



UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE
ENSINO
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS ONLINE

***RPG Tático em Texto com Métricas de BI:
Exercitando a Inovação em Jogos Digitais***

EAD.101_1-16.A

Francisco Eduardo Barros Tangerino, 25002174

SÃO JOÃO DA BOA VISTA, SP
Período: Outubro/2025 – Dezembro/2025

2. Introdução

O mercado de jogos digitais é um dos segmentos que mais cresce na indústria do entretenimento, impulsionado tanto pela popularização de plataformas de desenvolvimento quanto pela disponibilidade de ferramentas de análise de dados. Pequenos estúdios e desenvolvedores independentes, entretanto, frequentemente carecem de soluções simples e acessíveis para monitorar o comportamento dos jogadores e utilizar essas informações como insumo para decisões de melhoria contínua.

Neste contexto, o Projeto Integrado propõe o desenvolvimento de um protótipo de RPG tático em texto, executado em terminal e inteiramente escrito em Python, que combina exploração em grade, combate em turnos e coleta estruturada de métricas para Business Intelligence (BI). O jogo registra dados de sessão tanto em arquivo local (JSON) quanto em um banco de dados PostgreSQL, permitindo que essas informações sejam futuras fontes para dashboards analíticos em ferramentas como o Power BI.

O projeto se conecta ao tema da disciplina Formação para a Vida, “EXERCITANDO A INOVAÇÃO”, por meio do Tópico 3 – “Testei minha ideia inovadora... e agora?”. O RPG funciona como um Mínimo Produto Viável (MVP) de um jogo orientado a dados, permitindo testar uma ideia inovadora de coleta de métricas em jogos e refletir sobre seus próximos passos: modelagem de negócios, escalabilidade, experiência do usuário e potencial de comercialização pela empresa própria do aluno.

3. Objetivos

3.1 Objetivo Geral

Desenvolver um protótipo de RPG tático em texto, integrado a um módulo de coleta e persistência de métricas de jogo em JSON e PostgreSQL, de forma a exercitar a inovação na criação de jogos digitais orientados a dados e preparar o sistema para futura evolução como produto comercializável pela empresa própria do aluno.

3.2 Objetivos Específicos

- Projetar a mecânica básica de um RPG tático em grade 5x5, com exploração, eventos de mapa e combate em turnos.
- Implementar classes jogáveis, inimigos e eventos interativos utilizando Programação Orientada a Objetos em Python.
- Definir e registrar métricas de jogo relevantes (passos, combates, uso de habilidades, mortes, tempo de sessão, etc.) em um dicionário estruturado.
- Persistir os dados analíticos em arquivo JSON local e em um banco de dados PostgreSQL, utilizando boas práticas de modelagem e acesso a dados.
- Favorecer o uso futuro de ferramentas de BI, como Power BI, por meio de uma base de dados organizada em esquema relacional.
- Explorar o projeto como oportunidade de inovação pessoal e profissional, avaliando seu potencial como produto comercializável da empresa própria.
- Aplicar versionamento de código (Git/GitHub) e práticas de desenvolvimento incremental ao longo do trimestre letivo.

4. Fundamentação Teórica

A fundamentação teórica deste projeto integra conceitos de inovação, desenvolvimento de software, jogos digitais, programação em Python, bancos de dados relacionais e Business Intelligence.

Do ponto de vista da inovação, o projeto se alinha à ideia de experimentação contínua e desenvolvimento de protótipos, na qual soluções são testadas em pequena escala, avaliadas a partir de evidências e, então, refinadas. O tema “Testei minha ideia inovadora... e agora?” reforça a necessidade de transformar um experimento inicial em oportunidades concretas de melhoria e de geração de valor, seja educacional, seja comercial.

Em desenvolvimento de software, destaca-se o uso de Programação Orientada a Objetos (POO) em Python, que permite modelar entidades do jogo (jogador, inimigos, habilidades, mapa) como classes e objetos, encapsulando atributos e comportamentos. Esse paradigma facilita a manutenção, a evolução do código e a reutilização de componentes.

A área de jogos digitais contribui com conceitos de mecânicas de jogo (game mechanics), como combate por turnos, progressão de nível, exploração de mapa e sistemas de recompensa. Tais mecânicas são essenciais para manter o engajamento do jogador e, ao mesmo tempo, gerar eventos que alimentam o conjunto de métricas armazenadas.

