se deseja utilizar. Existem outras abordagens para caso não se queira implementar algum dos métodos padrões, uma dessas formas consiste em fazer a herança de classes que representam cada método HTTP, por exemplo, se deseja implementar somente o método *GET*, o usuário pode optar por fazer a herança somente da classe que implementa este método.

Figura 8 – Exemplo de criação de pontos de extremidade somente com os métodos desejados

```
class TesteViewSet(ListModelMixin, GenericViewSet):  
    queryset = Teste.objects.all()  
    serializer = TesteSerializer
```

Fonte: Elaborado pelo Autor

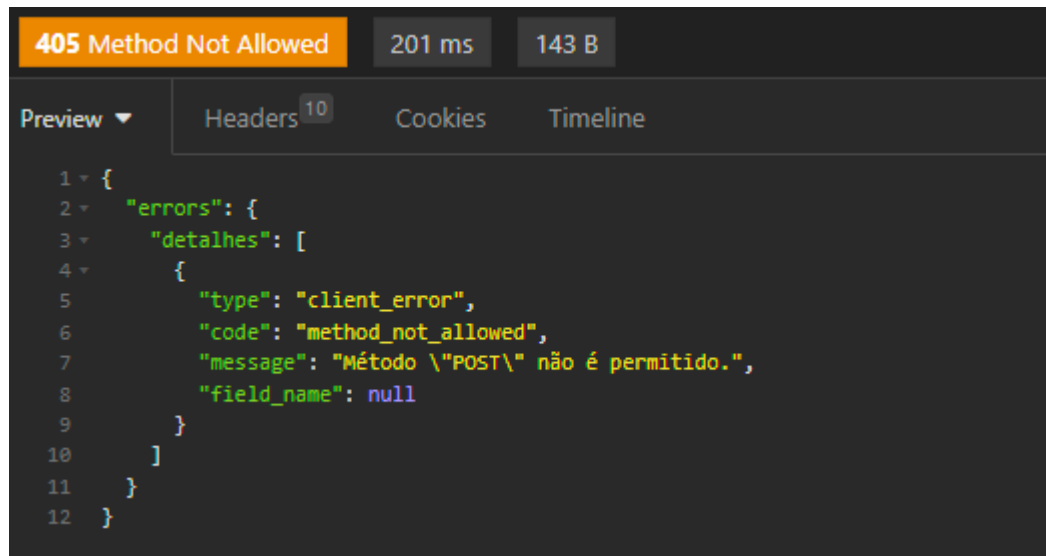
Outra abordagem é implementar o método que não deseja fazendo com que o mesmo sempre que for acessado retorne um erro ao usuário. Geralmente retorna o status HTTP 405, conhecido como *Method Not Allowed*.

Figura 9 – Exemplo de criação de pontos de extremidade somente com os métodos desejados

```
class TesteViewSet(ModelViewSet):  
    queryset = Teste.objects.all()  
    serializer = TesteSerializer  
  
    def delete(self, request, *args, **kwargs):  
        raise MethodNotAllowed(request.method)
```

Fonte: Elaborado pelo Autor

Figura 10 – Exemplo de criação de pontos de extremidade somente com os métodos desejados



Delete (CRUD) de tarefas agendadas, podendo ser possível realizar esta ação através do painel administrativo do Django[20]. Dessa maneira, o agendamento de tarefas não fica exclusivo do código.

3.4.1 Flower

É uma aplicação *Web* utilizada para monitorar em tempo real os eventos do Celery, a título de exemplo, visualizar o progresso e o histórico da tarefa, visualização dos detalhes da tarefa, permite também visualizar estatísticas do *broker*[21]. Além do mais, esta aplicação disponibiliza uma API e permite que seja possível realizar integração com o Prometheus, sendo este uma ferramenta para monitoramento, e o Grafana, este por sua vez é uma plataforma que permite a visualização de dados e métricas por meio de gráficos.

3.4.2 RabbitMQ

Como já falado, um dos componentes principais envolvidos no processo de tarefas assíncronas, é o *broker*, para este projeto foi escolhido o RabbitMQ, sendo um *software* de código aberto lançado no ano de 2007 e que trabalha com o protocolo *Advanced Message Queuing Protocol* (AMQP). Deste modo, o RabbitMQ foi utilizado neste projeto para garantir a comunicação assíncrona rápida. Uma alternativa viável a este *software*, é o Redis que será detalhado e explicado posteriormente foi utilizado neste projeto.

3.5 POSTGRESQL

PostgreSQL é um Sistema Gerenciador de Banco de Dados (SGBD) relacional de código aberto que utiliza a linguagem *Structured Query Language* (SQL). Teve sua origem por volta dos anos 1986 como parte de um projeto da Universidade de Califórnia[22], atualmente encontra-se na versão 16.2. Este banco de dados foi utilizado para armazenar todos os dados da aplicação e por trabalhar com objetos relacionais, poderia ter-se utilizado de outros banco de dados, tendo como exemplo, o MySQL, SQLite e o SQL Server.

Entretanto, o Postgre possui algumas características que se sobressaem em relação aos demais SGDB:

- **Segurança:** O PostgreSQL é uma SGBD totalmente transacional, ou seja, no momento em que uma transação é iniciada, todas suas atualizações são armazenadas em disco até que seja finalizada;
- **Suporte a tipos de dados:** O PostgreSQL, oferece um suporte a diversos tipos de dados modernos, como o Javascript Object Notation (JSON) e Extensible Markup Language (XML). Essa característica permitiu que o desenvolvimento do projeto fosse acelerado, uma vez que alguns dados não foi necessário criar modelos de banco de dados, foram criados apenas um campo que armazena estes valores em JSON;

- **Django:** O Django possui uma ótima relação com o PostgreSQL, visto que é disponibilizado uma série de funcionalidades que só é possível de se utilizar com este SGBD, sendo uma delas o *ArrayField*⁵[23].

3.6 REDIS

Redis é um banco de dados não relacional (NOSQL) de código aberto que armazena os valores no formato de chave-valor diretamente na memória RAM oferecendo alta performance[24], atualmente encontra-se na versão 7.2.4. Neste projeto, o Redis foi utilizado para armazenar os resultados das execuções das tarefas periódicas. Por padrão, o Django fornece um sistema de cache conhecido como *Local-memory caching*, entretanto, o Redis permite a recuperação de dados caso o servidor seja reiniciado enquanto o *Local-memory* não. Por fim, o Redis possui diversos casos de uso, como por exemplo, poderia ser utilizado como alternativa ao RabbitMQ como um *broker* e até mesmo para gerenciar sessões em jogos, ou lojas eletrônicas.

3.7 VULTR OBJECT STORAGE

Object Storage da Vultr⁶ é um serviço de armazenamento em nuvem de arquivos. Os casos de uso mais comuns são o armazenamento de arquivos estáticos, como Javascript, imagens, CSS, automação de *backups* e criação de soluções personalizadas através de sua API. Este serviço é compatível com o S3 Storage da *Amazon Web Service* (AWS) e trabalha de forma semelhante possuindo *buckets*, sendo esse similar aos diretórios em sistemas de arquivos onde estes serão armazenados, podendo ter acesso público ou privado[25]. Através de tarefas agendadas, neste projeto, este serviço foi utilizado para armazenar os *backups* do banco de dados, em que sua execução não interfere na experiência do usuário, um vez que é realizada em segundo plano.

Como alternativas a este serviço, tem-se a disponibilidade do próprio sistema de arquivos do aplicativo, entretanto, esta é uma solução que pode se tornar inviável a medida que a quantidade de dados vai crescendo. Ademais, o próprio S3 Storage da AWS, contudo, o Vultr Object Storage disponibiliza uma grande quantidade de espaço por um preço acessível ao passo que o S3 Storage não.

3.8 GIT

O Git é um sistema de código aberto que permite o controle de versões que trabalha com repositórios em que cada um deste é um histórico completo de alterações no código, atualmente encontra-se na versão 2.45.0. Sua história teve origem quando o projeto do *kernel* do Linux começou a utilizar um software de versionamento chamado BitKeeper, entretanto, este software passou a ser pago, dessa forma Linus Torvalds, criador do Linux, junto com a comunidade Linux,

⁵ ArrayField é um campo de models que permite armazenar um array de dados no banco de dados.

⁶ Empresa que fornece serviços de computação em nuvem

desenvolveram uma nova ferramenta para ser alternativa ao BitKeeper, surgindo assim o Git, tendo seu lançamento em 19 de abril de 2005[26].

O Git possui diversos comandos, dentre alguns deles que são comumente utilizados estão:

- **git-init:** Este comando permite inicializar ou reinicializar, se já existir localmente, um repositório git[27];
- **git-add:** Adiciona os arquivos modificados ao índice[27];
- **git-clone:** Permite copiar um repositório remoto dentro de um diretório[27];
- **git-commit:** Este comando é responsável por registrar as mudanças no repositório[27];
- **git-push:** Realiza a atualização do repositório remoto com base no repositório local[27];

3.9 DOCKER

Docker é uma plataforma de código aberto que permite a criação e o gerenciamento ambientes isolados, à vista disso, o código e suas dependências são empacotados e transformados em uma imagem Docker, tornando-se portátil com qualquer máquina que possua este aplicativo instalado. Outro conceito importante é de *Containers*, que são basicamente, ambientes de execução com todo o código, componentes e dependências necessárias para executar o código da aplicação. Este software teve sua primeira versão lançada em 2013 e atualmente encontra-se na versão 25.0.4. Para este projeto, foi-se utilizado com o objetivo de facilitar o desenvolvimento local, sem que seja necessário realizar o *download* e a configuração de diversos serviços, tais como PostgreSQL, RabbitMQ e Redis, e posteriormente o deploy do projeto com o mesmo objetivo, sem que precise demandar tempo para configuração de serviços.

4 MODELAGEM DO PROJETO

Neste capítulo iremos abordar alguns conceitos da engenharia de *software* para explicar o processo de desenvolvimento, tais como características e comportamentos da aplicação.

4.1 REQUISITOS

Os requisitos fazem parte do processo de desenvolvimento de software, sendo tratados como uma sub-área da engenharia de *software*, chamada de engenharia de requisitos. O levantamento de requisitos é importante para entender as necessidades iniciais do cliente e são muitas vezes de forma sistemática, envolvendo áreas como documentação, negociação e coleta de dados[28]. Dessa forma, os requisitos de *software* são descrições detalhadas das funcionalidades, comportamentos e restrições que uma aplicação deve cumprir. Ademais, ajudam a definir o que o aplicativo deve fazer.

Uma vez coletados, os requisitos devem ser documentados e validados, para esta atividade existem diversos modelos de documentação de requisitos, ficando a critério do profissional que esta realizando esta coleta. Na etapa de validação é conferido a viabilidade dos requisitos, se são de fato necessários e úteis para o desenvolvimento como também se podem ser implementados do jeito que o cliente deseja. Dessa maneira, a organização pode escolher sua forma de categorizar os requisitos, a mais generalista é a de requisitos funcionais e não funcionais[28].

Os requisitos funcionais descrevem as funcionalidades que o *software* deve possuir. Sendo esta uma forma de explicar para o time de desenvolvimento o que cliente deseja de maneira formal. Os requisitos não funcionais abordam sobre as qualidade e características do sistema como desempenho, usabilidade, segurança, confiabilidade dentre outros aspectos. Estes requisitos podem ou não ser especificados pelo cliente, exemplificando, o cliente deseja que seu sistema tenha um bom desempenho para não atrapalhar a experiência do usuário, ou até mesmo desejar que o sistema esteja disponível apenas na internet.

Para o projeto Marketing Analytics, os requisitos foram definidos utilizando conceitos da metodologia SCRUM como base. Sendo assim, foi atribuído um número de 1 a 8 que indica o nível de complexidade para o seu desenvolvimento e também foram divididos em "**Baixa prioridade**", "**Média prioridade**" e "**Alta prioridade**". Ao todo, foram identificados e coletados um total de 50 requisitos funcionais.

A tabela abaixo mostra como foi distribuído as prioridades e níveis de complexidade para os requisitos.

Análise dos Requisitos do Projeto			
Níveis/Prioridade	Baixa	Média	Alta
1	1	0	4
2	0	8	1
3	1	3	21
5	0	1	7
8	0	0	3

O detalhamento dos requisitos está disponível no Anexo B - Detalhamento dos Requisitos.

4.2 ENTENDENDO A ESTRUTURA BASE

Antes de entender sobre a arquitetura do Marketing Analytics, é necessário compreender sobre a arquitetura do Facebook, Instagram e Google. Desse modo, a Graph API do Facebook é a estrutura raiz que representa os dados da plataforma em forma de grafo social, ou seja, é composta por nós, bordas e campos. Sendo aquele usado para obter informações de um objeto específico com um identificador único, este é utilizado para se obter coleções de dados associadas a um único objeto, por exemplo, realizar a consulta de fotos associadas a um perfil de usuário e esse é utilizado para se obter dados sobre um único objeto ou sobre coleções de dados, tendo como, obter a descrição de cada foto associada ao perfil de usuário. É uma API baseada em HTTP e todos os pontos de extremidade exigem Hyper Text Transfer Protocol Secure (HTTPS). Contudo, para entender como esta e outras APIs do Meta se relacionam com este trabalho é necessário entender alguns conceitos que estão relacionados.

Seguem abaixo alguns desses significados:

- **Aplicativo:** Para utilizar os recursos da API é necessário criar um Aplicativo dentro da plataforma do Facebook, no geral esse aplicativo fornece algumas permissões básicas.
- **Tokens:** Os Tokens permitem que o Aplicativo acesse as APIs. Por padrão, todos tem duração de 1 hora, podendo ser estendida. Os Tokens são divididos em 5 tipos diferentes, porém, para este trabalho foram utilizados apenas 3.

Seguem abaixo as especificações:

- **Token de acesso do aplicativo:** Este token é utilizado para se ter acesso as configurações do aplicativo.
- **Token de acesso à Página:** Este token é necessário quando se quer ter acesso aos dados das páginas do Facebook, como por exemplo, *insights* de uma página.
- **Token de acesso do usuário:** Este token é bastante utilizado neste trabalho, uma vez que executa ações com base na entrada do usuário.
- **Modos do Aplicativo:** O aplicativo é dividido em dois modos, publicado e desenvolvimento e a diferença entre os dois se dá basicamente nas permissões em que o modo

publicado só pode solicitar permissões por meio de análises do aplicativo enquanto, o desenvolvimento só podem solicitar permissões de usuários com uma função e apenas permissões que tenham níveis de acesso padrão ou avançado.

