



# Entrega Semana 24 - Projeto Edubot - Robótica e Tutoria Pedagógica com LLMs



# Time Dados

# Monorepo - Iterações



- **OBJETIVO**
  - Integrar as novas entregas (rebase) no PR do monorepo (84)
- **RESULTADOS DA TAREFA**
  - Squashei os 60 e poucos commits para evitar repetição de trabalho
    - Dá pra dividir depois se necessário
- **DIFICULDADES**
  - Consertar algumas quebras de fluxo que surgiram de outras entregas
    - Deixar argumentos explícitos ajudou a visualizar os problemas
- **PRÓXIMOS PASSOS**
  - Ajustes finais: doc, renomear serviços, renomear libs, algumas validações, check de consistência de compose em CI, busca de consenso
  - Merge

# Correção do PR #40



- **OBJETIVO**

- A partir das revisões, realizar as mudanças necessárias nos serviços da branch.

- **RESULTADOS DA TAREFA**

- Retirada da pasta referente ao fab.
- Ajustes das variáveis de ambiente
- Criação de um novo endpoint para 'database\_extractor'
- Refatoração da chamada da query para mitigar riscos de SQL Injection, utilizando o padrão repository.

```
{ } Teen - Module 3_student.json  
{ } Teen - Module 4_student.json  
{ } Teen - Module 6_student.json  
{ } Teen - Modulo 3_teacher.json  
{ } Teen - Modulo 8_student.json  
{ } Workbook 1st - Part 2_student.json  
{ } Workbook 4Age - Part 2_student.json  
{ } Workbook 5Age - Part 2_teacher.json
```

# Correção do PR #83



- **OBJETIVO**

- A partir das correções feitas no PR #83, realizar as modificações necessárias no PR

- **RESULTADOS DA TAREFA**

- Ajustes no código e alterações para com a 20250124-monorepo
- Resolução dos conflitos com a 20250124-monorepo
- Correções e melhorias solicitadas pelo Lucas
- Refazer a branch para diminuir as diferenças com a 20250124-monorepo - em andamento



# Time LLM

- **OBJETIVO**

- Implementar um mecanismo para gerar um UUID único para cada arquivo enviado, baseado no hash do conteúdo do arquivo. Essa abordagem tem como objetivo evitar a duplicação de documentos na base de dados e garantir um controle eficiente de versões.
- Migrar persistência de arquivos para o AWS S3

- **RESULTADOS DA TAREFA**

- Implementação concluída: prevenção de duplicação e controle de versões aplicados.
- Persistência de dados migrada com sucesso para o AWS S3.

- **PRÓXIMOS PASSOS**

- Remoção de dados no bucket a partir da api

## Criação do uuid para o arquivo baseado no hash do arquivo



Data Science & Computer Vision  
Industry

F.A.B. Segurança - Gerenciamento - Visualização - admin user -

### Documento já existe

O Documento que você está tentando enviar já existe no sistema. Deseja adicionar uma nova versão?

UUID do Documento:

4c13943117fdf16aea2f9a305359c95014c680ec4f809059e2257b5393871018

Escolha o arquivo:

Escolher arquivo Nenhum arquivo escolhido

Adicionar Nova Versão

F.A.B. Segurança - Gerenciamento - Visualização - admin user -

### Versão já existe

O arquivo que você está tentando enviar já existe como uma versão do documento. Deseja fazer algo diferente?

Tentar novamente [Ver versões existentes](#)



# Realizar revisão de pull request de outro desenvolvedor



- **OBJETIVO**

- Analisar detalhadamente pelo menos um pull request aberto por outro desenvolvedor, verificando a qualidade do código, aderência às boas práticas, correção de eventuais problemas e compatibilidade com o restante do sistema

- **RESULTADOS DA TAREFA**

- Fiz a avaliação e apontei ajustes/melhorias a serem realizados no Pull Request (PR) #84 (<https://github.com/edubot-dev/robotica-pedagogica-com-llm/pull/84>).

- **DIFICULDADES**



- Pull Request complexo por alterar 180 arquivos.

- **PRÓXIMOS PASSOS**

- Analisar detalhadamente a refatoração e como ela padronizou as APIs.

