

TECNOLÓGICO DE COSTA RICA
ADMINISTRACIÓN DE TECNOLOGIA
DE INFORMACIÓN
Lenguajes de Programación
Profesor: Andrei Fuentes L

TPI

*Documentación
Externa*

elchatmasgenialdelmundo

Josué Espinoza C
Mauricio Gamboa C
Andrés Pacheco Q

Marzo, 2014

Tabla de Contenidos

• Descripción del problema.....	3
• Diseño del programa	4
• Librerías usadas.....	7
• Análisis de resultados	7
Objetivos alcanzados:	7
Objetivos no alcanzados:	8
• Manual de usuario	8
• Conclusión.....	12

El siguiente escrito pretende documentar la información del primer proyecto programado realizado en C.

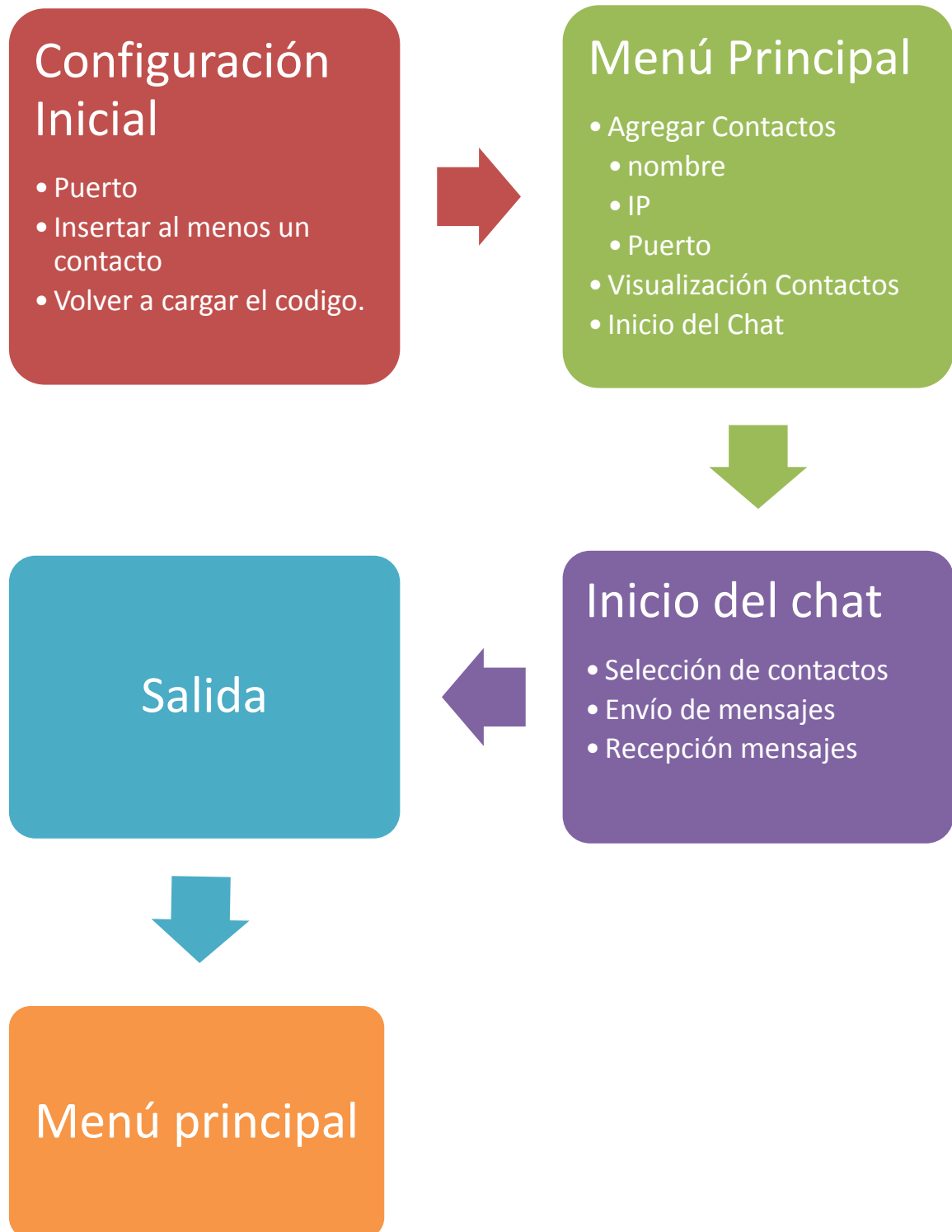
- **Descripción del problema**

Este proyecto consiste en la creación de un programa de mensajería instantánea similar a Google Chat, o a Facebook Messenger en C. Para la comunicación entre una computadora y otra se utilizarán sockets. Además para su completo entendimiento es necesario conocer los siguientes conceptos aplicados para la realización del chat.

