
CONTROL DE ACCESO

INFORME FINAL

Dany Medina

20192730

Mauricio Salazar

20200392

Sistemas Digitales – IEE256

Horario: 052F

Noviembre 2021

Índice

1. Resumen del proyecto	1
2. Requerimientos, especificaciones y restricciones	1
3. Metodología empleada	2
4. Descripción del Proyecto	3
4.1. Hardware	3
4.2. Software	4
5. Resultados	7
6. Conclusiones	7
7. Recomendaciones	7
8. Referencias	8
9. ANEXO: Código del Programa	8

1. Resumen del proyecto

El sistema consiste en un mecanismo de verificación de identidad para usuarios permitidos dentro de un área limitada por este dispositivo. Tendrá tres funciones principales: registrar nuevas credenciales a usuarios, borrar credenciales de usuarios ya registrados, y emplear las claves registradas para permitir el acceso. Al confirmar una contraseña correcta se indicará mediante la pantalla Nokia 5110 y un led que está permitido el ingreso. Asimismo, se incluirá una contraseña de administrador solicitada al almacenar nuevas contraseñas en el sistema y al borrar un usuario ya registrado con su respectiva clave.

Los objetivos del proyecto desarrollado fueron:

- Crear un almacén editable de usuarios asociados a sus respectivas claves.
- Agregar requisitos de longitud, caracteres, y no repetición sobre credenciales a registrar.
- Transmitir información del celular al microcontrolador inalámbricamente.
- Diseñar funciones modulares que simplifiquen el proceso.
- Crear menús de selección interactivos para el usuario en la pantalla Nokia 5110.
- Implementar físicamente el sistema diseñado.

2. Requerimientos, especificaciones y restricciones

El sistema esta compuesta por una pantalla LCD NOKIA 5110 para transmisión de mensajes o información; leds rojo y verde, para indicar si hay error o la contraseña es correcta; UART2, para escribir o leer información a través de un módulo Bluetooth. Con estos periféricos, el sistema deberá proceder de la siguiente forma:

1. Se transmite un mensaje preguntandole al usuario si desea mantener o cambiar la clave de administrador ADMIN.
2. Si se ingresa la opción (1)Cambiar, la pantalla solicitará una nueva clave, permitiendole al usuario modificarla. Sin embargo, si se ingresó la opcion (2)Mantener, la clave predeterminada ADMIN se mantendrá.
3. A continuación se transmitirán las siguientes opciones: "(1)Ingresar, (2)Nuevo usuario, (3)Borrar usuario"Se saltará al paso 4), 5) o 7) dependiendo de cuál se ingrese.
4. Si se ingresa la opción (1)Ingresar, se le pedirá al usuario ingresar su clave. En caso esta clave sea incorrecta el led rojo se encenderá; caso contrario, se enciende el led verde y en la pantalla aparecerá un mensaje dandole la bienvenida al usuario.
5. Si se ingresa la opción (2)Nuevo, se le pedirá al usuario la contraseña de administrador. En caso sea incorrecta se enciende led rojo y se retornaría al menú de inicio. Caso sea correcta, se procederá al paso 6).
6. Si hay 10 o menos usuarios registrados, se enciende led verde y se pedirá al usuario que ingrese su nombre y su nueva clave. En caso el nombre o la clave sean repetidas se encenderá el led rojo y se le pedirá nuevamente su nombre o clave. Caso contrario, el led verde se enciende y se le informará al usuario que se registró de manera exitosa. En el caso que ya existan 10 usuarios, únicamente se notificará que la memoria está llena.
7. Si se ingresa la opcion (3)Borrar, se le pide al usuario que ingrese la contraseña de administrador. Si es correcto se le solicitará el nombre de un usuario registrado. En caso ingrese un usuario invalido, aparecerá una notificación informando lo sucedido y se volverá a pedir el nombre de un usuario a borrar. En caso contrario,se enciende el led verde y aparecerá una notificación de que se ha borrado los credenciales del usuario elegido.

Las restricciones del desarrollo de este sistema fueron las siguientes:

- Restricciones del curso como el uso del lenguaje C como único lenguaje de programación o el empleo del microcontrolador TM4C123GH6PM.

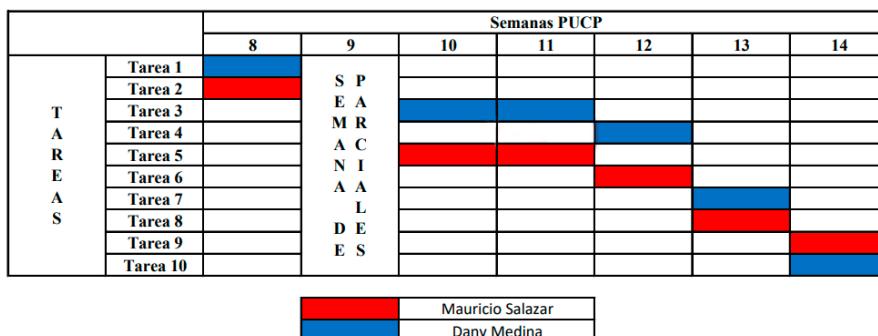
- Restricciones del proyecto como el uso de una pantalla nokia o la necesidad de una contraseña de fábrica.
- Restricciones debidas al estado de pandemia, ya que, en caso se quiera implementar, los componentes para armar el sistema son muy limitados. Debido a esto, se tuvo que reemplazar el Keypad del diseño original por un módulo Bluetooth.

