HUBNEWS

Documento de Requisitos

Durval Ferreira Sobrinho Junior
John Lennon de Souza Galdino
Luis Fernando Teixeira Oliveira
Vanúbia Santos Lima
Wallas Oliveira Almeida

Versão 1.0

Histórico de Alterações

Data	Versão	Descrição	Autor
31/10/2023	1.0	Versão inicial do documento	Vanúbia
04//11/2023	1.0	Adição de Requisitos Funcionais e não Funcionais	Durval, Vanúbia, John, Wallas e Luis Fernando

Conteúdo

1. Introdução

Este documento especifica os requisitos do Sistema HubNews, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

- **Seção 2 – Descrição geral do sistema**: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- **Seção 3 – Requisitos funcionais (casos de uso)**: especifica todos os casos de uso do sistema, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado.
- **Seção 4 – Requisitos não-funcionais**: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho e software.
- **Seção 5 – Referências:** apresenta referências para outros documentos utilizados para a confecção deste documento.

1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.2.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do identificador do requisito, de acordo com a especificação a seguir:

[*identificador do requisito*]

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] para os requisitos funcionais e [NF001] para os não-funcionais e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

Esta seção descreve superficialmente o cliente, os futuros usuários e fornece uma visão geral do HubNews.

2.1 Usuários

Existem dois tipos de usuários no sistema, sendo eles o Cliente e o Admin. O cliente poderá utilizar o sistema para gerar consultas de notícias de assuntos que sejam de seu interesse, filtrando palavras-chaves e categorias.

2.2 Visão Geral do Sistema

3. Referencial Teórico

3.1 Node.js

Em dias atuais, aplicações web são amplamente construídas, e sempre visando a melhoria e facilidade de criá-las, é comum que desenvolvedores se empenhem em criar frameworks que visem melhores formas de desenvolvimento das aplicações.

Criado em 2009 por Ryan Dahl, nasce o Node.js, surgindo como uma solução poderosa e barata para a criação e a manutenção de ambientes de tecnologia com altas demandas (Alura, 2023).

O Node.js revolucionou a forma como o JavaScript é usado, permitindo que ele fosse executado no lado do servidor, não apenas no navegador. Sua arquitetura orientada a eventos e baseada em I/O não bloqueante torna o Node.js extremamente eficiente para lidar com um grande número de conexões simultâneas. Essa tecnologia é amplamente utilizada para construir aplicativos web, APIs, servidores e outros sistemas distribuídos, permitindo um desenvolvimento ágil e eficiente, principalmente para aplicações em tempo real e com grande volume de tráfego.

Como um tempo de execução do JavaScript assíncrono orientado a eventos, o Node.js foi projetado para construir aplicativos de rede escalonáveis. No exemplo ”olá mundo”a seguir, muitas conexões podem ser tratadas simultaneamente. A cada conexão, o retorno de chamada é acionado, mas se não houver trabalho a ser feito, o Node.js irá dormir (Node.js, 2023).

Figura 1 – Conexões sendo tratadas simultaneamente no node.js

```
const http = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Fonte: (Node.js, 2023)

A popularidade do Node tem sido considerável desde seu surgimento, alguns dos motivos para esse crescimento são:

- ***JavaScript* Unificado,** que permitiu que os desenvolvedores usassem JavaScript tanto no frontend quanto no backend, unificando a linguagem de programação.

- **Ecossistema NPM:** o npm (Node Package Manager) é um gerenciador de pacotes, um dos maiores repositórios de software do mundo, oferecendo uma grande variedade de módulos e bibliotecas prontos para uso
- **Escalabilidade e Desempenho:** O Node.js é altamente escalável e oferece um bom desempenho, especialmente em aplicações que exigem alta concorrência e manipulação rápida de I/O.

O Node.js continua a ser uma força impulsionadora no mundo do desenvolvimento de software, transformando a maneira como as aplicações são construídas e executadas. Sua arquitetura eficiente e orientada a eventos, aliada à unificação da linguagem JavaScript, permite um desenvolvimento ágil e escalável.

3.2 *Selenium*

Selenium é um conjunto de ferramentas de código aberto multiplataforma, usado para testar aplicações web pelo *browser* de forma automatizada. Ele executa testes de funcionalidades da aplicação *web* e testes de compatibilidade entre *browser* e plataformas diferentes (TREINAWEB, 2023).

Ele fornece extensões para emular a interação do usuário com os navegadores, um servidor de distribuição para dimensionar a alocação de navegadores e a infraestrutura para implementações da especificação *W3C WebDriver* que permite escrever código intercambiável para todos os principais navegadores da *web* (Selenium, 2023).

O Selenium iniciou-se em 2004 na *ThoughtWorks* em Chicago, com Jason Huggins construindo o modo Core como "*JavaScriptTestRunner*"para o teste de uma aplicação interna de *Time and Expenses* (Python, Plone) (Selenium, 2023).

O Selenium suporta várias linguagens de programação, como Java, *Python*, *JavaScript*, *C#*, entre outras, o que permite aos desenvolvedores escolher a linguagem com a qual se sintam mais confortáveis para escrever seus testes automatizados.

Sua versatilidade e capacidade de lidar com diversas situações em aplicações *web*, juntamente com sua natureza de código aberto, o tornaram uma escolha popular para equipes de desenvolvimento que buscam automatizar seus testes de software, garantindo a qualidade e confiabilidade de suas aplicações.

3.3 *Beautiful Soup*

Beautiful Soup facilita significativamente o processo de extração de dados de arquivos HTML e XML em ambientes Python. Ao integrar-se perfeitamente com analisadores HTML/XML, a biblioteca oferece uma abordagem idiomática que simplifica a navegação, pesquisa e modificação da árvore de análise.

Com sua interface intuitiva, Beautiful Soup se destaca ao proporcionar uma economia notável de tempo para os desenvolvedores. Ao eliminar a necessidade de tarefas repetitivas e complexas, a biblioteca permite que os programadores atinjam seus objetivos de forma mais eficiente, acelerando o processo de desenvolvimento. Seja na busca por informações específicas ou na manipulação da estrutura do documento, Beautiful Soup emerge como uma ferramenta valiosa que contribui para uma experiência mais suave e produtiva no manuseio de dados HTML e XML (Beautiful-Soup, 2023)**.**

Aqui estão algumas maneiras simples de navegar nessa estrutura de dados:

```
 soup.title
# <title>The Dormouse's story</title>

soup.title.name
# u'title'

soup.title.string
# u'The Dormouse's story'

soup.title.parent.name
# u'head'

soup.p
# <p class="title"><b>The Dormouse's story</b></p>

soup.p['class']
# u'title'

soup.a
# <a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>

soup.find_all('a')
# [<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,
#  <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
#  <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]

soup.find(id="link3")
# <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>
```

3.3.1 *Tipos de objetos*

Beautiful Soup transforma um documento HTML complexo em uma árvore complexa de objetos Python. Mas você só terá que lidar com cerca de quatro tipos de objetos: `Tag`, `NavigableString`, `BeautifulSoup`, e `Comment`.

3.4 *MongoDB*

MongoDB é um sistema de gerenciamento de banco de dados (DBMS) não relacional, baseado em software livre, que utiliza documentos flexíveis em vez de tabelas e linhas para processar e armazenar várias formas de dados. Como uma solução de banco de dados NoSQL, o MongoDB não requerer um sistema de gerenciamento de banco de dados relacional (RDBMS), portanto, ele oferece um modelo de armazenamento de dados elástico, que permite aos usuários armazenar e consultar tipos de dados variados com facilidade. Isso não apenas simplifica o gerenciamento do banco de dados para os desenvolvedores, como também cria um ambiente altamente escalável para aplicativos e serviços multiplataforma(IBM, 2023).

Os documentos ou coleções de documentos do MongoDB são as unidades básicas de dados. Formatados como Binary JSON (Javascript Object Notation), esses documentos podem armazenar vários tipos de dados e ser distribuídos para diversos sistemas. Como o MongoDB emprega um design de esquema dinâmico, os usuários têm uma flexibilidade incomparável ao criar registros de dados, consultar coleções de documentos por meio da agregação do MongoDB e analisar grandes quantidades de informações (MongoDB, 2023).

4. Requisitos funcionais (casos de uso)

4.1.1 [RF001] Registrar

Descrição do caso de uso: O usuário deve poder realizar o cadastro no sistema informando o nome, email e senha, endereço, cpf e foto de perfil.

Ator: Cliente

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário entra com os dados solicitados no formulário de cadastro.

Saídas e pós-condição:

O sistema retorna uma mensagem informando o resultado da operação.

Fluxo de eventos principal

1. Usuário escolhe a ação Realizar Cadastro.
2. O sistema oferece o formulário de cadastro para o cliente.
3. O usuário entra com os dados e submete o formulário.
4. O sistema insere os dados submetidos no banco de dados.
5. O sistema retorna para o usuário uma mensagem informando que a operação foi realizada com sucesso.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.2 [RF 002] Realizar Login

Descrição do caso de uso: O usuário deve poder realizar o login no sistema fornecendo seu email e senha que foi fornecido na hora de realizar o cadastro.

Ator: Cliente

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário cliente já deve ter concluído o processo de cadastro no sistema, incluindo o fornecimento de informações de contato, como endereço de e-mail e uma senha de acesso.

Saídas e pós-condição:

O acesso ao sistema é concedido, permitindo que o cliente utilize os recursos e serviços disponíveis.

O sistema pode exibir informações adicionais, como o nome do cliente ou detalhes da conta, após o login bem-sucedido.

Fluxo de eventos principal

1. Usuário escolhe a ação Realizar Login.
2. O sistema apresenta um formulário de login que solicita ao usuário que forneça seu endereço de e-mail e senha registrados durante o cadastro.
3. O usuário insere as informações de login (endereço de e-mail e senha) nos campos apropriados.
4. O sistema verifica as credenciais inseridas pelo usuário.
5. Se as credenciais estiverem corretas e correspondentes às informações registradas, o sistema concede o acesso ao usuário e permite que ele utilize os recursos e serviços disponíveis.

Fluxos secundários

