

Práctica “TIMEOUT SOLICITUD-Respuesta confiable parte 1”



GRUPO: 4CM3

EQUIPO 3

INTEGRANTES:

• MAYA ROCHA LUIS EMMANUEL

• OSUNA BANDA ITZEL ARELY

• MIRANDA MOJICA ERICK

• BARBOSA PEÑA XAVIER MARISTIN

**DIRECCIONES IP**

**Primera computadora:**

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Segunda computadora:**

**Captura de pantalla de un celular

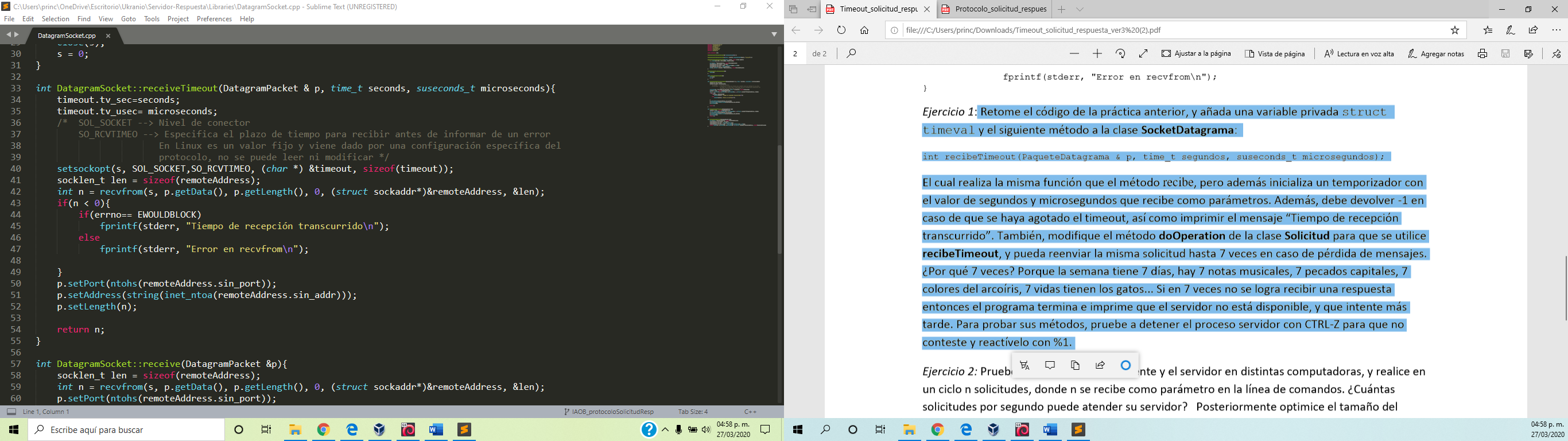
Descripción generada automáticamente**

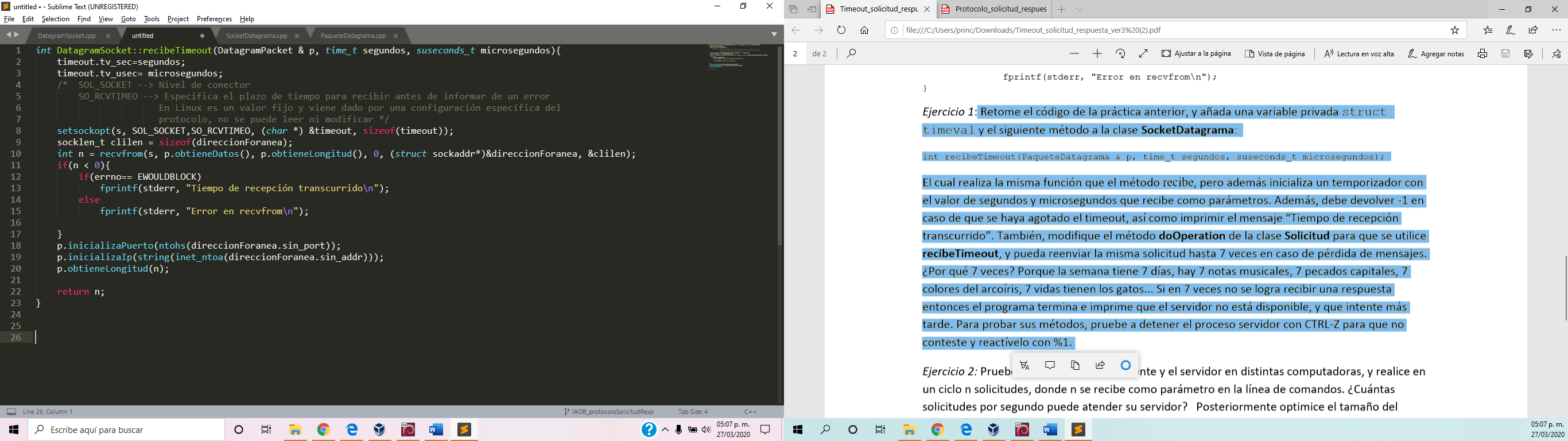
**Ejercicio 1: Retome el código de la práctica anterior, y añada una variable privada struct timeval y el siguiente método a la clase SocketDatagrama:**

