

# MANUAL DE USUARIO PROYECTO FINAL PROGRAMACIÓN DISTRIBUIDA

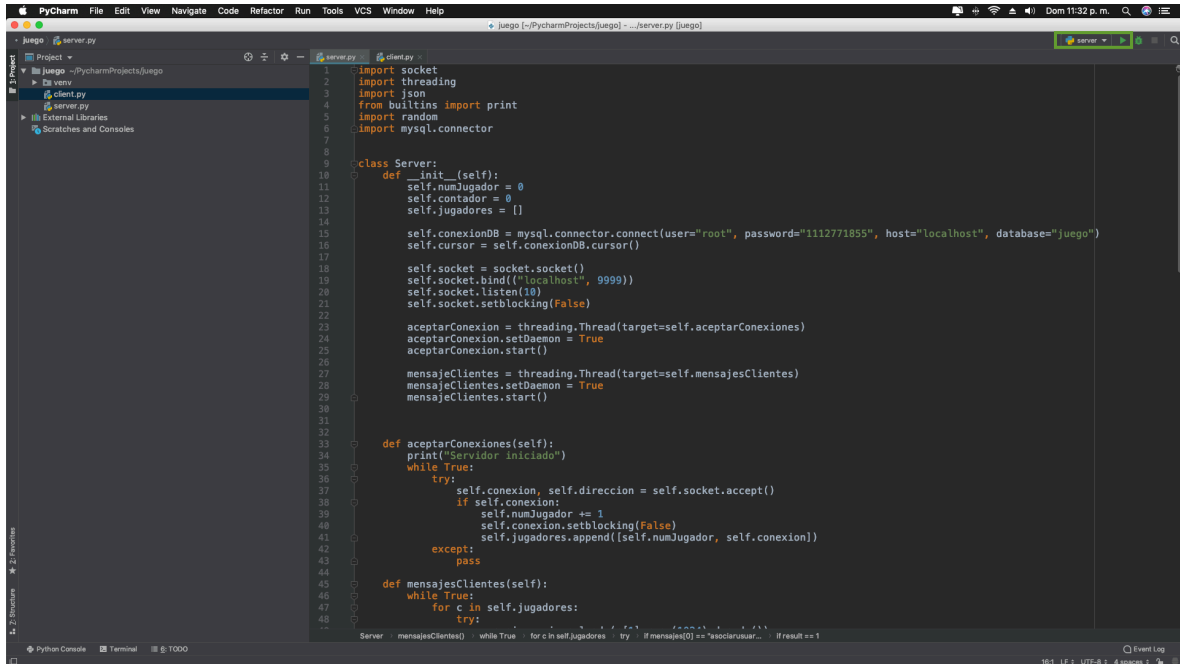
PRESENTADO POR: CRISTIAN RESTREPO ZULUAGA  
JAIRO ANDRES SALAZAR RUIZ  
ADRIAN VARGAS CASTRILLON

DIRIGIDO A: ING. CARLOS ALBERTO LONDOÑO LOAIZA

CORPORACIÓN DE ESTUDIOS TECNOLÓGICOS DEL NORTE DEL VALLE  
CARTAGO VALLE  
15-06-2019

# CONTENIDO

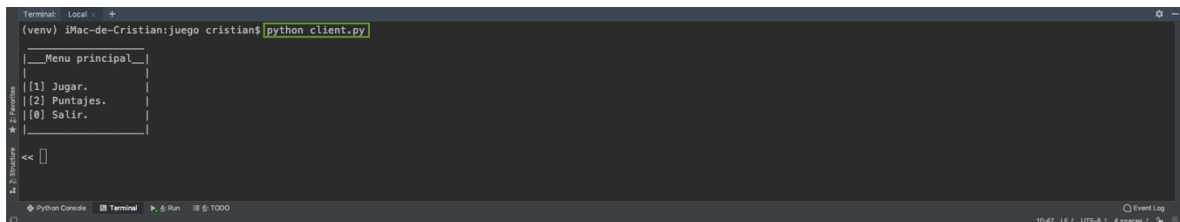
## 1. INICIO Y EJECUCION DEL SERVIDOR:



