

**Documentación Tarea
Programada 1.**

**TI3404 Lenguajes de
Programación**

II SEMESTRE 2013

Estudiantes:
Gerardo Calderón
Jose David Hidalgo
Juliana Mora
Allan Porras

Profesor:
Andrei Fuentes

Índice

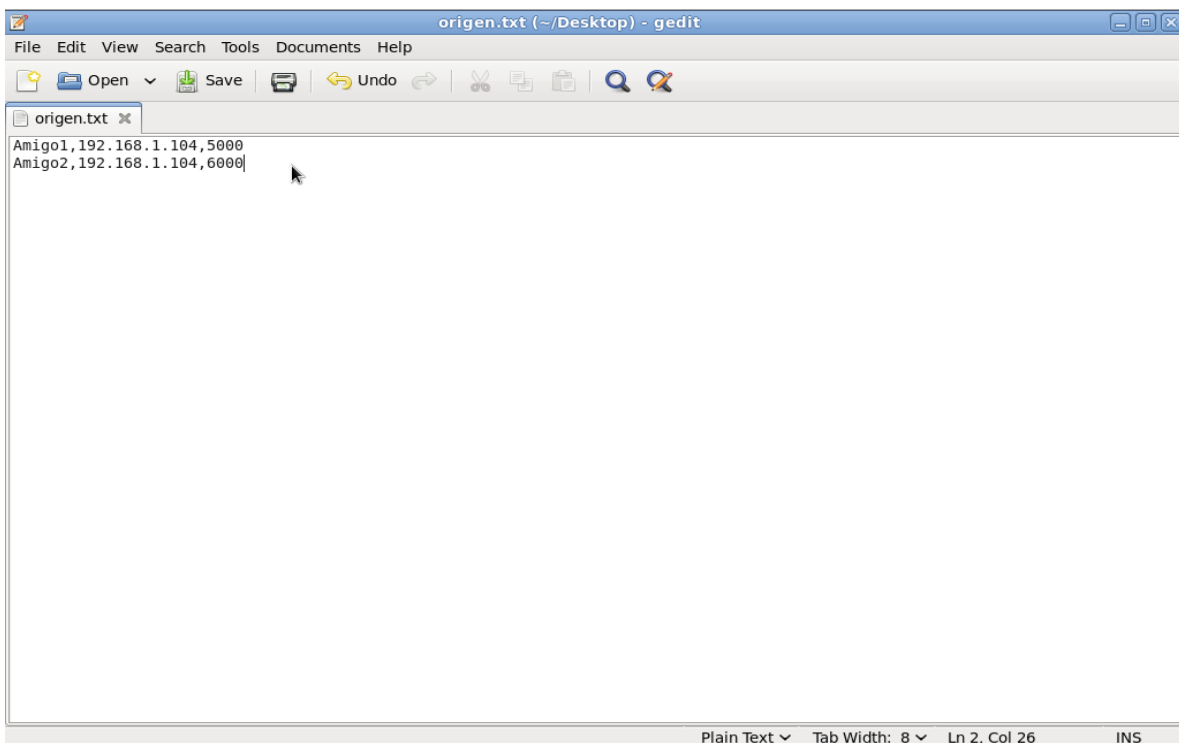
Descripción del problema	2
Funcionalidades del programa	2
Librerías usadas:	6
Análisis de Resultados:.....	6
Manual de usuario:	7
Conclusión:	10

Descripción del problema

Crear un mini programa en el lenguaje de programación C que envíe y reciba mensajes simulando la aplicación WhatsApp. Para esto se utilizarán sockets (mecanismos de comunicación a la capa de transporte de modelo OSI) los cuales permitirán conectar dos PC's distintas solicitando la IP y nombre de usuario para lograr enviar el mensaje al puerto. El mecanismo para agregar amigos es sencillo y cada usuario tanto el A como el B deberán de registrarse mutuamente (de manera manual). Asimismo el programa deberá de ser compatible en el ambiente Linux para su respectiva revisión.

