**Relatório de Jogo - Sistema de Armas (Jogador Fictício + Classe)**

**1. Identificação do Projeto**

**Data de Criação:** 05/05/2025

**Autores:** Equipe de Documentação

**Linguagem:** Python

**2. Resumo do Jogo**

O jogo é uma aventura no estilo medieval, em que o jogador controla um personagem capaz de utilizar armas como lança, Machado e entre outras armas para enfrentar inimigos. Cada arma possui características próprias, como dano, alcance, velocidade e durabilidade.

O principal objetivo é sobreviver aos ataques dos inimigos, derrotando-os utilizando as armas disponíveis.

**Público-alvo:** Estudantes, jovens e jogadores fãs de RPG, aventura e jogos de sobrevivência.

**Inspirações:** Jogos de RPG clássicos.

**3. Mecânicas do Jogo**

**- Movimento do personagem:** O personagem pode se movimentar livremente no mapa.

**- Colisões:** Não foram implementadas colisões físicas avançadas, mas há interações básicas, como ataque aos inimigos.

**- Pontuação:** Não possui pontuação tradicional, o objetivo é sobreviver e derrotar inimigos.

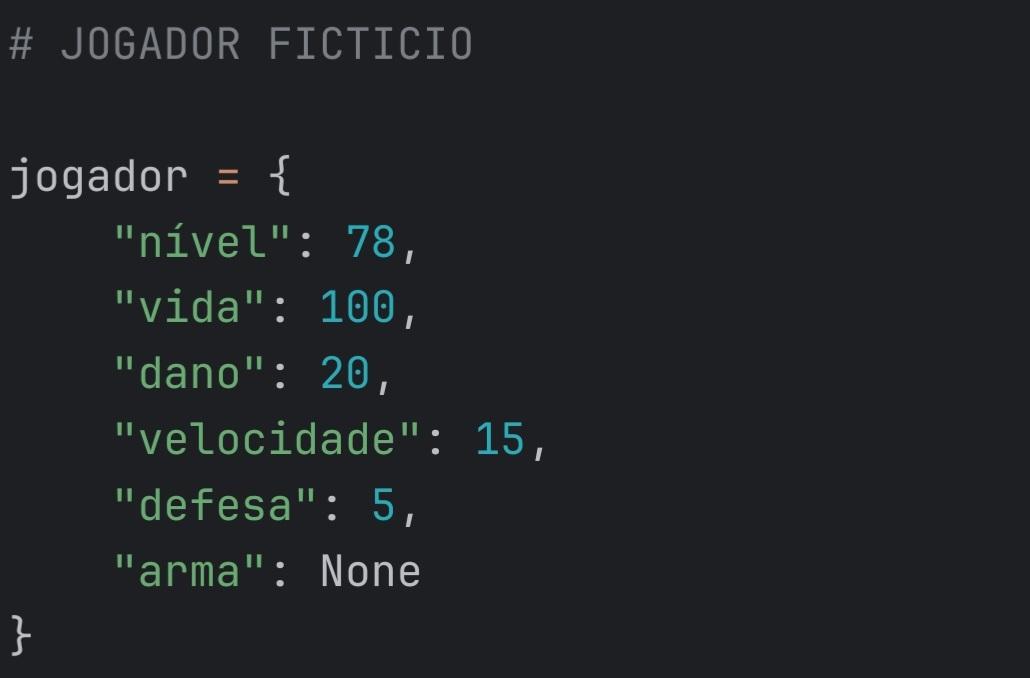
**- Fases/Níveis:**

Mundo aberto simples, sem fases.

**- Dificuldades:** Aumenta conforme o tempo de jogo, com mais inimigos surgindo.

**Relatório de Desenvolvimento — Jogador Fictício**

**Bloco 1 — Jogador Fictício:**

****

Este bloco é responsável por criar a estrutura de um jogador fictício, que será utilizado durante a execução do jogo. Este jogador possui atributos fundamentais que definem seu desempenho nas mecânicas, combates e interações dentro do sistema.

**Funcionamento:**

O dicionário chamado "jogador" contém os seguintes parâmetros:

**"nível":** Indica o nível atual do jogador, que influencia diretamente nos cálculos de força, resistência e poder de ataque.

**"vida":** Determina a quantidade de pontos de vida. Quando chega a zero, o jogador é considerado derrotado.

**"dano":** Valor que representa o quanto de dano o jogador consegue causar.

**"velocidade":** Reflete a agilidade do jogador, afetando o tempo de reação, movimentação ou ordem dos turnos.

**"defesa":** Quantidade de proteção que reduz parte dos danos recebidos.

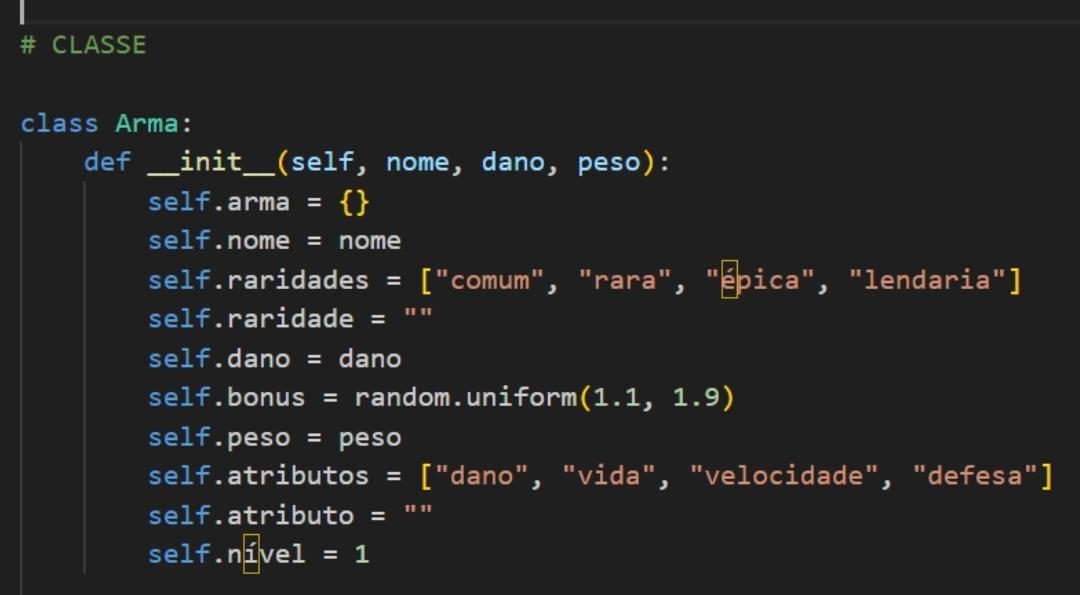
**"arma":** Define qual arma o jogador possui equipado. Inicialmente, este valor é None, indicando que o jogador começa sem nenhuma arma.

**Importância no jogo:**

Este bloco é essencial, pois configura a base do jogador no jogo, permitindo que ele interaja com as mecânicas, receba melhorias, armas e tenha seus atributos alterados conforme o desenvolvimento da partida.

