



Projeto **AFIAG-DB**

Avaliação de Fluência com Inteligência Artificial Generativa em Big
Data (FAGAI-BD - Fluency Assessment with Generative Artificial
Intelligence on Big Data)

Instituto Tecnológico de Aeronáutica - ITA

**Programa de Pós-Graduação em
Engenharia Eletrônica e Computação -
Informática**

CE-229 - Teste de Software

Prof. Luiz Alberto Vieira Dias

Prof. Adilson Marques da Cunha

Prof. Lineu F. Stege Mialaret

Relatório Final Individual

Strauss

DEV do Time Scrum #2

1 - Introdução

1.1 - Motivação

A participação na disciplina CE-229 e do projeto acadêmico AFIAG-BD/2025 foi motivada pelo interesse do aluno em testes de software e também em tecnologias relacionadas a engenharia de software, inteligência artificial generativa e big data.

Este aluno espera aplicar os conhecimentos no ambiente de trabalho, bem como em sua pesquisa, em nível de doutorado.

1.2 - Contexto

O Projeto acadêmico, denominado AFIAG-BD Avaliação de Fluência com Inteligência Artificial Generativa em Big Data)

representa uma iniciativa para se "dotar o ensino fundamental brasileiro de uma sistemática apropriada, visando melhor avaliar fluência em leitura de palavras, pseudopalavras e textos, bem como melhor avaliar, diagnosticar e propor correções para aumentar os níveis de alfabetização atuais no país".

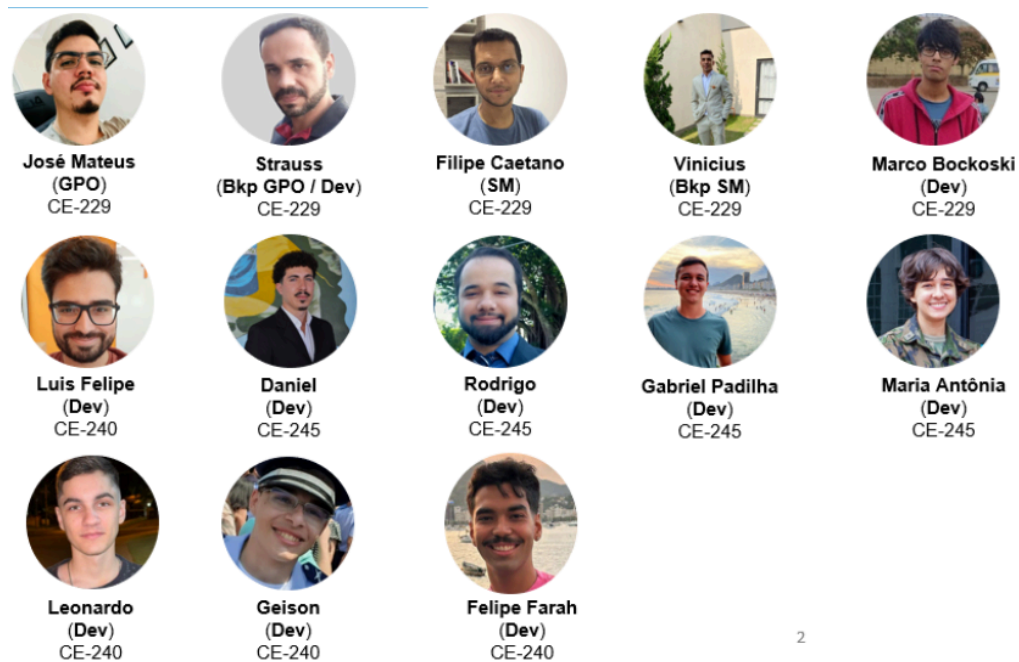
Este projeto deverá ser desenvolvido, a partir das experiências obtidas nos anos anteriores, envolvendo soluções tecnológicas emergentes e inovadoras, para atender às necessidades de detecção e avaliação de níveis de fluência de alfabetização em alunos do 2º ano do ensino fundamental brasileiro, no município de São José dos Campos - SP.

O autor deste relatório exerceu os papéis de DEV, Tester e Backup do General PO no Time Scrum #2.

1.3 - Objetivo do Time Scrum

O Time Scrum #2 ficou responsável pelas necessidades inerentes às escolas e infraestruturas. Assim, focando na experiência do backoffice e na exploração de novas tecnologias.

A Figura F1 apresenta os integrantes do Time Scrum #2.



