

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. In the bottom-left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

22-11-2013

Tarea Programada

Clonación WhatsApp

Fabiana

LENGUAJES DE PROGRAMACION

Tabla de contenido

Descripción del Problema	2
Diseño del programa	3
Librerías utilizadas.....	6
Análisis de Resultados.....	7
Manual de Usuario.....	8
Conclusión Personal	11

Descripción del Problema

La presente tarea programada tiene como objetivo desarrollar un programa similar al funcionamiento de la aplicación WhatsApp, este programa debe permitir al usuario poder enviar mensajes y recibir simultáneamente. Debe contener un archivo donde se almacene los contactos del usuario este archivo debe tener:

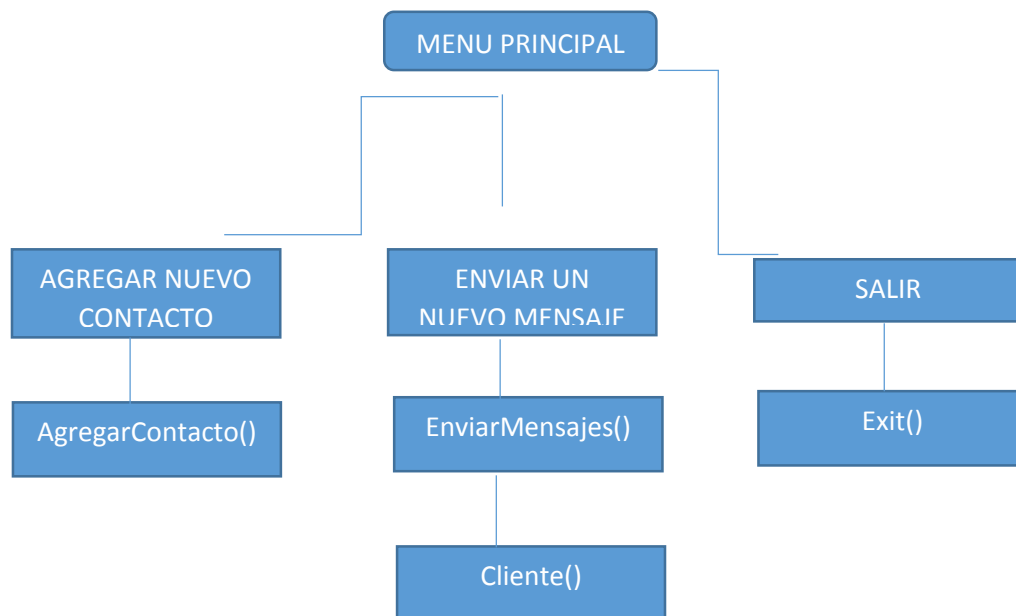
- Nombre
- IP
- Puerto

De igual manera, el programa debe permitir al usuario agregar nuevos contactos, solicitándole la información necesaria para guardarla en el archivo respectivo de los contactos.

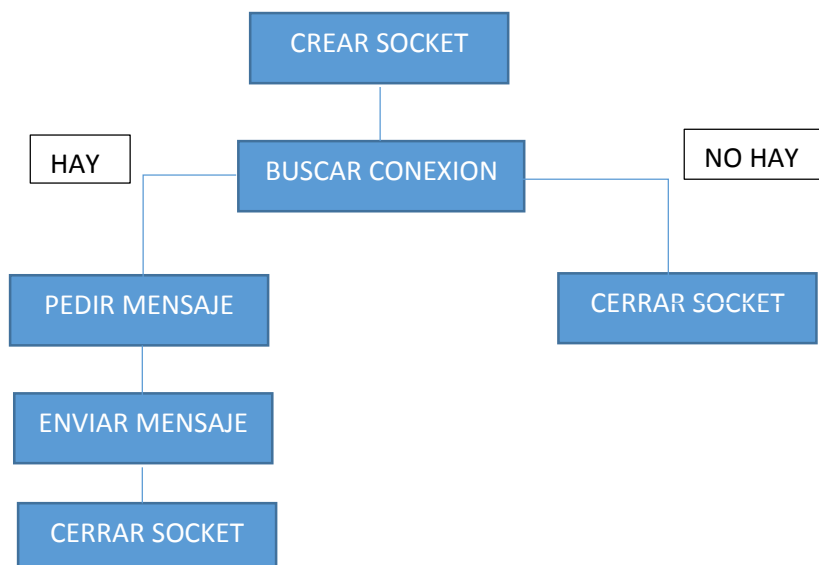
El puerto por el cual el usuario reciba los mensajes de otras personas debe de estar siempre en escucha, de este modo en el momento de recibirlo se debe mostrar.