**int recibeTimeout(PaqueteDatagrama & p, time\_t segundos, suseconds\_t microsegundos);**

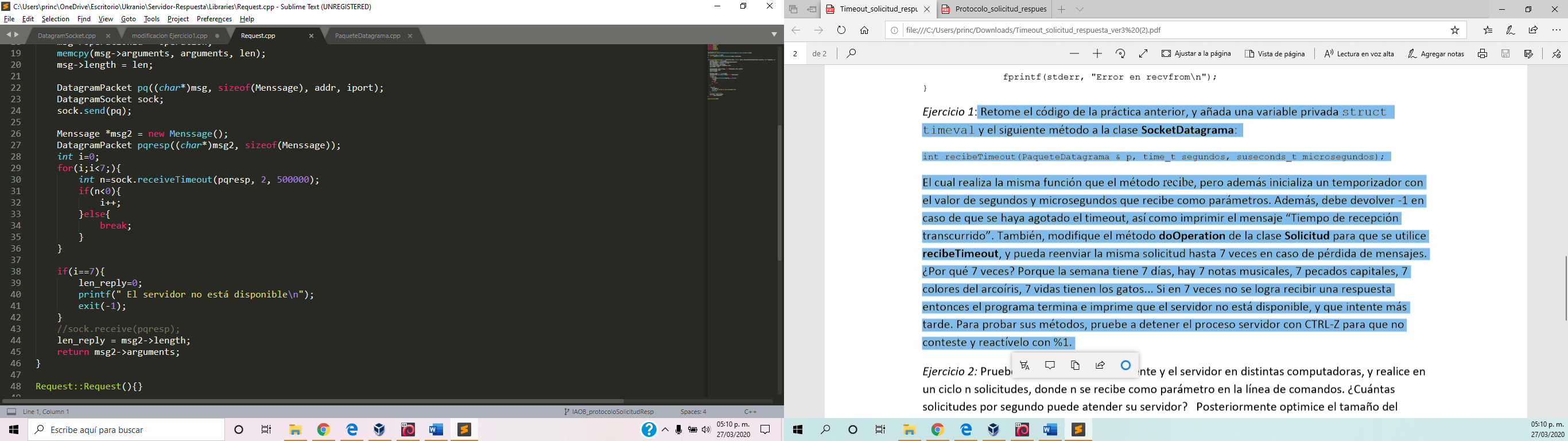
**El cual realiza la misma función que el método recibe, pero además inicializa un temporizador con el valor de segundos y microsegundos que recibe como parámetros. Además, debe devolver -1 en caso de que se haya agotado el timeout, así como imprimir el mensaje “Tiempo de recepción transcurrido”. También, modifique el método doOperation de la clase Solicitud para que se utilice recibeTimeout, y pueda reenviar la misma solicitud hasta 7 veces en caso de pérdida de mensajes. ¿Por qué 7 veces? Porque la semana tiene 7 días, hay 7 notas musicales, 7 pecados capitales, 7 colores del arcoíris, 7 vidas tienen los gatos... Si en 7 veces no se logra recibir una respuesta entonces el programa termina e imprime que el servidor no está disponible, y que intente más tarde. Para probar sus métodos, pruebe a detener el proceso servidor con CTRL-Z para que no conteste y reactívelo con %1.**

