

```
import random
import time
import pickle
import os
```



```
sim=["sim", "s"]
nao=["nao", "n", "nÃ£o"]
```

```
# ===== FunÃ§ÃÃes =====
def batalha(x,y,z): #FunÃ§ÃÃo de batalha
    return x-y+z
```

```
def funcaoexp(xp,xp_ganho): #FunÃ§ÃÃo de experiÃªncia
    xp=xp+xp_ganho
    return xp
```

```
def pokemon_inicial(lista,pokemon,Nindex): #FunÃ§ÃÃo que define o prÃ³prio inspermon
    ataque1=lista[pokemon]["ataque"]
    defesa=lista[pokemon]["defesa"]
    vida_inicial=lista[pokemon]["vida"]
    index=Nindex
    evolucao=1
    return(ataque1,defesa,vida_inicial,index,evolucao)
```

```
def escolha_adversÃrio(lista): # funÃ§ÃÃo de randomizaÃÃo do oponente e definiÃÃo das suas caracteri

    nome_oponente=random.choice(list(lista.keys())) # Escolha aleatÃ³ria do inspermon adversÃrio
    ataque_oponente=lista[nome_oponente]["ataque"]
    defesa_oponente=lista[nome_oponente]["defesa"]
    vida_oponente=lista[nome_oponente]["vida"]
    return (nome_oponente,ataque_oponente,defesa_oponente,vida_oponente)
```

```
def MudaNome(nome): #FunÃ§ÃÃo para mudanÃça de nome do inspermon
    while True:

        x = input("Deseja mudar o nome do seu InspermÃ³n?\n ")
        x = x.lower()

        if x in sim:
            nome = input("Qual serÃ¡ o nome do seu InspermÃ³n?\n ")
            print("Agora o nome do seu InspermÃ³n Ã© {}".format(nome))
            break

        elif x in nao:
            print("O nome do seu InspermÃ³n ainda Ã© {}".format(nome))
            break

        else:
```

```
        print("Digite um comando vÃ¡lido (Sim ou NÃ£o) ")
        continue

    nome = nome

    return nome

def funcao_evolucao(nome,pokemon,evolucao,ataque_extra): # FunÃ§Ã£o para a evoluÃ§Ã£o dos inspermons

    print("{} estÃ¡ se preparando para evoluir".format(nome))
    time.sleep(1)
    print(".")
    time.sleep(1)
    print(".")
    time.sleep(1)
    print(".")
    time.sleep(1)
    print("{} evoluiu para {}".format(nome, evolucao))

    ataque1=dicionario_inspermons_iniciais[evolucao]["ataque"] + ataque_extra
    defesa=dicionario_inspermons_iniciais[evolucao]["defesa"]
    vida_inicial=dicionario_inspermons_iniciais[evolucao]["vida"]

    nome = evolucao

    vida = vida_inicial

    time.sleep(1)
    print("Seu InspermÃ³ tem os seguintes atributos: Ataque:{}, Defesa:{}, Vida:{}".format(ataque1,defesa,vida))

    return(ataque1,defesa,vida_inicial,nome)

# ===== Arquivos abertos com os inspermons =====
===

with open("inspermons_faceis.pickle","rb") as arquivo_inspermons_faceis:
    dicionario_inspermons_faceis=pickle.load(arquivo_inspermons_faceis)

with open("inspermons_iniciais.pickle","rb") as arquivo_inspermons_iniciais:
    dicionario_inspermons_iniciais=pickle.load(arquivo_inspermons_iniciais)

with open("inspermons_medios.pickle","rb") as arquivo_inspermons_medios:
    dicionario_inspermons_medios=pickle.load(arquivo_inspermons_medios)

with open("inspermons_dificeis.pickle","rb") as arquivo_inspermons_dificeis:
    dicionario_inspermons_dificeis=pickle.load(arquivo_inspermons_dificeis)

with open("inspermons_lendarios.pickle","rb") as arquivo_inspermons_lendarios:
    dicionario_inspermons_lendarios=pickle.load(arquivo_inspermons_lendarios)
```

```

lista_faceis = ["Pikaxu", "Tatata", "Pigey", "Dogerpie", "Magipark", "Sharmander", "Skuirtle", "Bulbatauro"]
lista_medios = ["Machups", "Geogrude", "Goalbate", "Harboqui", "Bidriu", "Sharmilion", "Ivitauro", "Uarturtle"]
lista_dificeis = ["Nidorrei", "Esnorlacs", "Vaporneon", "Dragoday", "Arcanove", "Cheirazard", "Venustauro", "Blastoide"]
lista_lendarios = ["Articuno", "Zapdos", "Moltres"]

