



2023
INTELI

INSTALAÇÃO

MONITORAMENTO DE CAMPANHAS DE MARKETING
UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL



Autores:

- Dayllan Alho
- Eric Tachdjian
- Gabriela da Silva
- Giovanna Furlan
- Lucas Britto
- Michel Mansur

Orientador:

- Renato Penha

Coordenador:

- Egon Daxbacher

Instituição:

- Instituto de Tecnologia e Liderança

NATURALFIVE

SUMÁRIO

Introdução.....	4
Deploy Local.....	5
Configuração Jupyter Notebook.....	5
Backend.....	10
Tecnologias Utilizadas.....	10
Ambiente de Programação.....	11
Instruções VS Code.....	13
Bibliotecas.....	14
Executar Backend.....	21
Frontend.....	22
Tecnologias Utilizadas.....	22
Executar Frontend.....	23
Integração.....	26
Rotas Backend.....	27



INTRODUÇÃO

Este documento fornece instruções passo a passo sobre como instalar e integrar o projeto, juntamente com todas as suas dependências e bibliotecas. Além disso, explica-se detalhadamente como realizar a conexão entre o frontend e o backend.

Antes de iniciar o processo de instalação, exibe-se as dependências e bibliotecas necessárias para o projeto, juntamente com suas respectivas versões. Os usuários receberão instruções sobre como instalar e configurar cada uma delas.

Uma parte fundamental do documento é a integração entre o frontend e o backend. Aqui, os usuários encontrarão uma descrição clara da interação entre os dois, incluindo informações sobre a comunicação e a definição de endpoints de API.

Após a conclusão das etapas de instalação e integração, o documento fornece instruções sobre como executar o projeto, detalhando a sequência de comandos necessários e demonstrando como testar o projeto para garantir seu funcionamento adequado.



DEPLOY LOCAL

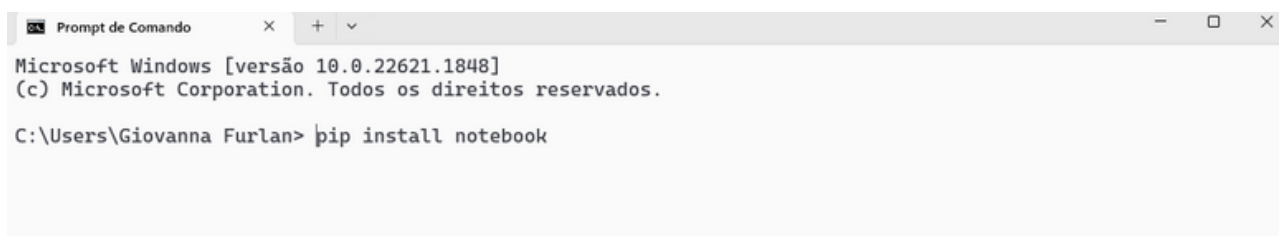
20 de junho de 2023 às 18h.

CONFIGURAÇÃO DO JUPYTER NOTEBOOK

Para o deploy local do projeto, utilizou-se o jupyter notebook, segue a configuração feita.

- Acessar o site : <https://jupyter.org/install>;
- Navegar até o índice "Jupyter Notebook";
- Abrir o `cmd` do computador;
- Digitar o comando "`pip install notebook`".

Figura 1 : Instalação notebook



```
Prompt de Comando
Microsoft Windows [versão 10.0.22621.1848]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\Giovanna Furlan> pip install notebook
```

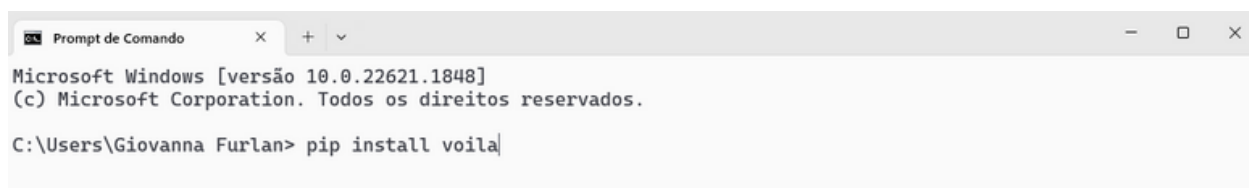
Fonte: Autores.

Após baixar as dependências:

- Digite o comando "`cls`", para limpar o terminal.
- Volte no navegador e navegue até o índice "Voilà";
- Abra o `cmd` do computador novamente;
- Digitar o comando "`pip install voila`".



Figura 2 : Instalação Voilà

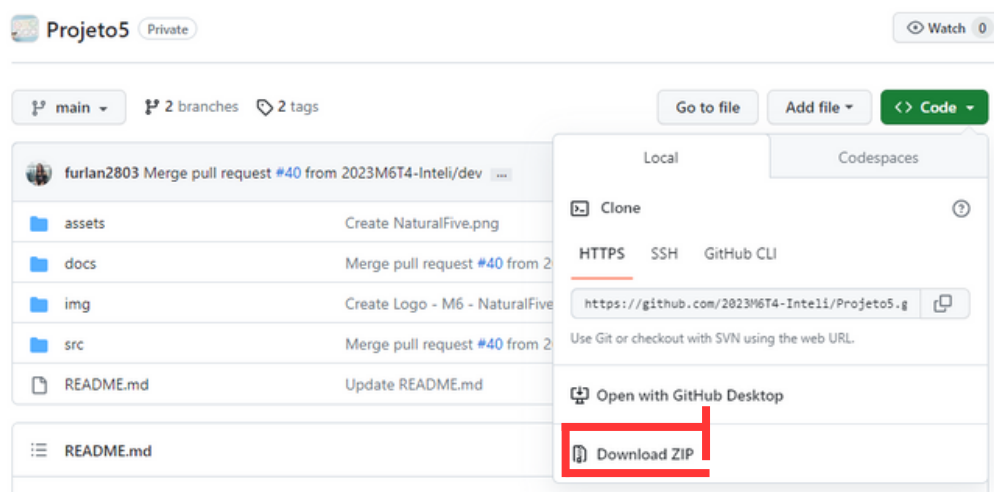


Fonte: Autores.