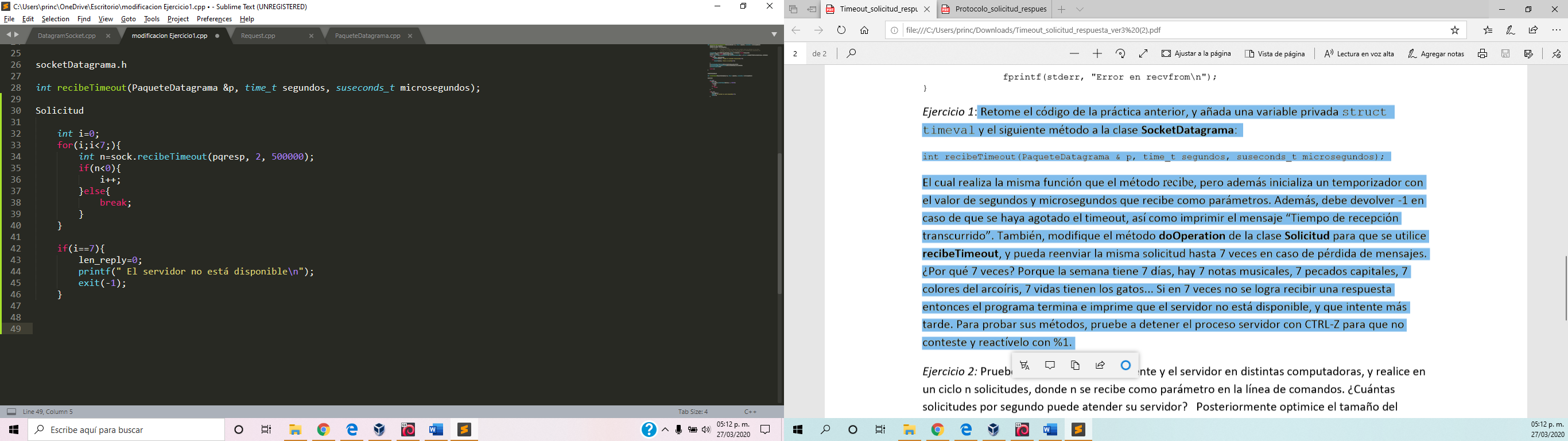
Se agregó el siguiente método a SocketDatagrama:





Modificación de la clase Solicitud para poder reenviar la misma solicitud hasta 7 veces:





**Pruebas realizadas:**

**Detener servidor y continuar**

Captura de pantalla de un celular

Descripción generada automáticamente

**Cliente:**

Se muestra como no detenemos el servidor y la solicitud se realiza en primer tiempo, posteriormente se va dejando uno por uno el tiempo de recepción hasta 7 timeouts donde se manda el mensaje que el servidor no está disponible.

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Captura de pantalla de un celular