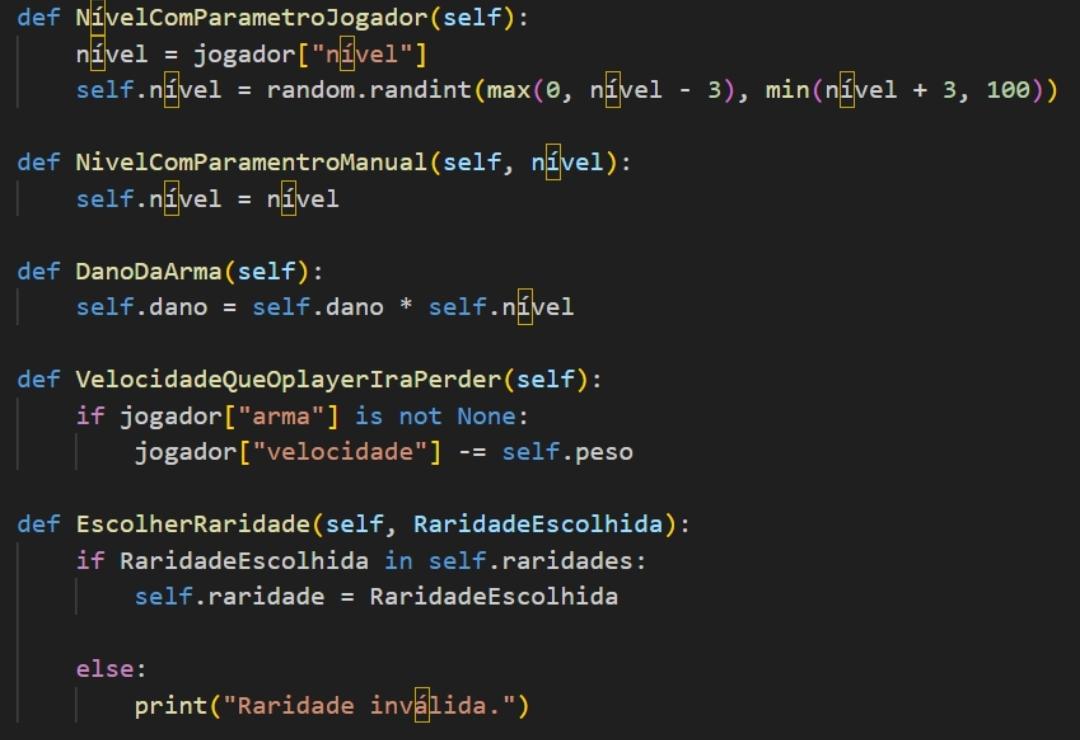
**Relatório de Desenvolvimento — Classe: Arma**

**Bloco 1 — Inicialização da Classe Arma**



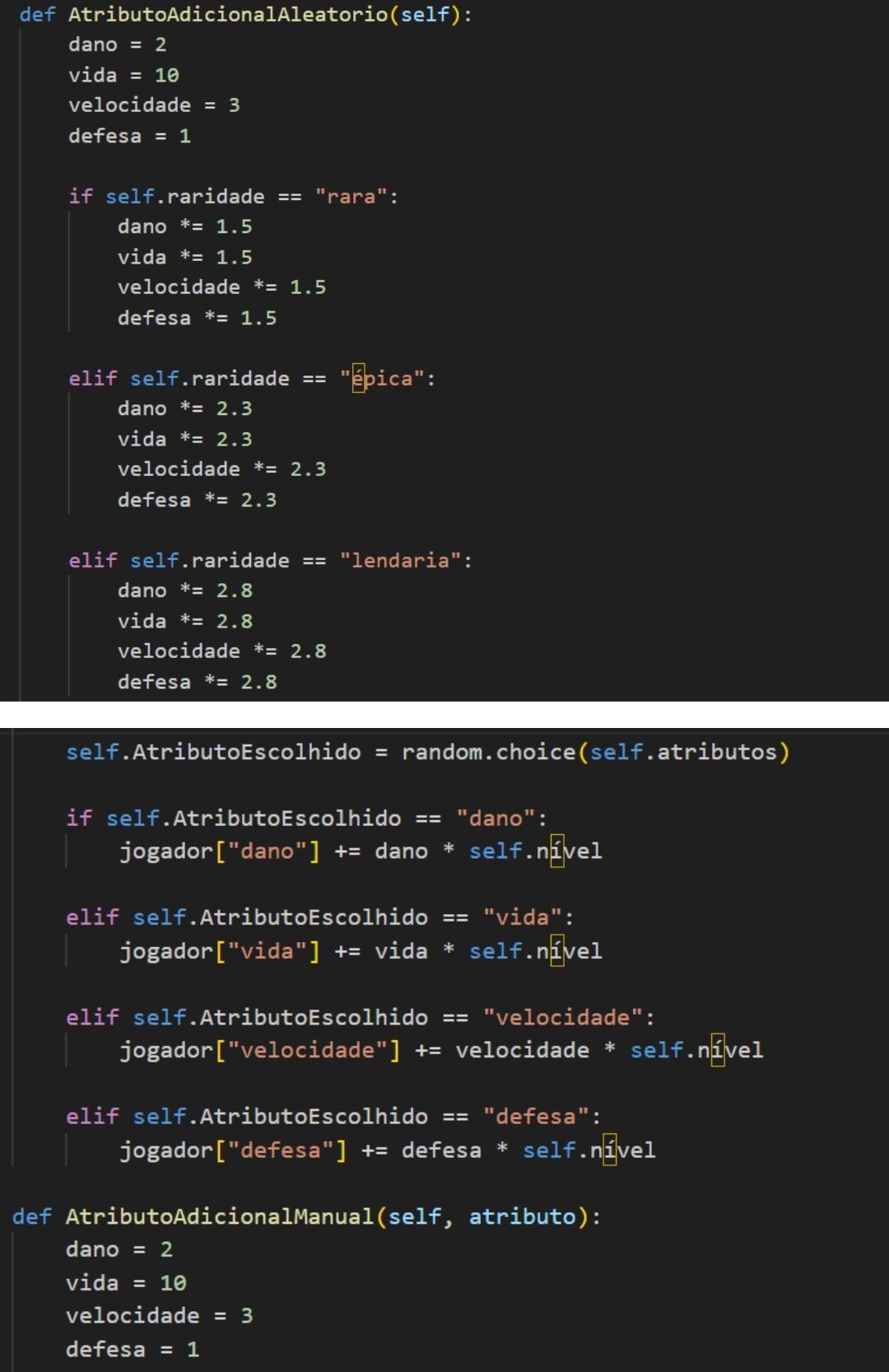
Este bloco define o início da classe Arma, responsável por armazenar os dados básicos de uma arma dentro do jogo. Nele são configurados os atributos principais, como nome, dano base e peso da arma. Também são definidos os tipos de raridade possíveis, os atributos adicionais (como vida, dano, velocidade e defesa), um bônus de experiência aleatório e o nível inicial da arma. Esse bloco prepara a estrutura da arma para ser personalizada nos próximos passos.

