

---

**INSTITUTO DE COMPUTAÇÃO - UNICAMP**

**MC536 – Banco de Dados: Teoria e Prática**

---

## **Fase 4 – Aplicação**

**Equipe: MinhaVidaÉUmDeepFake - Grupo 09**

**Chris Araújo Felipe Souza - 250453**

**Guilherme Azevedo Horn - 247127**

**Guilherme Marconi Mugnai - 241277**

**Joshua Matheus Ramos da Silva - 220670**

**Lucas de Lima Guarnieri - 119756**

2024

## 1. Descrição da Aplicação

A aplicação desenvolvida tem como objetivo realizar consultas no banco de dados desenvolvido nas outras etapas do projeto, proporcionando uma interface eficiente para pessoas da área técnica. A principal função da aplicação é realizar consultas pré-estabelecidas sobre as mídias, como tipo, distribuição geográfica, plataformas associadas, e outros atributos importantes. Essas consultas são feitas de forma simples, com os resultados sendo retornados diretamente no terminal bash, o que facilita a integração em outros ambientes de desenvolvimento e produção..

O banco de dados foi projetado para armazenar informações sobre as mídias, sendo um componente essencial para a aplicação, já que ele centraliza e organiza os dados necessários para gerar as respostas às consultas solicitadas. O uso de um banco de dados permite que a aplicação acesse informações de maneira eficiente, mantendo a integridade e a consistência dos dados através de um gerenciamento estruturado. A relação entre a aplicação e o banco de dados é direta e crucial, pois as consultas são executadas utilizando queries SQL, e os resultados são processados e apresentados ao usuário de forma clara e organizada.

A aplicação é voltada para o fornecimento de dados para outras aplicações e/ou instituições, em um modelo de negócio mais próximo ao b2b, dessa forma não há a preocupação em tornar a interface acessível ao público geral, já que o foco está em facilitar o trabalho daqueles que lidam com grandes volumes de dados, como analistas, desenvolvedores e pesquisadores. O acesso aos dados é restrito e controlado pelo banco de dados PostgreSQL, garantindo que apenas usuários autorizados possam interagir com as informações.

O objetivo geral do banco de dados é fornecer uma base organizada e eficiente para armazenar e recuperar os dados das mídias de forma rápida e consistente. A aplicação auxilia nesse objetivo ao fornecer uma interface simples para a execução de consultas específicas, permitindo que o usuário acesse rapidamente as informações de que precisa sem a necessidade de interagir diretamente com o banco de dados ou com a linguagem SQL.

A interação entre o usuário e o banco de dados se dá por meio das consultas feitas no terminal, onde o usuário apenas insere a opção desejada e recebe os resultados correspondentes. Como a aplicação oferece apenas funcionalidades de leitura, o usuário tem permissão somente para consultar os dados, sem a possibilidade de realizar alterações no banco de dados. Isso garante a integridade das informações e evita alterações não autorizadas.

## 2. Metodologia

A aplicação foi projetada para fornecer um meio simples e eficiente para consultas em uma grande base de dados. O modelo escolhido para a implementação foi uma aplicação de linha de comando (CLI) com interação direta entre o usuário e o banco de dados por meio de consultas SQL. A decisão de optar por uma interface de linha de comando foi motivada pela necessidade de fornecer uma solução rápida e flexível para profissionais técnicos, que frequentemente trabalham com dados e preferem utilizar ferramentas que operam de forma eficiente em ambientes de terminal. Outro ponto de extrema importância que motivou essa escolha foi criar uma aplicação altamente adaptável, para que diferentes instituições e negócios possam rapidamente adaptá-la às suas necessidades, seja

criando novas categorias de consultas, direcionando os dados de saída para um formato particular ou até mesmo usar a aplicação como base para desenvolvimento de soluções próprias

A aplicação oferece uma interface simples e direta, onde o usuário escolhe uma consulta a ser realizada através de um menu numérico. Após a seleção, a aplicação executa a query correspondente no banco de dados PostgreSQL e exibe os resultados no terminal. Esse modelo foi escolhido devido à sua facilidade de uso em ambientes técnicos, além de garantir que a interação com o banco de dados fosse realizada de maneira rápida e eficiente. O formato de saída no terminal permite que os dados sejam visualizados de forma concisa, sem a necessidade de interfaces gráficas mais complexas.

Em relação às interações esperadas, o usuário pode fazer consultas sobre mídias, distribuição geográfica, plataformas, motivos, e interações entre agentes e alvos. Por exemplo, ao selecionar a opção de consultar mídias por tipo, o usuário será solicitado a inserir o tipo de mídia desejado (vídeo, imagem, áudio, etc.), e a aplicação retornará uma lista de resultados com as informações pertinentes. Esse tipo de interação facilita a exploração de dados sem a necessidade de digitar comandos SQL manualmente, tornando a aplicação mais acessível para usuários técnicos que não desejam lidar com a complexidade das queries diretamente. No entanto, o modelo permite que rapidamente possam ser adicionadas novas consultas personalizadas para necessidades específicas.

O desenvolvimento da aplicação seguiu uma abordagem modular, com a definição de funções específicas para cada consulta ao banco de dados. A aplicação foi estruturada de forma a garantir que a lógica de consulta estivesse isolada da interface, permitindo que o código fosse mais fácil de manter e expandir. A utilização de módulos para cada consulta (como `consulta_midia_por_tipo`, `consulta_distribuicao_geografica`, entre outros) foi uma decisão estratégica para manter o código organizado e facilitar futuras modificações.