```



```

#===== JOGO =====
=

while True:
    print("Bem vindo ao Inspermã³n")
    time.sleep(1)
# ===== Inspermon Inicial =====

    while True:
        while True:

            abrir_salvo=input("Deseja abrir jogo salvo? \n ") #Verifica se o usÃ¡rio quer abrir um jogo salvo
            time.sleep(0.5)

            if abrir_salvo in sim:

                try:
                    salvo=pickle.load(open("jogo_salvo","rb")) # Traz o jogo salvo com os dados guardados
                    nome=salvo["Inspermon"][0]
                    ataque1=salvo["Inspermon"][1]
                    defesa=salvo["Inspermon"][2]
                    vida_inicial=salvo["Inspermon"][3]
                    xp=salvo["Inspermon"][4]
                    dinheiro=salvo["Inspermon"][5]
                    lista_nomes = salvo["Inspermon"][6]
                    ataque_extra = salvo["Inspermon"][7]
                    defesa_extra = salvo["Inspermon"][8]
                    vida_extra = salvo["Inspermon"][9]
                    evolucao=salvo["Inspermon"][10]
                    carregado=sim
                    print("Carregando dados...")
                    time.sleep(2)
                    break

                except:
                    print("NÃ£o existe nenhum jogo salvo") # Aviso caso nÃ£o exista jogo salvo
                    time.sleep(0.5)
                    carregado=nao
                    break

```

```

        if abrir_salvo in nao: # Caso o usuário não queira abrir o jogo salvo
            carregado=nao
            break

    else:
        print("Digite um comando válido: sim ou não")
        continue

if carregado==sim: # Checa se houve o carregamento de dados para pular o processo de escolha de um inspermon
    break

if carregado==nao: # Caso não tenha nenhum arquivo salvo aberto

    ataque_extra = 0
    defesa_extra = 0
    vida_extra = 0

    xp = 0
    dinheiro = 0

    lista_nomes = []

    print("Com qual Inspermon você deseja iniciar?") # apresenta o dos inspermons
    time.sleep(0.5)
    print("- Sharmander: Ataque = {} , Defesa = {} , Vida = {}".format(dicionario_inspermons_iniciais["Sharmander"]["ataque"],dicionario_inspermons_iniciais["Sharmander"]["defesa"],dicionario_inspermons_iniciais["Sharmander"]["vida"])))
    time.sleep(0.5)
    print("- Skuirtle: Ataque = {} , Defesa = {} , Vida = {}".format(dicionario_inspermons_iniciais["Skuirtle"]["ataque"],dicionario_inspermons_iniciais["Skuirtle"]["defesa"],dicionario_inspermons_iniciais["Skuirtle"]["vida"])))
    time.sleep(0.5)
    print("- Bulbatauro: Ataque = {} , Defesa = {} , Vida = {}".format(dicionario_inspermons_iniciais["Bulbatauro"]["ataque"],dicionario_inspermons_iniciais["Bulbatauro"]["defesa"],dicionario_inspermons_iniciais["Bulbatauro"]["vida"])))
    time.sleep(0.5)

    inspermon_inicial = input("->") # definição de qual inspermon será escolhido
    inspermon_inicial = inspermon_inicial.lower()

    if inspermon_inicial == "sharmander": #Caso o usuário opte pelo sharmander
        ataque1,defesa,vida_inicial,index,evolucao=pokemon_inicial(dicionario_inspermons_iniciais,"Sharmander",1)

        nome="Sharmander"
        lista_nomes.append("Sharmander")
        break

    if inspermon_inicial == "skuirtle": #Caso o usuário opte pelo skuirtle