3. Metodología empleada

Para la división del trabajo inicialmente se planteó una organización basada en una lista de tareas con el diseño básico original en mente. Es decir, un control de acceso con solo las funciones de ingreso y registro de usuarios. Sin embargo, a lo largo del desarrollo del proyecto se realizó un cambio del uso del Keypad por un módulo Bluetooth para el ingreso de datos. Asimismo, se agregaron funciones adicionales tales como la eliminación de credenciales y la configuración de una clave de administrador personalizada. Por ello, finalmente la distribución del trabajo se realizó como presentado a continuación.

1. Investigación sobre funcionamiento de pantalla LCD
2. Programación de comunicación serial por UART2
3. Escritura de mensajes en la pantalla LCD
4. Asociación de contraseñas con usuarios
5. Lectura de claves y usuarios por terminal serial
6. Implementación de clave maestra configurable
7. Verificación de cumplimiento de requisitos para credenciales
8. Validación de credenciales correctas para ingreso y notificación con LEDs
9. Borrado de credenciales de usuarios seleccionados
10. Verificación de claves y/o usuarios no repetidos en la memoria

Diagrama de Gantt



Por otro lado, la división de la redacción del presente informe se realizó de la siguiente forma:

Distribución para informe final

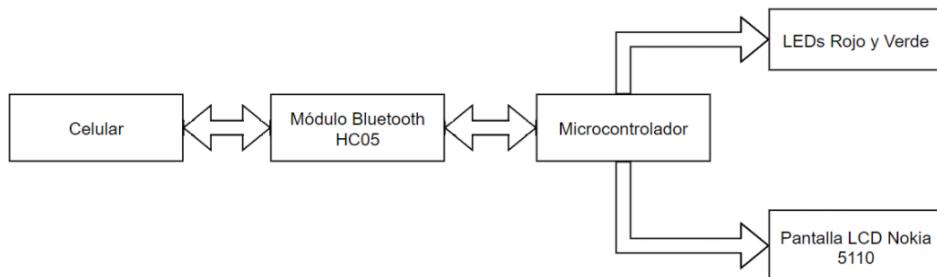
Mauricio Salazar	Dany Medina
Metodología	Resumen
Descripción de Hardware	Requerimientos
Resultados	Descripción de Software
Recomendaciones	Conclusiones
Anexo	Bibliografía

4. Descripción del Proyecto

4.1. Hardware

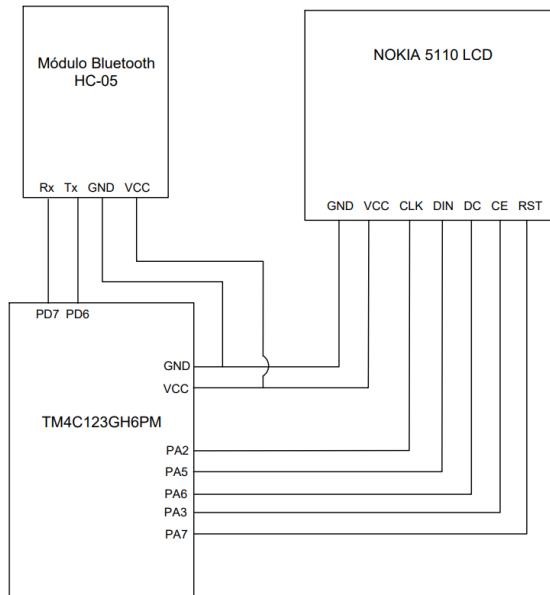
Para el diseño del Hardware del sistema, se comenzó planteando el diagrama de bloques presentado debajo. De esta manera, se inició estableciendo las conexiones que habrían entre componentes. El microcontrolador controlará a los LEDs y la pantalla LCD, mientras que habrán conexiones de ida y vuelta entre el microcontrolador y el módulo Bluetooth al transmitir y recibir mensajes. Finalmente, existe una conexión inalámbrica entre el módulo Bluetooth y el celular para conectarse para controlar el sistema y recibir indicaciones.

Diagrama de Bloques



Con el tipo de conexiones ya planteadas, se pudo diseñar un diagrama esquemático para ordenar los pines a conectarse entre la tarjeta Tiva del microcontrolador. Las conexiones con LEDs no fueron incluidas dado que estas se realizan en la tarjeta misma al usar el puerto F.

