PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Sistema Web de Conexão entre Influenciadores e Marcas

João Gabriel Pereira Pinheiro de Lemos

RELATÓRIO DA DISCIPLINA DE PROJETO FINAL

CENTRO TÉCNICO CIENTÍFICO – CTC
DEPARTAMENTO DE INFORMÁTICA
Curso de Graduação em Ciência da Computação

Rio de Janeiro, dezembro de 2024

João Gabriel Pereira Pinheiro de Lemos

RELATÓRIO DE PROJETO FINAL DE GRADUAÇÃO

Relatório de Projeto Final 2, apresentada ao
programa de Ciência da Computação da PUC-Rio
como requisito parcial para a obtenção do título de
Bacharel em Ciência da Computação.

Orientador: Edmundo Bastos Torreão
Departamento de Informática

Rio de Janeiro, dezembro de 2024

Resumo

Lemos, João Gabriel Pereira Pinheiro. Torreão, Edmundo Bastos. Sistema Web de Conexão entre Influenciadores e Marcas. Rio de Janeiro, 2024. (Número de páginas:63) Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto desenvolve um sistema web para a gestão de influenciadores e campanhas de marketing, focando na otimização das estratégias de divulgação de marcas por meio do marketing de influência. A plataforma permite que marcas de diferentes segmentos lancem campanhas de forma organizada, onde influenciadores podem se inscrever.

O sistema facilita o processo de inscrição e seleção de influenciadores para campanhas, com um orçamento previamente definido pela marca. Uma vez selecionados, os influenciadores realizam postagens nas redes sociais, seguindo um post ‘template’ elaborado pela marca. Então, o crawler da aplicação analisa as redes sociais e registra para a marca quando identifica um post compatível com o post ‘template’. Ao término da campanha, a marca realiza o pagamento ao influenciador de forma prática e flexível, podendo optar por métodos como cartão de crédito, PIX ou outras opções que sejam mais convenientes tanto para a marca quanto para o influenciador. A avaliação do influenciador é baseada na quantidade de posts entregues dentro do prazo da campanha que sejam compatíveis com o post ‘template’.

Palavras-chave: gestão de campanhas; marketing de influência; otimização de divulgação.

Abstract

Lemos, João Gabriel Pereira Pinheiro. Torreão, Edmundo Bastos. Web System for Connecting Influencers and Brands. Rio de Janeiro, 2024. (Number of pages:63) Final Project Report – Department of Informatics. Pontifical Catholic University of Rio de Janeiro.

This project develops a web system for managing influencers and marketing campaigns, focusing on optimizing brand promotion strategies through influencer marketing. The platform enables brands from various sectors to launch campaigns in an organized manner, allowing influencers to apply.

The system streamlines the process of registering and selecting influencers for campaigns, with a budget predefined by the brand. Once selected, influencers create social media posts following a 'template' post designed by the brand. The application’s crawler then analyzes social media and notifies the brand when a post matches the 'template' post. At the end of the campaign, the brand processes the payment to the influencer in a practical and flexible manner, choosing from methods such as credit card, instant transfer (PIX), or other options that best suit both the brand and the influencer. The influencer’s evaluation is based on the number of posts delivered within the campaign’s timeframe that comply with the 'template' post.

Keywords: campaign management; influencer marketing; promotion optimization.

Sumário

1. Introdução..... 1
2. Mercado Profissional de Influencer..... 2
3. Situação Atual..... 4
4. Propostas e Objetivos..... 6
5. Plano de Ação..... 7
6. Cronograma Previsto Inicialmente..... 8
7. Diagrama de Classes..... 9
8. Diagrama de Casos de Uso..... 10
9. Requisitos Não Funcionais..... 11
10. Mapa de Navegação entre as telas do sistema..... 12
11. Descrição de Casos de Uso..... 13
UC01 – Cadastro de Influencer..... 13
UC02 – Alterar Informações do Influencer..... 16
UC03 – Deleção de Conta do Influencer..... 18
UC04 – Lançar campanha..... 19
UC05 - Aplicar para campanha..... 21
UC06 – Aprovar aplicação..... 22
UC07 – Verificar Post..... 24
UC08 – Discordar de post inicialmente recusado..... 26
UC09 – Realiza Pagamento..... 27
UC10 – Avalia influencer..... 29
12. Outras Funcionalidades do Projeto..... 31
12.1 Funcionalidades para o Usuário Marca..... 31
12.1.1 Gerenciar Campanha Completas ou Finalizadas..... 31
12.1.2 Definir Posts Exemplo..... 31
12.1.3 Configurar Crawler..... 31
12.1.4 Visualizar Relatório de Campanha..... 32
12.1.5 Visualizar Notificações..... 32
12.2 Funcionalidades para o Usuário Influenciador..... 33
12.2.1 Visualizar Campanhas Disponíveis..... 33
12.2.2 Visualizar Status da Aplicação..... 34
12.2.3 Adicionar um Novo Perfil de Media Social..... 34
12.2.4 Editar Perfis de Media Social..... 35
12.2.5 Excluir Perfis de Media Social..... 35
12.2.6 Visualizar Notificações..... 36
12.3 Funcionalidades do Sistema..... 36
12.3.1 Comparar Posts com Post ‘Template’..... 36
12.3.2 Atualizar Status da Campanha..... 36

12.3.3 Exibir Notificações..... 36
12.3.4 Crawler.....37
13. Conceitos das Tecnologias Usadas no Sistema..... 37
14. Estrutura do Projeto..... 38
14.1 Estrutura do Django e o Padrão MVT(Model-View-Template).....38
14.2 Conexão entre Front-end e Back-end no Django.....39
14.2.1 Interação com o Ícone do Perfil..... 39
14.2.2 Requisição HTTP GET para o Servidor.....40
14.2.3 Django URL Configuration (urls.py).....40
14.2.4 Execução da View Function..... 40
14.2.5 Consulta ao Banco de Dados.....41
14.2.6 Processamento dos Dados.....42
14.2.7 Renderização do Template HTML..... 42
14.2.8 Resposta HTTP com o HTML Renderizado..... 42
15.3 Estrutura dos Arquivos “Back-end”..... 43
15.3.1 Account App..... 43
14.3.2 Campaign App..... 44
15.3.3 Crawler App..... 45
15.3.4 Home App.....46
15.3.5 Mysite..... 47
15.4 Estrutura dos Arquivos “Front-end”.....48
16. Desafios encontrados..... 49
16.1 Post ‘Template’..... 50
16.2 Crawler..... 50
17. Testes Realizados.....51
17.1 Deleção de Conta do Influencer..... 51
17.2 Lançar Campanha..... 53
17.3 Verificação de Post (Crawler)..... 55
17.4 Aplicar para a Campanha..... 57
17.5 Notificações.....58
17.6 Avaliação do Influencer.....59
18. Considerações Finais..... 62
19. Trabalhos Futuros..... 63
20. Referências Bibliográficas..... 64

1. Introdução

A ascensão das redes sociais e a popularidade dos influenciadores digitais mudaram drasticamente o cenário da publicidade e do marketing, criando novas oportunidades e desafios para marcas em todos os setores. Neste contexto, conectar eficazmente marcas e influenciadores é crucial para maximizar o impacto das campanhas publicitárias e alcançar públicos específicos de forma autêntica e envolvente

O projeto visa conceber e implementar uma aplicação que facilite a ligação entre influenciadores digitais e marcas, proporcionando uma plataforma eficaz para a realização de campanhas publicitárias. O objetivo do projeto é ajudar as marcas a identificar e selecionar influenciadores dispostos a promover os seus produtos, com o objetivo de otimizar o processo de criação e gestão de campanhas publicitárias.

No âmbito deste projeto, o aplicativo é concebido como uma ferramenta multifacetada, capaz de envolver uma variedade de etapas e recursos para garantir a seleção eficaz dos influenciadores. Isso pode incluir desde a análise de métricas de desempenho e engajamento dos influenciadores até a facilitação de negociações contratuais e a gestão de campanhas publicitárias.

A escolha certa de influenciadores desempenha um papel fundamental na eficácia de uma campanha publicitária, mantendo a imagem da sua marca e criando uma ligação genuína com o seu público-alvo. Nesse contexto, este trabalho de conclusão de curso propõe o desenvolvimento de uma solução tecnológica que otimize o processo de seleção e gestão de influenciadores, contribua significativamente para o sucesso das campanhas publicitárias e fortaleça o relacionamento entre marcas e influenciadores.

A criação deste aplicativo visa não apenas facilitar as interações comerciais, mas também proporcionar uma plataforma robusta que sustente o crescimento sustentável das marcas e a eficácia das suas estratégias de marketing digital.

2. Mercado Profissional de Influencer

Com o crescimento do uso da internet e das redes sociais, emergiu um novo mercado digital: influenciador digital (criador de conteúdo). Esses influenciadores se tornaram peças-chave nas estratégias de publicidade e comunicação das marcas de pequenas e grandes empresas, tornando-se assim muito significativos para o marketing nos últimos anos. Mas afinal, o que são os influenciadores digitais?

O advento das mídias sociais criou novas oportunidades para pessoas com uma grande quantidade de seguidores ou alto engajamento nessas mídias, permitindo-lhes influenciar comportamentos de compra e opiniões de uma audiência específica. Um influenciador digital é alguém que construiu uma reputação online e possui autoridade em um determinado nicho, ganhando a confiança de seguidores que buscam suas recomendações e opiniões.

Mas antes de explorarmos o papel dos influenciadores, é importante refletir sobre como a publicidade era feita anteriormente. Antes do surgimento dos influenciadores, a publicidade tradicional se concentrava em veículos como outdoors, rádio que mantém uma grande penetração, basta verificar a quantidade de rádios FM que o Rio de Janeiro possui (obs: as rádios FM tem um alcance de 100km), ou seja, é um público local e sobretudo na televisão, devido o alcance de sua abrangência. Os anunciantes presumiam conhecer seu público-alvo, mas muitas vezes essa verba publicitária possivelmente era mal alocada, sem um retorno direto mensurável.[1]

Com os influenciadores, a dinâmica mudou. Esse novo mercado começou inicialmente com blogs e canais no YouTube, contudo, hoje em dia cresceu para outras plataformas como Twitter, Instagram e TikTok, que atualmente são as mais usadas no Brasil. Com esse crescimento foi possível notar o aumento de influenciadores, abrangendo desde celebridades tradicionais, como figuras da televisão ou rádio, até microinfluenciadores com um número menor de seguidores, mas com alto nível de engajamento. O engajamento refere-se à participação ativa, interação e comprometimento das pessoas com determinado assunto e é medido pelo envolvimento, como curtidas, comentários e compartilhamentos.

A estratégia de uma marca em fazer parceria com influenciadores ocorre porque eles oferecem um meio mais eficaz de se conectar com o público-alvo de forma pessoal e direta, diferentemente dos meios tradicionais, como a TV. Grande parte dessas parcerias pode ampliar o alcance das campanhas, melhorar a percepção da marca e impulsionar as vendas.

Existem diversas formas de colaboração entre marcas e influenciadores, incluindo publicações patrocinadas, parcerias de longo prazo, presença em eventos, entre outros. A escolha do influenciador mais indicado depende do público-alvo da marca, dos objetivos da campanha e da sinergia da conexão entre influenciador e produto ou serviço.

Essa mudança vem gradualmente realocando os investimentos publicitários. Os grandes anunciantes são, por natureza, conservadores, buscam resultados comprovados ao longo do tempo. Imagine se toda a verba publicitária estivesse direcionada apenas para o Twitter, por exemplo, e essa plataforma fosse bloqueada no Brasil devido a problemas que vemos hoje na nossa política. Ou se o YouTube enfrentasse desafios semelhantes. Essa vulnerabilidade é algo que nunca enfrentamos com rádio e televisão, que têm sido canais tradicionais de publicidade por décadas.

Em suma, o mercado de influenciadores digitais representa uma mudança significativa no panorama do marketing contemporâneo. Sua capacidade de influenciar comportamentos de compra e opiniões os torna uma parte essencial das estratégias de marketing de muitas marcas. No entanto, é importante reconhecer os desafios e considerar, dentre outras, as implicações éticas à medida que esse mercado continua a se desenvolver e se transformar.[2][3]

3. Situação Atual

Hoje em dia, na era digital, o marketing de influência tornou-se uma ferramenta poderosa para as marcas alcançarem e engajarem seus públicos-alvo de maneira autêntica e eficaz. O advento das redes sociais e plataformas de conteúdo online proporcionou um espaço para indivíduos carismáticos e criativos, os influenciadores, que alcançam uma audiência própria e influenciam no seu poder de compra. Ao mesmo tempo, as marcas reconhecem o valor de colaborar com esses influenciadores para promover seus produtos e ideias.

Neste contexto, aplicativos ou sites que auxiliam a comunicação entre influenciadores e marcas surgiram como uma solução eficaz que simplifica e otimiza o processo de interação. Essas plataformas oferecem uma variedade de recursos e funcionalidades projetadas para atender às necessidades de ambas as partes envolvidas.

Atualmente, um exemplo bem conhecido de aplicativo que executa essa função é o The Lyfe [4]. Com origem em Paris, o The Lyfe construiu uma reputação como plataforma de referência no mercado de marketing de influenciadores, oferecendo uma ampla gama de recursos e serviços para marcas e influenciadores.

As marcas podem postar descrições de trabalhos que desejam que o influenciador execute, indicando se será uma permuta ou remunerado. A permuta é uma forma de negociação entre a marca e o influenciador, na qual o influenciador promove um produto ou serviço em troca de receber itens da marca. Em contrapartida, na remuneração, o influenciador é pago em dinheiro para realizar a propaganda.

Os influenciadores, previamente analisados e aprovados pela equipe do aplicativo, podem então explorar as oportunidades disponíveis e dar "match" com aquelas que se alinham com seus interesses e valores de marca. A decisão final sobre a parceria é tomada pela empresa, proporcionando transparência e controle ao processo.

Em conclusão, os aplicativos que conectam influenciadores com marcas representam uma resposta inovadora às necessidades do mercado de marketing de influenciadores. Através de funcionalidades avançadas e abordagens centradas no

usuário, essas plataformas têm o potencial de fortalecer e melhorar as relações entre influenciadores e marcas, impulsionar o sucesso das campanhas e gerar resultados mensuráveis para todas as partes envolvidas. No entanto, é crucial que essas aplicações continuem a evoluir e a adaptar-se para permanecerem relevantes no ambiente digital em constante mudança.

4. Propostas e Objetivos

Este projeto visa desenvolver um aplicativo cujo objetivo é fornecer uma estrutura organizada e eficiente para o gerenciamento de influenciadores e campanhas de marketing. Com a crescente importância do marketing de influência nas estratégias de divulgação de marcas, é essencial ter um sistema robusto que possa armazenar e relacionar informações de forma precisa e acessível.

O aplicativo tem como propósito permitir que marcas de diferentes segmentos lancem campanhas para promover seus produtos ou serviços. Os influenciadores podem se inscrever nessas campanhas, as quais incluem um orçamento definido pela marca. Após serem selecionados, os influenciadores realizam postagens nas redes sociais, destacando o produto ou serviço em questão, segundo um ‘template’ definido pela marca.

Após as postagens, a marca realiza o pagamento ao influenciador, oferecendo diversas opções de pagamento como cartão de crédito ou PIX, a opção que seja mais conveniente tanto para a marca quanto para o influenciador. Além disso, a marca também fornece uma avaliação do desempenho do influenciador, com base em critérios predefinidos, para garantir a qualidade da parceria e o alcance desejado da campanha, ou seja, o crawler da aplicação, analisando o que foi postado em redes sociais, registra para a marca quando houver um post compatível com o post ‘template’. A avaliação do influencer é dada pela quantidade de posts que o mesmo conseguiu entregar dentro do prazo da campanha compatíveis com o post ‘template’

5. Plano de Ação

Projeto Final I

1. Documentação
    a. Documentação da pesquisa e análise que foi feita a fundo sobre assunto abordado, no intuito de entender melhor o mercado e situação atual do assunto em questão;
2. Proposta de Projeto Final I
    a. Criação do arquivo de proposta de projeto final I, contendo parte da documentação e o plano de ação;
3. Definição do escopo
    a. Definir as funcionalidades que o aplicativo irá ter;
4. Modelagem
    a. Criação do modelo de classes do domínio do problema;
    b. Elaboração do diagrama de casos de uso;
    c. Especificação de requisitos
5. Relatório de Projeto Final I

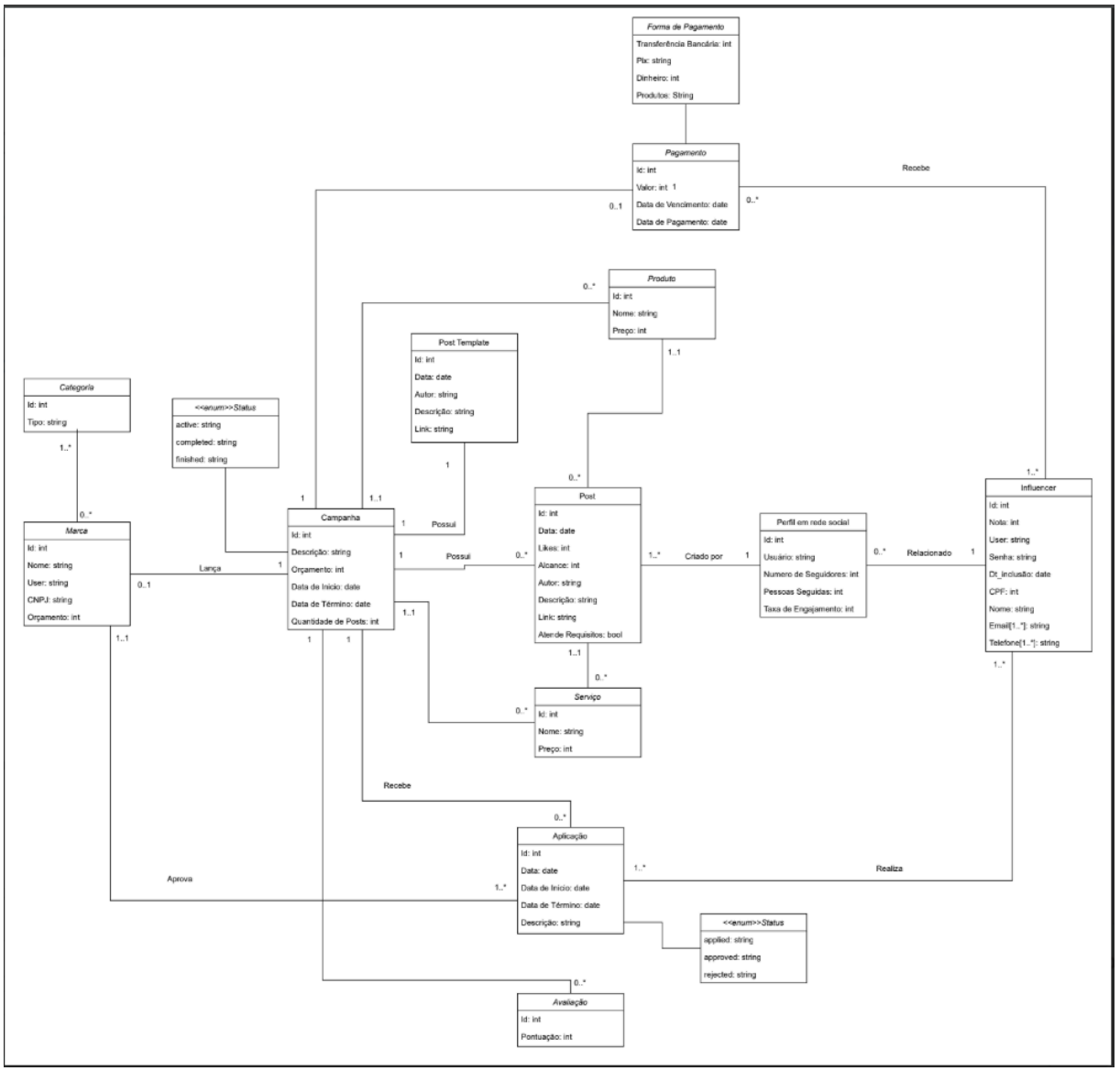
Projeto Final II

6. Pesquisa de tecnologias
    a. Pesquisa de tecnologias para o desenvolvimento do projeto;
7. Estudo das tecnologias escolhidas
8. Projeto de arquitetura
9. Revisão e modificações
    a. Após estudos de tecnologias e criação de projeto de arquitetura, revisar e realizar alterações na Modelagem [4]
10. Implementação
    a. Implementação de tudo estudado e realizado até o momento na criação do projeto;
11. Testes
    a. Executar bateria de testes sobre a implementação;
12. Relatório de Projeto Final II

