

Chat seguro

Manual de usuario

Contenido

[Contenido](#)

[Aspectos generales](#)

[Requerimientos importantes](#)

[Instalación de Protocol Buffer en Linux](#)

[Instalación manual de Protobuf](#)

[Instalación de Protobuf por ShellScript](#)

[Compilar nuevamente la aplicación](#)

[Ejecución del chat](#)

Aspectos generales

La aplicación ChatSeguro le permitirá entablar comunicación con una persona por medio de una red local. Al usar el Chatpodrá intercambiar mensajes cifrados que sólo usted y la persona con la que conversa podrán leer. La aplicación hace uso de un algoritmo de cifrado y un protocolo de intercambio de claves de sesión, lo cual dificultará a personas mal intencionadas la obtención de los mensajes que intercambia con su compañero(a) de chat.

Requerimientos importantes

- Sistema basado en Linux o Unix
- Compiladores gcc y g++
- Protocol Buffer v2.0 +

Instalación de Protocol Buffer en Linux

Protocol buffer o *Protobuf* de google es un “lenguaje neutral” que ofrece un mecanismo extensible para la serialización de datos estructurados. Es similar a XML aunque más simplificado y rápido. En el chat seguro juega un papel muy importante ya que los mensajes están definidos usando esta herramienta.

Para la instalación y configuración de protobuf es necesario seguir los pasos mostrados en alguna de las secciones detalladas a continuación.

NOTA: Es importante que, la ejecución de los comandos mostrados, se realice en modo superusuario o con el usuario root del sistema.

Instalación manual de Protobuf

1) Descargar protobuf-2.6.1 (o su versión más reciente) desde:
<https://developers.google.com/protocol-buffers/docs/downloads>

- 2) Descomprimir el archivo
- 3) Abrir la aplicación **Terminal** en su sistema Linux
- 4) Entrar a la carpeta generada con la ejecución del paso 2.
- 5) Ejecutar los comandos:

```
$ ./configure --prefix=/usr
$ make
$ make check
```

- 6) Después de ejecutar make check, debería aparecer algo como esto:

```
=====
Testsuite summary for Protocol Buffers 2.6.1
=====
# TOTAL: 5
# PASS: 5
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

- 7) Ejecutar el comando:

```
$ sudo make install
```

Al finalizar la instalación podrá ejecutar y ,si lo desea, compilar nuevamente la aplicación.

Instalación de Protobuf por ShellScript

- 1) Copie el siguiente código y guárdelo con la extensión sh. Por ejemplo, “protoInstall.sh”

1	<code>wget</code>
	<code>https://github.com/google/protobuf/releases/download/v2.6.1/protobuf-2.6.1.tar.gz</code>
2	<code>tar xvzf protobuf-2.6.1.tar.gz</code>
3	<code>cd protobuf-2.6.1</code>
4	<code>./configure --prefix=/usr</code>
5	<code>make</code>
6	<code>make check</code>
7	<code>sudo make install</code>

- 2) Abra la aplicación **Terminal** en su sistema Linux.
 - 3) Cambie el directorio de trabajo en la terminal por el directorio en que desea descargar los archivos necesarios para la instalación.
 - 4) Ejecute el shell script que creó en el paso 1 con el comando `bash`. Para el caso mostrado en este manual el comando sería: `bash protoInstall.sh`.
- Al finalizar la instalación podrá ejecutar y ,si lo desea, compilar nuevamente la aplicación.

Compilar nuevamente la aplicación

Antes de proceder con la compilación de la aplicación asegúrese de satisfacer los requerimientos mostrados en la sección anterior *“Requerimientos importantes”*.

- 1) Abra la aplicación **Terminal** en su sistema Linux.
- 2) Cambie el directorio de trabajo en la terminal por el directorio de la aplicación.
- 3) Ejecute con el comando bash el archivo Make.sh del siguiente modo:

