

**CAF VISION SYNC**

**RELATÓRIO DO PROJECTO**

**Autores:**

1. João Conquia
2. Maurício Benvindo
3. Leonel

Luanda, 2024/2025

### **Introdução**

O **CAF VisionSync** é uma plataforma de feedback voltada para a Escola CAF, com o objetivo de coletar e analisar as opiniões dos alunos de forma direta e eficiente. A plataforma é dividida em três módulos principais: cadastro de alunos, envio de feedbacks e análise de sentimentos. Para garantir a segurança dos dados e a integridade do sistema, diversas práticas de segurança foram implementadas, protegendo a plataforma contra ataques comuns, como brute force, XSS, CSRF e SQL Injection.

Neste relatório, vamos apresentar uma visão detalhada das funcionalidades da plataforma, arquitetura do sistema, medidas de segurança implementadas e o funcionamento da análise de sentimentos.

### **Arquitetura do Sistema**

A plataforma CAF VisionSync foi desenvolvida utilizando a arquitetura Flask, com um banco de dados SQLite para armazenar as informações dos alunos e feedbacks. Abaixo, descrevemos as principais partes do código:

#### **Tecnologias Utilizadas**

1. Front-End: HTML, CSS, JavaScript
2. Back-End: Python com Flask
3. Banco de Dados: SQLite, utilizando SQLAlchemy para ORM
4. Análise de Sentimentos: TextBlob
5. Criptografia de Senhas: WerkZeug (módulo generate\_password\_hash e check\_password\_hash)

#### **Classes e Funcionalidades**

Estudante: A classe Estudante armazena as informações de cada aluno, incluindo nome, email e senha criptografada.

Feedback: A classe Feedback armazena o conteúdo do feedback enviado pelo aluno e a classificação do sentimento (Positivo, Negativo, Neutro).

#### **Fluxo do Cadastro**

O processo de cadastro de um aluno é dividido em duas etapas. Na primeira, são coletados nome, email e senha, sendo que a senha é criptografada utilizando o algoritmo generate\_password\_hash. Na segunda etapa, são coletadas informações adicionais, como a classe/turma e sexo do aluno.