**Bloco 2 — Nível da Arma (Automático e Manual)**



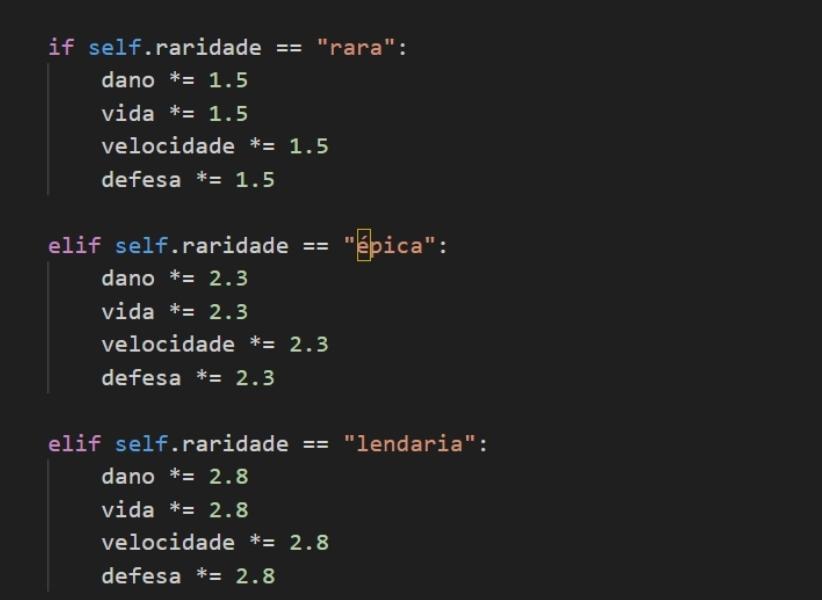
Este bloco reúne duas formas de definir o nível da arma. A primeira forma calcula o nível com base no nível do jogador, usando um valor aleatório dentro de um intervalo. A segunda permite definir o nível manualmente, de forma direta. O nível da arma influencia diretamente no dano e nos atributos adicionais que ela poderá aplicar.

**Bloco 3 — Cálculo de Dano da Arma**

****

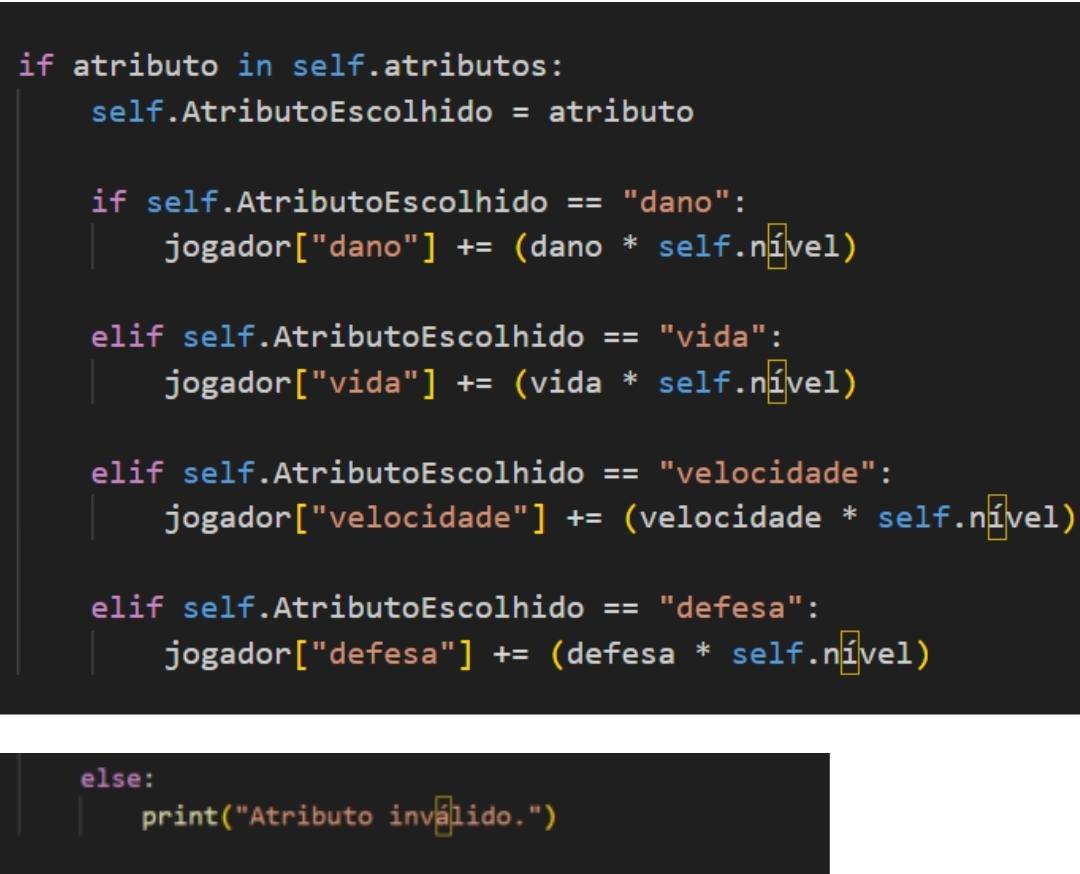
Este bloco é responsável por atualizar o dano da arma com base no seu nível. O valor de dano inicial é multiplicado pelo nível definido anteriormente, fazendo com que armas de mesmo tipo possam ter danos diferentes de acordo com a progressão do jogador ou a raridade.