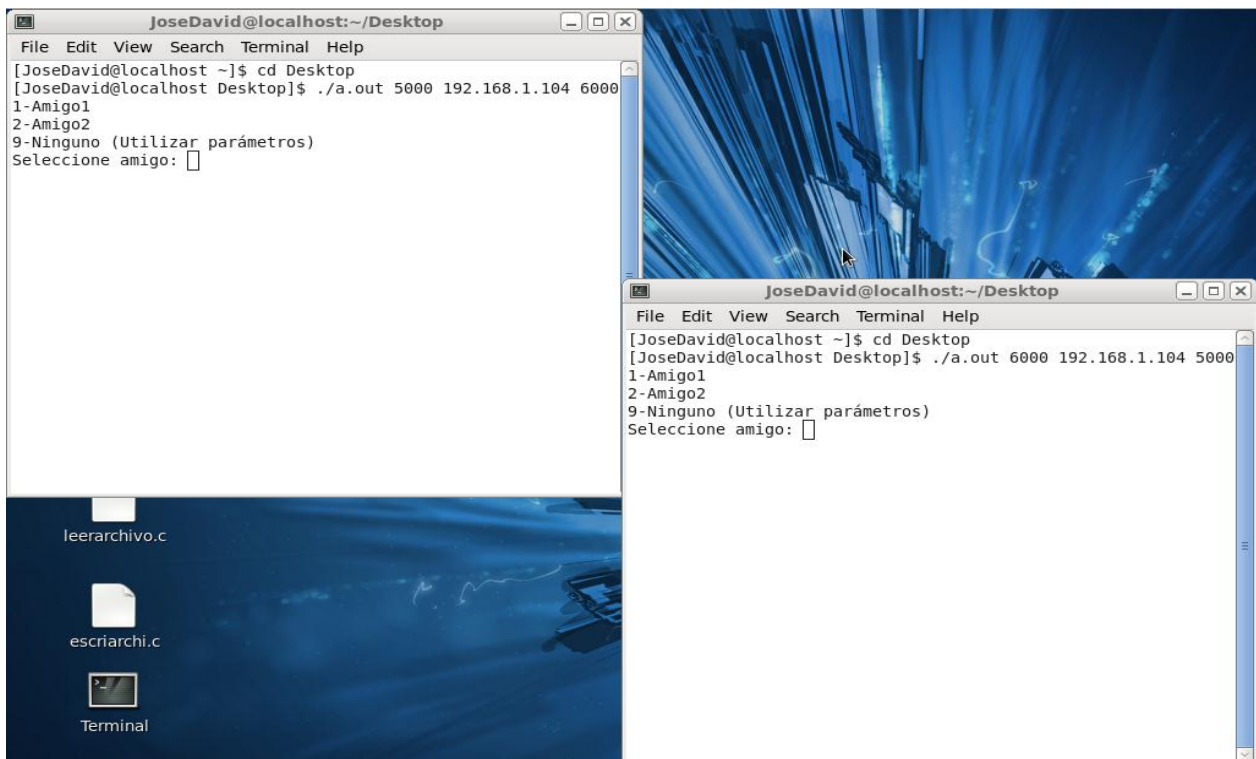
Funcionalidades del programa

Agregar amigos: se solicita el IP, puerto y nombre de usuario de forma manual, tanto amigo A como amigo B deben de agregarse.