1. Se ocorrer uma falha de comunicação com o banco de dados durante a verificação das credenciais de login, o sistema deve tratar essa situação.
2. Uma mensagem de erro é gerada e exibida ao usuário, informando-o sobre a falha na comunicação com o banco de dados.
3. A operação de login é cancelada devido à falha.
4. O sistema pode oferecer a opção de tentar novamente ou entrar em contato com o suporte técnico para resolver o problema de comunicação com o banco de dados.

4.1.3 [RF 003] Realizar Logout

Descrição do caso de uso: O usuário pode efetuar o logout no sistema, encerrando sua sessão ativa.

Ator: Cliente

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:
Antes de efetuar o logout, o Cliente deve estar autenticado no sistema, ou seja, já ter realizado o login com sucesso.

Saídas e pós-condição:
Após o logout, o Cliente não terá mais acesso ao sistema. Qualquer sessão anterior será encerrada, e ele deverá realizar um novo login para acessar o sistema novamente.

Fluxo de eventos principal

1. Usuário escolhe a ação Realizar Logout.
2. O sistema confirma a ação do Cliente e encerra a sessão ativa.
3. O Cliente é redirecionado para fora do sistema, e a página de login ou outra página inicial do sistema é exibida

Fluxos secundários

1. Se ocorrer uma falha de comunicação com o banco de dados, o sistema deve tratar essa situação.
2. Uma mensagem de erro é gerada e exibida ao usuário, informando-o sobre a falha na comunicação com o banco de dados.
3. A operação de logout é cancelada devido à falha.
4. O sistema pode oferecer a opção de tentar novamente ou entrar em contato com o suporte técnico para resolver o problema de comunicação com o banco de dados.

4.1.4 [RF 004] Editar Perfil

Descrição do caso de uso: O usuário, no caso o Cliente, tem a capacidade de visualizar sua conta e alterar seus dados de cadastro no sistema podendo adicionar uma foto .

Ator: Cliente

Prioridade: □ Essencial □ Importante ■ Desejável

Entradas e pré-condições:

Antes de visualizar seu perfil, o Cliente deve estar autenticado no sistema, ou seja, já ter realizado o login com sucesso.

Saídas e pós-condição:

Após a visualização do seu perfil, o Cliente permanecerá logado no sistema e poderá tomar ações, como editar seu perfil.

Fluxo de eventos principal

1. Usuário escolhe a ação Editar perfil.
2. O sistema recupera as informações vinculadas à conta do Cliente.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.5 [RF 005] Alterar Senha

Descrição do caso de uso: o usuário Cliente, pode alterar a senha de acesso ao sistema caso deseje e tenha a senha atual.

Ator: Cliente

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O cliente deve estar logado no sistema.

O cliente deve estar na pagina de perfil no hubNews para acesso a mudança de senha.

O usuário precisa saber a senha atual para mudança da mesma.

Saídas e pós-condição:

A senha será alterada instantaneamente e a mensagem de confirmação será exibida ao usuário, e após essa execução, a senha de acesso ao sistema passa a ser a nova senha inserida.

Fluxo de eventos principal

1. Usuário seleciona a página de perfil
2. O sistema apresenta todas suas informações em tela
3. O Cliente seleciona a opção de mudança de senha
4. O sistema oferece o formulário para a inserção da senha atual e senha nova para o usuário inserir as informações.
5. Se as credenciais estiverem corretas e a verificação for bem-sucedida, o sistema atualiza a nova senha no banco de dados.
6. O sistema pode exibir uma confirmação de sucesso ou erro caso as credenciais falhe.

Fluxos secundários

1. Se ocorrer uma falha de comunicação com o banco de dados durante a verificação das credenciais da conta, o sistema invalidará a atualização.

2. Uma mensagem de erro é gerada e exibida ao Cliente, informando-o sobre a falha na comunicação com o banco de dados.
3. A operação de atualização da senha é cancelada devido à falha.
4. O sistema pode oferecer a opção de tentar novamente ou entrar em contato com o suporte técnico para resolver o problema de comunicação com o banco de dados.

4.1.6 [RF 006] Visualizar análise de sentimentos por Grupos de palavras