**Bloco 4 — Atributo Adicional Aleatório**

****

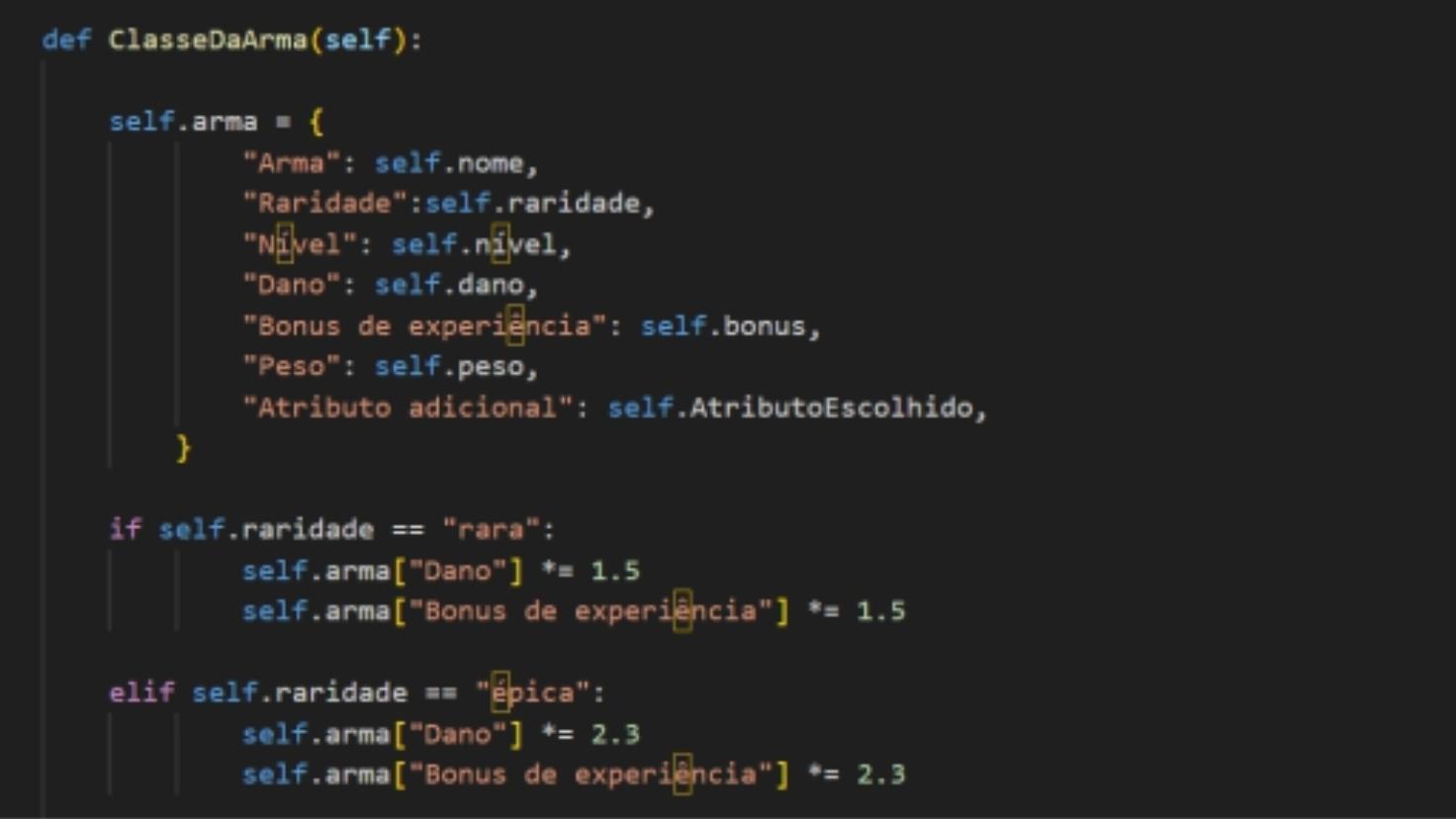
Neste bloco, a arma recebe um atributo extra aleatório que pode ser dano, vida, velocidade ou defesa. O valor atribuído depende da raridade da arma — quanto mais rara, maior o bônus aplicado. O sistema escolhe o atributo automaticamente e aplica ao jogador, ajustando conforme o nível da arma.

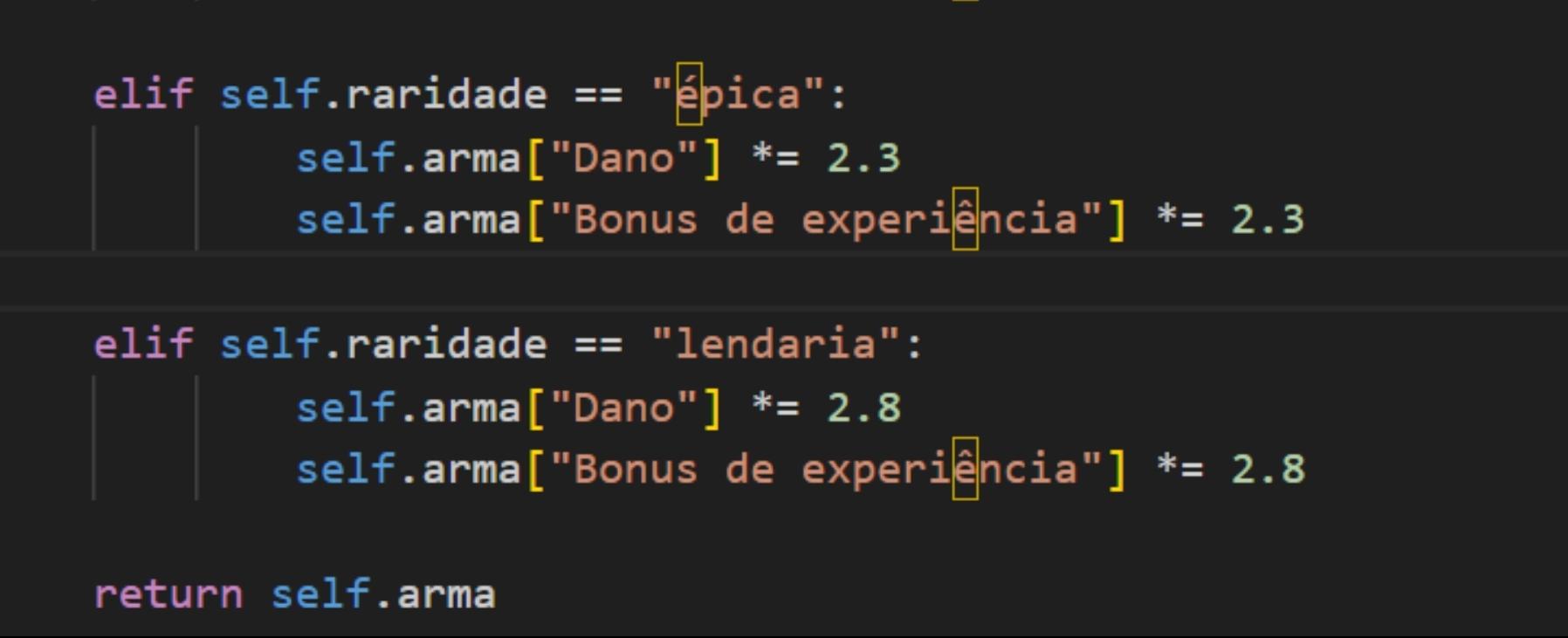
**Bloco 5 — Atributo Adicional Manual**

****

Este bloco oferece ao desenvolvedor a opção de escolher manualmente o atributo adicional da arma, ao invés de ser aleatório. Ele verifica se o atributo informado é válido e, se for, aplica ao jogador com um valor proporcional ao nível da arma. É uma forma de controle direto sobre o bônus da arma.

**Bloco 6 — Classe Final da Arma**

****

****

Este bloco monta e retorna o dicionário final da arma, reunindo todas as informações que foram definidas: nome, raridade, nível, dano final, bônus de experiência, peso e o atributo adicional aplicado. Esse bloco também aplica multiplicadores extras ao dano e ao bônus caso a arma seja rara, épica ou lendária, finalizando a customização da arma.

**Conclusão:**

A classe Arma é uma estrutura completa que permite criar armas personalizadas dentro do jogo. Com suas seis etapas bem definidas, ela garante controle total sobre atributos, raridade e escalonamento de poder. A possibilidade de escolher entre configurações automáticas ou manuais torna a classe versátil, adequada tanto para testes quanto para aplicação em jogos completos.