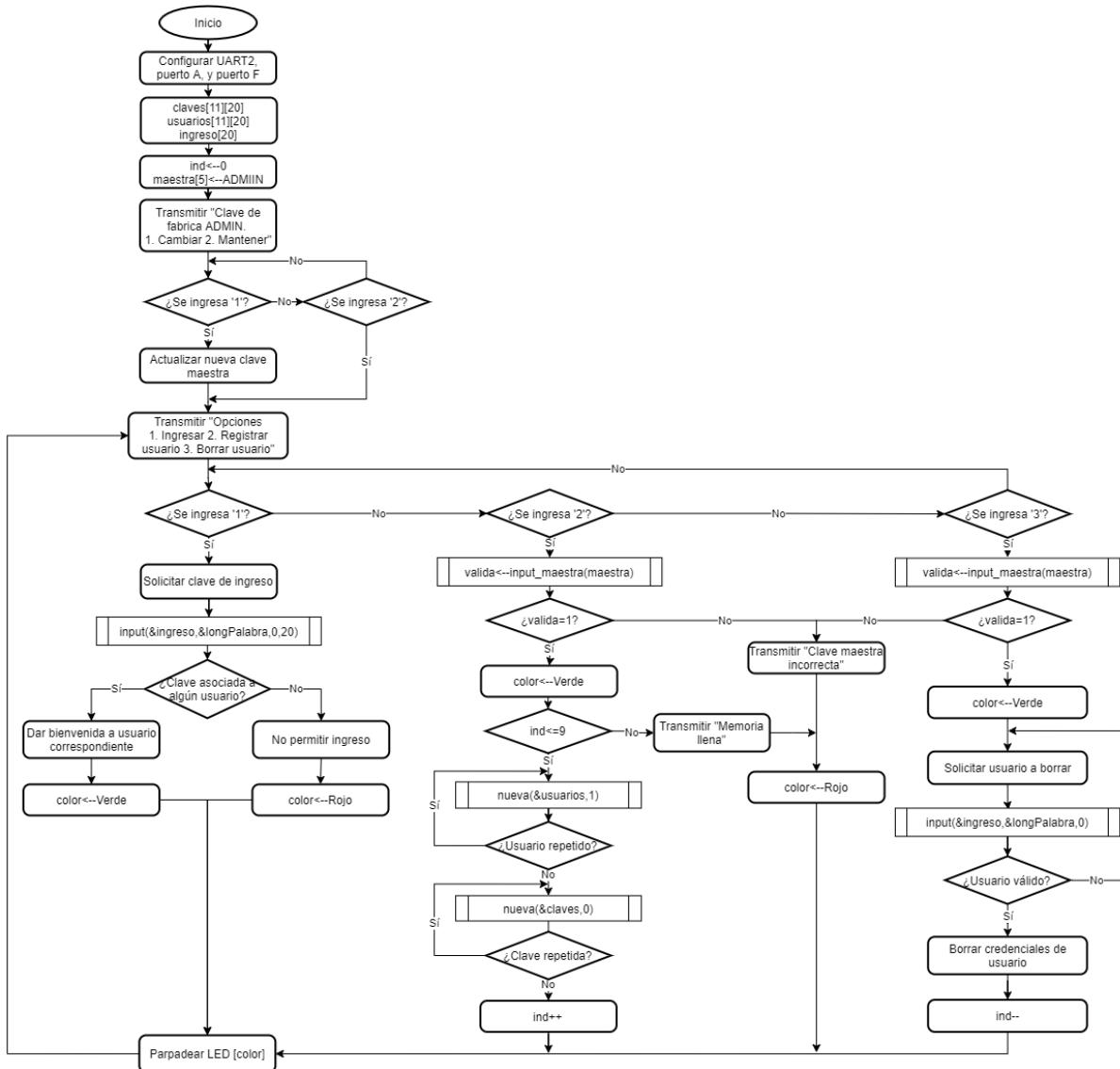
Diagrama Esquemático



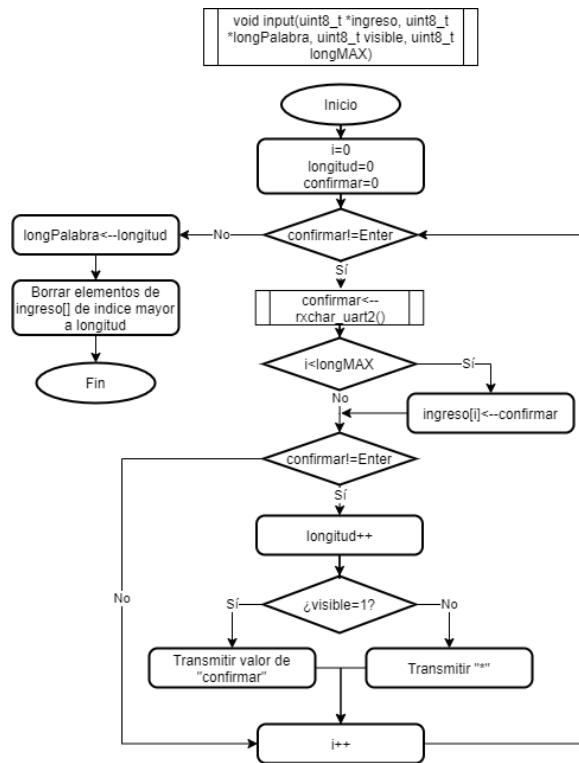
Se conectó así el módulo Bluetooth con el UART 2 del microcontrolador mediante los pines PD6 y PD7 para recibir y mandar información. Mientras tanto, dado que la conexión con la pantalla LCD se realiza mediante protocolo SPI, se requirió también de una señal de reloj conectada a CLK para poder controlarla.