```
bash Make.sh
```

1	#Correr en terminal con: bash Make.sh
	echo ""
	echo " Chat Seguro "
2	echo ">> Compilando fuentes... "
3	if [-f obj/Chat.o];
4	then
5	echo ">> Chat.o Borrando..."
6	rm obj/Chat.o
7	fi
8	
9	echo ">> Chat.cpp Compilando... "
10	g++ Chat.cpp -c proto/MPI.pb.cc proto/MPI.pb.h -L /usr/lib/ -I
11	/usr/include/ -lprotobuf -pthread
12	mv Chat.o obj/
13	
14	echo ">> Anfitrión.cpp Compilando... "
15	g++ Anfitrión.cpp obj/Chat.o proto/MPI.pb.cc proto/MPI.pb.h -L
16	/usr/lib/ -I /usr/include/ -lprotobuf -pthread -o ANFITRION
17	
18	echo ">> Invitado.cpp Compilando... "
19	g++ Invitado.cpp obj/Chat.o proto/MPI.pb.cc proto/MPI.pb.h -L
20	/usr/lib/ -I /usr/include/ -lprotobuf -pthread -o INVITADO
21	
22	echo "Proceso terminado."
23	
24	
25	
26	

La ejecución de los pasos anteriores puede verse en la siguiente imagen:

```

isaac@ArgusPC: ~/Desktop/ChatSeguro
File Edit View Search Terminal Help
isaac@ArgusPC:~$ cd Desktop/ChatSeguro/
isaac@ArgusPC:~/Desktop/ChatSeguro$ bash Make.sh

>>Compilando fuentes...
>>Borrando Mensaje.o
>>Borrando Chat.o
>>Compilando Mensaje.cpp
>>Compilando Chat.cpp
>Compilando Anfitrión.cpp
>>Compilando Invitado.cpp
Invitado.cpp: In function 'int main(int, char**)':
Invitado.cpp:54:21: warning: deprecated conversion from string constant to 'char
*' [-Wwrite-strings]
    char* host_name="127.0.0.1";
                        ^
Proceso terminado.
isaac@ArgusPC:~/Desktop/ChatSeguro$

```

El compilador puede mostrar algunos mensajes de tipo Warning. Ésto puede deberse a la versión del compilador.

Ejecución del chat

- 1) Abra la aplicación **Terminal** en su sistema Linux.
- 2) Cambie el directorio de trabajo en la terminal por el directorio de la aplicación.
- 3) Si desea iniciar sesión y esperar a que un compañero de chat se una, ejecute la aplicación en modo “anfitrión” con el comando de ejecución

```
$ ./ANFITRION {1/0}
```

Donde la opción 1 habilitará el modo de demostración, por lo que se mostrarán varios aspectos del chat durante su ejecución.

La opción 0 permite mostrar la interfaz simple del chat, es decir, sólo se mostrará la conversación y los mensajes en claro.

- 4) En caso de querer unirse a un usuario que ya está esperando ejecute la aplicación en modo “anfitrión” con el comando de ejecución

```
$ ./INVITADO direcciónIP {1/0}
```

Donde direcciónIP corresponde a la dirección del anfitrión con quien se desea chatear.

La opción 1/0 sigue el mismo principio mencionado en el paso 3.

- 5) Finalmente, posterior a la ejecución de la aplicación en 2 terminales diferentes, los usuarios podrán intercambiar mensajes hasta que uno detenga la aplicación.

```
root@ArgusPC: /home/isaac/Desktop/CHAT
File Edit View Search Terminal Help
root@ArgusPC:/home/isaac/Desktop/CHAT# ./ANFITRION 1
Nombre de Usuario: Isaac
Esperando alguien con quien conversar...
-----
Petición de chat desde 192.168.1.79
-----