Após baixar as dependências:

- Feche o **cmd**;
- Acesse o site : "<https://github.com/2023M6T4-Inteli/Projeto5>";
- Clique no botão "<> Code";
- Clique em "**Download ZIP**"

Figura 3 : Download pasta GitHub



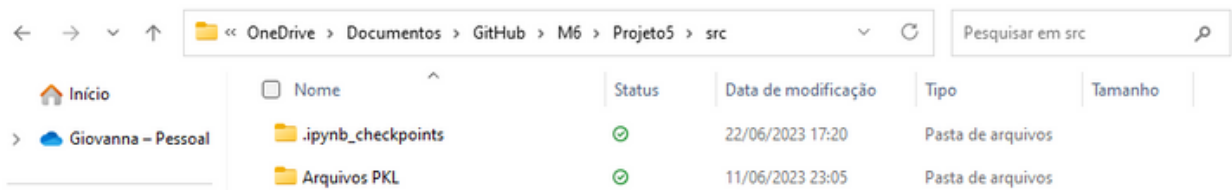
Fonte: Autores.

Após o download do arquivo:

- Descompacte a pasta;
- Abra-a e acesse o seguinte caminho: "**Projeto5\src**";



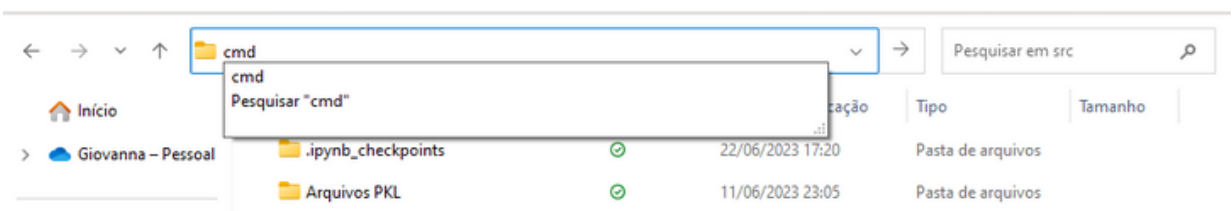
Figura 4 : Caminho do Projeto



Fonte: Autores.

- Digite **cmd** no local do caminho;

Figura 5 : Abrindo cmd



Fonte: Autores.

- Digite no **cmd** "jupyter notebook"

Figura 6 : Rodando o Jupyter



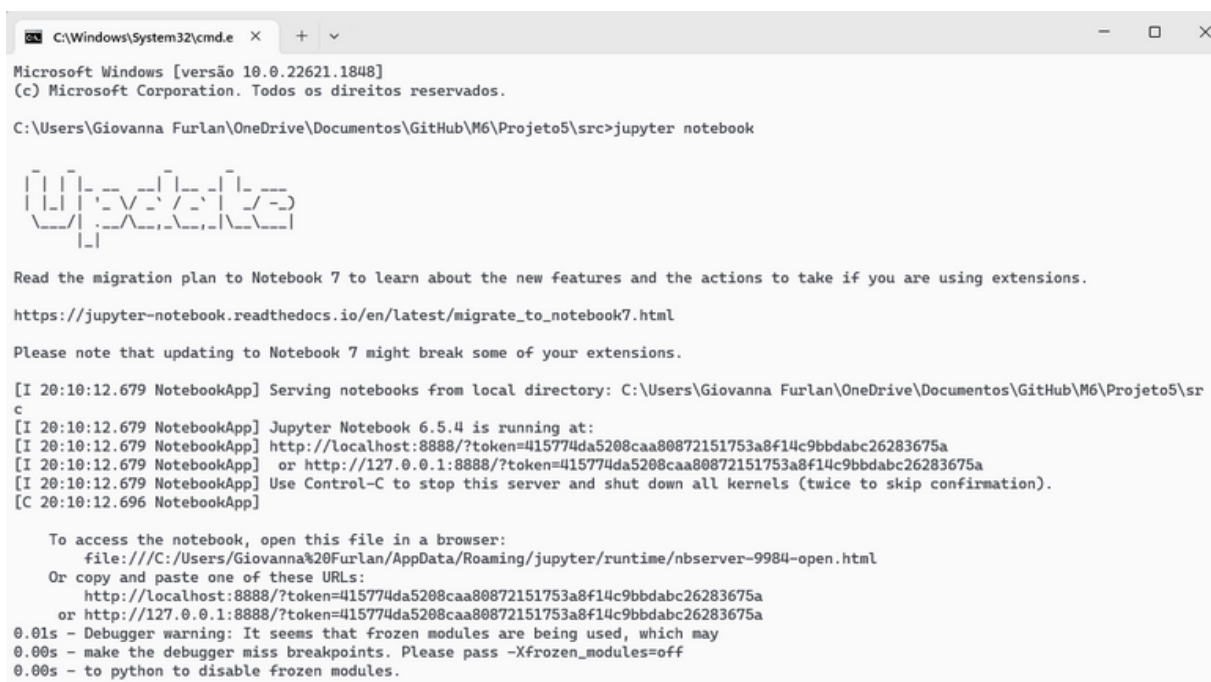
Fonte: Autores.