6. Cronograma Previsto Inicialmente

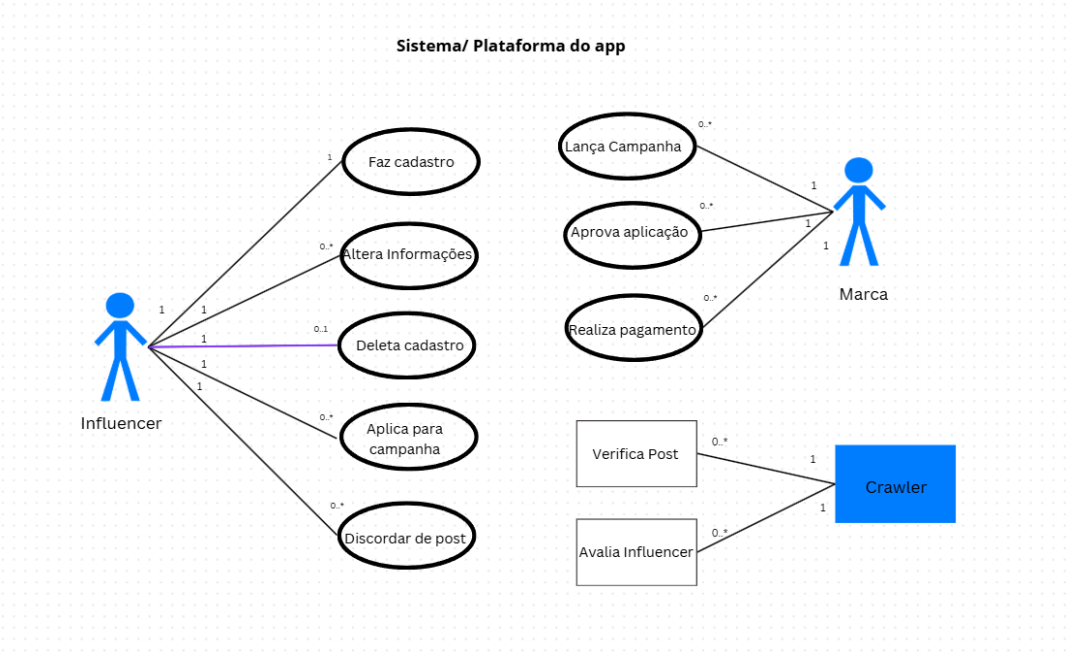
	mai.2024	jun. 2024	jul. 2024	ago. 2024	set. 2024	out.2024	nov. 2024	dez. 2024
1. Documentação								
2. Proposta de Projeto Final I								
3. Definição do escopo								
4. Modelagem								
5. Relatório Projeto Final I								
6. Pesquisa de tecnologias								
7. Estudo das tecnologias escolhidas								
8. Projeto de arquitetura								
9. Revisão e modificações								
10. Implementação								
11. Testes								
11. Relatório de Projeto Final II								

7. Diagrama de Classes



Link - Diagrama de Classes.drawio.pdf

8. Diagrama de Casos de Uso

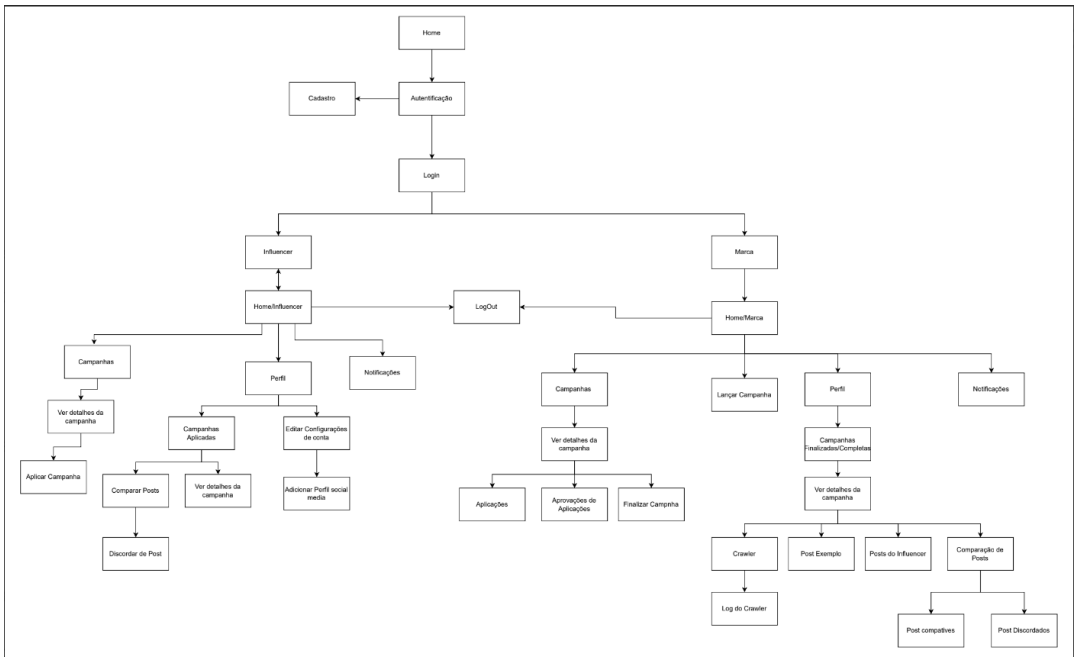


Link - Casos de Uso.pdf

9. Requisitos Não Funcionais

Segurança	O sistema deve garantir a segurança dos dados dos usuários.Os dados pessoais dos influenciadores e das marcas devem ser armazenados de forma segura
Confiabilidade	O sistema deve estar disponível para acesso durante a maior parte do tempo, com um tempo de inatividade limitado.
Usabilidade	O sistema deve ter uma interface deve ser simples e intuitiva, oferecendo uma boa experiência ao usuário.
Manutenibilidade	O sistema deve possuir o código organizado para facilitar manutenções, correções e extensões futuras.
Implementação	O sistema deve ser implementado em Python, JavaScript, CSS e HTML.

10. Mapa de Navegação entre as telas do sistema

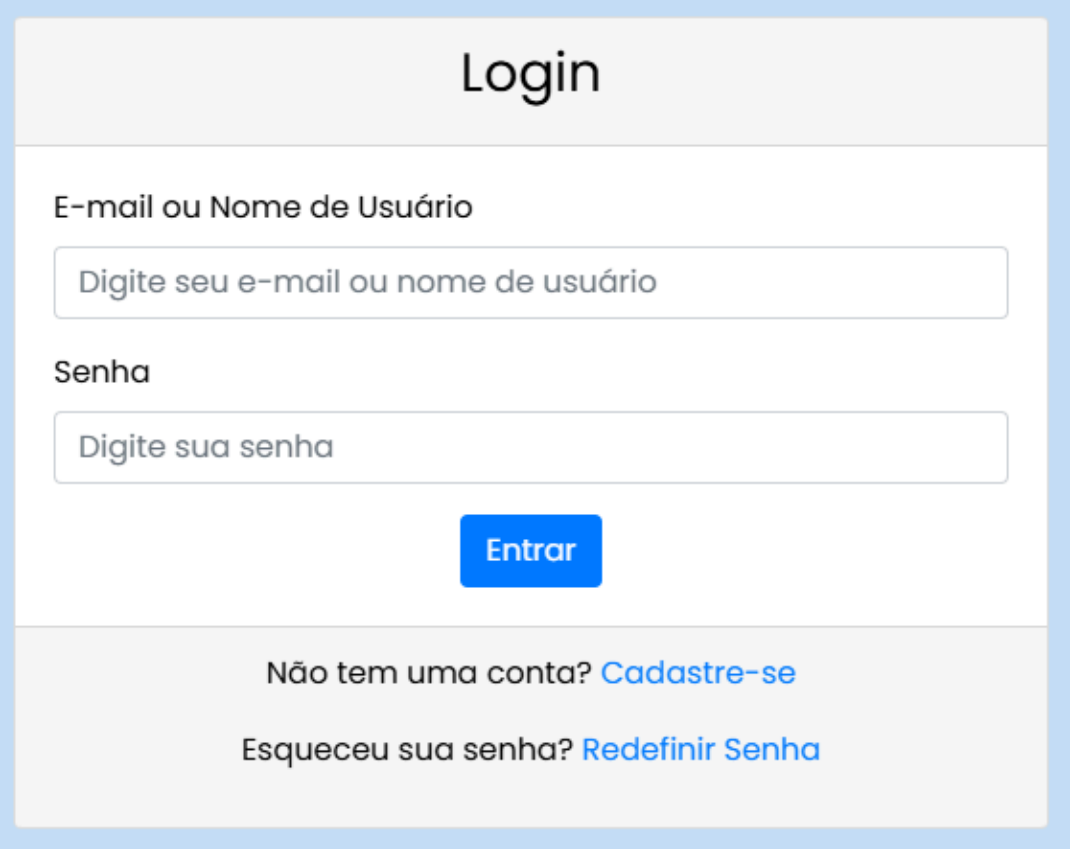


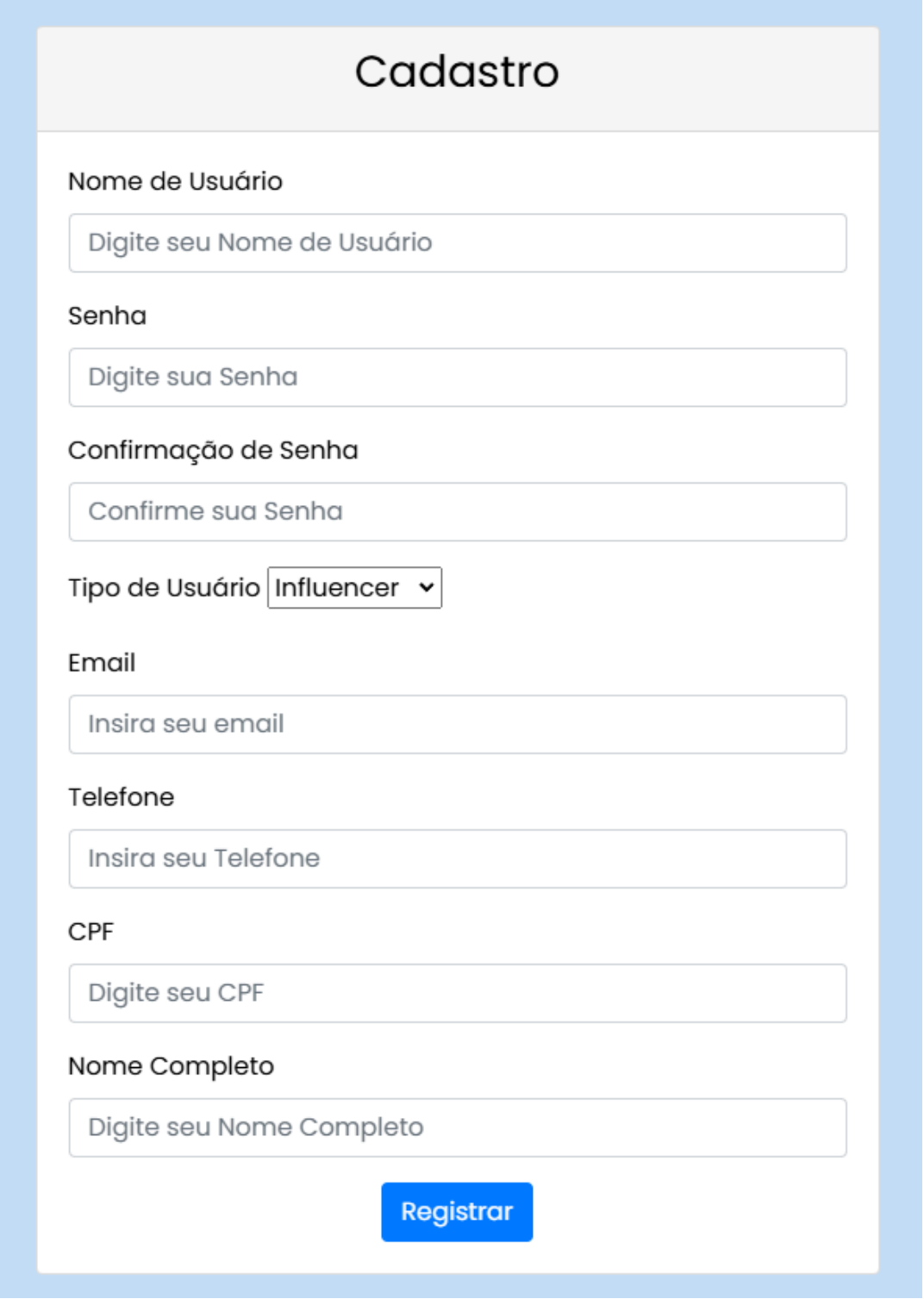
Link - Mapa de Navegacao.drawio.pdf

11. Descrição de Casos de Uso

UC01 – Cadastro de Influencer

Descrição de Casos de Uso - UC01 – Cadastro de Influencer



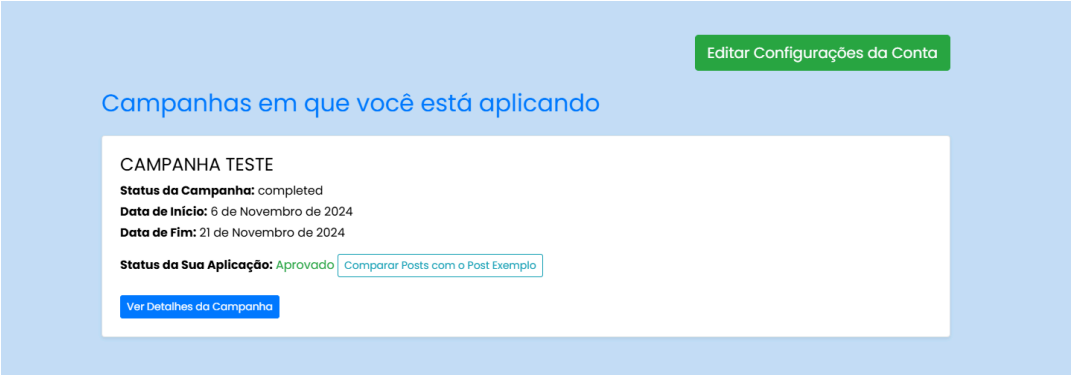


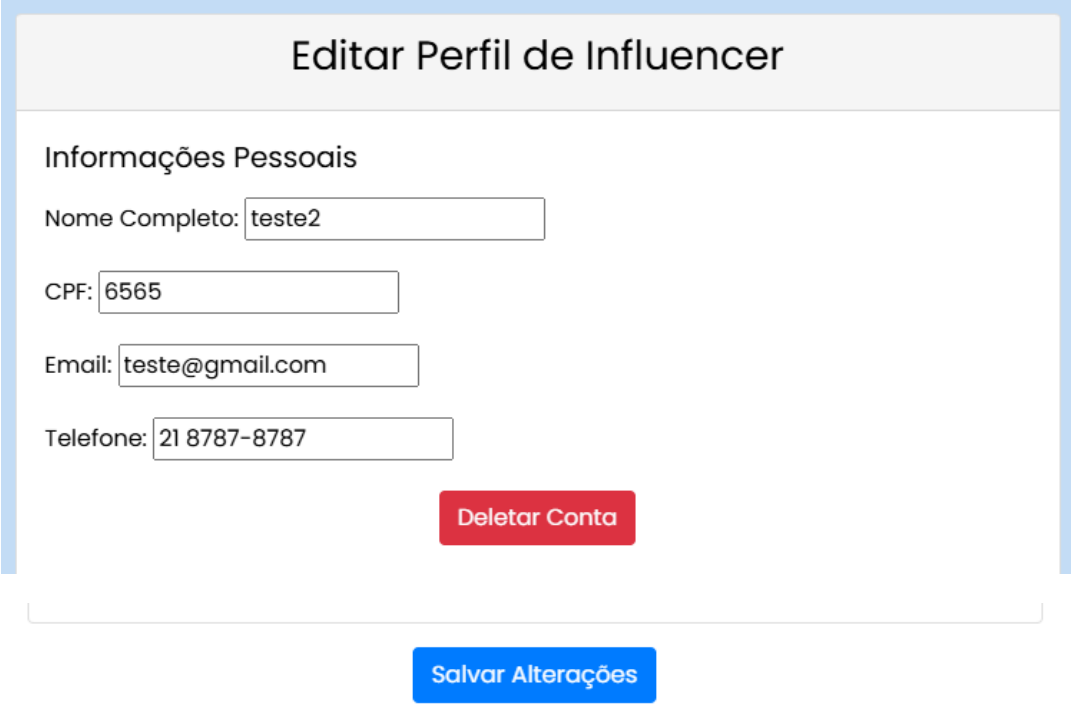
Nome:	Cadastro de Influencer
Objetivo:	Permitir que um novo Influencer se cadastre no sistema.
Atores:	Influencer
Pré-condição:	O usuário não está logado no sistema.
Pós-condição (cenário de sucesso):	Novo Influencer cadastrado no sistema.
Pós-condição (cenário de insucesso):	Influencer não cadastrado.
Trigger:	O usuário clica no botão "Cadastrar" na tela de login ou na tela de home.

Fluxo Principal:	1. Sistema exibe formulário de cadastro. 2. Usuário preenche as informações obrigatórias (Nome, CPF, Login, Senha.). Outros dados como (id, nota, nr_avaliacoes e dt_inclusao serão gerados automaticamente pelo sistema. 3. Usuário clica em "Cadastrar". 4. Sistema valida as informações e cria uma nova conta. 5. Sistema exibe mensagem de confirmação "Cadastro realizado com sucesso".
Fluxos Alternativos:	[A1] Usuário decide cancelar o cadastro. 1. Usuário clica em "Cancelar". 2. Sistema exibe mensagem "Cadastro cancelado". 3. Sistema retorna para a tela de login.
Fluxos de Exceção	[E1] Informações inválidas ou incompletas. 1. Sistema exibe mensagem de erro "Informações inválidas ou incompletas". 2. Sistema solicita que o usuário revise as informações e tente novamente.
Pontos de Extensão:	
Regras de Negócio	RN01: CPF deve ser único e válido. RN02: Login deve ser único. RN03: Senha deve cumprir requisitos mínimos de segurança.