```

```

        ataque,defesa,vida_inicial,index,evolucao=pokemon_inicial(dicionario_inspermons_iniciais,"Skui
irtle",2)

        nome="Skuiurtle"
        lista_nomes.append("Skuiurtle")
        break
    elif inspermon_inicial == "bulbatauro": #Caso o usu rio opte pelo bulbatauro
        ataque,defesa,vida_inicial,index,evolucao=pokemon_inicial(dicionario_inspermons_iniciais,"Bul
batauro",3)

        nome="Bulbatauro"
        lista_nomes.append("Bulbatauro")
        break

    else:
        print("Digite um Insperm n v lido")
        time.sleep(0.5)
        continue

    nome=MudaNome(nome) # Checa se o usu rio quer mudar o nome do pr prio inspermon

    vida = vida_inicial
    time.sleep(0.5)
    print("Seu Insperm n tem os seguintes atributos: Ataque:{}, Defesa:{}, Vida:{}".format(ataque,defesa,vida)) # Apre
senta   o do Inspermon
    time.sleep(0.5)

# ===== In cio do jogo =====
    while True: # Jogo
        vida = vida_inicial
        comando = input("O que voc  deseja fazer? (Passear, Loja, Dormir ou Insperdex?):\n ") # Pergunta sobre o que
o usu rio deseja fazer
        comando = comando.lower()

#DORMIR
        if comando == "dormir": # Dormir: para o jogo e salva os dados
            break

#INSPERDEX
        elif comando == "insperdex": # Mostra a insperdex

            print("====Insperdex====")

            for i in lista_nomes: # Checa uais inspermons ja foram encontrados

                if i in lista_faceis:

                    time.sleep(0.3)
                    print("{}: ".format(i),end="")
                    print("Ataque: {}, ".format(dicionario_inspermons_faceis[i]["ataque"]),end="")

```

```

        print("Defesa: {}", ".format(dicionario_inspemons_faceis[i]["defesa"]),end="")
        print("Vida: {}",".format(dicionario_inspemons_faceis[i]["vida"]))

    elif i in lista_medios:

        time.sleep(0.3)
        print("{}: ".format(i),end="")
        print("Ataque: {}", ".format(dicionario_inspemons_medios[i]["ataque"]),end="")
        print("Defesa: {}", ".format(dicionario_inspemons_medios[i]["defesa"]),end="")
        print("Vida: {}",".format(dicionario_inspemons_medios[i]["vida"]))

    elif i in lista_dificeis:

        time.sleep(0.3)
        print("{}: ".format(i),end="")
        print("Ataque: {}", ".format(dicionario_inspemons_dificeis[i]["ataque"]),end="")
        print("Defesa: {}", ".format(dicionario_inspemons_dificeis[i]["defesa"]),end="")
        print("Vida: {}",".format(dicionario_inspemons_dificeis[i]["vida"]))

    elif i in lista_lendarios:

        time.sleep(0.3)
        print("{}: ".format(i),end="")
        print("Ataque: {}", ".format(dicionario_inspemons_lendarios[i]["ataque"]),end="")
        print("Defesa: {}", ".format(dicionario_inspemons_lendarios[i]["defesa"]),end="")
        print("Vida: {}",".format(dicionario_inspemons_lendarios[i]["vida"]))

    time.sleep(0.3)
    print("=====")

    continue

#LOJA

    elif comando == "loja":

        while True: #verifica qual atributo quer comprar
            time.sleep(0.5)
            compra=input("Você está na loja e tem {} inpermoedas, o que quer comprar: ataque, defesa ou vida? Você também pode sair digitando: sair\n ".format(dinheiro))

            if compra=="ataque": # caso o atributo escolhido seja o ataque
                time.sleep(1)
                print("Você pode aumentar seu ataque em 1 por 10 inpermoedas, 5 por 50 inpermoedas ou 10 por 100 inpermoedas. Você também pode sair digitando: sair")

                while True:

```

```

compra_ataque=input("Quanto deseja comprar?\n ")
time.sleep(1)

if compra_ataque=="1": # Compra de apenas um ponto

    if dinheiro >= 10:
        ataque_extra = ataque_extra + 1
        ataquel = ataquel + 1
        print("Agora o ataque do seu Inspermon Ã©: {}".format(ataquela))

        dinheiro = dinheiro - 10
        print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
        break
    else:
        print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
        continue

if compra_ataque=="5": #compra de 5 pontos

    if dinheiro >= 50:
        ataque_extra = ataque_extra + 5
        ataquel = ataquel + 5
        print("Agora o ataque do seu Inspermon Ã©: {}".format(ataquela))

        dinheiro = dinheiro - 50
        print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
        break
    else:
        print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
        continue

if compra_ataque=="10": # Compra de 10 pontos

    if dinheiro >= 100:
        ataque_extra = ataque_extra + 10
        ataquel = ataquel + 10
        print("Agora o ataque do seu Inspermon Ã©: {}".format(ataquela))

        dinheiro = dinheiro - 100
        print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
        break
    else:
        print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
        continue