As ferramentas utilizadas para o desenvolvimento da aplicação foram:

- Python: A linguagem escolhida por sua simplicidade e robustez em manipulação de dados, além de sua excelente integração com bancos de dados.
- PostgreSQL: O banco de dados utilizado devido à sua confiabilidade e capacidade de manipular grandes volumes de dados de maneira eficiente.

A equipe de desenvolvimento se organizou de forma a dividir as tarefas de maneira eficiente: uma parte do grupo ficou responsável pela estruturação do banco de dados e definição das queries, enquanto outra parte cuidou da implementação da lógica da aplicação em Python, conectando a interface com o banco de dados. Em resumo, a aplicação foi desenvolvida com foco na simplicidade e na eficiência, com uma interface de linha de comando que atende às necessidades do público técnico. As decisões de projeto foram orientadas pela busca por flexibilidade, organização do código e eficiência na interação com o banco de dados.

### 3. Uso do Banco de Dados

Os resultados da fase 3 são utilizados pela aplicação de modo a realizar consultas por meio de filtros que selecionam e fornecem resultados relevantes ao usuário, como a distribuição geográfica e as relações entre agentes e alvos. A interação com o banco de dados usando a aplicação se dá por meio das opções oferecidas, que podem ser selecionadas e executadas por números digitados no terminal por parte do usuário. As consultas ao banco de dados na aplicação CLI são designadas por meio da função `mostrar_menu`, que elenca opções para a exibição das mídias com base em filtros específicos, além da `executar_opcao`, que por sua vez, executa a busca selecionada no menu.

### 4. Acesso à Aplicação

#### Estrutura do Projeto

- `app.py`: Arquivo principal da aplicação CLI. Ele exibe o menu interativo no terminal, gerencia a interação com o usuário e chama as funções de consulta de acordo com as opções selecionadas.
- `database config.py`: Arquivo de configuração da conexão com o banco de dados PostgreSQL. Ele contém a função de conectar banco(), que utiliza variáveis de ambiente para estabelecer uma conexão segura com o banco de dados.
- `queries.py`: Este arquivo contém funções para cada uma das consultas SQL usadas na aplicação. Cada função encapsula uma consulta específica e retorna os resultados para que o `app.py` possa exibi-los ao usuário.
- `utils.py`: Contém funções utilitárias para formatar e manipular os dados obtidos do banco de dados. Isso inclui a formatação de resultados em tabelas e o tratamento de dados complexos, como blobs de imagem, que são salvos como arquivos temporários para visualização.

#### Configuração do Ambiente

Para rodar este projeto, é necessário instalar a biblioteca `python-dotenv` para gerenciar variáveis de ambiente. Execute o seguinte comando no terminal para instalá-la:

```
bash
Copy code
pip install python-dotenv
```

Crie um arquivo `.env` na pasta raiz do projeto para configurar as variáveis de conexão ao banco de dados. Insira as seguintes configurações no arquivo `.env`:

```
makefile
Copy code
DB_NAME=nome_do_banco
DB_USER=usuario
DB_PASSWORD=senha
DB_HOST=localhost
DB_PORT=5432
```

## Execução

Certifique-se de que todos os arquivos estejam organizados conforme abaixo:

```
meu_projeto/
├── app.py          # Arquivo principal da aplicação CLI
├── database_config.py # Configuração de conexão ao banco de dados
├── queries.py      # Arquivo com as funções de consulta SQL
├── utils.py        # Funções utilitárias para formatação de resultados
├── .env.example    # Exemplo das variáveis de ambiente
└── requirements.txt # Lista de dependências
```

O arquivo requirements.txt lista as dependências do projeto. Gere-o automaticamente com o comando:

```
bash
Copy code
pip freeze > requirements.txt
```

Ou insira manualmente as dependências principais:

```
Copy code
psycpg2
python-dotenv
pillow
```

Recomenda-se incluir um arquivo README.md com instruções detalhadas para configuração e execução do projeto.

Exemplo de conteúdo do README.md e os documentos em anexo.

## Anexo

### Exemplo de uso:

```
# Projeto de Banco de Dados CLI

Este projeto é uma aplicação de linha de comando que permite executar consultas SQL
em um banco de dados PostgreSQL.

## Pré-requisitos
- Python 3.7+
- PostgreSQL
- Dependências listadas em 'requirements.txt'

## Instalação
1. Clone o projeto ou extraia os arquivos em um diretório local.
2. Instale as dependências:
    ''bash
    pip install -r requirements.txt

Configure as variáveis de ambiente:
Crie um arquivo .env baseado no arquivo .env.example.
Preencha as variáveis com as credenciais do banco de dados.

Estrutura do Projeto
app.py: Arquivo principal da aplicação CLI.
database_config.py: Configuração de conexão ao banco de dados.
queries.py: Funções para consultas SQL.
utils.py: Funções utilitárias para formatação de resultados.
.env.example: Exemplo das variáveis de ambiente.
requirements.txt: Lista de dependências do projeto.

Executando o Projeto
Certifique-se de que o banco de dados PostgreSQL está em execução e acessível.
Execute a aplicação com o comando:
bash
Copiar código
python app.py
```

### Link de acesso a aplicação:

<https://drive.google.com/file/d/1TVV9hK8gUGOG8bfvFWcAAEqNML3i-P-H/view?usp=sharing>