4.2. Software

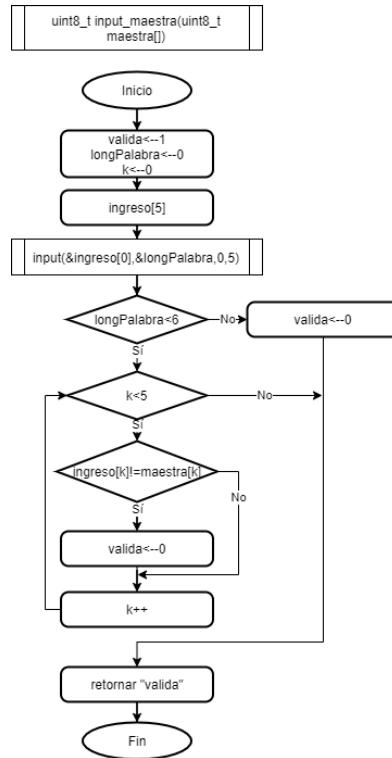
En el programa principal se usaron arreglos para almacenar las credenciales (nombre/claves) usuarios; asimismo, para la recepción de caracteres ingresados y transmisión de mensajes. Tambien, se usó la pantalla LCD mediante su biblioteca para transmitir indicaciones y los mensajes escritos por el usuario. Adicionalmente se emplearon bucles, en caso de la opción 2, para impedir caracteres inválidos y contraseñas o usuarios repetidos; en caso la opción 3, si se escribió correctamente un usuario a borrar. Finalmente, para la configuración del UART 2 se establecieron los parametros predeterminados del módulo HC05 indicados en su hoja de datos (9600 bps, 8 bits, sin paridad, 1 bit de parada).



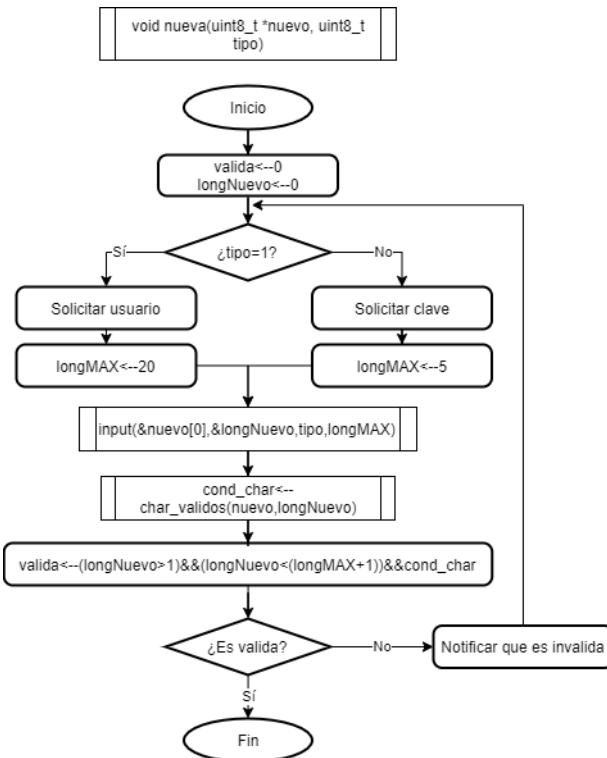
En la función input() se modificará las entradas ingreso[], que representará lo que se está leyendo en el UART2; longPalabra, que representará la longitud de la palabra escrita; visible, que define si se muestra en UART2 el mensaje original o asteriscos;longMAX, que limitará la longitud de los caracteres ingresados. En caso el usuario haya terminado de escribir, al apretar Enter la función finalizará. Tambien finaliza si la longitud de la palabra ha superado la Longitud máxima.



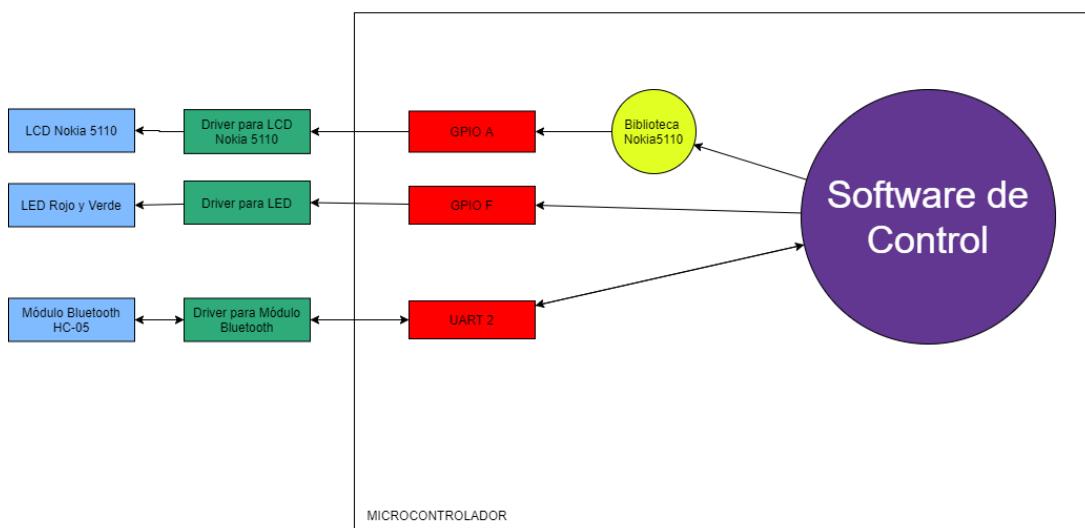
La función `inputMaestra` compara todos los elementos de un arreglo con la contraseña de administrador. En caso dicho arreglo tiene la misma longitud y mismos caracteres que la contraseña de administrador, la validez es 1. En caso contrario, la validez es 0.