UC02 – Alterar Informações do Influencer

Descrição de Casos de Uso - UC02 – Alterar Informações do Influencer





Nome:	Alterar Informações do Influencer
Objetivo:	Permitir que o Influencer altere suas informações pessoais e de perfil.
Atores:	Influencer.
Pré-condição:	Influencer está logado no sistema.
Pós-condição (cenário de sucesso):	Informações do Influencer são atualizadas.
Pós-condição (cenário de insucesso):	Informações do Influencer não são atualizadas.

Trigger:	Influencer clica no botão "Editar Configurações da Conta" na tela de perfil.
Fluxo Principal:	1. Sistema exibe formulário com as informações atuais do Influencer. 2. Influencer altera as informações desejadas. 3. Influencer clica em "Salvar Alterações". 4. Sistema valida as informações e salva as alterações. 5. Sistema exibe mensagem de confirmação "Informações atualizadas com sucesso".
Fluxos Alternativo:	[A1] Influencer decide cancelar a alteração. 1. Influencer clica em "Cancelar". 2. Sistema exibe mensagem "Alteração cancelada". 3. Sistema retorna para a tela anterior.
Fluxos de Exceção	[E1] Informações inválidas ou incompletas. 1. Sistema exibe mensagem de erro "Informações inválidas ou incompletas". 2. Sistema solicita que o Influencer revise as informações e tente novamente.
Pontos de Extensão:	
Regras de Negócio	RN01: Informações obrigatórias devem ser preenchidas corretamente. RN02: O email deve ser único e válido. RN03: O Influencer não pode alterar seu CPF após o cadastro inicial.

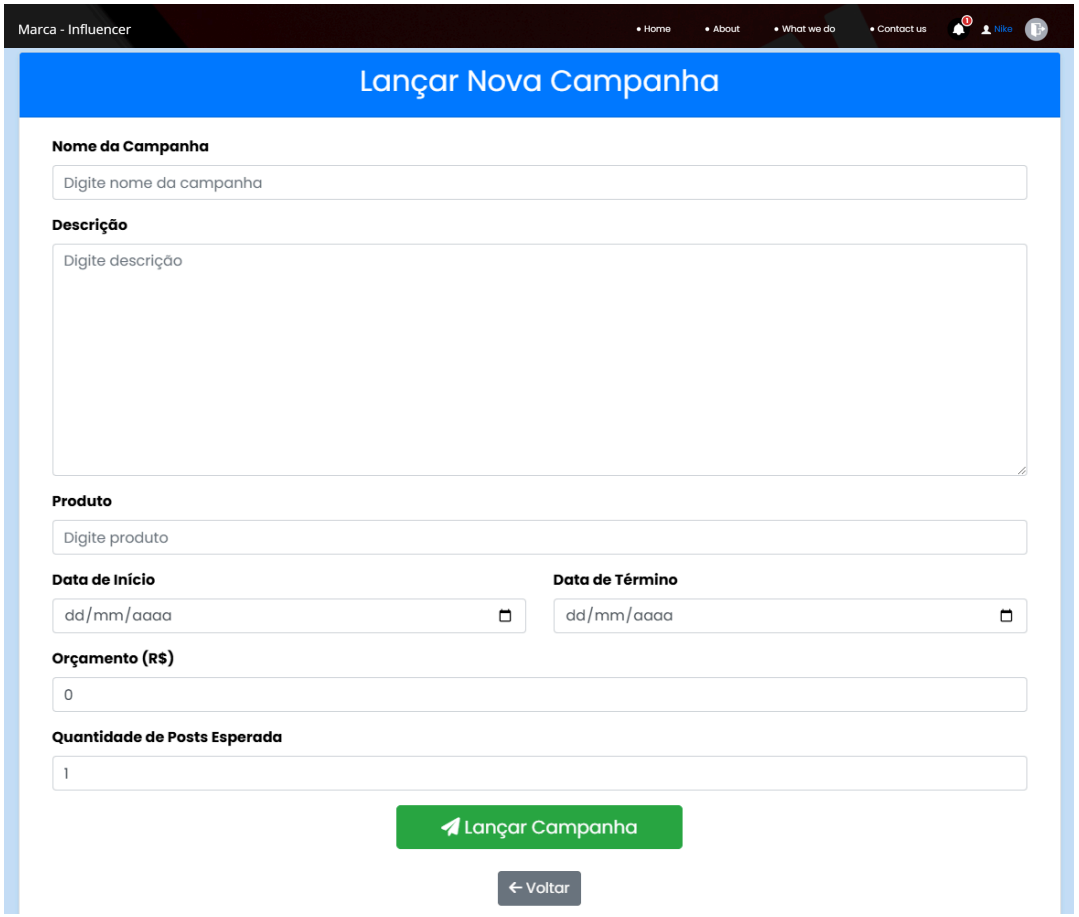
UC03 – Deleção de Conta do Influencer

Descrição de Casos de Uso - UC03 – Deleção de Conta do Influencer

Nome:	Deleção de Conta do Influencer
Objetivo:	Permitir que o Influencer exclua sua conta do sistema.
Atores:	Influencer
Pré-condição:	Influencer está logado no sistema.
Pós-condição (cenário de sucesso):	Conta do Influencer excluída do sistema.
Pós-condição (cenário de insucesso):	Conta do Influencer não excluída.
Trigger:	Influencer clica no botão "Excluir Conta" na tela editar configurações de conta.
Fluxo Principal:	1. Sistema exibe mensagem de confirmação "Tem certeza que deseja excluir sua conta?". 2. Influencer clica em "Confirmar". 3. Sistema exclui a conta do Influencer. 4. Sistema exibe mensagem de confirmação "Conta excluída com sucesso".
Fluxos Alternativos:	[A1] Influencer decide não excluir a conta. 1. Influencer clica em "Cancelar". 2. Sistema exibe mensagem "Exclusão cancelada". 3. Sistema retorna para a tela de perfil.
Fluxos de Exceção	[E1] Falha ao excluir a conta. 1. Sistema exibe mensagem de erro "Erro ao excluir a conta". 2. Sistema solicita que o Influencer tente novamente mais tarde.
Pontos de Extensão:	
Regras de Negócio	RN01: Contas excluídas devem ser removidas permanentemente do sistema. RN02: Dados relacionados ao Influencer (posts, avaliações, etc.) devem ser mantidos para fins de auditoria.

UC04 – Lançar campanha

Descrição de Casos de Uso - UC04 – Lançar campanha

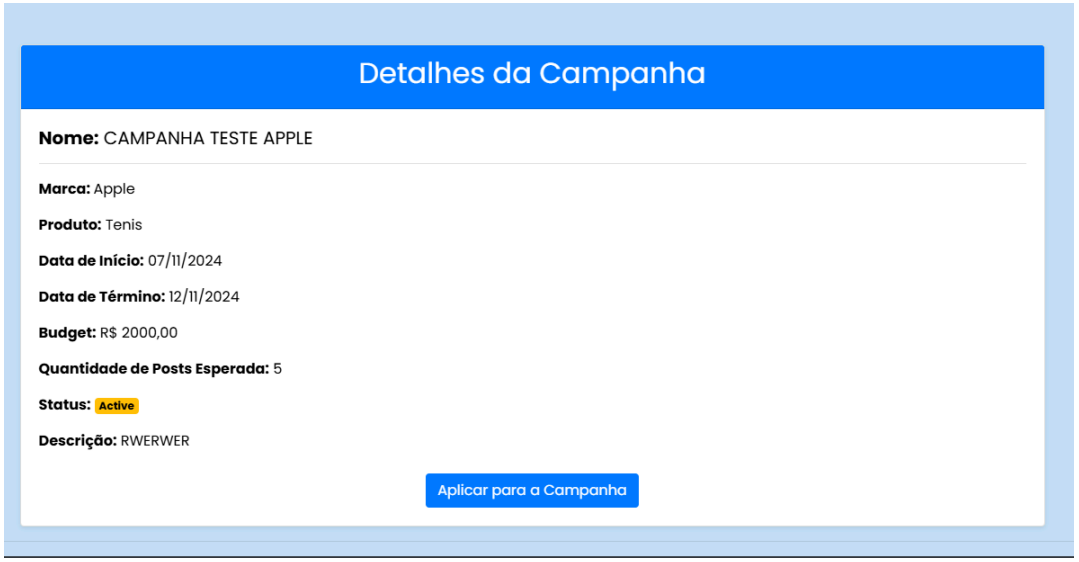


Nome:	Lançar campanha
Objetivo:	Lançar campanha para divulgação de produtos pelos influencers.
Atores:	Marca
Pré-condição:	Marca estar logada no sistema Produto(s) da campanha estarem cadastrados no sistema
Pós-condição (cenário de sucesso):	Campanha lançada
Pós-condição (cenário de insucesso):	Campanha não lançada
Trigger:	Botão de 'Lançar Nova Campanha' é clicado na tela home
Fluxo Principal:	1. Sistema apresenta tela exibindo os campos 'Id', 'Descrição', 'Orçamento', 'Data de início', 'Data de término', 'Produto' e 'Quantidade de Posts Esperada'. 2. Marca preenche 'Descrição', 'Orçamento', 'Data de início', 'Data de término', 'Produto' e 'Quantidade de Posts Esperada'. [RN1] [RN2] [RN3] [RN4][RN5] [A1] 3. Marca clica no botão 'Confirmar'. [A3] 4. Sistema salva as informações e exibe mensagem de sucesso. 5. Sistema fecha a tela de aplicação e volta para a tela de campanhas.
Fluxos Alternativos:	[A1] Marca deseja cancelar lançamento de campanha. 1. Marca clica no botão 'Cancelar'. 2. Sistema encerra tela de Lançamento e retorna à tela de Campanha. [A3] Marca deseja corrigir informações. 1. Marca clica na opção 'Corrigir dados'. 2. Sistema volta à etapa 1 do Fluxo Principal.

Fluxos de Exceção	[E1] Marca preenche dados de forma incorreta. 1. Sistema exibe mensagem de erro e sinaliza campos preenchidos incorretamente. 2. Sistema retorna à etapa 1 do fluxo principal.
Pontos de Extensão:	Incluir restrições de aplicação para campanha com base em indicadores do perfil do instagram e do influencer.
Regras de Negócio	[RN1] 'Id' é um autonumber. [RN2] 'Data de início' é a data de lançamento da campanha. [RN3] 'Orçamento' é um número maior que 0. [RN4] Data de término deve ser posterior ao dia de lançamento da campanha. [RN5] A Quantidade de Posts Esperada deve ser maior que 0.

UC05 - Aplicar para campanha

Descrição de Casos de Uso - UC05 – Aplicar para campanha

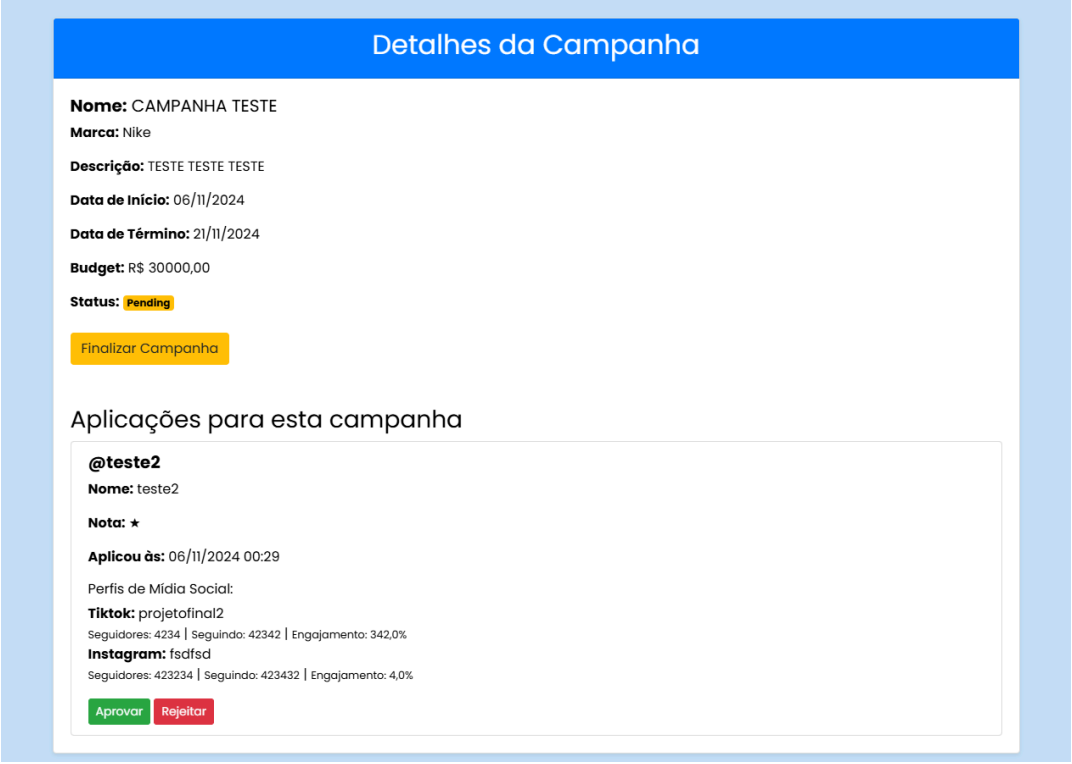


Nome:	Aplicar para campanha
Objetivo:	Aplicar para campanha lançada
Atores:	Influencer
Pré-condição:	Influencer está logado na plataforma. Campanha foi lançada pela marca e influencer está na tela da campanha. Influencer é da mesma categoria da Marca.
Pós-condição (cenário de sucesso):	Aplicação realizada
Pós-condição (cenário de insucesso):	Aplicação não realizada
Trigger:	Botão de 'Aplicar' na tela de detalhes da campanha é selecionado.
Fluxo Principal:	1. Sistema apresenta tela contendo 'Orçamento', 'Data de início' e 'Data de término' [A1] 2. O influencer clica na opção 'Aplicar para Campanha' [E1] 3. Sistema salva as informações e exibe mensagem de sucesso. 4. Sistema fecha a tela de aplicação e volta para a tela de campanhas
Fluxos Alternativos:	[A1] Influencer desiste da aplicação 1. Influencer clica no botão de 'Voltar' 2. Sistema fecha tela da aplicação e retorna para a tela de campanhas.
Fluxos de Exceção	[E1] Influencer já se encontra aplicado para essa campanha 1. Sistema exibe mensagem de erro e sinaliza que o influencer já está aplicado na campanha em questão.
Pontos de Extensão:	

Regras de Negócio	

UC06 – Aprovar aplicação

Descrição de Casos de Uso - UC06 – Aprovar aplicação

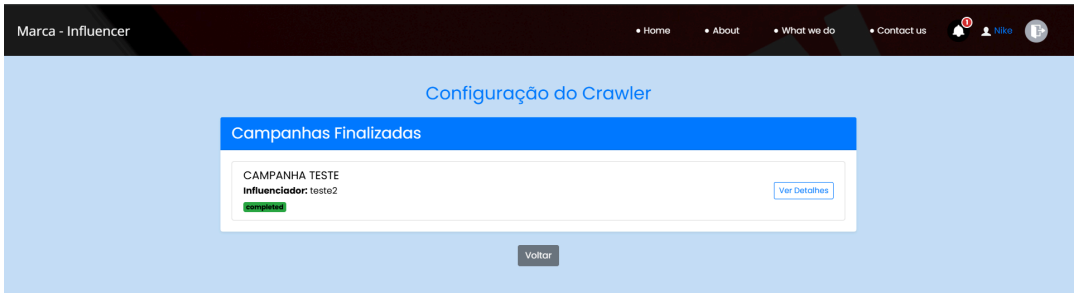


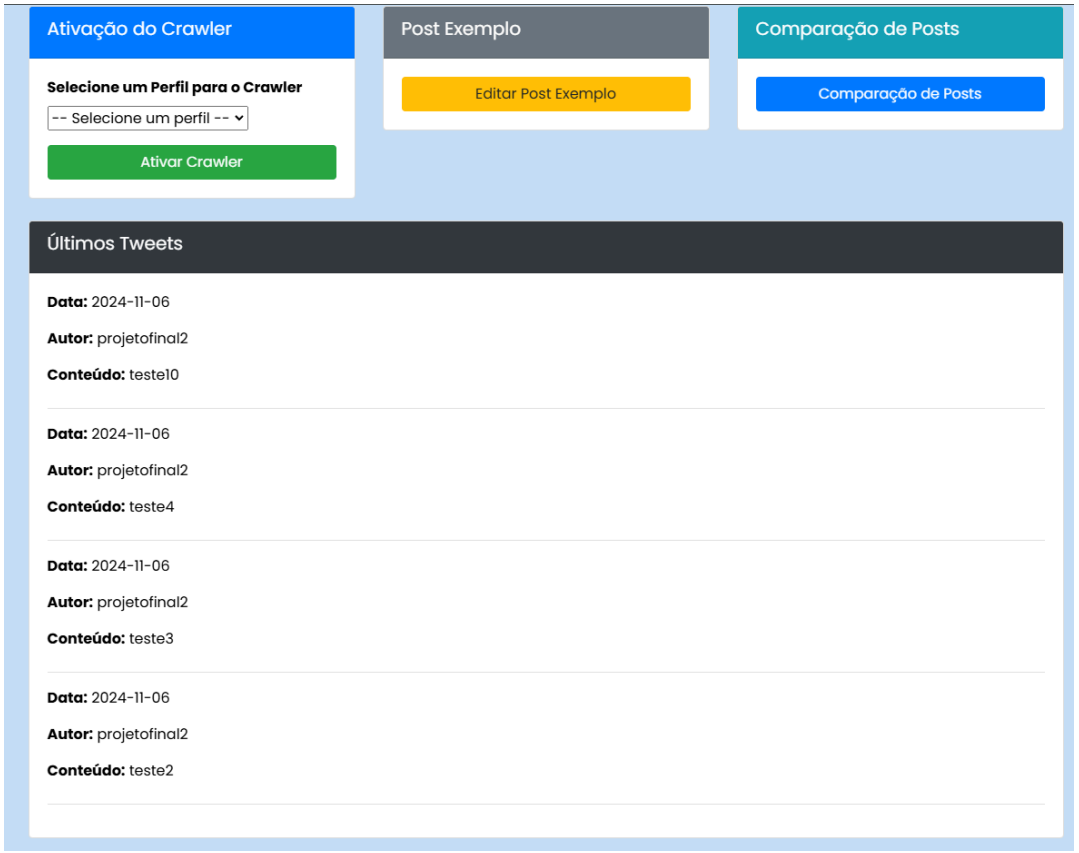
Nome:	Aprovar aplicação
Objetivo:	Aprovar aplicação submetida por influencer em referência a uma campanha da marca.
Atores:	Marca.
Pré-condição:	Marca estar logada no sistema. Existir aplicação para campanha lançada pela marca.
Pós-condição (cenário de sucesso):	Aplicação aprovada.
Pós-condição (cenário de insucesso):	Aplicação não aprovada.
Trigger:	Botão de 'Aprovar' no quadro de aplicações pendentes da tela de detalhes da campanha é clicado.
Fluxo Principal:	1. Sistema exibe informações da aplicação, contendo: nome do influencer, nota do influencer, Perfis do influencer, número de seguidores, taxa de engajamento e data de aplicação. 2. Sistema exibe opção de "Aprovar" ou "Rejeitar". 3. Marca clica no botão "Aprovar" [A1]