- > IP
- > Puerto
- > Socket
- > Fork
- > Manejo de procesos (Programación de Múltiples Procesos)
- > Manejo de archivos
- > Modelo TCP/IP
- > Utilización de Github (Git)
- > Modelo OSI

- **Diseño del programa**

- › Agregar contactos: en esta opción si es por primera vez se decidió agregar a un contacto y para crear el archivo de contactos, el usuario debe de volver a ejecutar el código. Para almacenar los contactos se utilizó la estructura Struct donde se almacena el nombre, el puerto y la dirección IP.
- › Envío y recepción de mensajes: se utilizó el protocolo TCP/IP para la creación de los sockets. Para bifurcar los procesos se utilizó fork para el manejo de los procesos del cliente y del servidor. El servidor se encarga de la recepción de mensajes y el cliente del envío de los mismos
- › Enviar mensajes: el programa permitirá a los usuarios enviarle mensajes a alguno de sus contactos, para lo cual se especificará el nombre de usuario, y el mensaje, y el programa deberá enviar el mensaje al puerto y a la IP asociados a dicho usuario.
- › Recibir mensajes: el programa deberá estar escuchando en algún puerto para poder recibir mensajes. El puerto que usará el programa deberá estar especificado en un archivo de configuración (.txt). Este archivo de configuración solamente contiene el número del puerto.
- › Función sleep: se utilizó esta función en el cliente para esperar el cambio del proceso en el otro chat.
- › Mensajes con colores: para su realización se utilizaron macros donde se definieron los colores deseados.



Estructura de diseño

Configuración Inicial

- Ingresar el puerto de escucha
- Volver a cargar el programa

Configuración de Contactos

- Creación del archivo de contactos (internamente)
- Volver a cargar el programa

Ejecución del programa

- Menú de opciones
 - 1- Agregar un contacto
 - * Ingresar un nombre válido
 - * Ingresar una IP válida
 - * Ingresar un puerto de escucha del contacto
 - 2- Visualizar contactos
 - * Impresión de cada contacto
 - 3- Chat
 - * Búsqueda del contacto
 - * Bifurcación del código
 - * Proceso padre: servidor
 - > crea y configura los sockets (escucha y conexión)
 - > acepta conexión entrante
 - > muestra el mensaje leído
 - * Proceso hijo: cliente
 - > crea y configura el socket
 - > realiza la conexión
 - > envía un mensaje (cierra conversación si es el mensaje de salida)

• Librerías usadas.

Las librerías utilizadas para la creación del chat son las siguientes:

sys/types.h	Librería utilizada para la creación de sockets.
sys/socket.h	Librería utilizada para la creación de sockets.
stdio.h	Librería utilizada para las entradas y salidas.
unistd.h	Librería para obtener acceso a variables como PID.
netinet/in.h	Librería necesaria para la implementación de la función bind().
netdb.h	Librería para el uso de primitivas unix.
stdlib.h	Proporciona las opciones de entrada y salida, permite el uso de printf.
string.h	Se utiliza para manipulación de cadenas de caracteres.
ctype.h	Contiene funciones para clasificar caracteres según sus tipos o para convertir entre mayúsculas y minúsculas independientemente del conjunto de caracteres.

• Análisis de resultados

Objetivos alcanzados:

- Crear de manera correcta los sockets y la conexión de una máquina a otra.
- Conocer a fondo la función "fork" y lograr con éxito su implementación dentro del código.
- Utilizar el modelo "TCP/IP" y aplicarlo correctamente en la conexión de la tarea programada.
- Enviar y recibir mensajes, efectivamente, de un ordenador a otro por medio de un puerto y su IP.

- Agregar y consultar contactos al chat, guardándolos en un archivo “.txt”, y posteriormente, enviarles mensajes de la forma esperada.
- Lograr las funciones dentro de un mismo código de “servidor” y “cliente”, de manera que un usuario deba esperar el mensaje (función de servidor) y el otro envíe el mismo (función de cliente)

Objetivos no alcanzados:

- Al implementar la salida del chat, ésta únicamente se ejecuta en el usuario que ingresa a la conversación (función de cliente). Es decir, si se llama a la salida del chat (cualquiera de los dos usuarios), únicamente saldrá con éxito del usuario que efectúa la función de cliente. Por lo tanto el servidor sigue escuchando hasta que se presione 0 en el menú principal para salir de la aplicación.

• Manual de usuario

Para utilizar el chat se debe compilar primeramente el código, una vez realizado este paso desde la consola se debe de ejecutar el archivo, el programa solicitará que se ingrese el número del puerto por el cual se va a dar el envío de los mensajes. Después se debe de volver a ejecutar el código ya que es necesaria la creación del archivo de configuración. Seguidamente se solicitará la inserción de al menos un contacto, debe de ingresar el nombre del contacto, su dirección IP (Ej: 192.168.33.46) y el puerto del mismo. Nuevamente se debe de volver a correr el programa para permitir la creación del archivo de contactos.