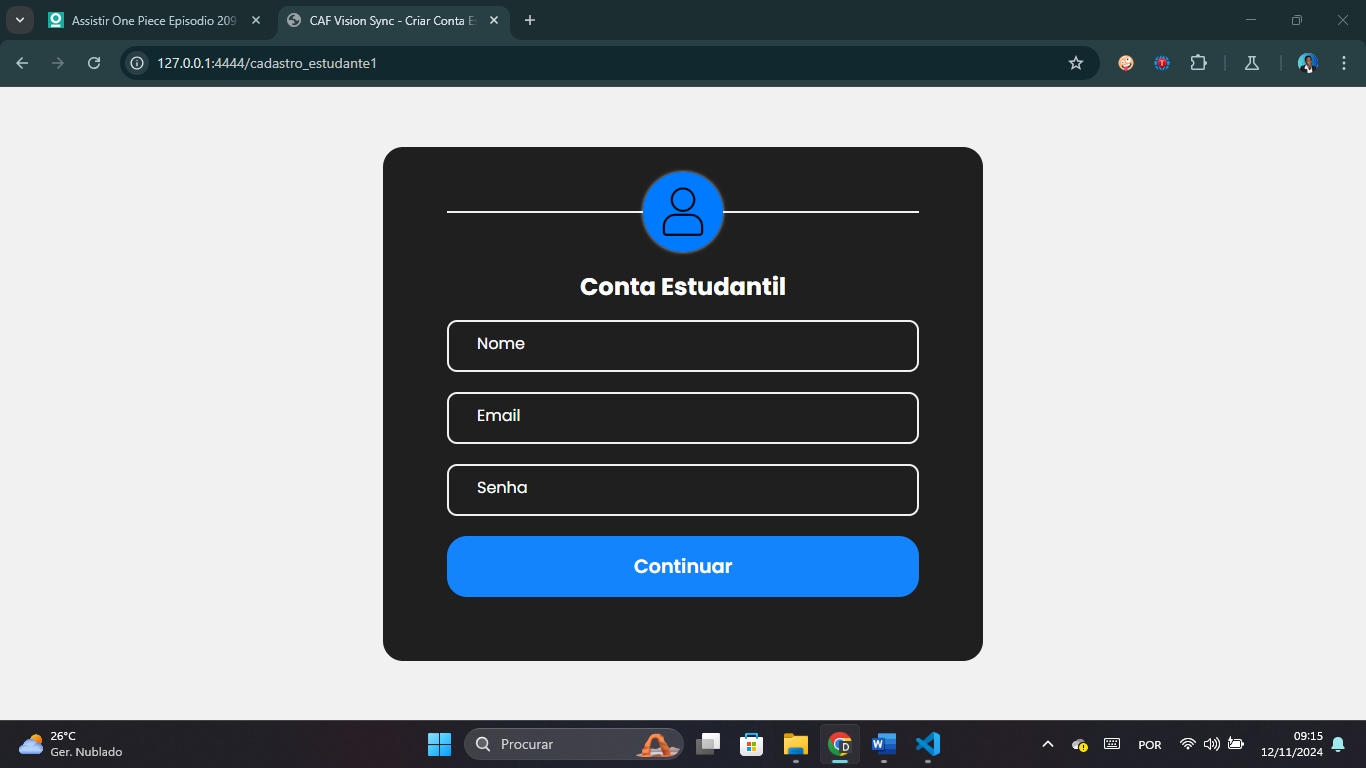
### **Funcionalidades da Plataforma**

#### **Cadastro de Aluno**

O aluno se registra por meio de um formulário que coleta seu nome, email e senha na primeira etapa, e informações adicionais (classe e sexo) na segunda. O código responsável por isso está nas rotas /cadastro\_estudante1 e /cadastro\_estudante2.

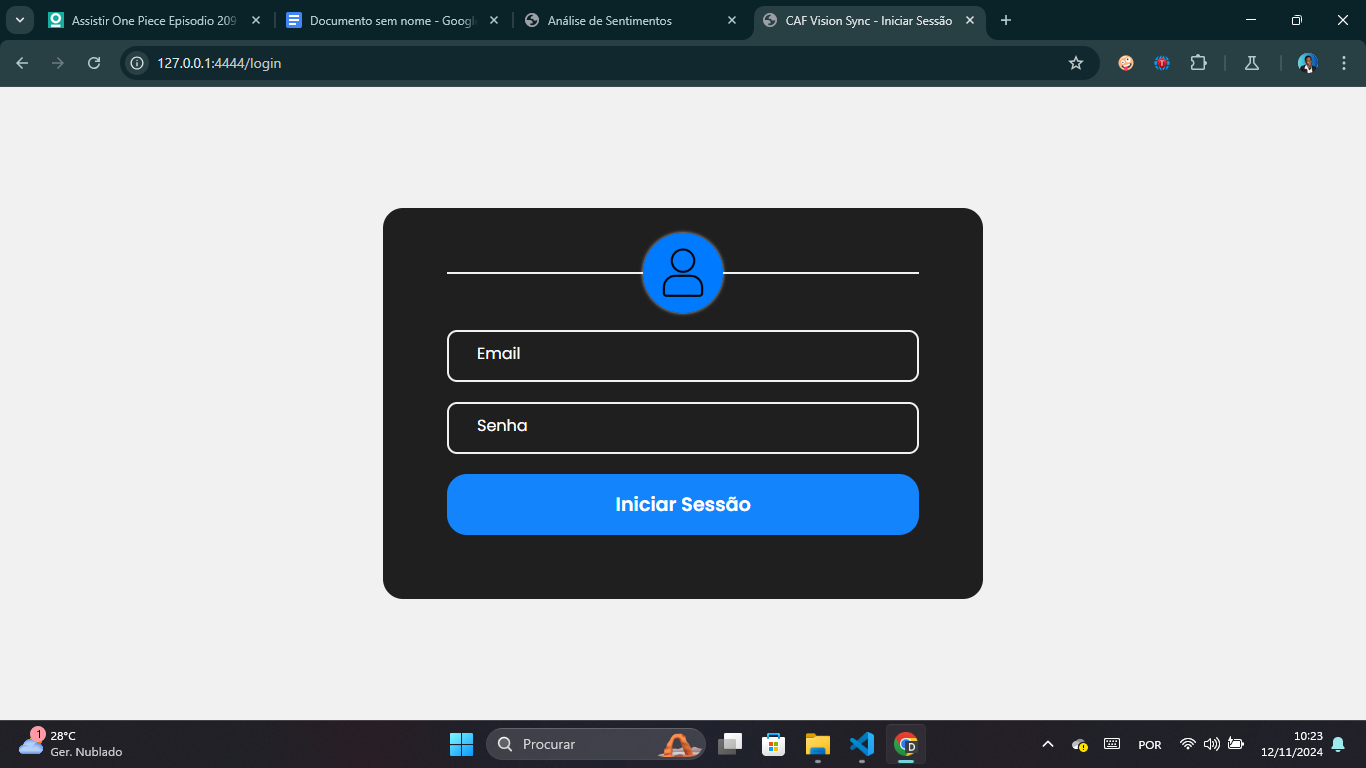
Tela 1 de Cadastro (cadastro\_estudante1): A tela coleta o nome, email e senha do aluno. A senha é criptografada antes de ser salva no banco de dados.

