

# RA1

## Tarea 1

### Corrección de código

```
平淡无奇的文本框显示了main.py的内容，包含从1到8行的Python代码，用于将分钟数转换为小时和分钟。
```

## Tarea 2

```
平淡无奇的文本框显示了tarea2.py的内容，包含从1到37行的Python代码，用于处理用户输入的整数、浮点数和字符串，并检查它们的类型。
```

## RA2

### Tarea 1

#### Juego ataque por turno

```
Exámen > RA2 > tarea1.py > ...
22     import random
23
24     vida = 3
25     puntos = 0
26     nvl = 0
27     print("- -- Partida comenzada -- -")
28     print("Tienes ", vida, " vidas, nivel ", nvl, " y ", puntos, " puntos")
29
30     cont = 0
31
32     while(vida != 0):
33         # Menú
34         print("=====")
35         print ("1. Fuerza")
36         print ("2. Precisión")
37         print ("3. Riesgo")
38         ataque = input("Atacar (introduce dígito): ")
39         ataque = int(ataque)
40
41         if (ataque == 1):
42             jug = random.randrange(5, 11) # Num entre 5 y 10
43             ord = random.randrange(3, 11) # Num entre 3 y 10
44         elif (ataque == 2):
45             jug = random.randrange(3, 9) # Num entre 3 y 8
46             ord = random.randrange(2, 10) # Num entre 2 y 9
47         elif (ataque == 3):
48             jug = random.randrange(1, 11) # Num entre 1 y 10
49             ord = random.randrange(1, 9) # Num entre 1 y 8
50         else:
51             print("Ataque introducido inválido")
52
53         if (ataque > 0 and ataque < 4):
54             print("-----")
55             print("Jugador: ", jug)
56             print("Enemigo: ", ord)
57             if ( jug > ord):
58                 puntos += 1
```

```
59     print("+ 1 punto")
60
61     cont += 1
62     if ( cont == 3 ):
63         cont = 0
64         nvl += 1
65         if ( vida < 3 ):
66             vida += 1
67             print("+ 1 nivel")
68             print("+ 1 vida")
69     elif ( jug < ord ):
70         vida -= 1
71         print("- 1 vida")
72
73     print("=====")
74     print("Juego finalizado")
75     print("Usted ha alcanzado el nivel ", nvl, " con ", puntos, " puntos")
76     print("=====")
```

## RA3

main

```
Exámen > RA3 > main.py > ...
13  from batalla import jugador
14  from batalla import enemigo
15
16  print("====")
17  print(" - -- Introducir datos -- - ")
18  jug = input("Nombre del jugador: ")
19  conocJug = 0
20  enerJug = 0
21
22  # Bucle repite si conocJug menor de 1 y mayor de 10, y, enerJug menor de 1 y mayor de 5
23  while ( (conocJug < 1 or conocJug > 10) or (enerJug < 1 or enerJug > 5) ):
24      conocJug = input("Conocimiento del jugador (entre 1 - 10): ")
25      enerJug = input("Energía del jugador (entre 1 - 5): ")
26      try:
27          conocJug = int(conocJug)
28          enerJug = int(enerJug)
29
30          if ((conocJug < 1 or conocJug > 10) or (enerJug < 1 or enerJug > 5)):
31              print("Datos incorrectos")
32
33      except (ValueError):
34          print("Valores no numéricos introducidos")
35          conocJug = 0
36          enerJug = 0
37
38  print()
39  print(" - -- Jugador generado -- -")
40  jugador.mostrar_jugador(jug, conocJug, enerJug)
41
42
43  print("-----")
44
45  print("Generando enemigo...")
46  ene, conocEne, enerEne = enemigo.generar_enemigo()
47  print()
48  print(" - -- Enemigo generado -- -")
49  jugador.mostrar_jugador(ene, conocEne, enerEne)
```

Ln 67, Col 54 Spaces: 4 UTF-8 CRLF { } Python

```
51 print()
52 print("====")
53 for x in range(3): # Empieza por 0 y termina en 2
54     print("-----")
55     ataqJug = jugador.ataque_jugador(conocJug, enerJug)
56     print("Jugador ataca: ", ataqJug)
57
58     ataqEne = enemigo.ataque_enemigo(conocEne, enerEne)
59     print("Enemigo ataca: ", ataqEne)
60
61     if ( ataqJug > ataqEne ):
62         enerEne -= 1
63     elif ( ataqJug < ataqEne ):
64         enerJug -= 1
65
66     print("-----")
67 print["===="]
68
69 if ( enerJug > enerEne ):
70     print("Gana ", jug, " con ", enerJug, " de energía")
71     print("Pierde ", ene, " con ", enerEne, " de energía")
72 elif ( enerJug < enerEne ):
73     print("Gana ", ene, " con ", enerEne, " de energía")
74     print("Pierde ", jug, " con ", enerJug, " de energía")
75 else:
76     print("Empatados")
77     print("Energía ", enerJug)
78
79 print("=====")
```

enemigo

```
8 import random
9
10 def generar_enemigo():
11     nombres = ["Hydra", "Kraken", "Minotauro", "Gorgona", "Titán"]
12     nombre = random.choice(nombres)
13
14     conocimiento = random.randrange(1, 11) # Entre 1 y 10
15     energia = random.randrange(1, 6) # Entre 1 y 5
16
17     return nombre, conocimiento, energia
18
19 def ataque_enemigo(conocimiento, energia):
20     return (conocimiento*energia)*(random.randrange(1, 4))
21
22 def mostrar_enemigo(nombre, conocimiento, energia):
23     print("Nombre: ", nombre)
24     print("Conocimiento: ", conocimiento)
25     print("Energia: ", energia)
```

jugador

```
5 import random
6
7 def ataque_jugador(conocimiento, energia):
8     return (conocimiento*energia)*(random.randrange(1, 4))
9
10 def mostrar_jugador(nombre, conocimiento, energia):
11     print("Nombre: ", nombre)
12     print("Conocimiento: ", conocimiento)
13     print("Energia: ", energia)
```

init

```
平淡 _init_.py 2, U X  
Exámen > RA3 > batalla > 平淡 _init_.py > ...  
1 print("Inicializando mi_paquete...")  
2  
3 from .jugador import ataque_jugador, mostrar_jugador  
4 from .enemigo import generar_enemigo, ataque_enemigo, mostrar_enemigo  
5  
6 # Controlar lo que se importa si hacemos: from matemáticas import *  
7 __all__ = [  
8     "ataque_jugador", "mostrar_jugador",  
9     "generar_enemigo", "ataque_enemigo", "mostrar_enemigo"  
10 ]  
11  
12 ## No me funciona, no usaré el init
```