Diseño del programa

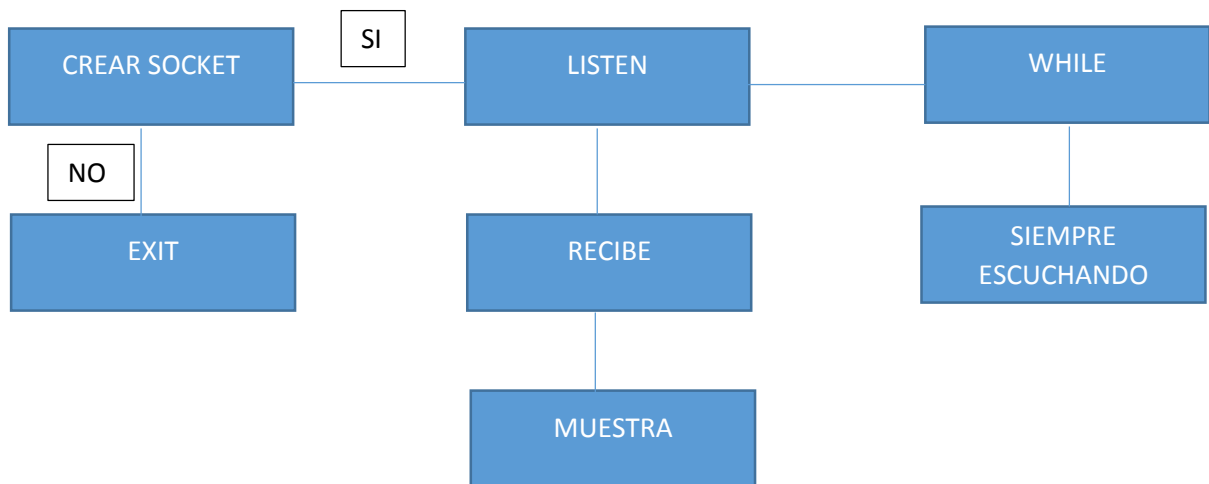
A continuación se muestran los algoritmos de lógica que se siguieron para el funcionamiento del programa:



El siguiente esquema muestra con más detalle la lógica del algoritmo empleado en la función Cliente() la cual trabaja con sockets:



También se muestra la estructura lógica base para el desarrollo de la función Servidor() que de igual manera trabaja con el manejo de sockets:



Seguidamente se mencionarán las funciones que componen el código del programa desarrollado. Adjunto a cada una, una breve descripción de cada una:

AgregarContacto(): Esta función permite al usuario ingresar los datos necesarios para ingresar un nuevo contacto al documento que los almacena (Contactos.txt).

EnviarMensaje(): Por medio de este método se solicita el nombre del contacto al cual se desea enviar un nuevo mensaje, se procede a partir de este dato buscar en el documento que contiene la información de cada contacto la extracción del IP y Puerto para enviar esta información como parámetro de la función Cliente()

Cliente(): Esta función se encarga de crear el socket y si este tiene conexión pide el mensaje que se desea enviar al usuario para enviarlo a la IP y Puerto que ya vienen como parámetros

RecibirMensajes(): En ella, se localiza del documento “Config.txt” la información del IP y Puerto propio para ser enviados como parámetro a la función Servidor()

Servidor(): Acá se crea un socket, si se logra la conexión, dentro de un ciclo while se habilita que este escuche la entrada de un nuevo mensaje y sea mostrado al usuario.