Tela 2 de Cadastro (cadastro\_estudante2): A tela coleta a classe/turma e o sexo do aluno, e ao final redireciona o aluno para a tela de login.



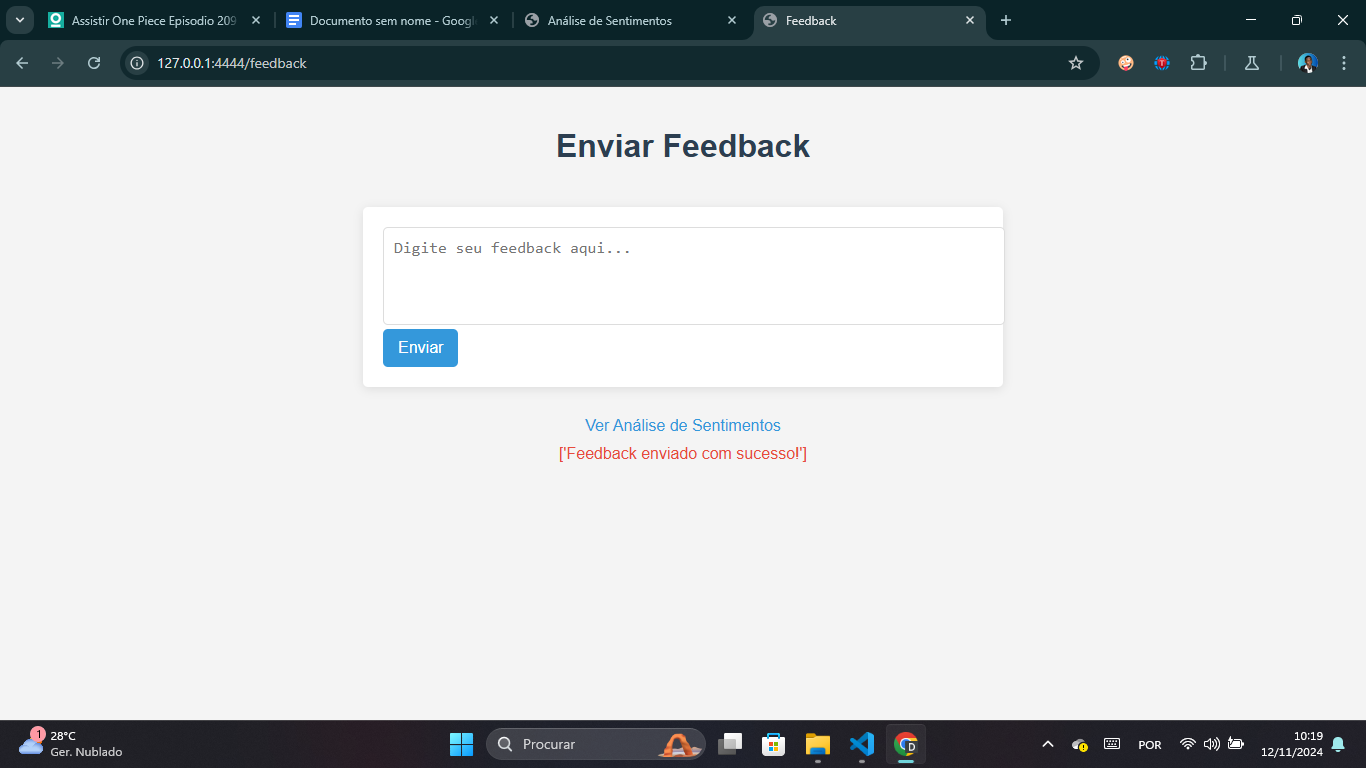
#### **Login**

O aluno se autentica na plataforma através de um formulário de login, onde seu email e senha são validados contra os dados armazenados no DB. Se as credenciais estiverem corretas, o aluno é redirecionado para a página de feedback.