```

```
if compra_ataque=="sair":
    break

else:
    print("Digite um comando vÃ¡lido")
    time.sleep(0.5)
    continue

elif compra=="defesa": # Caso o atributo escolhido seja defesa
    time.sleep(1)
    print("VocÃª pode aumentar sua defesa em 1 por 10 inpermoedas, 5 por 50 inpermoedas ou
10 por 100 inpermoedas. VocÃª tambÃ©m pode sair digitando: sair")

while True:
    compra_defesa=input("Quanto deseja comprar?\n ")
    time.sleep(1)

    if compra_defesa=="1": # Compra de 1 ponto

        if dinheiro >= 10:
            defesa_extra = defesa_extra + 1
            defesa = defesa + 1
            print("Agora a defesa do seu Inspermon Ã©: {}".format(defesa))
            dinheiro = dinheiro - 10
            print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
            break

        else:
            print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
            continue

    if compra_defesa=="5": # Compra de 5 pontos

        if dinheiro >= 50:
            defesa_extra = defesa_extra + 5
            defesa = defesa + 5
            print("Agora a defesa do seu Inspermon Ã©: {}".format(defesa))
            dinheiro = dinheiro - 50
            print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
            break

        else:
            print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
            continue

    if compra_defesa=="10": # Compra de 10 pontos

        if dinheiro >= 100:
            defesa_extra = defesa_extra + 10
```



```
defesa = defesa + 10
print("Agora a defesa do seu Inspermon Ã©: {}".format(defesa))
dinheiro = dinheiro - 100
print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
break

else:
    print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
    continue

if compra_defesa=="sair":
    break

else:
    print("Digite um comando vÃ¡lido")
    time.sleep(0.5)
    continue

elif compra=="vida": # Caso o atributo escolhido seja a vida
    time.sleep(1)
    print("VocÃª pode aumentar sua vida em 1 por 10 inpermoedas, 5 por 50 inpermoedas ou 1
0 por 100 inpermoedas. VocÃª tambÃ©m pode sair digitando: sair")

while True:
    compra_vida=input("Quanto deseja comprar?\n ")
    time.sleep(1)

    if compra_vida=="1": # Compra de 1 ponto

        if dinheiro >= 10:
            vida_extra = vida_extra + 1
            vida = vida + 1
            print("Agora a vida do seu Inspermon Ã©: {}".format(vida))
            dinheiro = dinheiro - 10
            print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
            break

        else:
            print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")
            continue

    if compra_vida=="5": # Compra de 5 pontos

        if dinheiro >= 50:
            vida_extra = vida_extra + 5
            vida = vida + 5
            print("Agora a vida do seu Inspermon Ã©: {}".format(vida))
```

```

        dinheiro = dinheiro - 50
        print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
        break
    else:
        print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")

        continue

    if compra_vida=="10": # Compra de 10 pontos

        if dinheiro >= 100:
            vida_extra = vida_extra + 10
            vida = vida + 10
            print("Agora a vida do seu Inspermon Ã©: {}".format(vida))
            dinheiro = dinheiro - 100
            print("Seu saldo Ã© de {} Inspermoedas".format(dinheiro))
            break

        else:
            print("Infelizmente vocÃª nÃ£o tem inspermoedas suficientes pa
ra comprar esta item")

            continue

    if compra_vida=="sair":
        break

    else:
        print("Digite um comando vÃ¡lido")
        time.sleep(0.5)
        continue

elif compra=="sair":
    time.sleep(0.5)
    print("Volte sempre!")
    time.sleep(0.5)
    break

else:
    print("Digite um comando vÃ¡lido")
    continue

#PASSEAR

elif comando == "passear": # Passear em busca de novos inspermons
    print("Passeando...")
    time.sleep(1)

    print("VocÃª encontrou um InspermÃ³n") # Encontro com um inspermon
    time.sleep(1)

```

```

        if evolucao == 1: #Chance de encontrar tipos de inspermon com o pokemon inicial
            nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicionario_ins
permons_faceis)

        elif evolucao == 2: #Chance de encontrar tipos de inspermon com a primeira evoluã$ãfo

            chance_2 = random.randint(1,5)

            if chance_2 == 1: # Chance de inspermon facil
                nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicion
ario_inspermons_faceis)

            else: # Chance de inspermon medio
                nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicion
ario_inspermons_medios)

        elif evolucao == 3: #Chance de encontrar tipos de inspermon com a segunda evoluã$ãfo
            chance_3_facil = [2, 3, 4]
            chance_3_medio = [5, 6, 7, 8, 9, 10]

            chance_3 = random.randint(1, 20)

            if chance_3 == 1: # Chance de inspermon lendario
                nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicion
ario_inspermons_lendarios)

            elif chance_3 in chance_3_facil: # Chance de inspermon facil
                nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicion
ario_inspermons_faceis)

            elif chance_3 in chance_3_medio: # Chance de inspermon medio
                nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicion
ario_inspermons_medios)

            else: # Chance de inspermon dificil
                nome_oponente, ataque_oponente, defesa_oponente, vida_oponente = escolha_adversã; rio(dicion
ario_inspermons_dificeis)

        print("Voce esta batalhando contra {}".format(nome_oponente))
        time.sleep(1)

        print("Este Inspermã³n possui os seguintes atributos: Ataque: {}, Defesa: {} e Vida: {}".format(ataque
_oponente, defesa_oponente, vida_oponente))
        time.sleep(1)

        if nome_oponente not in lista_nomes: #Verifica se o oponente ja esta na insperdex e o adiciona em caso
negativo