Após rodar o comando abrirá uma aba em seu navegador, com o deploy do projeto realizado.

- O cmd após o comando "jupyter notebook";

Figura 7 : cmd após rodar o Jupyter



```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [versão 10.0.22621.1848]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Giovanna Furlan\OneDrive\Documentos\GitHub\M6\Projeto5\src>jupyter notebook

Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions.
https://jupyter-notebook.readthedocs.io/en/latest/migrate_to_notebook7.html

Please note that updating to Notebook 7 might break some of your extensions.

[I 20:10:12.679 NotebookApp] Serving notebooks from local directory: C:\Users\Giovanna Furlan\OneDrive\Documentos\GitHub\M6\Projeto5\src
[I 20:10:12.679 NotebookApp] Jupyter Notebook 6.5.4 is running at:
[I 20:10:12.679 NotebookApp] http://localhost:8888/?token=415774da5208caa80872151753a8f14c9bbdabc26283675a
[I 20:10:12.679 NotebookApp] or http://127.0.0.1:8888/?token=415774da5208caa80872151753a8f14c9bbdabc26283675a
[I 20:10:12.679 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[C 20:10:12.696 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Giovanna%20Furlan/AppData/Roaming/jupyter/runtime/nbserver-9984-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=415774da5208caa80872151753a8f14c9bbdabc26283675a
or http://127.0.0.1:8888/?token=415774da5208caa80872151753a8f14c9bbdabc26283675a
0.01s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
```

Fonte: Autores.

Após realizar o deploy local do projeto, você poderá acessá-lo no endereço "http://localhost:8888/tree", através do navegador. Essa URL leva à página do Jupyter Notebook, onde você pode interagir com o projeto, explorar os arquivos e executar o código fonte. Certifique-se de que o deploy tenha sido concluído com sucesso antes de acessar o endereço.

Figura 8 : Página Jupyter Notebook



Fonte: Autores.



BACKEND

20 de junho de 2023 às 19h.

TECNOLOGIAS UTILIZADAS

Python

Linguagem de programação de alto nível, interpretada. Suporta múltiplos paradigmas de programação, incluindo programação orientada a objetos, programação funcional e programação imperativa. Também pode ser utilizada em desenvolvimento web, ciência de dados, automação de tarefas, inteligência artificial, entre outros. Neste projeto utilizou-se a versão utilizada - 3.11, o endereço para realizar o download é "<https://www.python.org/downloads/>"

Postman

Postman é uma plataforma de colaboração para desenvolvimento de APIs (Application Programming Interfaces). Ele permite que os desenvolvedores criem, testem, documentem e compartilhem APIs de maneira fácil. Com o Postman, você pode enviar solicitações HTTP para uma API, visualizar e inspecionar as respostas, automatizar fluxos de trabalho e colaborar com outros membros da equipe.

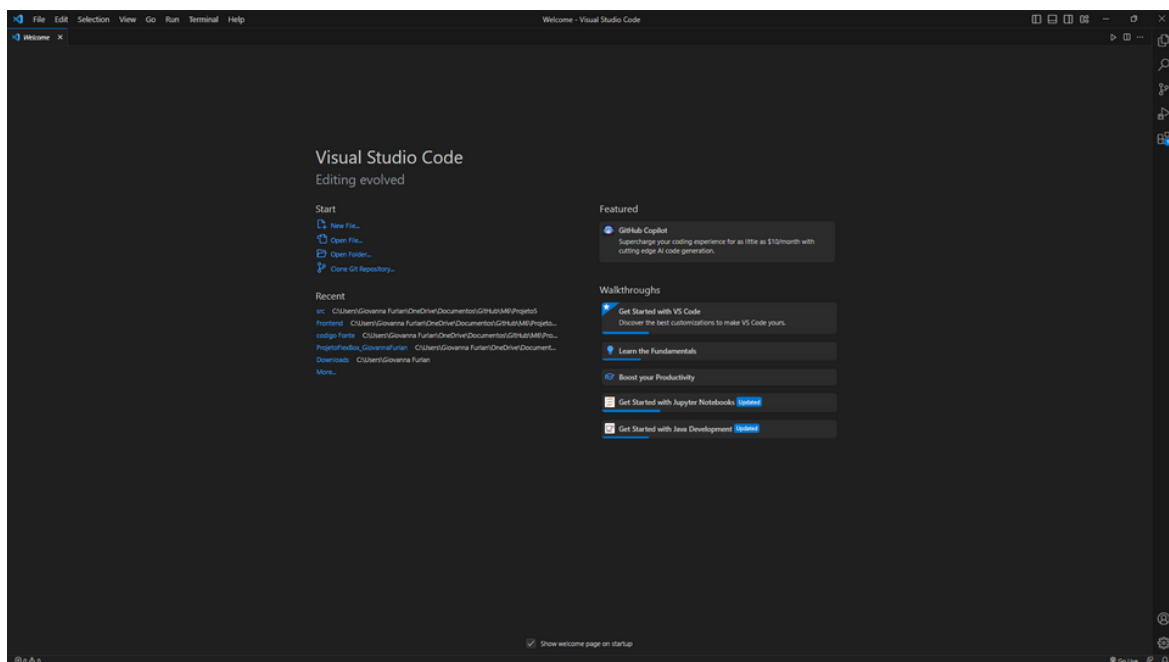


AMBIENTE DE PROGRAMAÇÃO

Apresenta-se abaixo um passo a passo para baixar o Visual Studio Code (VS Code).

