



HI-CHAT

Chat Puerto Serie

Maxi Guzman
Maxiguz17@gmail.com

Contenido

- Introducción 2
- Puertos serie 3
- Ficha DB9 hembra 3
- Dev C ++..... 4
 - Librería winbgim.h 4
- Programa 5
 - Pseudocódigo 6
 - Inicio 6
 - Fin 6
- Propuesta de Proyecto 7
- Funcionamiento específico del programa..... 8
- Conclusión 15
- Bibliografía 16

Introducción

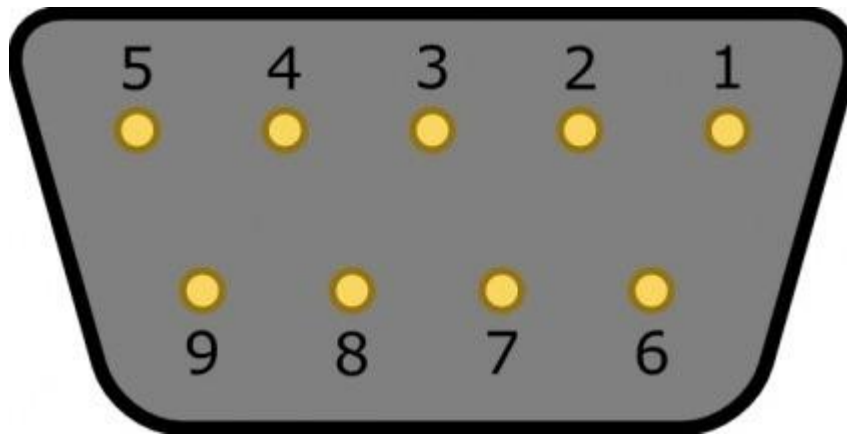
En el informe que se presenta a continuación se mostraran los procedimientos para confeccionar una comunicación entre dos computadoras por medio de un puerto serie o RS-232, utilizando el programa DEV C++, el cual presentará una interfaz gráfica que nos permitirá interactuar entre las mismas.

Seguido de esto detallaré brevemente cada uno de los componentes y su funcionamiento necesario para lograrlo.

Puertos serie

Los puertos serie posibilitan la transmisión de datos entre computadoras, lo que caracteriza a este tipo de puerto es que manda la información bit por bit.

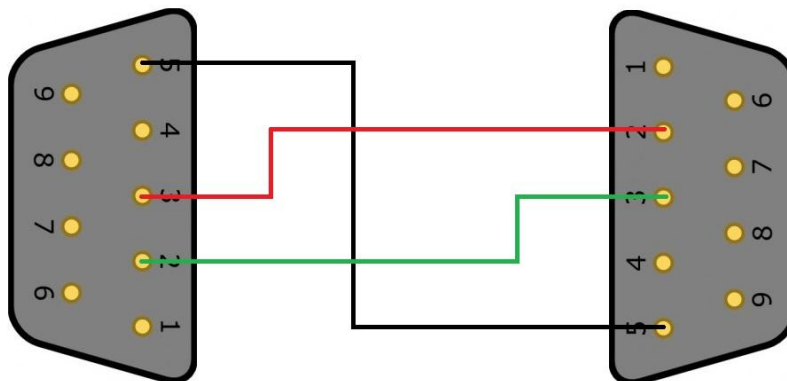
Para lograr esa comunicación se inicia con un bit inicial, antes de primer byte de la letra o palabra que se quiere enviar, al finalizar el envío, se manda inmediatamente un bit de alto al final de la cadena enviada.



Ficha DB9 hembra

Como se observa en la imagen, el conector cuenta con 9 pines, de los cuales para nuestro proyecto hemos usado solamente tres, que son pin N° 2, pin N° 3 y pin N° 5 de los cuales el pin 2 es el que recibe los datos, el pin 3 es el que envía los datos y el pin 5 es el GND.

Para nuestro proyecto nos vimos en la necesidad de hacer un cable con dos fichas DB9 hembra a las cuales les cruzamos los cables entre los pines 2 y 3 para que envíen y reciban correctamente las computadoras como se mostrara en la siguiente imagen.



Dev C ++

El Dev C++ es un programa que nos permite generar algoritmos en lenguaje C/C++, para diversas utilidades. En este caso manejar puertos seriales entre dos maquinas, el mismo contiene herramientas tales como librerías que habilitan distintas funciones, tales como la libreria winbgim.h que emula funciones de graphics.h, siendo la misma una librería de borland C++. Dicha herramienta la utilizamos para generar la parte grafica del programa.

Librería winbgim.h

La librería grafica winbgim.h ha sido creada originalmente por Konstantin Knizhnik's winbgi shareware posteriormente modificada por Mark Richardson y Michael Main, esta librería tiene como objetivo emular la librería graphics.h de Borland C++ y facilitar algunas funciones del Mouse. La ventaja de esta librería es que nos permite crear aplicaciones gráficas con compiladores GNU tales como el ambiente de desarrollo Dev C++ de la misma manera como se crean en el compilador comercial Borland C++.

Programa

Se basa en el tratamiento de archivos por medio de puertos, trabajándolos tanto en escritura como en lectura. Donde pueden ser enviados o recibidos por medio de bits, pudiendo así realizar una comunicación eficaz.