Vale ressaltar que as APIs disponibilizadas pelo Meta, podem ter diversos casos de uso, do quais podem ser necessário outros tipos de tokens que não foram citados. Dessa forma, para este trabalho entendido estes conceitos é necessário realizar solicitações nos pontos de extremidade da API, no qual todos exigem autenticação. Desse modo, autenticação e autorização garantem a segurança e a privacidade, haja vista que os desenvolvedores precisam obter tokens de acesso e definir as permissões para obter os dados. No geral, esta API é mais ampla e fornece acesso a dados e recursos gerais do Facebook[29]. Como forma de testar a API, o Facebook disponibiliza o Explorador da Graph API, uma ferramenta que pode ser acessada através da interface do aplicativo e pode ser utilizada para gerar Tokens de acesso, obter códigos de consultas e pode realizar consulta nas APIs disponíveis.

4.2.1 Facebook Pages API

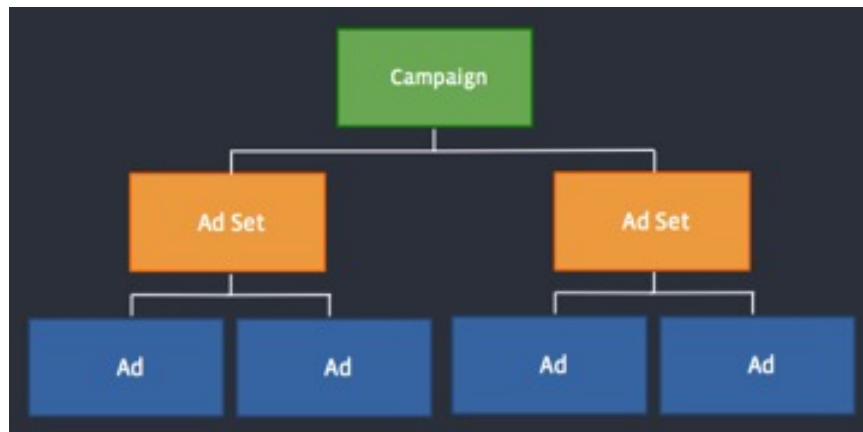
A API de Páginas é mais específica, pois é voltada para gerenciamento de acesso aos dados de páginas do Facebook. Concede aos desenvolvedores a possibilidade de criar aplicativos que podem realizar diversas ações relacionadas as páginas do Facebook, tais como o gerenciamento de conteúdo, possibilitando que os desenvolvedores possam criar, editar e excluir postagens nas páginas por meio desta API. Sendo bastante útil para automatizar o agendamento de postagens e garantir que o conteúdo seja publicado no momento certo. Ademais, esta API fornece acesso a métricas de desempenho das páginas, incluindo estatísticas de alcance, envolvimento do público e interações.

Em resumo, esta API é uma extensão da Graph API que se concentra especificamente no gerenciamento e na automação de ações relacionadas às páginas no Facebook. Vale ressaltar, que para ter acesso as dados das páginas é necessário utilizar o token de acesso a página, caso contrário será retornado erro de permissão. Apesar de possuir diversos casos de uso, neste projeto, esta API foi utilizada como forma de estabelecer a conexão inicial com o Facebook e de obter dados de páginas do Facebook, tais como, *insights* das páginas, postagens e *stories*. Dessa forma, esta API serve como base para ter acesso a outras APIs e funcionalidades disponíveis.

4.2.2 Marketing API

A API de *Marketing* é voltada especificamente para gerenciamento de dados de Campanhas, Anúncios e Conjuntos de Anúncios, na qual a mesma possui sua estrutura como mostra a imagem a seguir:

Figura 11 – Estrutura da Marketing API



Fonte: <https://developers.facebook.com/docs/marketing-api/campaign-structure>

Dessa forma, as campanhas são o mais alto nível da estrutura e representam os objetivos dos clientes, dos quais podemos citar o reconhecimento de marca que tem por finalidade alcançar o maior número de pessoas, sendo este objetivo bastante importante para empresas novas no mercado. Outro objetivo associado as campanhas é o engajamento, este tem por finalidade de encontrar pessoas que tem altas chances de interagir com o conteúdo do cliente[30]. Os Ad Sets são caracterizados por serem um conjunto de anúncios e são utilizados para configurar os orçamentos e períodos de exibição dos anúncios. As campanhas são o mais alto nível da estrutura e representam o objetivo do cliente. Por fim, tem-se os anúncios que são objetos responsáveis por conter todas as informações necessárias para serem exibidos no Facebook.

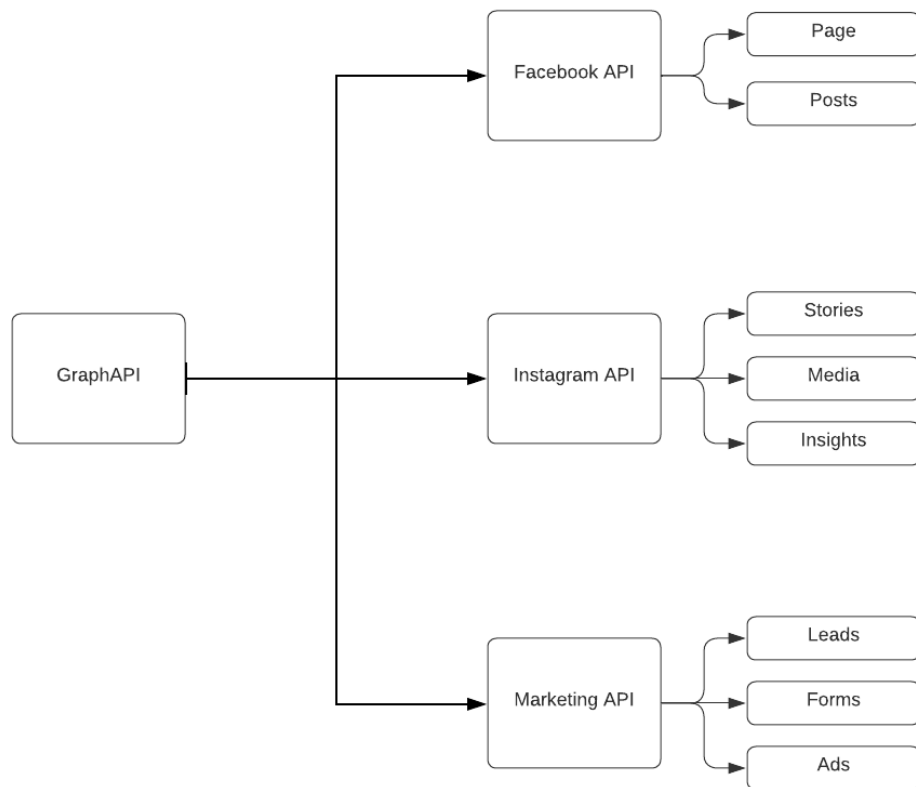
Por conseguinte, o relacionamento da Graph API com a API de Marketing se dá pela seguinte forma, API de *Marketing* pode usar a Graph API para acessar informações de perfil de usuários e direcionar campanhas publicitárias. No entanto, cada uma delas tem seu foco e finalidade distintas, atendendo a diferentes necessidades de desenvolvedores e anunciantes na plataforma do Facebook.

4.2.3 Instagram API

Semelhante a API de *marketing* e de *paginas*, esta também é específica, sendo voltada apenas para o gerenciamento de dados do Instagram. Para utilizar deste recurso é necessário que as contas do Instagram estejam associadas as Páginas do Facebook. Além do mais, esta API permite obter dados de perfil dos usuários, mídias e *stories* postados e permite a publicação de vídeos e fotos[31]. Para acessar os dados desta API, pode-se utilizar tanto o token de acesso a página quanto o token de acesso do usuário. Semelhante as outras APIs, esta também é relacionada com a Graph API, porém ambas tem finalidades diferentes.

4.2.4 Representação da relação da Graph API com as demais APIs

Figura 12 – Como funciona a Graph API



Fonte: Elaborado pelo Autor

Esta representação demonstra de forma simples o relacionamento entre as APIs. Através da Graph API é possível obter dados gerais das demais APIs, porém dados específicos, como por exemplo, foto de perfil da página do Facebook, exigem que sejam utilizadas além do Token adequado, a API específica.

4.2.5 Webhooks

Ao trabalhar com plataformas diretamente dependentes, é necessário que ambas estejam de fato sincronizadas. Diante disso, as plataformas disponibilizam de gatilhos também conhecidos como *Webhooks*, que são chamados por meio de algum evento. A definição de evento, neste contexto é quando ocorre a troca de informações entre sistemas.

Segundo Cotovicz[32], "Os webhooks são retornos de chamadas HTTP via POST¹ criados para expandir comportamentos de um aplicativo ou página web". O Facebook disponibiliza

¹ Método HTTP utilizado quando se quer enviar dados para o servidor com o intuito de se criar um novo registro.

uma série de *Webhooks* que podem ser utilizados para potencializar o produto. Para utilizar essa funcionalidade, deve-se criar um ponto de extremidade que suporte HTTPS, conceder permissões ao aplicativo, mudar para modo de publicado e configurar um ponto de extremidade, feito isso será possível receber chamadas no *endpoint* informado ao Facebook.

Como já falado, os *Webhooks* são utilizados para incrementar a qualidade do produto, uma vez que pode-se dizer que ao trabalhar com este recurso está se trabalhando com dados em tempo real. Apesar dessa importância e da vasta quantidade de gatilhos disponíveis, neste trabalho pouco foi usado desse recurso devido também a plataforma não disponibilizar de gatilhos essenciais para este produto, tendo que recorrer a alternativas para sanar esta deficiência.

4.3 DIAGRAMAS DE CASOS DE USO

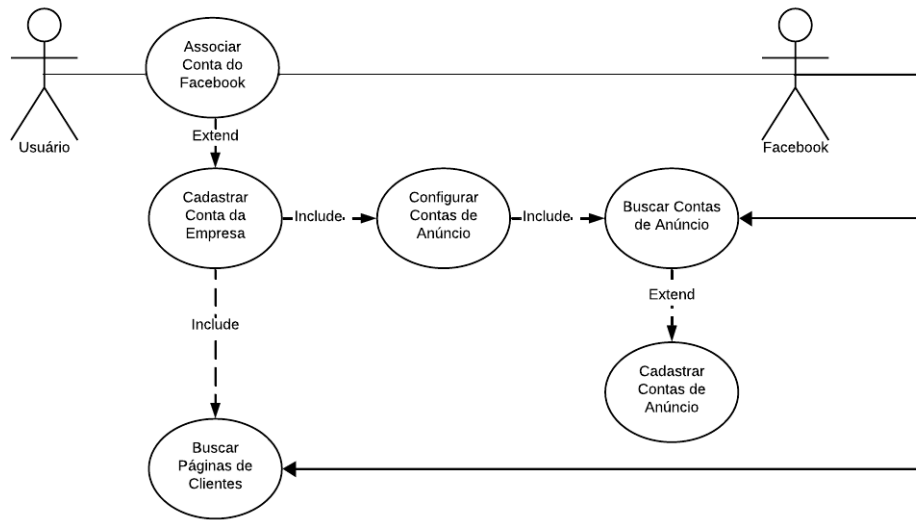
Esta técnica é utilizada para representar as interações dos atores com o sistema e o relacionamento das funcionalidades entre si. No contexto de engenharia de requisitos, assim como diagrama de classes, ela pode ser utilizada para validar requisitos e fluxos de funcionamento. Entretanto, esta técnica é mais do que apenas desenhar bonecos e bolhas, a especificação dos casos de uso garante que seja explicado passo a passo de como o usuário irá realizar tal caso de uso. Ainda nas especificações, são expostos possíveis fluxos alternativos e de erros.

Apesar desta aplicação possuir diversos casos de uso, apenas 4 deles serão explicados e detalhados, pois através destes que serão citados é possível realizar a configuração padrão do sistema. Sendo assim, é possível fazer o uso do aplicativo por completo e os demais casos de uso referem-se, geralmente, a CRUD de entidades no banco de dados.

Abaixo estão a listagem dos casos de uso selecionados:

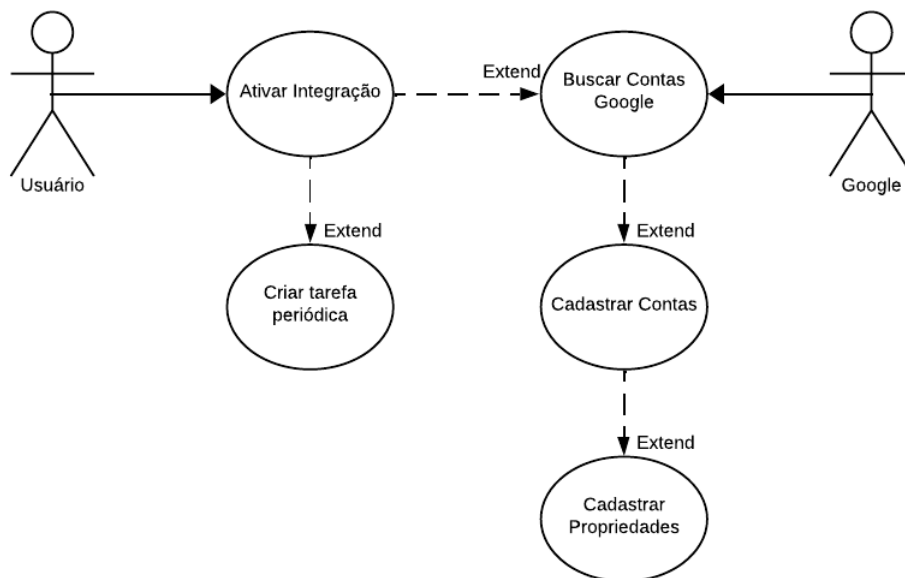
- **Caso de Uso 01 - Associar Conta do Facebook:** Para ter acesso as funcionalidades envolvendo o Facebook e Instagram, é necessário que seja feito a associação da conta da empresa de *marketing*;
- **Caso de Uso 02 - Ativar integração com o Google:** Para ter acesso as funcionalidades relacionadas ao Google, é necessário que seja ativado a integração com o mesmo.
- **Caso de Uso 03 - Executar tarefas periódicas:** É importante para o usuário ter uma maneira de poder executar tarefas em segundo plano, sem necessariamente prejudicar a sua experiência;
- **Caso de Uso 04 - Cadastro de páginas do Facebook:** Para se ter acesso aos dados dos clientes, é necessário que seja realizado o cadastro destas páginas no sistema.