**Relatório de Classes: Dardos, Escudo, Flechas e Armadura**

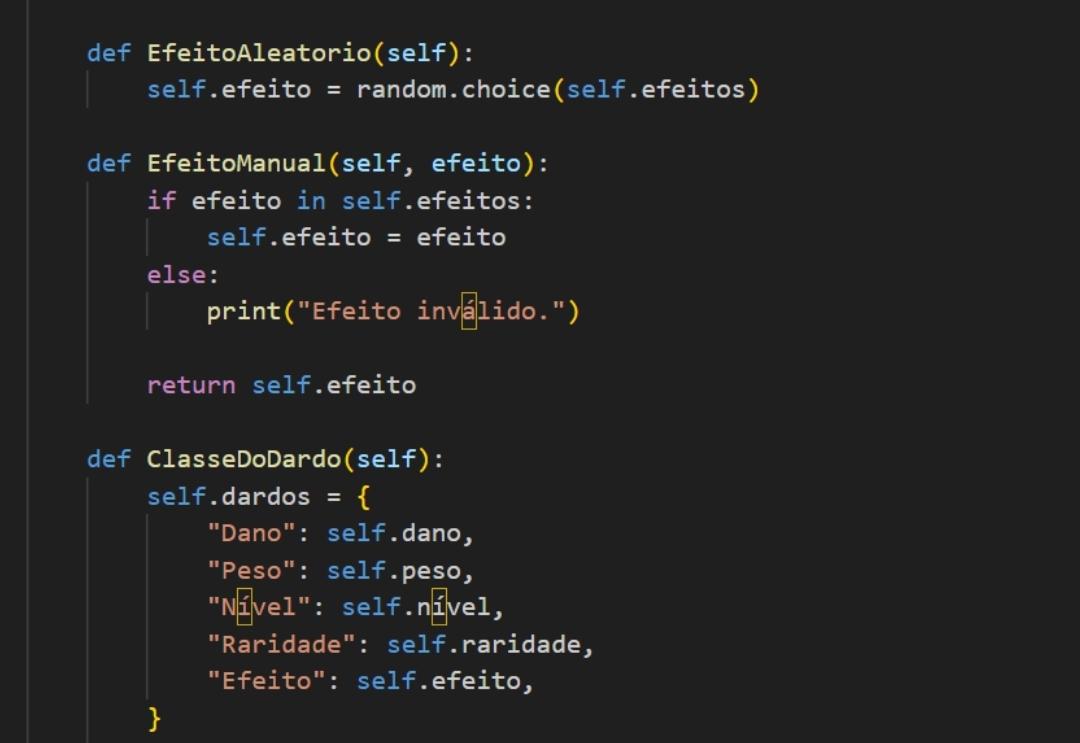
**Classe: Dardos**

**Bloco 1 — Inicialização da Classe Dardos**



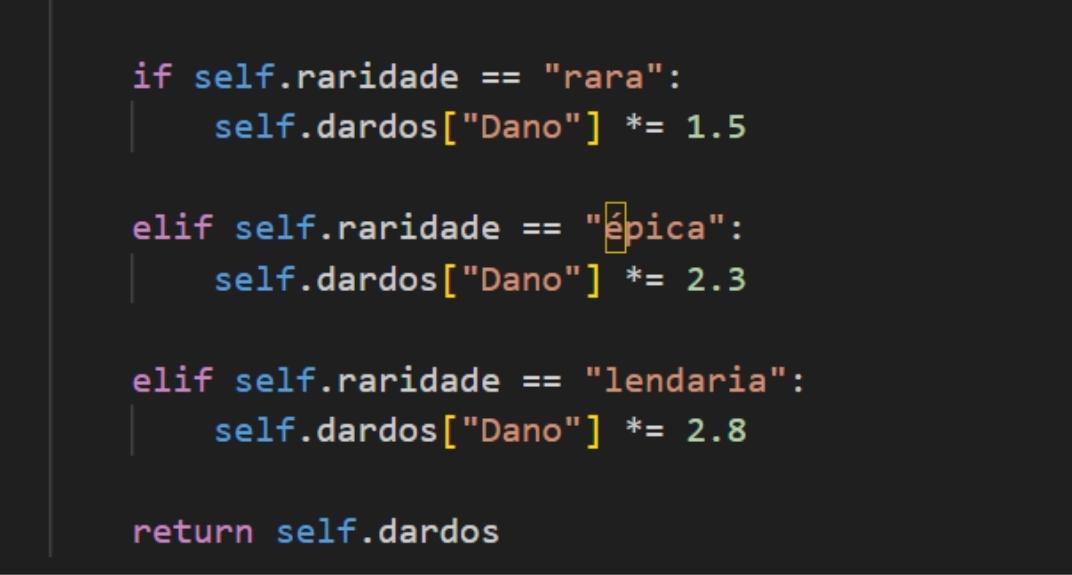
Este bloco define a estrutura básica da classe Dardos, responsável por representar uma arma de arremesso no jogo. São definidos os parâmetros essenciais como dano, peso e nível, além da lista de raridades disponíveis (comum, rara, épica e lendária). Também é configurada uma lista de possíveis efeitos especiais que os dardos podem aplicar, como veneno, explosivo, assoviante, entre outros. Um dicionário interno (self.dardos) é reservado para armazenar os dados finais do objeto.

**Bloco 2 — Configuração de Nível, Raridade e Efeito**



Este bloco cuida da personalização do dardo. A raridade é escolhida aleatoriamente, e com base nela, o dano da arma é ajustado proporcionalmente — armas raras causam mais dano. Também é definido um nível aleatório dentro de uma faixa com base no nível do jogador. Por fim, é aplicado um efeito especial aleatório, que afeta o comportamento da arma durante o uso (como causar dano extra ou aplicar efeitos negativos no inimigo).

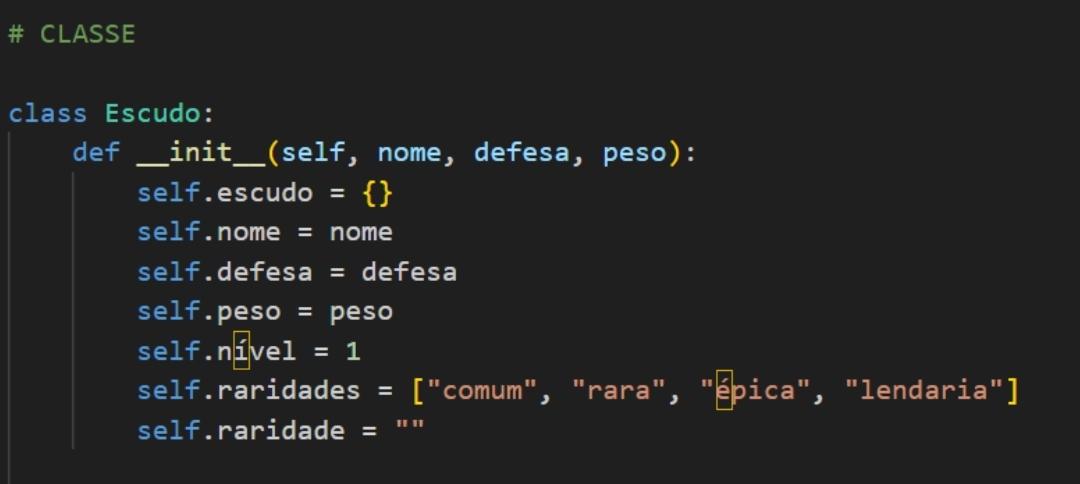
**Bloco 3 — Geração da Classe Final do Dardo**



Este bloco organiza todas as informações configuradas anteriormente e cria um dicionário final com os dados da arma. Ele inclui nome, raridade, dano ajustado, nível, efeito especial e peso. Esses dados são retornados prontos para serem utilizados no sistema do jogo, seja para exibição, comparação ou aplicação ao personagem.