# Realizar revisão de pull request de outro desenvolvedor




 **victorbento** requested changes 1 hour ago [View reviewed changes](#)

**victorbento** left a comment ...


Parabéns pelo árduo trabalho!

Após revisar o PR, identifiquei que as mudanças sugeridas pelo Otávio são pertinentes, especialmente no que diz respeito à padronização e organização do código. Acredito que essas correções vão ser importantes para evoluções futuras no projeto.


Sugeri também algumas mudanças pontuais no "docker-compose.core.yml" para garantir que o projeto seja iniciado corretamente.




`rag_sys/docker-compose.core.yml`

 **victorbento** 1 hour ago ...

- Adicione no .ENV a configuração LANGFUSE\_HOST (está fixo langfuse-server:3000)
- Sugiro renomear LANGFUSE\_INIT\_PROJECT\_PUBLIC\_KEY para LANGFUSE\_PUBLIC\_KEY que é o padrão já utilizado e padronizado no .ENV
- Sugiro renomear LANGFUSE\_INIT\_PROJECT\_SECRET\_KEY para LANGFUSE\_SECRET\_KEY que é o padrão já utilizado e padronizado no .ENV





[Resolve conversation](#)

# Testar todos os componentes



- **OBJETIVO**

- Avaliar os erros e o que precisa melhorar em cada componente da solução para garantir uma aplicação robusta

- **RESULTADOS DA TAREFA**

- Fiz um mapeamento inicial de alguns pontos de instalação
- Identifique alguns erros (talvez sejam resolvido no #84)
- Coloquei no rag\_sys/README.md

- **DIFICULDADES**

- Tudo kkkkk, muitos componentes e demora.

- **PRÓXIMOS PASSOS**

- Continuar testando

# Testar todos os componentes



Data Science & Computer Vision  
for Industry



## Edubot RAG System for chatbot

### Pre requisitos

- Python3.9+
- Docker compose

### Instalação

1. Clone o repositório:

```
git clone git@github.com:edubot-dev/robotica-pedagogica-com-llm.git
```

**NOTA 1:** Recomendamos fazer isso em uma env para não conflito de pacotes

2. Build as imagens dos container utilizados:

```
cd robotica-pedagogica-com-llm/  
ENV=dev docker compose build #para buildar as imagens utilizadas na aplicação
```

3. Configure as variáveis de ambiente Copie o `.env.example` para um arquivo com o nome `.env` e lá será necessa

- RABBITMQ\_DEFAULT\_USER (talvez de para deixar predefinido)
- RABBITMQ\_DEFAULT\_PASS (talvez de para deixar predefinido)
- TELEGRAM\_TOKEN(para usar o bot pelo telegram)
- OPENAI\_API\_KEY para a LLM funcionar, pergunte a alguns do membro do grupo para adquirir
- SENTRY\_DSN\_API\_CLIENTE gerada no dashboard do Sentry, util para monitoramento.

No `.env` existe comentario que podem te auxiliar

**NOTA 2:** Em cada pasta, geralmente há um `requirements.txt` com os pacotes necessario para rodar localmente, por

```
pip install -r requirements.txt
```

### Como usar

Inicie os containers

```
ENV=dev docker compose up
```

### Acesso ao streamlit

Quando os container subirem, entre <http://localhost:81/>. Acesso:

```
login: awsedu  
senha: chatedu
```

Se essas credencias não funcionarem, entre na pasta do [streamlit](#) e veja a senha e usuario predefinido no código.

Ao acessar a interface, e so digitar que o bot irá responder.

### Erros e problemas que eu achei(ordem temporal):

- em `docker-compose.core.yml` no api-extractor esta escrito "restar" e não "restart"
- variavel AMQP\_URI não foi settada no `.env` - isso levanta um warning
- pgadmin esta chamando postgresql, sendo que era postgres
- ENV=dev docker compose up levanta o erro:

```
failed to solve: process "/bin/sh -c pip install --no-cache-dir --upgrade -r /code/requirements.txt" did not complete successfully: exit code: 1
```

- Resolvi com imagem base da api extractor mudando do python3.9-slim para o python3.12.0-slim
- `/api-transcricao/app` no docker file
- `/api-extractor/code` no docker file

robotica\_1

1 2025-07-05 16:22:07 074877488-00 [warning] 20 733 84 Generated features:



Obrigado!