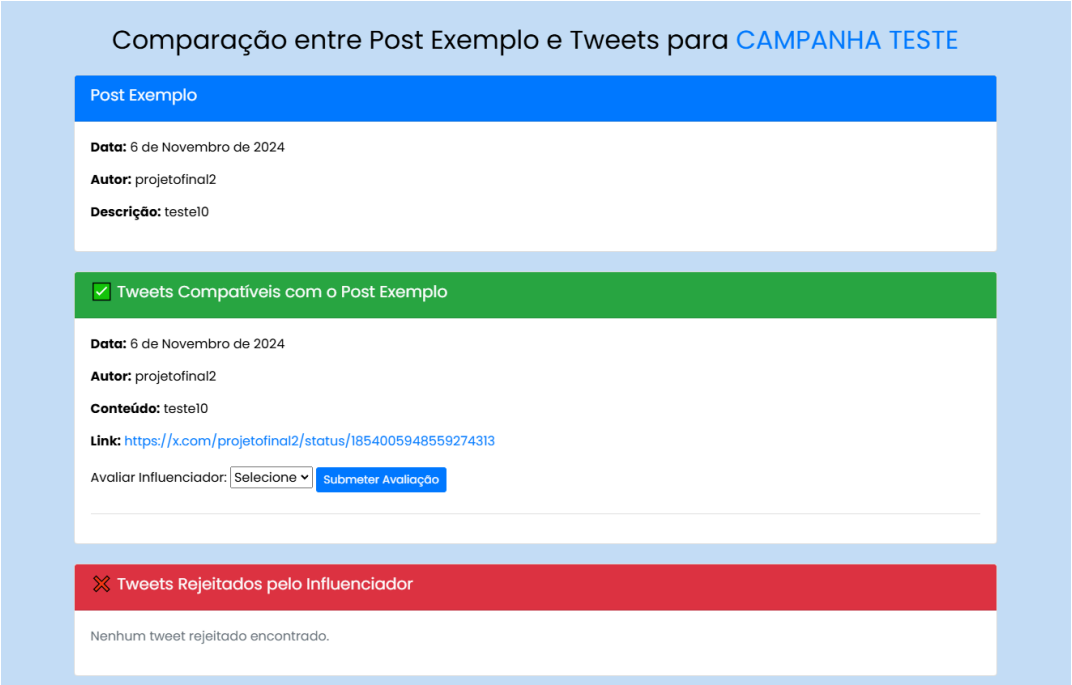
	4. Sistema envia notificação ao Influencer de que a notificação foi aprovada. A notificação pode ser acessada no canto superior direito. 5. Sistema exibe mensagem confirmando à marca a aprovação e contendo o contato do Influencer. 6. Sistema fecha a tela de aprovação.
Fluxos Alternativos:	[A1] Marca não deseja aprovar a solicitação 1. Marca clica no botão “Não”. 2. Sistema exibe a mensagem “Aprovação cancelada”. 3. Sistema fecha a tela de aprovação.
Fluxos de Exceção	
Pontos de Extensão:	Incluir comparativo no quadro de aplicações quanto à viabilidade financeira da aplicação.
Regras de Negócio	

UC07 – Verificar Post

Descrição de Casos de Uso - UC07 – Verificar Post



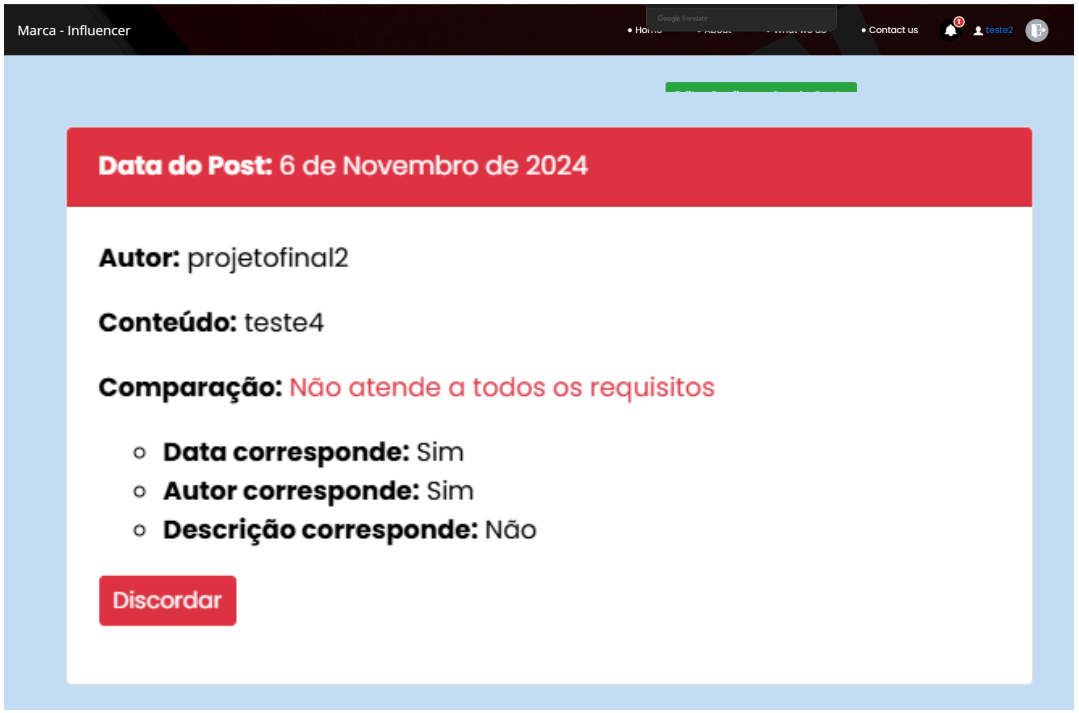




Nome:	Verificar Post
Objetivo:	Automatizar a verificação de postagens de influencers em redes sociais utilizando um crawler para capturar e validar informações da postagem.
Atores:	Tempo
Pré-condição:	O crawler está configurado e funcional. Marca está logada no sistema.
Pós-condição (cenário de sucesso):	A postagem é validada com sucesso pelo crawler e aprovada pelo sistema.
Pós-condição (cenário de insucesso):	Postagem não é validada ou o crawler não conseguiu capturar as informações corretamente.
Trigger:	A marca escolhe o perfil de social media do influencer e dispara o crawler clicando no botão de "Ativar Crawler"
Fluxo Principal:	1. O sistema dispara o crawler. 2. O crawler verifica os critérios de qualidade, número de curtidas, comentários, engajamento e que a postagem está de acordo com o conteúdo esperado da campanha. [RN1] [RN2] [RN3] 3. Sistema exibe os resultados da verificação automática realizada pelo crawler para a marca.[E1] 4. Sistema envia uma notificação para a Marca sinalizando que um post compatível foi encontrado.
Fluxos Alternativos:	
Fluxos de Exceção	**[E1] Falha ao capturar as informações da postagem:** 1. Sistema exibe mensagem de erro "Erro ao capturar informações da postagem". 2. Marca pode tentar novamente ou cancelar o processo de verificação.
Pontos de Extensão:	
Regras de Negócio	**RN1**: A postagem deve seguir as diretrizes de conteúdo estabelecidas pela campanha. **RN2**: O crawler deve verificar se a postagem foi feita no perfil correto do influencer.

UC08 – Discordar de post inicialmente recusado

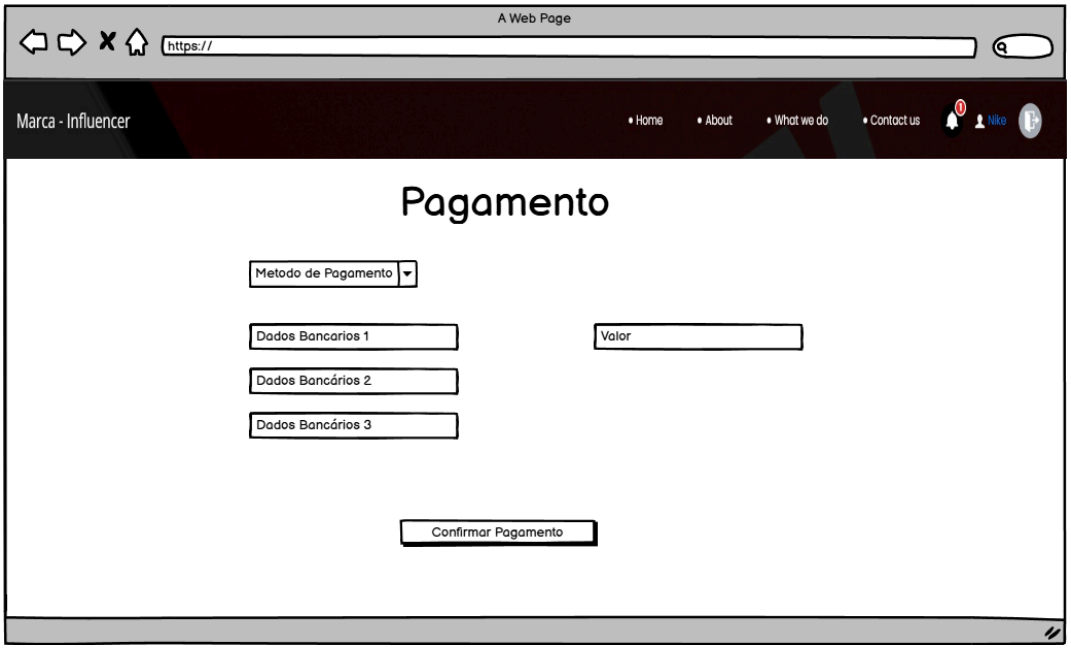
Descrição de Casos de Uso - UC08 – Discordar de post inicialmente recusado



Nome:	Discordar de post inicialmente recusado
Objetivo:	Influencer questionar a rejeição do Crawler.
Atores:	Influencer
Pré-condição:	Influencer está logada no sistema. Crawler verificou o post.
Pós-condição (cenário de sucesso):	Postagem discordada.
Pós-condição (cenário de insucesso):	Postagem não discordada.
Trigger:	Marca clica no botão "Discordar" na tela de Comparar Posts com o Post 'template'.
Fluxo Principal:	1. Sistema exibe a postagem do influencer. [RN1] 2. Influecer verifica os detalhes da postagem. 3. Influencer clica em "Discordar". 4. Sistema envia notificação para a Marca.
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Falha ao dicordar postagem. 1. Sistema exibe mensagem de erro "Erro ao aprovar a postagem". 2. Influencer pode tentar aprovar novamente.
Pontos de Extensão:	
Regras de Negócio	RN1: Postagens devem cumprir os critérios de qualidade estabelecidos pela marca. RN2: Feedbacks de rejeição devem ser detalhados para melhorias futuras.

UC09 – Realiza Pagamento

Descrição de Casos de Uso - UC09 – Realiza Pagamento	
Nome:	Realiza pagamento
Objetivo:	Efetuar o pagamento ao influencer após a postagem.
Atores:	Marca
Pré-condição:	Marca está logada no sistema. Postagem foi realizada pelo influencer. Informações bancárias do influencer estão disponíveis.
Pós-condição (cenário de sucesso):	Pagamento realizado e confirmado.
Pós-condição (cenário de insucesso):	Pagamento não realizado.
Trigger:	Marca clica no botão "Realizar Pagamento"..
Fluxo Principal:	1. Sistema verifica se meta foi atendida.[E1] 2. Sistema exibe informações do pagamento, incluindo valor e dados bancários do influencer.[RN1][RN2][RN3] 3. Marca confirma as informações e clica em "Confirmar Pagamento". 4. Sistema processa o pagamento.[E2] 5. Sistema exibe mensagem de confirmação do pagamento. 6. Sistema atualiza o status do pagamento para "Realizado".
Fluxos Alternativos:	[A1] Marca cancela o pagamento. 1. Marca clica em "Cancelar". 2. Sistema exibe mensagem "Pagamento cancelado". 3. Sistema retorna para a tela anterior.
Fluxos de Exceção	[E1] Meta não foi atendida. 1. Sistema exibe mensagem dizendo “Meta não foi atendida.” 2. Sistema informa Influencer. [E2] Falha no processamento do pagamento. 1. Sistema exibe mensagem de erro "Falha no processamento do pagamento". 2. Sistema solicita que a marca verifique as informações e tente novamente. 3. Marca pode optar por tentar novamente ou cancelar o pagamento.
Pontos de Extensão:	
Regras de Negócio	Regras de Negócio: [RN1] Pagamentos só podem ser realizados para postagens aprovadas. [RN2]: Informações bancárias do influencer devem estar completas e validadas. [RN3]: Pagamentos devem ser registrados no sistema.


A Web Page
https://
Marca - Influencer
• Home
• About
• What we do
• Contact us
Nike
Pagamento
Metodo de Pagamento
Dados Bancarios 1
Dados Bancários 2
Dados Bancários 3
Valor
Confirmar Pagamento

UC10 – Avalia influencer

Descrição de Casos de Uso - UC10 – Avalia influencer

```python
# Compara com o Post Exemplo
if post_exemplo:
    comparison_result = post_exemplo.comparar_com_post(post)
    post.atende_requisitos = comparison_result['all_match']
    post.save()

    # Se o post atende aos requisitos, incrementa o post_count do influenciador
    if post.atende_requisitos:
        influencer_rating, created = InfluencerRating.objects.get_or_create(
            influencer=influencer,
            campaign=campaign,
            defaults={'post_count': 0}
        )
        influencer_rating.post_count += 1
        influencer_rating.save()
```

```python
@receiver(post_save, sender=Campaign)
def calculate_influencer_ratings(sender, instance, **kwargs):
    # Verifica se o status da campanha foi atualizado para 'finished'
    if instance.status == 'finished':
        approved_applications = instance.applications.filter(application_status='approved')

        # Para cada influenciador aprovado, calcula a nota com base no número de posts
        for application in approved_applications:
            influencer = application.influencer
            influencer_posts_count = instance.posts.filter(influencer=influencer).count()
            required_posts = instance.required_posts

            # Calcula a nota atual da campanha (limita ao máximo de `required_posts`)
            current_campaign_rating = min(influencer_posts_count, required_posts)

            # Cria um registro da nota para a campanha atual
            InfluencerRating.objects.create(
                influencer=influencer,
                campaign=instance,
                rating=current_campaign_rating,
                post_count = influencer_posts_count
            )

            # Calcula a média das notas considerando as notas anteriores e a nova nota
            new_average_rating = InfluencerRating.objects.filter(influencer=influencer).aggregate(avg_rating=Avg('rating'))['avg_rating']

            # Atualiza a média no modelo de influenciador
            influencer.average_rating = new_average_rating
            influencer.save()
```

Nome de Usuario: teste2

Nome: teste2

Nota Média: 2,0

Aplicou às: 06/11/2024 21:02

Perfis de Mídia Social:

Twitter: projetofinal2

Seguidores: 4234 | Seguindo: 42342 | Engajamento: 342,0%

Instagram: fsdfsd

Seguidores: 423234 | Seguindo: 423432 | Engajamento: 4,0%

Approved

Nome:	Avalia influencer
Objetivo:	Calcular e registrar automaticamente a nota de desempenho do influencer após o término da campanha, com base na quantidade de posts realizados em comparação ao solicitado pela marca.
Atores:	Sistema
Pré-condição:	A Campanha foi finalizada.
Pós-condição (cenário de sucesso):	A nota do influencer é calculada e registrada no sistema.
Pós-condição (cenário de insucesso):	Nota não registrada no sistema.
Trigger:	A campanha muda de status para "finalizada".
Fluxo Principal:	1. O sistema detecta que a campanha foi finalizada e verifica o número de posts realizados pelo influencer. 2. O sistema calcula a nota do influencer com base na fórmula: 3. Nota = min(post_count, required_posts), onde post_count é o número de posts realizados e required_posts é a quantidade solicitada pela marca. 4. O sistema atualiza o perfil do influencer com a média de todas as avaliações de campanhas concluídas. 5. O sistema salva a avaliação e exibe uma mensagem de confirmação.
Fluxos Alternativos:	[A1] Nenhuma avaliação: 1. Se a marca não especificou uma quantidade de posts mínima, o sistema exibe uma mensagem de que a avaliação não é aplicável.
Fluxos de Exceção	[E1] Falha ao registrar a avaliação: 1. O sistema exibe a mensagem de erro "Erro ao registrar a avaliação". 2. O sistema poderá tentar registrar novamente ou notificar o administrador.
Pontos de Extensão:	
Regras de Negócio	[RN01]: A avaliação só deve ocorrer após a conclusão da campanha. [RN02]: A avaliação é baseada na proporção de posts realizados em relação ao solicitado pela campanha. [RN03]: As avaliações devem ser registradas no sistema e atualizadas no perfil do influencer para referência futura.

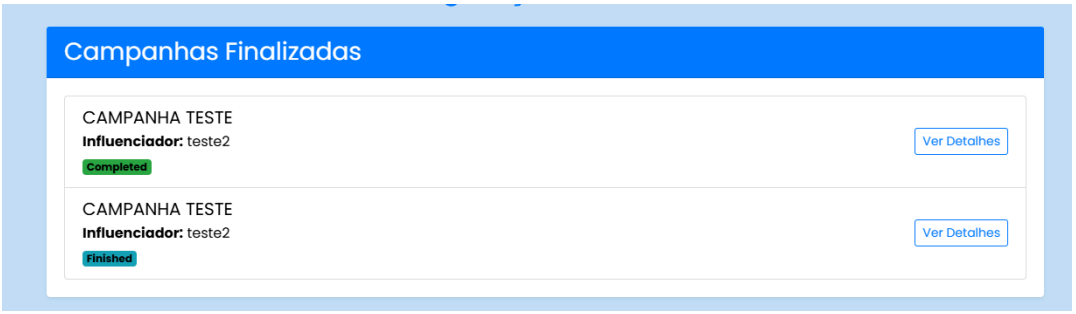
12. Outras Funcionalidades do Projeto

Além das funcionalidades já apresentadas, o sistema oferece diversas outras ferramentas voltadas para otimizar a interação entre marcas e influenciadores e garantir o bom funcionamento da aplicação.
Foram priorizadas as descrições dos casos de uso acima (que estão no capítulo 12) e as funcionalidades abaixo não foram descritas como casos de uso conforme o acordado entre o aluno e o professor.

12.1 Funcionalidades para o Usuário Marca

12.1.1 Gerenciar Campanha Completas ou Finalizadas

- Visualizar as campanhas completas ou finalizadas.

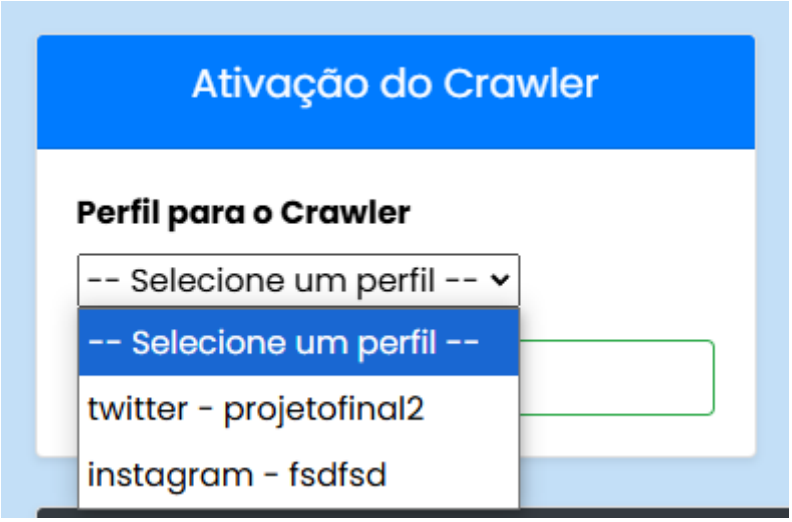


12.1.2 Definir Posts Exemplo

- Criar e editar posts exemplo que servirão como referência para avaliar a compatibilidade dos posts dos influenciadores.