Descrição do caso de uso: O usuário poderá visualizar a quantidade de notícias positivas, negativas e neutras de determinado grupo de palavras.

Ator: Cliente.

Prioridade: □ Essencial ■ Importante □ Desejável

Entradas e pré-condições:

O usuário deverá estar logado com sua conta para poder visualizar a análise de sentimentos é preciso ter uma analytics cadastrada.

Saídas e pós-condição:

O sistema retorna os dados baseados no grupo de palavras, assim mostrando a quantidade notícias de positivas, negativas e neutras.

Fluxo de eventos principal

1. O usuário acessa o painel de analytics.
2. Clica na analytics que deseja visualizar a análise.
3. O usuário clica na opção de análise de sentimentos.
4. O sistema retorna a análise baseada no grupo de palavras.

Fluxos secundários

1. Caso o sistema não consiga fazer a análise será exibido uma mensagem para o usuário informando que não é possível executar essa ação.

4.1.7 [RF 007] Visualizar análise de sentimentos de Notícias

Descrição do caso de uso: O usuário poderá visualizar a quantidade de palavras positivas, negativas e neutras baseada no grupo de notícias de determinada fonte.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário deverá estar logado com sua conta para poder visualizar a análise de sentimentos é preciso ter uma analytics cadastrada.

Saídas e pós-condição:

O sistema retorna os dados baseados nas notícias de cada fonte, assim mostrando a quantidade notícias de positivas, negativas e neutras.

Fluxo de eventos principal

1. O usuário acessa o painel de analytics.
2. Clica na analytics que deseja visualizar a análise.
3. O usuário clica na opção de análise de sentimentos.
4. O sistema retorna a análise baseada nas notícias de cada fonte.

Fluxos secundários

1. Caso o sistema não consiga fazer a análise será exibido uma mensagem para o usuário informando que não é possível executar essa ação.

4.1.8 [RF008] Criar analytics

Descrição do caso de uso: O usuário pode criar analytics de duas formas, tanto por categorias, sendo que ele terá que selecionar uma categoria já criada do sistema, e por palavras chaves, dessa forma ele precisa escolher as palavras que ele deseja usar na analytics.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário deverá estar logado com sua conta para poder criar a sua analytics.

Saídas e pós-condição:

O sistema redireciona o usuário para a página de criação das analytics, sendo por palavras ou categorias.

Fluxo de eventos principal

1. Usuário clica em analytics no menu superior.
2. Em seguida clica na opção criar analytics.
3. O sistema exibe o painel dando as opções de criar por categoria ou palavras chaves
4. O usuário informa qual será sua opção desejada.
5. Clica em criar analytics.
6. O sistema retorna uma mensagem informando que foi criada com sucesso.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.9 [RF 009] Deletar analytics

Descrição do caso de uso: O usuário pode deletar sua analytics caso não queira mais utilizá-la.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário deve estar logado no sistema e ter uma analytics já criada para poder deletar.

Saídas e pós-condição:

O sistema informa se a ação de deletar foi bem sucedida.

Fluxo de eventos principal

1. Usuário acessa a analytics que deseja deletar.
2. Clica no botão de deletar analytics na parte superior do sistema.
3. O sistema informa para o usuário que a analytics foi deletada com sucesso.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.10 [RF 010] Iniciar Expansão por "Minhas Categorias"

Descrição do caso de uso: O usuário pode iniciar uma expansão por "Minhas Categorias".

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário deve estar logado no sistema e não ter nenhuma expansão por "Minhas Categorias" já criada.

Saídas e pós-condição:

O sistema informa se a ação de iniciar expansão foi bem sucedida.

Fluxo de eventos principal

1. Usuário escolhe a ação de iniciar expansão.
2. O sistema apresenta ao usuário as categorias que podem ser selecionadas.
3. O usuário escolhe uma das categorias.
4. O sistema informa para o usuário que a inicialização foi bem sucedida.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.11 [RF 011] Filtrar dados