- Entre em um navegador e acesse: <https://code.visualstudio.com>
- Clique no botão "Download" para baixar o instalador do VS Code compatível com o seu sistema operacional (Windows, macOS ou Linux).
- Após o download ser concluído, execute o instalador e siga as instruções de instalação padrão para o seu sistema operacional.
- Após a conclusão da instalação, abra o VSCode.

Figura 9 : Interface VS Code



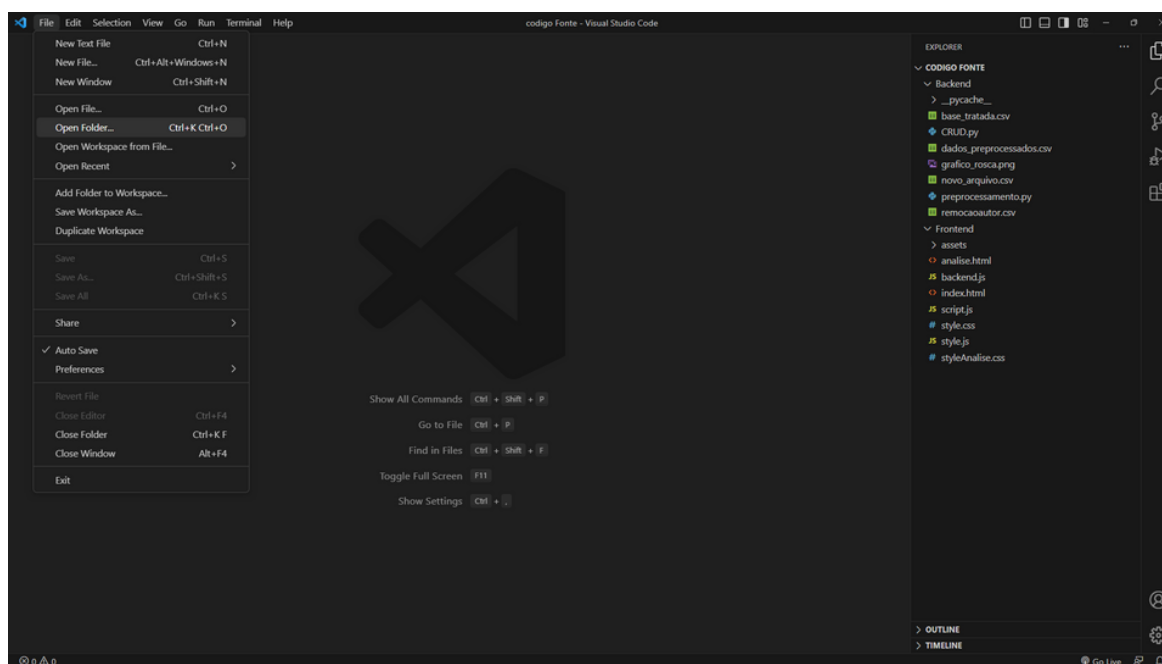
Fonte: Autores.



Agora, para abrir uma pasta no VS Code, acesse:

- Menu superior do VS Code, clique em "File" (Arquivo) e selecione "Open Folder" (Abrir Pasta). Em seguida, navegue até a pasta desejada e clique em "Open" (Abrir).
- Após selecionar a pasta, ela será aberta no VS Code. Você verá a estrutura de arquivos e pastas da pasta no painel lateral esquerdo do VS Code.

Figura 10 : Projeto no VS Code



Fonte: Autores.

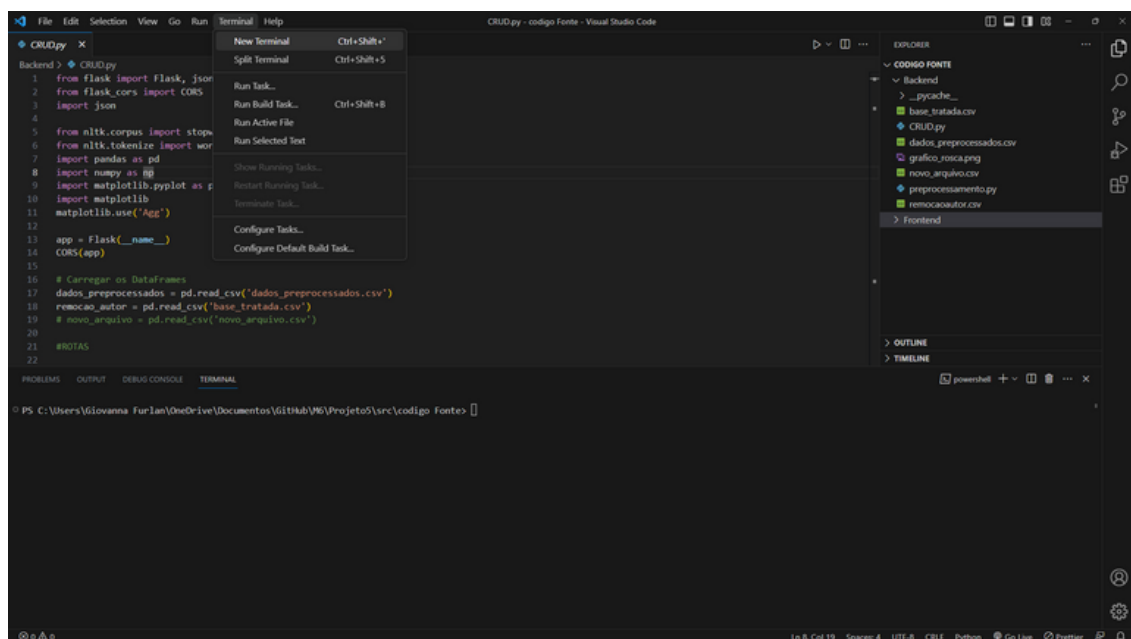


INSTRUÇÕES VS CODE

Apresenta-se abaixo um passo a passo para executar o backend da solução.