Figura 13 – Diagrama UML de Casos de Uso 01 do Marketing Analytics



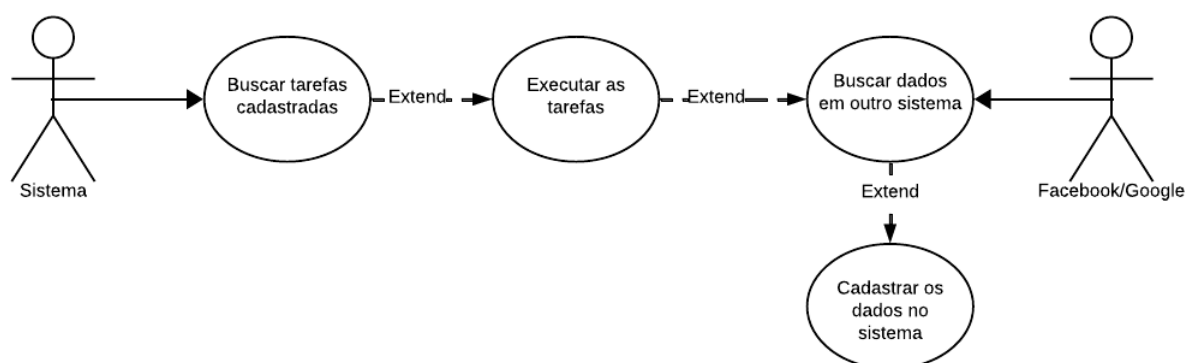
Fonte: Elaborado pelo Autor

Figura 14 – Diagrama UML de Casos de Uso 02 do Marketing Analytics



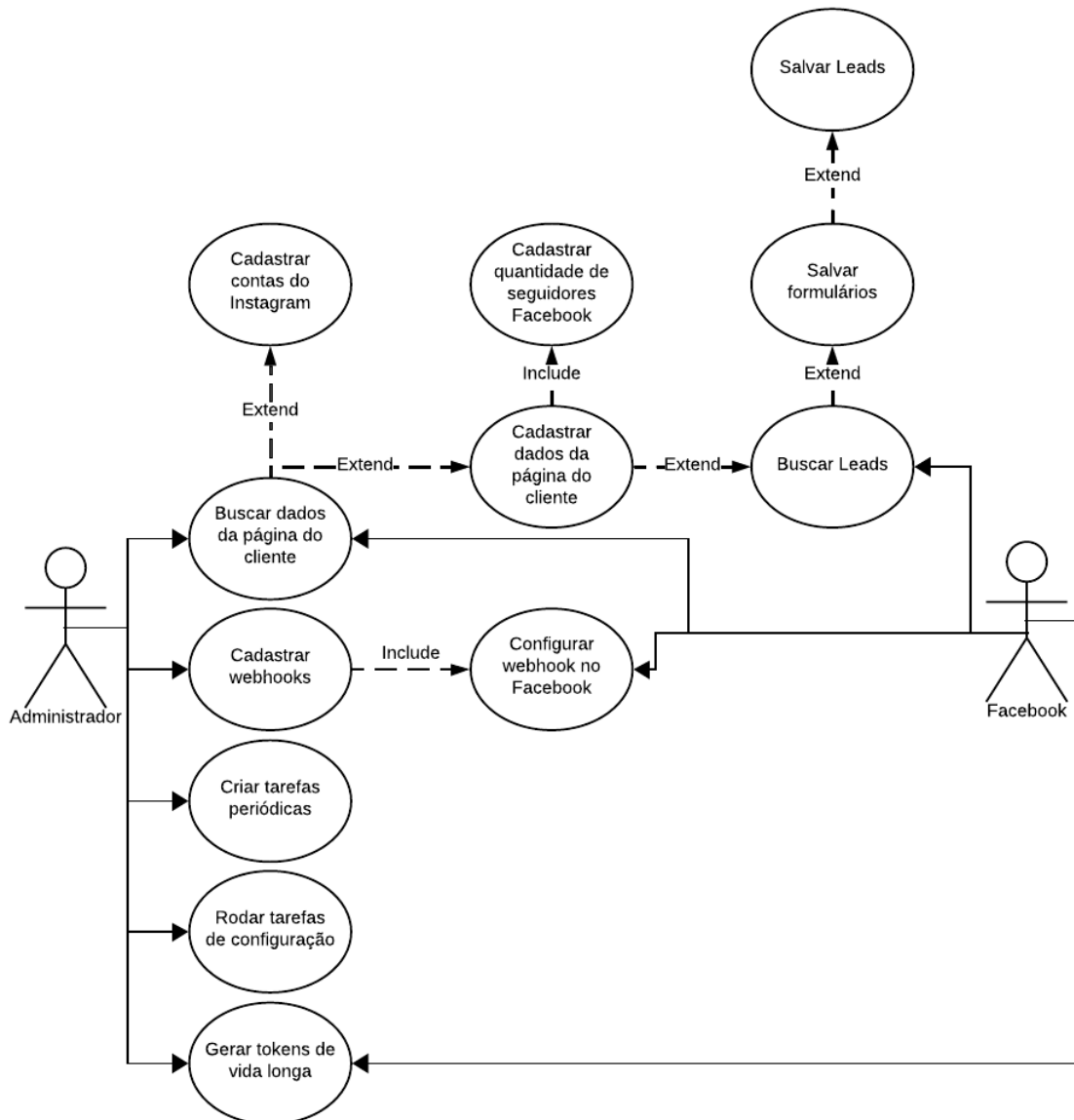
Fonte: Elaborado pelo Autor

Figura 15 – Diagrama UML de Casos de Uso 03 do Marketing Analytics



Fonte: Elaborado pelo Autor

Figura 16 – Diagrama UML de Casos de Uso 04 do Marketing Analytics

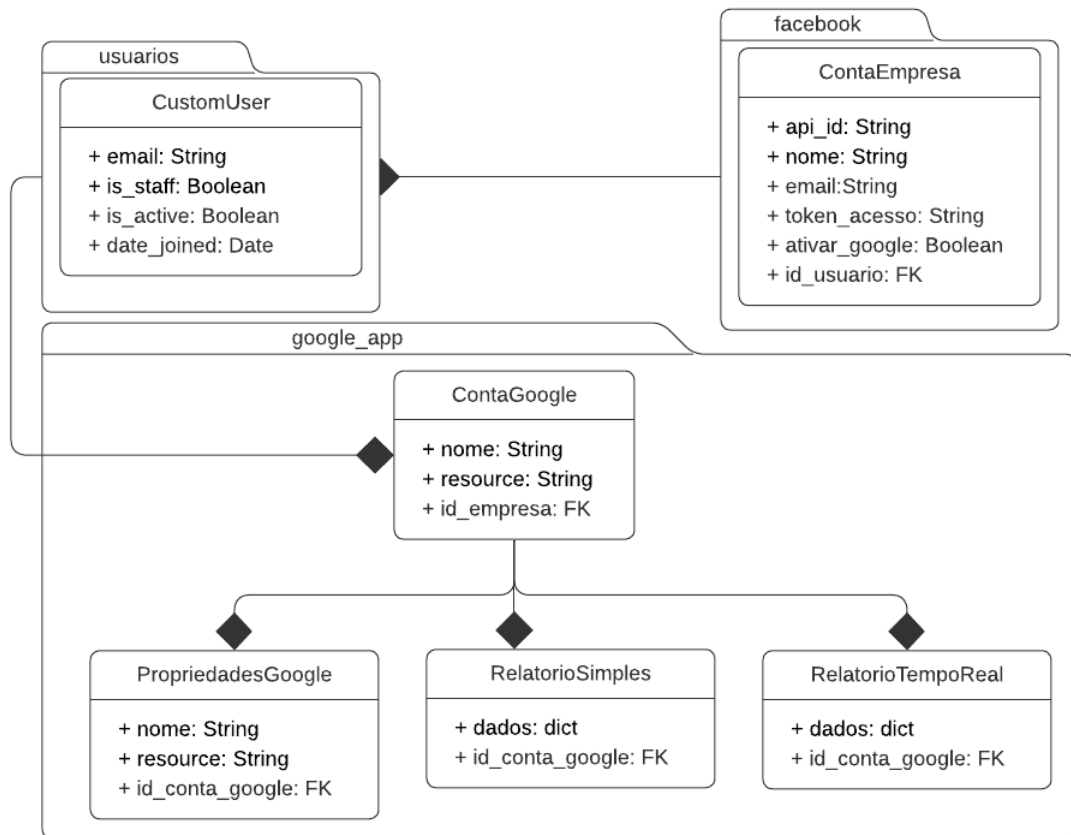


Fonte: Elaborado pelo Autor

4.4 DIAGRAMAS DE CLASSE

São componentes essenciais no desenvolvimento de *software* pois podem ajudar a compreender os requisitos do sistema. Porém, para desenhar um diagrama de classes é necessário identificar as classes do sistema, determinar o relacionamento de cada classe e por fim criar a estrutura. Este projeto possui diversas classes que poderiam ser utilizadas para este diagrama, todavia foi escolhido as classes que estão associadas ao modelos de dados do *Google Analytics* conforme demonstra a imagem abaixo:

Figura 17 – Diagrama de Classes



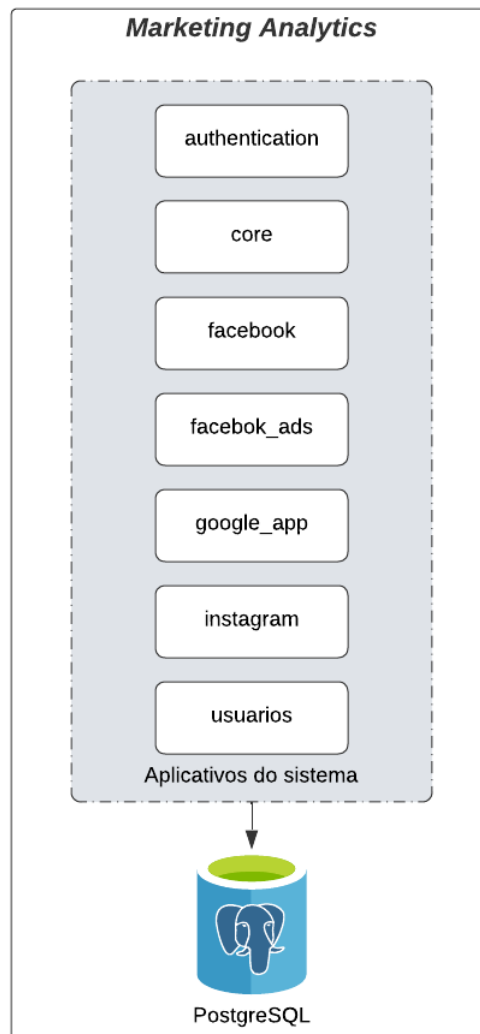
Fonte: Elaborado pelo Autor

4.5 ARQUITETURA DO SISTEMA

Ao falar sobre arquitetura de sistemas, entende-se sobre o projeto no mais alto nível, falando sobre aplicativos, microserviços, camadas de abstrações, dentre outros[33]. Quando se trata do *framework* Django[17], é possível ver como este conceito é aplicado por meio da arquitetura MTV, já falado anteriormente, sendo este a arquitetura principal de uma aplicação que se utiliza deste *framework*. Contudo, este projeto utiliza de outras duas arquiteturas, a Monolítica e a *REST*.

A arquitetura monolítica consiste em todo o sistema ser desenvolvido em uma "única base de códigos", dessa forma as telas do usuário, a lógica da regra de negócio e até mesmo conexão e criação de tabelas no banco de dados, é toda realizada em um único código sob o mesmo processo de execução. Esta arquitetura, facilita o desenvolvimento inicial e a implementação de aplicações.

Figura 18 – Arquitetura monolítica



Fonte: Elaborado pelo Autor

Como mostra a imagem acima, o sistema possui 7 aplicativos, em que cada um deles tem uma responsabilidade:

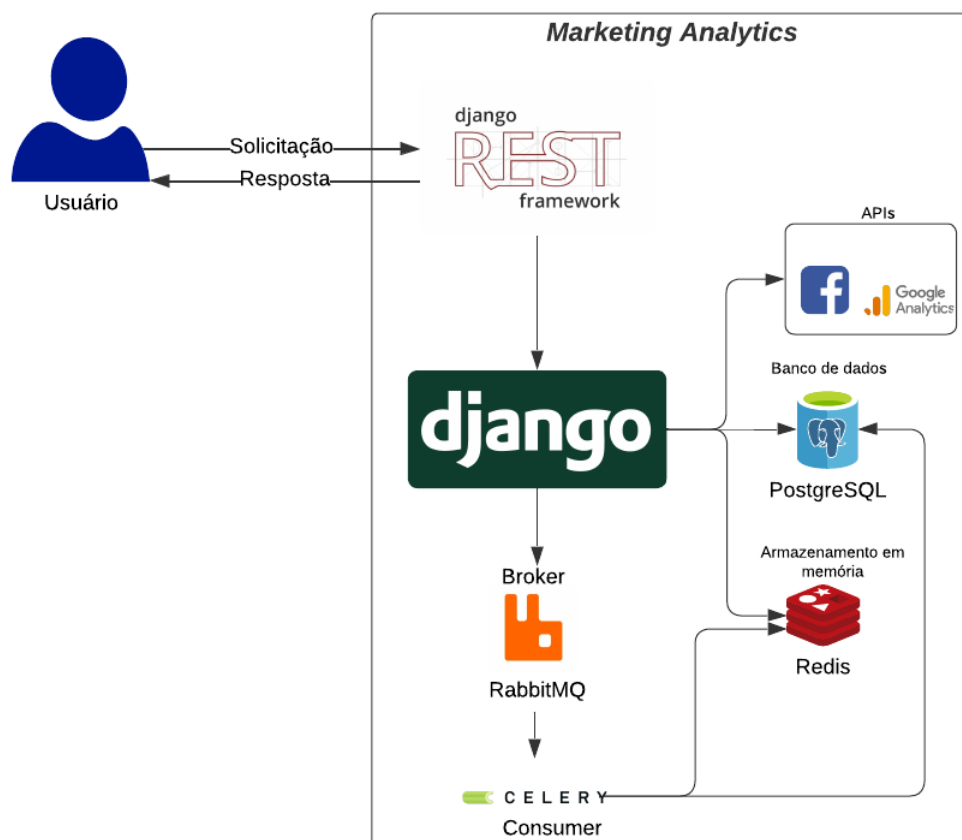
- **authentication:** Este aplicativo é responsável por toda a lógica envolvendo autenticação no sistema, por exemplo, login;
- **core:** Este aplicativo é voltado para funcionalidades genéricas do sistema, tendo por exemplo, a definição de metas para usuários;
- **facebook:** Este aplicativo é responsável por conter a lógica que está associada a busca e exibição de dados provenientes de API de páginas do Facebook;

- **facebook_ads:** Este aplicativo é responsável por conter a lógica que está associada a busca e exibição de dados provenientes de API de *marketing* do Facebook;
- **google_app:** Este aplicativo tem responsabilidade unicamente na captura e exibição de dados que são fornecidos pela API do Google Analytics;
- **instagram:** De forma semelhante ao aplicativo do facebook, este é responsável pela busca e exibição de dados provenientes da API do Instagram;
- **usuarios:** Aplicativo voltado exclusivamente para tratar operações envolvendo os usuários.

