Guia do Avaliador

Como acessar o projeto e entender sua arquitetura

MathMorphosis
EMPRESA JÚNIOR

2025

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Sumário

Resumo

Link de acesso: https://avaliacaoinstitucionalmath.streamlit.app

Hospedamos o site na plataforma do streamlit. Usamos o recurso de teste gratuito para realizar esse projeto. A plataforma quando vê inatividade do site acaba por retirar ele do ar. Caso isso ocorra contate: (41) 991168835 para podermos reativa-lo.

Tecnologias usadas:

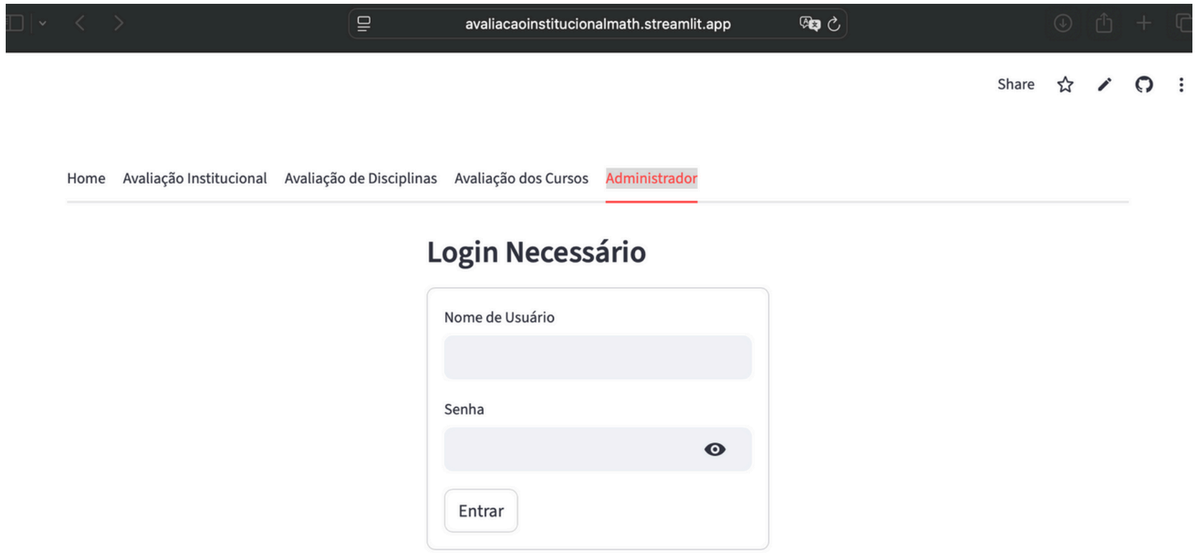
Python +3.10 *e Framework* ***Streamlit*** *para criação das telas do site. Além de bibliotecas comuns dentro da área de analise de dados como* ***pandas*** *e* ***plotly***.

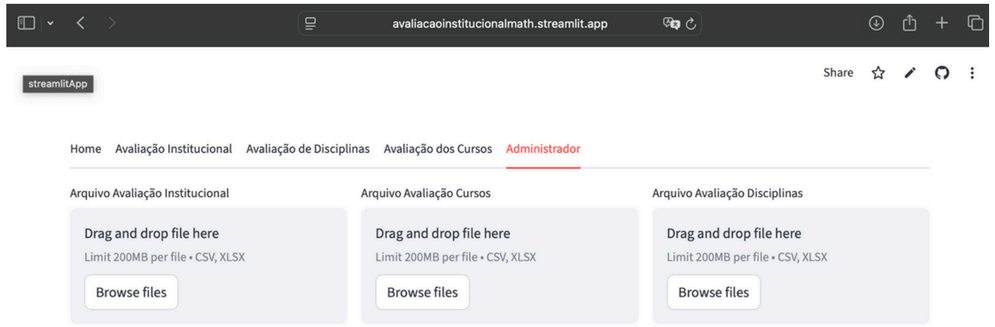
Resumo

versão compatível com celular também!