2

Figura 01: Time Scrum #2

1.4 - Redução do Escopo

O Time Scrum #2 resolveu reduzir o escopo do desenvolvimento por meio da reusabilidade parcial do código desenvolvido pela turma anterior. Também, com relação aos dados, foram reutilizados os data set de alunos, professores e escolas.

Quanto à abrangência da aplicabilidade do projeto, decidiu-se utilizar dados de escolas do município de São José dos Campos, focando os experimentos num contexto municipal.

1.5 - Especificação de Requisitos

Os principais requisitos endereçados pelo Time Scrum #2, na qual houve participação deste aluno, foram:

- Realização da modelagem conceitual das entidades relacionadas à operacionalização do projeto, sendo: Alunos, Professores e Escolas;
- Elaboração de scripts de criação das tabelas;
- Implantação e configuração do Banco de Dados e disponibilização para os outros times;
- Geração das respectivas massa de dados, sendo 1.000.000 de alunos, 13.000 professores e aproximadamente 250.000 escolas de nível fundamental.
- Implementação e publicação em nuvem gratuita (on-render) de uma API com 3 endpoints (Alunos, Professores e Escolas).
- Elaboração de Plano de testes e execução de casos de testes.
- Implementação de um modelo de IA para avaliação de fluência a partir de áudios de alunos de escolas do ensino fundamental.
- Documentação do desenvolvimento para Meta Quest 3 em ambiente Unity e StellarX.

1.6 - Ordem de Apresentação

Na seção Desenvolvimento, encontra-se o detalhamento das atividades realizadas pelo aluno durante o desenvolvimento do projeto AFIAG-DB. Na seção Conclusão, apresenta-se as conclusões, recomendações e sugestões para trabalhos futuros.

2 - Desenvolvimento

2.1 - Visão do projeto AFIAG-BD

O Time Scrum #2 desenvolveu, testou e documentou funcionalidades envolvendo a implementação de um software para realidade virtual simulando uma avaliação da fluência de um aluno e embarcando-o no Meta Quest 3.

2.1 - User Stories

O projeto AFIAG-FB foi desenvolvido em 3 Sprints, nas quais o time scrum #2 ficou responsável pelas funcionalidades que envolvem escolas e infraestrutura. Segue abaixo a lista das histórias desenvolvidas.

Sprint #1

- **Como usuário do backoffice**, quero realizar o

login no sistema para acessar a interface administrativa onde poderá cadastrar escolas, usuários e gerenciar o desempenho das turmas.

- **Como diretor de escola,** quero cadastrar, visualizar, editar e excluir professores na escola, para garantir que a equipe docente esteja organizada e os dados atualizados.
- **Como administrador da Secretaria de Educação,** quero cadastrar, visualizar, editar e excluir escolas no sistema, para garantir que todas as instituições estejam devidamente registradas e possam acessar a plataforma.

Sprint #2

- **Como um membro da Equipe Técnica (TS#02),** eu quero explorar e testar as principais funcionalidades do Meta Quest 3, para que eu possa me familiarizar com o hardware e entender suas capacidades para o desenvolvimento em VR.

- **Como o Time Scrum #2**, eu quero criar um Vídeo-demo mostrando o progresso na integração das Personas P3 e P4 em um Ambiente Virtual na Nuvem, para que possamos demonstrar o estado atual da implementação desta funcionalidade ao professor, mesmo que ainda não esteja completa.
- **Como um Aluno integrante de um Time Scrum (TS)**, eu quero criar um Relatório Individual detalhando minha participação no desenvolvimento das US sob responsabilidade do meu TS e nas entregas de valor da 2ª Sprint (AFIAG-BD), para que o professor possa avaliar minha contribuição individual e meu entendimento das atividades realizadas.

Sprint #3

- **Como um membro da equipe técnica**, eu quero compilar e organizar todos os artefatos de código e documentação técnica gerados durante a sprint, para que possamos ter um registro completo e acessível do trabalho técnico realizado.

- **Como membro da equipe de QA ou desenvolvedor(a)**, eu quero elaborar pelo menos 3 casos de teste para cada User Story Funcional do projeto AFIAG-BD da Sprint 3, para garantir a qualidade, segurança e confiabilidade do sistema.
- **Como um Gestor Municipal (P3) ou Estadual (P4)**, eu quero me conectar a um ambiente de escritório virtual e interagir com o outro gestor, para que possamos ter um espaço dedicado para discussão e análise de dados de fluência.
- **Como um analisador de fluência**, desejo investigar e propor quais metadados dos arquivos de áudio (avaliação), atualmente salvos no sistema de arquivos, são relevantes e devem ser armazenados no banco de dados. Essa mudança permitirá uma maior rastreabilidade e integridade dos dados.

