

# Proyecto programado 1

## Creación clon whatsapp

Yaxiri Azofeifa García  
Carlos Campos Fuentes  
Fabricio Soto Mejías  
Michelle García Campos  
Daniela Valerio Cerdas

El objetivo es familiarizarse con el lenguaje de programación C. este se pretende con la implementación del clon de WhatsApp donde se deberá realizar la comunicación entre computadoras  
Y poder realizar el intercambio de mensajes por medio de sockets.



## Tabla de contenido

Descripción del programa: .....	1
Métodos Estudiados: .....	2
Socket: .....	2
TCP/IP(Transmisión Control Protocolo): .....	2
Método Fork: .....	2
Diseño del programa: .....	2
Decisión de diseño .....	3
Algoritmos usados (Funciones principales del programa): .....	3
Diagramas Lógico .....	4
Librerías de C utilizadas .....	4
Stdio.h: .....	4
Stdlib.h: .....	5
Unistd.h: .....	5
string.h: .....	5
sys/types.h: .....	5
sys/socket.h: .....	5
netinet/in.h: .....	5
netdb.h: .....	5
Análisis de resultados .....	6
Objetivos alcanzados .....	6
Manual de usuario .....	6
Conclusión Grupal .....	6

## Descripción del programa:

Al querer implementar un sistema de información al nivel computacional, es de vital importancia el manejo de los datos, para cubrir con las necesidades que requiere la implementación. Entre otros es importante un control eficaz, de la cual dependerá la concurrencia de la aplicación. El desarrollo de la tarea se establecerá bajo las reglas del lenguaje Imperativo C.

Las especificaciones de la tarea nos llevarán a desarrollar un sistema de comunicación entre dos nodos pertenecientes a una red específica, la cual permitirá el envío de mensajes entre los diferentes usuarios.

La comunicación debía usar el concepto de sockets, que nos permitía el intercambio de flujo de datos en el que es importante ciertos requisitos y características para que los programas puedan comunicarse, en la modalidad cliente-servidor, en el que podríamos utilizar dos protocolos, por un lado TCP orientado a conexión y por otro UDP no orientado a conexión.

La comunicación debe proveer al usuario la facilidad de recibir mensajes al mismo tiempo que lo está enviando, de la misma manera, tener conexión con uno o más nodos de la red. Esto se hará mediante el uso de la función *fork()*, el cual se encargara de copiar el mismo programa en dos distintos procesos, diferenciando a cada uno con un número de identificación, que nos permite crear procesos padres e hijos. Este proceso nos permite una bifurcación en el hilo de ejecución en un entorno multi-hilo.

## Métodos Estudiados:

**Socket:** Herramienta que ofrece a los sistemas operativos para comunicación entre diferencias procesos de flujo de datos, estos constituyen un mecanismo de entrega de paquetes de datos provenientes de una tarjeta de red a los procesos apropiados. En el control de socket es importante el manejo de ciertos requisitos para que la comunicación pueda darse con éxito por ejemplo: Conocer el IP de ambos programas, además del puerto.

**TCP/IP(Transmisión Control Protocolo):** En una red con este protocolo cada máquina está identificada por medio de su dirección IP que debe ser única. Sin embargo en cada máquina se encuentran múltiples procesos ejecutándose simultáneamente. Los procesos deben de ser asociados al número de puerto para identificar los paquetes que reciba la máquina. La conexión TCP tiene tres etapas: Establecimiento de conexión, transferencia de datos y fin de conexión.

**Método Fork:** Permite crear una copia del programa, generando un proceso hijo y uno padre los cuales serán ejecutados al mismo tiempo bajo un diferente PID. Esto lo permite lo que permite es crear un proceso encargado de realizar el envío de archivos hacia el servidor remoto y al mismo tiempo ejecutar el proceso de recepción el cual se encontrara activo hasta que una señal de recepción de archivo, así el mismo programa podrá realizar las dos tareas al mismo tiempo, bajo el mismo contexto y código.

## Diseño del programa:

A continuación de especificar el diseño usado en el programa, además de la manera en que se maneja las diferentes funciones implementadas.