```

```
lista_nomes.append(nome_oponente)
```

```
# ===== Batalha =====
    while True:
        while True:
            x=2
            if evolucao == 1: #Caracteriza  o dos ganhos para inspermon inicial
                xp_ganho=10
                dinheiro_ganho = 2

            elif evolucao == 2: #Caracteriza  o dos ganhos para primeira evolu  o
                if chance_2 == 1:#inspermon facil
                    xp_ganho = 10
                    dinheiro_ganho = 2
                else:
                    xp_ganho = 20#inspermon medio
                    dinheiro_ganho = 4

            elif evolucao == 3: #Caracteriza  o dos ganhos para segunda evolu  o
                if chance_3 == 1: #inspermon lendario
                    xp_ganho = 100
                    dinheiro_ganho = 20
                elif chance_3 in chance_3_facil:#inspermon facil
                    xp_ganho = 10
                    dinheiro_ganho = 2
                elif chance_3 in chance_3_medio:#inspermon medio
                    xp_ganho = 20
                    dinheiro_ganho = 4
                else:
                    xp_ganho = 30 #inspermon dificil
                    dinheiro_ganho = 6

        atacar ou fugir

        a  o=input("O que voc   deseja fazer? (Atacar ou Fugir):\n ")    # Pergunta se deseja

        if a  o == "fugir":
            chancefugir=random.randint(1,2)    # Sorte de fuga

            if chancefugir==2:
                print("Voc   conseguiu fugir!")    # Fuga com sucesso
                time.sleep(0.5)
                x=1 # X
                break

            else:
```

```

        print("Você não conseguiu fugir!") # Fuga sem sucesso
        time.sleep(0.5)
        x=2
        break

    if aÃ§Ã£o == "atacar": # Ataque
        print("1- Ataque normal, 2- Ataque de sorte ")
        time.sleep(0.5)
        ataque_ativo=int(input("Qual ataque deseja usar? 1 ou 2: \n ")) # Escolha do
ataque a ser utilizado

        critico=random.randint(1,5) # Chance de ataque critico

        if ataque_ativo==1:

            if critico==1:

                ataque1=ataque1*1.5
                print("ATAQUE CRÍTICO".format(nome))
                time.sleep(0.5)
                critico=random.randint(1,5)

                if vida_oponente>0 and ataque1-defesa_oponente>=0: # RealizaÃ
vida_oponente= batalha(vida_oponente,ataque1,defesa_op
onente)

                if vida_oponente>0: #Verifica se o oponente tem vida
e atualiza caso exista dano
                    print("{} deu {} de dano em {}, agora ele tem
{} de vida".format(nome,ataque1-defesa_oponente,nome_oponente,vida_oponente)) # Cálculo da vida restante
                    time.sleep(0.5)
                    ataque1=ataque1/1.5
                    break

                else: # Fim da batalha, vitória
                    print("{} deu {} de dano em {}, ele desmaiou".
format(nome,ataque1-defesa_oponente,nome_oponente)) # Cálculo da vida restante
                    time.sleep(0.5)
                    print("Voce derrotou {}!".format(nome_oponent
e))

                    time.sleep(0.5)
                    xp=funcaoxp(xp,xp_ganho) # AtualizaÃ§Ã£o da
experiência
                    print("Voce recebeu {} de experiencia, agora v
ocê tem {}".format(xp_ganho,xp))

                    time.sleep(0.5)
                    dinheiro = dinheiro + dinheiro_ganho
                    print("Voce recebeu {} Inspermoedas! Agora seu
saldo é de {} Inspermoedas".format(dinheiro_ganho,dinheiro))

