

## RA1

1)

```
minutos = (int)(input("Introduce una cantidad de minutos: "))

horas = minutos // 60
resto = minutos % 60

print("Equivale a",horas,"horas y",resto,"minutos")
```

2)

```
entero = (int)(input("Introduce un número entero: "))
decimal = (float)(input("Introduce un número decimal: "))
cadena = input("Introduce una cadena de texto: ")

print(entero, type(entero))
print(decimal, type(decimal))
print(cadena, type(cadena))

es_entero = isinstance(entero, int)
es_decimal = isinstance(decimal, float)
es_cadena = isinstance(cadena, str)

comprobaciones = es_entero and es_decimal and es_cadena
print(comprobaciones)
```

## RA2

1)

```
import random

vida_jugador = int(3)
puntos_jugador = int(0)
nivel_jugador = int(1)
puntos_nivel = int(0)

while (vida_jugador > 0):
    try:
        #Impresion de las estadisticas actuales del jugador
        print("Vida actual:",vida_jugador)
        print("Nivel actual:",nivel_jugador)
        print("Experiencia actual:",puntos_nivel,"/3")

        #Control del tipo de ataque, respecto a ello sacar la carta respectiva del jugador y del enemigo
        print("Elige un ataque: \n -Fuerza 1 \n -Precisión 2 \n -Riesgo 3")
        ataque = int(input(""))

        match ataque:
            case 1:
                carta_jugador = random.randint(5,10)
                carta_enemigo = random.randint(3,10)
                tipo_ataque = "Fuerza"
            case 2:
                carta_jugador = random.randint(3,8)
                carta_enemigo = random.randint(2,9)
                tipo_ataque = "Precision"
            case 3:
                carta_jugador = random.randint(1,10)
                carta_enemigo = random.randint(1,8)
                tipo_ataque = "Riesgo"
            case _:
                raise ValueError("El valor debe estar entre 1 y 3.")

        #Impresion de la jugada, la carta del jugador, la carta del enemigo, comprobación de quien ha ganado a quien y asignacion de puntos y vida
        print("")
        print("-----")
        print("Tu carta:",carta_jugador)
        print("Carta del enemigo:",carta_enemigo)
        if (carta_jugador > carta_enemigo):
            puntos_jugador = puntos_jugador+1
            puntos_nivel = puntos_nivel+1
            print("Has ganado al enemigo usando",tipo_ataque,".\nGanas un punto")
        elif (carta_jugador < carta_enemigo):
            vida_jugador = vida_jugador-1
            print("Has perdido contra el enemigo usando",tipo_ataque,".\nPierdes una vida")
        else:
            print("Has quedado empate con el enemigo, por lo tanto no ocurre nada")
        print("-----")

        #Control de niveles del jugador, por cada 3 puntos sube un nivel, recupera vida siempre que no tenga el maximo
        #y se reinicia el contador de puntos nivel.
        if (puntos_nivel == 3):
            puntos_nivel = 0
            print("Has subido de nivel")
            nivel_jugador = nivel_jugador+1
        if (vida_jugador < 3):
            vida_jugador = vida_jugador+1
            print("Recuperas vida por subida de nivel")

    except ValueError as e:
        print("Error:", e)

#Impresion final, despues de haber perdido todas las vidas
print("")
print("-----")
print("Has perdido todas las vidas")
print("Puntuación final:",puntos_jugador)
print("Nivel alcanzado:",nivel_jugador)
print("-----")
```

## RA3

1)

```
from batalla import (
    ataque_jugador, mostrar_jugador,
    generar_enemigo, ataque_enemigo, mostrar_enemigo
)

print("== BATALLA DE MAGIA ==")
nombre_jugador = str(input("Introduce tu nombre:"))
conocimiento_jugador = int(input("Introduce tu conocimiento (1-10):"))
energia_jugador = int(input("Introduce tu energía (1-5):"))
ronda = 1

nombre_enemigo, conocimiento_enemigo, energia_enemigo = generar_enemigo()

mostrar_jugador(nombre_jugador, conocimiento_jugador, energia_jugador)
mostrar_enemigo(nombre_enemigo, conocimiento_enemigo, energia_enemigo)

while (ronda < 4):
    magia_jugador = ataque_jugador(conocimiento_jugador, energia_jugador)
    magia_enemigo = ataque_enemigo(conocimiento_enemigo, energia_enemigo)

    print(f"--- RONDA{ronda} ---")
    print(f"Magia de {nombre_jugador}: {magia_jugador}")
    print(f"Magia de {nombre_enemigo}: {magia_enemigo}")

    if (magia_jugador > magia_enemigo):
        print(f"{nombre_jugador} lanza un poderoso hechizo (-1 de energía enemigo)")
        energia_enemigo = energia_enemigo - 1
    elif (magia_jugador < magia_enemigo):
        print(f"{nombre_enemigo} lanza un poderoso hechizo (-1 de energía jugador)")
        energia_jugador = energia_jugador - 1
    else:
        print("Ambos han lanzado un hechizo de la misma potencia, por lo tanto nadie pierde energía")
        print("")

    ronda = ronda + 1

print("== FIN DEL DUELO ==")
if (energia_jugador > energia_enemigo):
    print(f"{nombre_jugador} ha vencido a {nombre_enemigo}")
elif (energia_jugador < energia_enemigo):
    print(f"{nombre_enemigo} ha vencido a {nombre_jugador}")
else:
    print(f"{nombre_jugador} y {nombre_enemigo} han quedado empate")
```

```
# __init__.py

from .jugador import ataque_jugador, mostrar_jugador
from .enemigo import generar_enemigo, ataque_enemigo, mostrar_enemigo

__all__ = [
    "ataque_jugador", "mostrar_jugador",
    "generar_enemigo", "ataque_enemigo", "mostrar_enemigo",
]
```

```
import random

def ataque_jugador(conocimiento, energia):
    return (conocimiento * energia) * random.randint(1,3)

def mostrar_jugador(nombre, conocimiento, energia):
    print("-----")
    print("Jugador:", nombre)
    print(" -Conocimiento:", conocimiento)
    print(" -Energia", energia)
    print("-----")
    print("")
```

```
import random

def generar_enemigo():
    nombres = ["Hydra", "Kraken", "Minotauro", "Gorgona", "Titán"]
    nombre = random.choice(nombres)
    conocimiento = random.randint(1,10)
    energia = random.randint(1,5)
    return nombre, conocimiento, energia

def ataque_enemigo(conocimiento, energia):
    return (conocimiento * energia) * random.randint(1,3)
    *
def mostrar_enemigo(nombre, conocimiento, energia):
    print("-----")
    print("Enemigo:", nombre)
    print(" -Conocimiento:", conocimiento)
    print(" -Energia", energia)
    print("-----")
    print("")
```