- Após abrir os arquivos no VS Code, navegue até a pasta do código fonte.
- Dentro da pasta do código fonte, você encontrará uma subpasta chamada "backend". Acesse essa pasta.
- Dentro da pasta "backend", localize o arquivo chamado "CRUD.py". Abra este arquivo no VS Code.
- Agora, é necessário abrir um terminal no VS Code e navegar até a pasta "backend". No menu superior do VS Code, clique em "Terminal" e selecione "New Terminal" (Novo Terminal).

Figura 11 : Terminal do VS Code



Fonte: Autores.




BIBLIOTECAS

Realize a instalação das dependências abaixo no terminal do seu VS Code antes de iniciar a execução do backend.

emoji (2.2.0)


A biblioteca emoji fornece um conjunto de funções para trabalhar com emojis em Python. Permite a pesquisa, conversão e manipulação de emojis em textos. A versão utilizada é a 2.2.0. Para instalar a biblioteca emoji no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install emoji==2.2.0
```

enelvo (0.15)

A biblioteca envelvo é um corretor ortográfico para o idioma português. Ela implementa um modelo estatístico para sugerir correções de palavras escritas incorretamente. A versão utilizada é a 0.15. Para instalar a biblioteca envelvo no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install envelvo==0.15
```



Flask (2.3.2)

Framework para desenvolvimento web em Python. Permite criar aplicativos web de forma rápida e simples, com suporte a roteamento, gerenciamento de solicitações e respostas HTTP, e integração com outros componentes. A versão utilizada é a 2.3.2. Para instalar o Flask no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install Flask==2.3.2
```

Flask_Cors (3.0.10)

Extensão para o Flask que fornece suporte para lidar com solicitações de recursos cruzados (CORS - Cross-Origin Resource Sharing). Simplifica a configuração das políticas de CORS em aplicativos Flask. A versão utilizada é a 3.0.10. Para instalar a biblioteca Flask_Cors no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install Flask_Cors==3.0.10
```



gensim (4.3.1)

Biblioteca de processamento de linguagem natural (NLP - Natural Language Processing) em Python. Fornece algoritmos para modelagem de tópicos, similaridade de documentos, criação de word embeddings, etc. A versão utilizada é a 4.3.1. Para instalar a biblioteca gensim no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install gensim==4.3.1
```

keras (2.12.0)

Biblioteca de alto nível para aprendizado de máquina e redes neurais em Python. Fornece uma interface simples e intuitiva para criar e treinar redes neurais profundas. A versão utilizada é a 2.12.0. Para instalar a biblioteca keras no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install keras==2.12.0
```



matplotlib (3.7.1)

Biblioteca de visualização de dados em Python. Ele fornece uma ampla variedade de gráficos e visualizações, permitindo que você crie gráficos estáticos, interativos e animados. A versão utilizada é a 3.7.1. Para instalar a biblioteca matplotlib no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install matplotlib==3.7.1
```

nlk (Natural Language Toolkit) (3.8.1)

Biblioteca em Python para processamento de linguagem natural. Ele fornece uma coleção de ferramentas e recursos para tarefas como tokenização, lematização, análise sintática, classificação de texto, etc. A versão utilizada é a 3.8.1. Para instalar a biblioteca nltk no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install nltk==3.8.1
```



numpy (1.23.5)

Biblioteca fundamental para computação científica em Python. Ela fornece suporte para arrays multidimensionais, funções matemáticas de alto desempenho, manipulação de dados e integração com outras bibliotecas. A versão utilizada é a 1.23.5. Para instalar a biblioteca numpy no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install numpy==1.23.5
```

pandas (2.0.2)

Biblioteca em Python para análise e manipulação de dados. Ele fornece estruturas de dados flexíveis e eficientes, como o DataFrame, para lidar com tabelas e séries temporais, além de oferecer funcionalidades para limpeza, transformação e agregação de dados. A versão utilizada é a 2.0.2. Para instalar a biblioteca pandas no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install pandas==2.0.2
```



Requests (2.31.0)

Biblioteca HTTP elegante e simples para Python. Ela permite que você envie solicitações HTTP de maneira fácil, fazendo requisições a APIs ou obtendo conteúdo de páginas da web. A versão utilizada é a 2.31.0. Para instalar a biblioteca Requests no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install requests==2.31.0
```

tensorflow (2.12.0)

Biblioteca de código aberto para aprendizado de máquina e computação numérica em Python. Ele é amplamente utilizado para construir e treinar modelos de aprendizado de máquina, especialmente redes neurais. A versão utilizada é a 2.12.0. Para instalar a biblioteca tensorflow no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install tensorflow==2.12.0
```



textblob (0.17.1)

Biblioteca em Python para processamento de linguagem natural (NLP). Fornece uma API simples para realizar tarefas como análise de sentimentos, correção ortográfica, extração de frases-chave, etc. A versão utilizada é a 0.17.1. Para instalar a biblioteca textblob no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install textblob==0.17.1
```

Unidecode (1.3.6)

Utilizada para converter caracteres Unicode em caracteres ASCII. Ela é útil quando se trabalha com texto em diferentes idiomas e é necessário normalizar caracteres especiais. A versão utilizada é a 1.3.6. Para instalar a biblioteca Unidecode no terminal do VS Code, pode-se usar o seguinte comando:



```
pip install unidecode==1.3.6
```



EXECUTAR BACKEND

Após a instalação das dependências acima, abra o terminal novamente. Se o terminal não estiver na pasta "backend", altere o diretório de trabalho usando o comando:

A terminal window with a light gray background and three colored window control buttons (red, yellow, green) in the top left corner. The text 'cd backend' is displayed in a monospaced font, with 'cd' in red and 'backend' in black.