The screenshot shows the PyCharm IDE with the 'server.py' file open. The code defines a 'Server' class that initializes a MySQL database connection, sets up a socket listener on port 9999, and starts two threads: 'aceptarConexion' (accept connections) and 'mensajeClientes' (handle client messages). The 'aceptarConexion' thread prints 'Servidor iniciado' and enters a loop to accept connections, incrementing the number of players and adding them to a list. The 'mensajeClientes' thread handles messages for each player in the list.

```
1 import socket
2 import threading
3 import json
4 from builtins import print
5 import random
6 import mysql.connector
7
8
9 class Server:
10     def __init__(self):
11         self.numJugador = 0
12         self.contador = 0
13         self.jugadores = []
14
15         self.conexionDB = mysql.connector.connect(user="root", password="1112771855", host="localhost", database="juego")
16         self.cursor = self.conexionDB.cursor()
17
18         self.socket = socket.socket()
19         self.socket.bind(("localhost", 9999))
20         self.socket.listen(10)
21         self.socket.setblocking(False)
22
23         aceptarConexion = threading.Thread(target=self.aceptarConexiones)
24         aceptarConexion.setDaemon = True
25         aceptarConexion.start()
26
27         mensajeClientes = threading.Thread(target=self.mensajesClientes)
28         mensajeClientes.setDaemon = True
29         mensajeClientes.start()
30
31
32     def aceptarConexiones(self):
33         print("Servidor iniciado")
34         while True:
35             try:
36                 self.conexion, self.direccion = self.socket.accept()
37                 if self.conexion:
38                     self.numJugador += 1
39                     self.conexion.setblocking(False)
40                     self.jugadores.append((self.numJugador, self.conexion))
41             except:
42                 pass
43
44     def mensajesClientes(self):
45         while True:
46             for c in self.jugadores:
47                 try:
48                     # ... (code for handling client messages) ...
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

## 2. EJECUCION DE CLIENTE:



The screenshot shows a terminal window with the command 'python client.py' executed. The output displays a 'Menu principal' with three options: '[1] Jugar.', '[2] Puntajes.', and '[0] Salir.'. The cursor is positioned at the bottom, ready for user input.

```
(venv) iMac-de-Cristian:juego cristian$ python client.py
Menu principal
[1] Jugar.
[2] Puntajes.
[0] Salir.
<< 
```

### 3. INICIAR UNA PARTIDA (CLIENTE 1):

```
Terminal: Local +
(venv) iMac-de-Cristian:juego cristian$ python client.py

Menu principal
[1] Jugar.
[2] Puntajes.
[0] Salir.

<< 1
Usuario << cristian
```

```
Terminal: Local +
Esperando otro jugador
```

### 4. INICIAR UNA PARTIDA (CLIENTE 2):

```
Terminal: Local Local (2) +
(venv) iMac-de-Cristian:juego cristian$ python client.py

Menu principal
[1] Jugar.
[2] Puntajes.
[0] Salir.

<< 1
Usuario << jairo
```

### 5. PARTIDA INICIADA (ARRANCA JUGANDO EL CLIENTE 1 PARA ESTE EJEMPLO):

```
Terminal: Local Local (2) +
Iniciando partida [jugador cristian]
[Vida] = 100%
El jugador cristian tiene la ventaja.

Menu
[1] Atacar.
[0] Salir.

