

## BUG HUNTERS

Integrantes:

- Maria Luiza Pin
- Anna Clara Zuany
- Gustavo Gama

### Descrição do Projeto: RPG Interativo em Python com Eventos Dinâmicos

Nosso projeto consiste na criação de um jogo de RPG interativo desenvolvido em Python, com foco em ser uma ferramenta de aprendizado para iniciantes. O propósito é construir uma aventura narrativa na qual o jogador toma decisões que afetam o rumo da história. Para tornar a experiência mais dinâmica e imprevisível, o projeto também integra uma API externa para simular eventos aleatórios, garantindo que cada partida seja uma experiência única.

O funcionamento do jogo se baseia em uma série de etapas interativas. O jogador inicia personalizando seu personagem com um nome e começa a jornada com uma quantidade pré-definida de pontos de vida. A cada rodada, o jogo apresenta um cenário e opções de ação. A grande diferença é que as consequências dessas escolhas não são fixas; para determinar resultados de eventos, como o dano recebido em uma armadilha ou o tesouro encontrado em um baú, o jogo se conecta a uma API de geração de números aleatórios. Isso introduz um elemento de sorte e rejogabilidade, já que uma mesma escolha pode ter desfechos diferentes a cada nova partida. O jogo termina quando a vida do personagem chega a zero (derrota) ou quando o jogador decide encerrar a aventura.

Este projeto é uma forma lúdica e prática de aplicar conceitos de programação em Python. Ele exercita o uso de variáveis, estruturas de decisão e laços de repetição. Além disso, o projeto agora introduz um conceito fundamental do desenvolvimento moderno: o consumo de APIs externas para buscar dados e adicionar funcionalidades dinâmicas. Isso demonstra como aplicações podem interagir com serviços na internet para criar interações mais ricas e menos previsíveis, abrindo portas para futuras expansões.

### Detalhamento do Funcionamento Lógico

Para dar vida ao jogo, duas estruturas da linguagem Python são essenciais: o laço de repetição `while` e as estruturas de decisão `if`, `elif` e `else`.

#### O Laço `while` (O Coração do Jogo)

O laço `while` é o que mantém o jogo em execução. Ele funciona como o motor da aventura, repetindo as rodadas de jogo (apresentar cenário, pedir escolha, mostrar resultado) enquanto as condições para continuar jogando forem verdadeiras. No nosso caso, o jogo continua enquanto a vida do personagem for maior que zero e ele não tiver decidido desistir. A cada final de rodada, o laço verifica novamente essas condições antes de iniciar a próxima, controlando o fluxo principal da partida.

#### As Estruturas `if`, `elif` e `else` (As Escolhas do Jogador)

Dentro de cada rodada do laço `while`, precisamos processar a decisão do jogador. É aqui que a estrutura `if/elif/else` se torna a ferramenta principal. Ela permite que o programa tome diferentes caminhos com base na entrada do usuário, criando a interatividade central do RPG.

`if`: É usado para verificar a primeira condição possível. Por exemplo, se o jogador escolheu a opção "1", o bloco de código dentro do `if` será executado.

`elif`: Caso a primeira condição seja falsa, o `elif` (abreviação de "else if") testa uma segunda condição. Podemos usar vários `elif` em sequência para criar múltiplos caminhos na história.

`else`: Se nenhuma das condições anteriores (do `if` e dos `elif`) for verdadeira, o bloco `else` é executado. Isso é útil para lidar com casos inesperados, como quando o jogador digita uma opção inválida.

## URLs e APIs Utilizadas

API: CSRNG API ([csrng.net](https://csrng.net))

Descrição: Uma API pública e gratuita utilizada para gerar números aleatórios. No projeto, é usada para simular a rolagem de um dado (gerando um número entre 1 e 6) e introduzir um elemento de sorte nos eventos do jogo.

URL de Exemplo:

<https://csrng.net/csrng/csrng.php?min=1&max=6>