Por sua vez, a arquitetura *REST* é um estilo que disponibiliza métodos padrões conhecidos como HTTP. Como já falado anteriormente, para este projeto foi escolhido a biblioteca DRF, que como mostrado com exemplos, permite o desenvolvimento de APIs de forma rápida e prática.

A imagem abaixo mostra a arquitetura completa do sistema proposto:

Figura 19 – Arquitetura do Marketing Analytics



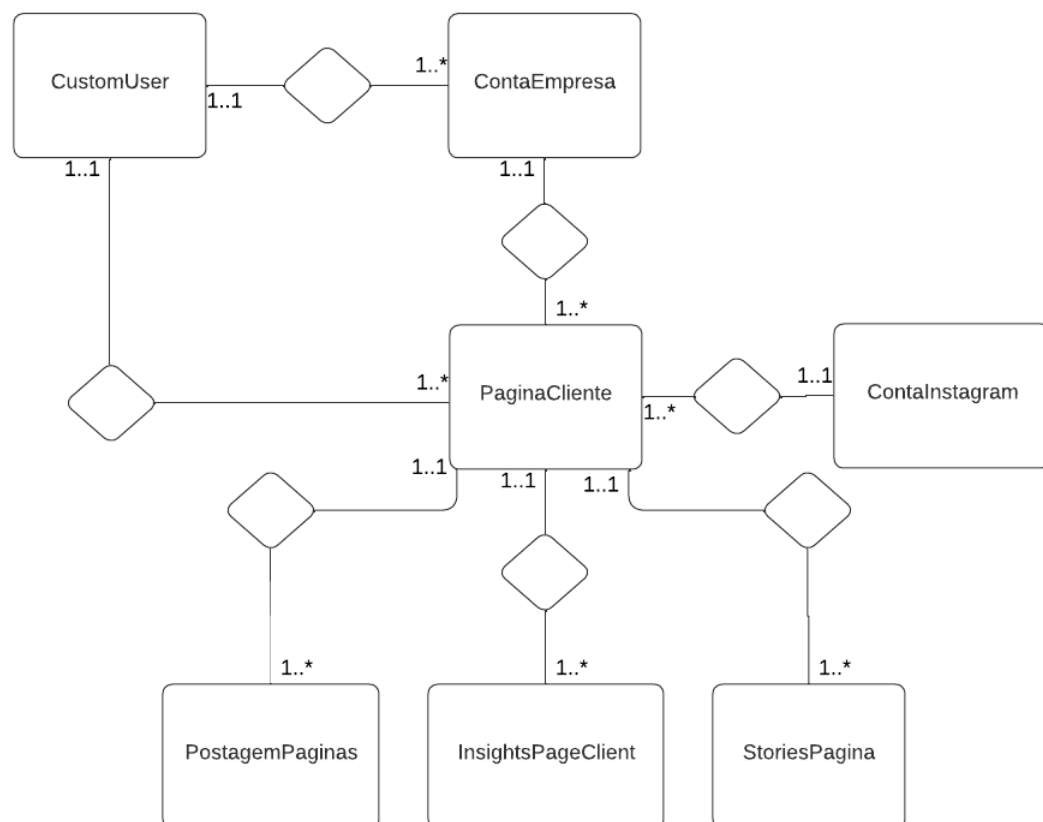
Fonte: Elaborado pelo Autor

4.6 DIAGRAMA DE ENTIDADES-RELACIONAMENTOS

Quando se trabalha com sistema com várias entidades, é necessário adotar a utilização de técnicas para que não se tenha um banco de dados mal projetado. Dessa forma, estes diagramas são focados nas entidades de um sistema e seus relacionamento, ou seja, cada tabela no banco de dados é representado por um quadrado neste diagrama, com exceção os relacionamentos fortes que são representados por losangos. São de suma importância para modelagem de bancos de dados relacionais, sendo na maioria dos casos, uma das primeiras etapas para o desenvolvimento de *softwares*.

Abaixo é mostrado uma imagem que representa os principais relacionamentos do sistema proposto:

Figura 20 – Diagrama de Entidade e Relacionamento

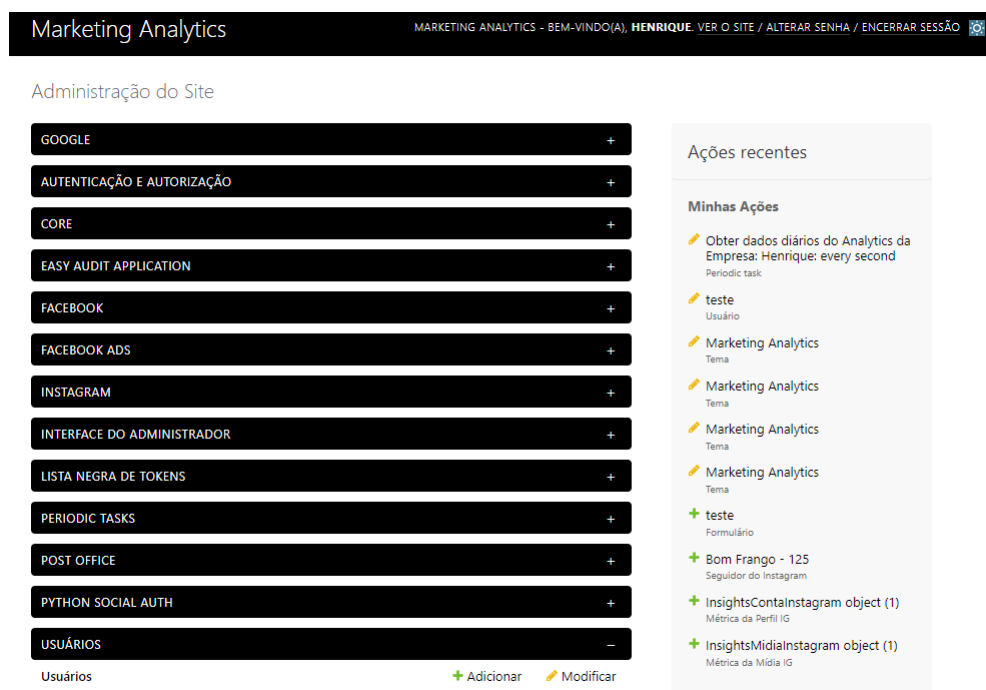


Fonte: Elaborado pelo Autor

4.7 INTERFACE

Este trabalho teve como objetivo principal o desenvolvimento de uma API e a criação de seus testes. Dessa maneira, não foram desenvolvidas telas para mostrar visualmente o seu funcionamento. Todavia, o Django disponibiliza um painel administrativo que permite a realização das operações CRUD dos modelos de banco de dados. Para este painel, foi escolhido a utilização de uma biblioteca Python, de código aberto, chamada de Django Admin Interface que permite customizar o painel, adicionando cores e textos de forma dinâmica, ou seja, ela abstrai a utilização de CSS e HTML5.

Figura 21 – Captura de tela da tela inicial do painel administrativo do Marketing Analytics



Fonte: Elaborado pelo Autor

A imagem acima mostra a tela inicial do painel administrativo, nela é exposto todos os aplicativos criados para o sistema e os aplicativos de pacotes de código aberto Python. Ademais, possui uma secção que lista as ações recentes do usuário.

Por fim, o cabeçalho do painel é exibido em todas as telas em que é possível ver o nome do sistema, o nome do usuário e opções para ver o site, caso tenha, alterar a senha e encerrar a sessão.

Figura 22 – Captura de tela da tela que lista as tarefas periódicas do Marketing Analytics

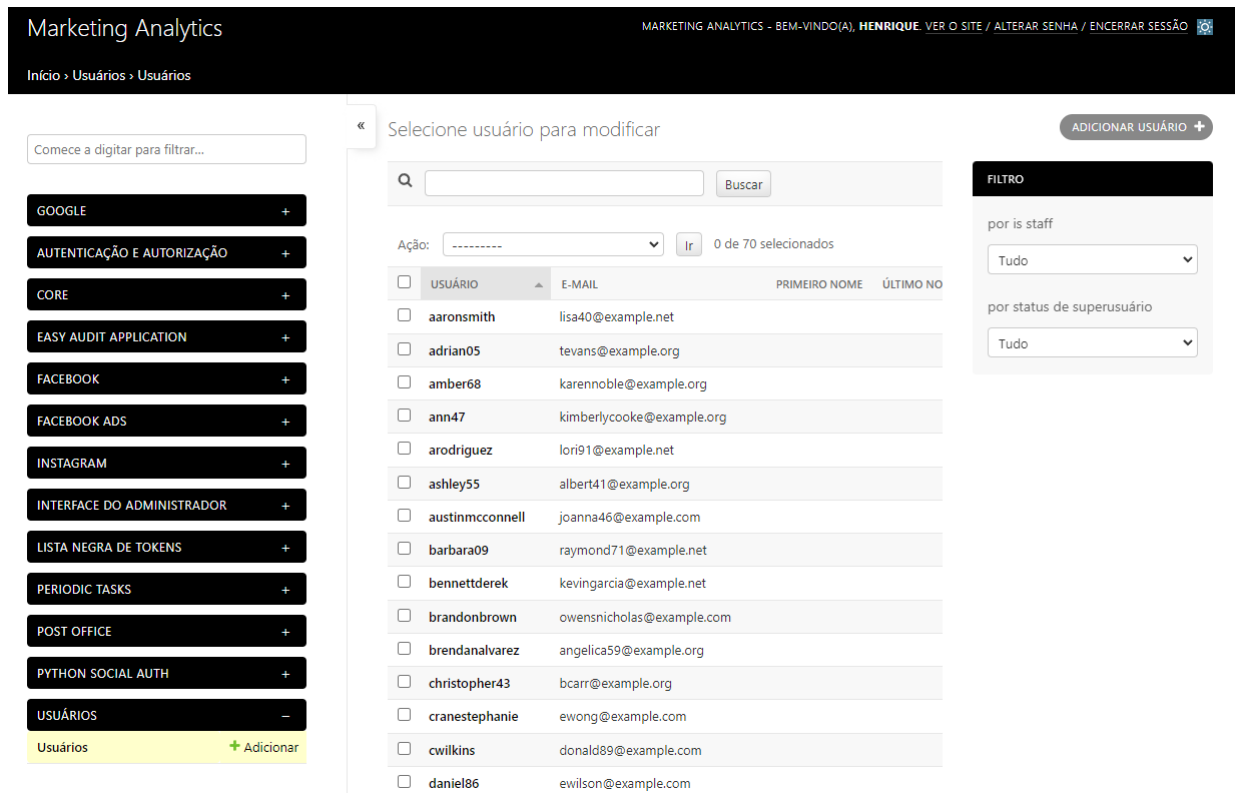
The screenshot displays the 'Marketing Analytics' interface for managing periodic tasks. The top header shows the user 'HENRIQUE' and navigation links. The breadcrumb trail is 'Início > Periodic Tasks > Periodic tasks'. On the left, a sidebar lists various system components, with 'PERIODIC TASKS' currently selected. The main content area is titled 'Selecione periodic task para modificar' and features a search bar and a table of tasks. The table has columns for 'NAME', 'ENABLED', 'SCHEDULER', 'INTERVAL SCHEDULE', and 'START'. Five tasks are listed, including 'Obter dados diários do Analytics da Empresa: Henrique', 'celery.backend_cleanup', 'Criar Metas Semanais dos Clientes', 'Limpeza de Logs do Sistema', and 'Backup do Banco de Dados'. A right-hand panel titled 'FILTRO' provides options to filter tasks by 'Enabled' status, 'One-off Task', 'Task Name', 'Start Datetime', and 'Last Run Datetime'. At the bottom of the table, it indicates '5 periodic tasks'.

NAME	ENABLED	SCHEDULER	INTERVAL SCHEDULE	START
☐ Obter dados diários do Analytics da Empresa: Henrique	●	every second	every second	-
☐ celery.backend_cleanup	●	0 4 * * * (m/h/dM/MY/d) UTC	-	-
☐ Criar Metas Semanais dos Clientes	●	0 3 * * 1 (m/h/dM/MY/d) America/Sao_Paulo	-	-
☐ Limpeza de Logs do Sistema	●	0 3 * * * (m/h/dM/MY/d) America/Sao_Paulo	-	-
☐ Backup do Banco de Dados	●	0 3 * * * (m/h/dM/MY/d) America/Sao_Paulo	-	-

Fonte: Elaborado pelo Autor

Com exceção da tela inicial, todas as demais telas do painel administrativo possuem um menu lateral a esquerda que permite acessar os demais aplicativos do sistema, ademais, nas telas de listagem pode-se ou não ser exibido uma seção a direita que será utilizada para disponibilizar filtros para se utilizar na listagem. Nesta tela é exibido as tarefas periódicas do sistema, podendo visualizar seu nome, o seu status, o horário em que será executada, dentre outros atributos.

Figura 23 – Captura de tela da tela que lista os usuários do Marketing Analytics



Fonte: Elaborado pelo Autor

Acima é mostrado a listagem de usuários seguindo o padrão descrito anteriormente. Nesta listagem é possível visualizar o nome do usuário, seu e-mail, se é super usuário, dentre outros atributos.

O painel administrativo possui mais de 20 telas disponíveis, entretanto, é possível expor suas funcionalidades através das 3 telas que foram exibidas, uma vez que, exibir tela por tela iria se tornar redundante e desnecessário.

4.8 PADRÕES DE PROJETO

Durante o desenvolvimento de *softwares* é comum encontrar problemas, a título de exemplo, toda vez que é realizada uma integração a uma API externa com sucesso, é necessário tratar os dados para que eles possam ser armazenados e trabalhados de forma adequada. Diante disso, surgiu um conceito na engenharia de *software*, os padrões de projeto, que são soluções que resolvem problemas dentro do processo de desenvolvimento de sistemas.

