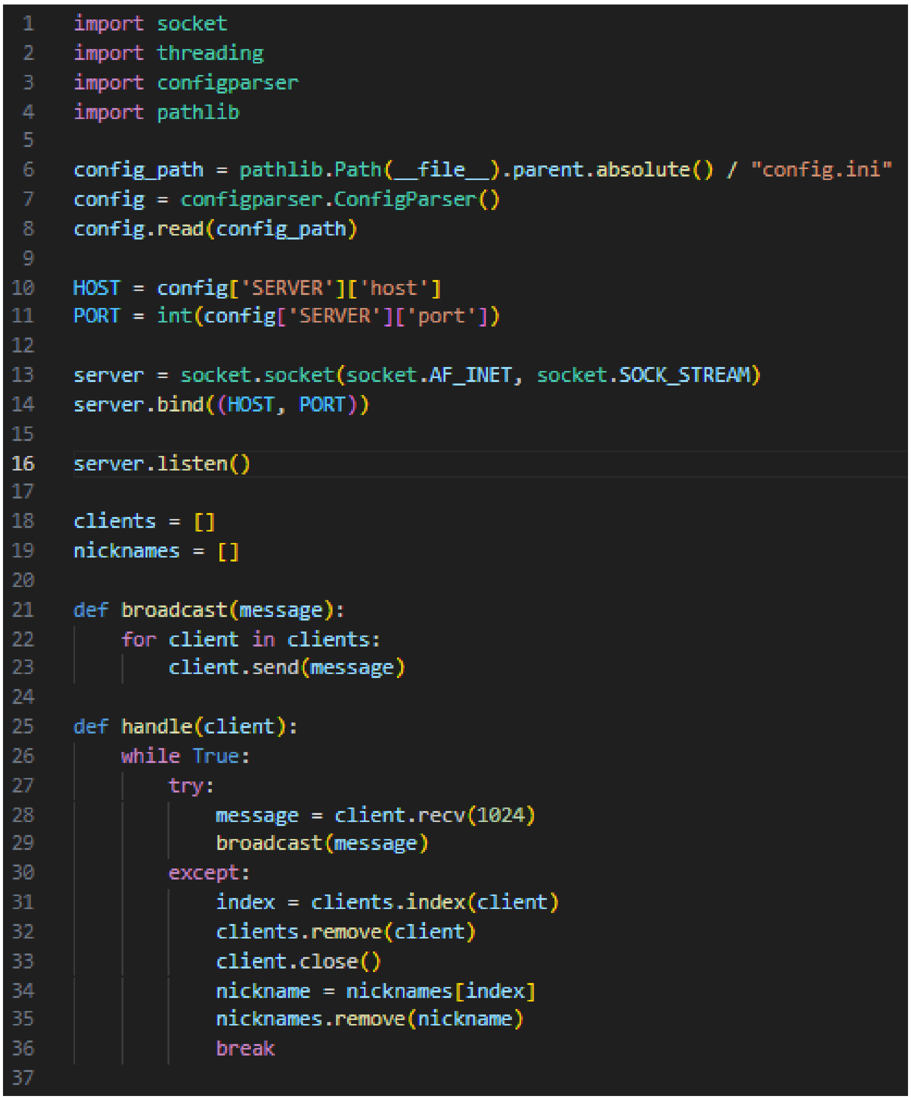
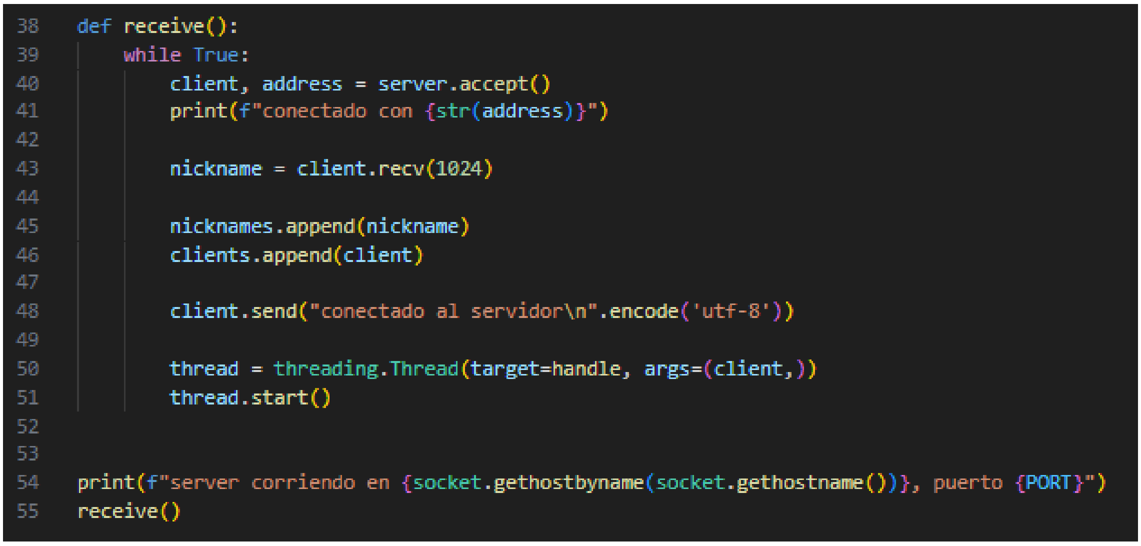
Código fuente