2.4 - Arquitetura da solução

No desenvolvimento deste projeto, foi utilizada uma stack de tecnologias, a fim de propiciar a execução de todo

pipeline necessário à análise de fluência. A Figura 02 apresenta a arquitetura de sistema empregada no projeto AFIAG-BD. Nela, observam-se quatro camadas arquiteturais, sendo: a) camada de interface; b) camada de negócios; c) camada de persistência (dados); e d) camada de inteligência artificial.

A camada de interface (front-end) — IO-Bounding — tem a função de propiciar uma interação homem-máquina com as interfaces desenvolvidas, sendo consulta a dados, telas de CRUD, captação de áudio e dashboard para visualizações analíticas dos dados.

A camada de negócios (back-end) — CPU-Bounding — possui a engine de processamento do projeto. Nela, abarcam-se o Web Server da API AFIAG-BD, o módulo de Blockchain e a API de Análise de fluência.

Na camada de persistência, armazenam-se os dados relacionais, os dados não-estruturados, por meio de arquivos de áudio e os dados de Blockchain.

Por fim, a camada de inteligência artificial é responsável pela operacionalização de LLMs utilizados no projeto.

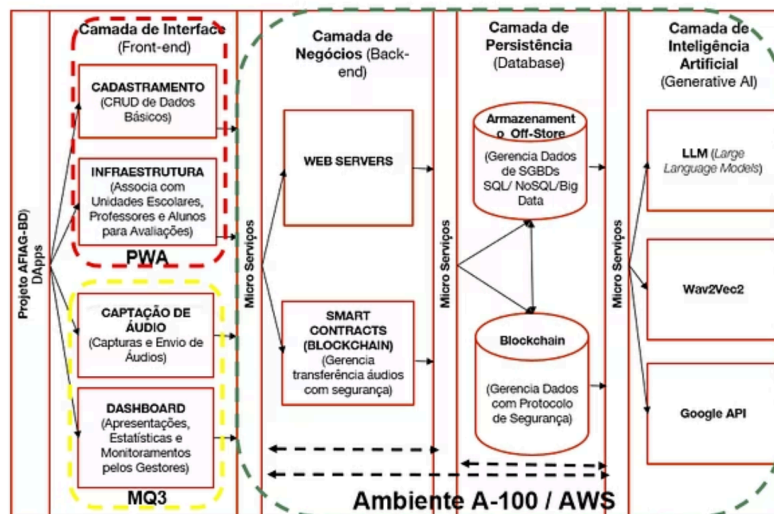


Figura 02: Arquitetura de Sistema

A Figura 03 apresenta a arquitetura do módulo de blockchain. Neste módulo do sistema, o time especialista optou pelo uso da Hathor, uma rede escalável, com alto throughput e baixo custo transacional, propiciando uma interface de comunicação entre o software desenvolvido em Unity.

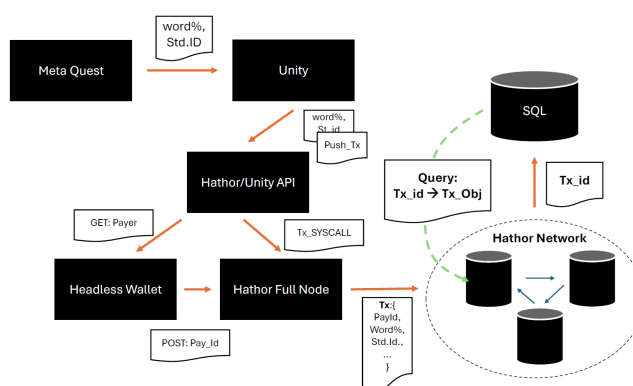


Figura 03: Arquitetura do módulo de Blockchain

2.5 - Casos de teste

Na Sprint #1, o Time Scrum #2 desenvolveu um conjunto de casos de testes, elaborados em planilhas e executados manualmente ao final da sprint. Segue abaixo um screenshot de um destes casos de testes.

Na Sprint #2, o Time Scrum #2 desenvolveu um conjunto de casos de testes. Foi então desenvolvido um mini-framework Python para execução automática das planilhas de caso de testes, visando automatizar os testes de regressão, conforme figuras abaixo. Após a automatização, a execução dos casos de testes caiu de 30 min. (manual) para menos de 1 minuto.

Na Sprint #3 o Time Scrum #2 desenvolveu um conjunto de casos de testes, resultando ao todo 64 casos de teste automatizados.