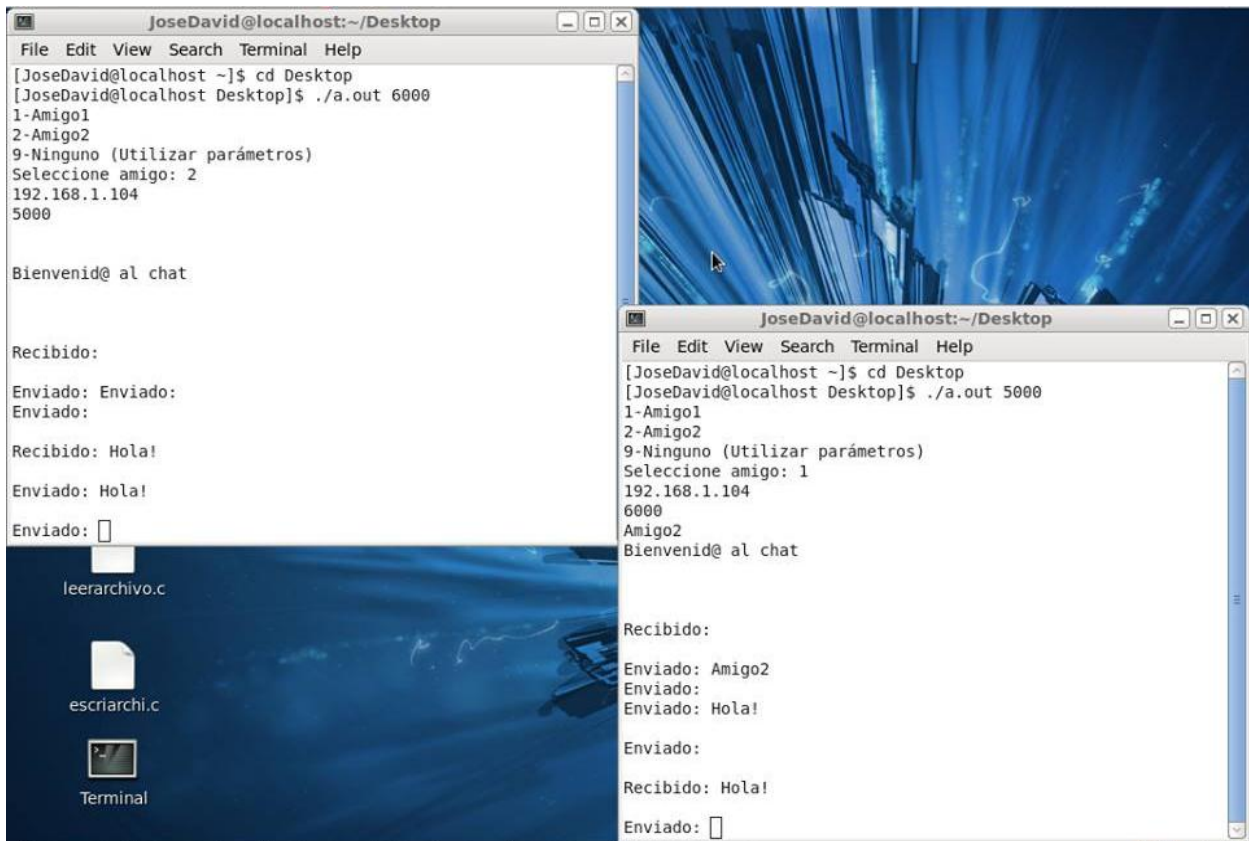


```
File Edit View Search Terminal Help
[JoseDavid@localhost ~]$ cd Desktop
[JoseDavid@localhost Desktop]$ gcc msn.c
[JoseDavid@localhost Desktop]$ ./a.out 5000 192.168.1.104 6000
9-Ninguno (Utilizar parámetros)
Seleccione amigo: 9
¿Desea agregar un nuevo amigo? [1 si lo desea agregar, 0 si no]1
Escriba el nombre del amigo: Amigo1

Amigo1
Bienvenid@ al chat
█
```



Enviar mensajes: Se envía un mensaje al puerto e IP asociados al nombre de usuario especificados.



Recibir mensajes: El puerto para recibir mensajes se encuentra especificado en el archivo de configuración, de esta manera lo encuentra.

Diseño del programa

Diseñado para que tuviera parámetros, en ese caso, uno mismo define el puerto de escucha, y sería el mismo cada vez que utiliza el programa, por ejemplo:

✚ *Si un amigo lo tiene agregado a usted, el puerto suyo en el archivo “origen” de su amigo, tiene el puerto que usted especificó.*

A la vez que uno mismo empieza el programa, y el archivo origen tiene amigos agregados:

✚ *Se desplegará una lista en la pantalla con los amigos que tenga usted.*

✚ *Si usted quiere agregar a un amigo, tiene que saber la dirección IP y el puerto de su amigo, ya que así se guardaría en su archivo “origen”. Para que en la próxima vez que utilice el programa, solo se cargue y lo seleccione.*

Se utilizaron algoritmos específicos para:

✚ *La lectura inicial del archivo de texto.*

✚ *La agregación de amigos, validación de nombres repetidos.*

✚ *Lectura de mensajes: Entre estos, se implementaron métodos estudiados en clase para la comparación de “Strings” (Cadenas de Caracteres), al igual que de utilidad.*

Librerías usadas:

Entrada y Salida:

- `#include <stdio.h>`
- `#include <stdlib.h>`

Manejo de “Strings”:

- `#include <strings.h>`
- `#include <string.h>`

Sockets y Network:

- `#include <sys/socket.h>`
- `#include <netinet.h>`
- `#include <netdb.h>`

Manejo de tipos:

- `#include <unistd.h>`
- `#include <sys/types.h>`

Análisis de Resultados:

Se cumplió con los objetivos más importantes, que eran: La funcionalidad de los sockets, por medio del protocolo TCP/IP, la utilización del método “fork ()” para el manejo general y bifurcación entre sockets, además del envío y recepción de mensajes, el mismo envío y recepción de mensajes entre usuarios del programa (amigos), la lectura del archivo de amigos al inicio del programa, el agregar amigos y que se guarden y actualicen en el mismo archivo de texto.

El envío y recepción de mensajes se ve notoriamente como: “Recibido: (mensaje recibido) \n Enviado: (mensaje enviado)” no se pudo implementar el cambio de colores en las letras, ya que no existe una librería específica para hacerlo, y mediante la búsqueda en Internet sobre cómo hacerlo, provocaría agregar líneas de código muy extensas, sólo para esa implementación, según los resultados de la búsqueda, entonces se dejó así.

A veces ocurre un pequeño error que dice: “error en el bind”, y esto pasa cuando se ejecuta el programa múltiples veces, ya que queda basura en la memoria.

Manual de usuario:

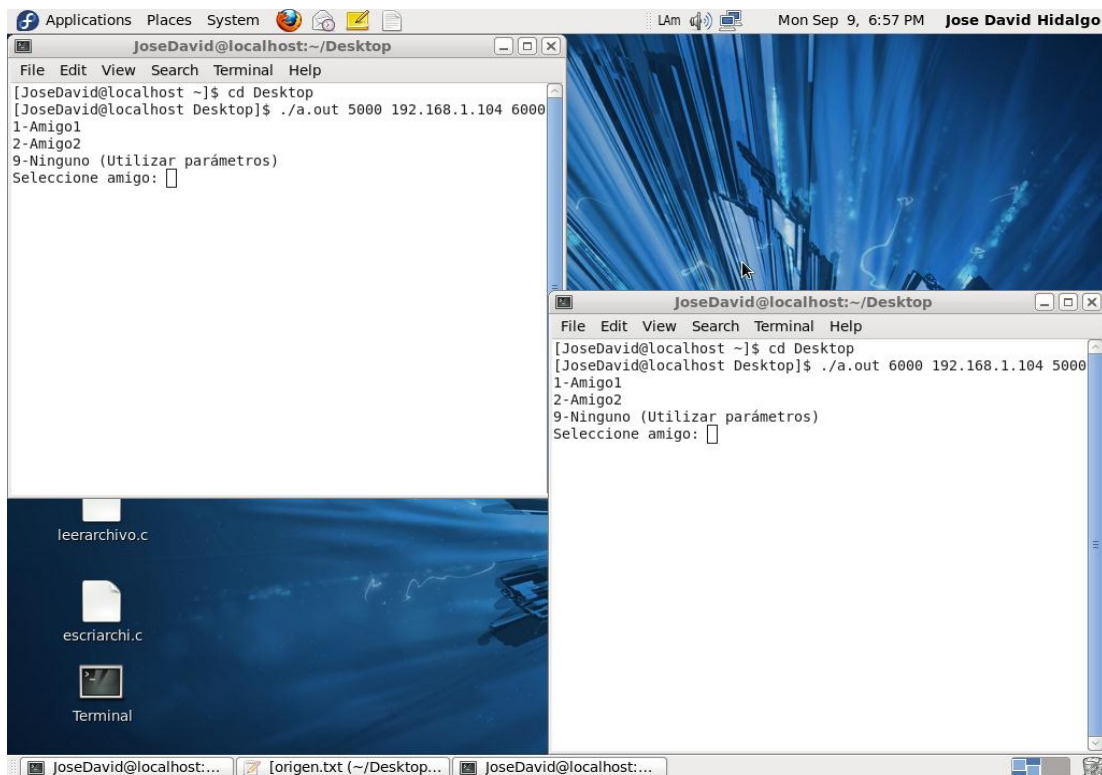
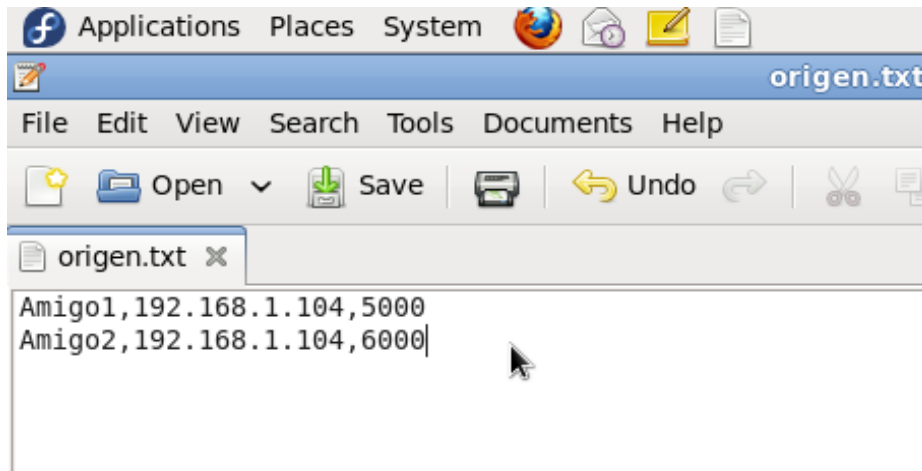
Bienvenido a su programa de chat con amigos “msn”, completamente trabajado por líneas de comandos para su comodidad, y así no tenga usted que lidiar con alguna interfaz que le resulte difícil de utilizar.