```
cd backend
```

Uma vez que esteja no diretório correto no terminal (a pasta "backend"), execute o comando a seguir para iniciar o backend:

A terminal window with a light gray background and three colored window control buttons (red, yellow, green) in the top left corner. The text 'python CRUD.py' is displayed in a monospaced font, with 'python' in red and 'CRUD.py' in blue.

```
python CRUD.py
```

Agora, o backend do projeto está em execução. Certifique-se de que todas as dependências necessárias estejam instaladas corretamente e que você esteja usando a versão correta do Python. Caso ocorra algum erro durante a execução, verifique as mensagens de erro no terminal.



FRONTEND

21 de junho de 2023 às 8h.

TECNOLOGIAS UTILIZADAS

HTML (HyperText Markup Language) 5

Linguagem de marcação padrão para criar e estruturar conteúdo em páginas da web. Fornece elementos e tags que definem a estrutura, os componentes e o layout de uma página da web.

CSS (Cascading Style Sheets)

Linguagem de estilo utilizada para definir a aparência e o layout de uma página da web. Permite definir cores, fontes, margens, posicionamento de elementos e outros estilos visuais. O CSS trabalha em conjunto com o HTML para fornecer uma apresentação estilizada e consistente para as páginas da web.

JavaScript (JS)

Linguagem de programação que permite adicionar interatividade e comportamento dinâmico às páginas da web. Permite criar scripts que respondem a eventos, manipulam elementos HTML, fazem solicitações a servidores, validam formulários, etc. Desempenha um papel fundamental na criação de aplicativos web interativos.

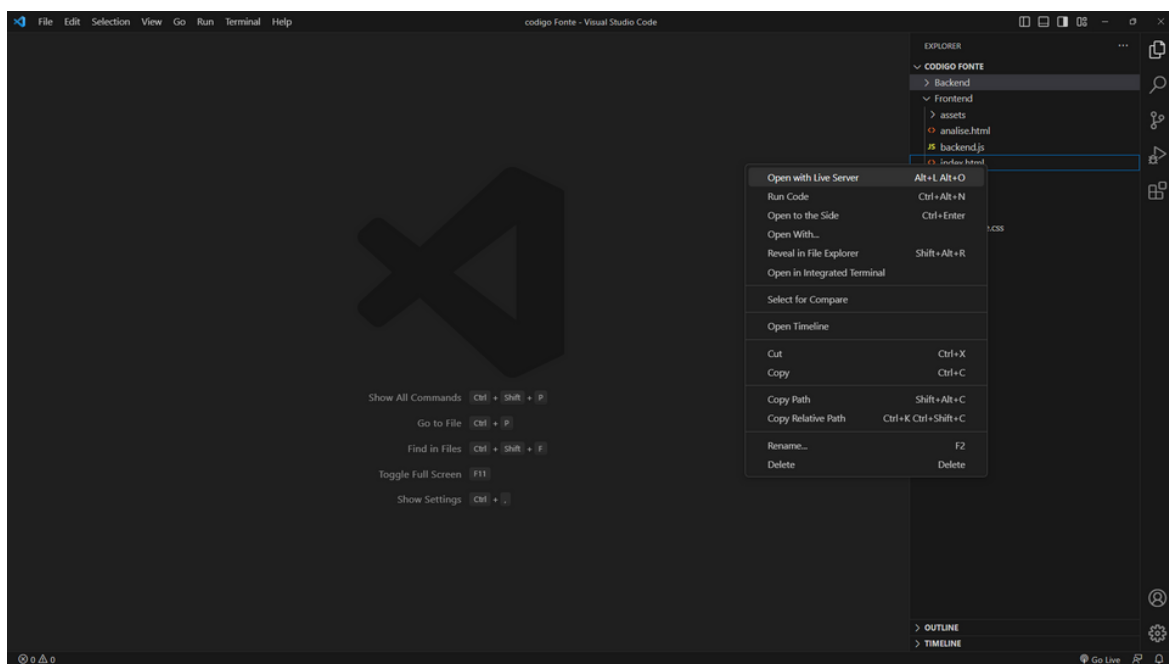


EXECUTAR FRONTEND

- Abra a pasta do projeto no VS Code, localize o arquivo chamado "index.html", dentro da pasta "frontend".
- Clique com o botão direito do mouse no arquivo "index.html" e selecione a opção "Open with Live Server" (Abrir com o Live Server).

Observação: O "Live Server" é uma extensão do VS Code que permite visualizar as alterações no frontend em tempo real. Ele atualiza automaticamente a página do navegador sempre que você salva um arquivo no VS Code. Caso não possua a extensão "Live Server" instalada, você pode instalá-la a partir do marketplace do VS Code.

Figura 12 : Live Share no VS Code

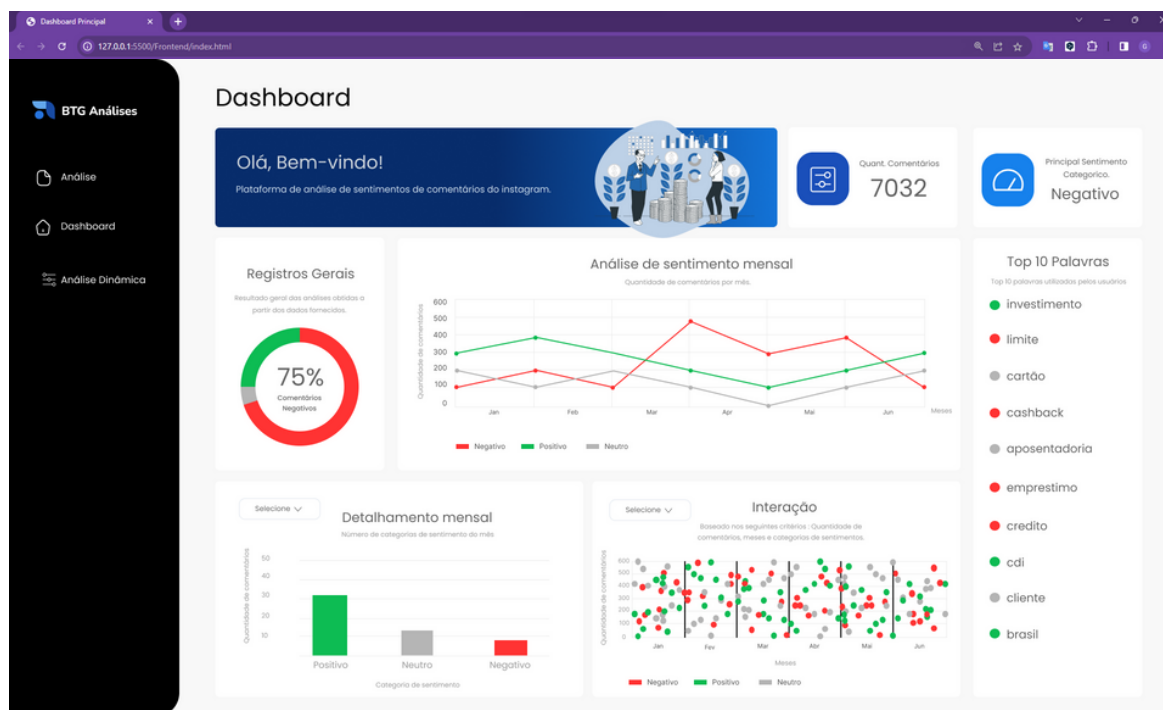


Fonte: Autores.



Quando clicado, uma nova janela abrirá no navegador padrão, exibindo a página "index.html" renderizada.

Figura 13 : Exemplo visualização Frontend

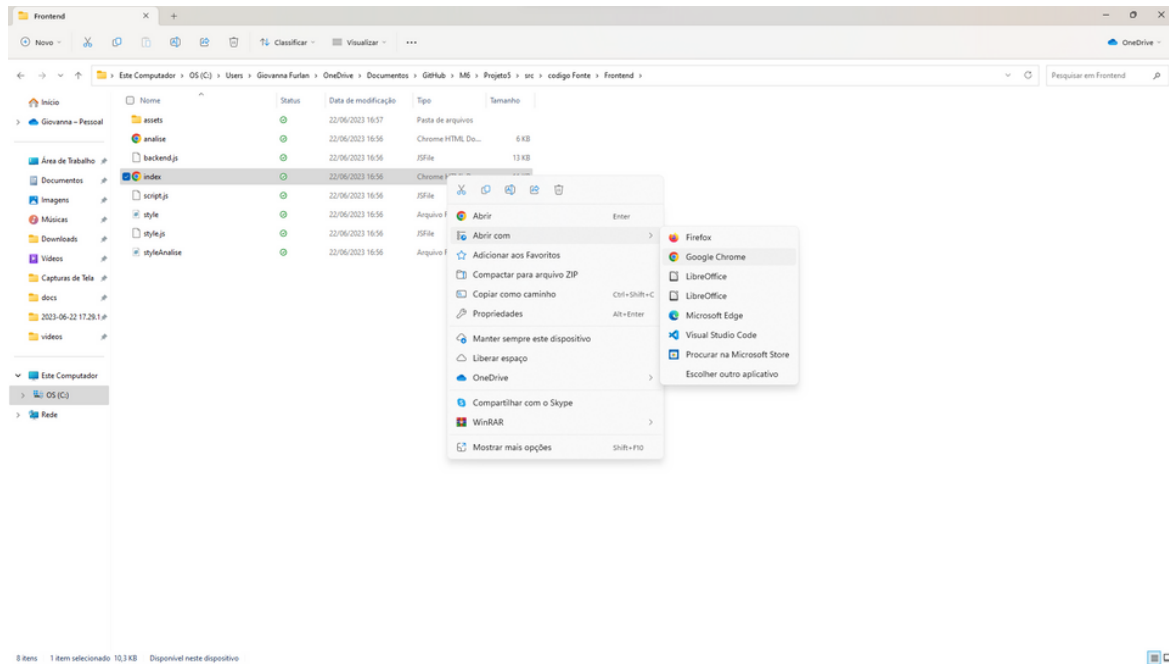


Fonte: Autores.

Caso prefira abrir o frontend sem uma extensão específica, dentro dos documentos de seu computador localize a pasta frontend e o arquivo "index.html", clique com o botão direito do mouse e selecionar a opção "Open with" (Abrir com) e, em seguida, escolher o navegador da sua preferência.