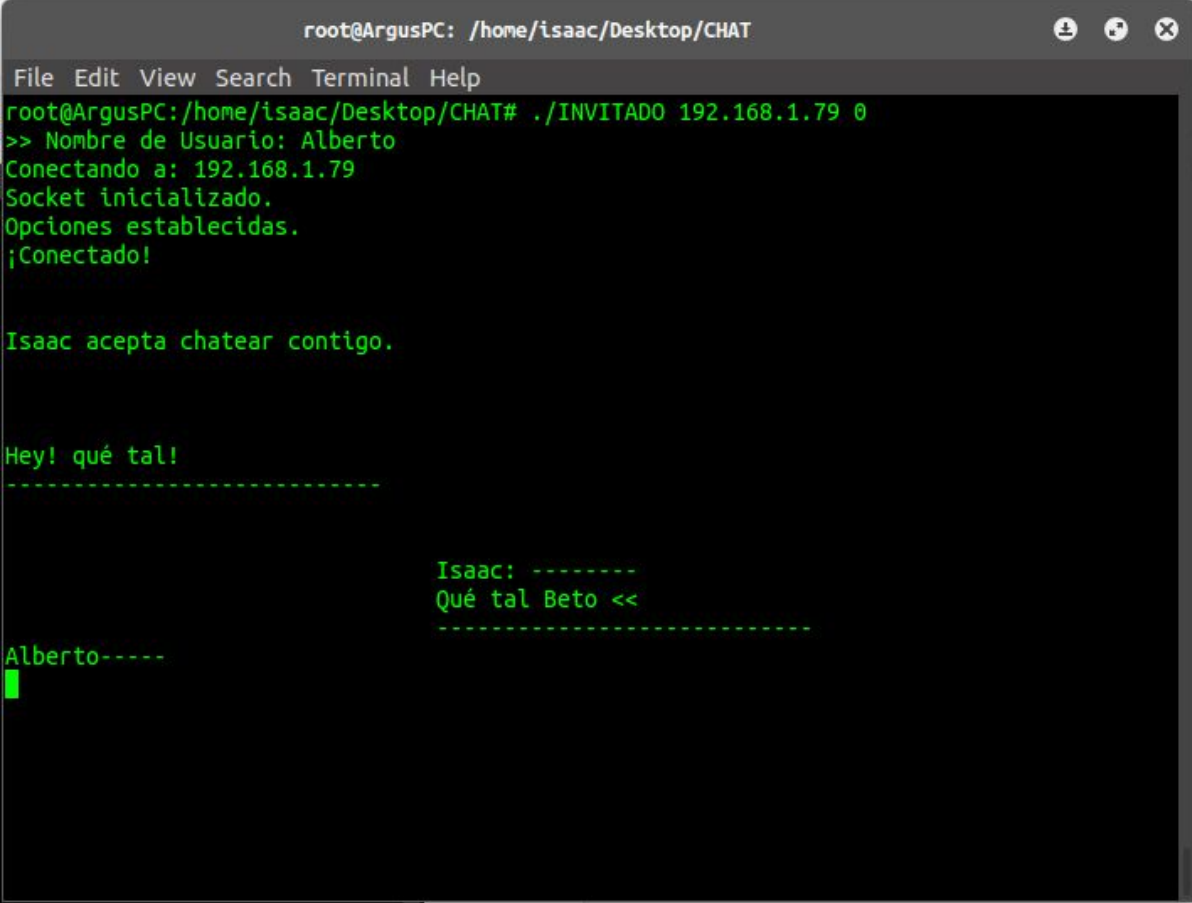
Alberto quiere chatear contigo.

Punto recibido:
-----
Contenido de mensaje:
msg_id: 2
dpt {
  x_val: 3
  y_val: 106
  tipo: 1
}
-----
Parametros actuales:
a: 5
p: 113
pxPub: 39
pyPub: 14
n: 67
Llave Sesión 10 bits: 534

                                Alberto: -----
                                Hey! qué tal! <<
                                -----

Isaac-----
Qué tal Beto
-----
Isaac-----
>> █
```

Ejecución del chat como Anfitrión



```
root@ArgusPC: /home/isaac/Desktop/CHAT
File Edit View Search Terminal Help
root@ArgusPC:/home/isaac/Desktop/CHAT# ./INVITADO 192.168.1.79 0
>> Nombre de Usuario: Alberto
Conectando a: 192.168.1.79
Socket inicializado.
Opciones establecidas.
¡Conectado!

Isaac acepta chatear contigo.

Hey! qué tal!
-----

Isaac: -----
Qué tal Beto <<
-----

Alberto-----
█
```

Ejecución del chat como Invitado