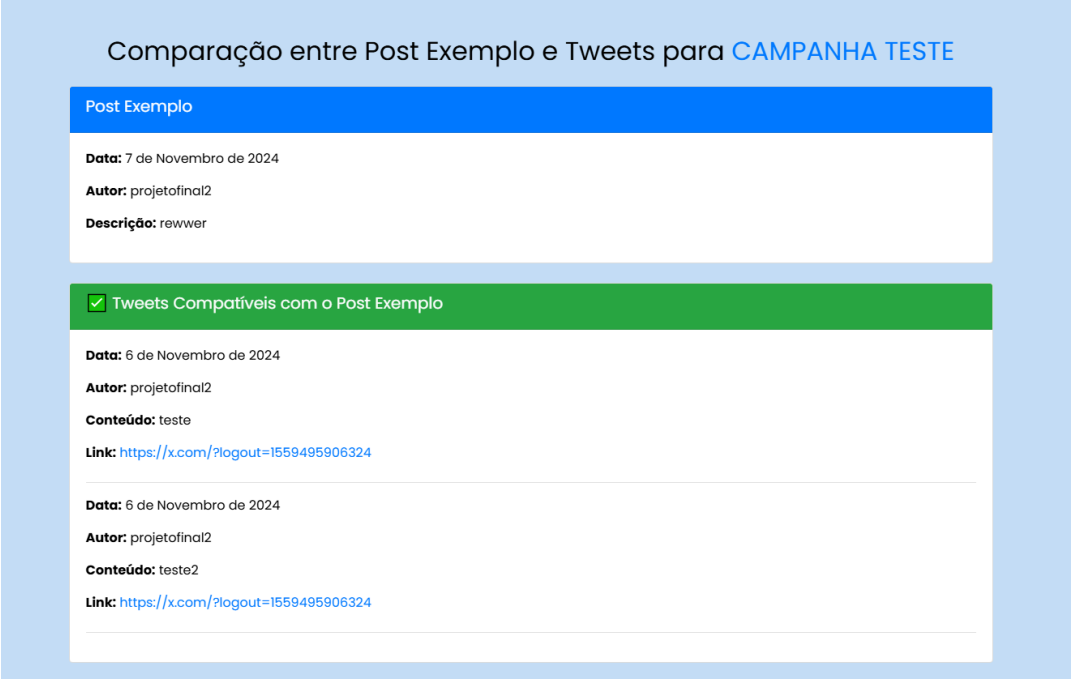
12.1.3 Configurar Crawler

- Configurar e ativar o crawler para coletar dados de redes sociais, associando a campanha a perfis específicos de influenciadores.



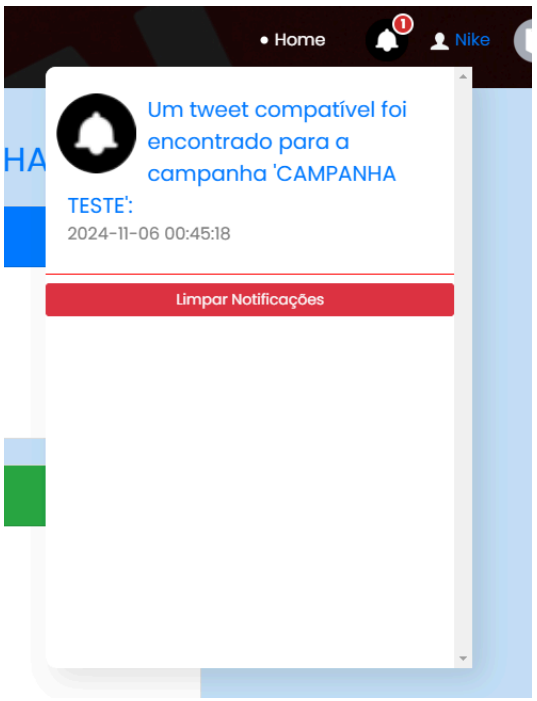
12.1.4 Visualizar Relatório de Campanha

- Acessar um relatório completo da campanha, incluindo o número de posts compatíveis



12.1.5 Visualizar Notificações

- Acessar um aba contendo todas as notificações



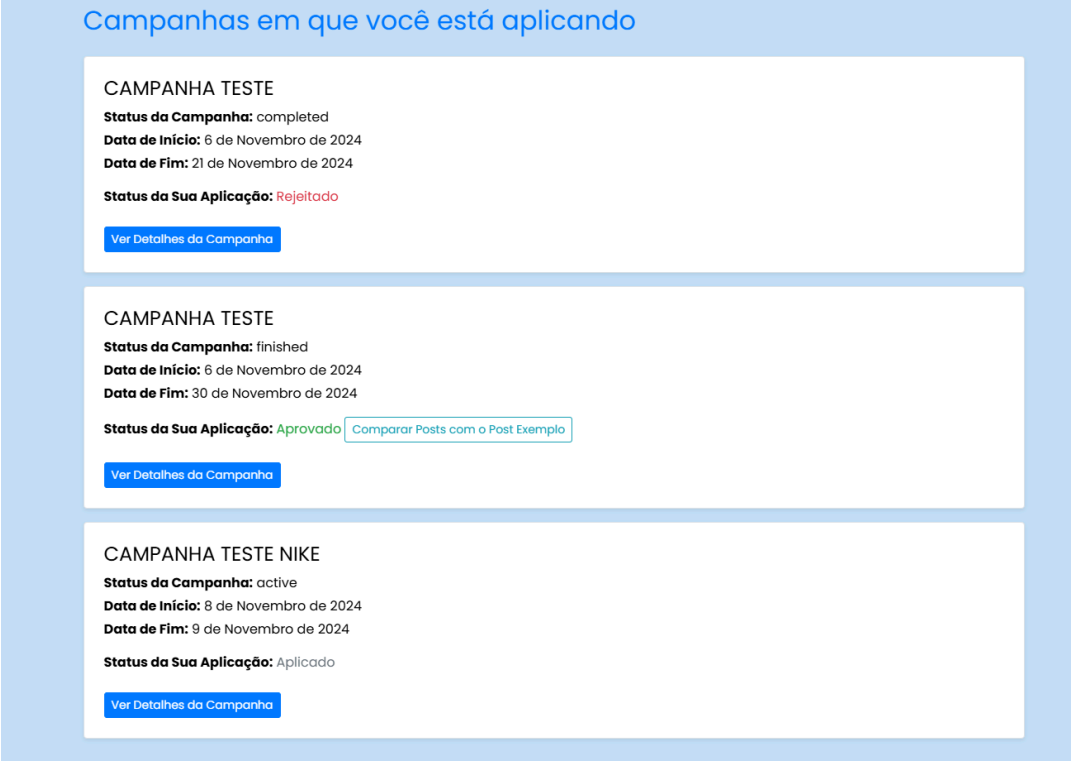
12.2 Funcionalidades para o Usuário Influenciador

12.2.1 Visualizar Campanhas Disponíveis

- Acessar campanhas disponíveis e ver detalhes como nome, descrição, e critérios.

12.2.2 Visualizar Status da Aplicação

- Acompanhar o status de sua aplicação (pendente, aprovada, rejeitada).

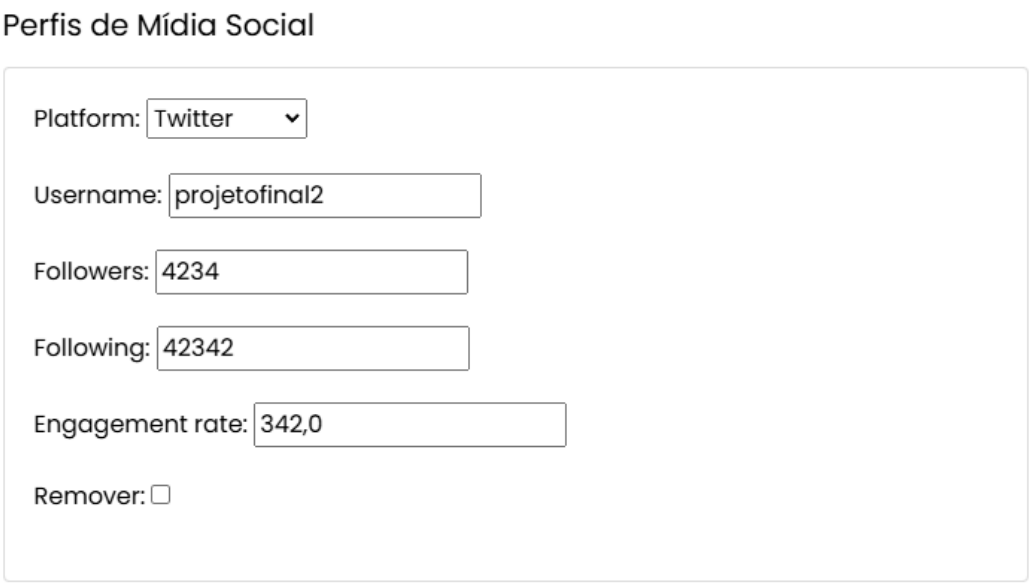


12.2.3 Adicionar um Novo Perfil de Media Social

- Cadastrar um novo perfil de social media.

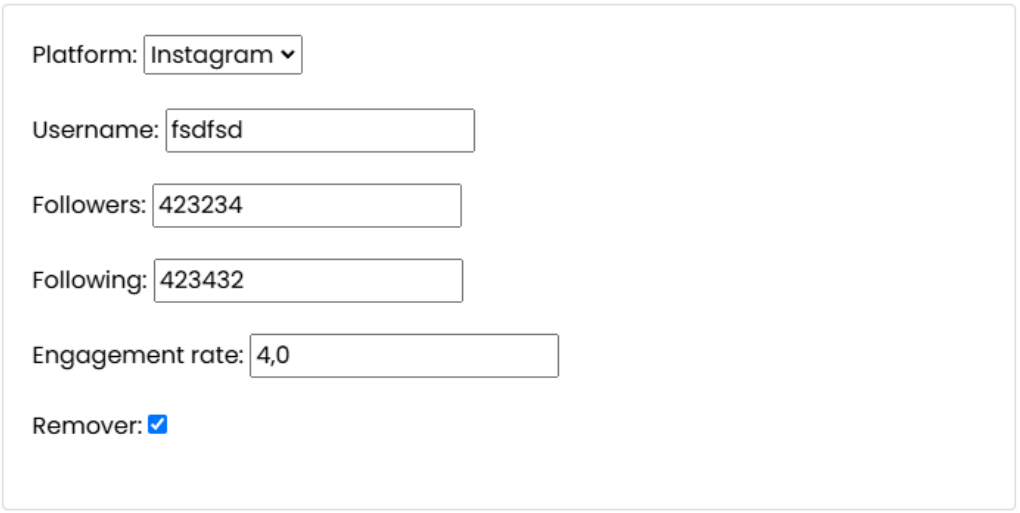
12.2.4 Editar Perfis de Media Social

- Editar e atualizar informações de seus perfis de social media.



12.2.5 Excluir Perfis de Media Social

- Excluir um perfil de social media.



12.2.6 Visualizar Notificações

- Acessar um aba contendo todas as notificações

12.3 Funcionalidades do Sistema

12.3.1 Comparar Posts com Post ‘Template’

- A comparação entre os posts coletados e o Post ‘Template’ verifica a compatibilidade com base em três critérios: data, autor e conteúdo. A data do post é comparada diretamente com a do template para assegurar que foi publicado no período correto. O autor é validado através do nome de perfil de mídia social já associado ao influenciador. Já o conteúdo é analisado para verificar se as palavras-chave definidas na descrição do template estão presentes no texto do post, mesmo que parcialmente. Quando todos esses critérios são atendidos, o post é considerado compatível, assegurando que ele cumpre os requisitos estabelecidos pela campanha.

12.3.2 Atualizar Status da Campanha

- Altera automaticamente o status da campanha para “finalizada” ao alcançar a data final.

12.3.3 Exibir Notificações

- Notifica influenciadores e marcas sobre eventos importantes, como aprovação de aplicações, início e término de campanhas.

12.3.4 Crawler

- O crawler é um script que automatiza a extração de dados de tweets de um perfil específico na plataforma X. Foi utilizado a biblioteca Playwright para automatizar a navegação dentro do site, ele realiza o login, acessa o perfil do influencer e rola a página para carregar novos tweets. Os dados de cada tweet, como ID, autor, conteúdo, data, link, imagem e vídeo, são extraídos e armazenados em um arquivo JSON. Após isso, o script compara os tweets recém-extraídos com os previamente salvos para identificar novos dados. Se houver novos tweets, ele atualiza o arquivo de armazenamento, caso contrário, continua o monitoramento, recarregando a página e repetindo o processo de extração. O fluxo é contínuo, garantindo a captura de novos tweets conforme eles são postados no perfil monitorado.

13. Conceitos das Tecnologias Usadas no Sistema

JavaScript : é uma linguagem de programação interpretada usada para adicionar interatividade a páginas web. Como linguagem estruturada, é frequentemente utilizada no front-end para tornar o conteúdo das páginas mais dinâmico, responder a eventos do usuário e manipular elementos da interface diretamente. [5]

CSS : é a linguagem de estilo usada para formatar a aparência visual dos elementos em uma página web. Ela permite aplicar cores, espaçamento, tipografia e outros estilos a documentos HTML, e auxilia no projeto e áreas no site. [6]

Bootstrap : é um framework open-source de front-end usado para acelerar o desenvolvimento de páginas web responsivas e visualmente atraentes. Ele oferece uma biblioteca consistente de componentes CSS e JavaScript, incluindo botões, formulários e grades de layout, e ajuda a acelerar o desenvolvimento da interface de usuário. [7]

Django : é um framework web modular de alto nível, que é escrito em Python e segue o padrão MVT. O Django acelera o desenvolvimento de aplicações web robustas, oferecendo recursos como autenticação, roteamento de URLs e um ORM integrado que permite que os desenvolvedores se concentrem na lógica do aplicativo. [8]

Python : Python é uma linguagem de programação de alto nível fácil de ler usada para desenvolvimento back-end, automação de tarefas e ciência de dados.

Para web, é a linguagem na qual o Django, Flask e outros frameworks são baseados, e ajuda a desenvolver aplicativos complexos de maneira eficaz. [9]

Playwright: é uma biblioteca escrita para a linguagem Python que permite a navegação controlada e testes de automação em navegadores da web. No contexto do crawler, ele é programado para acessar páginas web e, de modo controlado, grava a captura dos dados em tempo real e repete ações do usuário, como clicks e rolar para pegar informações específicas. [10]

14. Estrutura do Projeto

14.1 Estrutura do Django e o Padrão MVT(Model-View-Template)

Django é um framework de desenvolvimento web que usa Python e permite a criação de aplicações web com um fluxo bem organizado em três camadas: Model, View e Template. Com essa abordagem, uma aplicação aderente pode permitir que Django gerencie o back-end da aplicação de forma eficaz. Simultaneamente, a aplicação é integrada perfeitamente com o front-end. [11]

Model: Representa a camada de dados da aplicação e é responsável por mapear os dados do banco de dados para objetos no código. Cada classe de model no Django representa uma tabela no banco de dados, e suas propriedades são traduzidas em colunas dessa tabela. Essa camada permite que o Django se conecte e manipule dados com consultas SQL automáticas, tornando o processo mais rápido e seguro.

Figura 1 - Arquivo models.py

Fonte: Elaborado pelo autor (2024)

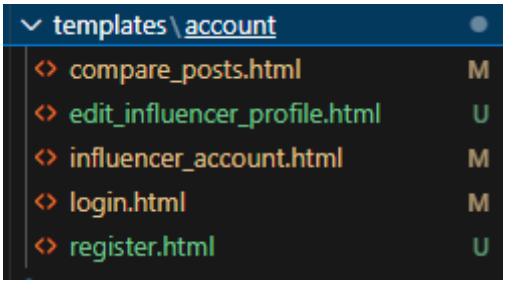
View: Contém a lógica de negócios da aplicação. Uma view é uma função ou classe que processa uma solicitação HTTP, interage com o modelo para obter ou alterar dados e decide o que será exibido ao usuário. As views retornam respostas HTTP, que podem ser renderizadas como páginas HTML, JSON ou outros tipos de dados.

Figura 2 - Arquivo views.py

Fonte: Elaborado pelo autor (2024)

Template: A camada de apresentação da aplicação, onde ficam os arquivos HTML que serão enviados para o navegador. Nos templates, você pode usar a linguagem de template do Django para integrar dados dinâmicos vindos do back-end, exibindo, por exemplo, o conteúdo de variáveis, iterando sobre listas, ou aplicando condições lógicas.

Figura 3 - Pasta de templates



Fonte: Elaborado pelo autor (2024)

14.2 Conexão entre Front-end e Back-end no Django

O Django permite conectar o front-end e o back-end de forma fluida, facilitando a exibição de dados dinâmicos em páginas web e manipulando a lógica do lado do servidor. Nesse projeto foi utilizado a Renderização de Templates no Django (Back-end Rendering), que o Django usa seus templates para renderizar HTML no servidor e enviar a página completa para o navegador. A página HTML já vem pronta para exibição e carrega os dados diretamente do servidor.

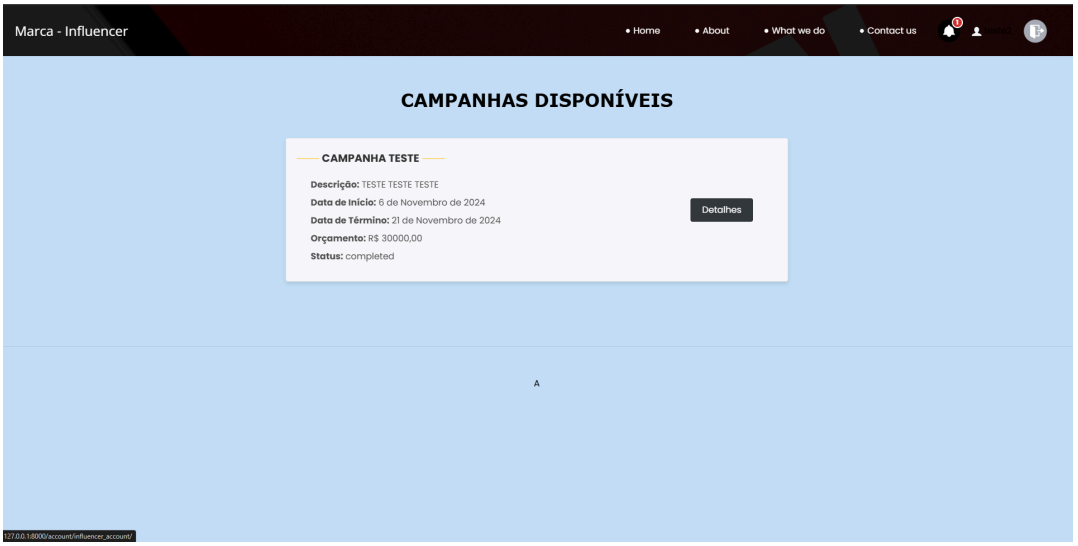
Como funciona: Quando o usuário acessa uma URL da aplicação, uma requisição HTTP é enviada para o servidor. O Django processa essa requisição na camada de view, onde obtém dados necessários do model. Depois, a view passa esses dados para um template. O Django renderiza o template com os dados, e o HTML final é enviado para o navegador do usuário.

Exemplo: Acessar o Perfil do Usuário

14.2.1 Interação com o Ícone do Perfil

- O usuário clica no ícone de perfil, que tem um link configurado para a URL /account/influencer_account/.
- Essa URL é associada a uma *view* Django que carrega a página de perfil do influenciador.