Figura 14 : Abrir Frontend fora do VS Code



Fonte: Autores.



INTEGRAÇÃO

22 de junho de 2023 às 16h.

A Integração entre o Frontend (HTML, CSS e JS) e o Backend (Flask) tem como objetivo puxar os dados de um arquivo CSV e gerar gráficos interativos para análise de sentimentos de campanhas de marketing do BTG Pactual.

O backend, construído com Flask, fornece as rotas e lógica necessárias para processar as requisições do frontend e retornar os dados apropriados. Já o frontend, é responsável por solicitar os dados ao backend e exibir os gráficos gerados. Ele utiliza JavaScript para fazer as chamadas assíncronas às rotas do backend e manipular os dados, utilizando a biblioteca Chart.js para criar os gráficos interativos com os dados fornecidos pelo backend.

Tem que ocorrer uma comunicação adequada entre o frontend e o backend, pois assim define-se a estrutura dos dados que serão enviados e recebidos nas requisições. Neste caso, utilizou-se arquivo JSON para facilitar o processamento e manipulação dos dados.



ROTAS BACKEND

Abaixo apresenta-se as rotas e as lógicas utilizadas para a integração do front-end com o back-end. Caso seja necessário mais detalhes, elas se encontram na pasta código fonte.

- Rota principal ("/"): Retorna um JSON indicando que a API está conectada.
- Rota para contar a quantidade de comentários ("/quantidadecomentarios"): Retorna o número de comentários presentes no DataFrame "dados_preprocessados". palavra: A função comentarios_por_palavra() utiliza as palavras principais obtidas na rota /palavras para realizar a contagem de sentimentos associados a cada uma dessas palavras no DataFrame dados_preprocessados. O resultado é retornado na rota /sentimentoPalavra.
- Lógica de análise de sentimentos por mês: A função sentimento_por_mes() realiza a análise de sentimentos por mês, contando a quantidade de sentimentos positivos, negativos e neutros para cada mês. O resultado é retornado na rota /sentimento-mes.
- Lógica de contagem de categorias por mês: A função categorias_por_mes() realiza a contagem das categorias (sentimentos) por mês, contando a quantidade de cada categoria para cada mês. O resultado é retornado na rota /categorias-por-mes.



- Rota para obter as categorias de sentimento por mês ("/categorias-por-mes"): Agrupa os comentários por mês e retorna a quantidade de comentários classificados como POSITIVE, NEGATIVE e NEUTRAL para cada mês.
- Rota para obter a quantidade de comentários por tipo de interação por dia ("/comentarios-dia"): Agrupa os comentários por dia e retorna a quantidade de comentários, marcações e respostas classificadas como POSITIVE, NEGATIVE e NEUTRAL para cada dia.