Estes padrões podem ser categorizados em três tipos. Sendo eles, os criacionais que são bastante utilizados para a criação de objetos através da abstração, permitindo uma maior reutilização de código[34]. Os estruturais por sua vez, são focados na montagem de objetos, ou seja, atuam regras que ajudam a manter a flexibilidade e eficiência[34]. Por fim, os comportamentais,

que são focados na comunicação e distribuição de responsabilidades[34]. Atualmente existem um total de 22 padrões de projetos e por modelo, o Django[17] já implementa alguns desses padrões, como por exemplo, a implementação de um *Singleton*² para realizar a conexão com o banco de dados e garantir que só haverá uma instância dessa conexão enquanto o projeto é executado. Este projeto se utiliza de dois padrões, o *Adapter* e o *Factory Method*.

4.8.1 Adapter

Ao realizar integrações com APIs externas os dados podem ser retornados em diversos formatos, tendo como exemplo, JSON e XML ou então pode-se retornar um objeto muito complexo. Dessa maneira, pode ser necessário tratar estes dados para que se possa trabalhar sobre eles de forma adequada, ou seja, este padrão de projeto é utilizado quando se deseja que os dados possam ser compatíveis com todas as classes envolvidas no processo.

A API do Google Analytics através das bibliotecas Python que é recomendado de se utilizar, retorna os dados no formato Protobuf, este é um método de serialização de dados criado pelo próprio Google e fornece adaptadores que permitem transformar esses dados em JSON ou dicionários Python. Além de utilizar essas funções que as bibliotecas do Google fornecem, foi criado um adaptador para tratar situações em que o retorno dos dados do Facebook eram muito complexos, permitindo assim uma melhoria no código em questão de leitura.

4.8.2 Factory Method

Na realização de testes é necessário que se crie dados durante a execução dos testes para simular um cenário o mais próximo possível da realidade. Entretanto, quando se tem muitos casos de testes, a criação desses dados vai se tornando uma tarefa desgastante e repetitiva. Dessa forma, o *Factory Method* é utilizado quando se deseja criar diversos objetos em vários lugares do código, os objetos são criados com os parâmetros definidos. Neste projeto, o mesmo foi utilizado para simular cenários reais, atendendo as regras de negócio, porém de maneira prática e organizada. Posteriormente, a utilização deste padrão será explicada de forma detalhada.

² Este padrão de projeto é utilizado quando se quer ter apenas uma instância de uma classe enquanto o projeto é executado.

5 SOFTWARE

Neste capítulo iremos falar sobre o processo de desenvolvimento da aplicação, abordando técnicas utilizadas para garantir a qualidade do produto.

5.1 O QUE SÃO METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Para garantir a qualidade de um *software* deve-se analisar diversas métricas, dentre elas, podemos citar a segurança, sendo esta uma das métricas de maior relevância, visto que um *software* seguro transmite confiança. Outro medidor pode ser o desempenho, em lojas online que trabalham com o *e-commerce*, como a Amazon, Magazine Luiza, Casas Bahia, onde possuem picos de acesso e uso em determinadas situações, sendo uma delas a *Black Friday*, dessa forma as aplicações devem ser robustas e performáticas para que não atrapalhe a experiência do usuário. Entretanto, a análise dessas e outras métricas não garantem de fato o sucesso do *software*, uma vez que estas acabam sendo inúteis se a aplicação não executar exatamente o que lhe foi proposto, ou seja, se não atender as regras de negócio. Além do mais, para atingir tal objetivo é necessário, inicialmente, entender o problema exposto e a partir deste ponto será possível documentar os requisitos e as necessidades do cliente de maneira clara e objetiva[35].

Estudos da década de 70 expuseram problemas que os sistemas da época possuíam, indo de dificuldades para realizar manutenções nos mesmos ocasionando uma baixa qualidade até a ausência de técnicas para reaproveitar código e por consequência aumentar a agilidade no desenvolvimento[36], este marco ficou conhecido como "Crise de Software". Para exemplo, temos o *software* nomeado de Therac-25, que era uma aplicação utilizada no tratamento de pacientes com câncer. Mas, este sistema acabou apresentando falhas no seu funcionamento e no seu processo de desenvolvimento, o que acabou ocasionando sérias consequências aos pacientes. Dessa forma, observando os problemas enfrentados pelos desenvolvedores, a engenharia de *software* surgiu como forma de sanar esses problemas garantindo que o processo de desenvolvimento seria realizado de forma correta e respeitado.

Diante desse cenário, surgiram as metodologias de desenvolvimento de *software* que segundo Koscianski[37], "Uma metodologia de desenvolvimento de software é um conjunto de atividades que auxiliam na produção de um software". As metodologias tradicionais foram as primeiras a surgirem devido ao problema da época, sua principal característica era ser totalmente orientadas a documentação. Devido a este aspecto, realizar alterações em softwares que adotavam essas metodologias era algo extremamente custoso, fazendo com que o software fosse inteiramente planejado e documentado antes de ser implementado[38]. Apesar das metodologias tradicionais resolverem os problemas enfrentados na década de 1970, elas possuíam os seus próprios desafios, tais como, prazos de entrega e clientes confusos. Segundo dados de 1995, um

total de 8380 projetos utilizados como base, sendo que 16,2% foram entregues, 31% cancelados e 52,7% entregues, mas com os prazos estendidos[39].

Nesse contexto, foram surgindo as metodologias ágeis que tinham o foco na entrega rápida do produto e eram mais voláteis as mudanças de requisitos, ou seja, caso o cliente desejasse realizar alguma alteração isso não seria tão trabalhoso se estivesse utilizando uma metodologia ágil. Apesar de existirem várias metdologias, como A SCRUM e o modelo Cascata, todas elas possuem algumas atividades em comum.

O levantamento de requisitos é essencial para compreender o que o cliente deseja para o sistema. Esta etapa envolve a análise e documentação dos requisitos funcionais e não funcionais. Porém, para desenvolver um *software* é necessário mais do que apenas ter os requisitos coletados e validados, deve-se ter um escopo definido, identificando recursos necessários, estabelecendo cronogramas, definindo assim a fase de planejamento do projeto. Por fim, o ciclo de testes que são realizados nesta fase, incluindo testes de unidade, integração, sistema, aceitação dentre outros tipos de testes que podem ser solicitados. A realização destes testes é para garantir que o *software* funcione da forma como foi solicitado, ou seja, atendendo aos requisitos. Vale destacar que existem diversas outras atividades em comum, tais como implementação, documentação, integração e implantação.

5.1.1 Metodologias tradicionais

5.1.1.1 Modelo Cascata

O modelo Cascata foi a primeira metodologia a surgir no contexto de desenvolvimento de sistemas. Caracterizado por ser em sequências, ou seja, sua próxima etapa só poderia ter início quando a etapa anterior estivesse devidamente documentada e aprovada.

5.1.2 Métodos ágeis

5.1.2.1 SCRUM

O SCRUM é uma metodologia ágil que divide o processo de desenvolvimento de sistemas em várias etapas, chamadas de *Sprints*, onde cada um deles possui seu objetivo e suas atividades relacionadas, tendo o foco em entrega do produto e na participação constante do cliente. Esta metodologia estimula as equipes a aprender com as experiências, a se organizar para resolver um problema e a refletir sobre os êxitos e fracassos para melhorias.

Dentre todos os conceitos relacionados a metodologia SCRUM, um foi bastante utilizado para o desenvolvimento deste produto que foi a forma como as tarefas foram organizadas, no qual cada tarefa podia ter uma numeração dentre os seguintes números 1, 2, 3, 5, 8, indicando o seu grau de dificuldade. No SCRUM este conceito está relacionado ao *Planning Poker*.

Para Cohn[40], esses são alguns dos benefícios da metodologia SCRUM:

- **Maior produtividades e menores custos:** Uma vez que o SCRUM é volátil a mudanças de requisitos, isso não será problema para o desenvolvimento do sistema.
- **Maior qualidade:** Devido a utilização de algumas práticas de desenvolvimento;
- **Maior satisfação dos *stakeholders***¹: Este benefício está relacionado a metodologia SCRUM ser volátil aos desejos dos *stakeholders* em que eles podem definir novas prioridades sem grandes impactos no projeto;
- **Maior engajamento e satisfação no trabalho por parte dos funcionarios:** Um dos principais motivos citados por Cohn[40], é que algumas pesquisas revelaram que os funcionários trabalhavam mais antes de ser adotado o SCRUM.

5.1.3 Aplicação neste projeto

Para garantir a qualidade do *software*, foram utilizados conceitos da metodologia SCRUM juntamente com a utilização do aplicativo Trello. Dessa maneira, as atividades deste projeto receberam um nível de prioridade para determinar a ordem da sua resolução. Como forma de identificar o tipo de atividade, as tarefas receberam uma etiqueta e por fim, seguindo os princípios da metodologia adotada, cada tarefa recebeu um número que representa seu nível de complexidade. Abaixo, estas formas de organização serão detalhadas.

Com relação a priorização de tarefas, foi implementado o seguinte nível de prioridade:

- **Baixa:** Geralmente são tarefas que não tem uma certa urgência para serem desenvolvidas, ou seja, elas não tem um impacto muito grande na aplicação, mas enriquecem o produto;
- **Média:** São tarefas que estão no meio termo, geralmente são *bugs* que não estão atrapalhando o desenvolvimento mas que também pode acabar prejudicando o produto final;
- **Alta:** São tarefas que devem ser desenvolvidas com caráter de urgência, devido a sua importância para o produto final.

Esta abordagem proporciona uma diretriz clara sobre quais tarefas devem receber atenção imediata e quais podem ser abordadas posteriormente.

Como citado anteriormente, cada tarefa recebeu uma etiqueta com o seu respectivo tipo de atividade visando permitir uma compreensão rápida do propósito de cada tarefa.

Os tipos de atividade são:

- **Refactor:** Atividades que recebem esta etiqueta, são funcionalidades já implementadas que irão sofrer alguma mudança;

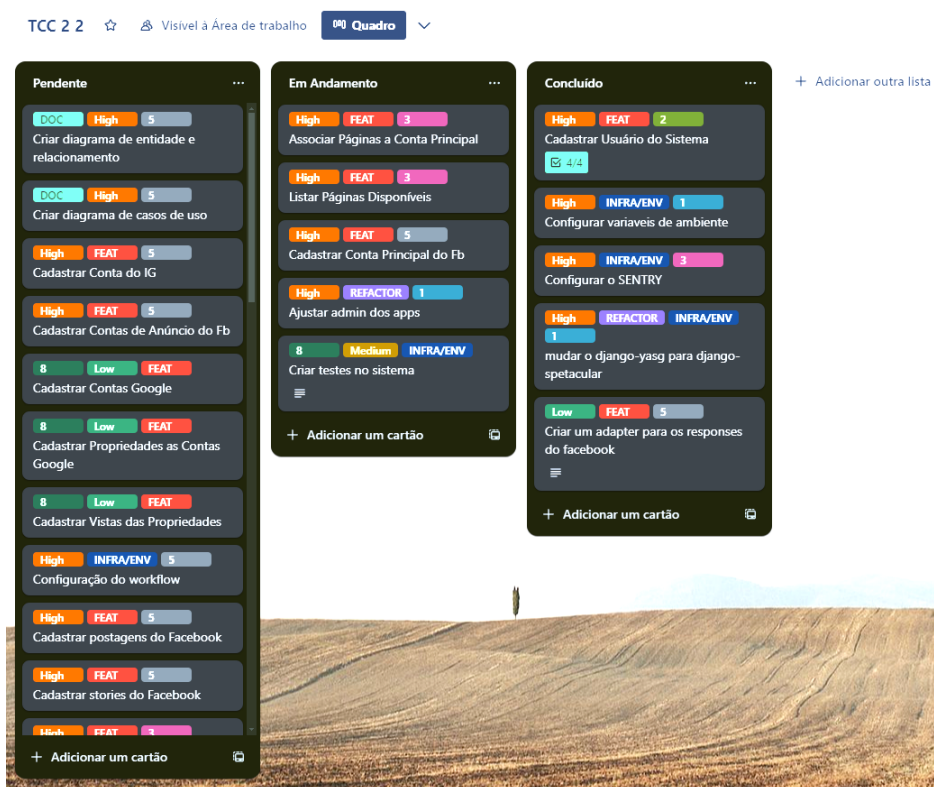
¹ São aqueles que tem alto grau de envolvimento com o projeto, podendo ser clientes, gerentes dentre outros.

- **Feat:** Refere-se a atividades que estão relacionadas a novas funcionalidades do sistema;
- **Infra/Env:** Atividades que recebem esta etiqueta, estão relacionadas a funções de infraestrutura do projeto;
- **Bug:** Refere-se a atividades que estão relacionadas a algum erro que ocorreu no sistema;
- **Doc:** Refere-se a atividades que estão relacionadas a documentação do sistema.
- **Anotação:** Esta etiqueta, faz alusão a tarefas que servem apenas de lembretes.

Seguindo os princípios do SCRUM, foi incorporado uma avaliação de dificuldade para cada tarefa utilizando a escala de pontos de complexidade (1, 2, 3, 5, 8), foi possível estimar a complexidade relativa para as atividades. Isto facilitou o planejamento do desenvolvimento do projeto. Ao integrar essas práticas organizacionais no Trello, foi possível criar uma abordagem transparente, ágil e eficiente para o gerenciamento de tarefas.

Abaixo uma imagem de como foi a organização no trello:

Figura 24 – Captura de Tela do Quadro do Trello



Fonte: Elaborado pelo Autor

5.2 TESTE DE SOFTWARE

A entrega do *software* é uma etapa importante, uma vez que o produto final deve chegar para o cliente totalmente operante e sem falhas. Dessa forma, no processo de desenvolvimento

de sistemas os testes são mais uma etapa com as suas atividades visando garantir que o produto final irá funcionar de acordo com o que foi planejado tendo o seu principal objetivo de expor as falhas do produto para que possam ser corrigidas antes da entrega final[41].

Para garantir que esta etapa seja realizada, existe um profissional exclusivo para esta atividade, conhecido como *Quality Assurance* (QA). Este pode realizar tanto testes automatizados, que são testes que podem ser realizados através do próprio *software* ou com o auxílio de ferramentas embutidas, tais como Cypress, Jest e o Pytest. Sendo assim, não é necessário um profissional para realizar o passo a passo do sistema e também pode realizar os testes manuais, em que um profissional executa a aplicação sem auxílio de programas, geralmente estes testes simulam a experiência do usuário e esta sujeito a falhas, tendo em vista que o usuário muitas das vezes é imprevisível.

Entretanto, engane-se ao imaginar que para garantir a qualidade do produto basta apenas testá-lo de qualquer forma. Estes profissionais acabam criando planos de testes, ou seja, é criado um caminho que deve ser seguido, dessa forma o aplicativo não será testado de qualquer maneira, vale resaltar que este plano pode envolver tanto os desenvolvedores quanto os QA.