La configuración de puertos que utilicé es la siguiente:

```
h=CreateFile("COM1",GENERIC_READ|GENERIC_WRITE,0,NULL,OPEN_EXISTING,0,NULL);

if(h == INVALID_HANDLE_VALUE) {
    /* ocurrió un error al intentar abrir el puerto */
}

if(!GetCommState(h, &dcb)) {
    /* error: no se puede obtener la configuracion */
}

dcb.BaudRate = 9600;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
dcb.StopBits = ONESTOPBIT;
dcb.fBinary = TRUE;
dcb.fParity = TRUE;

if(!SetCommState(h, &dcb)) {
}
SetCommMask(h, EV_RXCHAR);

COMMTIMEOUTS timeouts;
timeouts.ReadIntervalTimeout = 50;
timeouts.ReadTotalTimeoutMultiplier = 1;
timeouts.ReadTotalTimeoutConstant = 1;
timeouts.WriteTotalTimeoutMultiplier = 1;
timeouts.WriteTotalTimeoutConstant = 1;
if(!SetCommTimeouts(h,&timeouts))
printf("Error al intentar abrir el puerto");
```

Pseudocódigo

Inicio

- Configurar Puertos
- Abrimos ventana grafica
- Mientras no se presione ninguna tecla
- Leer archivo de puerto
- Mostrar el “mensaje”
- Fin Mientras
- Mientras no se presione Enter
- Generar cadena y mostrar en pantalla
- Fin Mientras
- Desde Inicio de cadena
- Hasta Fin de la cadena
- Enviar cadena de texto

Fin

Lo que hace el programa es estar constantemente en espera verificando si existe algún tipo de dato hasta que se oprima una tecla. Una vez ocurrido esto el usuario podrá escribir su mensaje, que luego será guardado en una cadena de caracteres por la acción del comando “kbhit”. La misma será enviada por medio de la función “writefile” y volver a revisar, a la espera de algún otro dato con el “readfile”.

El comando “COMMTIMEOUTS” (timeouts) nos permite evitar que el programa se trabe sino recibe datos en un determinado intervalo de tiempo, saltando la acción de readfile.

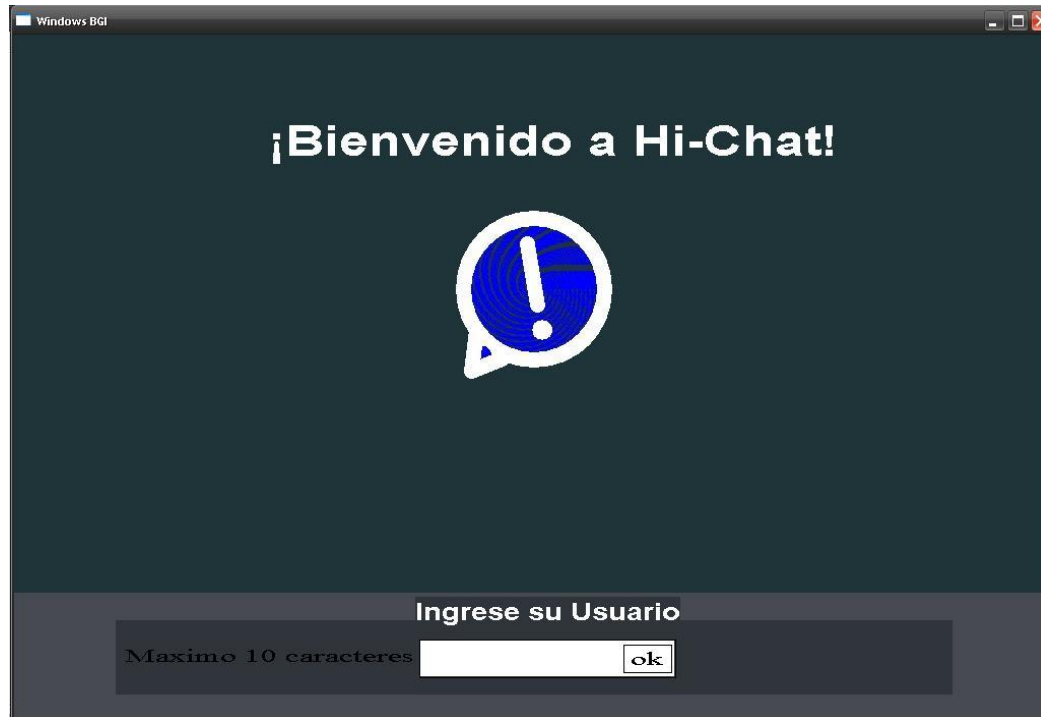
Propuesta de Proyecto

La propuesta de trabajo fue crear un chat, el cual se basaría de una primera ventana grafica que me da la bienvenida al programa, donde podré cargar mi usuario para luego ir a la pantalla principal, donde se llevará a cabo el inicio de la comunicación; o bien salir del chat si fue abierto por equivocación.