Descrição do caso de uso: O usuário pode filtrar os dados que vão ser utilizados para criar o gráfico de linhas, assim sendo mais preciso no que ele quer visualizar.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário deve estar logado no sistema e ter uma analytics criada com mais de uma opção para poder filtrar.

Saídas e pós-condição:

O sistema exibe o gráfico em linhas baseadas na filtragem do usuário.

Fluxo de eventos principal

1. O usuário acessa o painel de analytics.
2. Escolhe a analytics que deseja visualizar o gráfico em linhas.
3. O sistema retorna gráfico em linhas, e tendo as opções de filtragem logo acima.
4. O usuário clica na filtragem desejada para o gráfico.
5. O sistema atualiza o gráfico com as informações baseadas na filtragem selecionada.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário informando que não foi possível fazer a filtragem.

4.1.12 [RF012] Visualizar gráfico em linha

Descrição do caso de uso: O usuário poderá visualizar o gráfico de linhas referente á analytics que ele criou, assim tendo um entendimento melhor do dados através dos gráficos.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário precisa estar logado em sua conta, e ter uma analytics criada para poder visualizar o gráfico em linha.

Saídas e pós-condição:

O sistema retorna o gráfico em linha com base nas analytics que o usuário criou.

Fluxo de eventos principal

1. O usuário acessa o painel de analytics.
2. Escolhe a analytics que deseja visualizar o gráfico em linhas.
3. O sistema retorna o gráfico em linhas baseadas nos dados que o cliente escolheu para analytics.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário informando que não foi possível exibir o gráfico.

4.1.13 [RF 013] Visualizar analytics

Descrição do caso de uso: O usuário poderá acessar o painel de analytics, assim podendo ver todas analytics todas que ele criou, e podendo criar novas.

Ator: Cliente.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário deverá estar logado com sua conta para poder visualizar o seu painel de analytics.

Saídas e pós-condição:

O sistema retorna todas as analytics que o usuário criou, e também a opção de poder criar mais.

Fluxo de eventos principal

1. Usuário acessa seu perfil.
2. Acesse na barra de menu a opção de analytics.
3. Clicar na analytics desejada para visualizar.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário informando que não foi possível exibir os dados.

4.1.14 [RF018] Raspar notícias

Descrição do caso de uso: O Crawler deve ser capaz de acessar sites de notícias externas, extrair informações sobre as notícias publicadas e armazenar essas informações para posterior processamento ou exibição.

Ator: Crawler

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

1. Lista de fontes de notícias a serem rastreadas.
2. Critérios de seleção de notícias, como palavras-chave, categoria, intervalo de datas.

Saídas e pós-condição:

1. Conjunto de notícias extraídas das fontes especificadas.

Fluxo de eventos principal

1. O Crawler inicia o processo de raspagem de notícias.
2. O sistema acessa as fontes de notícias especificadas.
3. O Crawler aplica os critérios de seleção para identificar as notícias relevantes.
4. O sistema extrai as informações das notícias selecionadas, como título, autor, data e conteúdo.

5. O Crawler armazena as notícias extraídas em um formato adequado para posterior processamento ou exibição.

Fluxos secundários

1. Se uma fonte de notícias estiver inacessível ou apresentar problemas de conexão, o sistema registra o incidente e continua a busca nas outras fontes especificadas.
2. Se não houver notícias correspondentes aos critérios de seleção fornecidos, o sistema registra a ausência de notícias e encerra o processo de raspagem.

4.1.15 [RF019] Salvar notícias na base de dados

Descrição do caso de uso: O Crawler deve ser capaz de salvar as notícias raspadas na base de dados MongoDB para posterior utilização.

Ator: Crawler

Prioridade: □ Essencial ■ Importante □ Desejável

Entradas e pré-condições:

1. Conjunto de notícias raspadas pelo Crawler.

Saídas e pós-condição:

1. Notícias salvas na base de dados com sucesso.