Dentro deste plano, existem os níveis de teste de *software*, dos quais citaremos três deles:

- **Teste de unidade:** Estes testes consistem em testar a menor unidade do sistema. Visando encontrar falhas causadas pelo defeito na lógica implementada. Um exemplo da aplicação deste teste é quando se deseja verificar se um método de um objeto no sistema está realizando exatamente aquilo que lhe foi designado. Geralmente este teste é realizado pelos próprios desenvolvedores a medida que vão criando as funcionalidades;
- **Teste de integração:** Este teste por sua vez, tenta identificar falhas entre os relacionamentos do sistema. Um exemplo de sua implementação é verificar se as chamadas de um método dentro de uma outra classe estão funcionando de maneira esperada. Semelhante aos testes de unidades, os testes de integração podem ser realizados pelos próprios desenvolvedores.
- **Teste de sistema ou *End-to-End* (E2E):** Este teste tem por objetivo simular o funcionamento do sistema. Para exemplo, muitos profissionais utilizam ferramentas como o Cypress para simular o usuário utilizando o sistema. Neste teste podem ser encontrados tanto falhas em telas quanto na lógica da aplicação. Sendo recomendado que seja realizados pelos QA, uma vez que os mesmos irão simular o usuário final.

Somados a estes níveis, os profissionais QA também fazem uso de técnicas sempre focando em buscar falhas no produto. Estas técnicas abrangem diferentes perspectivas da aplicação visando definir estratégias que explorem ao máximo as vantagens destas perspectivas[41]. Apesar de existirem diversas técnicas, iremos falar apenas de duas técnicas que foram utilizadas no desenvolvimento desta aplicação.

Segue abaixo as técnicas que serão detalhadas:

- **Teste Caixa Branca:** Esta técnica está voltado para testar internamente o código do sistema, ou seja, o usuário que está testando deve ter conhecimento do código. Tendo o foco em testar condições, fluxos de dados[41], esta técnica está associada aos Testes de Unidade e Integração, uma vez que fica na responsabilidade dos desenvolvedores realizarem estas verificações. Um exemplo desta técnica, é a utilização do Pytest, uma ferramenta para criação de testes na linguagem Python.
- **Teste Caixa Preta:** Sendo o oposto do Teste Caixa Branca, este por sua vez tem por objetivo testar a aplicação sem ter o conhecimento do código. Dessa maneira, este teste é realizado com a entrada de dados e a comparação das saídas com o que é esperado.

Por fim, ao final de toda a etapa de testes, um relatório pode ser gerado pelo QA, ele garante quais foram os testes realizados e fornece sua cobertura, dependendo dessa avaliação definida pelo profissional o produto é aceito e é lançada uma nova versão. Para este projeto, foi utilizada a ferramenta conhecida como Coverage, que permite gerar um relatório em HTML5 desmonstrando o que foi e o que não foi testado. Dessa forma, concluí-se que a etapa de teste de *software* é importante para validar o produto final e impedir que falhas cheguem até o usuário final, podendo envolver uma grande quantidade de recursos e tempo para que a mesma seja aplicada de maneira correta e prática. Ademais, vale ressaltar que cada aplicação pode seguir planos diferentes para realizar testes, tudo irá depender da regra de negócio e do profissional QA que será responsável por implementar esta etapa.

Este projeto teve seus testes desenvolvidos com base na combinação de técnicas caixa branca e caixa preta, combinados com o pacote Python de código aberto *Covarage*, o sistema foi testado visando no mínimo 85% de cobertura de testes. Além disso, foi utilizado de um padrão de projeto conhecido como *Factory Method* através da biblioteca Python Factory boy[42], que permitiu definir classes que ao serem instanciadas, geravam, de forma automática todos os dados necessários para os testes, juntamente com esta biblioteca foi utilizada outra chamada *Model Bakery*[43], esta por sua vez, permite que sejam criados objetos com dados aleatórios sem que necessariamente seja implementado a regra de negócio, bastante útil para quando se deseja criar um registro no banco de dados de forma rápida e prática.

A seguir, imagens que mostram exemplos de como se utilizar estas bibliotecas para facilitar os testes:

Figura 25 – Exemplo básico da utilização do Model Bakery

```
from model_bakery import baker

objeto_teste = baker.make(Teste)
```

Fonte: Elaborado pelo Autor

Deve-se realizar a importação da classe *baker*, que disponibiliza um método chamado de *make*, este por sua vez recebe a classe de modelo de banco de dados que se deseja gerar os dados de forma automática. Por fim, é retornado uma instância da classe.

Figura 26 – Exemplo básico da utilização da Factory Boy

```
import factory

class TesteFactory(factory.django.DjangoModelFactory):
    class Meta:
        model = Teste
```

Fonte: Elaborado pelo Autor

Ademais, caso o usuário queira especificar os campos, deve-se criar um atributo com o nome do campo e inserir os dados de forma manual ou então solicitar que o pacote crie de forma aleatória.

Figura 27 – Exemplo básico da utilização da Factory Boy especificando campos

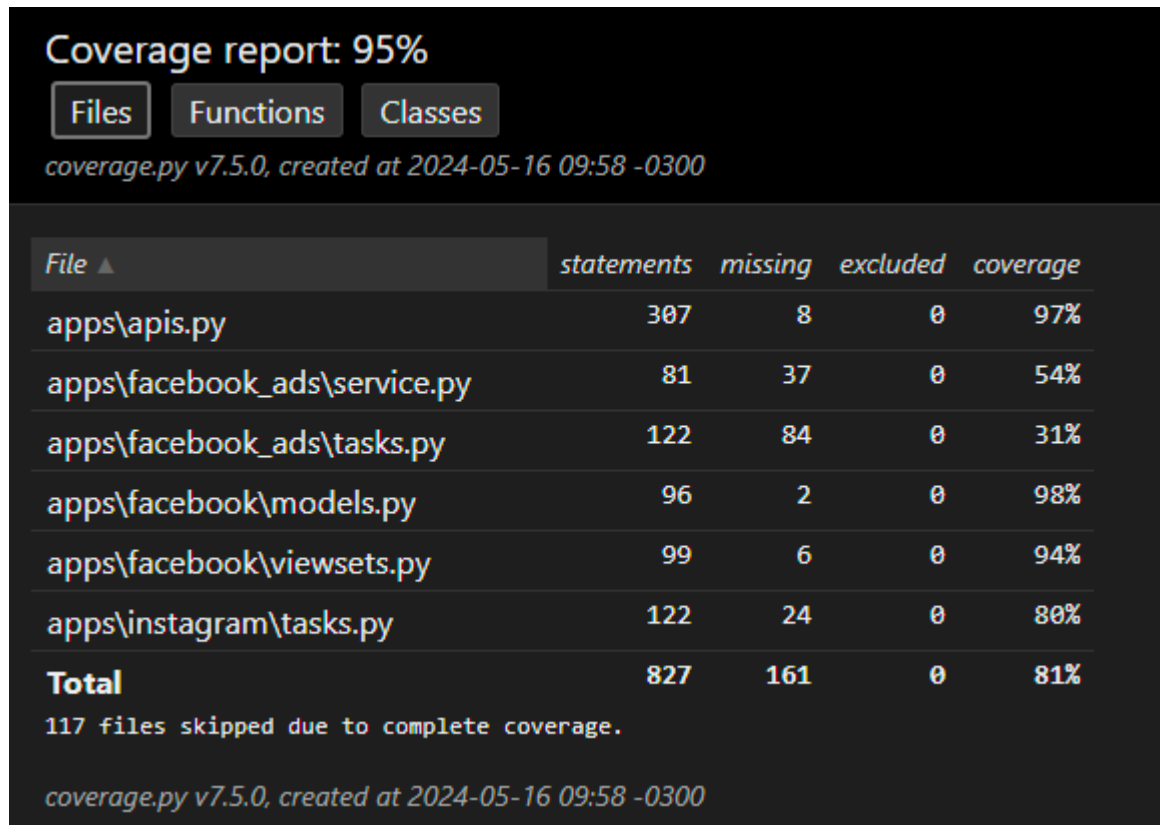
```
import factory

class TesteFactory(factory.django.DjangoModelFactory):
    class Meta:
        model = Teste
        campo_teste = 'teste'
        campo_teste_dois = factory.Faker('name')
```

Fonte: Elaborado pelo Autor

Por fim, abaixo é mostrado uma captura de tela que mostra os arquivos que não tiveram 100% de cobertura nos testes, no qual é possível ver a cobertura total, sendo de 95% e a cobertura de cada arquivo.

Figura 28 – Relatório de cobertura de testes gerado pela biblioteca Coverage



Fonte: Elaborado pelo Autor

6 CONCLUSÃO

Este trabalho apresentou o *software* Marketing Analytics, que é uma API que busca, armazena e disponibiliza dados de plataformas de *marketing* digital, como Facebook, Instagram e Google, com o objetivo de centralizar estes dados, permitindo que profissionais de *marketing* possam gerenciar de forma prática e rápida os seus clientes em um único lugar. Para desenvolver este sistema, foi feito um estudo sobre as estruturas do Meta e do Google, entendendo como funcionam e como utilizá-las, focando bastante na utilização das APIs que cada plataforma disponibiliza. Ademais, o produto foi desenvolvido utilizando técnicas e conceitos importantes no mercado de desenvolvimento, tais como SCRUM, padrões REST, testes de *software* e tecnologias que permitem um desenvolvimento ágil, sempre visando a entrega de um produto com qualidade.

Considera-se que este projeto alcançou seu objetivo principal, que foi centralizar os dados das plataformas de *marketing*, em que o sistema conclui bem o seu papel ao realizar as integrações com o Facebook, Instagram e Google de forma automatizada e prática. Além do mais, a arquitetura do projeto foi pensada em permitir que várias empresas do ramo de *marketing* digital possam realizar o cadastro no sistema para poder trabalhar com os dados de seus clientes sem que haja a interferência técnica no processo e também que estas empresas e outros profissionais, como desenvolvedores, para que possam criar suas próprias telas e relatórios deixando ainda mais customizados.

Por fim, este trabalho contribuiu para a ampliação de conhecimento no desenvolvimento de *softwares* com a utilização de tecnologias que estão consolidadas no mercado de desenvolvimento, tais como o *framework* Django e o SGBD PostgreSQL, a utilização de conceitos de uma metodologia de *software*, a SCRUM, permitiu que fosse possível entender o que estava em desenvolvimento e como estava sendo desenvolvido. Uma coleta e documentação de requisitos desmonstrando que é necessário mapear os desejos do cliente para que não tenha problemas no desenvolvimento do sistema, a utilização de técnicas de diagramação, para entender as necessidades dos clientes de um ponto de vista técnico bem como o entender como o sistema irá se comportar, a utilização de técnicas de teste de *software*, visando sempre garantir a qualidade do produto e finalmente o funcionamento das estruturas do Google e Meta, que foram a base para o desenvolvimento deste projeto.

6.1 TRABALHOS FUTUROS

As APIs citadas possuem uma grande quantidade de atributos que podem ser explorados, bem como o Facebook disponibiliza várias APIs que podem ser utilizadas para melhorarem o produto. Nesse sentido, planeja-se melhorar a integração com o Instagram, realizando a captura de mais dados, como por exemplo, as postagens que estão agendadas para um determinado perfil

e melhorar as funcionalidades já existentes. Com relação aos dados da API de *marketing* do Facebook, pretende-se automatizar relatórios permitindo que os mesmos possam ser enviados em determinados horários do dia em e-mails de quem tiver interesse, assim como deseja-se ter acesso a mais métricas que esta API fornece.

Planeja-se também melhorar a integração com o Google Analytics, tornando-a mais robusta e permitindo que mais dados sejam obtidos com o objetivo melhorar os relatórios já existentes. Ademais, tem-se o interesse em realizar uma integração com o Google AdSense, este por sua vez é voltado para anúncios no Google, dessa forma novas métricas iriam ficar disponíveis para serem coletadas e armazenadas. Somando a essas melhorias, tem-se o desenvolvimento de telas do usuário, no qual a criação dessas telas trará para o produto uma maior relevância no mercado de *marketing* digital.

Por fim, como um último trabalho futuro, pretende-se integrar o Marketing Analytics a algumas inteligência artificial disponíveis no mercado, visando a análise de dados dos clientes para dessa forma ter acesso a relatórios que podem ser utilizados para ter um melhor resultado nas campanhas e anúncios dos clientes, assim como nas postagens do Facebook e Instagram.

REFERÊNCIAS

- 1 SANTOS, T. et al. O desenvolvimento do marketing: uma perspectiva histórica. *REGE Revista de Gestão*, v. 16, n. 1, p. 89–102, 2009. Citado na página 14.
- 2 SAKURAI, R.; ZUCHI, J. D. As revoluções industriais até a indústria 4.0. *Revista Interface Tecnológica*, v. 15, n. 2, p. 480–491, 2018. Citado na página 14.
- 3 LINS, B. F. E. A evolução da internet: uma perspectiva histórica. *Cadernos Aslegis*, v. 48, p. 11–45, 2013. Citado na página 14.
- 4 CRUVINEL, I. B. Marketing digital em tempos de pandemia digital marketing in pandemic times. *Gestão Tecnologia Faculdade Delta, Goiânia 1.1*, 2020. Citado na página 14.
- 5 MENDES, L. Z. R. E-commerce: origem, desenvolvimento e perspectivas. 2013. Citado na página 14.
- 6 SAMPAIO, V.; TAVARES, C. V. C. C. *Marketing digital: O poder da influência das redes sociais na decisão de compra do consumidor universitário da cidade de Juazeiro do Norte-CE*. [S.l.]: Revista Científica Semana Acadêmica, 2017. Citado na página 15.
- 7 MLABS. <<https://www.mlabs.com.br/blog/gerenciamento-de-redes-sociais>>. Acessado: 2023-12-01. Citado na página 15.
- 8 AMARAL, S. A. D. Os 4ps do composto de marketing na literatura de ciência da informação. *Transinformação*, SciELO Brasil, v. 12, p. 51–60, 2000. Citado na página 18.
- 9 LUCCA, D. d.; GALEAZZI, T. L. O instagram como estratégia de marketing digital na empresa nutrativa. 2019. Citado na página 19.
- 10 PINTO, M. M. F. Estudo das vantagens de utilização do tráfego pago em uma empresa de médio porte de máquinas, ferramentas e suprimentos industriais. Pontifícia Universidade Católica de Goiás, 2022. Citado na página 19.
- 11 FAUSTINO, P. *Marketing Digital na Prática: como criar do zero uma estratégia de marketing digital para promover negócios ou produtos*. [S.l.]: DVS Editora,, 2019. Citado 2 vezes nas páginas 19 e 20.
- 12 KINDER, F. H. *Marketing digital e marketing tradicional: uma análise comparativa*. Tese (Doutorado) — Universidade do Minho (Portugal), 2012. Citado na página 20.
- 13 SILVA, G. R. B. Métricas de marketing digital e sua aplicação na gestão das ações de marketing de uma indústria de calçados. Universidade Tecnológica Federal do Paraná, 2017. Citado na página 21.
- 14 USAR mLabs é bom? Saiba tudo sobre a ferramenta de gestão de redes sociais! <<https://www.rdstation.com/blog/marketing/mlabs/>>. Acessado: 2024-04-30. Citado na página 22.
- 15 PYTHON® – the language of today and tomorrow. <<https://pythoninstitute.org/about-python>>. Acessado: 2024-04-30. Citado na página 26.