Figura 4 - Tela home da aplicação

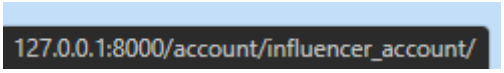


Fonte: Elaborado pelo autor (2024)

14.2.2 Requisição HTTP GET para o Servidor

- Ao clicar, o navegador do usuário envia uma **requisição HTTP GET** para o servidor com a URL /account/influencer_account/.
- Essa requisição inclui informações sobre o usuário autenticado (caso o usuário esteja logado).

Figura 5 - Requisição Url do perfil do usuário

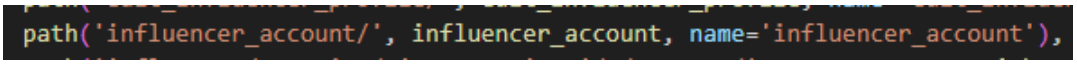


Fonte: Elaborado pelo autor (2024)

14.2.3 Django URL Configuration (urls.py)

- No Django, a URL /account/influencer_account/ é configurada no arquivo urls.py do aplicativo account.
- Django identifica qual *view function* (ou classe) deve ser chamada para processar essa URL, geralmente com algo assim:

Figura 6 - Path da url no arquivo urls.py



Fonte: Elaborado pelo autor (2024)

14.2.4 Execução da View Function

- O Django chama a função *view* associada, por exemplo, influencer_account(request), passando o objeto request que contém informações sobre a requisição HTTP.
- A *view* realiza várias tarefas para preparar os dados a serem exibidos na página, como verificar se o usuário está autenticado.

Figura 7 - Metodo influencer_account na views.py

```python
def influencer_account(request):
    if not request.user.is_authenticated or not hasattr(request.user, 'influencer'):
        return redirect('login')

    influencer = request.user.influencer
    applications = Application.objects.filter(influencer=influencer)

    campaigns_with_status = [
        {'campaign': app.campaign, 'status': app.application_status}
        for app in applications
    ]

    social_profiles = influencer.social_media_profiles.all()

    return render(
        request,
        'account/influencer_account.html',
        {
            'campaigns_with_status': campaigns_with_status,
            'social_profiles': social_profiles
        }
    )
```

Fonte: Elaborado pelo autor (2024)

14.2.5 Consulta ao Banco de Dados

- A *view* consulta o banco de dados para buscar as informações do perfil do influenciador, como nome e perfis de social media.
- Essas consultas são feitas usando o ORM do Django, por exemplo:

Figura 8 - Consultas no banco de dados

```
influencer = request.user.influencer
applications = Application.objects.filter(influencer=influencer)
```

Fonte: Elaborado pelo autor (2024)

14.2.6 Processamento dos Dados

- A *view* processa os dados recebidos
- Os dados são então preparados para serem enviados ao template HTML como *contexto*.

14.2.7 Renderização do Template HTML

- A *view* usa a função render() para combinar os dados com o template HTML associado como o influencer_account.html.
- O Django substitui variáveis no template pelos dados do *contexto*, criando o HTML final que será enviado ao navegador:

Figura 9 - Retorno com os dados para o template na views.py

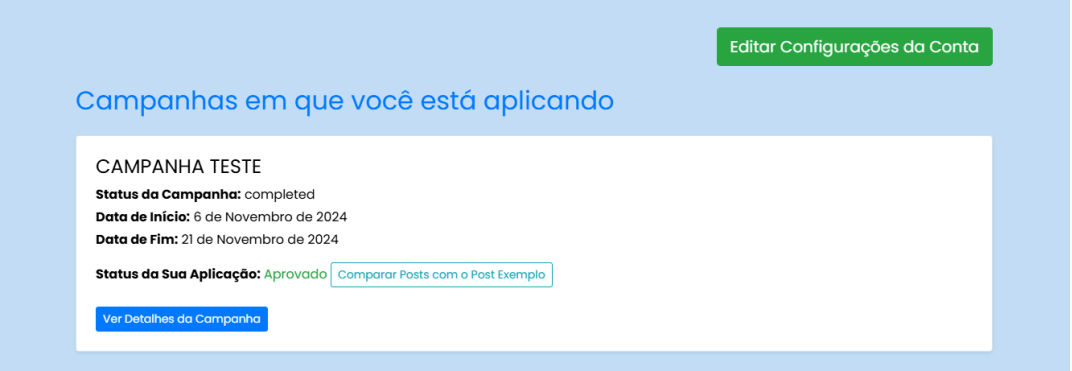
```
return render(
    request,
    'account/influencer_account.html',
    {
        'campaigns_with_status': campaigns_with_status,
        'social_profiles': social_profiles
    }
)
```

Fonte: Elaborado pelo autor (2024)

14.2.8 Resposta HTTP com o HTML Renderizado

- O servidor Django envia uma **resposta HTTP** ao navegador, contendo o HTML renderizado da página de perfil do influenciador.
- Essa resposta é recebida pelo navegador, que exibe o conteúdo para o usuário.

Figura 9 - Tela do perfil do usuário com os dados

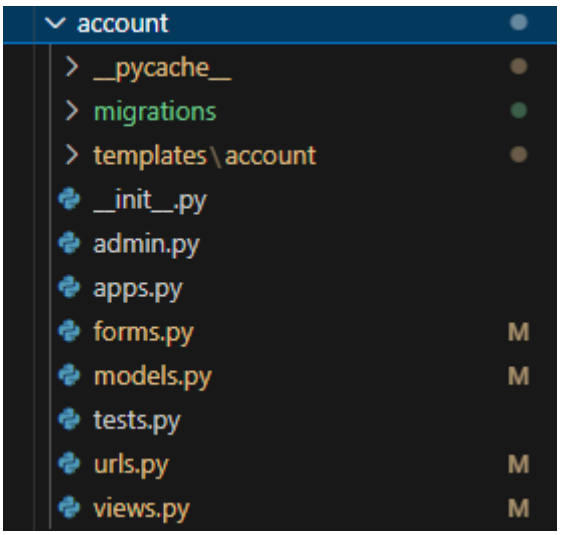


Fonte: Elaborado pelo autor (2024)

15.3 Estrutura dos Arquivos “Back-end”

O Django organiza o projeto em **apps**, módulos independentes que agrupam funcionalidades específicas, facilitando a manutenção e reutilização de código. Cada app possui uma estrutura padrão com arquivos como `models.py`, `views.py`, e `urls.py` para gerenciar banco de dados, lógica e rotas.

Figura 10 - Estrutura de um app



Fonte: Elaborado pelo autor (2024)

15.3.1 Account App

O Account lida com todas as operações relacionadas às contas de usuário, incluindo perfis, edição de dados pessoais e configurações específicas para

influenciadores e marcas. Ele permite a personalização do perfil, atualizações de informações e o gerenciamento de interações entre usuários e o sistema.

Paths:

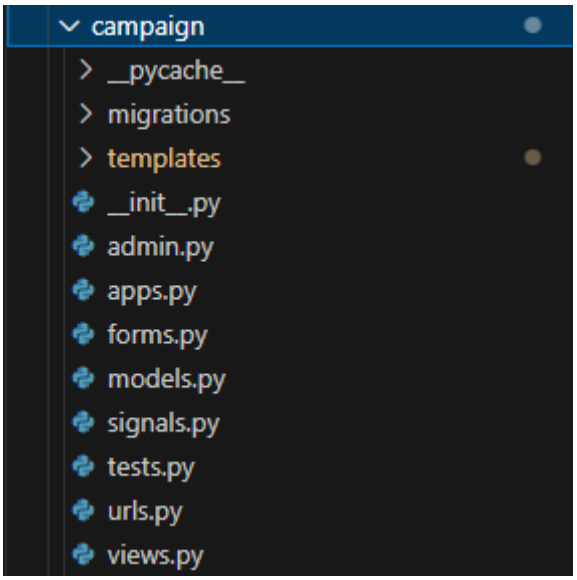
- path('logout/'), - Executa o logout do usuário
- path('login/'), - Executa o login do usuário
- path('register/'), - Faz o cadastro do usuário
- path('delete_account/') - Deleta a conta do usuario
- path('edit_influencer_profile/'), - Edita o perfil do usuário
- path('influencer_account/'), - Recupera os dados do usuário
- path('influencer/campaign/<int:campaign_id>/compare/'), - Compara os posts do influencer com o post 'template'
- path('rate-influencer/') - Avalia o influencer

Pode ser observado na Figura 10.

14.3.2 Campaign App

O Campaign gerencia o ciclo de vida das campanhas, desde sua criação até sua finalização. Ele controla a exibição e aplicação para campanhas, o status das campanhas e interações entre influenciadores e marcas, facilitando o lançamento, a edição e o acompanhamento de campanhas.

Figura 11 - Estrutura do App campaign



Fonte: Elaborado pelo autor (2024)

Paths:

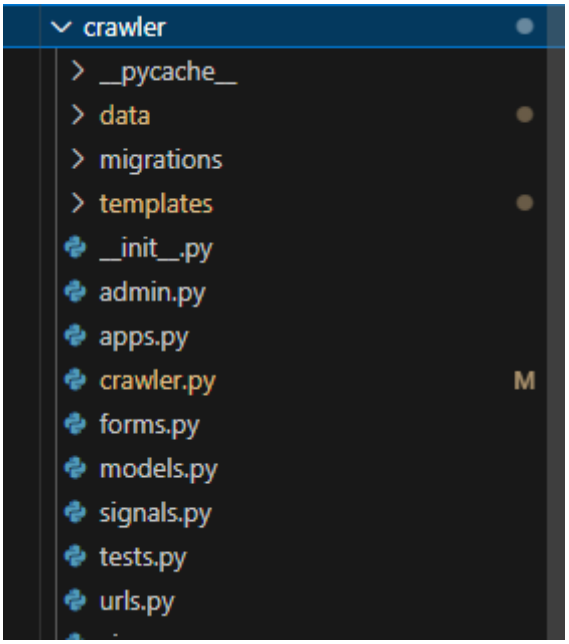
- path('<int:campaign_id>/') – Exibe os detalhes da campanha específica.

- path('<int:campaign_id>/apply/') – Permite que um usuário aplique para a campanha específica.
- path('application/<int:application_id>/update/<str:status>/') – Atualiza o status da aplicação de um influenciador em uma campanha para aprovado, rejeitado, etc.
- path('<int:campaign_id>/finalize/') – Finaliza a campanha específica, alterando seu status para concluído.
- path('new/) – Cria uma nova campanha no sistema.
- path('disapprove_post/<int:post_id>/<int:campaign_id>/') – Influencer desaprova um post em uma campanha específica.
- path('create_post_exemplo/<int:campaign_id>/') – Cria um post 'template' para comparação na campanha específica.
- path('edit_post_exemplo/<int:post_exemplo_id>/') – Edita um post 'template' já existente em uma campanha.
- path('comparisons/<int:campaign_id>/') – Realiza uma comparação entre posts do influenciador e o post 'template' para a campanha específica.
- path('<int:campaign_id>/manage_posts/') - Gerencia os Posts Exemplos
- path('reject_influencer_post_disagree/<int:post_id>/') - Rejeita a desacordo do influencer sobre a comparação automática.

15.3.3 Crawler App

O Crawler é dedicado à coleta e atualização de dados externos relacionados às campanhas. Ele configura, ativa e gerencia o processo de scraping, obtendo informações relevantes de redes sociais ou outras plataformas para comparação e validação de posts com os exemplos definidos pelas campanhas.

Figura 12 - Estrutura do app crawler



Fonte: Elaborado pelo autor (2024)

Esse app possui um arquivo adicional chamado signals.py, cuja função é permitir a execução de ações específicas em resposta a eventos automáticos do sistema, como a criação, atualização ou exclusão de registros no banco de dados. Por exemplo, em um contexto de gerenciamento de usuários, o signals.py pode ser configurado para enviar um e-mail de boas-vindas quando um novo usuário se cadastra, ou no caso desse projeto enviar notificações para Influenciadores e Marcas.

Paths:

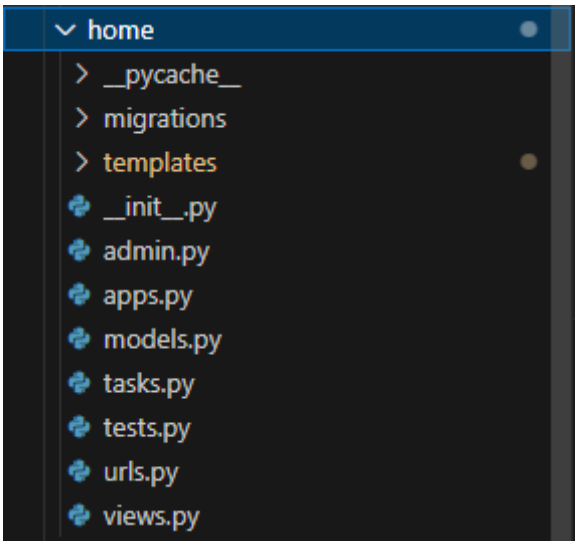
- path('') – Exibe a interface de controle principal para gerenciar o status dos crawlers nas campanhas.
- path('<int:campaign_id>/') – Exibe detalhes do crawler para uma campanha específica.
- path('toggle/<int:campaign_id>/') – Ativa ou desativa o crawler para a campanha específica.
- path('crawler/log/<int:campaign_id>/') – Recupera e exibe o log do crawler para a campanha específica.

15.3.4 Home App

O Home gerencia a interface principal e páginas públicas do sistema, incluindo a página inicial e conteúdos exibidos para usuários não autenticados. Ele atua como ponto de entrada para o sistema, fornecendo informações e orientações

para novos usuários e apresentando o propósito do sistema.Além de ser o organizador da central de notificações.

Figura 13 - Estrutura do app home



Fonte: Elaborado pelo autor (2024)

Paths:

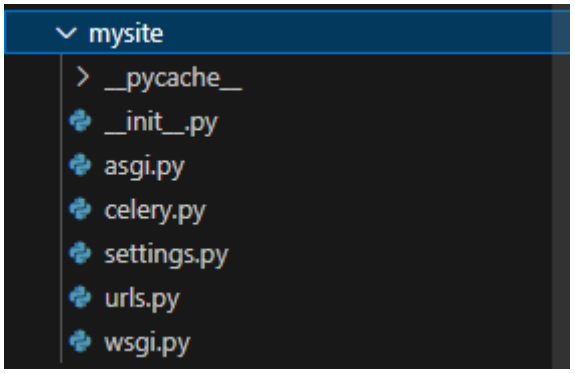
- path('notifications/get/') – Recupera as notificações não lidas para o usuário autenticado.
- path('notifications/read/') – Marca todas as notificações como lidas para o usuário.

Esse app possui um arquivo adicional chamado tasks.py, cujo é usado para definir tarefas automáticas ou assíncronas, como enviar notificações periódicas ou realizar operações em segundo plano, permitindo que ações sejam executadas sem bloquear o fluxo principal do sistema. No caso desse projeto ele é utilizado para atualizar o model de post caso o crawler ache um novo post na rede social.

15.3.5 Mysite

O Mysite atua como a configuração principal do projeto Django, centralizando as configurações globais, incluindo as configurações de INSTALLED_APPS, URLs, banco de dados e autenticação. Ele coordena a inicialização do projeto e a integração de todos os apps dentro de uma estrutura coesa.

Figura 14 - Estrutura do mysite



Fonte: Elaborado pelo autor (2024)

O Mysite possui um arquivo adicional chamado celery.py, que configura o Celery, uma ferramenta para gerenciar tarefas assíncronas e agendadas. Ele define o ambiente e integra o Celery ao projeto Django, permitindo o agendamento e a execução de tarefas de forma eficiente e em segundo plano.

Paths:

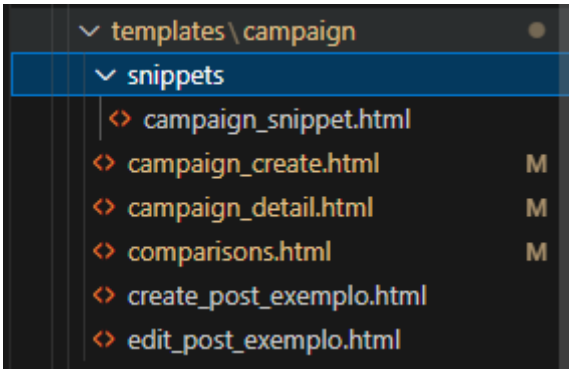
- path('admin/', admin.site.urls) – Fornece acesso ao painel de administração do Django para gerenciar dados e configurações do sistema.
- path('', home_screen_view, name="home") – Exibe a página inicial do sistema.
- path('account/', include(('account.urls', 'account'), namespace='account')) – Inclui e organiza todas as rotas relacionadas ao gerenciamento de contas de usuário.
- path('campaign/', include(('campaign.urls', 'campaign'), namespace='campaign')) – Inclui e organiza todas as rotas relacionadas à criação e gerenciamento de campanhas.
- path('crawler/', include(('crawler.urls', 'crawler'), namespace='crawler')) – Inclui e organiza todas as rotas relacionadas ao controle e configuração dos crawlers.
- path('home/', include(('home.urls', 'home'), namespace='home')) – Inclui e organiza todas as rotas relacionadas às funcionalidades da página inicial, header e footer.

15.4 Estrutura dos Arquivos “Front-end”

Cada app possui uma pasta templates que armazena os arquivos HTML associados às suas views. Dentro dessa pasta, é comum ter uma subpasta chamada snippets, onde ficam os componentes HTML reutilizáveis, como o layout

de uma campanha, que podem ser incluídos em várias páginas (Figura 15), mantendo o código organizado e evitando duplicações.

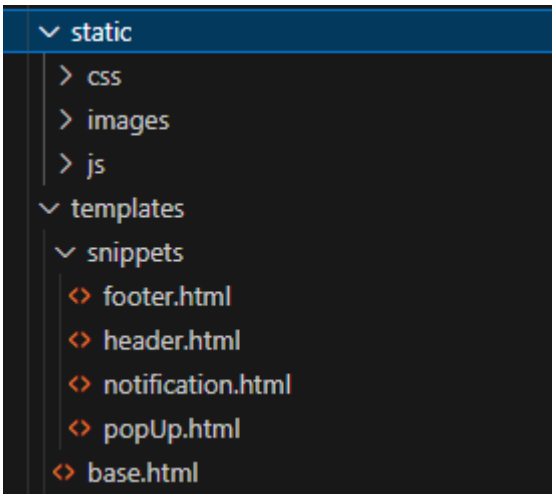
Figura 15 - Estrutura de uma pasta templates



Fonte: Elaborado pelo autor (2024)

Os arquivos de CSS, imagens e JavaScript ficam em uma pasta separada no mesmo nível dos apps chamada static, organizando todos os recursos estáticos do projeto. Também há uma pasta de templates nesse nível que armazena os HTMLs base e componentes comuns, como header e footer, que servem de estrutura para várias páginas do site.

Figura 16 - Estrutura dos arquivos de front-end



Fonte: Elaborado pelo autor (2024)

16. Desafios encontrados

Na elaboração deste trabalho, foram encontrados diversos desafios, tanto no desenvolvimento do projeto quanto na pesquisa e análise dos dados, abrangendo diversos aspectos, que exigiram adaptações e decisões estratégicas para garantir o

cumprimento dos objetivos propostos. Em seguida, serão detalhados os principais obstáculos enfrentados.