MenuPrincipal(): Desde esta función se le muestra al usuario el menú del programa, pidiendo un número del 1 al 3. El número 1 es si desea agregar un nuevo contacto, el número 2 si desea enviar un nuevo mensaje y el número 3 si desea salir del programa. Según la selección que el usuario indique se reenvía a la función pertinente para realizar lo solicitado.

Main(): En esta función se utiliza el fork(), permitiendo realizar simultáneamente el envío y recibo de mensajes, como también se utiliza hilos para enviar mensajes y escuchar desde el puerto.

Colores en la terminal

Para diferenciar los mensajes de entrada y salida se investigó el recurso con el cual se podía alcanzar esta tarea, los resultados fueron agregar un código antes del string que se deseaba mostrar al usuario, el código varía según el color requerido. Los encontrados fueron:

Negro	00;30m	Gris oscuro	01;30m
Rojo	00;31m	Rojo claro	01;31m
Verde	00;32m	Verde claro	01;32m
Café	00;33m	Amarillo	01;33m
Azul	00;34m	Azul claro	01;34m
Púrpura	00;35m	Púrpura claro	01;35m
Cian	00;36m	Cian claro	01;36m
Gris claro	00;37m	Blanco	01;37m

Los utilizados en nuestro programa fueron el color blanco, azul claro y amarillo.

Librerías utilizadas

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

Contiene tipos, macros y funciones para la realización de tareas de entrada y salida

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

Contiene tipos, macros y funciones para la manipulación de cadenas de caracteres

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

Contiene tipos, macros y funciones para la conversión numérica, generación de números aleatorios, búsquedas y ordenación, gestión de memoria y tareas similares.

Sockets

Para el desarrollo de sockets dentro del programa se utilizaron las siguientes librerías:

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

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

Análisis de Resultados

Los objetivos planteados para este proyecto son:

- Crear un documento (dinámico) para el almacenamiento de contactos
- Utilizar sockets en C para el envío y recibo de mensajes
- Recurrir al fork() para mantener simultáneamente la mensajería
- Diferencia mensajes, a través de color identificado

Análisis de cada objetivo:

El primer objetivo fue alcanzado con éxito. Se puede manipular el archivo "Contactos.txt" ya sea para leer información desde este o bien editarlo y agregar nuevos a el. El archivo cuanta en cada línea con la siguiente información: IP, puerto y nombre.

La utilización de sockets se pudo alcanzar, el programa tiene la capacidad de enviar mensajes y de recibirlos. La creación de esto se logró guiado por un documento que explica paso a paso el procedimiento y la justificación del código necesario

El fork() fue investigado e implementado, en perspectiva se logra exitosamente pero en funcionalidad, da a parecer que no funciona pues el recibo de mensajes se ve condicionado a enviar un nuevo mensaje.

Los colores mostrados en consola, se alcanzaron utilizando color blanco para el menú y demás enunciados, color azul claro para el envío de mensajes y color amarillo para mostrar los mensajes nuevos.

Manual de Usuario

Compilación: para esto vamos a localizar primero la carpeta donde guardamos los archivos del programa. Seguidamente ponemos gcc “Nombre del archivo .c” –o “Nombre para ejecutar” –lpthread (activar los hilos).

```
fabiana@fabiana: ~/C/WhatsApp
fabiana@fabiana:~$ cd /home/fabiana/C/WhatsApp
fabiana@fabiana:~/C/WhatsApp$ gccCodigo.c -o whatsapp -lpthread
fabiana@fabiana:~/C/WhatsApp$ ./whatsapp
```

Se muestra el menú,

```
fabiana@fabiana: ~/C/WhatsApp
fabiana@fabiana:~$ cd /home/fabiana/C/WhatsApp
fabiana@fabiana:~/C/WhatsApp$ gccCodigo.c -o whatsapp -lpthread
fabiana@fabiana:~/C/WhatsApp$ ./whatsapp

-----Bienvenido a WhatsApp-----

----Agregar un nuevo contacto[1]---
----Enviar un nuevo mensaje[2]-----
----Salir[3]-----
```