- 16 SUPPORTED Versions. <<https://www.php.net/supported-versions.php>>. Acessado: 2024-05-05. Citado na página 26.
- 17 DJANGO makes it easier to build better web apps more quickly and with less code. <<https://www.djangoproject.com/>>. Acessado: 2024-05-02. Citado 3 vezes nas páginas 26, 44 e 51.
- 18 ABOUT Swagger Specification Documentation. <<https://swagger.io/docs/specification/about/>>. Acessado: 2024-05-31. Citado na página 27.
- 19 DJANGO REST Framework. <<https://www.django-rest-framework.org/>>. Acessado: 2024-04-23. Citado 2 vezes nas páginas 29 e 31.
- 20 DJANGO Documentação do Celery. <<https://github.com/celery/django-celery-beat>>. Acessado: 2023-12-03. Citado na página 32.
- 21 FEATURES. <<https://flower.readthedocs.io/en/latest/features.html>>. Acessado: 2024-05-02. Citado na página 32.
- 22 ABOUT PostgreSQL. <<https://www.postgresql.org/about/>>. Acessado: 2024-05-02. Citado na página 32.
- 23 DJANGO Documentação Postgres. <<https://docs.djangoproject.com/en/4.2/ref/contrib/postgres/>>. Acessado: 2023-12-03. Citado na página 33.
- 24 QUAL é a diferença entre o Redis e o MongoDB? <<https://aws.amazon.com/pt/compare/the-difference-between-redis-and-mongodb/>>. Acessado: 2024-05-02. Citado na página 33.
- 25 OBJECT Storage. <<https://www.vultr.com/docs/vultr-object-storage/>>. Acessado: 2023-12-03. Citado na página 33.
- 26 1.2 Começando - Uma Breve História do Git. <<https://git-scm.com/book/pt-br/v2/Começando-Uma-Breve-História-do-Git>>. Acessado: 2024-05-04. Citado na página 34.
- 27 DOCUMENTAÇÃO GIT. <<https://git-scm.com/docs/git>>. Acessado: 2024-05-04. Citado na página 34.
- 28 NARDI, J. C.; FALBO, R. de A. Uma ontologia de requisitos de software. In: *CibSE*. [S.l.: s.n.], 2006. p. 111–124. Citado na página 35.
- 29 GRAPH API. <<https://developers.facebook.com/docs/graph-api/overview>>. Acessado: 2024-03-30. Citado na página 37.
- 30 COMO escolher o objetivo do anúncio certo do Gerenciador de Anúncios da Meta. <<https://www.facebook.com/business/help/1438417719786914>>. Acessado: 2024-03-31. Citado na página 38.
- 31 INSTAGRAM API. <<https://developers.facebook.com/docs/instagram-api/>>. Acessado: 2023-12-04. Citado na página 38.
- 32 COTOVICZ, L. A. C. Uso do modelo incremental para o desenvolvimento de um módulo de captura de leads para o vtiger crm. *BS thesis*, 2021. Citado na página 39.
- 33 VALENTE, M. T. Engenharia de software moderna. *Princípios e Práticas para Desenvolvimento de Software com Produtividade*, v. 1, p. 24, 2020. Citado na página 44.

- 34 CLASSIFICAÇÃO dos padrões. <<https://refactoring.guru/pt-br/design-patterns/classification>>. Acessado: 2024-05-13. Citado 2 vezes nas páginas 50 e 51.
- 35 LOPES, L. T. et al. Um modelo de processo de engenharia de requisitos para ambientes de desenvolvimento distribuído de software. Pontifícia Universidade Católica do Rio Grande do Sul, 2005. Citado na página 52.
- 36 FILHO, C. F. *História da computação: O Caminho do Pensamento e da Tecnologia*. [S.l.]: EDIPUCRS, 2007. Citado na página 52.
- 37 KOSCIANSKI, A.; SOARES, M. dos S. *Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. [S.l.]: Novatec Editora, 2007. Citado na página 52.
- 38 SOARES, M. dos S. Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. *INFOCOMP Journal of Computer Science*, v. 3, n. 2, p. 8–13, 2004. Citado na página 52.
- 39 SOARES, M. dos S. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. *Revista Eletrônica de Sistemas de Informação*, v. 3, n. 1, 2004. Citado na página 53.
- 40 COHN, M. *Desenvolvimento de software com Scrum: aplicando métodos ágeis com sucesso*. Bookman. [S.l.]: Bookman, 2000. Citado 2 vezes nas páginas 53 e 54.
- 41 NETO, A.; CLAUDIO, D. Introdução a teste de software. *Engenharia de Software Magazine*, v. 1, p. 22, 2007. Citado 2 vezes nas páginas 56 e 57.
- 42 FACTORY Boy. <<https://factoryboy.readthedocs.io/en/stable/>>. Acessado: 2024-05-09. Citado na página 57.
- 43 MODEL Bakery. <https://model-bakery.readthedocs.io/en/latest/basic_usage.html>. Acessado: 2024-05-09. Citado na página 57.

ANEXO A – ESPECIFICAÇÃO DE CASOS DE USO

ESPECIFICAÇÃO DO CASO DE USO 01 — ASSOCIAR CONTA DO FACEBOOK

Descrição

Este documento tem por objetivo descrever todos os fluxos envolvidos no Caso de Uso 01 - Associar conta do Facebook

Atores

Esta seção lista e descreve todos os atores envolvidos nos fluxos que compõem o Caso de Uso 01 - Doação.

- **Usuário:** Qualquer pessoa que utiliza o Marketing Analytics;
- **Facebook:** API do Facebook;

Pré-condições

- O usuário deve estar cadastrado no sistema;
- O usuário deve estar logado no sistema;
- O usuário deve possuir uma conta que gerencie páginas no Facebook;

Pós-condições

- Será possível ter acesso às páginas dos clientes e às suas contas de Instagram;
- Será possível ter acesso às contas de anúncios, campanhas, adsets e anúncios;

Fluxo de eventos

- **Associar conta do Facebook:** Este fluxo especifica a ação de associar a conta do Facebook. **Fluxo Básico**
 1. O caso de uso tem início quando o ator realiza a chamada no ponto de extremidade que realiza a associação com o Facebook;
 2. O ator é redirecionado para a tela do Facebook
 3. O ator insere os dados de sua conta do Facebook e confirma a associação;
 4. O sistema realiza o cadastro da conta da empresa;
 5. O sistema chama uma função responsável por realizar a configuração das contas de anúncio em segundo plano;

6. O sistema busca as contas de anúncio, as campanhas associadas a estas contas, os adsets associados a estas campanhas e os anúncios associados a estes adsets;
7. O sistema realiza o cadastro de cada entidade no banco de dados;
8. O sistema realiza a busca das páginas de clientes associadas a conta da empresa;
9. O sistema lista as páginas de clientes associadas;
10. O caso de uso termina;

Fluxos Alternativos

1. Usuário não realizou a associação com o Facebook;
 - a) O sistema emite uma mensagem informando que, o usuário que está tentando fazer a operação, não realizou a associação com o Facebook;
2. Falha na comunicação com a API do Facebook;
 - a) O sistema emite uma mensagem informando que ocorreu um erro ao realizar a operação;
 - b) Os responsáveis pelo sistemas são informados no e-mail sobre o erro ocorrido, bem como é salvo no banco de dados e fica disponível para visualização no Sentry;

ESPECIFICAÇÃO DO CASO DE USO 02 — ATIVAR INTEGRAÇÃO COM O GOOGLE

Descrição

Este documento tem por objetivo descrever todos os fluxos envolvidos no Caso de Uso 02 - Ativar integração com o Google

Atores

Esta seção lista e descreve todos os atores envolvidos nos fluxos que compõem o Caso de Uso 02 - Ativar integração com o Google.

- **Usuário:** Qualquer pessoa que utiliza o Marketing Analytics;
- **Google:** API do Google Analytics;

Pré-condições

- O usuário deve estar cadastrado no sistema;
- É necessário que, o e-mail informado no arquivo de credenciais da conta do Google, esteja cadastrado no gerenciamento de contas do Google Analytics;
- O usuário deve ter acesso a plataforma do Google Analytics;
- O usuário deve estar logado no sistema;

Pós-condições

- Será possível obter os relatórios em tempo real e consolidados do Google Analytics;

Fluxo de eventos

- **Ativar integração com o Google:** Este fluxo especifica a ação de ativar a integração com o Google. **Fluxo Básico**
 1. O caso de uso tem início quando o ator acessa o ponto de extremidade que realiza a integração com o Google;
 2. O sistema realiza a busca pelas contas do Google disponíveis;
 3. O sistema realiza o cadastro das contas do Google Analytics;
 4. O sistema realiza o cadastro das propriedades associadas às contas do Google;
 5. O sistema realiza o cadastro das tarefas periódicas responsáveis por obter os relatórios consolidados;
 6. O caso de uso termina;

Fluxos Alternativos

1. Integração já está ativa;
 - a) O sistema apenas retorna o status code 200;
2. Falha na comunicação com a API do Google Analytics;
 - a) O sistema emite uma mensagem informando que ocorreu um erro ao realizar a operação;
 - b) Os responsáveis pelo sistemas são informados no e-mail sobre o erro ocorrido, bem como é salvo no banco de dados e fica disponível para visualização no Sentry;

ESPECIFICAÇÃO DO CASO DE USO 03 — EXECUTAR TAREFAS PERIÓDICAS

Descrição

Este documento tem por objetivo descrever todos os fluxos envolvidos no Caso de Uso 03 - Executar tarefas periódicas

Atores

Esta seção lista e descreve todos os atores envolvidos nos fluxos que compõem o Caso de Uso 03 - Executar tarefas periódicas.

- **Sistema:** Sistema do Marketing Analytics;
- **Facebook:** API do Facebook;
- **Google:** API do Google Analytics;

Pré-condições

- Deve existir uma tarefa periódica cadastrada e ativa no sistema;

Pós-condições

- A tarefa será executada, e dependendo da sua atribuição, podendo ser operações CRUD ou até mesmo consultas em APIs externas, será realizado.

Fluxo de eventos

- **Relizar o cadastro de tarefas periódicas:** Este fluxo especifica a ação de executar uma tarefa periódica.

Fluxo Básico

1. O caso de uso pode ter início a qualquer momento;
2. O sistema busca por tarefas periódicas agendadas;
3. O sistema executa as tarefas periódicas;
4. O sistema busca os dados em uma API externa;
5. O sistema realiza o cadastro destes dados no banco de dados;
6. O caso de uso termina;

Fluxos Alternativos

1. Falha na comunicação com as APIs externas;

- a) O sistema emite uma mensagem informando que ocorreu um erro ao realizar a operação;
- b) Os responsáveis pelo sistemas são informados no e-mail sobre o erro ocorrido, bem como é salvo no banco de dados e fica disponível para visualização no Sentry;

ESPECIFICAÇÃO DO CASO DE USO 04 — CADASTRO DE PÁGINAS DO FACEBOOK

Descrição

Este documento tem por objetivo descrever todos os fluxos envolvidos no Caso de Uso 04 - Cadastro de páginas do Facebook

Atores

Esta seção lista e descreve todos os atores envolvidos nos fluxos que compõem o Caso de Uso 04 - Cadastro de páginas do Facebook.

- **Usuário:** Qualquer pessoa que utiliza o PASIM;

Pré-condições

- O usuário deve ter associado sua conta do Facebook;
- O usuário deve gerenciar ao menos uma página de cliente no Facebook;
- O usuário deve estar logado;

Pós-condições

- O usuário terá acesso as funcionalidades que permitem a conexão com as APIs do Facebook;

Fluxo de eventos

- **Relizar o cadastro de metas para clientes:** Este fluxo especifica a ação de realizar o Criar metas para clientes.

Fluxo Básico

1. O caso de uso tem início quando o usuário seleciona as contas que deseja cadastrar no Marketing Analytics;
2. O sistema realiza a busca pelas informações das páginas;
3. O sistema realiza o cadastro da conta do Instagram dos clientes;
4. O sistema realiza o cadastro das informações das páginas de clientes;
5. O sistema cria um cadastro relacionado a quantidade de seguidores das páginas;
6. O sistema realiza a busca pelos leads associado as páginas de clientes;
7. O sistema realiza o cadastro dos formulários associados aos leads encontrados;
8. O sistema realiza o cadastro dos leads encontrados;

9. O sistema realiza o cadastro de pontos de extremidades;
10. O sistema faz uma requisição no Facebook informando quais pontos de extremidade deseja configurar na plataforma;
11. O sistema cria as tarefas periódicas necessárias para o sistema funcionar como deve;
12. O sistema chama as funções responsáveis pelas configurações iniciais;
 - a) O sistema executa a função responsável por obter as métricas do perfil do dia anterior;
 - b) O sistema executa a função responsável por obter stories e postagens do Facebook;
 - c) O sistema executa a função responsável por obter stories do Instagram;
 - d) O sistema executa a função responsável por obter mídias do Instagram;
 - e) O sistema executa a função responsável por obter informações do perfil do Instagram, tais como métricas;
13. O sistema chama a função responsável por realizar uma conexão com o Facebook com o objetivo de gerar tokens de vida longa;
14. O sistema atualiza o token de acesso de cada cadastro de página de cliente;
15. O sistema retorna informações das contas que foram cadastradas;
16. O caso de uso termina;

Fluxos Alternativos

1. Falha na comunicação com as APIs externas;
 - a) O sistema emite uma mensagem informando que ocorreu um erro ao realizar a operação;
 - b) Os responsáveis pelo sistemas são informados no e-mail sobre o erro ocorrido, bem como é salvo no banco de dados e fica disponível para visualização no Sentry;

ANEXO B – DETALHAMENTO DOS REQUISITOS

B.1 INTRODUÇÃO

Este documento visa expor os requisitos coletados para o desenvolvimento do Marketing Analytics, software que irá coletar, armazenar e expor os dados provenientes de plataformas como o Facebook, Instagram e Google Analytics.