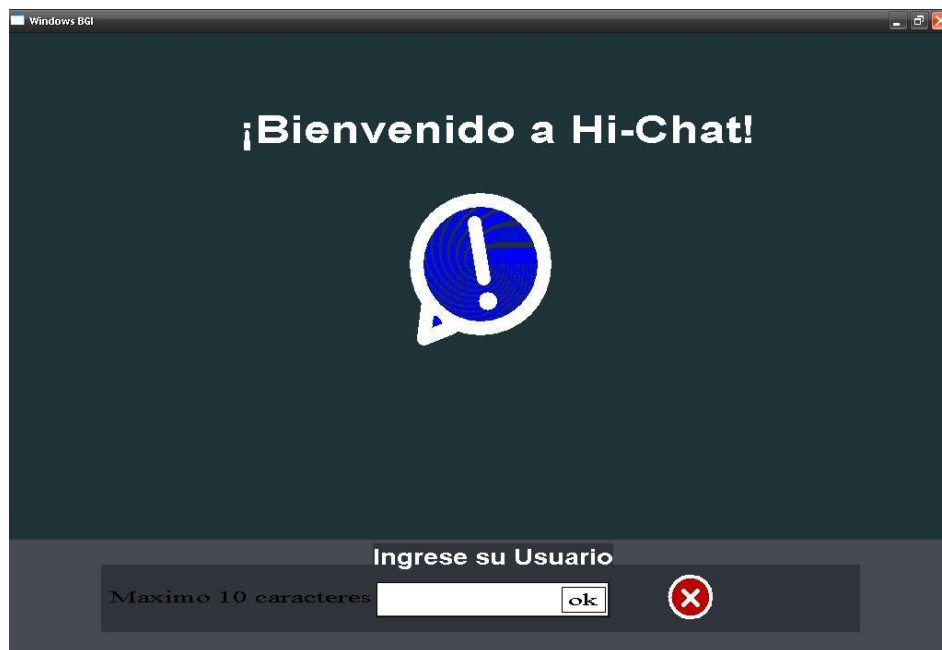
Al cargar mi usuario me dirigirá a la pantalla principal donde podré establecer dicha comunicación, donde existen diferentes opciones.

Funcionamiento específico del programa

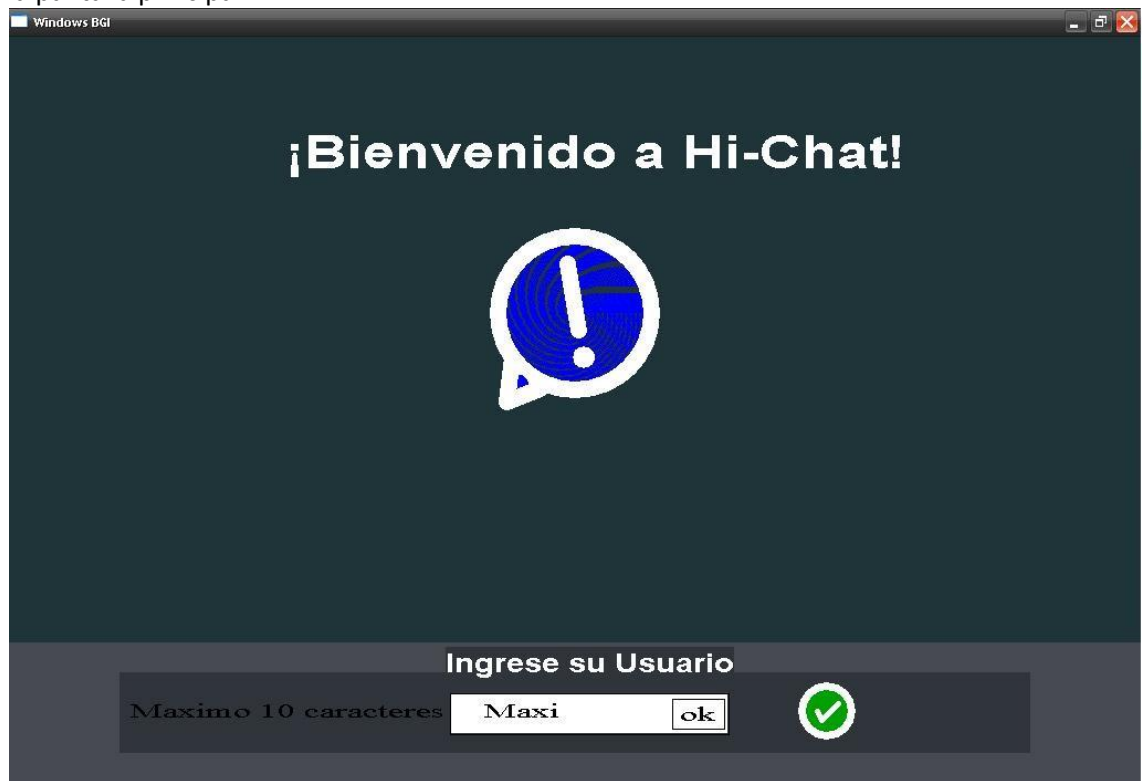
Como comenté anteriormente de forma breve, el programa se basará en dos pantallas. La primera dando la bienvenida al programa. Pidiéndome cargar un nombre de usuario, con un máximo de 10 (diez) caracteres.