Ahora, comencemos con lo básico, lo primero que tiene que verificar es que usted tenga, este manual de usuario, el archivo de texto “MAKEFILE” (importante para su ejecución), el archivo de texto “origen”, y el programa “msn” en el formato de C (msn.c).

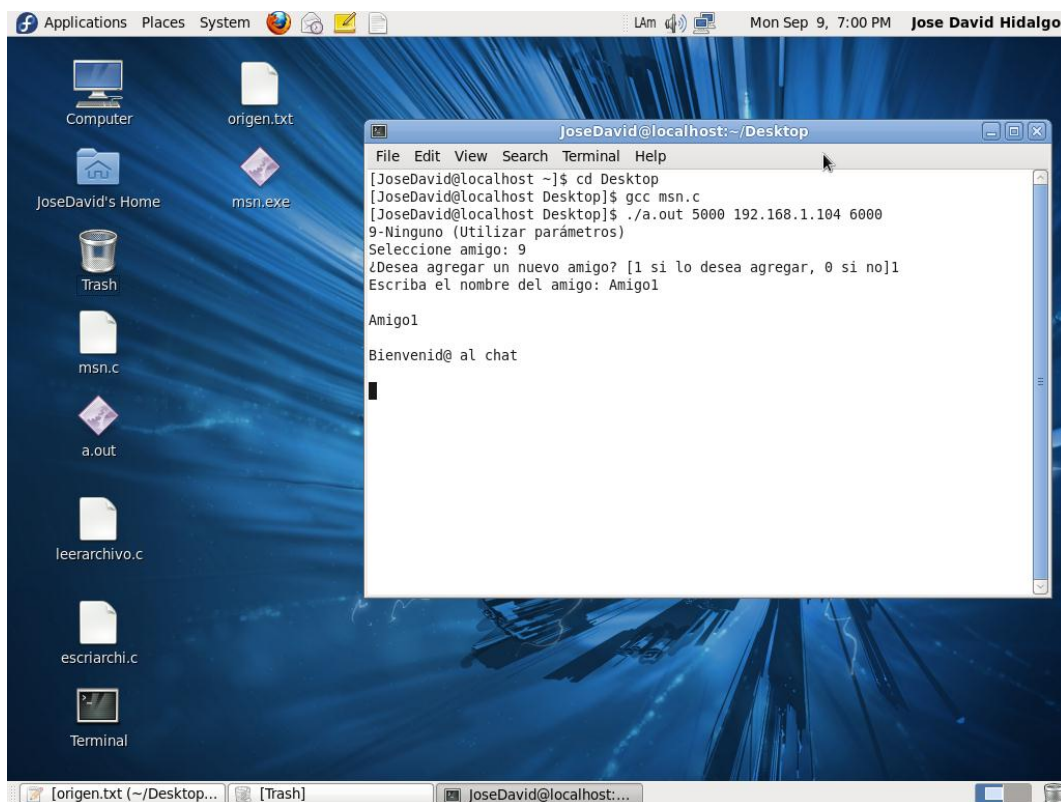
Según el archivo de texto MAKEFILE, en la línea de comandos, usted tiene que escribir 2 códigos diferentes, y para asegurarse de que funcione, el archivo “origen” y el programa:”msn.c”, estrictamente deben estar en el mismo directorio, ya sea en “Documentos”, “Escritorio”, o alguna carpeta específica. Una vez que usted escribe los 2 códigos, usted ya tiene el código ejecutable, y no funciona con solo darle click.