La función nueva(), cuya entrada decide qué se solicitará(Un usuario de longitud máxima 20 o una clave de longitud máxima 5), requiere de la función input() y una función que valide los caracteres ingresados(En este caso solo serán validos las letras mayúsculas o minúsculas y números).Se usará un bucle para evitar los caracteres inválidos.



Se construyó así el diagrama de flujo de datos para el sistema, empleando 3 puertos para los 3 periféricos y una sola biblioteca. El microcontrolador controla a la pantalla Nokia LCD mediante el puerto A empleando la librería Nokia5110 incluida en el ejemplo de demostración de la guía del curso sobre el uso del módulo LCD. Para los LEDs emplea el puerto F, usando los ya incluidos en la tarjeta. Y finalmente para el módulo Bluetooth emplea el UART 2 transfiriendo y recibiendo información con la configuración realizada.



5. Resultados

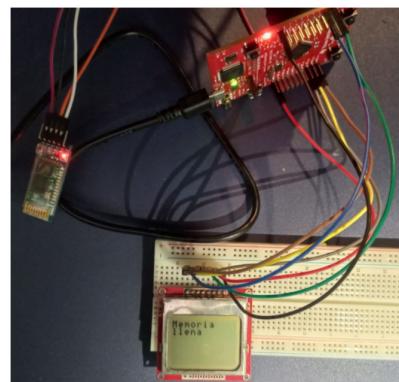
Implementando el código realizado junto con el diseño entre componentes planteado, se consiguió un sistema de control de acceso funcional como se esperaba. Al verificarlo, se encontró que todas sus funciones operaban sin problemas, tanto la configuración de clave de administrador como todas las opciones del menú de inicio (ingreso, registro de usuarios y borrado de usuarios). Las pruebas de su funcionamiento exitoso se muestran a continuación.



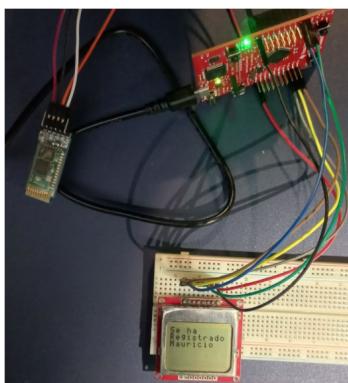
Menú Admin



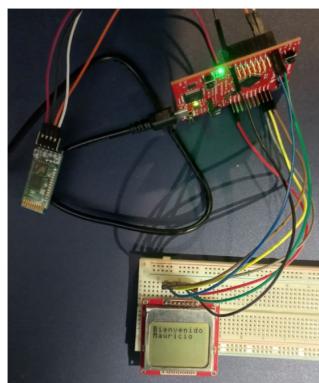
Menú Inicio



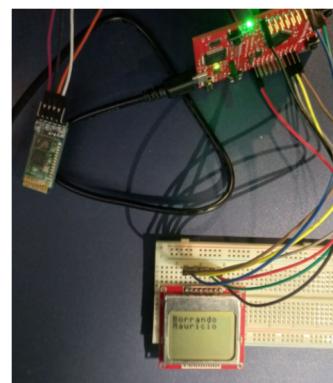
Memoria Llena



Registro usuario



Ingreso Exitoso



Borrar usuario

Adicionalmente, se grabó un video demostrando la operación correcta de todas sus funciones empleando el terminal serial Bluetooth del celular.

Link: <https://www.youtube.com/watch?v=81Ui3mBGVC8>

6. Conclusiones

En principio se cumplieron todos los objetivos base del proyecto. Por ello, se adicionaron más funciones, tales como agregar la opción de modificar al principio del programa la contraseña de administrador y verificar que credenciales nuevas no coincidan con otras ya existentes. También se logró implementar físicamente el sistema propuesto.

7. Recomendaciones

Si bien el sistema llegó a operar correctamente, existe una serie de ajustes y adiciones que se podrían implementar para mejorarlo. Por este motivo, se incluye la siguiente lista de recomendaciones a considerar de querer ampliar la funcionalidad del sistema de control de acceso.