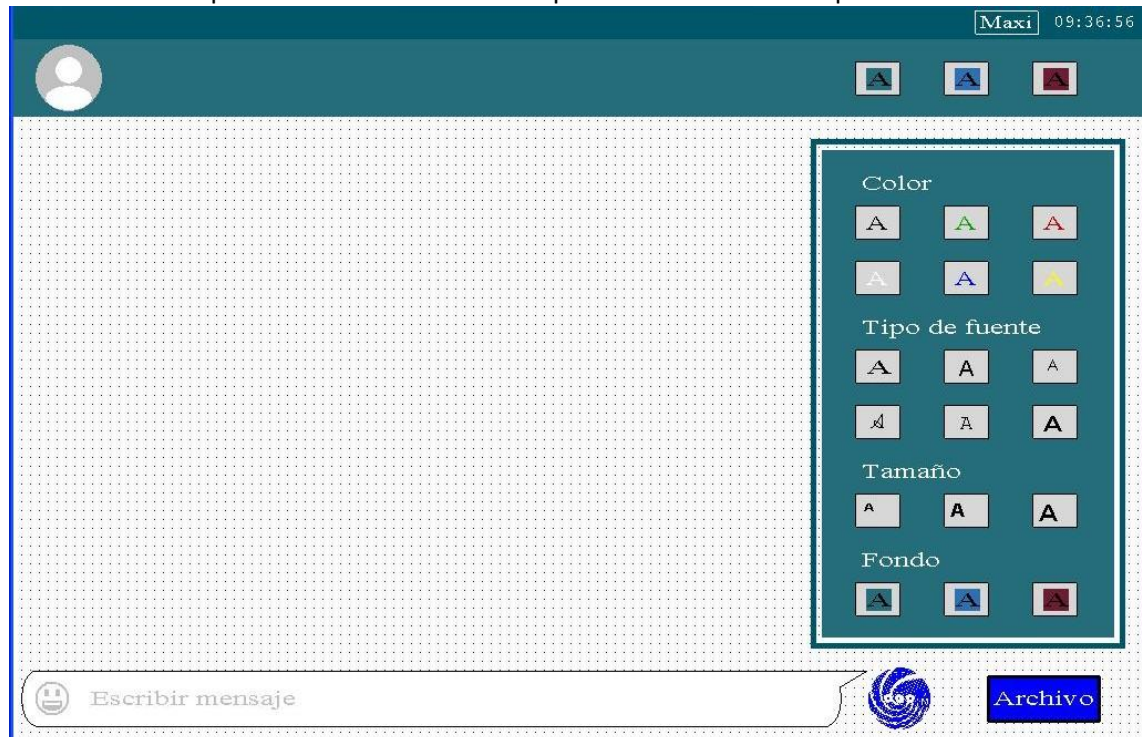
En caso que no ingrese ningún carácter en el usuario y se presione la tecla "ENTER" o el "CLIK", sobre el botón OK, no nos permitirá ingresar a la ventana principal mostrándonos un símbolo con una cruz. Haciendo referencia a lo dicho anteriormente.



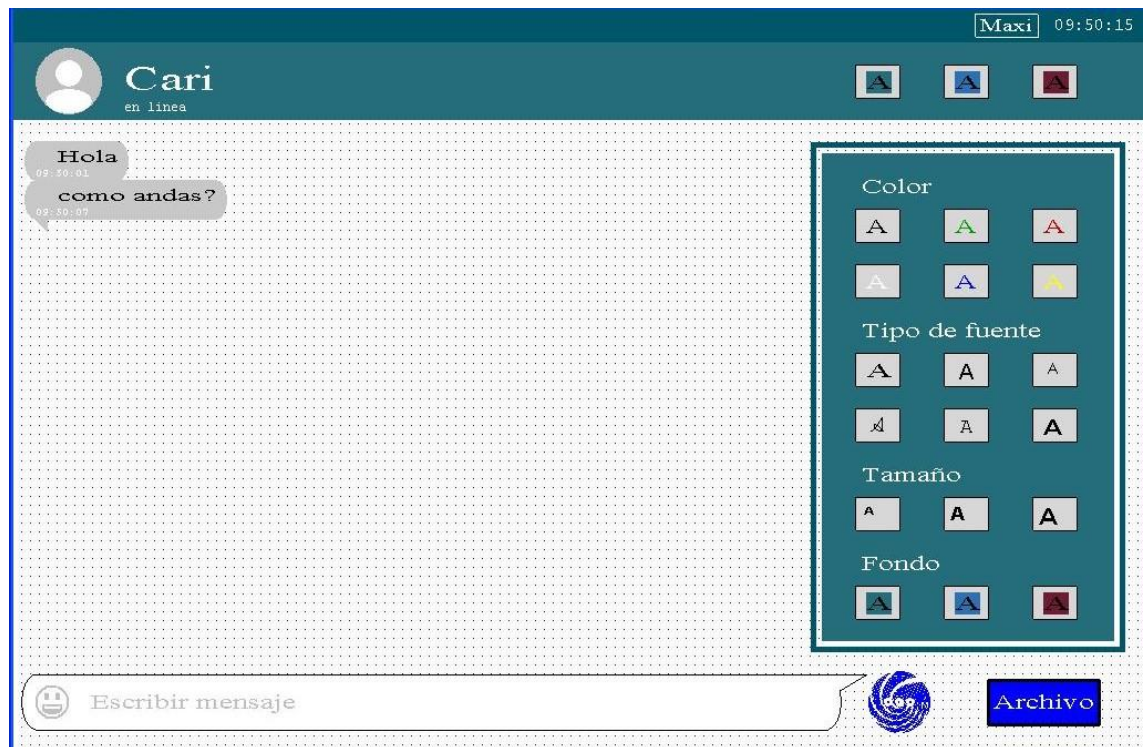
Por contrario, si ingreso mi usuario, me dará el okey (mostrándome una tilde) para pasar a la pantalla principal.