server.py:

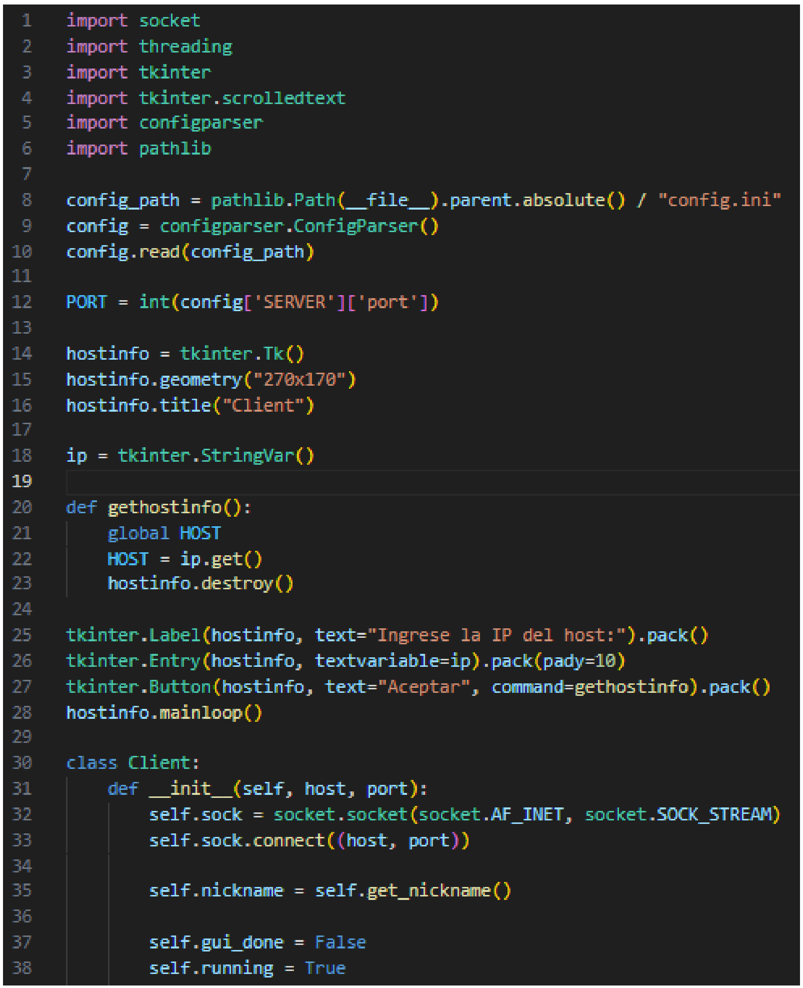


```python
import socket
import threading
import configparser
import pathlib

config_path = pathlib.Path(__file__).parent.absolute() / "config.ini"
config = configparser.ConfigParser()
config.read(config_path)

HOST = config['SERVER']['host']
PORT = int(config['SERVER']['port'])

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))

server.listen()

clients = []
nicknames = []

def broadcast(message):
    for client in clients:
        client.send(message)

def handle(client):
    while True:
        try:
            message = client.recv(1024)
            broadcast(message)
        except:
            index = clients.index(client)
            clients.remove(client)
            client.close()
            nickname = nicknames[index]
            nicknames.remove(nickname)
            break