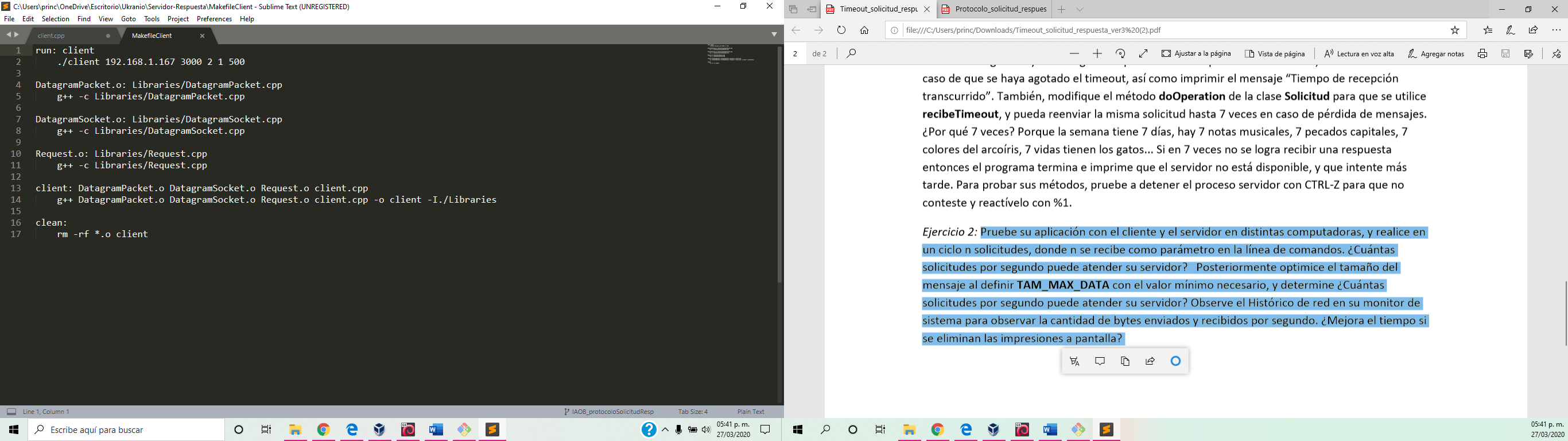
Descripción generada automáticamente**

**Captura de pantalla de un celular

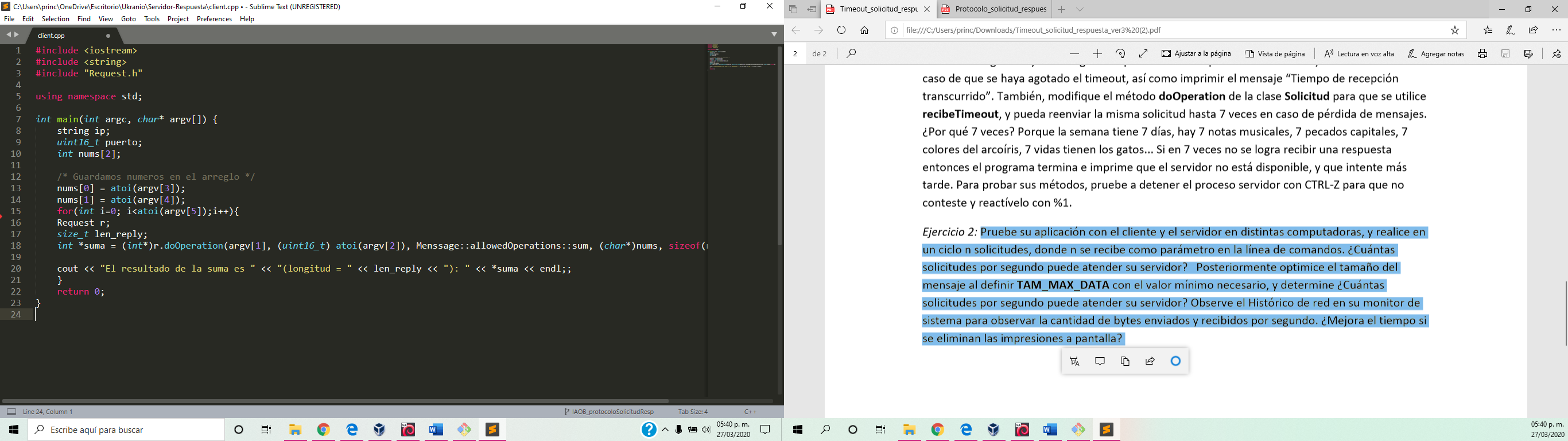
Descripción generada automáticamente**

**Ejercicio 2: Pruebe su aplicación con el cliente y el servidor en distintas computadoras, y realice en un ciclo n solicitudes, donde n se recibe como parámetro en la línea de comandos. ¿Cuántas solicitudes por segundo puede atender su servidor? Posteriormente optimice el tamaño del mensaje al definir TAM\_MAX\_DATA con el valor mínimo necesario, y determine ¿Cuántas solicitudes por segundo puede atender su servidor? Observe el Histórico de red en su monitor de sistema para observar la cantidad de bytes enviados y recibidos por segundo. ¿Mejora el tiempo si se eliminan las impresiones a pantalla?**

**Se inicializó con 500 solicitudes en la línea de comandos:**



**For para las n solicitudes:**



**Se tomó el tiempo de las 500 solicitudes ingresadas:**

**Una captura de pantalla de una red social

Descripción generada automáticamente**