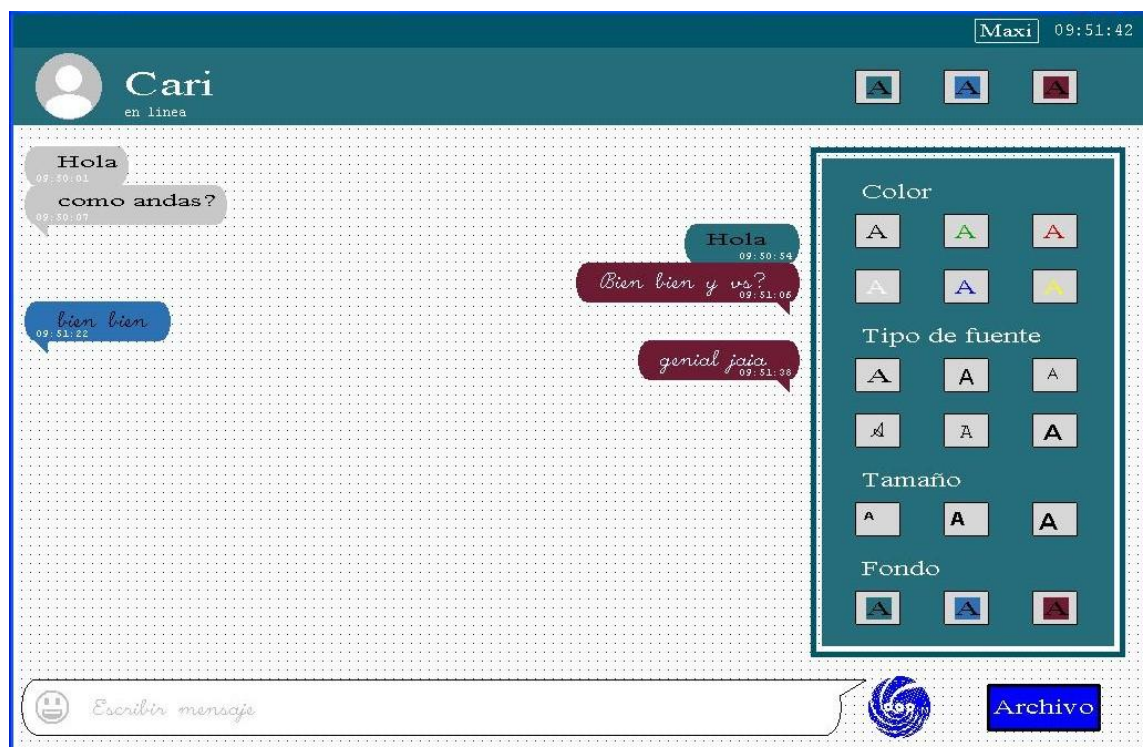
Al ingresar a la pantalla principal visualizare mi usuario. Junto a él observaré la hora en tiempo real. También se puede ver a la derecha de la pantalla un cuadro de opciones.



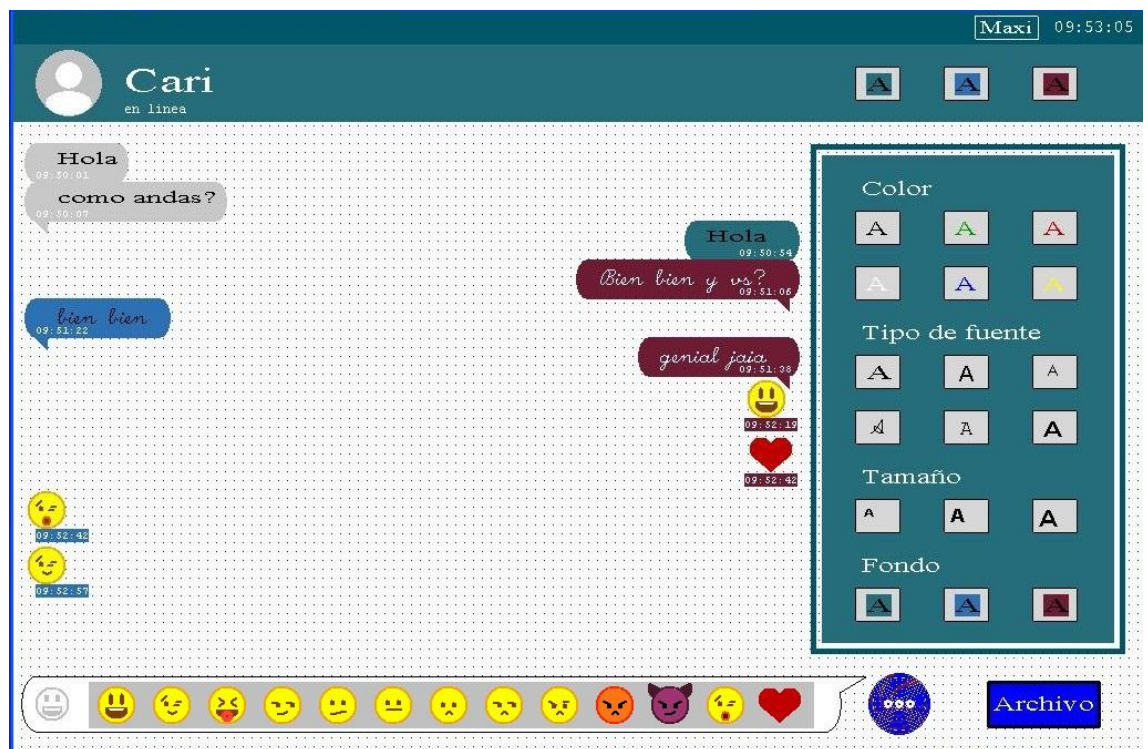
Una vez iniciada la comunicación, observo el usuario del otro equipo y me informa si está conectado.