Fluxo de eventos principal

1. O Crawler recebe o conjunto de notícias raspadas.
2. Para cada notícia, o sistema verifica se ela já existe no MongoDB.
3. Se a notícia já existe no MongoDB:
   a. O sistema verifica se há atualizações nas informações.
   b. Se houver atualizações, o sistema atualiza os dados existentes no MongoDB com as informações mais recentes.
4. Se a notícia não existe no MongoDB, o sistema insere um novo documento representando a notícia no MongoDB.
5. O sistema confirma a conclusão do processo de salvamento das notícias no MongoDB.

Fluxos secundários

1. Se ocorrer algum erro durante o processo de salvamento no MongoDB, o sistema registra o incidente e tenta novamente após um intervalo de tempo determinado, garantindo a integridade dos dados no MongoDB.

4.1.16 [RF020] Visualizar Categorias

Descrição do caso de uso: O Admin poderá visualizar todas as categorias existentes no sistema.

Ator: Admin.

Prioridade: □ Essencial ■ Importante □ Desejável

Entradas e pré-condições:

O Admin deverá estar logado no sistema como admin para ter acesso a funcionalidade.

Saídas e pós-condição:

Após o login, o administrador visualiza todas as categorias que foram criadas/cadastradas.

Fluxo de eventos principal

1. O usuário faz o login como Admin.
1. O Admin escolhe a opção de visualizar todas as categorias
2. O sistema retorna todas as categorias cadastradas em forma de listagem para o usuário Admin.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.17 [RF021] Adicionar Categoria

Descrição do caso de uso: O usuário Admin poderá adicionar uma categoria para a expansão.

Ator: Admin

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

O usuário terá que estar logado como Admin.

Saídas e pós-condição:

O sistema após o cadastrado, além de listar a nova categoria para o Admin, aparecerá para o cliente/usuário a nova opção de categoria para análise de notícias, com gráficos detalhados e ocorrências de palavras.

Fluxo de eventos principal

1. O usuário Admin, na página de visualização de categorias, irá selecionar a opção de adicionar nova categoria.
2. O sistema irá direcioná-lo para um formulário contendo todos os campos necessários de inserção para o usuário preencher.
3. Após salvar, o usuário Admin deverá ser redirecionado novamente para a página inicial de admin e poderá visualizar a nova categoria cadastrada.

Fluxos secundários

1. Caso ocorra uma falha de comunicação com o banco de dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.18 [RF022] Editar Categoria

Descrição do caso de uso: O Admin deve poder editar as categorias, dessa forma podendo mudar o nicho ou informações da categoria.

Ator: Admin

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

A categoria a ser editada deve estar cadastrada no sistema.

Saídas e pós-condição:

O sistema retorna a categoria atualizada, seja por atualização do nicho ou informações da categoria, de acordo com as alterações realizadas.

Fluxo de eventos principal

1. O Admin escolhe a ação "Editar Categoria".
2. O sistema apresenta uma interface com as palavras-chaves que fazem parte da categoria que foi selecionada.
3. O Admin realiza as alterações desejadas, como podendo alterar o nicho.
4. O sistema atualiza as informações no banco de dados.
5. O sistema retorna a interface de "Categorias".

Fluxos secundários

1. Caso ocorra uma falha na atualização dos dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.19 [RF023] Deletar Categoria

Descrição do caso de uso: O Admin deve poder deletar as categorias, dessa forma podendo deletar categorias não relevantes para o sistema.

Ator: Admin

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

A categoria a ser deletada deve estar cadastrada no sistema.

Saídas e pós-condição:

O sistema retorna as categorias exceto a categoria que foi removida anteriormente.

Fluxo de eventos principal

1. O Admin escolhe a ação "Deletar Categoria".
2. O sistema interface com detalhes da categoria selecionada para remoção.
3. O Admin clica no botão de deletar categoria, o sistema retorna um pop-up com a mensagem "Deseja realmente remover categoria?";
4. O Admin clica no botão deletar categoria.
