

In [43]:

```
1 #codigo de un constructor de pantalla de inicio de un juego
2
3 # creo un diccionario con todos los posibles personajes que puede tener mi j
4
5 #diccionario={"key1":"valor1","key2":"valor2"}
6
7 personajes = {
8     "raza":{
9         "humanos":{
10             "clase":{
11                 "gerreros":{
12                     "stats":{
13                         "vida":"1200",
14                         "mana":"850"
15                     }
16                 },
17                 "cazador":{
18                     "stats":{
19                         "vida":"800",
20                         "mana":"900"
21                     }
22                 },
23                 "druida":{
24                     "stats":{
25                         "vida":"1000",
26                         "mana":"1200"
27                     }
28                 }
29             }
30         },
31     },
32     "orcos":{
33         "clase":{
34             "gerreros":{
35                 "stats":{
36                     "vida":"1200",
37                     "mana":"850"
38                 }
39             },
40             "cazador":{
41                 "stats":{
42                     "vida":"800",
43                     "mana":"900"
44                 }
45             },
46             "chaman":{
47                 "stats":{
48                     "vida":"1000",
49                     "mana":"1200"
50                 }
51             }
52         }
53     }
54 }
55
56
```

```
57         }
58     }
59 }
60 }
61
62 #print(personajes["raza"]["humanos"]["clase"]["cazador"]["stats"]["mana"])
63
64 import uuid
65 import json
66 import time
67
68
69 class personaje():
70
71     def __init__(self,nombre,raza,clase,vida,mana):
72         self.nombre=nombre
73         self.raza=raza
74         self.clase=clase
75         self.vida=vida
76         self.mana=mana
77
78     def configurar_personaje(self):
79
80         print("Que raza quieres ser?: \n 1. humanos \n 2. orcos \n")
81         raza=input("\n")
82
83         if raza=="1" or raza=="humanos":
84             raza="humanos"
85
86         print("Que clase quieres ser?: \n 1. guerrero \n 2. cazador \n 3")
87         clase=input("\n")
88
89         if clase=="1" or clase=="guerrero":
90             clase="guerrero"
91
92         #aqui llamamos un metodo que cree al personaje
93         self.crear_personaje(raza,clase)
94
95         elif clase=="2" or clase=="cazador":
96             clase="cazador"
97
98         #aqui llamamos un metodo que cree al personaje
99         self.crear_personaje(raza,clase)
100
101         elif clase=="3" or clase=="druida":
102             clase="druida"
103
104         #aqui llamamos un metodo que cree al personaje
105         self.crear_personaje(raza,clase)
106
107         else:
108             print("clase no valida")
109             #control de errores
110
111         elif raza=="2" or raza=="orcos":
112             raza="orcos"
113
```

```

114 print("Que clase quieres ser?: \n 1. guerrero \n 2. cazador \n 3
115 clase=input("\n")
116
117 if clase=="1" or clase=="guerrero":
118     clase="guerrero"
119
120     #aqui llamamos un metodo que cree al personaje
121     self.crear_personaje(raza,clase)
122
123 elif clase=="2" or clase=="cazador":
124     clase="cazador"
125
126     #aqui llamamos un metodo que cree al personaje
127     self.crear_personaje(raza,clase)
128
129 elif clase=="3" or clase=="chaman":
130     clase="chaman"
131
132     #aqui llamamos un metodo que cree al personaje
133     self.crear_personaje(raza,clase)
134
135 else:
136     print("clase no valida")
137     #control de errores
138
139 else:
140     print("raza no valida")
141     #control de errores
142
143 def crear_personaje(self,raza,clase):
144     #accedo al diccionario
145     vida=personajes["raza"][raza]["clase"][clase]["stats"]["vida"]
146     mana=personajes["raza"][raza]["clase"][clase]["stats"]["mana"]
147
148     nombre = input("Adalid! , como deseas ser llamado : \n")
149     #llamo al constructor
150     nuevo_pj=personaje(nombre=nombre,raza=raza,clase=clase,vida=vida,mana=mana)
151
152     #creo un diccionario para guardar la info de mi personaje creado
153     datos={
154         "id":str(uuid.uuid1()),
155         "nombre":nuevo_pj.nombre,
156         "raza":nuevo_pj.raza,
157         "clase":nuevo_pj.clase,
158         "vida":nuevo_pj.vida,
159         "mana":nuevo_pj.mana
160     }
161
162     #aqui tenemos que llamar un metodo para guardar al personaje
163     print("El personaje {} ha sido creado y guardado satisfactoriamente"
164
165     pj=personaje("Shigueru","humano","druida","1000","1200")
166     pj.configurar_personaje()
167
168
169

```

Que raza quieres ser?:

1. humanos
2. orcos

humanos

Que clase quieres ser?:

1. guerrero
2. cazador
3. druida

3

Adalid! , como deseas ser llamado :

shigueru

El personaje shigueru ha sido creado y guardado satisfactoriamente

In [18]:

```
1 import uuid
2
3 u1 = uuid.uuid1() #me otorga un codigo unico de identificacion
4 u2 = uuid.uuid1()
5 u3 = uuid.uuid1()
6 print(u1)
7 print(u2)
8 print(u3)
```

06f23768-2f09-11ed-aa32-cf4bf1144f19

06f23769-2f09-11ed-be73-cf4bf1144f19

06f2376a-2f09-11ed-a407-cf4bf1144f19

In [21]:

```
1 import json # es un formato para guardar datos, como archivo
2 print(json.dumps({'4': 5, '6': 7}, sort_keys=True, indent=4))
3 {
4     "4": 5,
5     "6": 7
6 }
```

```
{
  "4": 5,
  "6": 7
}
```

Out[21]: {'4': 5, '6': 7}

In [24]:

```
1 import time
2
3 while True:
4     localtime = time.localtime()
5     result = time.strftime("%I:%M:%S %p", localtime)
6     print(result)
7     time.sleep(1) # es un delay, un retraso temporal en la ejecucion del loop
```

07:11:34 PM

07:11:35 PM

07:11:36 PM

07:11:37 PM

07:11:38 PM

07:11:39 PM

KeyboardInterrupt

Traceback (most recent call last)

Input In [24], in <cell line: 4>()

5 result = time.strftime("%I:%M:%S %p", localtime)

6 print(result)

----> 7 time.sleep(1)

KeyboardInterrupt:

In [36]:

```
1 raza="elfo";
```