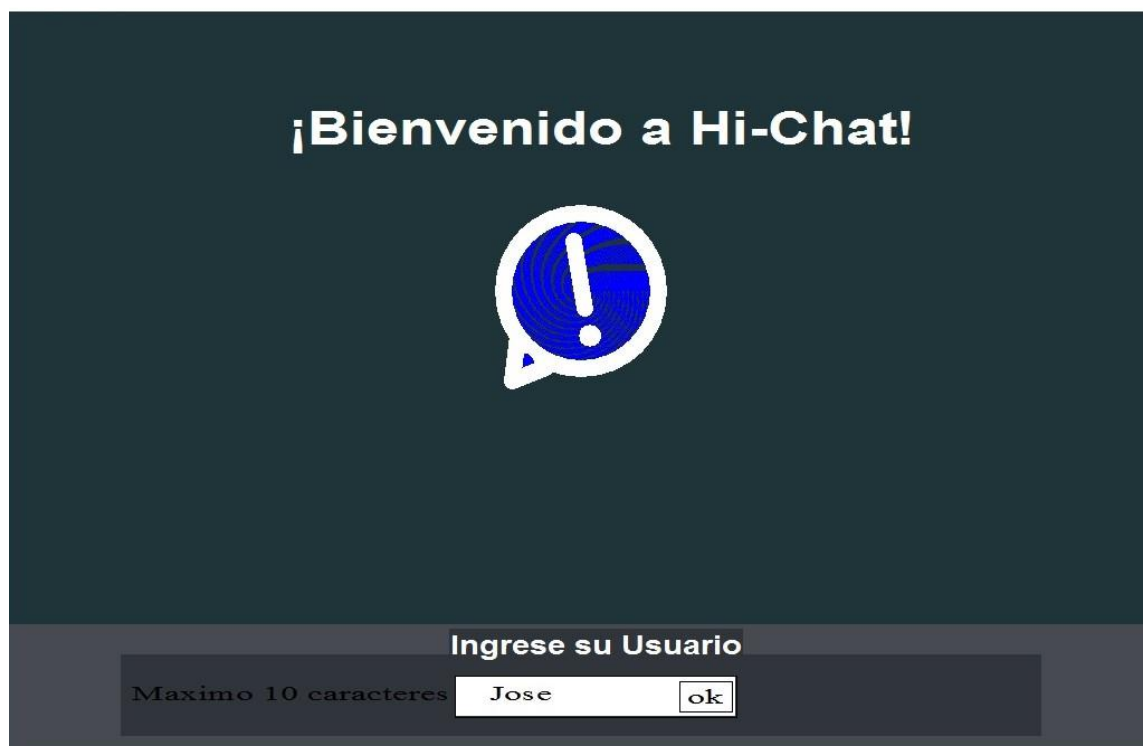
En este caso se muestra el uso de las diferentes opciones. Donde también me encuentro con que puedo cambiar el color de fondo de lo que recibo, con los botones que se ubican a la derecha debajo de mi usuario.



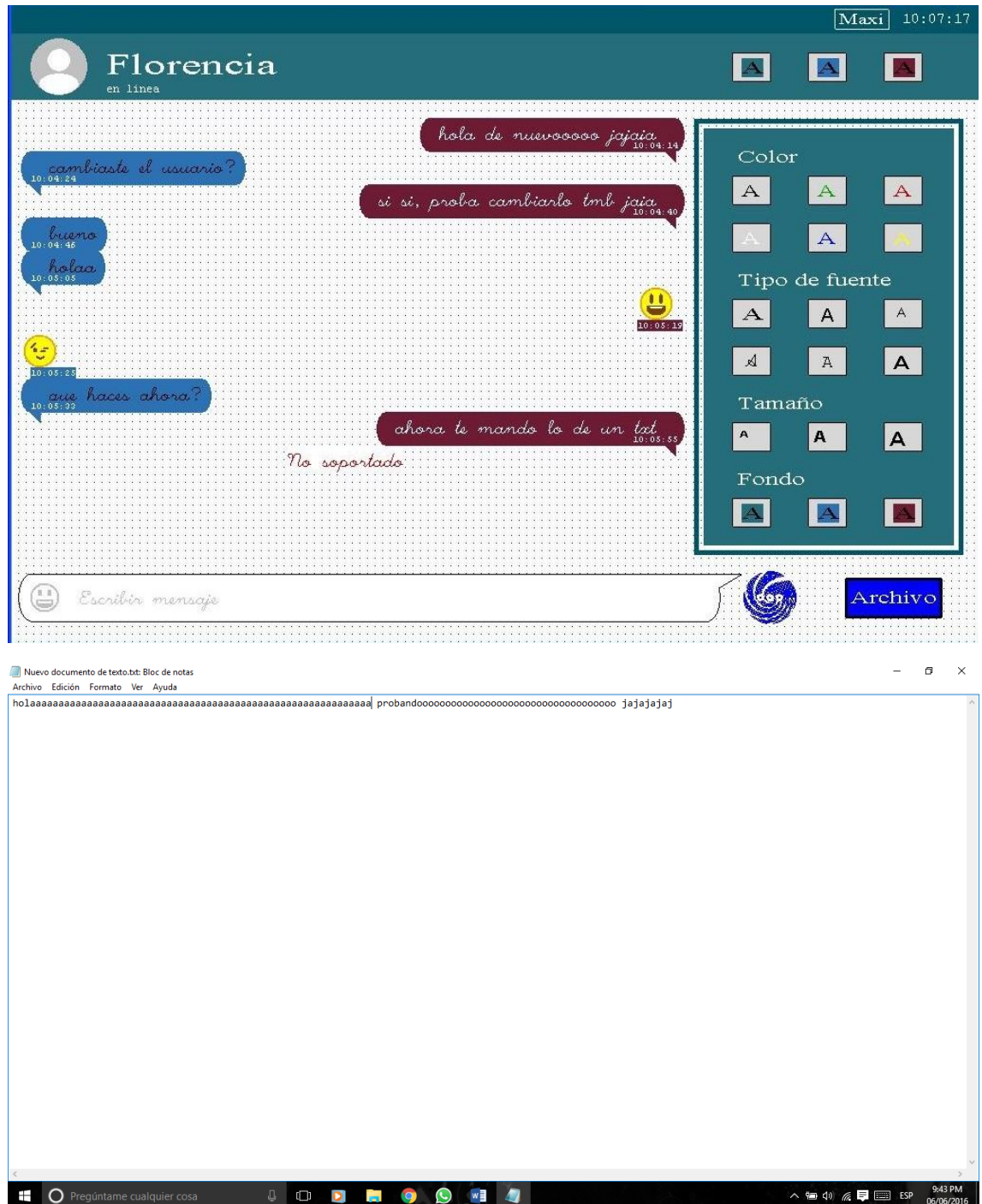
También al presionar el CLICK sobre la carita que está en el extremo inferior izquierdo, visualizo las opciones de emojis.