5. O sistema atualiza as informações no banco de dados.
6. O sistema retorna para interface de "Categorias".

Fluxos secundários

1. Caso ocorra uma falha na atualização dos dados, uma mensagem de erro é retornada para o usuário e a operação é cancelada.

4.1.20 [RF024] Visualizar clientes

Descrição do caso de uso: O Admin deve poder visualizar todos clientes cadastrados no sistema.

Ator: Admin

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

1. O Administrador está autenticado no sistema.

Saídas e pós-condição:

1. Lista de clientes cadastrados.

Fluxo de eventos principal

1. O administrador acessa a funcionalidade de "Visualizar Clientes".
2. O sistema exibe a lista de clientes cadastrados, incluindo informações como nome, contato, endereço, etc.
3. O Administrador pode navegar pela lista de clientes conforme necessário.

Fluxos secundários

1. Nenhum.

4.1.21 [RF025] Adicionar admins

Descrição do caso de uso: O Administrador pode adicionar novos usuários Administradores

Ator: Admin.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

1. O Administrador está autenticado no sistema.

Saídas e pós-condição:

1. Novo usuário administrador adicionado com sucesso ao sistema.

Fluxo de eventos principal

1. O administrador acessa a funcionalidade de "Adicionar administrador".
2. O sistema apresenta um formulário para inserção dos dados do novo Administrador, como nome, e-mail, senha, telefone, CPF e senha.
3. O Administrador preenche os dados do novo Administrador.
4. O administrador confirma a adição do novo administrador.
5. O sistema valida os dados e adiciona o novo administrador ao sistema.
6. O sistema exibe uma mensagem de confirmação de que o novo administrador foi adicionado com sucesso.

Fluxos secundários

1. Se os dados inseridos não estiverem corretos ou incompletos, o sistema exibe uma mensagem de erro e solicita que o Administrador corrija os dados antes de prosseguir.

4.1.22 [RF026] Ativar usuário

Descrição do caso de uso: O Administrador pode ativar um usuário no sistema caso ele esteja desativado por não ter renovado sua assinatura no sistema.

Ator: Admin

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

1. O Administrador está autenticado no sistema.
2. O usuário que será ativado está cadastrado no sistema, mas está inativo.

Saídas e pós-condição:

1. O usuário é ativado com sucesso e pode acessar o sistema.

Fluxo de eventos principal

1. O Administrador acessa a funcionalidade de "Ativar Usuário".

2. O sistema exibe uma lista de usuários inativos.
3. O Administrador seleciona o usuário que deseja ativar.
4. O Administrador confirma a ativação do usuário.
5. O sistema ativa o usuário selecionado.
6. O sistema exibe uma mensagem de confirmação de que o usuário foi ativado com sucesso.

Fluxos secundários

1. Se o usuário selecionado já estiver ativo, o sistema exibe uma mensagem de aviso informando que o usuário já está ativo e não realiza nenhuma ação adicional.

4.1.23 [RF027] Desativar usuário

Descrição do caso de uso: O Administrador pode desativar um usuário no sistema caso ele não tenha renovado sua assinatura no sistema.

Ator: Admin.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

1. O Administrador está autenticado no sistema.
2. O usuário que será desativado está cadastrado e ativo no sistema.

Saídas e pós-condição:

1. O usuário é desativado com sucesso e não pode mais acessar o sistema.

Fluxo de eventos principal

1. O Administrador acessa a funcionalidade de "Desativar Usuário".
2. O sistema exibe uma lista de usuários ativos.
3. O Administrador seleciona o usuário que deseja desativar.
4. O Administrador confirma a desativação do usuário.
5. O sistema desativa o usuário selecionado.
6. O sistema exibe uma mensagem de confirmação de que o usuário foi desativado com sucesso.

Fluxos secundários

1. Se o usuário selecionado já estiver desativado, o sistema exibe uma mensagem de aviso informando que o usuário já está desativado e não realiza nenhuma ação adicional.

4.1.24 [RF028] Buscar usuário

Descrição do caso de uso: O administrador pode buscar um usuário no sistema a fim de ativar ou desativar do sistema.

Ator: Admin.