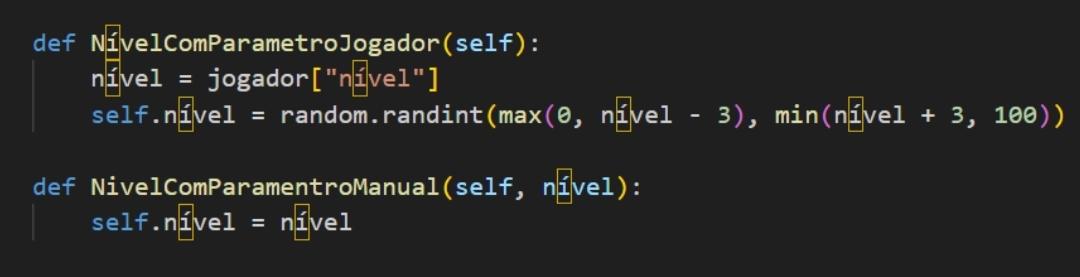
**Classe: Escudo**

**Bloco 1 — Inicialização da Classe Escudo**



Este bloco define os parâmetros básicos da classe Escudo, responsável por representar a defesa do jogador no jogo. São configurados os atributos principais: nome, defesa base, peso, além da lista de raridades possíveis (comum, rara, épica, lendária). O nível inicial do escudo é definido como 1, e um dicionário interno é criado para armazenar os dados finais do objeto.

**Bloco 2 — Definição de Nível (Automático e Manual)**



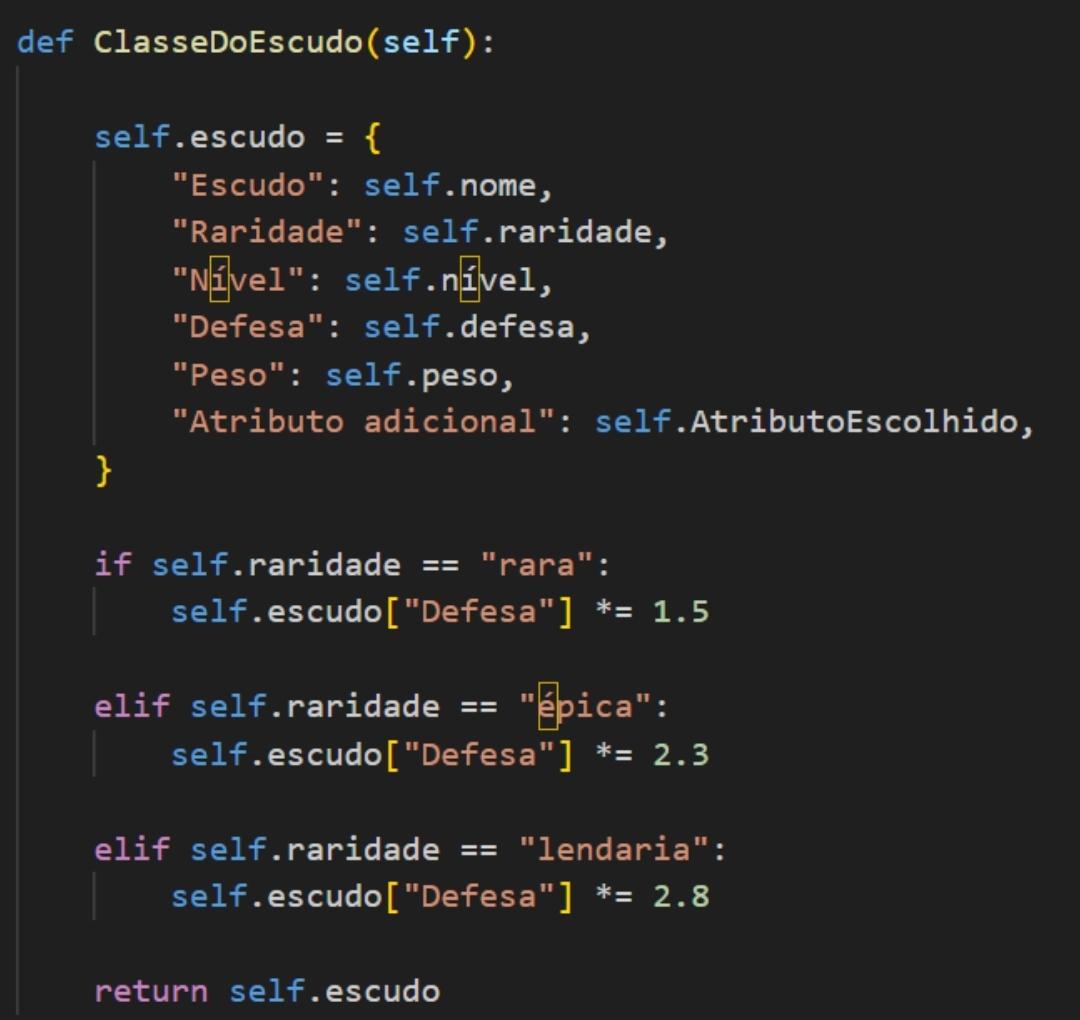
Neste bloco são apresentados dois métodos. O primeiro define o nível do escudo automaticamente, com base no nível do jogador, dentro de um intervalo de até três níveis acima ou abaixo, limitado ao máximo de 100. O segundo método permite definir o nível manualmente, oferecendo maior controle ao desenvolvedor. O nível influenciará diretamente na eficácia defensiva do escudo.

**Bloco 3 — Cálculo da Defesa Final**



Este bloco é responsável por calcular o valor final da defesa do escudo, multiplicando a defesa base pelo nível definido anteriormente. Isso garante que escudos de mesmo tipo possam ter performances diferentes, conforme a progressão do jogador ou a raridade aplicada ao equipamento.