16.1 Post ‘Template’

Foi decidido com o orientador que o Post ‘Template’ elaborado pela marca seria algo simplificado, visando facilitar o desenvolvimento e a usabilidade. No modelo atual, ao criar o Post ‘Template’, a marca precisa fornecer apenas o perfil de mídia social do influencer, uma data e uma descrição. Embora existam outras abordagens mais detalhadas para estruturar o Post ‘Template’, como incluir uma imagem do logotipo da marca ou exigir campos adicionais, a opção escolhida priorizou a simplicidade dentro que era objetivo ser testado dentro do escopo deste projeto.

No processo de verificação, o crawler avalia a compatibilidade de cada post com o ‘Template’ fornecido, verificando se o nome de usuário (do perfil do social media), a descrição e a data correspondem aos critérios do Post ‘Template’.

Figura 17 - Tela de criação para a campanha teste

Fonte: Elaborado pelo autor (2024)

16.2 Crawler

No início do projeto, a ideia era desenvolver um crawler que pudesse acessar e coletar dados de diversas redes sociais. No entanto, ao longo das pesquisas, encontrei vários desafios significativos. Muitas das redes sociais populares

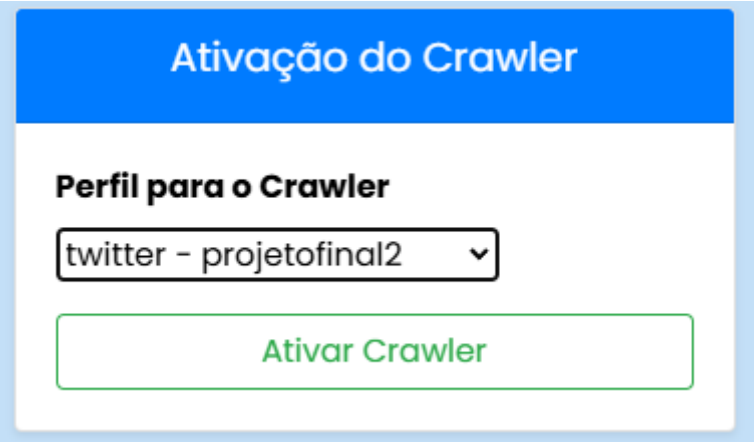
implementam métodos avançados de segurança para impedir atividades de web scraping e o uso de crawlers não autorizados. Para contornar essas barreiras, a ideia inicial era utilizar as APIs oficiais fornecidas por essas plataformas.[12]

Essas APIs de desenvolvedor, no entanto, costumam ser bastante caras. Por exemplo, a API do Twitter (atualmente chamada de rede social X) tem um custo mensal de 175 (cento e setenta e cinco) dólares para o plano básico, que permite apenas 3.000 (três mil) posts escritos e 10.000 (dez mil) posts lidos por mês. Embora o uso da API tornasse a aplicação mais rápida e eficiente para obter dados, o alto custo inviabilizou essa abordagem para o projeto. [13]

Diante dessas restrições, a solução foi desenvolver um crawler específico para a rede social X, utilizando a biblioteca Playwright em Python. Essa ferramenta permite automatizar interações com a interface da web, simulando a navegação humana. O crawler foi configurado para acessar uma conta criada especificamente para este projeto, chamada “projetofinal2”. Ele navega até a página do influenciador desejado e captura os últimos quatro posts do usuário em questão.

Essa abordagem, embora limitada em comparação com uma integração via API, possibilitou a coleta dos dados necessários para o projeto de forma funcional e acessível, evitando os custos elevados e as limitações impostas pelas APIs oficiais.

Figura 18 - Tela de ativação do crawler

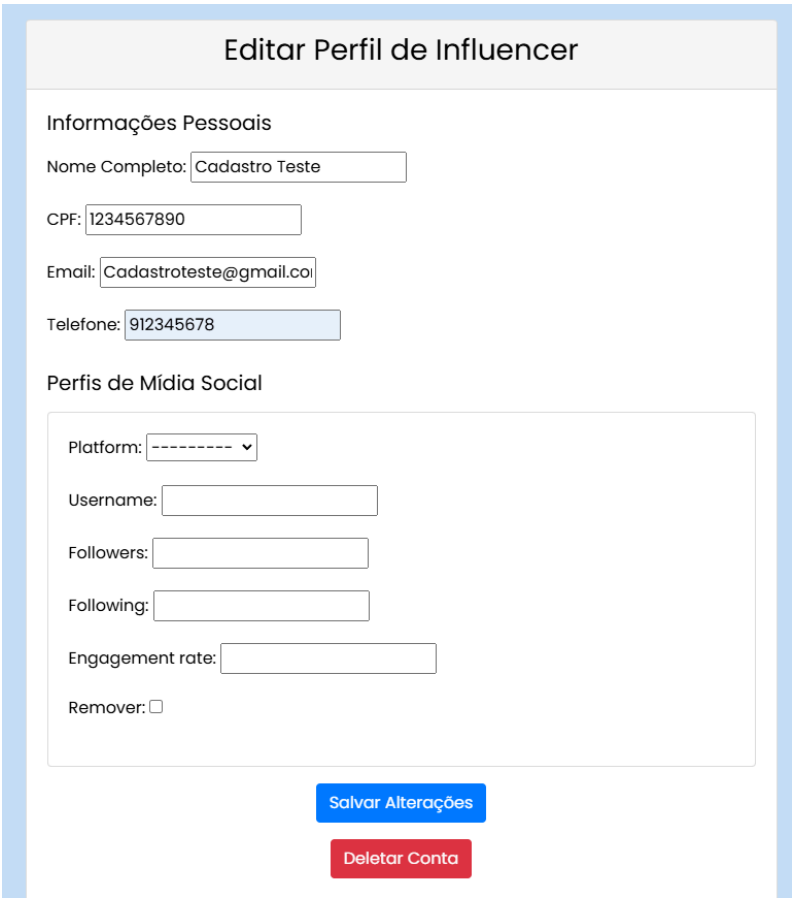


Fonte: Elaborado pelo autor (2024)

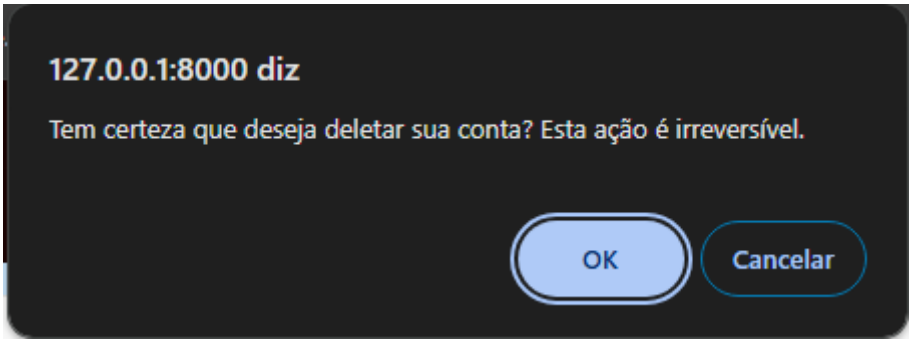
17. Testes Realizados

17.1 Deleção de Conta do Influencer

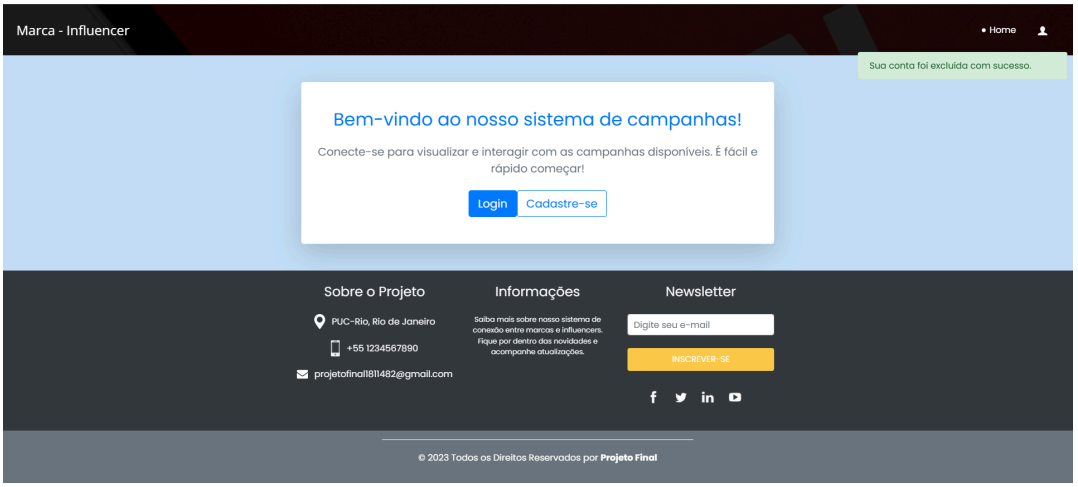
Abrindo a tela de Editar Perfil de Influencer temos um botão vermelho escrito “Deletar Conta”.



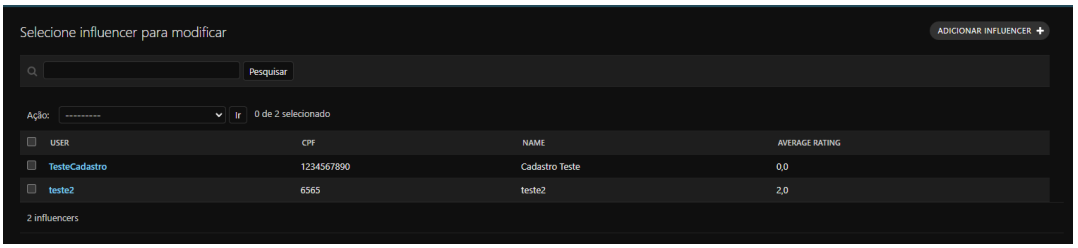
Clicando nele uma mensagem vai aparecer pedindo confirmação:



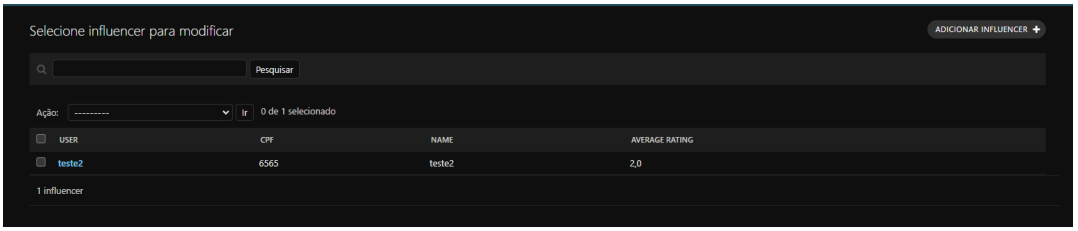
Após confirmar:



Visão do Admin antes da confirmação:

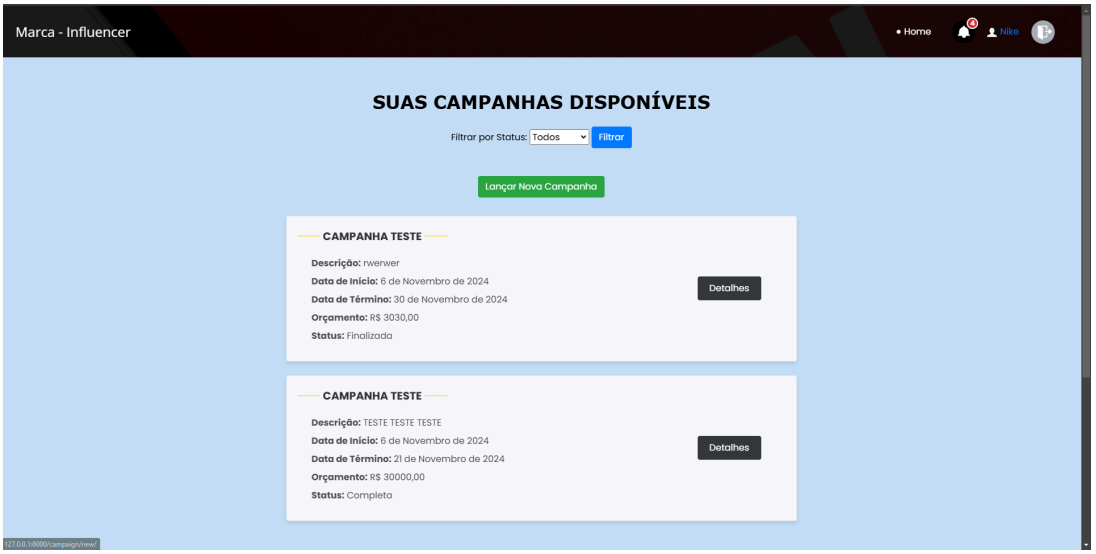


Visão do Admin após a confirmação:

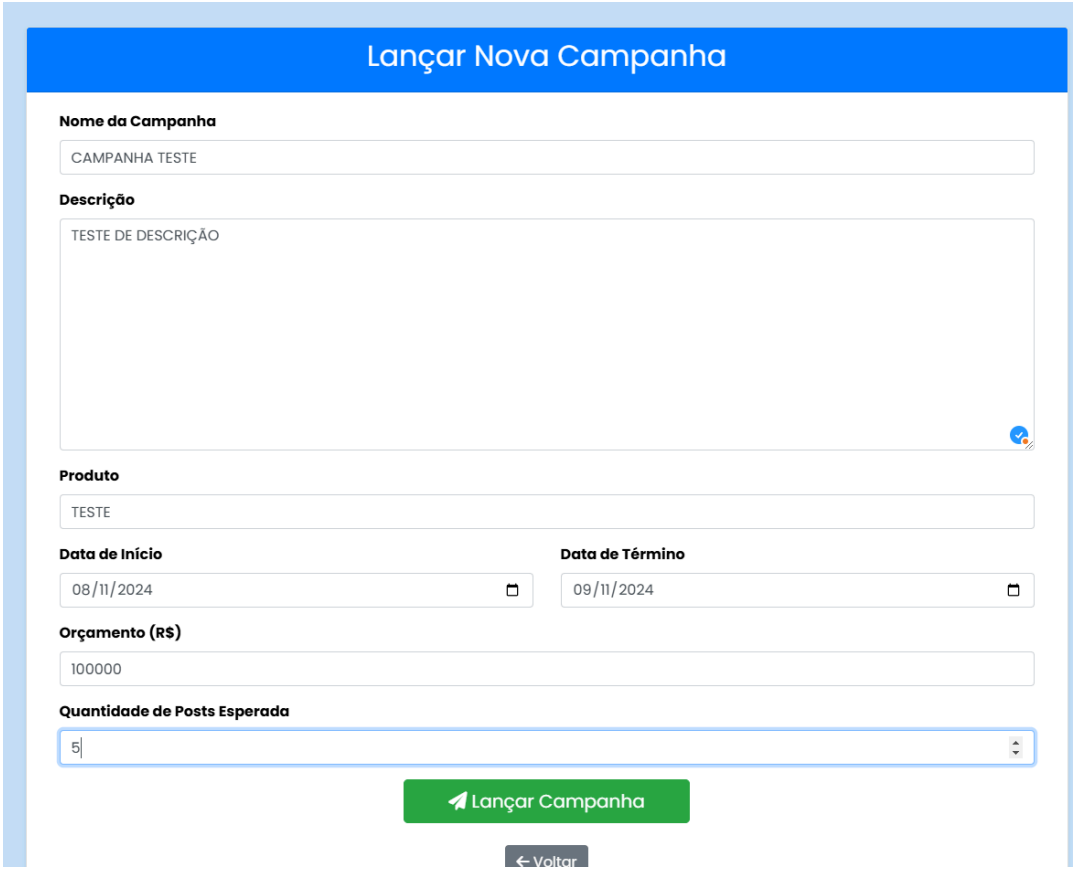


17.2 Lançar Campanha

Clicando em “Lançar Campanha”



Preencher o formulário e clicar em enviar:



Aparecerá uma mensagem de confirmação no canto superior direito e a campanha criada poderá ser vista na home:

17.3 Verificação de Post (Crawler)

Na tela de configuração do crawler para campanha, selecionando o perfil do influencer:

Quando o crawler começar a rodar o log irá aparecer:

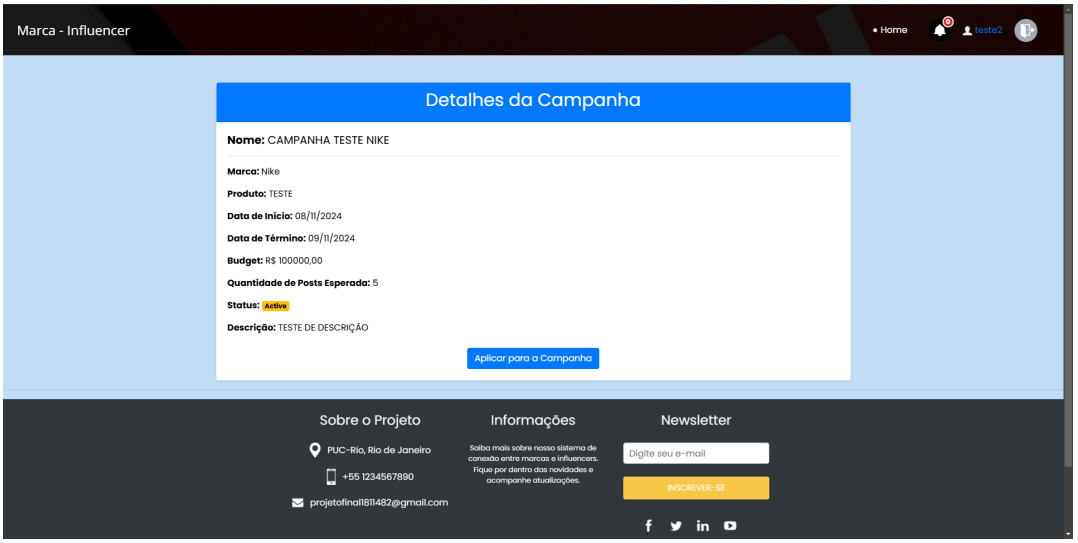
Quando o crawler resgatar os posts ele serão exibidos na tela:

Após os posts serem resgatados poderá ser analisado a comparação automática:

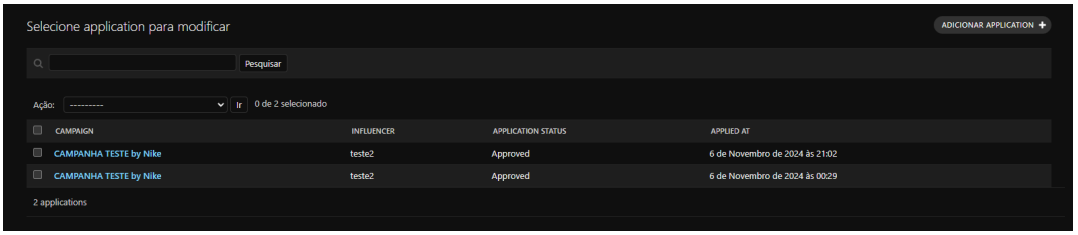
17.4 Aplicar para a Campanha

Na tela home o influencer pode ver todas as campanhas ativas:

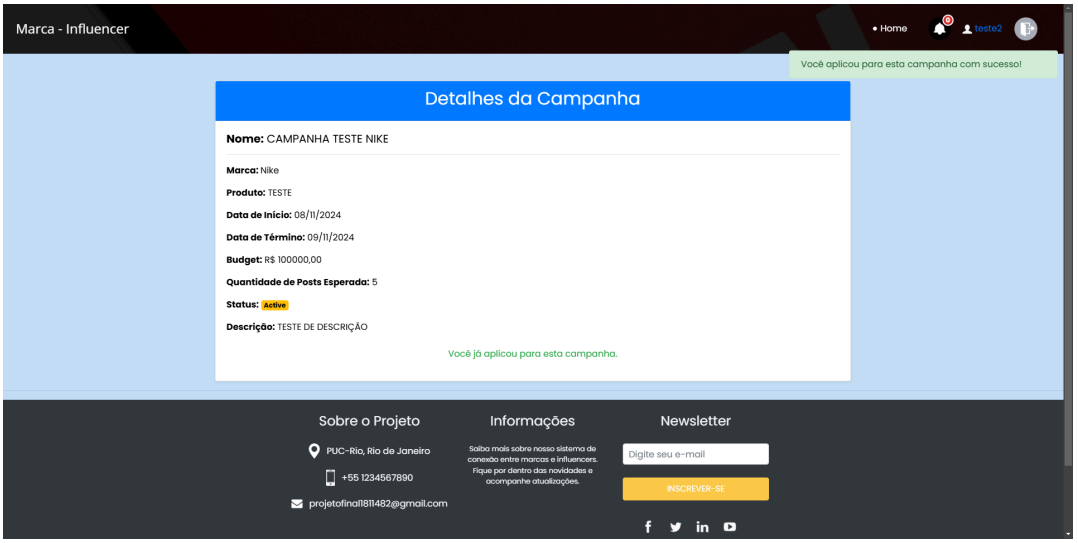
Clicando em detalhes, podemos aplicar para a campanha:

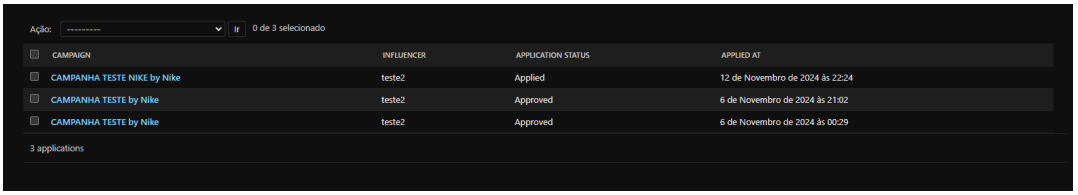


Antes de clicar no botão de aplicar o banco de dados:



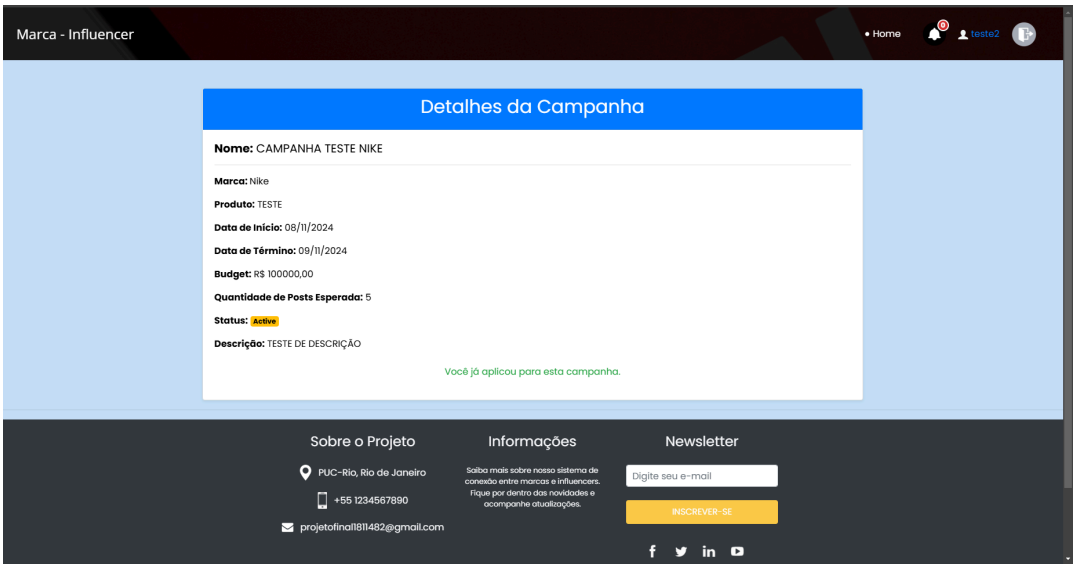
Após clicar no botão de aplicar:



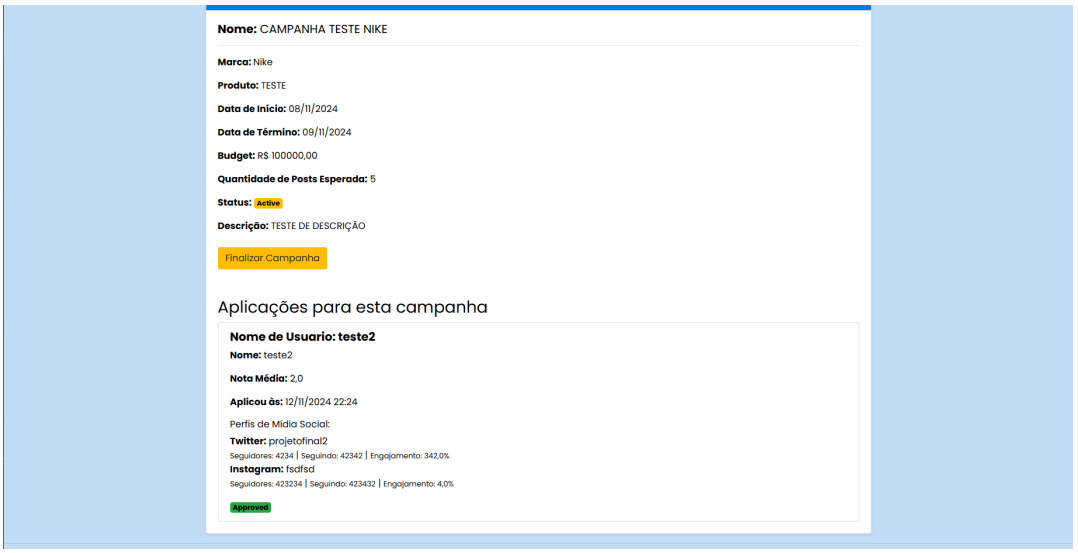


17.5 Notificações

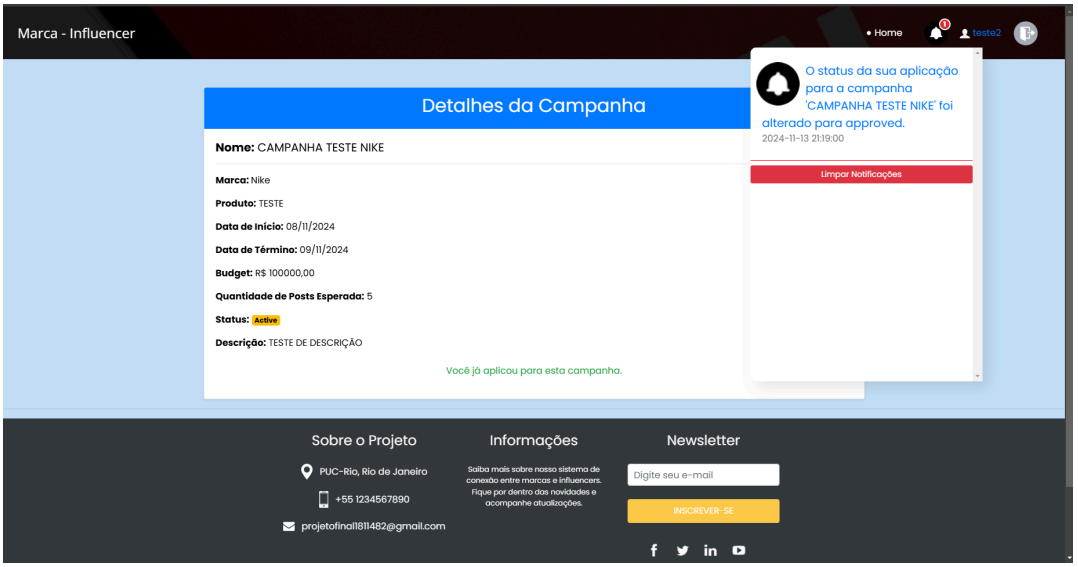
Ao realizar algumas ações ou dado algum acontecimento é enviado uma notificação ao usuário. Uma delas, por exemplo, é quando o status da campanha do influencer é alterada, o influencer recebe uma notificação avisando isso.
Influencer aplicou para campanha:



Marca aprova aplicação:



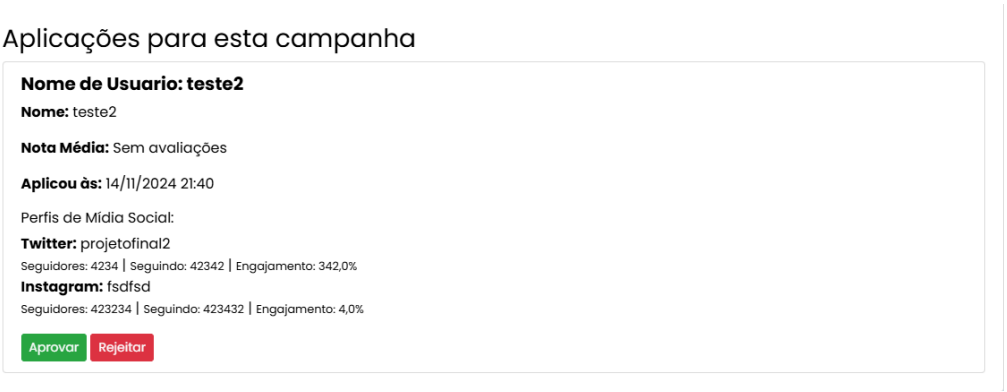
Influencer receba notificação:



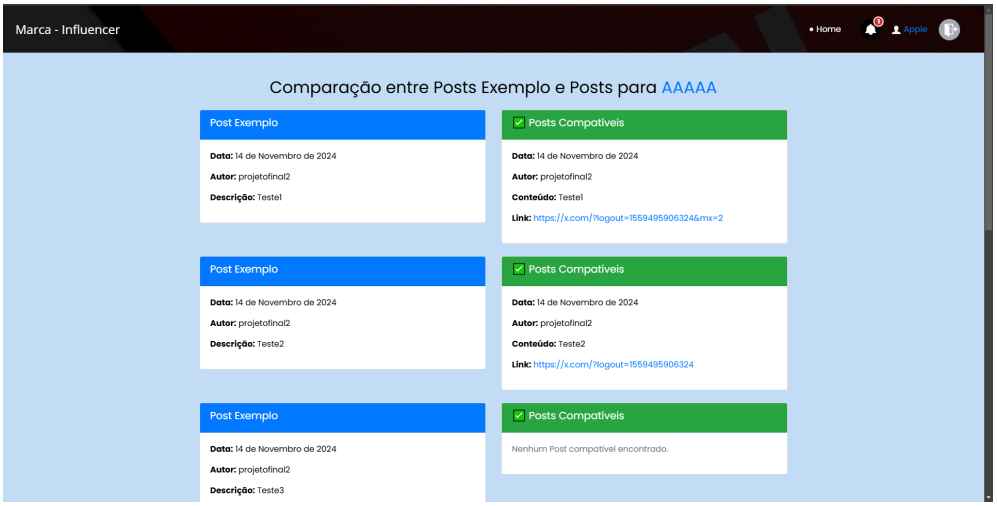
17.6 Avaliação do Influencer

A avaliação é realizada ao final da campanha, com a nota sendo baseada no número de posts entregues pelo influenciador em relação ao que foi acordado com a marca.

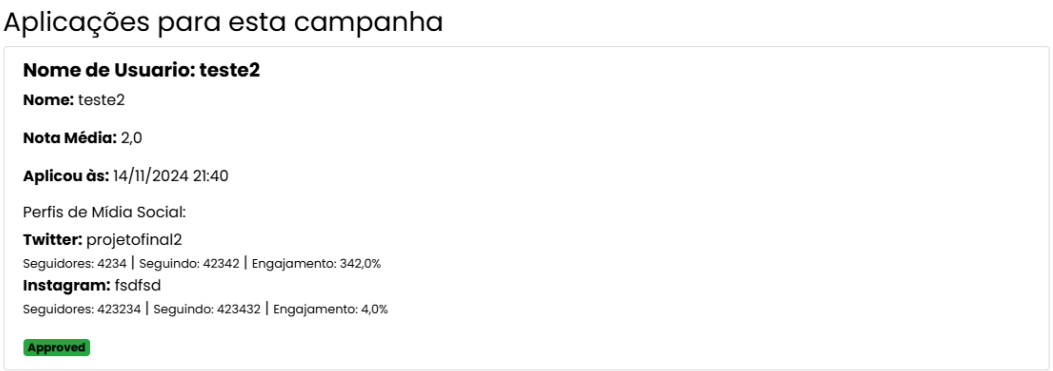
Influencer sem avaliação.



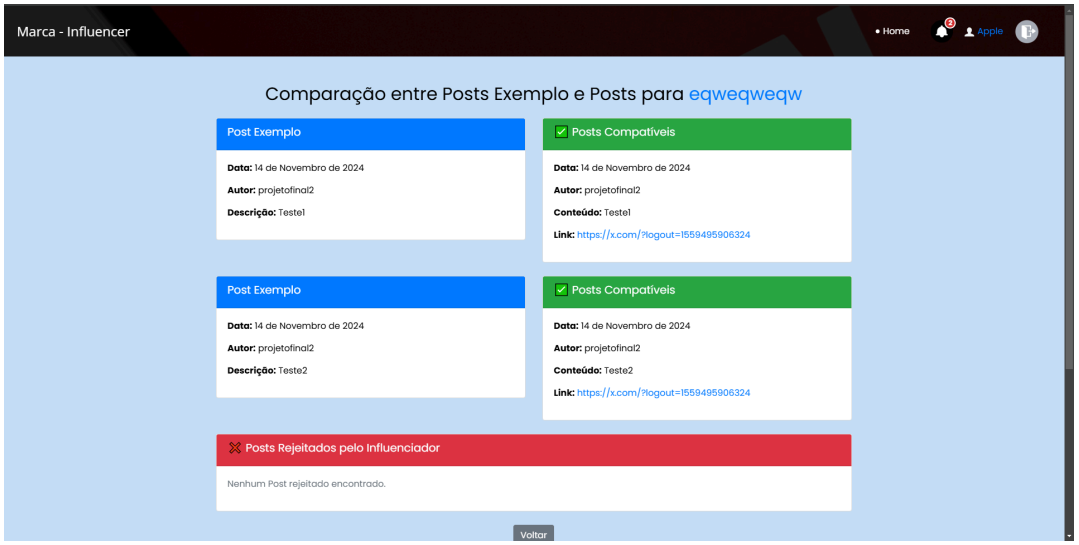
Influencer fez apenas dois posts de 5



Após a marca finalizar a campanha o influencer recebe a nota dele:



A nota dele foi alterada para dois. Caso futuramente o influencer entregue todos os posts de outra campanha ele receberá uma nota 5 e isso somará na média da sua nota total:



Aplicações para esta campanha

Nome de Usuario: teste2

Nome: teste2

Nota Média: 3,5

Aplicou às: 14/11/2024 23:22

Perfis de Mídia Social:

Twitter: projetofinal2

Seguidores: 4234 | Seguindo: 42342 | Engajamento: 342,0%

Instagram: fsdfsd

Seguidores: 423234 | Seguindo: 423432 | Engajamento: 4,0%

Approved

18. Considerações Finais

O objetivo principal foi criar um sistema que conecte influenciadores e marcas de maneira simples e rápida, além de que esse projeto trouxe uma solução eficiente para a automação da coleta de dados de influenciadores nas redes sociais, com foco na plataforma X (antigo Twitter). A estrutura do sistema foi projetada para ser flexível e escalável, com possibilidade de futuras integrações com outras redes sociais, desde que sejam superados os desafios técnicos e financeiros impostos pelas APIs oficiais.

Estou satisfeito com o resultado final, pois consegui implementar todas as funcionalidades planejadas, ainda que de forma diferente do conceito inicial. Acredito que me adaptei bem às dificuldades que surgiram ao longo do desenvolvimento, o que trouxe um grande aprendizado.

O desenvolvimento deste projeto proporcionou um aprendizado significativo em áreas como web scraping, criação de crawlers e segurança web. Além disso, foi uma oportunidade valiosa para reforçar e aplicar conhecimentos adquiridos em Django, HTML, CSS, JavaScript e Python, consolidando a experiência prática obtida ao longo do curso de Ciência da Computação.

19. Trabalhos Futuros

Futuramente, seria interessante aprimorar o crawler com APIs oficiais para ampliar a coleta de dados e integrar outras redes sociais, como Instagram e TikTok, aumentando o alcance e a análise do engajamento dos influenciadores. Implementar algoritmos de análise de ‘sentimento’ e NLP traria insights mais profundos sobre o conteúdo dos posts.

Além disso, desenvolver um painel com visualizações de dados e relatórios detalhados facilitaria o acompanhamento das campanhas em tempo real. Melhorias na segurança e escalabilidade da aplicação também são essenciais para suportar um ambiente de produção com alto volume de dados.

20. Referências Bibliográficas

[1] RÁDIO FM. [S. l.], 27 mar. 2024. Disponível em: https://pt.wikipedia.org/wiki/R%C3%A1dio_FM. Acesso em: 24 maio 2024.

[2] CREATOR economy: qual é o futuro do marketing de influência?. [S. l.], 3 out. 2023. Disponível em: https://portal.fgv.br/artigos/creator-economy-qual-e-futuro-marketing-influencia. Acesso em: 24 maio 2024.

[3] OS INFLUENCIADORES Digitais no Processo de Tomada de Decisão de Seus Seguidores. Os Influenciadores Digitais no Processo de Tomada de Decisão de Seus Seguidores, [s. l.], 28 fev. 2019. Disponível em: https://www.rasi.vr.uff.br/index.php/rasi/article/view/322/89. Acesso em: 24 maio 2024.

[4] THE LYFE Influence. [S. l.], 9 fev. 2022. Disponível em: https://thelyfe.com.br/. Acesso em: 24 maio 2024.

[5] JavaScript Documentation. **JS**. Disponível em: https://devdocs.io/javascript/. Acesso em: 16 out. 2024.

[6] CSS Documentation. **CSS**. Disponível em: https://devdocs.io/css/. Acesso em: 16 out. 2024.

[7] Bootstrap Documentation. **Bootstrap**. Disponível em: https://getbootstrap.com/docs/4.1/getting-started/introduction/. Acesso em: 16 out. 2024.

[8] Django Documentation. **Django**. Disponível em: https://docs.djangoproject.com/en/5.1/. Acesso em: 09 out. 2024.

[9] Python Documentation. **Python**. Disponível em: https://docs.python.org/3/. Acesso em: 09 out. 2024.

[10] Playwright Documentation. **Playwright**. Disponível em: https://playwright.dev/docs/intro. Acesso em: 11 out. 2024.

[11] Arquitetura MTV. **MTV,** 3 jan. 2020. Disponível em: https://diandrasilva.medium.com/como-funciona-a-arquitetura-mtv-django-86af916f1f63. Acesso em: 13 out. 2024.

[12] WebScraping Dificuldades. **proxyscrape,** 6 mar. 2024. Disponível em: https://pt-br.proxyscrape.com/blog/desafios-da-raspagem-da-web. Acesso em: 14 out. 2024.

[13] X Developer. ***X***. Disponível em: https://developer.x.com/en/portal/petition/essential/basic-info?plan=basic. Acesso em: 14 out. 2024.

[14] RIBEIRO, Rafael Auad Medeiros. **Sistema Web de Vendas de Produtos para Organizações**. Rio de Janeiro, 2023. 89 p. Relatório de Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.