Prioridade: ■ Essencial □ Importante □ Desejável

Entradas e pré-condições:

1. O Administrador está autenticado no sistema.

Saídas e pós-condição:

1. Exibição dos resultados da busca conforme critérios fornecidos.

Fluxo de eventos principal

1. O Administrador acessa a funcionalidade de "Buscar Usuário".
2. O sistema apresenta um campo de busca onde o Administrador pode inserir critérios de busca, como nome, e-mail, ou outro identificador.
3. O Administrador insere os critérios de busca desejados.
4. O Administrador solicita a execução da busca.
5. O sistema realiza a busca e exibe os resultados correspondentes aos critérios fornecidos.
6. O Administrador pode visualizar os detalhes do usuário encontrado ou realizar outras ações, conforme necessário.

Fluxos secundários

1. Se nenhum usuário corresponder aos critérios de busca fornecidos, o sistema não exibe nenhum usuário.

5. Requisitos não-funcionais

5.1 Usabilidade

Esta seção descreve os requisitos não funcionais associados à facilidade de uso da interface com o usuário.

5.1.1 [NF001] Interface Amigável

O sistema terá uma interface amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes.

Prioridade: ■ Essencial □ Importante □ Desejável

5.1.2 [NF002] Componentes WEB

A interface deverá utilizar elementos comuns a usuários de sistemas web, como campos de texto, *combo-boxes*, *links* e botões, cards. A idéia é focar nos aspectos operacionais e na apresentação limpa e responsiva dos elementos visuais, de modo que o sistema fique intuitivo e de fácil uso.

Prioridade: ■ Essencial □ Importante □ Desejável

5.2 Software

Esta seção descreve os requisitos não-funcionais associados aos softwares que devem ser utilizados para o desenvolvimento do sistema.

5.2.1 [NF003] Banco de Dados "NoSQL" MongoDB

O sistema deve utilizar um banco de dados não relacional, MongoDB, para fazer a permanência de dados.

Prioridade: ■ Essencial □ Importante □ Desejável

5.2.2 [NF004] Linguagem Python

Visando criar um produto com maior desempenho em tarefas de Web Scraping, deve-se adotar Python como linguagem principal de desenvolvimento, seguindo cuidadosamente as técnicas de orientação a objetos.

Prioridade: ■ Essencial □ Importante □ Desejável

5.2.3 [NF005] Framework NodeJs

Visando o desenvolvimento rápido e uma boa qualidade do sistema, será utilizado para o desenvolvimento da aplicação o framework NodeJs.

Prioridade: ■ Essencial □ Importante □ Desejável

5.3 Desempenho

Esta seção descreve os requisitos não-funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

5.3.1 [NF005] Agilidade na Execução das Operações

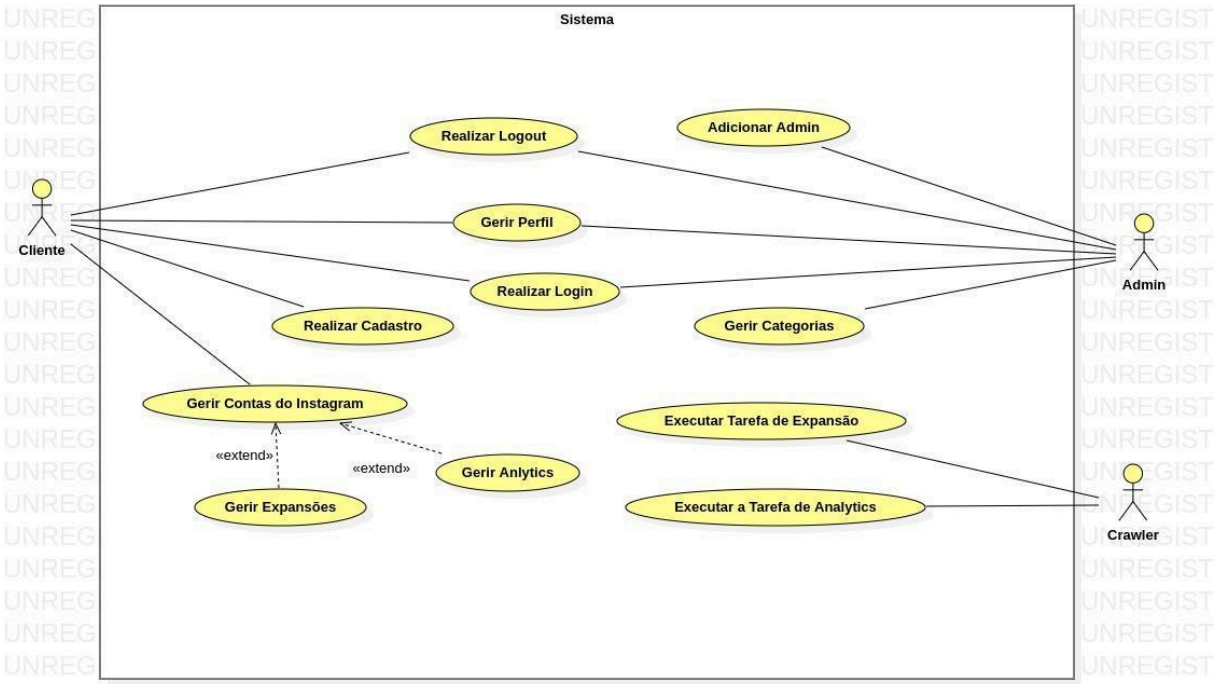
O sistema deve executar as operações no menor tempo possível, visando dar uma maior agilidade ao processo.

Prioridade: □ Essencial ■ Importante □ Desejável

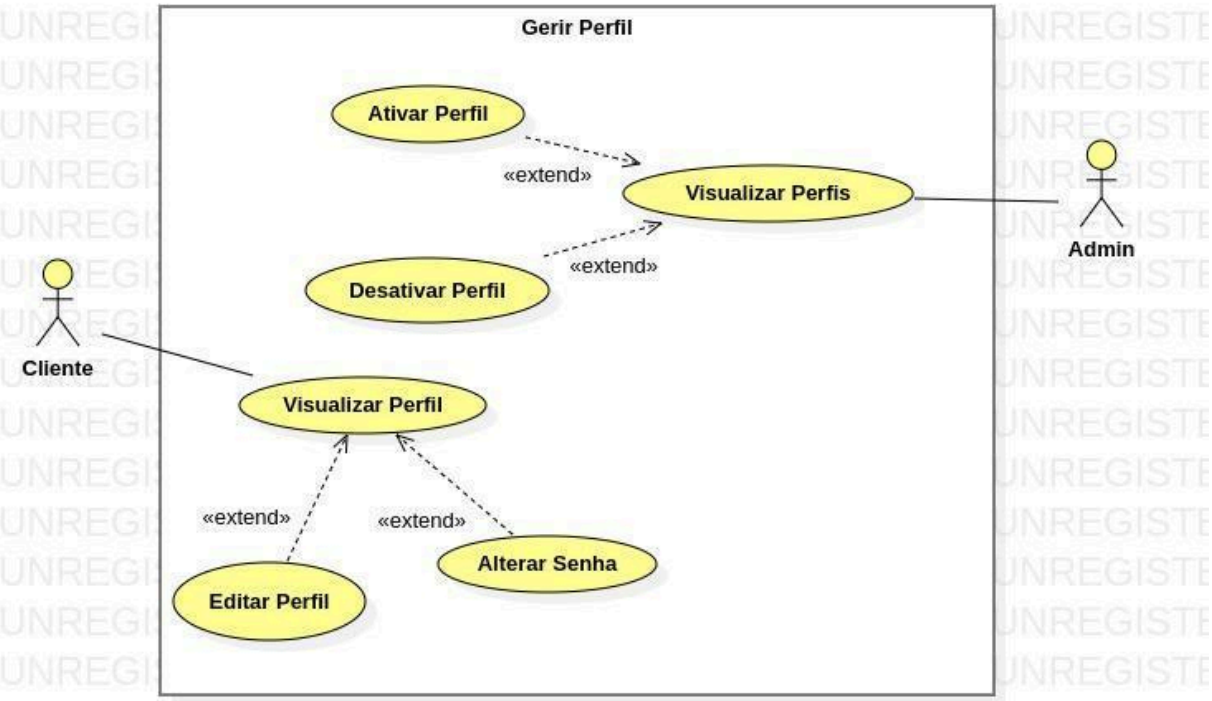
6. Diagramas

6.1 Diagramas de Caso de Uso

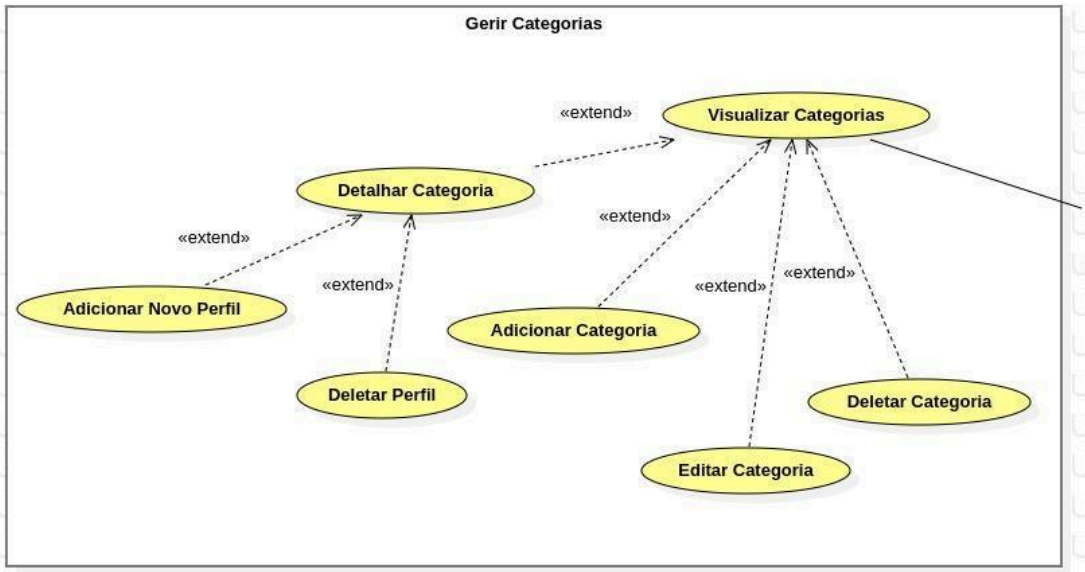
5.1 Sistema



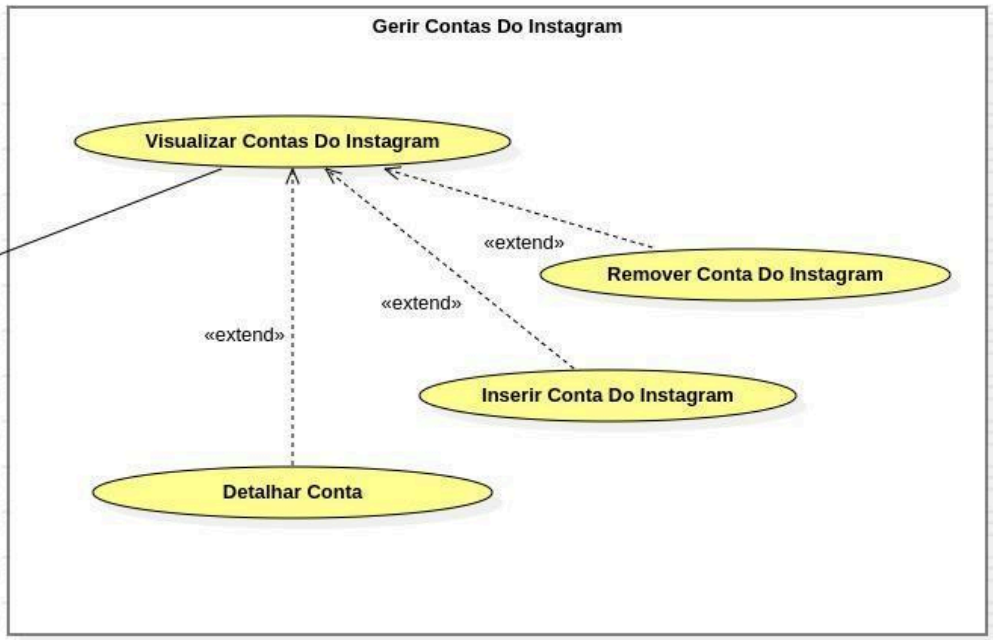
5.2 Gerir Perfil



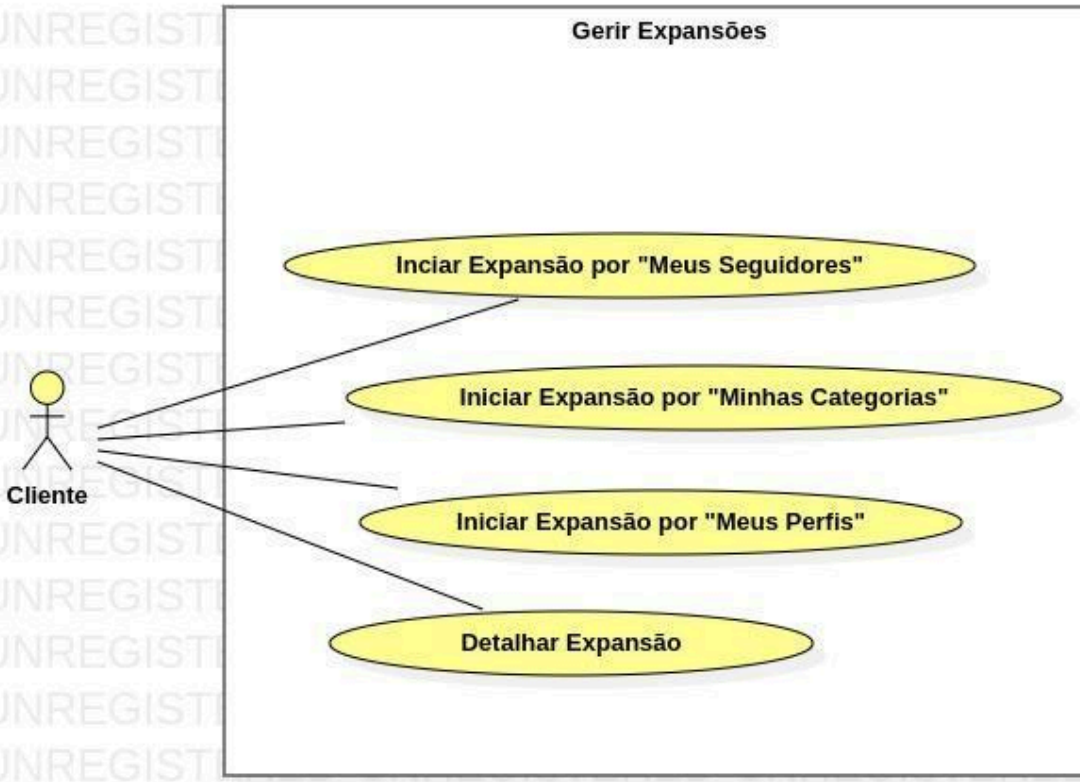
5.3 Gerir Categorias



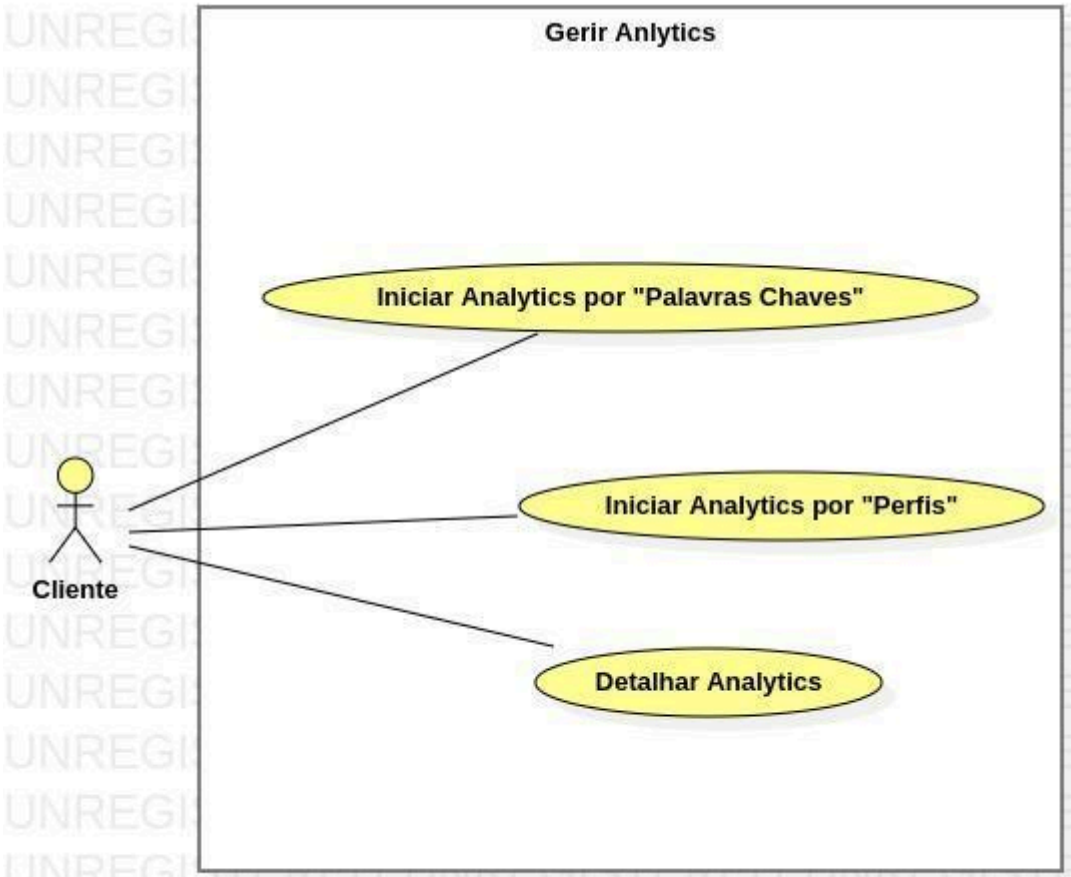
5.4 Gerir Contas do Instagram



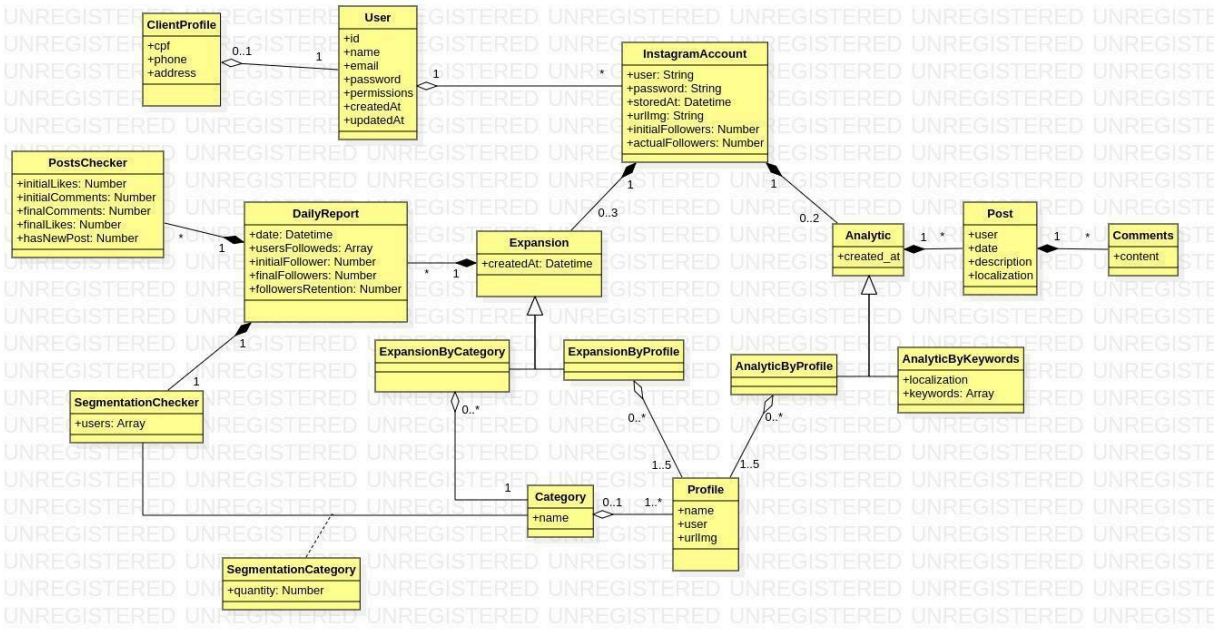
5.5 Gerir Expansões



5.6 Gerir Analytics



6.2 Diagrama de Classe



7. Referências

[1] Rumbaugh, J., "UML – Guia do Usuário". Editora Campus.

[2] Alura. Node.js: O que é e para que serve? 2023 https://www.alura.com.br/artigos/node-js . Acesso em: 03 dez. 2023.

[3] Node.js. About | Node.js. 2023. https://nodejs.org/en/about. Acesso em: 03 dez. 2023.

[4] Selenium. History | Selenium. 2023. https://www.selenium.dev/history/ . Acesso em: 03 dez. 2023.

[5] TREINAWEB. O que é Selenium? 2023.https://www.treinaweb.com.br/blog/o-que-e-selenium . Acesso em: 03 dez. 2023.

[6] Beautiful Soup Documentation. Beautiful Soup Documentation. 2023. Disponível em: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Acesso em: 04 dez. 2023.

[7] MongoDB Documentation. MongoDB Documentation. 2023. Disponível em: https://docs.mongodb.com/. Acesso em: 04 dez. 2023.

[8] IBM. MongoDB | IBM. 2023. Disponível em: https://www.ibm.com/br-pt/topics/mongodb. Acesso em: 04 dez. 2023.