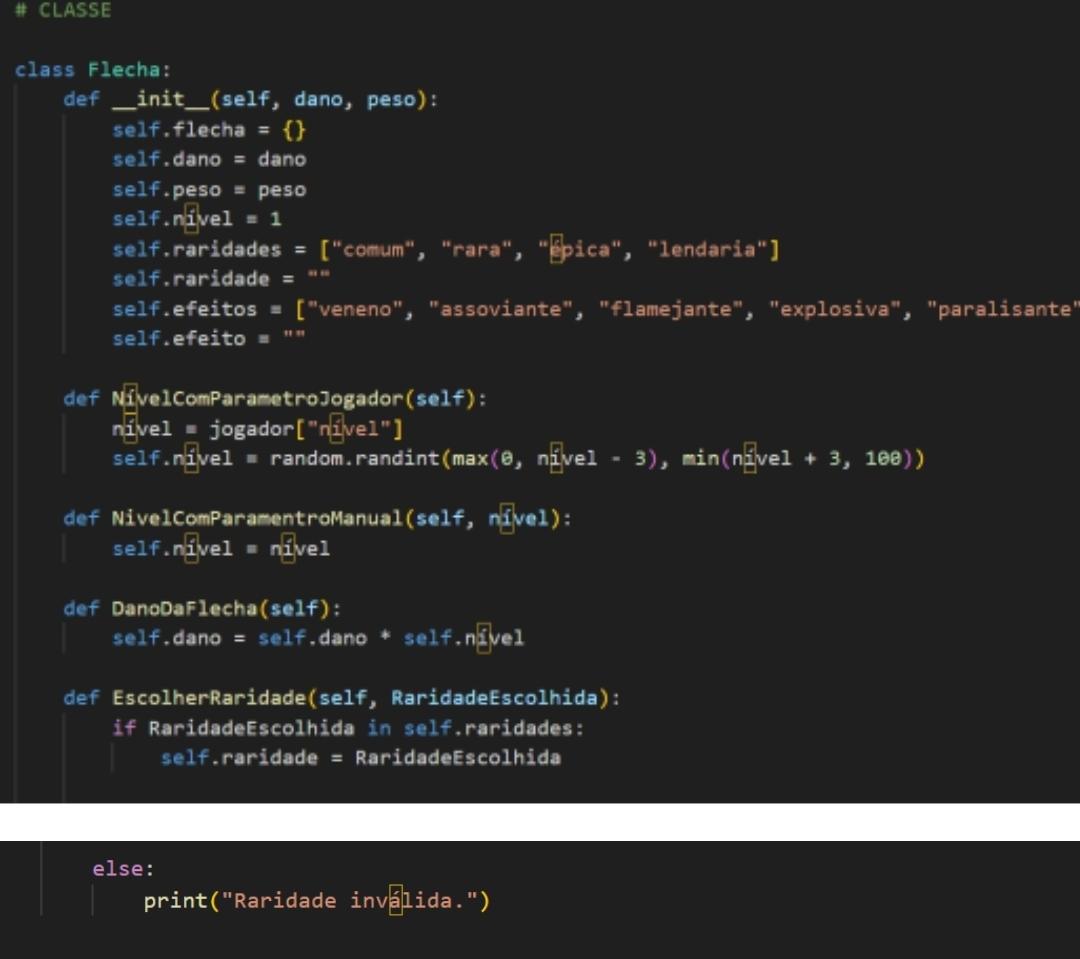
**Bloco 4 — Geração da Classe Final do Escudo**



O último bloco reúne todos os dados configurados anteriormente e os organiza em um dicionário final, com os seguintes campos: nome, raridade, nível, defesa e peso. Esse dicionário representa o escudo final, pronto para ser equipado ou exibido no sistema do jogo.