Já Business Intelligence (BI) envolve a coleta, organização e análise de dados para apoiar a tomada de decisão. No projeto, as métricas de sessão são registradas em JSON e em um banco PostgreSQL estruturado com tabelas de jogadores, classes, sessões e tabelas de métricas gerais, de combate e de eventos. Essa modelagem facilita a construção de indicadores de desempenho (KPIs), como tempo médio de sessão, número de inimigos derrotados por classe ou taxa de acerto de charadas.

Por fim, o uso de controle de versões com Git e repositório no GitHub contribui com práticas de engenharia de software profissional, permitindo rastreabilidade, histórico de mudanças e colaboração futura.

5. Metodologia

A metodologia adotada foi de natureza aplicada, com abordagem tecnológica e caráter exploratório. O projeto foi desenvolvido individualmente, ao longo do 4º trimestre letivo de 2025, seguindo etapas incrementais de análise, concepção, implementação e testes.

As principais etapas foram:

1. Levantamento de requisitos do jogo e definição das métricas de BI a serem coletadas em cada sessão.
2. Desenho da mecânica principal de exploração de mapa 5x5, com movimentação do jogador, posicionamento aleatório de inimigos e eventos.

3. Modelagem lógica do banco de dados PostgreSQL, definindo tabelas para classe, jogador, sessão e métricas associadas (gerais, de combate e de eventos).
4. Implementação do loop principal do jogo em Python, com criação das classes de domínio (Jogador, Mapa, EnumInimigos, EnumClasses, EnumEventos).
5. Desenvolvimento das funções de combate, movimentação, resolução de charadas, abertura de baús e interação com fontes de cura.
6. Implementação dos mecanismos de coleta de métricas em um dicionário de estatísticas (stats) e gravação em arquivo JSON.
7. Implementação da função de persistência em banco de dados (salvar_progresso_db), utilizando a biblioteca psycopg2 para comunicação com o PostgreSQL.
8. Testes manuais com diferentes classes de personagem, caminhos no mapa e campanhas, verificando a integridade das métricas e o correto salvamento em JSON e no banco.
9. Uso de Git e GitHub para controle de versões, acompanhamento do progresso e organização profissional do código.

Foram utilizadas as seguintes ferramentas e tecnologias:

- Linguagem Python 3.12;
- Biblioteca psycopg2 para acesso ao PostgreSQL;
- Banco de dados PostgreSQL com schema jogo_pi;
- Formato JSON para registro local de pontuações;
- Ambiente de desenvolvimento VS Code;
- Sistema de controle de versões Git, com repositório hospedado no GitHub.

6. Desenvolvimento

O desenvolvimento do projeto iniciou-se com a definição dos parâmetros globais do jogo, como o tamanho do tabuleiro (5x5) e as credenciais de conexão com o banco de dados PostgreSQL. Em seguida, foi construída a estrutura de entrada do jogador, incluindo a função de solicitação de nome e a escolha de classe, baseada em um enumerador (EnumClasses) que carrega

atributos como HP, MP, força, destreza, inteligência, defesa, sorte e ícone visual do personagem.

A classe Jogador foi responsável por representar o personagem principal, incluindo métodos de movimentação no mapa, uso de habilidades, cálculo de ataque básico e ganho de experiência. O mapa foi modelado pela classe Mapa, que gerencia a matriz de posições, a inserção e remoção de inimigos e a exibição visual do tabuleiro no terminal.

A geração procedimental de fases foi implementada por meio das funções `gerar_inimigos_fase` e `gerar_eventos_fase`. Essas funções distribuem inimigos e eventos (fontes de cura, charadas e baús de tesouro) aleatoriamente pelo mapa, respeitando restrições como não ocupar a posição inicial do jogador. Duas campanhas principais foram definidas ("Trilha do Reino" e "Rota das Cinzas"), cada uma composta por três fases com combinação própria de inimigos, eventos e chefes finais.

As mecânicas de combate foram implementadas na função `executar_batalha`, que controla a troca de ataques entre jogador e inimigo em turnos. Foram considerados elementos como chance de acerto baseada em destreza, chance de crítico baseada em sorte, falhas críticas que causam dano a quem ataca e habilidades especiais com custo de MP. Durante o combate, diversas métricas são atualizadas, como inimigos derrotados, fugas, críticos acertados e sofridos, desvios e falhas críticas.