```

```

        ataque1=ataque1/1.5
        x=3
        break

    if critico!=1 and ataque1-defesa_oponente>0:
        if vida_oponente>0: # Realiza o ataque e atualiza a vida
da vida restante
                                vida_oponente= batalha(vida_oponente,ataque1,defesa_op
onente)

                                if vida_oponente>0: #Verifica se o oponente tem vida
e mostra o dano causado
                                    print("{} deu {} de dano em {}, agora ele tem
{} de vida".format(nome,ataque1-defesa_oponente,nome_oponente,vida_oponente)) # Cálculo da vida restante
                                    time.sleep(0.5)
                                    break

                                else: # Fim da batalha, vitória
                                    print("{} deu {} de dano em {}, ele desmaiou".
format(nome,ataque1-defesa_oponente,nome_oponente)) # Cálculo da vida restante
                                    time.sleep(0.5)
                                    print("Voce derrotou {}!".format(nome_oponent
e))

                                    time.sleep(0.5)
                                    xp=funcaoexp(xp,xp_ganho) # Atualiza a xp da
experiência e dinheiro
                                    print("Voce recebeu {} de experiencia, agora v
ocê tem {}".format(xp_ganho,xp))

                                    time.sleep(0.5)
                                    dinheiro = dinheiro + dinheiro_ganho
                                    print("Voce recebeu {} Inspermoedas! Agora seu
saldo é de {} Inspermoedas".format(dinheiro_ganho,dinheiro))

                                    x=3
                                    break

                                else:
                                    print("{} não deu dano em {}, ele continua com {} de vida".fo
rmat(nome,nome_oponente,vida_oponente))

                                    break

if ataque_ativo == 2: # Ataque 2 valor de ataque aleatório
    ataque2=random.randint(1,10)

    if critico==1: #Caso seja um ataque crítico
        ataque2=ataque2*1.5
        print("ATAQUE CRÍTICO".format(nome))
        time.sleep(0.5)
        critico=random.randint(1,5)

```

```

liza a vida restante

onente)

sou dano e da o resultado em caso afirmativo
ele tem {} de vida".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente))

sou dano e avisa caso não tenha causado
tinua com {} de vida".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente))

format(nome,ataque2-defesa_oponente,nome_oponente)) # Cálculo da vida restante
e))

experiência e dinheiro
ocia tem {}".format(xp_ganho,xp))

saldo de {} Inspermoedas".format(dinheiro_ganho,dinheiro))

if critico!=1:
    if ataque2-defesa_oponente>0 : # Realiza o ataque e atua
        vida_oponente= batalha(vida_oponente,ataque2,defesa_oponente)

        if vida_oponente>0: #Verifica se o oponente tem vida
            if ataque2-defesa_oponente>0: #verifica se causou
                print("{} deu {} de dano em {}, agora ele tem {} de vida".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente))
                # Cálculo da vida restante
                time.sleep(0.5)
                ataque2=ataque2/1.5
                break
            if ataque2-defesa_oponente<=0:#verifica se causou
                print("{} não deu dano em {}, ele continua com {} de vida".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente))
                # Cálculo da vida restante
                time.sleep(0.5)
                ataque2=ataque2/1.5
                break

        else:
            # Fim da batalha, vitória
            print("{} deu {} de dano em {}, ele desmaiou".format(nome,ataque2-defesa_oponente,nome_oponente)) # Cálculo da vida restante
            time.sleep(0.5)
            print("Voce derrotou {}!".format(nome_oponente))
            time.sleep(0.5)
            xp=funcao(xp,xp_ganho) # Atualiza a vida
            print("Voce recebeu {} de experiencia, agora verifique o saldo de {} Inspermoedas".format(dinheiro_ganho,dinheiro))
            time.sleep(0.5)
            dinheiro = dinheiro + dinheiro_ganho
            print("Voce recebeu {} Inspermoedas! Agora seu saldo é de {} Inspermoedas".format(dinheiro_ganho,dinheiro))
            ataque2=ataque2/1.5
            x=3
            break

    if critico!=1:
        if ataque2-defesa_oponente>0 : # Realiza o ataque e atua
            vida_oponente= batalha(vida_oponente,ataque2,defesa_oponente)

            if vida_oponente>0:
                print("{} deu {} de dano em {}, agora ele tem {} de vida".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente))
                # Cálculo da vida restante
                time.sleep(0.5)
                ataque2=ataque2/1.5
                break
            else:
                # Fim da batalha, vitória
                print("{} deu {} de dano em {}, ele desmaiou".format(nome,ataque2-defesa_oponente,nome_oponente))
                time.sleep(0.5)
                print("Voce derrotou {}!".format(nome_oponente))
                time.sleep(0.5)
                xp=funcao(xp,xp_ganho)
                print("Voce recebeu {} de experiencia, agora verifique o saldo de {} Inspermoedas".format(dinheiro_ganho,dinheiro))
                time.sleep(0.5)
                dinheiro = dinheiro + dinheiro_ganho
                print("Voce recebeu {} Inspermoedas! Agora seu saldo é de {} Inspermoedas".format(dinheiro_ganho,dinheiro))
                ataque2=ataque2/1.5
                x=3
                break