```

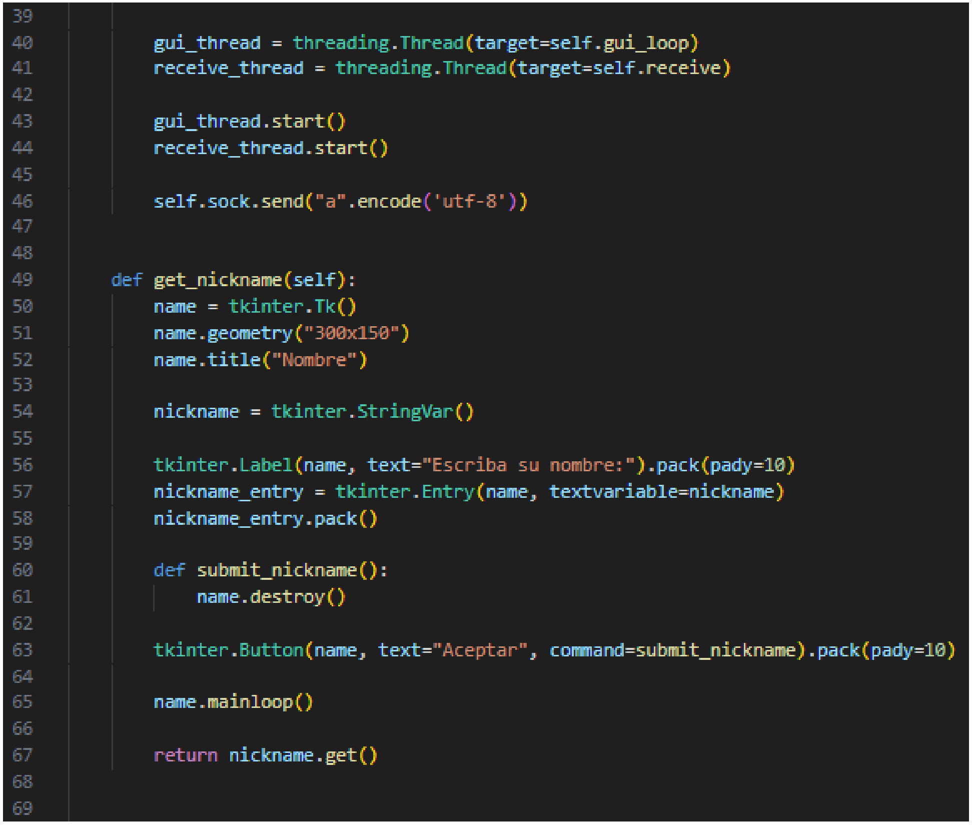


```python
def receive():
    while True:
        client, address = server.accept()
        print(f"conectado con {str(address)}")

        nickname = client.recv(1024)

        nicknames.append(nickname)
        clients.append(client)

        client.send("conectado al servidor\n".encode('utf-8'))

        thread = threading.Thread(target=handle, args=(client,))
        thread.start()


print(f"server corriendo en {socket.gethostbyname(socket.gethostname())}, puerto {PORT}")
receive()
```