## Decisión de diseño

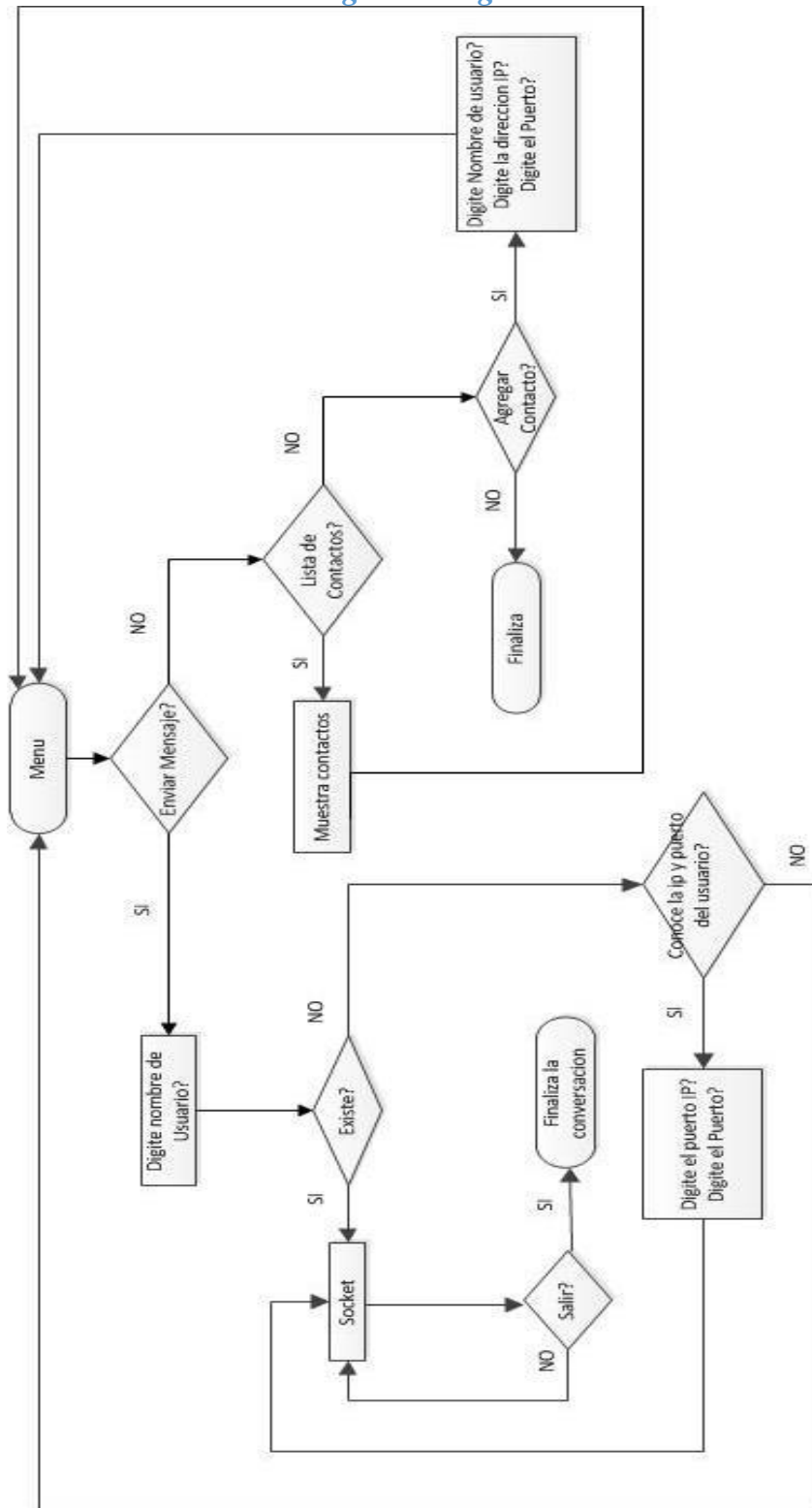
- Se desarrolló en el lenguaje de Programación C debido a que era un requisito de la tarea.
- Se decidió utilizar el protocolo orientado a conexión Proceso de Control de Transmisión (TCP, por sus siglas en inglés) ya que esto representaba mayor seguridad y confiabilidad en la transmisión de datos.
- Se utilizó un menú en el cual se le solicita al usuario insertar una selección mediante una serie de números para acceder a las diferentes funciones del programa.
- Se diseñó el programa de manera tal que pudiera correrse desde consola solamente.

### Algoritmos usados (Funciones principales del programa):

Nombre de Función	Funcionamiento	Valor de retorno
Lee()	Consiste en recorrer el documento .txt donde se almacenaron los contactos junto con la información necesaria. Se recorre mediante un puntero de tipo FILE, y junto con la implementación de fgets(), se lee cada línea del documento y se almacena la información necesaria para enviar mensajes en arreglos estáticos.	No existe valor de retorno, El resultado son datos recuperados del documento donde se encuentra la información de contactos.
Escribir()	Es la encargada de agregar al documento los contactos que son guardados por el usuario. Mediante fprintf(), se escribe en el archivo .txt, especificando durante el escaneo de datos que se vaya almacenando en una línea por dato insertado.	No existe un valor de retorno, lo que concluye es el almacenamiento de los datos insertados por el usuario en el documento de texto.
Sockets()	Es el encargado de abrir la conexión entre los usuarios para que estos puedan enviar y recibir mensajes. En esta función se implementa el fork(). Cuando los usuarios se comunican ingresan a un ciclo infinito, la comunicación se cierra si un usuario escribe la palabra "salir".	No posee valor de retorno, la función comunica a los usuarios.
Muestra()	Le corresponde recorrer el documento donde el usuario tiene la lista de contactos, y mostrar la información de los mismos. Para lograr el objetivo se modificó la función de lee().	No existe un valor de retorno, más cumple la función de mostrar en consola todos los datos que han sido almacenados.
ValidaContactos()	Consiste en verificar si el usuario que se está intentando agregar ya existe en el archivo de contactos.	El retorno de esta función es 1, si el usuario no existe en el documento, 0 si, el usuario se encuentra en el documento.