Adendos

Solução intermediária para atualização dos Dados





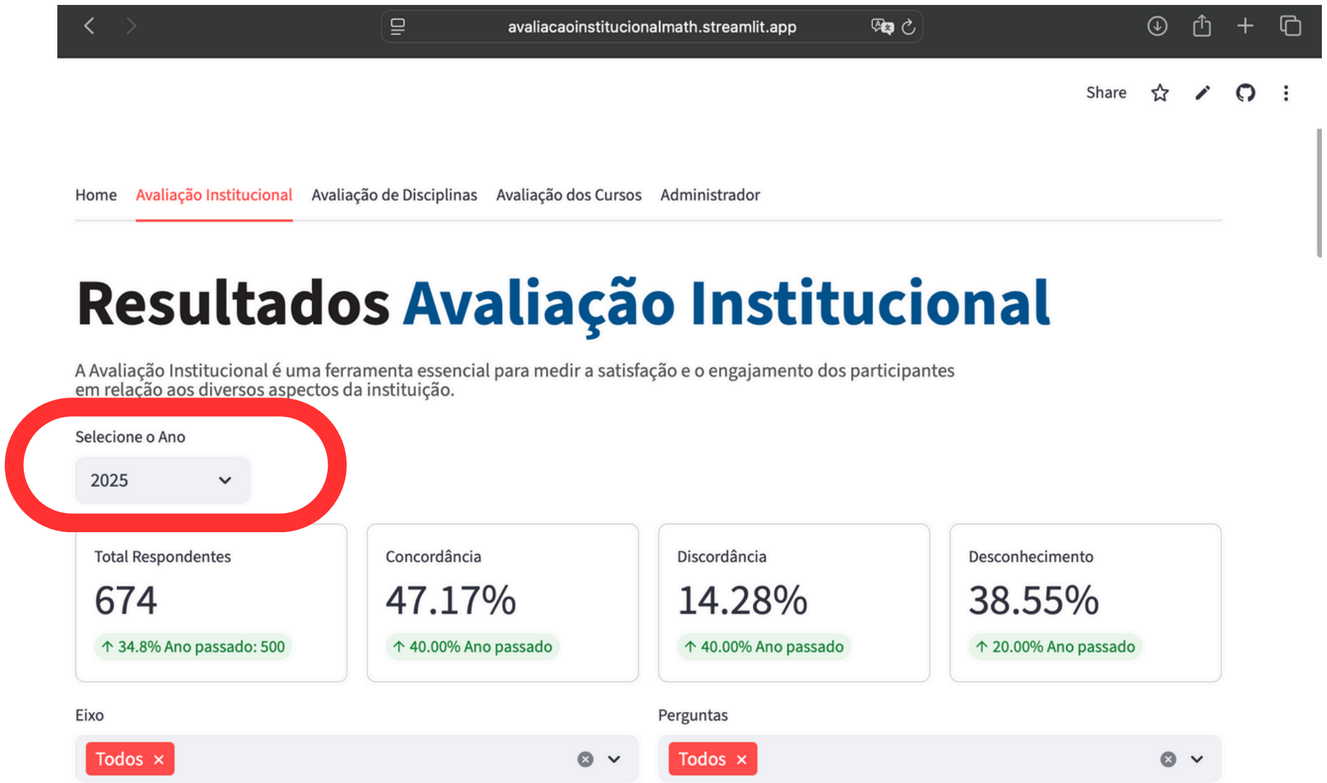
Em Desenvolvimento :3

Nome de usuário admin **Senha** 123

Propusemos uma **solução intermediária para facilitar a atualização dos dados**. Criamos o conceito de uma **página administrativa onde o servidor poderia apenas enviar os arquivos Excel** de cada pesquisa, e o sistema **realizaria automaticamente todo o pré-processamento e atualização da plataforma**. No entanto, não consideramos essa abordagem ideal nem a implementamos totalmente, pois fugia do escopo. Embora simplificasse as atualizações iniciais, o envio manual de arquivos se tornaria inviável com o crescimento da plataforma. Para escalar, o caminho mais adequado seria reprojetar a infraestrutura dos bancos de dados devido ao volume crescente de informações.