Este documento adota um padrão em que cada requisito funcional possui um identificador, um nome, sua dificuldade, prioridade e uma descrição sobre o que é esperado do requisito.

B.2 REQUISITOS FUNCIONAIS

Identificação	Nome	Dificuldade	Prioridade
RF01	Associar Conta do Facebook	8	ALTA
Descrição			
- Possibilita que o sistema possa ter acesso as páginas de clientes e as contas de anúncio dos clientes.			

Identificação	Nome	Dificuldade	Prioridade
RF02	Gerar tokens sem expiração	2	ALTA
Descrição			
- Possibilita que os tokens de acesso, tanto ta conta principal quanto das contas associadas, não tenha data de expiração, uma vez que os tokens padrões possuem apenas duração de 1 hora.			

Identificação	Nome	Dificuldade	Prioridade
RF03	Cadastrar Pontos de Extremidades das Páginas de Clientes	5	ALTA
Descrição			
- Possibilita que seja permitido realizar uma inscrição em um ponto de extremidade do Facebook.			
- Possibilita a utilização dos webhooks do Facebook.			

Identificação	Nome	Dificuldade	Prioridade
RF04	Buscar pontos de extremidades	3	Baixa
Descrição			
- Possibilita que seja buscado quais pontos de extremidade estão associados a cada página de cliente.			

Identificação	Nome	Dificuldade	Prioridade
RF05	Buscar dados de Páginas de Clientes	8	ALTA
Descrição			
- Possibilita que sejam obtidos dados de páginas dos clientes do Facebook com o objetivo de armazenar estes dados no banco de dados. Através desta funcionalidade é possível ter acesso a outras funcionalidades do sistema.			

Identificação	Nome	Dificuldade	Prioridade
RF06	Buscar leads	3	MÉDIA
Descrição			
- Permite que seja possível obter os <i>leads</i> que já estão cadastrados no Facebook e não estão cadastrados no sistema.			

Identificação	Nome	Dificuldade	Prioridade
RF07	Capturar Leads	5	ALTA
Descrição			
- Possibilita que os <i>leads</i> e entidade associadas a ele, como anúncio, formulário, possam ser cadastrados no sistema através de uma requisição que o Facebook envia para um ponto de extremidade no Marketing Analytics.			

Identificação	Nome	Dificuldade	Prioridade
RF08	Buscar formulários do Facebook	3	ALTA
Descrição			
- Possibilita que seja realizado a busca de formulário que estão cadastrados no Facebook porém não estão cadastrados no Marketing Analytics. Esta funcionalidade também realiza o cadastro dos <i>leads</i> que não estão cadastrados no sistema.			

Identificação	Nome	Dificuldade	Prioridade
RF09	Buscar Postagens de clientes no Facebook	3	ALTA
Descrição			
- Possibilita que seja realizado uma busca na API de páginas do Facebook com o objetivo de se obter as postagens de um página de cliente específica.			

Identificação	Nome	Dificuldade	Prioridade
RF10	Buscar Stories de clientes no Facebook	3	ALTA
Descrição			
- Possibilita que seja realizado uma busca na API de páginas do Facebook com o objetivo de se obter os stories de um página de cliente específica.			

Identificação	Nome	Dificuldade	Prioridade
RF11	Buscar Métricas das páginas dos clientes no Facebook	3	ALTA
Descrição			
- Possibilita que seja realizado uma busca na API de páginas do Facebook com o objetivo de se obter as métricas de um página de cliente específica.			

Identificação	Nome	Dificuldade	Prioridade
RF12	Cadastrar Contato	1	BAIXA
Descrição			
- Possibilita que sejam cadastrados os contatos de pessoas que realizaram os cadastros campanhas ou anúncios do Facebook.			

Identificação	Nome	Dificuldade	Prioridade
RF13	Cadastrar Meta para o Usuário	5	MÉDIA
Descrição			
- Possibilita que sejam cadastradas metas para os usuários, permitindo assim a criação de relatórios customizados referente as postagens dos usuários.			

Identificação	Nome	Dificuldade	Prioridade
RF14	Verificar se o Usuário Cumpriu a Meta	3	MÉDIA
Descrição			
- Possibilita verificar se o usuário bateu a meta atribuída ou não.			

Identificação	Nome	Dificuldade	Prioridade
RF15	Cadastrar Usuários	1	ALTA
Descrição			
- Possibilita que os usuários possam se cadastrar no Marketing Analytics			

Identificação	Nome	Dificuldade	Prioridade
RF16	Backup do Banco de Dados Automatizado	3	ALTA
Descrição			
- Possibilita salvar os dados do sistema diariamente, é uma medida de prevenção para situações as quais possa ocorrer a perda de dados.			

Identificação	Nome	Dificuldade	Prioridade
RF17	Efetuar Login	1	ALTA
Descrição			
- Possibilita que os usuários possam acessar suas respectivas áreas ou a área administrativa.			

Identificação	Nome	Dificuldade	Prioridade
RF18	Buscar mídias de contas do Instagram	3	ALTA
Descrição			
- Possibilita que seja possível buscar as mídias de uma conta no Instagram com o objetivo de salva-las no banco de dados.			

Identificação	Nome	Dificuldade	Prioridade
RF19	Buscar Stories de contas do Instagram	3	ALTA
Descrição			
- Possibilita que seja possível buscar os stories de uma conta no Instagram com o objetivo de salva-los no banco de dados.			

Identificação	Nome	Dificuldade	Prioridade
RF20	Buscar dados da conta do Instagram	3	ALTA
Descrição			
- Possibilita que seja possível buscar as informações de uma conta no Instagram com o objetivo de salva-las no banco de dados.			

Identificação	Nome	Dificuldade	Prioridade
RF21	Buscar o tipo de mídia	3	ALTA
Descrição			
- Possibilita que seja possível buscar o tipo específico de uma mídia selecionada.			

Identificação	Nome	Dificuldade	Prioridade
RF22	Buscar Perfis de Instagram	5	ALTA
Descrição			
- Possibilita que os usuários possam realizar busca por concorrentes no Instagram e analisar seus dados.			

Identificação	Nome	Dificuldade	Prioridade
RF23	Buscar Contas de anúncio	3	ALTA
Descrição			
- Possibilita que seja possível buscar contas de anúncio no Facebook com o objetivo de salva-las no banco de dados para ter acesso as outras funcionalidades.			

Identificação	Nome	Dificuldade	Prioridade
RF24	Buscar campanhas no Facebook	3	ALTA
Descrição			
- Possibilita que seja possível buscar campanhas no Facebook com o objetivo de salva-las no banco de dados para ter acesso as outras funcionalidades.			

Identificação	Nome	Dificuldade	Prioridade
RF25	Buscar adsets no Facebook	3	ALTA
Descrição			
- Possibilita que seja possível buscar adsets no Facebook com o objetivo de salva-las no banco de dados para ter acesso as outras funcionalidades.			

Identificação	Nome	Dificuldade	Prioridade
RF26	Buscar anúncios no Facebook	3	ALTA
Descrição			
- Possibilita que seja possível buscar anúncios no Facebook com o objetivo de salva-los no banco de dados para ter acesso as outras funcionalidades.			

Identificação	Nome	Dificuldade	Prioridade
RF27	Buscar métricas das contas de anúncio	3	ALTA
Descrição			
- Possibilita que seja possível buscar as métricas contas de anúncio no Facebook com o objetivo de salva-las no banco de dados e criar relatórios com base nos dados salvos.			

Identificação	Nome	Dificuldade	Prioridade
RF28	Buscar métricas das campanhas no Facebook	3	ALTA
Descrição			
- Possibilita que seja possível buscar as métricas de campanhas no Facebook com o objetivo de salva-las no banco de dados e criar relatórios com base nos dados salvos.			

Identificação	Nome	Dificuldade	Prioridade
RF29	Buscar métricas dos adsets no Facebook	3	ALTA
Descrição			
- Possibilita que seja possível buscar as métricas de adsets no Facebook com o objetivo de salva-las no banco de dados e criar relatórios com base nos dados salvos.			

Identificação	Nome	Dificuldade	Prioridade
RF30	Buscar métricas dos anúncios no Facebook	3	ALTA
Descrição			
- Possibilita que seja possível buscar as métricas de anúncios no Facebook com o objetivo de salva-las no banco de dados e criar relatórios com base nos dados salvos.			

Identificação	Nome	Dificuldade	Prioridade
RF31	Buscar métricas dos reels do Instagram	5	ALTA
Descrição			
- Possibilita que seja possível buscar as métricas de reels de contas do instagram com o objetivo de salva-las no banco de dados e criar relatórios com base nos dados salvos.			

Identificação	Nome	Dificuldade	Prioridade
RF32	Buscar métricas dos mídias do Instagram	5	ALTA
Descrição			
- Possibilita que seja possível buscar as métricas de mídias de contas do Instagram com o objetivo de salva-las no banco de dados e criar relatórios com base nos dados salvos.			

Identificação	Nome	Dificuldade	Prioridade
RF33	Buscar Contas do Google Analytics	8	ALTA
Descrição			
- Possibilita que possa ser feita a busca por contas no Google Analytics. Esta funcionalidade também trás consigo as propriedades de uma conta do Google Analytics.			

Identificação	Nome	Dificuldade	Prioridade
RF34	Buscar relatório em tempo real do Google Analytics	5	ALTA
Descrição			
- Possibilita que seja realizada uma consulta em tempo real dos dados da conta do Google Analytics.			

Identificação	Nome	Dificuldade	Prioridade
RF35	Buscar relatório de dados consolidados do Google Analytics	5	ALTA
Descrição			
- Possibilita que seja realizada uma consulta dos dados consolidados de uma conta do Google Analytics.			

Identificação	Nome	Dificuldade	Prioridade
RF36	Exportar dados de Contas de Anúncio	3	ALTA
Descrição			
- Possibilita que os dados de contas de anúncio possam ser exportados em 4 formatos, excel, xlsx, csv e json.			

Identificação	Nome	Dificuldade	Prioridade
RF37	Exportar dados de Campanhas	3	ALTA
Descrição			
- Possibilita que os dados de campanhas possam ser exportados em 4 formatos, excel, xlsx, csv e json.			

Identificação	Nome	Dificuldade	Prioridade
RF38	Exportar dados de Adesets	3	ALTA
Descrição			
- Possibilita que os dados de adsets possam ser exportados em 4 formatos, excel, xlsx, csv e json.			

Identificação	Nome	Dificuldade	Prioridade
RF39	Exportar dados de Anúncios	3	ALTA
Descrição			
- Possibilita que os dados de anúncios possam ser exportados em 4 formatos, excel, xlsx, csv e json.			

Identificação	Nome	Dificuldade	Prioridade
RF40	Cadastrar tarefas periódicas	1	ALTA
Descrição			
- Possibilita que os usuários possam agendar a execução de tarefas no sistema.			

Identificação	Nome	Dificuldade	Prioridade
RF41	Ativar Integração com o Google	1	ALTA
Descrição			
- Possibilita que os usuários possam ativar a integração com o Google Analytics.			

Identificação	Nome	Dificuldade	Prioridade
RF42	Visualizar métricas de conta do Instagram	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar as métricas de contas do Instagram, possibilitando a filtragem por datas, e contas			

Identificação	Nome	Dificuldade	Prioridade
RF43	Visualizar métricas de mídias do Instagram	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar as métricas de mídias contas do Instagram, possibilitando a filtragem por datas, e contas			

Identificação	Nome	Dificuldade	Prioridade
RF44	Capturar Métricas de Stories	3	MÉDIA
Descrição			
- Possibilita que seja possível capturar as métricas dos Stories de contas do Instagram que são enviados pelo Facebook através de um ponto de extremidade.			

Identificação	Nome	Dificuldade	Prioridade
RF45	Visualizar métricas de Stories do Instagram	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar as métricas de stories de contas do Instagram, possibilitando a filtragem por datas, e contas			

Identificação	Nome	Dificuldade	Prioridade
RF46	Visualizar histórico de seguidores de contas do Instagram	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar a evolução de seguidores de contas do Instagram, permitindo a filtragem por datas e contas			

Identificação	Nome	Dificuldade	Prioridade
RF47	Visualizar dados de contas de anúncio	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar os dados das contas de anúncios.			

Identificação	Nome	Dificuldade	Prioridade
RF48	Visualizar dados de campanhas	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar os dados das campanhas, permitindo a filtragem por datas e contas de anúncio.			

Identificação	Nome	Dificuldade	Prioridade
RF49	Visualizar dados de adsets	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar os dados das contas de anúncios, permitindo a filtragem por datas, contas de anúncio e campanhas.			

Identificação	Nome	Dificuldade	Prioridade
RF50	Visualizar dados de anúncios	2	MÉDIA
Descrição			
- Possibilita que seja possível visualizar os dados dos anúncios, permitindo a filtragem por datas, contas de anúncio, campanhas e adsets.			

B.3 REQUISITOS NÃO-FUNCIONAIS

B.3.1 Requisitos de Produtos

B.3.1.1 Usabilidade

Identificação	Descrição
RNF01	O painel administrativo deverá ser responsivo
RNF02	Os O sistema não deve demorar mais que 20 segundos em uma requisição.
RNF03	Requisições demoradas, devem ser executadas em segundo plano.

B.3.1.2 Confiabilidade

Identificação	Descrição
RNF04	Os dados devem ser sempre validados antes de ser feito a interação com o banco de dados
RNF05	O sistema deve está disponível pelo máximo de tempo.

B.3.1.3 Segurança

Identificação	Descrição
RNF06	Área administrativa pode ser acessada tanto pelo funcionário quanto pelo cliente, porém as permissões devem ser definidas.
RNF07	O usuário será autenticado no sistema através de senha que será criptografada ao ser armazenada.

B.3.2 Requisitos de Processo

B.3.2.1 Padrões

Identificação	Descrição
RNF08	O sistema deve utilizar um banco de dados relacional para o armazenamento de dados dos clientes.
RNF09	Para o armazenamento de cache, deve-se utilizar o Redis, em contrapartida do sistema padrão de cache do Django.

B.3.2.2 Organização

Identificação	Descrição
RNF10	Os backups devem ser armazenados externamente.
RNF11	Os backups devem ser realizados sempre as 03:00 da manhã no horário de Brasília.