- Adicionar un historial de ingreso con reloj que almacene la hora de ingreso de los usuarios en un registro.
- Utilizar LEDs externos a la tarjeta Tiva en lugar de los que ya vienen montados.
- Implementar un modo de ingreso de datos alternativo mediante un Keypad.
- Integrar la función de ingreso mediante RFID.

8. Referencias

- Firdaus, M. (2017). *DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC05*. <http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>
- Gonzales, L. (2021) *MANEJOS DE MÓDULOS ADC, DAC, LCD y TFT*. Pontificia Universidad Católica del Perú.
- Trujillano, F., Fernández, J., Cucho, Z., Orihuela, F., Morales, M. & Raymi, L. (2021) *LABORATORIO 7 PROGRAMACIÓN DEL UART*. Pontificia Universidad Católica del Perú.

9. ANEXO: Código del Programa

```

1 /* **** */
2 /* **** PONTIFICIA UNIVERSIDAD CATOLICA DEL PER */
3 /* **** FACULTAD DE CIENCIAS E INGENIERIA */
4 /* **** SISTEMAS DIGITALES */
5 /* **** */
6 /* **** */
7 /* * Microcontrolador: TM4C123GH6PM */
8 /* * EvalBoard: Tiva C Series TM4C123G LaunchPad */
9 /* * Autores: Mauricio Salazar y Dany Medina */
10 /* * Fecha: Noviembre 2021 */
11 /* **** */
12 /* * Enunciado: */
13 /* * Control de acceso con memoria de 10 usuarios. Permite registrar */
14 /* * nuevos con una clave admin y cambiar dicha clave de administrador */
15 /* **** */
16
17 #include <stdint.h>
18 #include "tm4c123gh6pm.h"
19 #include "Nokia5110.h"
20
21 /* Funcion de configuracion del puerto F*/
22 void config_portF(void){
23     SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_R5;           // Activar el Reloj de
    Puerto F
24     while( (SYSCTL_PGPIO_R & SYSCTL_PGPIO_R5)==0) { } // Se espera a que se
    active el reloj
25
26     GPIO_PORTF_LOCK_R = 0x4C4F434B;                      // Activar configuracion
    // de PF0
27     GPIO_PORTF_CR_R |= 0x01;
28
29     GPIO_PORTF_DIR_R |= 0x0E;                            // Configura pines PF1-3
    como salidas
30     GPIO_PORTF_DIR_R &= ~(0x11);                      // Configura pines PF0+
    PF4 como entradas.
31     GPIO_PORTF_PUR_R = 0x11;                            // Activar pull-up para
    PF0 y PF4
32     GPIO_PORTF_AFSEL_R &= ~0x11;                      // Desactivar
    funciones alternas
33     GPIO_PORTF_DEN_R |= 0x1F;                          // Activar pines PF0-PF4
    como digital
34

```

```

35     GPIO_PORTF_DATA_R &= ~(0x0E);                                // Apaga todos los LEDs
36 }
37
38 // ***** Parpadeo *****
39 // Parpadeo de led de color seleccionado
40 // ENTRADAS: color
41 // SALIDA: ninguna
42 void Parpadeo(uint32_t color){
43     uint32_t i;
44
45     GPIO_PORTF_DATA_R |= color;           // Prender Led [color]
46     for(i=0;i<=3200000;i++);
47     GPIO_PORTF_DATA_R &= ~(color);       // Apagar Led [color]
48     for(i=0;i<=3200000;i++);
49 }
50
51 /* Funcion de configuracion del UART2*/
52 void config_uart2(void){
53     SYSCTL_RCGCUART_R |= SYSCTL_RCGCUART_R2;                // Habilitamos UART2
54     while((SYSCTL_RCGCUART_R & SYSCTL_RCGCUART_R2)==0){};    // Esperamos a que se active UART2
55     SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_R3;                // Activamos el puerto D
56     while((SYSCTL_PGPIO_R & SYSCTL_RCGCGPIO_R3)==0){};    // Esperamos a que se active el reloj de D
57
58     // Configuracion para 9600 bps, 1 parada, sin paridad (Default de HC-05)
59     UART2_CTL_R = 0x0;                                         // Se desactiva el UART
60     UART2_IBRD_R = (UART2_IBRD_R & (~0xFFFF)) + 104;        // Parte entera
61     UART2_FBRD_R = (UART2_FBRD_R & (~0x3F)) + 11;            // Parte decimal
62     UART2_LCRH_R = (UART2_LCRH_R & (~0xFF)) | 0x70;          // 8 bits de datos, sin paridad, 1 bit de parada, FIFOs habilitados
63     UART2_CTL_R &= ~(0x20);                                    // HSE=0,
64     UART2_CTL_R |= 0x301;                                     // Para habilitar UARTEN=1, RXE=1, TX=1
65
66     // Configuracion portD
67     GPIO_PORTD_LOCK_R = 0x4C4F434B;                          // Activar configuracion
68     GPIO_PORTD_CR_R |= 0xF0;
69
70     GPIO_PORTD_DIR_R |= 0x80;                                 // PD7 como salida
71     GPIO_PORTD_DIR_R &= ~(0x40);                            // PD6 como entrada
72     GPIO_PORTD_DEN_R |= 0xC0;                                // Funciones digitales en PD6 y PD7
73     GPIO_PORTD_AMSEL_R &= ~(0xC0);                           // Desactivamos modo analogico en PD6 y PD7
74     GPIO_PORTD_AFSEL_R |= 0xC0;                              // Activamos funciones alternas en PD6 y PD7
75     GPIO_PORTD_PCTL_R = (GPIO_PORTD_PCTL_R&0x00FFFFFF)|0x11000000; // Conecta
76     // UART2 a PD6 y PD7
77 }
78
79 /* Funcion para la transmision de un caracter*/
80 void txcar_uart2(uint32_t car){
81     while ((UART2_FR_R & UART_FR_TXFF)!=0); // Espera que este disponible para tx
82     UART2_DR_R = car;
83 }
84
85 /* Funcion para la recepcion de un caracter de 8 bits*/
86 uint8_t rxcar_uart2(void){
87     uint8_t temp;
88     while ((UART2_FR_R & UART_FR_RXFE)!=0){}; // Se espera que llegue un dato
89     temp= UART2_DR_R&0xFF; // Se toman solo 8 bits
90
91     return temp;
92 }