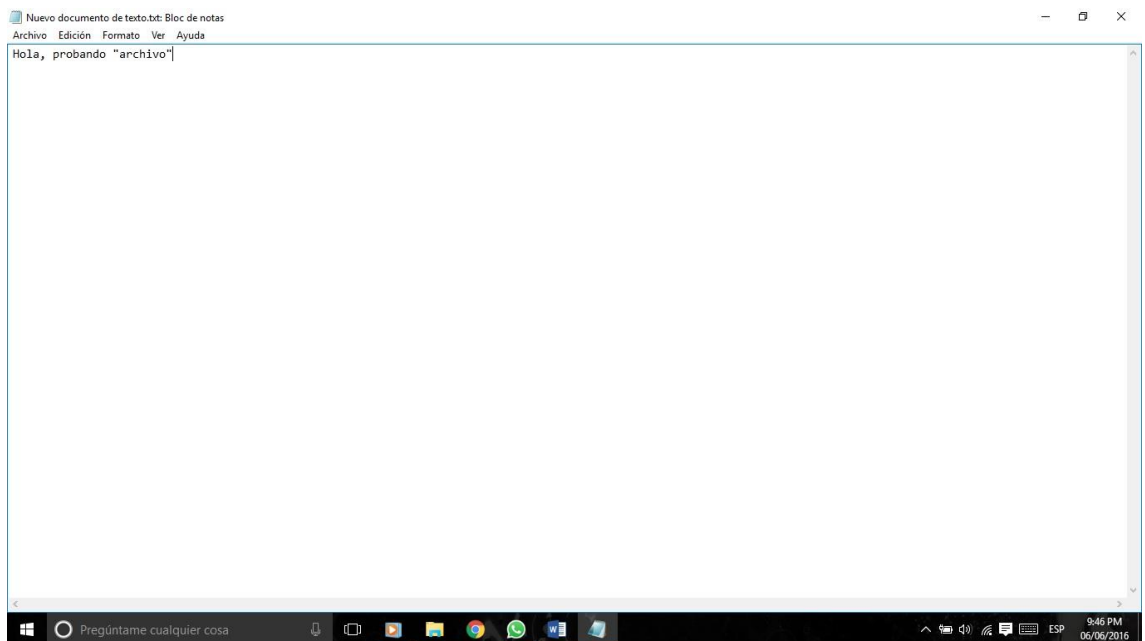


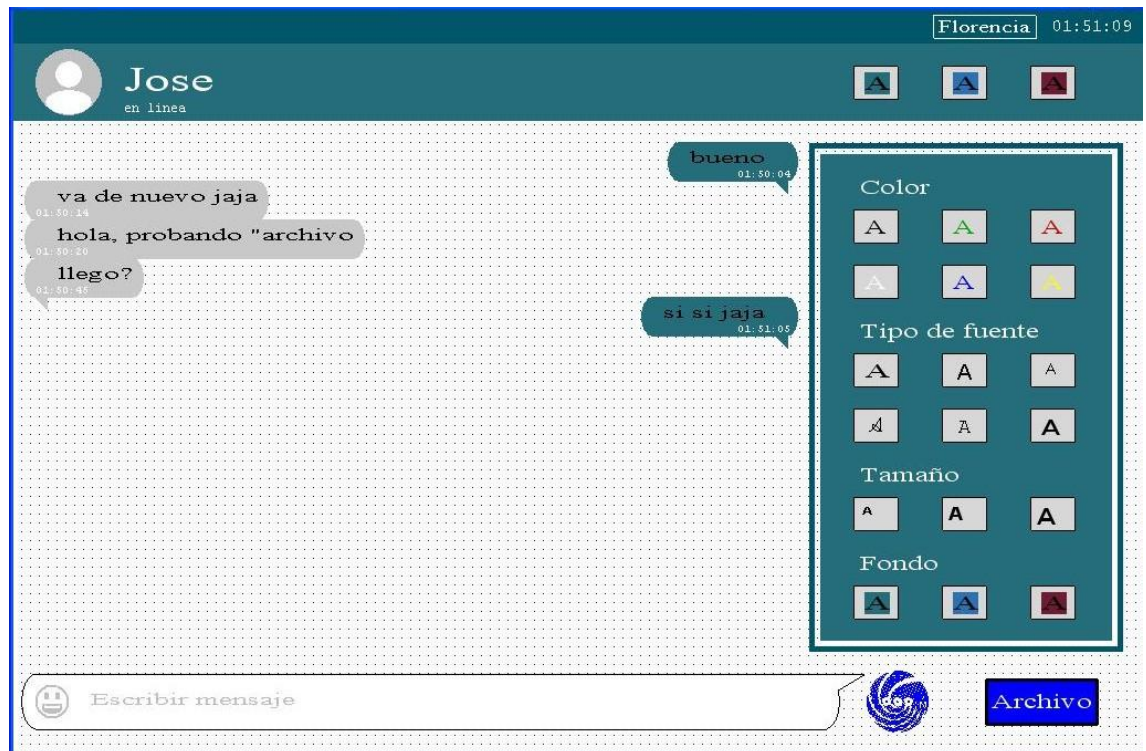
Otra opción es apretar el CLICK sobre mi usuario y me lleva a otra pantalla donde ingreso uno nuevo, para luego seguir con la comunicación.