Fork()	<p>Hace referencia a la creación de una copia de sí mismo por parte de un programa, que entonces actúa como un "proceso hijo" del proceso original, ahora llamado "padre". Los procesos resultantes son idénticos, salvo que tienen distinto número de proceso (PID).</p> <p>Más generalmente, una bifurcación en un entorno multihilo significa que un hilo de ejecución se bifurca.</p>	No existe un valor de retorno
--------	---	-------------------------------

## Diagrama Lógico



## Librerías de C utilizadas

**Stdio.h:** Es la biblioteca estándar del Lenguajes de programación en C , el archivo de cabecera que contiene las definiciones de macros, las constantes, las declaraciones de funciones y la definición de tipos usados por varias operaciones estándar de entrada y salida. La biblioteca estándar stdio.h nos permite manipular ficheros desde nuestros programas en C, para lo cual hemos de incluir la siguiente directiva al comienzo de nuestros ficheros de código: `#include <stdio.h>`

**Stdlib.h:** La biblioteca estándar stdlib.h es el archivo de cabecera de la biblioteca estándar de propósito general del lenguaje de programación de C. Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras.

Las funciones de la biblioteca stdlib.h pueden ser agrupadas en las categorías de:

- Conversión de tipos
- Generación de números pseudo-aleatorios
- Gestión de memoria Dinámica
- Control de procesos
- Ordenamiento y Búsqueda
- Aritmética.

**Unistd.h:** librería para obtener acceso a variables.

**string.h:** Es un archivo de la biblioteca estándar del lenguaje de programación en C que contiene la definición de macros, las constantes, funciones y tipos de utilidad para trabajar con cadenas de caracteres y algunas operaciones de manipulación de memoria. Se utiliza para la creación de sockets y las correspondientes constantes.

**sys/types.h:** Es una librería estándar del lenguaje de programación C. Esta librería define los tipos de datos de sistemas primitivos. Se define los tipos de datos utilizados en el código fuente del sistema. Debido a que algunos tipos de datos del sistema son accesibles a código de usuario, que pueden ser utilizados para mejorar la portabilidad a través de diferentes máquinas y sistemas operativos. También incluye las definiciones de tipos de enteros sin signo y direcciones.

**sys/socket.h:** Esta estructura se utiliza para definir una dirección de conector que se utilizara en las funciones. La cabecera `<sys/socket.h>` definirá la estructura. Esta estructura será la siguiente:

- Lo suficientemente grande como para dar cabida a todas las estructuras de dirección de protocolo específicos apoyados.
- Alineados en un límite adecuado para que los punteros a que pueden ser emitidos como punteros a estructuras de direcciones específicas de protocolo y se utilizan para acceder a los campos de las estructuras sin problemas de alineación.

**netinet/in.h:** La estructura funciona como una familia de direcciones de internet.

**netdb.h:** La cabecera `<netdb.h>` puede definir el tipo `in_port_t` y `in_addr_t` tipo descrito en



<netinet/in.h>. La cabecera <netdb.h> definirá la estructura **hostent** que incluye al menos los siguientes miembros:

- nombre oficial del host
- Tipo de dirección
- la longitud de la dirección
- Dirección estructurada de información
- Direcciones errores de información

## Análisis de resultados

### Objetivos alcanzados

- Se logró trabajar bajo el modelo OSI y el protocolo TCP que se requería para el proyecto programado.
- Se alcanzó la correcta creación y enlace de los sockets en la tarea Programada I.
- Se logró el almacenamiento de datos en los archivos de configuración.
- Se consiguió la lectura de datos almacenados para la implementación de los sockets.
- A la hora de pasar de computador a computador se alcanzó eficazmente el envío y recepción de mensajes mediante una red inalámbrica doméstica.
- La bifurcación, el proceso en recepción y envío simultáneamente, en otras palabras la implementación del Fork.
- Se obtuvo con éxito el makefile

### Manual de usuario

Se encuentra adjunto con el nombre Manual de Usuario.pdf.

## Conclusión Grupal

El proyecto programado 1, requirió de mucha investigación y análisis de comportamientos de gran cantidad de funciones. Además se necesitó de empeño y tiempo para alcanzar objetivos.

Se implementaron muchos de los temas vistos en clase y se aumentó el conocimiento de la gran mayoría de estos.

Se notó claramente con la investigación, la importancia de la implementación de sockets y fork, ya que actualmente el tema de las comunicaciones se encuentra en crecimiento.

Cabe mencionar que uno de las implementaciones más difíciles del proyecto fue la bifurcación (envió y recepción simultanea) en cuanto a los forks.