```

```

{} de vida".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente)) # Cálculo da vida restante
time.sleep(0.5)
break

else: # Fim da batalha, vitória
    print("{} deu {} de dano em {}, ele desmaiou".
format(nome,ataque2-defesa_oponente,nome_oponente)) # Cálculo da vida restante
time.sleep(0.5)
print("Voce derrotou {}!".format(nome_oponent
e))

time.sleep(0.5)
xp=funcaoxp(xp,xp_ganho) # Atualização da
experiência
print("Voce recebeu {} de experiencia, agora v
ocã tem {}".format(xp_ganho,xp))

time.sleep(0.5)
dinheiro = dinheiro + dinheiro_ganho
print("Voce recebeu {} Inspermoedas! Agora seu
saldo de {} Inspermoedas".format(dinheiro_ganho,dinheiro))

x=3
break

if critico==1: # Caso seja um ataque crítico
    ataque2=ataque2*1.5
    critico=random.randint(1,5)

if ataque2-defesa_oponente>0: # Verifica se o inspermon causou dano n
o adversário

    vida_oponente= batalha(vida_oponente,ataque2,defesa_oponente)

    if vida_oponente>0: # Realização do ataque e atualização
da vida restante
        print("{} deu {} de dano em {}, agora ele tem {} de vi
da".format(nome,ataque2-defesa_oponente,nome_oponente,vida_oponente)) # Cálculo da vida restante
        time.sleep(0.5)
        break

if ataque2-defesa_oponente<=0: # Notificação de não ter dado dano a
lgum
    print("{} não deu dano em {}, ele continua com {} de vida".fo
rmat(nome,nome_oponente,vida_oponente))

    time.sleep(0.5)
    break

if vida_oponente<=0: # Derrota do oponente
    print("{} deu {} de dano em {}, ele desmaiou".format(nome,ataq
uel-defesa_oponente,nome_oponente)) # Cálculo da vida restante

```



```

        time.sleep(0.5)
        print("Voce derrotou {}!" .format(nome_oponente))
        time.sleep(0.8)
        xp=funcaoexp(xp,xp_ganho)    # Atualiza o nível da experiência

        print("Voce recebeu {} de experiencia, agora você tem {}".for

mat(xp_ganho,xp))

        time.sleep(0.5)
        dinheiro = dinheiro + dinheiro_ganho
        print("Voce recebeu {} Inspermoedas! Agora seu saldo é de {}

Inspermoedas".format(dinheiro_ganho,dinheiro))

        x=3
        break

    else: # Erro de digitação
        print("Digite um ataque válido!")
        time.sleep(0.5)
        continue

    else:
        print("Digite um comando válido!")
        time.sleep(0.5)
        continue

    if x==2: # X para fuga sem sucesso
        if vida_oponente>0:
            time.sleep(0.5)
            print("Ataque a vez do seu oponente!")
            time.sleep(0.5)

            critico_oponente = random.randint(0,5)

            if critico_oponente==1: #Ataque crítico do oponente
                print("ATAQUE CRÍTICO")
                time.sleep(0.5)
                ataque_oponente=ataque_oponente*1.5

                if ataque_oponente-defesa>0: #verifica se o oponente causou dano
                    vida= batalha(vida, ataque_oponente, defesa)

                    if vida>0: # se o inspermon tem vida, atualiza a vida restante
                        print("{} deu {} de dano em {}! Agora ele tem {} de vi

da".format(nome_oponente,ataque_oponente-defesa,nome,vida))

                        time.sleep(0.5)
                        ataque_oponente=ataque_oponente/1.5
                        continue

                    if vida<=0: # se o inspermon ficou sem vida
                        print("Seu Inspermon levou {} de dano, desmaiou e foi

