



Aluno(a): _____ Nota: _____

Prof. Tiago Pessoa Ferreira de Lima

Questão 1: (3 pontos)

Análise o código abaixo, que modela um combate em um jogo:

```
# Dados do jogador
player_vida = 100
player_ataque = 15
player_nome = "Guerreiro"

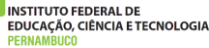
# Dados do inimigo
inimigo_vida = 50
inimigo_defesa = 5
inimigo_nome = "Goblin"

# Procedimento (função) para o ataque
def atacar_inimigo(ataque, defesa_inimigo, vida_inimigo):
    dano = max(0, ataque - defesa_inimigo)
    vida_inimigo -= dano
    print(f"{player_nome} causou {dano} de dano ao {inimigo_nome}.")
    return vida_inimigo

inimigo_vida = atacar_inimigo(player_ataque, inimigo_defesa,
inimigo_vida)
```

a) Porque o código representa o paradigma procedural e como ele difere da POO.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



b) Reescreva em POO usando classes Jogador e Inimigo, incluindo métodos como atacar.

This image shows a single page of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Questão 2: (3 pontos)

Analise a classe Musica:

```
class Musica:
    total_musicas = 0 # Atributo de classe

    def __init__(self, titulo, artista, duracao):
        self.titulo = titulo
        self.artista = artista
        self.duracao = duracao
        Musica.total_musicas += 1
```

a) Explique a diferença entre **atributo de classe** (`total_musicas`) e de **instância** (`titulo`).

b) Crie duas instâncias de Musica e imprima `Musica.total_musicas`. Modifique o código para adicionar um método que retorne a duração formatada.

Questão 3: (4 pontos)

Considere a classe ContaBancaria:

```
class ContaBancaria:
    def __init__(self, titular, saldo=0):
        self.titular = titular
        self.saldo = saldo # saldo acessível diretamente

    def depositar(self, valor):
        if valor > 0:
            self.saldo += valor

    def sacar(self, valor):
        if 0 < valor <= self.saldo:
            self.saldo -= valor
```

a) Qual é o problema de o atributo saldo ser acessado e modificado diretamente? Dê pelo menos um exemplo de consequência negativa num sistema bancário.

b) Crie o método transferencia que permita debitar um valor da conta atual e creditar em outra conta, garantindo que não seja possível transferir mais do que o saldo disponível.

Rascunho