Agregar contactos: Opción 1. Se ingresa la IP, el puerto y un nombre para el contacto

```
fabiana@fabiana: ~/C/WhatsApp
fabiana@fabiana:~$ cd /home/fabiana/C/WhatsApp
fabiana@fabiana:~/C/WhatsApp$ gccCodigo.c -o whatsapp -lpthread
fabiana@fabiana:~/C/WhatsApp$ ./whatsapp

-----Bienvenido a WhatsApp-----

----Agregar un nuevo contacto[1]---
----Enviar un nuevo mensaje[2]-----
----Salir[3]-----
1

Ingrese la IP del contacto: 127.0.0.1
Ingrese el número de puerto: 1654
Ingrese el nombre del contacto: Geiner
```

El contacto queda almacenado, procedemos a enviar un mensaje a este contacto. Se ingresa el nombre de la persona a la que se desea enviar el mensaje y el mensaje,

```
-----Bienvenido a WhatsApp-----  
  
----Agregar un nuevo contacto[1]---  
----Enviar un nuevo mensaje[2]-----  
----Salir[3]-----  
2  
  
Enviar mensaje a: Geiner  
Escriba su mensaje:  
Hola, como estas?
```

El mensaje ha sido enviado, y el menú vuelve aparecer. Volvemos a enviar un nuevo mensaje, y este es enviado, más adelante podemos observar que hemos recibido un nuevo mensaje,

```
fabiana@fabiana: ~/C/WhatsApp  
Hola, como estas?  
  
Mensaje enviado  
-----Bienvenido a WhatsApp-----  
  
----Agregar un nuevo contacto[1]---  
----Enviar un nuevo mensaje[2]-----  
----Salir[3]-----  
2  
  
Enviar mensaje a: Geiner  
Escriba su mensaje:  
Hola?  
  
Mensaje enviado  
-----Bienvenido a WhatsApp-----  
  
----Agregar un nuevo contacto[1]---  
----Enviar un nuevo mensaje[2]-----  
----Salir[3]-----  
Nuevo mensaje: Hola, como estas?  
|
```

La opción 3 nos permite salir del programa,

```
fabiana@fabiana: ~/C/WhatsApp

Mensaje enviado
-----Bienvenido a WhatsApp-----

----Agregar un nuevo contacto[1]---
----Enviar un nuevo mensaje[2]-----
----Salir[3]-----

2

Enviar mensaje a: Geiner
Escriba su mensaje:
Hola?

Mensaje enviado
-----Bienvenido a WhatsApp-----

----Agregar un nuevo contacto[1]---
----Enviar un nuevo mensaje[2]-----
----Salir[3]-----
Nuevo mensaje: Hola, como estas?
3
fabiana@fabiana:~/C/WhatsApp$
```

Conclusión Personal

En el desarrollo de este proyecto he adquirido conocimiento acerca del funcionamiento de aplicaciones que encontramos diariamente como WhatsApp, Messenger, o bien chats como el de Facebook, gmail, etc.

Durante este proceso, aprendí de una manera más práctica el funcionamiento del lenguaje de programación C, más conocimiento de su sintaxis y su compilación. Principalmente, el cuidado a la hora de disponer funciones que llaman a otras funciones, puesto que a la hora de compilar solo lee una vez el código, y solo se puede llamar dentro de una función a otra si esta ya esta escrita con anterioridad.

Con respecto al tema de los sockets, es un tema totalmente nuevo en el desarrollo de programas y logre entender el funcionamiento que estos tiene, así como el modo de realizarlos. Me parecen una herramienta sumamente prescindible para el manejo de datos de un computador a otro.

En términos generales, en esta tarea programada aprendí que el desarrollo de una aplicación, o de cualquier programa es posible alcanzarlo con la mentalidad "Divide y Venceras".