```

```

levado para o InsperCenter".format(ataque_oponente-defesa))

time.sleep(0.5)
vida=vida_inicial
ataque_oponente=ataque_oponente/1.5
break

if ataque_oponente-defesa<=0: # caso não tenha levado dano
    print("Seu Inspermão não levou dano")
    ataque_oponente=ataque_oponente/1.5
    continue

if critico_oponente!=1: #caso não seja um ataque crítico
    if ataque_oponente-defesa>0: #verifica se causou dano
        vida= batalha(vida, ataque_oponente, defesa)

        if vida>0: #caso o inspermon ainda tenha vida
            print("{} deu {} de dano em {}! Agora ele tem {} de vi
da".format(nome_oponente,ataque_oponente-defesa,nome,vida))

            time.sleep(0.5)
            continue

        if vida<=0: # caso o inspermon tenha desmaiado
            print("Seu Inspermão levou {} de dano, desmaiou e foi
levado para o InsperCenter".format(ataque_oponente-defesa))

            time.sleep(0.5)
            vida=vida_inicial
            break

        if ataque_oponente-defesa<=0: # se não houve dano no inspermon
            print("Seu Inspermão não levou dano")
            continue

    if vida<=0: # Derrota do seu inspermon
        print("Seu Inspermão desmaiou e foi levado para o InsperCenter")
        time.sleep(0.5)
        vida=vida_inicial
        break

elif x==1 or x==3: # X para fuga com sucesso/parada do loop de batalha
    x=2
    break

# ===== Evolução =====

time.sleep(0.5)
print("Seu Inspermão tem os seguintes atributos: Ataque:{}, Defesa:{}, Vida:{}".format(ataque1,defesa
,vida))

```

```

if evolucao == 1 and xp >= 200: #Checa se é a primeira evolução do inspermon e se ele tem a experi
    evolucao = 2

    if index == 1: #Caso o sharmander tenha sido a escolha inicial
        ataquel,defesa,vida_inicial,nome=funcao_evolucao(nome,"Sharmander","Sharmilion",ataque_ex
        nome = MudaNome(nome)
        lista_nomes.append("Sharmilion")

    if index == 2: # Caso o skuiurtle tenha sido a escolha inicial
        ataquel,defesa,vida_inicial,nome=funcao_evolucao(nome,"Skuiurtle","Uarturtle",ataque_ex
        nome = MudaNome(nome)
        lista_nomes.append("Uarturtle")

    if index == 3: # Caso o bulbatauro tenha sido a escolha inicial
        ataquel,defesa,vida_inicial,nome=funcao_evolucao(nome,"Bulbatauro","Ivitauro",ataque_e
        nome = MudaNome(nome)
        lista_nomes.append("Ivitauro")

elif evolucao == 2 and xp >= 500:

    evolucao = 3 #Checa se é a segunda evolução do inspermon e se ele tem a experi
    if index == 1: # Caso o sharmander tenha sido a escolha inicial
        ataquel,defesa,vida_inicial,nome=funcao_evolucao(nome,"Sharmilion","Cheirazard",ataque_ex
        nome = MudaNome(nome)
        lista_nomes.append("Cheirazard")
    if index == 2: # Caso o skuiurtle tenha sido a escolha inicial
        ataquel,defesa,vida_inicial,nome=funcao_evolucao(nome,"Uarturtle","Blastoide",ataque_e
        nome = MudaNome(nome)
        lista_nomes.append("Blastoise")
    if index == 3: # Caso o bulbatauro tenha sido a escolha inicial
        ataquel,defesa,vida_inicial,nome=funcao_evolucao(nome,"Bulbatauro","Venustauro",ataque_ex
        nome = MudaNome(nome)
        lista_nomes.append("Venustauro")

    continue

else:
    print("Digite um comando válido.")
    time.sleep(0.5)

```

```
# ===== Salvando o jogo =====
=

    while True:
        time.sleep(0.5)
        salvar_jogo=input("Deseja salvar o jogo? \n ") # Pergunta se deseja salvar o jogo
        salvar_jogo=salvar_jogo.lower()

        if salvar_jogo in sim: # Caso o usu rio queira salvar o jogo
            dados= open("jogo_salvo",'wb')
            pickle.dump({"Inspermon" : [nome, ataque1, defesa, vida_inicial, xp, dinheiro, lista_nomes, ataque_ext
ra, defesa_extra, vida_extra, evolucao]}, dados)
            dados.close()
            time.sleep(0.5)
            print("Jogo salvo com sucesso!")
            break

        if salvar_jogo in nao: # Caso n o queira salvar o jogo
            break

        else:
            print("Digite um comando v lido (Sim ou N o)")
            continue

#FINAL

    print("At  a proxima!")          # Fim do jogo
    break

# ===== FIM =====
```