- Rota para buscar uma palavra específica ("/buscar_palavra"): Recebe uma palavra enviada por um formulário POST e verifica a quantidade de vezes que essa palavra aparece nos textos processados presentes no DataFrame "remocao_autor". Retorna a quantidade de ocorrências da palavra ou uma mensagem de erro caso a palavra não seja encontrada.

Além das rotas existentes, o código contém algumas lógicas adicionais fora das rotas. Abaixo apresenta suas descrições.

- Carregamento dos DataFrames: No início do código, há a carga de dois DataFrames: dados_preprocessados e remocao_autor. Eles são carregados a partir de arquivos CSV usando a biblioteca pandas. Esses DataFrames contêm os dados com os quais as rotas irão trabalhar.



- Lógica para contagem de sentimentos: Existe uma lógica para contar a quantidade de cada sentimento presente no DataFrame dados_preprocessados. Essa lógica é usada nas rotas /dados-relacionados e /sentimentoPalavra.
- Lógica de pré-processamento de palavras: A função principal_palavras() realiza um pré-processamento dos textos contidos no DataFrame dados_preprocessados. Ela remove stopwords em português, palavras indesejadas e conta a frequência de ocorrência das palavras restantes. As 10 palavras mais frequentes são retornadas na rota /palavras.
- Lógica de análise de sentimento por palavra: A função comentarios_por_palavra() utiliza as palavras principais obtidas na rota /palavras para realizar a contagem de sentimentos associados a cada uma dessas palavras no DataFrame dados_preprocessados. O resultado é retornado na rota /sentimentoPalavra.
- Lógica de análise de sentimentos por mês: A função sentimento_por_mes() realiza a análise de sentimentos por mês, contando a quantidade de sentimentos positivos, negativos e neutros para cada mês. O resultado é retornado na rota /sentimento-mes.



- Lógica de contagem de categorias por mês: A função `categorias_por_mes()` realiza a contagem das categorias (sentimentos) por mês, contando a quantidade de cada categoria para cada mês. O resultado é retornado na rota `/categorias-por-mes`.
- Lógica de contagem de comentários por dia: A função `get_comentarios_por_dia()` agrupa os comentários por dia e conta a quantidade de cada tipo de interação (comentários, marcações e respostas) em cada dia. O resultado é retornado na rota `/comentarios-dia`.
- Lógica de busca de palavra: A função `buscar_palavra()` recebe uma palavra como entrada, processa o texto no DataFrame `remocao_autor` e conta a quantidade de ocorrências da palavra especificada. O resultado é retornado na rota `/buscar_palavra`.