Adendos

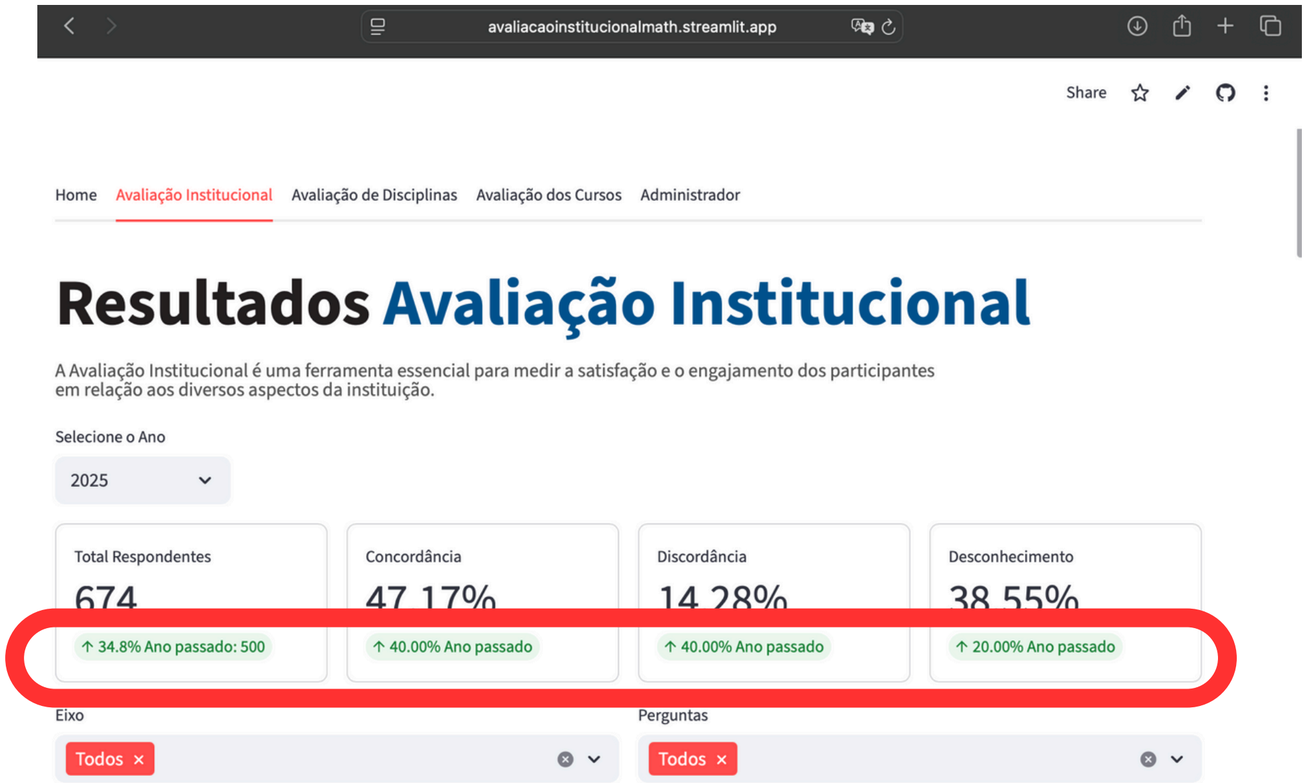
Filtros por período



Apesar da plataforma permitir selecionar o ano, para fins do hackathon não fizemos análises dos dados que não foram encaminhados

Adendos

Comparação com o ano anterior



Apesar da plataforma permitir ver a comparação com o ano anterior, estas são ficticias para fins de visualização do conceito. Todas as outras métricas são usadas dados reais.

Adendos

Legenda
Arquivos

app.py

Caso queira executar o projeto em seu próprio computador após clonar o repositório e baixar as dependências no requirements.txt.

pip install -r requirements.txt

o arquivo **app.py** orquestra a lógica de execução do programa. Então para executar localmente utilize o comando

streamlit run app.py

Fluxograma da arquitetura de projeto

Legenda
pastas principais

data

Responsável por conter todos os dados. Separa em dados **brutos (Raw)** e em **processados (Processed)**

Processed: contém a lógica de preprocessamento e tratamento

*ver Notebooks documentados de pré processamento

Services

Responsável por conter **classes de serviços**.
Classes essas que **recebem os filtros da view** possuem métodos que **correspondem as métricas calculadas.**

Data Loader (classe mãe):
Orquestra a lógica de carregamento dos dados de forma segura

view

Responsável por conter todos as **componentes de visualização do streamlit**

Conclusões

acesso fácil

https://avaliacaoinstitucionalmath.streamlit.app

Analise fácil de acessar através de um site. Podendo ser acessado de qualquer dispositivo.

arquitetura escalável

A arquitetura tem distinção clara entre responsabilidades e é baseada em princípio S.O.L.I.D e em padrões MVC. Além de que os services são documentados através de PEP8 (Boas práticas da comunidade python pra documentação). Isso permite um expecialista implementar novas features em pouco tempo.

soluções intermediárias

Pensamos em soluções que fossem LOW CODE, ou seja, sem a necessidade de expecialista para atualização da plataforma. Ressaltamos a limitação dos dataframes terem que possuir a mesma estrutura da que foi encaminhada e de que armazenar em excell limita a escalabilidade.

Obrigado :)

MathMorphosis
EMPRESA JÚNIOR

2025

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