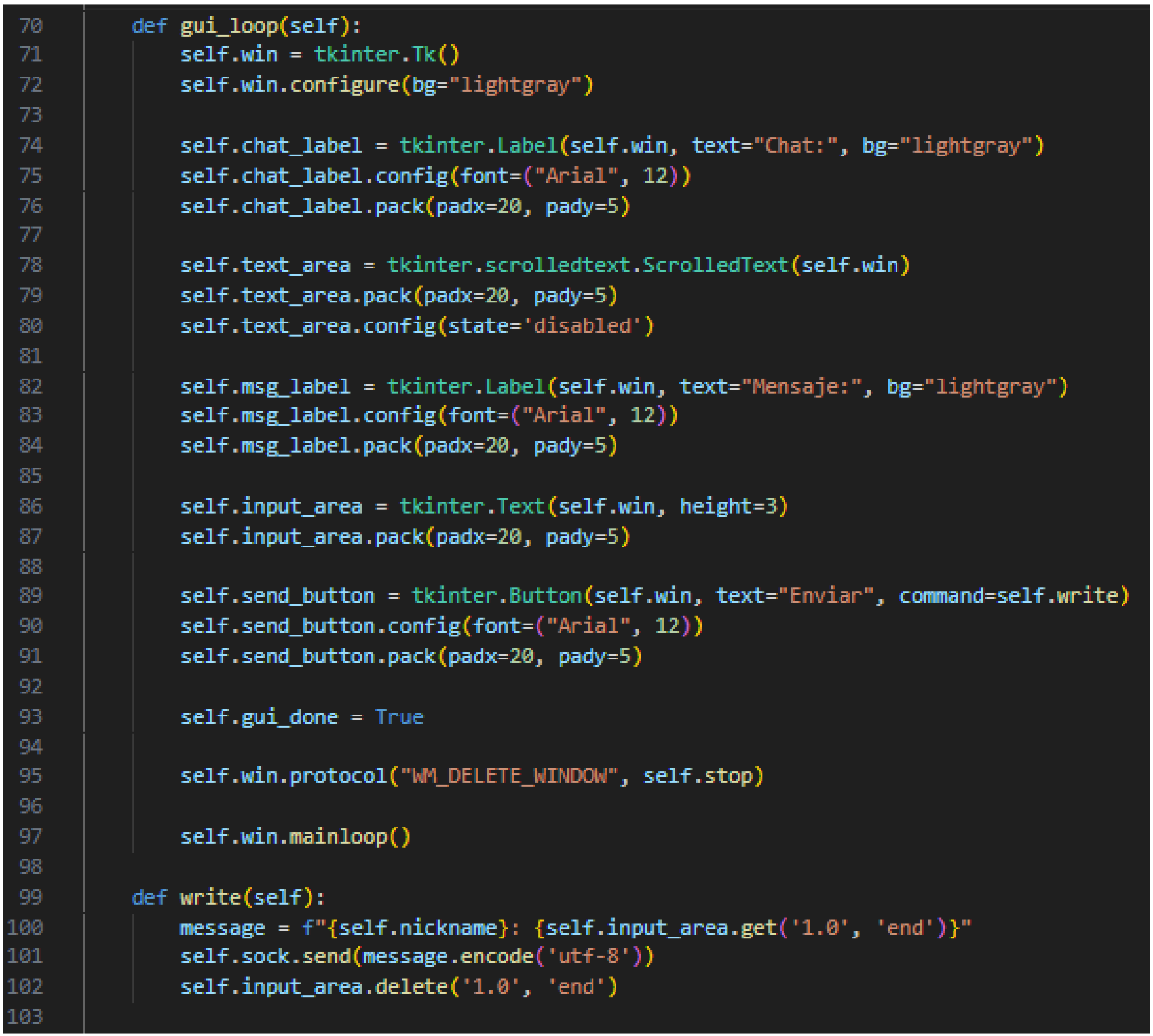
client.py:



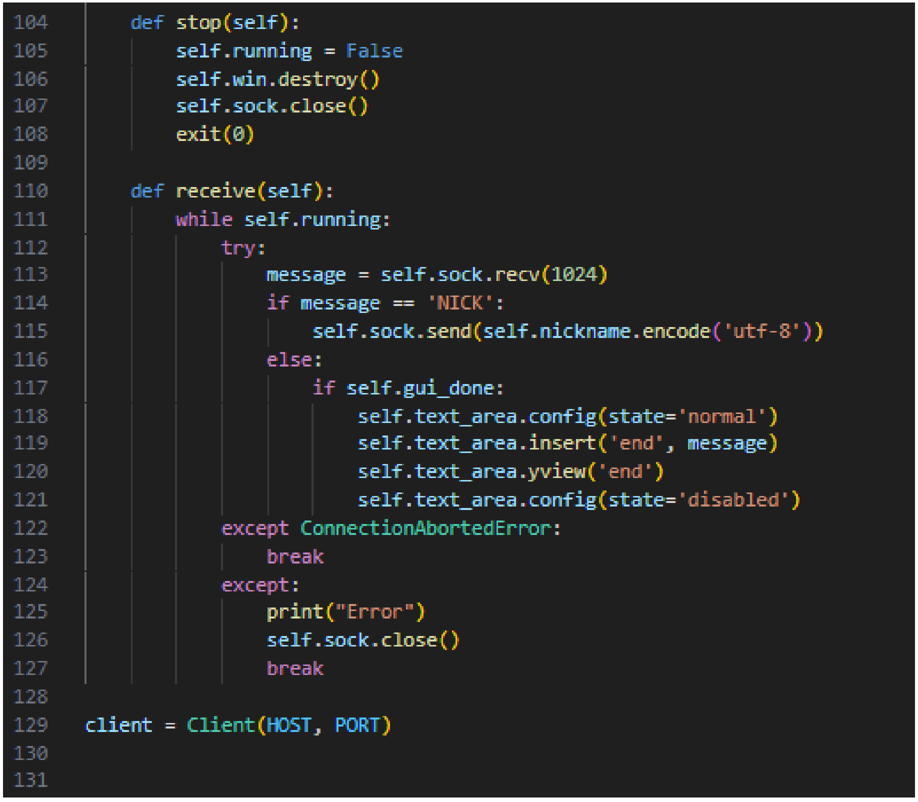
```python
import socket
import threading
import tkinter
import tkinter.scrolledtext
import configparser
import pathlib

config_path = pathlib.Path(__file__).parent.absolute() / "config.ini"
config = configparser.ConfigParser()
config.read(config_path)

PORT = int(config['SERVER']['port'])

hostinfo = tkinter.Tk()
hostinfo.geometry("270x170")
hostinfo.title("Client")

ip = tkinter.StringVar()

def gethostinfo():
    global HOST
    HOST = ip.get()
    hostinfo.destroy()

tkinter.Label(hostinfo, text="Ingrese la IP del host:").pack()
tkinter.Entry(hostinfo, textvariable=ip).pack(pady=10)
tkinter.Button(hostinfo, text="Aceptar", command=gethostinfo).pack()
hostinfo.mainloop()

class Client:
    def __init__(self, host, port):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.connect((host, port))

        self.nickname = self.get_nickname()

        self.gui_done = False
        self.running = True
```



```python

        gui_thread = threading.Thread(target=self.gui_loop)
        receive_thread = threading.Thread(target=self.receive)

        gui_thread.start()
        receive_thread.start()

        self.sock.send("a".encode('utf-8'))


    def get_nickname(self):
        name = tkinter.Tk()
        name.geometry("300x150")
        name.title("Nombre")

        nickname = tkinter.StringVar()

        tkinter.Label(name, text="Escriba su nombre:").pack(pady=10)
        nickname_entry = tkinter.Entry(name, textvariable=nickname)
        nickname_entry.pack()

        def submit_nickname():
            name.destroy()

        tkinter.Button(name, text="Aceptar", command=submit_nickname).pack(pady=10)

        name.mainloop()

        return nickname.get()

```

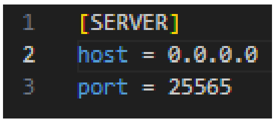


```python
    def gui_loop(self):
        self.win = tkinter.Tk()
        self.win.configure(bg="lightgray")

        self.chat_label = tkinter.Label(self.win, text="Chat:", bg="lightgray")
        self.chat_label.config(font=("Arial", 12))
        self.chat_label.pack(padx=20, pady=5)

        self.text_area = tkinter.scrolledtext.ScrolledText(self.win)
        self.text_area.pack(padx=20, pady=5)
        self.text_area.config(state='disabled')

        self.msg_label = tkinter.Label(self.win, text="Mensaje:", bg="lightgray")
        self.msg_label.config(font=("Arial", 12))
        self.msg_label.pack(padx=20, pady=5)

        self.input_area = tkinter.Text(self.win, height=3)
        self.input_area.pack(padx=20, pady=5)

        self.send_button = tkinter.Button(self.win, text="Enviar", command=self.write)
        self.send_button.config(font=("Arial", 12))
        self.send_button.pack(padx=20, pady=5)

        self.gui_done = True

        self.win.protocol("WM_DELETE_WINDOW", self.stop)

        self.win.mainloop()

    def write(self):
        message = f"{self.nickname}: {self.input_area.get('1.0', 'end')}"
        self.sock.send(message.encode('utf-8'))
        self.input_area.delete('1.0', 'end')

```



```python
    def stop(self):
        self.running = False
        self.win.destroy()
        self.sock.close()
        exit(0)

    def receive(self):
        while self.running:
            try:
                message = self.sock.recv(1024)
                if message == 'NICK':
                    self.sock.send(self.nickname.encode('utf-8'))
                else:
                    if self.gui_done:
                        self.text_area.config(state='normal')
                        self.text_area.insert('end', message)
                        self.text_area.yview('end')
                        self.text_area.config(state='disabled')
            except ConnectionAbortedError:
                break
            except:
                print("Error")
                self.sock.close()
                break

client = Client(HOST, PORT)


```

config.ini:



```ini
[SERVER]
host = 0.0.0.0
port = 25565
```