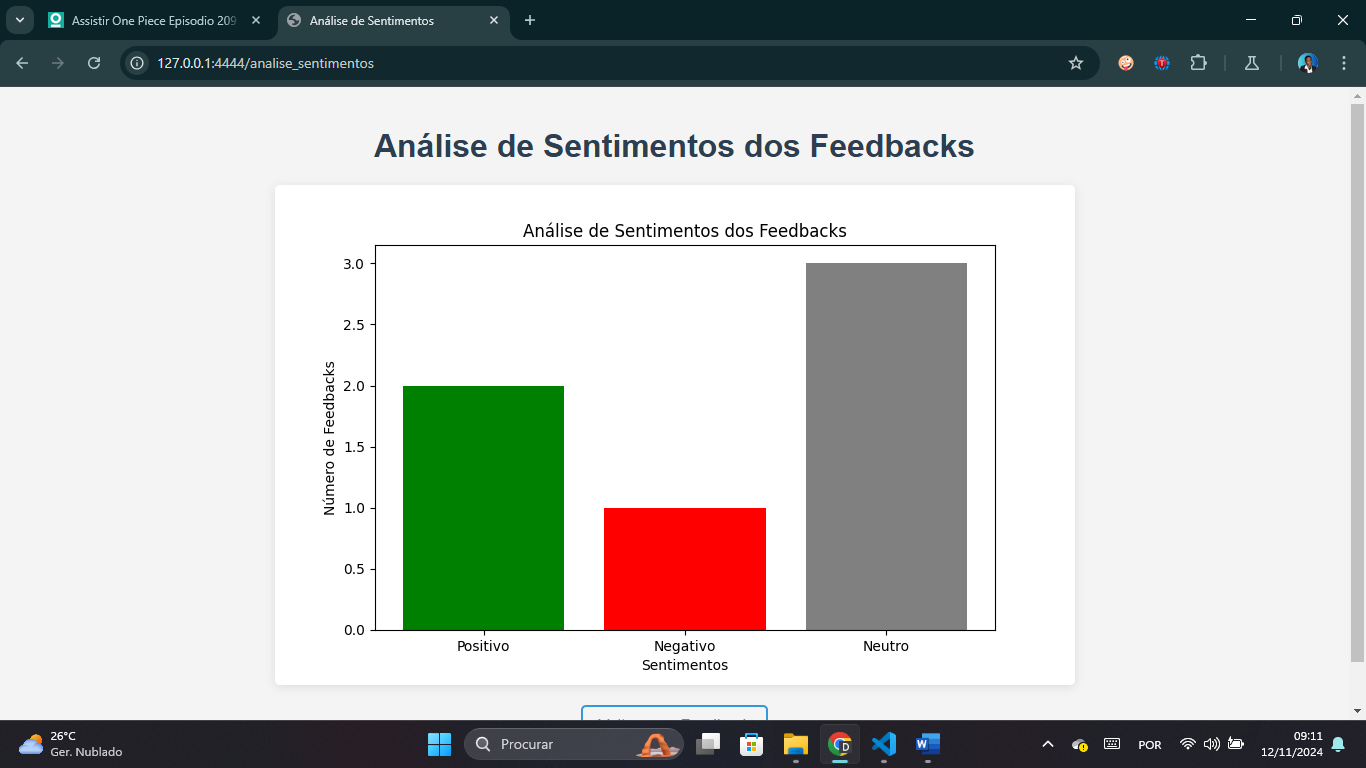
#### **Envio de Feedback**

Na página de feedback, o aluno pode escrever um comentário sobre a escola. O conteúdo do feedback é analisado utilizando a biblioteca TextBlob, que classifica o sentimento do texto como "Positivo", "Negativo" ou "Neutro". O feedback é então armazenado no banco de dados, com a classificação de sentimento correspondente.