A função `jogar_fase` orquestra a movimentação pelo mapa, a detecção de encontros com inimigos e a ativação de eventos. Ao final de cada fase, pontos são acumulados de acordo com vitórias em combate, uso de eventos e conclusão da fase. Todo o progresso é refletido tanto no dicionário de estatísticas quanto nas funções de persistência.

Para registro local, a função `registrar_pontuacao` grava um snapshot completo da sessão em arquivo JSON (`scores.json`), incluindo informações de horário, nome, classe, pontuação e um bloco de métricas analíticas organizadas em categorias (gerais, combate e eventos). Para persistência em

banco de dados, a função `salvar_progresso_db` realiza operações de inserção e atualização (upsert) nas tabelas de classe, jogador, sessão e métricas associadas, garantindo que cada sessão possua um identificador único (`sessao_id`) e possa ser consultada posteriormente em ferramentas de BI.

7. Resultados Obtidos

Ao término do período de desenvolvimento, os principais resultados alcançados com o projeto foram:

- Implementação de um RPG tático em texto plenamente jogável em terminal, com fases, campanhas, inimigos variados e eventos de mapa.
- Construção de um sistema de coleta de métricas de sessão, capaz de registrar tempo de jogo, passos, descansos, combates, uso de habilidades, magias e interações com eventos.
- Persistência bem-sucedida dos dados em arquivo JSON, facilitando a consulta rápida ao histórico de sessões.
- Integração com banco de dados PostgreSQL, utilizando um schema específico (`jogo_pi`) com tabelas normalizadas para jogadores, classes, sessões e métricas.
- Criação de uma base de dados adequada para futura construção de dashboards em ferramentas de BI, como o Power BI, permitindo análises de desempenho por classe, fase ou campanha.
- Prototipagem de um possível produto comercializável, em que a empresa própria do aluno poderá evoluir o jogo para uma solução de analytics em games voltada a pequenos estúdios.

8. Dificuldades e Aprendizados

Durante o desenvolvimento, algumas dificuldades se destacaram. A primeira foi a modelagem adequada das métricas no banco de dados PostgreSQL, especialmente para garantir que as tabelas de métricas gerais, de combate e de eventos estivessem corretamente ligadas à tabela de sessões. Também houve desafios na configuração da conexão com o banco em ambiente local e na definição de tratamentos de erro para situações de falha na persistência.

Outra dificuldade relevante foi o balanceamento da mecânica de combate, envolvendo probabilidades de acerto, dano crítico e falha crítica, de forma a

tornar as batalhas desafiadoras, porém justas. A implementação de habilidades especiais, com consumo de MP e diferentes tipos de dano, exigiu testes repetidos e ajustes finos.

Apesar dessas dificuldades, o projeto proporcionou diversos aprendizados. O aluno pôde consolidar conhecimentos de Programação Orientada a Objetos em Python, integração com bancos de dados relacionais, manipulação de arquivos JSON e boas práticas de versionamento com Git. Além disso, reforçou a importância de pensar em métricas desde o início do projeto, reconhecendo que a coleta estruturada de dados é fundamental para decisões futuras de melhoria e de inovação.

Também foi possível reutilizar e expandir aprendizados de projetos anteriores, especialmente no que diz respeito à organização modular do código, ao planejamento de funcionalidades e ao alinhamento com objetivos de formação pessoal e profissional.

9. Considerações Finais

O desenvolvimento do RPG tático em texto com métricas de BI permitiu integrar, na prática, conceitos de programação, bancos de dados, Business Intelligence e inovação. Ao transformar o jogo em um laboratório de dados, o projeto aproxima o universo do entretenimento digital das práticas de análise utilizadas em ambientes corporativos, mostrando que é possível extrair informações relevantes mesmo de aplicações simples, como um jogo em modo texto.

Sob a perspectiva da Formação para a Vida, o projeto exemplifica o Tópico 3 – “Testei minha ideia inovadora... e agora?”, ao demonstrar que testar uma ideia é apenas o primeiro passo. A partir dos resultados obtidos, abre-se espaço para novas perguntas: como melhorar a experiência do jogador, quais métricas são mais relevantes, que modelo de negócios pode sustentar o produto e como escalar a solução para outros contextos.

Como próximos passos, vislumbra-se a criação de dashboards em Power BI para visualização das métricas coletadas, a migração do jogo para uma

interface gráfica ou web e a validação do protótipo com usuários reais. Essas ações podem consolidar o projeto como um produto comercializável pela empresa própria do aluno, ao mesmo tempo em que fortalecem sua trajetória de inovação pessoal e profissional.