Antes de correr el programa (como está especificado en la documentación interna del código), tiene que deshabilitar el “Firewall” de su Sistema Operativo, ya que si no lo hace, no se establecería una conexión entre 2 usuarios. Luego, para correrlo, usted tiene 2 opciones, basadas en el archivo “origen”:

- Si usted quiere hablar con alguno de los amigos de la lista, debe correr el programa con Un (1) solo parámetro: con su puerto de escucha que usted habilite (de esta manera: “./msn.exe (puerto de escucha)”), con la condición de que si y solo si su amigo lo tenga registrado en su archivo con ese puerto, y con la dirección IP correcta. Entonces se puede establecer la conexión Y hablar sin problemas.



- La otra manera es, si no usted no quiere hablar con uno de sus amigos en la lista, sino, agregar a alguien a su lista. Para esto es necesario que usted conozca: la dirección IP de su amigo y el puerto que va a utilizar, ya que se corre el programa con Tres (3) parámetros: su puerto de escucha, la dirección IP del nuevo amigo, y el puerto del nuevo amigo (de esta manera: “./msn.exe (puerto de escucha) (dirección IP del amigo) (puerto del amigo)”). Luego el programa le pregunta también si desea agregar a ese nuevo amigo a su lista Digitando 1 si quiere agregarlo, 0 si no, y luego presionando “enter”, ya eso queda a criterio suyo, de todas maneras, si lo desea agregar: en el archivo “origen”, el cursor o línea de donde comienza a escribir debe estar al final de la línea del último amigo insertado, cada vez que se inserta uno nuevo, todo con el objetivo de evitar errores de escritura. Y a la hora de escribir el nombre del amigo, tiene que escribir el nombre 1 vez, presionar “enter”, escribir el nombre de nuevo, presionar “enter”, y listo, ya puede hablar con ese amigo.



Ya luego de estas simples y claras instrucciones, no debería tener problemas al utilizar este programa, Esperamos que lo disfrute tanto como nosotros. ¡Muchas Gracias!

-----Atentamente: El Equipo de Trabajo de la Aplicación-----

Conclusión:

Es un programa muy complejo, que permite explorar ligeramente el mundo de las redes, por medio de sockets, incluyendo las funcionalidades internas manejadas por archivos; además, la materia vista en clase fue de ayuda importante para la realización del trabajo.

Se comprendió más el lenguaje C, y sus extensas ramas que nos permitieron programar el trabajo, y además de la búsqueda de soluciones, por algún inconveniente que hubo en cuanto a funcionalidad de alguna parte específica.

Constantes pruebas fueron clave para determinar su funcionamiento interno, incluyendo la solución de problemas comunes que aparecían a lo largo del trabajo.

Para finalizar, fue una muy buena fuente de aprendizaje y la cantidad de esfuerzo no fue tomada en vano.