

# Gabriel Roberto Alves Coelho

## Desenvolvedor Full Stack

**Location:** Campo Grande, Mato Grosso do Sul, Brasil

**Contact:** (67) 99177-7363 | [Email](#) | [LinkedIn](#) | [Portfolio](#) | [GitHub](#)

## Profile

Sou desenvolvedor Full-Stack formado em Sistemas para Internet, graduação focada em desenvolvimento WEB. Tenho experiência profissional de mais de 1 ano em desenvolvimento Back-end, focado na criação de API REST em Python (FastAPI) e Javascript (Node). Também tenho experiência breve com desenvolvimento Front-end com React.js e bibliotecas como TailwindCSS, Bootstrap e derivados. Também tenho experiência profissional em automações com Python usando calls de API de LLMs (ChatGPT 4o) e Web Scraping (Selenium).

## Education

- **Graduação:** Tecnólogo em Sistemas para Internet.  
**Instituição:** Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul  
**Período:** 06/2021 – 12/2023.  
**Principais disciplinas:** Estrutura de Dados (Java), Programação Orientada a Objetos (Java), Banco de Dados (Postgres), Engenharia de Software, Construção de páginas WEB (Laravel, Spring Boot), Webservices e XML (SOAP, REST);

## Skills

- **Linguagens de Programação:** Python, Java, JavaScript, Typescript;
- **Frameworks:** Spring Boot, React, Fast API, NextJS, Laravel, Node;
- **Bancos de Dados:** Normalização, Diagramação, SQL, PostgreSQL, MySQL, ORMs, MSSQL;
- **Ferramentas e Tecnologias:** Docker, Git, GitHub, Selenium;
- **Metodologias:** APIs REST, MVC, Repository Pattern, OOP;
- **Outras Competências Técnicas:** HTML/CSS, Testes unitários (Pytest), Web Scraping, microserviços, devcontainers, Ubuntu, WSL;
- **Idiomas:** Inglês avançado;

## Additional Details

- **Java completo: Programação Orientada a Objetos + Projetos.** Udemy;
- **Estrutura de Dados e Algoritmos + Leetcode.** Augusto Galego;

## Work experience

- **Empresa:** [Rural Business](#).

**Cargo:** Analista de Desenvolvimento de Sistemas.

**Período:** 09/2023 – 12/2023 (3 meses).

**Tecnologias principais:** *React.js, Node.js, Express.js, MSSQL, ANTD.*

- **Projeto:** CRUD de commodities

O sistema interno legado de atualizações de *commodities* diárias era pouco intuitivo e feito em HTML puro com frequentes quedas e lentidão.

Finalizei o projeto do novo sistema, desenvolvendo o back-end com uma API REST em Node.js (Express.js) com arquitetura MVC, conforme lógica de negócios da base de dados antiga em MSSQL. Implementei o front-end usando React.js e bibliotecas de componentes.

O sistema novo facilitou o cadastro diário com uma experiência moderna e mais intuitiva, reduzindo o tempo gasto com cadastro em 50% (aproximado) sem quedas.

- **Empresa:** [Ministério Público de Mato Grosso do Sul](#).

**Cargo:** Residente de Tecnologia da Informação.

**Período:** 03/2024 – 03/2025 (1 ano).

**Tecnologias principais:** *Python, Docker, SQLAlchemy, Selenium, FastAPI, Postgres, WSL (Ubuntu).*

- **Projeto:** Web scraper.

A rotina de raspagem de dados de candidatos políticos (Robô da Ficha Limpa) parou de funcionar para algumas cidades, por atualização no funcionamento ou indisponibilidade de certos portais da transparência.

Atualizei o Web Scraping de portais da transparência usando requisições personalizadas (Python Requests) ou simulando acesso (Selenium), conforme caso do portal de cada cidade do estado. Por fim, tratando os dados de csv/planilha para JSON.

Como resultado, atingimos o funcionamento da rotina para acima de 95% das cidades do estado.

- **Projeto:** OCR de PDFs.

O registro manual de informações (saldo e demais dados) de PDFs de extratos bancários pelo setor de prestação de contas era trabalhoso e passível de erro manual.

Participei na implementação do projeto de OCR usando LLM (ChatGPT 4o) e AzureOCR para extração das informações bancárias. Com uma interface de uso do usuário final em Vaadin (Java) com back-end em Python (FastAPI) e SQLAlchemy (Postgres), com o fim de testar e personalizar os prompts usados para o OCR de cada tipo de extrato. Também, foram criados mais de 300 casos de teste para comparar o resultado esperado com o resultado retornado pela LLM, usando Pytest.

Como resultado, a rotina mostrou ter precisão acima de 90% na extração dos dados bancários, conforme testes unitários. Agilizando significativamente o processo de prestação de contas.