TP FINAL DE SEGURIDAD INFORMATICA

ALUMNOS:

MARQUES, LEONARDO DANIEL

RODRIGUEZ MARANO, JUAN MANUEL

DOCENTE: RENDE, DIEGO

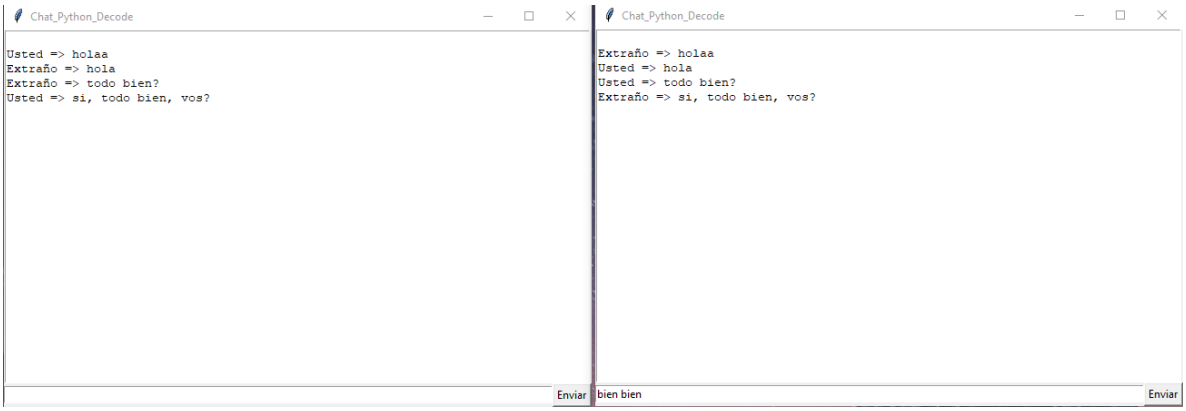
AÑO: 3RO

IFTS 16

2020

Chat_Python_Decode

Funcionamiento:



Captura 1.

El Chat_Python_Decode, es un chat básico de texto, en el cual consta de 2 archivos, uno de servidor y el otro cliente (se provee cliente como ejecutable .pyw sin consola y .py con consola), es de multichat, o sea se pueden agregar más de 2 chat a chatear, en si en esta muestra se utiliza la librería pybase64, en los cuales se utiliza sus herramientas para codificar y decodificar los mensajes antes de ser enviados. En esta demo, los mensajes son enviados como “Usted” y los mensajes provenientes del otro cliente aparecen como “Extraño”.

```
def send():
    msg = f"Usted => {self.e.get()}"
    self.txt.insert(END,"\n"+msg)

    b = self.e.get().encode("UTF-8")
    msg = pybase64.b64encode(b)
    self.send_msg(msg)

    self.e.delete(0,END)
```

Captura 2.

```
def msg_recv(self):
    while True:
        try:
            data = self.sock.recv(1024)
            if data:
                data = pickle.loads(data)
                print(data)
                data = pybase64.b64decode(data)
                data = data.decode("UTF-8")
                msg = f"Extraño => {data}"
                self.txt.insert(END,"\n"+msg)
        except:
            pass
```

Captura 3.

Aquí se puede apreciar el código utilizado para su correcto funcionamiento, tanto para él envió del mensaje encriptado (Captura 2) como para el mensaje recibido (Captura 3). En esta demo está configurado para que funcione en host=”localhost “ y port=4000. Para que funcione primero hay que ejecutar el archivo “servidor.py ” en cmd en la ruta donde se encuentran ambos archivos y luego ejecutar del de “cliente.py” o “cliente.pyw”. Si uno desea salir, simplemente debe cerrar la ventana del chat.