Redes de Computadoras

Proyecto 1: CREACIÓN E IMPLEMENTACIÓN DE UN PROTOCOLO DE LA CAPA DE APLICACIÓN

Reporte

Equipo Rocket

Integrantes:

- García Alcántara Karla Socorro
- Herrera Ortiz Oscar

Objetivo:

Implementación una trivia "¿Quién es este Pokemón?" sobre la capa de aplicación, funcional tanto el cliente como el servidor, de manera que el servidor envíe trivias en el formato clásico de los intermedios del programa Pokemón al cliente, y verifique correctamente si ha acertado este último.

Detalles de interpretación y ejecución:

- Para la correcta visualización de imágenes desde la terminal, usamos la aplicación de software libre EOG (Eye Of GNOME), que se puede obtener en el repositorio de git: git://git.gnome.org/eog
- Para interpretar y ejecutar el Servidor, usar desde la línea de comandos:
 \$ python server.py 9999
- Para interpretar y ejecutar el Cliente, usar desde la línea de comandos:
- \$ python client.py 127.0.0.1 9999

Descripción del código:

La aplicación fue dasarrollada en python con los siguientes programas:

common.py

Aquí implementamos la función 'getAsByte(value, n = 1)' en la que se recibe un mensaje, que consideramos que siempre se recibiría como una cadena, y se pasa a bytes. Esta función fue implementada aquí porque es utilizada por el servidor y el cliente, así que se llamará desde sus respectivos programas importando su módulo.

Server.py

Al ejecutarse, se verifica si se está usando correctamente, en caso contrario manda un mensaje en la terminal especificando su uso correcto; en seguida se verifica que se halla indicado el uso de un puerto válido.

Se crea el socket y se enlaza con el puerto y el host, y se hace un nuevo hilo de ejecución que cuando termina, se cierra la conexión del socket.

Función 'run(cn, addr)': inicia la ejecución de un intercambio de mensajes, 'cn' se refiere al socket del cliente que devuelve la aplicación de la función 'accept()' que es parte de la biblioteca socket de pyhton, 'addr' es la dirección del cliente que se obtiene con la misma función.

Durante la ejecución de esta función se imprimirá en la terminal todo el proceso de la trivia, y en caso de error o finalización del intercambio de mensajes se terminará la ejecución con la función 'killThread(cn)'.

Función 'killThread(cn)': cierra la conexión con el socket del cliente, 'cn' nuevamente se refiere a este socket, y finaliza este hilo de ejecución.

Client.py

Al ejecutar verifica que se hallan recibido los argumentos que se necesitan, en caso de error, se finaliza la ejecución luego de enviar un mensaje de error. Verifica que el host sea válido. Se crea el socket de conexión con el servidor y se conecta con el puerto y el host, entonces se reciben las trivias del servidor por medio de imágenes que se abren con la función 'openImage(data)' y en caso de error o finalización exitosa de toda la trivia se utiliza la función 'killClient(msj)'.

Función 'killClient(msj)': en caso de que el socket exista, cierra su conexión y envía el mensaje msj que fue hecho de acuerdo al caso en que se llamó a la función (error o éxito).

Función 'openImage(data)': hace una llamada al sistema para abrir las imágenes utilizando desde la línea de comandos la herramienta 'eog' o en caso de que se ejecute en windows, se utiliza 'explorer' (el visor de imágenes).

```
Proyecto_Redes>python server.py 9999
-
```

Fig 1 Inicialización de servidor

Fig 2 Inicialización de cliente (Puerto especifico, puerto default 9999)

Fig 3 Conexión aceptada por el servidor

```
Proyecto_Redes>python client.py 127.0.0.1
Proyecto_Redes>python client.py 127.0.0.1
Pto del cliente: 5231
Pto del cliente: 5231
With thread 4276
Codigo recibido: 10

txbytes: 1

Pto del cliente: 127.0.0.1
Pto del cliente: 127.0.0
Pto del cliente: 12
```

Fig 4 Envio de solicitud de trivia/Recibo de solicitud de trivia

Fig 5 Respuesta de servidor(trivia, cod 20) / Recibo de respuesta(cliente) en cliente

```
Codigo recibido: 10
rxbytes: 1
rxbytes: 1
Tamano de la imagen: 107342 bytes
Codigo enviado: 20
txbytes = 51 (codigo 20)
Enviando imagen ...
Paquetes enviados: 105
txbytes: 107342 (imagen)
```

Fig 6 Envío de imagen (servidor) / Imagen recibida (cliente)

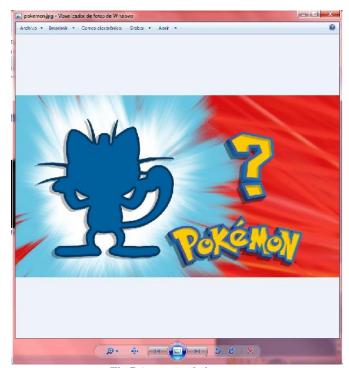


Fig 7 Apertura de imagen

```
SERVER
                                                                             IP del cliente: 127.0.0.1
Pto del cliente: 5231
Proyecto_Redes>python client.py 127.0.0.1
------- CLIENT ------
                                                                             with thread 4276
----- CLIENT
 Introduce codigo a enviar: 10
 txbytes: 1
                       SERVER
                                                                              Codigo recibido: 10
Codigo recibido: 20
rxbytes: 51
[d Pokemon: 52
                                                                              exbytes: 1
                                                                                                        SERVER -
ld Pokemon: 52
[amano imagen en bytes: 107342
Respuesta 1: Bulbasaur
Respuesta 2: Meowth
Respuesta 3: Voltorb
Recibiendo imagen ...
Paquetes recibidos: 105
[amano de la imagen en bytes: 107342
[magen recibida y guardada correctamente
                                                                             Tamano de la imagen: 107342 bytes
Codigo enviado: 20
txbytes = 51 (codigo 20)
                                                                             Enviando imagen ...
Paquetes enviados: 105
                                                                             txbytes: 107342 (imagen)
---- CLIENT ---
lmagen rect.
Abriendo imagen ...
CLIENT
                                                                             Codigo recibido: 11
                                                                              exbytes: 3
Introduce numero de respuesta: 2
txbytes: 3
                                                                             idPok: 52
                                                                             idResn: 2
```

Fig 8 Envio de respuesta (cod 11) cliente/ recibo de respuesta (servidor)

Fig 9 Envío de respuesta correcta (servidor, cod 21) / Recibo de respuesta (cliente). Finaliza la conexion

Fig 10 Manejo de múltiples conexiones