Después de realizar el proceso de configuración señalado anteriormente, el chat está listo para su utilización. El sistema mostrará un menú principal donde se solicita seleccionar una de las tres opciones.

- | | | |
|------------------------|----------------------------|------------------|
| 1. Agregar
contacto | 2. Visualizar
Contactos | 3. Abrir el chat |
|------------------------|----------------------------|------------------|

En la opción 1 se debe de ingresar el nombre del contacto, su dirección IP y el puerto. En la opción 2 se desplegarán los contactos registrados. Y finalmente en la opción 3 se podrá iniciar una conversación o conectarse al chat solamente. Una vez iniciada una conversación donde antes se ingresó el nombre del contacto, se debe de presionar <enter> para iniciar el chat y luego enviar el mensaje deseado. Se debe de tomar en cuenta que el chat permite un mensaje a la vez, si se envían más estos llegarán después del mensaje del otro contacto. Para salir, se debe de enviar el mensaje <salir>.

A continuación se presentan tres capturas de pantalla que muestran el correcto funcionamiento del programa y su forma de uso.

```
ubuntu: ~/ProgramasC
andres@ubuntu:~/ProgramasC$ ./chat
Bienvenido al ChatMásGenialDelMundo

*****Menú Principal*****

Seleccione una de las siguientes opciones:
1- Agregar contactos
2- Visualizar contactos
3- Abrir el Chat
Presione 0 para salir de la aplicación.
1
Ingrese el Nombre del contacto a guardar (No ingrese espacios): Fernando
Ingrese el IP del contacto a guardar (No ingrese espacios): 192.168.0.112
Ingrese el Puerto del contacto a guardar (Número y No ingrese espacios): 23
45
Se ingresó --> CONTACTO # 3
Nombre = Fernando
IP = 192.168.0.112
Puerto = 2345

Se ha añadido el contacto correctamente.

1- Agregar contactos
2- Visualizar contactos
3- Abrir el Chat
Presione 0 para salir de la aplicación.
█
```

```
ubuntu: ~/ProgramasC
andres@ubuntu:~/ProgramasC$ ./chat
Bienvenido al ChatMÁSGenialDelMundo

*****Menú Principal*****

Seleccione una de las siguientes opciones:
1- Agregar contactos
2- Visualizar contactos
3- Abrir el Chat
Presione 0 para salir de la aplicación.
2
-----CONTACTO # 0
Nombre = Peque
IP = 192.168.33.107
Puerto = 2006

-----CONTACTO # 1
Nombre = Chimy
IP = 172.26.99.95
Puerto = 4005

-----CONTACTO # 2
Nombre = Yo
IP = 192.168.0.111
Puerto = 4000

-----CONTACTO # 3
Nombre = Fernando
IP = 192.168.0.112
Puerto = 2345

1- Agregar contactos
2- Visualizar contactos
3- Abrir el Chat
Presione 0 para salir de la aplicación.

```

```
andres@ubuntu: ~/ProgramasC
andres@ubuntu:~/ProgramasC$ ./chat
Bienvenido al ChatMÁSGenialDelMundo

*****Menú Principal*****

Seleccione una de las siguientes opciones:
1- Agregar contactos
2- Visualizar contactos
3- Abrir el Chat
Presione 0 para salir de la aplicación.
3
Escriba 1 para enviar mensaje, 2 para conectarse al chat: 1
Ingrese el nombre del contacto a mensajear: Yo

Conversación válida con: Yo

Prueba: texto = 192.168.0.111
Bienvenido al servidor. Digite <Enter> para iniciar o <Salir> para terminar
la conversacion.

hola
todo bien?
Sólo bueno, por dicha
Ah que bien. Qué me tacuen?
Diay no aquí haciendo la progra, vos??
Igual que vos :(
Que madre, nos hablamos entonces!
bueno mae nos vemos ;)
salir

1- Agregar contactos
2- Visualizar contactos
3- Abrir el Chat
Presione 0 para salir de la aplicación.
█
```

line: 175/507 col: 2 sel: 0 ins TAB mode: Unix (LF) encoding: UTF-8 filetype: C scope: cti

• Conclusión

Al trabajar por primera vez en el lenguaje C, fue interesante utilizar punteros y acostumbrarse a diferentes restricciones al no poder usar strings, sino más bien chars. Al igual, trabajar con sockets por primera vez permitió aprender tanto de su implementación y uso, así como la importancia de investigar diferentes conceptos antes de iniciar cualquier proyecto, especialmente para conocer realmente la estructura a desarrollar. Por otra parte, fue interesante conocer como simples asignaciones de variables, funciones y otros datos alteraban completamente el código ya que existe gran diferencia en el proceso de compilación con respecto a los lenguajes utilizados en otros cursos anteriormente. Finalmente, este proyecto programado permitió mejorar las habilidades personales tanto en la administración del tiempo, el trabajo en equipo y entre otras necesarias para el desarrollo del código.