```

```

92
93 /* Funcion para el envio de una cadena*/
94 void txmens_uart2( uint8_t mens[] ){
95     uint8_t letra;
96     uint8_t i = 0;
97     letra = mens[ i++ ];
98     while( letra != '\0' ){
99         //Se envian todos los caracteres hasta el fin de cadena
100        txcar_uart2( letra );
101        letra = mens[ i++ ];
102    }
103 }
104
105 // *****char_validos*****
106 // Verificacion de que el input sea solo letras o numeros
107 // ENTRADA: ingreso[], longPalabra
108 // SALIDA: Condicion (0 o 1)
109 uint8_t char_validos(uint8_t ingreso[], uint8_t longPalabra){
110     uint8_t invalidos ,condicion ,char_invalido ,i;
111     invalidos=0;
112
113     for( i=0; i < longPalabra; i++ ){                                // Evalua cada caracter del
114         char_invalido=1; // Caracter invalido
115         if(( '0'<=ingreso[i])&&('z'>=ingreso[i])){
116             char_invalido=0; // Caracter valido si esta entre '0' y 'z' (ASCII)
117             if((( '9'<ingreso[i])&&('A'>ingreso[i]))||(( 'Z'<ingreso[i])&&('a'>ingreso[i]))){
118                 char_invalido=1; // Caracter invalido si es algun simbolo no letra/numero
119                 entre '0' y 'z' (ASCII)
120             }
121             invalidos=invalidos+char_invalido; //Se suma la cantidad de caracteres
122             invalidos
123         }
124
125         // Definiendo si se cumple la condicion
126         if(invalidos){      // Si hay 1+ caracteres invalidos , no se cumple
127             condicion=0;
128         } else{           // Si hay 0 caracteres invalidos , se cumple
129             condicion=1;
130         }
131     }
132
133 // *****input*****
134 // Ingreso de cadena de caracteres hasta presionar Enter contando su longitud .
135 // ENTRADAS: *ingreso (direccion de palabra a ingresar) , *longPalabra (registro
136 // de longitud de palabra ingresada)
137 // ENTRADAS: visible (Mostrar caracteres o *) , longMAX (Longitud maxima del
138 // arreglo a ingresar)
139 // SALIDA: ninguna
140 void input(uint8_t *ingreso , uint8_t *longPalabra , uint8_t visible , uint8_t
141 longMAX){
142     uint8_t confirmar ,i;
143     uint32_t retardo ;
144     uint8_t oculta[] = "*";
145     uint8_t longitud=0;
146
147     for( i=0; (confirmar!=0x0D); i++ ){
148         confirmar=rxcar_uart2();
149         if(i<longMAX){ ingreso[i] = confirmar;}
150         if(confirmar!=0x0D){ //Si no se presiona ENTER
151             longitud++; //Aumentando longitud

```

```

149     if(visible){
150         txcar_uart2(confirmar);
151         Nokia5110_OutChar(confirmar);
152     }
153     else{
154         txmens_uart2(oculta);
155         Nokia5110_OutString("*");
156     }
157 }
158 *longPalabra=longitud;
159
160 // Borrando el resto del arreglo donde se almacena
161 for( i=longitud; i<longMAX; i++ ){
162     ingreso[i]=0;
163 }
164 // Retardo para mostrar en LCD
165 for( retardo=0; retardo<=3200000; retardo++ );
166 }
167
168 // *****nueva*****
169 // Registro de nuevo usuario o clave
170 // ENTRADAS: *nuevo (direccion de nuevo usuario/clave a registrar)
171 // ENTRADAS: tipo (1: Usuario, 0: Clave)
172 // SALIDA: ninguna
173 void nueva(uint8_t *nuevo, uint8_t tipo){
174     uint8_t longNuevo, valida, longMAX;
175     uint8_t mUsuario[] = "\n\rUsuario: ";
176     uint8_t mClave[] = "\n\rNueva clave: ";
177     uint8_t mInvalida[] = "\n\rLongitud o caracteres invalidos";
178
179     do{
180         Nokia5110_Clear();
181         if(tipo){
182             // Pantalla LCD
183             Nokia5110_OutString("Usuario:");
184             //UART
185             txmens_uart2(mUsuario);
186             longMAX=20;
187         } else{
188             // Pantalla LCD
189             Nokia5110_OutString("Clave:");
190             //UART
191             txmens_uart2(mClave);
192             longMAX=5;
193         }
194         Nokia5110_SetCursor(0,1);
195
196         // Ingresando
197         input(&nuevo[0],&longNuevo, tipo, longMAX);
198         // Validando
199         valida=(longNuevo>1)&&(longNuevo<(longMAX+1))&&char_validos(nuevo, longNuevo);
200
201         if(!valida){ //Caso invalido
202             txmens_uart2(mInvalida);
203             Nokia5110_Clear();
204             if(tipo){
205                 Nokia5110_OutString("Usuario");
206                 Nokia5110_SetCursor(0,1);
207                 Nokia5110_OutString("Invalido");
208             } else{
209                 Nokia5110_OutString("Clave");
210                 Nokia5110_SetCursor(0,1);
211                 Nokia5110_OutString("Invalida");
}