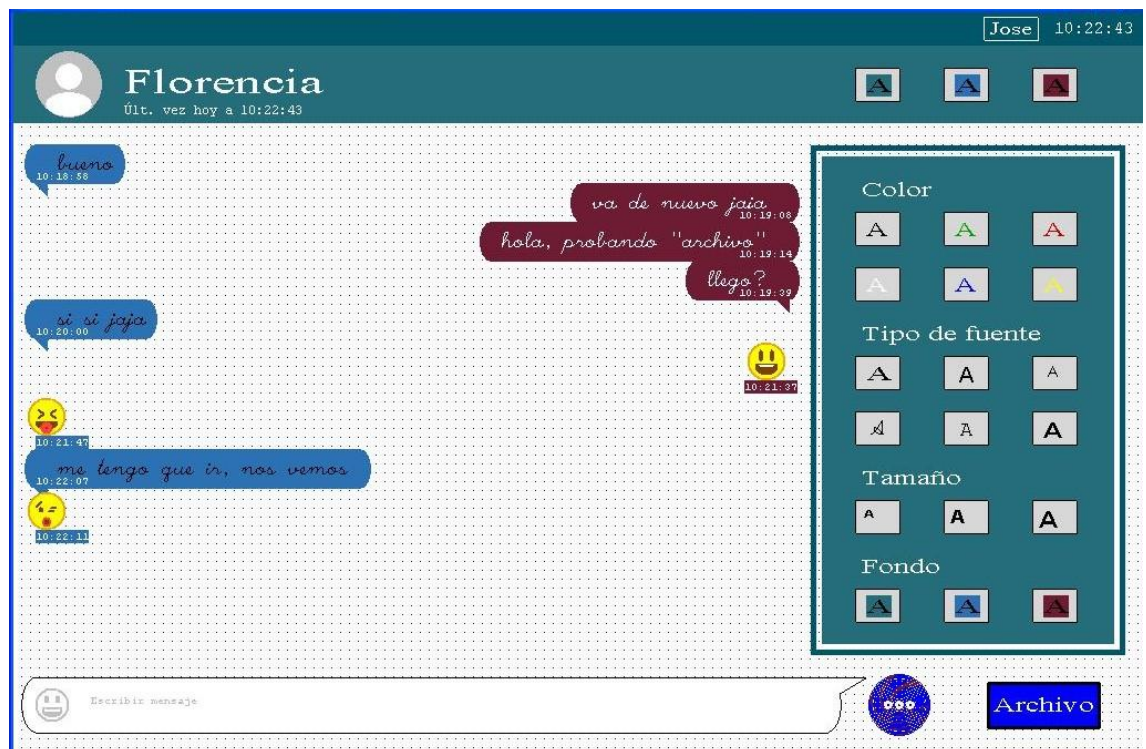
Si observo por debajo del cuadro de opciones se encuentra un botón azul donde puedo crear un archivo txt (ARCHIVO.txt). Y lo que escriba en él puedo enviárselo al otro usuario, presionándolo. Pero si lo escrito dentro del archivo, excede los límites de la ventana gráfica, me aparece el siguiente mensaje “NO SOPORTADO”, en color rojo.







Si el usuario del otro equipo decide finalizar con la comunicación, al apretar la tecla Escape (Esc), se puede observar que aparece un mensaje bajo su nombre, informándome su ultimo estado en línea.



Conclusión

Una vez finalizado el trabajo llego a la conclusión que es posible la comunicación entre dos ordenadores por medio de un cable de puertos en serie, y a su vez tener una visualización clara de la información a tratar.

Puedo agregar que fue una tarea complicada al no tener conocimientos previos de las herramientas a utilizar pero, de igual modo una vez concluido todo esto, veo que todo lo aprendido a medida que fui llevándolo a cabo me servirá para desenvolverme en futuros proyectos.

Bibliografía

<http://es.edaboard.com/topic-1295829.0.html>

<http://www.codeproject.com/Articles/992/Serial-library-for-C>

<http://www.elrincondelc.com/nuevorincon/foros/viewtopic.php?t=3772&sid=f9c477d874cbdc8906c36082837758d7>

<http://www.todopic.com.ar/foros/index.php?topic=25264.0>

http://www.mileti.com.ar/apuntes/modo_grafico_c.pdf