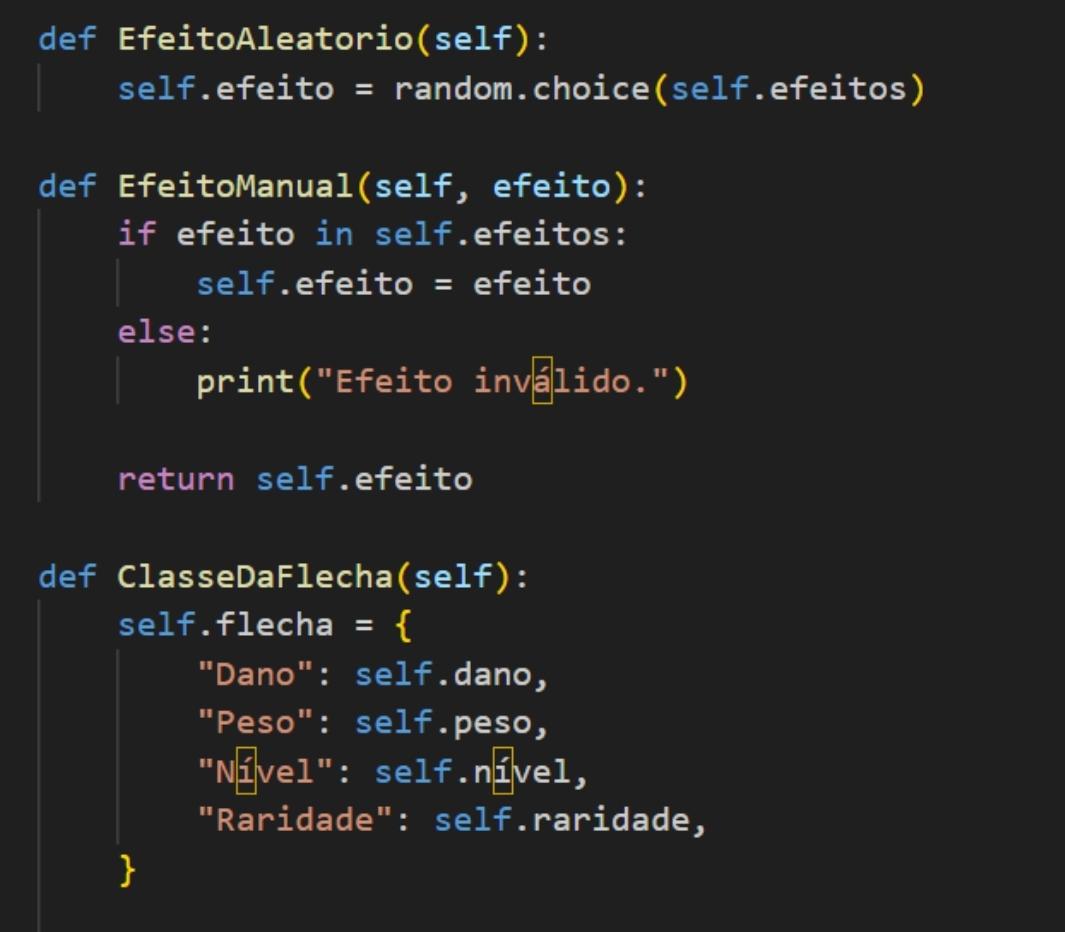
**Classe: Flechas**

**Bloco 1 — Inicialização da Classe Flechas**



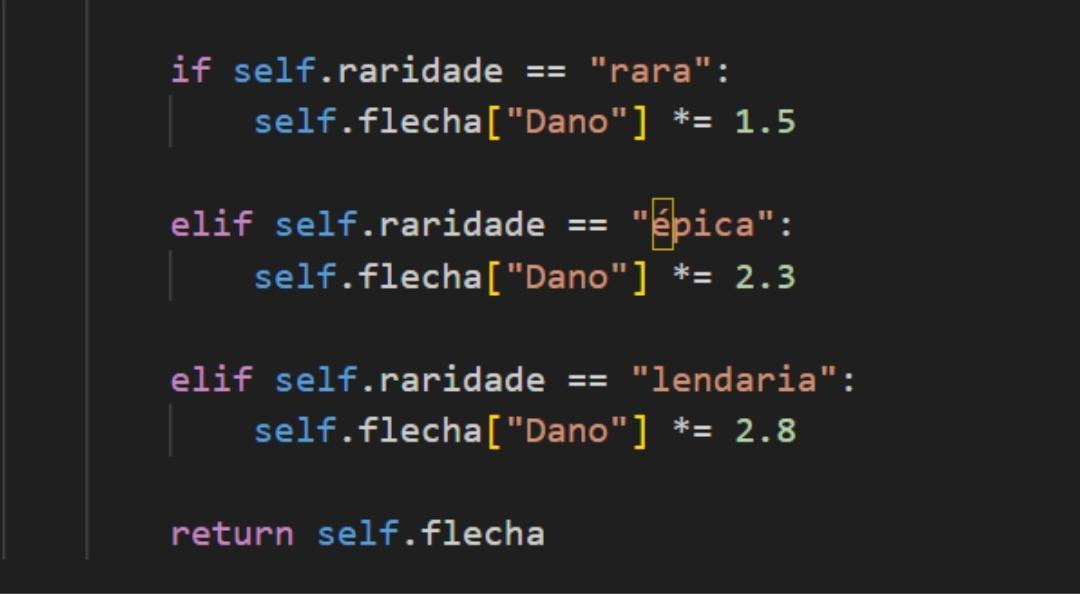
Este bloco define os atributos principais da classe Flechas, representando flechas que podem ser utilizadas pelo jogador. São definidos parâmetros como nome, dano base, tipo da flecha e uma lista de raridades possíveis (comum, rara, épica, lendária). Também é criado um dicionário interno chamado self.flechas, que armazenará todos os dados finais da flecha.

**Bloco 2 — Definição de Raridade e Ajuste de Dano**



Este bloco é responsável por definir a raridade da flecha de forma aleatória. Dependendo da raridade sorteada, o dano da flecha é multiplicado por um valor específico: raridades maiores resultam em flechas com dano significativamente superior. Esse sistema permite balancear a força das flechas conforme sua raridade.

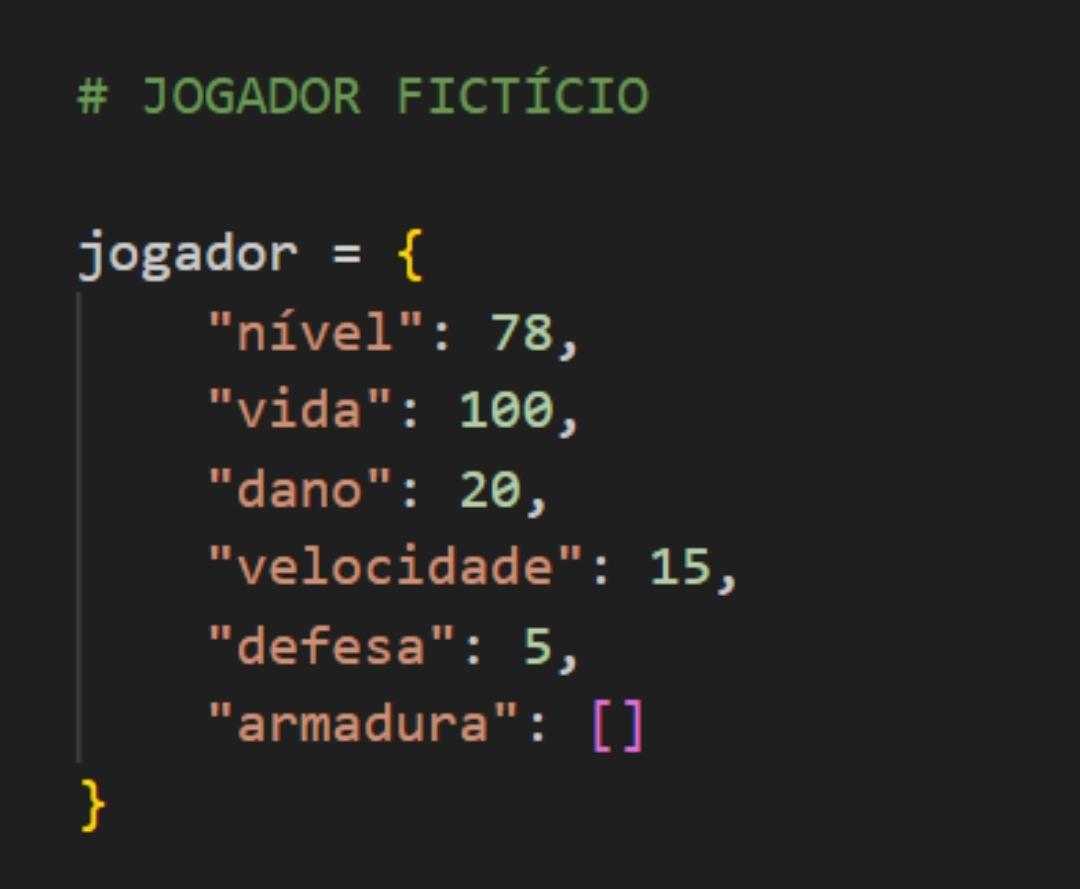
**Bloco 3 — Geração da Classe Final da Flecha**



Neste bloco, os dados da flecha são reunidos em um dicionário final, contendo o nome, tipo, raridade e dano ajustado. Esse dicionário representa a flecha pronta para ser usada no jogo, seja para exibição, ataque ou inventário.

**Classe: Armadura**

**Bloco 1 — Jogador Fictício**



Este bloco define um jogador fictício, usado como referência para aplicar os efeitos da armadura. Ele possui atributos básicos como nível, vida, dano, velocidade e defesa, que serão modificados conforme o tipo e a raridade da armadura equipada. Esse jogador é utilizado para testar e aplicar diretamente os resultados da armadura gerada.

**Bloco 2 — Inicialização da Classe Armadura**



Neste bloco é iniciada a classe que representa a armadura. São definidos os parâmetros básicos: nome, tipo de armadura (como capacete, peitoral, luvas, calça, botas) e raridade. O código também cria listas com os tipos possíveis de armaduras e raridades (comum, rara, épica, lendária), além de reservar um dicionário interno para armazenar os dados finais da armadura criada.

**Bloco 3 — Aplicação dos Atributos e Geração da Armadura**



Este bloco aplica os efeitos da armadura diretamente ao jogador, com base em sua raridade. A raridade determina o multiplicador dos atributos alterados — como vida, defesa e velocidade. Armas de raridade mais alta resultam em bônus mais significativos. Ao final, o código monta um dicionário completo da armadura, contendo seu nome, tipo, raridade, e os bônus aplicados ao jogador.

**Conclusão:**

**As quatro classes apresentadas :*Dardos, Escudo, Flechas e Armadura***, foram desenvolvidas para funcionar de forma integrada ao sistema de equipamentos do jogo. Cada uma possui estrutura própria, lógica de raridade, escalonamento de atributos e retorno em dicionário, facilitando a aplicação no personagem. A divisão em blocos permite organização, reutilização e fácil manutenção, além de garantir que cada item possua impacto direto e equilibrado na jogabilidade. O uso de aleatoriedade em efeitos e raridades torna a experiência mais dinâmica e recompensadora para o jogador.