<< 1
```

### 6. PRIMER ATAQUE PERPETRADO (CLIENTE 1 CONTRA CLIENTE 2:

```
Terminal: Local Local (2) +
Ataque realizado, por favor espere su turno.
```

7. ES EL TURNO DEL CLIENTE 2: (SE OBSERVA QUE LOS PUNTOS DE VIDA DE CADA JUGADOR SE VAN DISMINUTENDO CON CADA ATAQUE REALIZADO)

```
Terminal: Local Local (2) +
Has recibido un ataque de 15[puntos] de fuerza.
[Vida] = 85%
Es tu turno.
Menu
[1] Atacar.
[0] Salir.
<< 1
```

8. CLIENTE 2 PERPETRA EL ATAQUE CONTRA EL CLIENTE 1:

```
Terminal: Local Local (2) +
Ataque realizado, por favor espere su turno.
```

9. SE OBSERVA QUE LOS PUNTOS DE VIDA DEL CLIENTE 1 VAN DISMINUYENDO A MEDIDA QUE SE REALIZAN ATAQUES EN SU CONTRA:

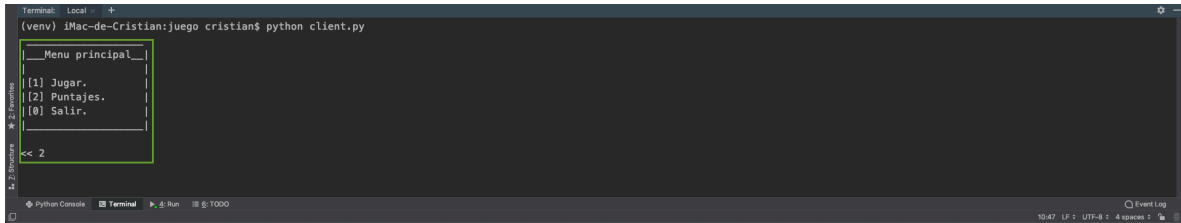
```
Terminal: Local Local (2) +
Has recibido un ataque de 16[puntos] de fuerza.
[Vida] = 84%
Es tu turno.
Menu
[1] Atacar.
[0] Salir.
<< 1
```

10. EL JUEGO CONCLUYE CUANDO UNO DE LOS DOS CLIENTES PIERDE TODOS LOS PUNTOS DE VIDA (PARA VOLVER A JUGAR CONTRA EL MISMO Oponente PRESIONE LA TECLA 2 O PARA SALIR PRESIONE LA TECLA 0):

```
Terminal: Local Local (2) +
[Vida] = 21%
cristian has ganado la partida.
Obtienes 20pts.
Menu
[2] Volver a jugar.
[0] Salir.
<<
```

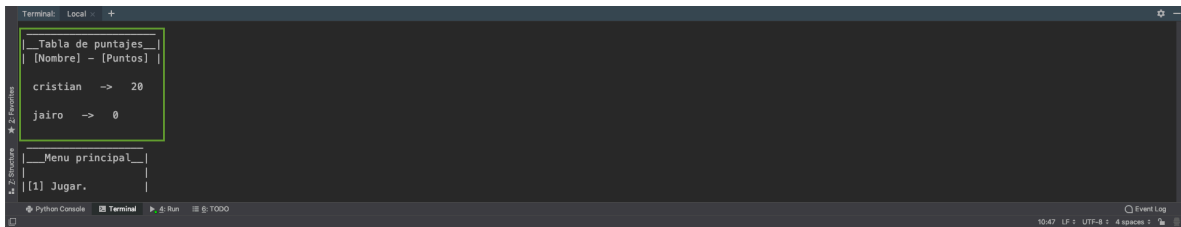
```
Terminal: Local Local (2) +
Has recibido un ataque de 24[puntos] de fuerza.
[Vida] = 0%
Has perdido la partida.
Menu
[2] Volver a jugar.
[0] Salir.
<<
```

## 11. REGISTRO DE JUGADORES Y VISUALIZACIÓN DE PUNTAJES GENERALES:



```
Terminal: Local +
(venv) iMac-de-Cristian:juego cristian$ python client.py

Menu principal
|
| [1] Jugar.
| [2] Puntajes.
| [0] Salir.
|
<< 2
```



```
Terminal: Local +

Tabla de puntajes
| (Nombre) - (Puntos) |
| cristian -> 20      |
| jairo -> 0          |

Menu principal
|
| [1] Jugar.
|
```