#### **Análise de Sentimentos**

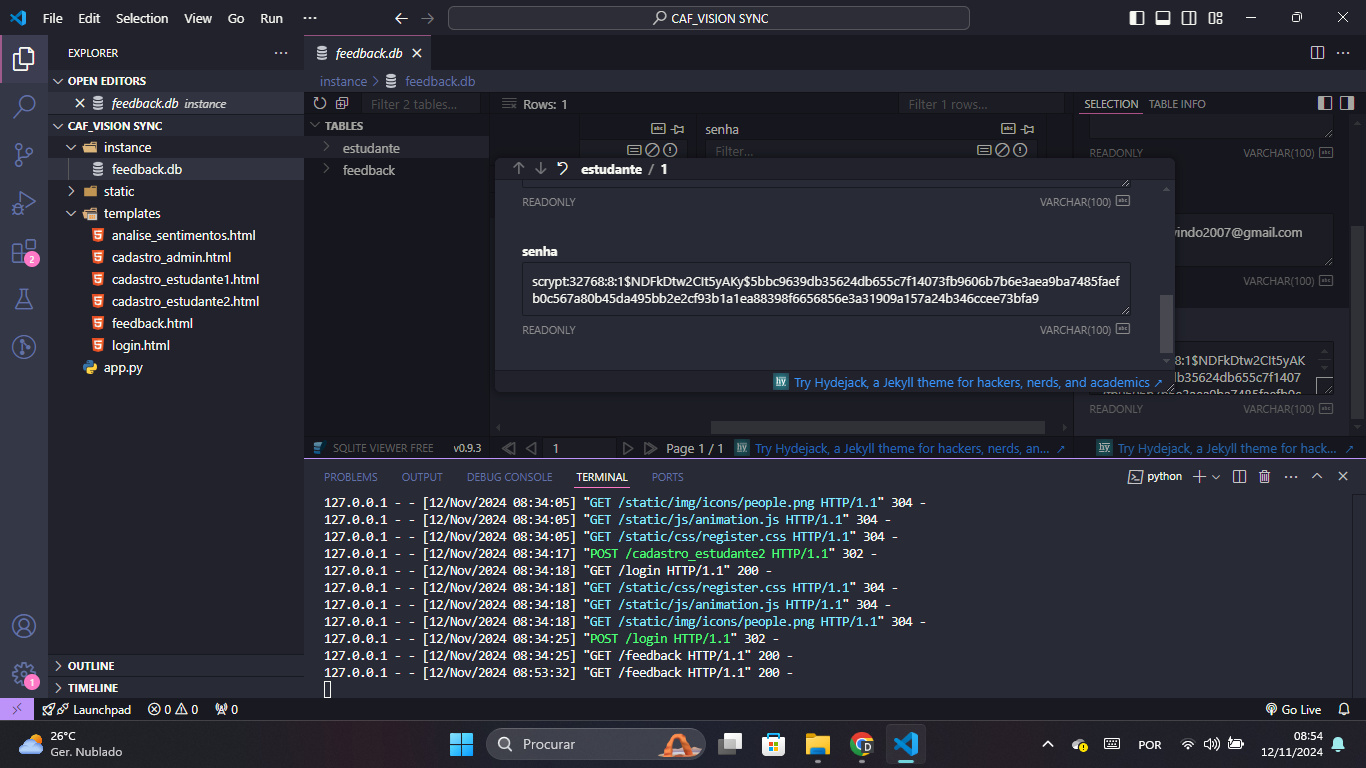
Na página /analise\_sentimentos, a plataforma coleta todos os feedbacks armazenados e gera um gráfico de barras mostrando a distribuição de sentimentos (Positivo, Negativo, Neutro). O gráfico é gerado utilizando a biblioteca Matplotlib e é exibido como uma imagem codificada em base64.



### **Implementação de Segurança**

#### **Criptografia de Senhas**

As senhas dos alunos são armazenadas de forma segura utilizando o algoritmo Scrypt para criptografá-las. Ao realizar o login, a senha fornecida é verificada através do método check\_password\_hash para garantir que a senha seja válida, sem a necessidade de armazená-la em texto puro no banco de dados.



#### **Proteção contra SQL Injection**

Para evitar ataques de SQL Injection, o código utiliza SQLAlchemy, que usa consultas preparadas e parametrizadas. Isso garante que os dados fornecidos pelos usuários sejam tratados de forma segura e não permitam a execução de comandos SQL maliciosos.

### **Conclusão**

O CAF VisionSync é uma plataforma robusta para coleta e análise de feedbacks, com foco na segurança e integridade dos dados. A implementação de criptografia de senhas e a utilização de práticas de segurança, como o uso de SQLAlchemy e análise de sentimentos com TextBlob, garantem que o sistema seja eficiente e seguro.

Apesar de a plataforma já ser funcional, algumas melhorias podem ser feitas, como a implementação de proteção contra Brute Force e CSRF, além de adicionar medidas de proteção mais rigorosas contra XSS.

### **Referências**

Flask Documentation: https://flask.palletsprojects.com/en/stable/

SQLAlchemy Documentation: https://docs.sqlalchemy.org/

TextBlob Documentation:<https://textblob.readthedocs.io/en/dev/>

WerkZeug Security Documentation: <https://werkzeug.palletsprojects.com/en/stable/>

Matplotlib Documentation: https://matplotlib.org/stable/plot\_types/index.html