```

```

212     }
213     Parpadeo(0x02);
214 } while(! valida);
215 }
216
217 // ***** input_maestra*****
218 // Ingreso y validacion de clave maestra
219 // ENTRADAS: maestra[] (Arreglo de clave maestra registrada)
220 // SALIDA: valida (1 o 0)
221 uint8_t input_maestra(uint8_t maestra[]){
222     uint8_t mMaestra[] = "Ingrese clave maestra: \n\r";
223     uint8_t valida ,longPalabra ,k;
224     uint8_t ingreso[5];
225
226     //UART
227     txmens_uart2(mMaestra);
228     //LCD
229     Nokia5110_Clear();
230     Nokia5110_OutString("Ingrese");
231     Nokia5110_SetCursor(0,1);
232     Nokia5110_OutString("clave");
233     Nokia5110_SetCursor(0,2);
234     Nokia5110_OutString("maestra:");
235     Nokia5110_SetCursor(0,3);
236     //Ingresar clave maestra
237     input(&ingreso[0],&longPalabra ,0,5);
238
239     //Resetear verificacion
240     valida=1;
241     //Verificar clave maestra
242     if(longPalabra<6){ //Si tiene misma longitud de clave maestra
243         for( k=0; k < 5; k++ ){
244             if(ingreso[k]!=maestra[k]){
245                 valida=0;
246             }
247         } else{ valida=0; }
248         return valida;
249     }
250
251
252 /* ****PROGRAMA PRINCIPAL****/
253 /* ****PROGRAMA PRINCIPAL****/
254 /* ****PROGRAMA PRINCIPAL****/
255 int main( void ){
256 /* ****ETAPA0: Inicializacion puertos y variables*/
257 /* ****ETAPA0: Inicializacion puertos y variables*/
258 /* ****ETAPA0: Inicializacion puertos y variables*/
259
260     //Mensajes a utilizar
261     uint8_t mAdmin[] = "Clave de administrador ADMIN:\n\r 1.Cambiar\n\r 2.Mantener\n\r";
262     uint8_t mNuevoAdmin[] = "\n\rNueva clave de administrador: ";
263     uint8_t mInicio[] = "\n\rMENU INICIO\n\r 1.Ingresar\n\r 2.Registrar usuario\n\r 3.Borrar usuario\n\r";
264     uint8_t mClave[] = "Ingrese clave de ingreso: ";
265     uint8_t mFin1[] = "\n\rSea bienvenido usuario ";
266     uint8_t mFin2[] = "\n\rClave incorrecta\n\r";
267     uint8_t mExistente[] = "\n\rCredencial ya existente\n\r";
268     uint8_t mLleno[] = "\n\rMemoria llena , borre clave para registrar nuevos\nusuarios\n\r";
269     uint8_t mUsuarioS[]=" \n\rIngrese Usuario: ";
270     uint8_t mUsuarioF[]=" \n\rUsuario no valido\n\r";
271     uint8_t mUsuarioR[]=" \n\rSe ha registrado al usuario ";
272     uint8_t mUsuarioB[]=" \n\rBorrando credenciales de usuario ";

```

```

273 // Verificacion + Claves
274 uint8_t verificar[11];
275 uint8_t claves[11][5], usuarios[11][20]; // usuario y clave 11
276
277 // Clave maestra
278 uint8_t maestra[5] = "ADMIN";
279 // Arreglo para guardar intento digitado
280 uint8_t ingreso[20];
281
282 // Variables a utilizar
283 uint8_t usuario, i, k, ind, longPalabra, longUsuario, color;
284 uint8_t confirmar, opcion, valida;
285 // uint32_t retardo;
286
287 // Inicializacion portF + LCD +UART
288 Nokia5110_Init();
289 Nokia5110_Clear();
290 config_portF();
291 config_uart2();
292
293 /* ****ETAPA1: Cambiar clave de administrador****/
294 /* ****ETAPA1: Cambiar clave de administrador****/
295 /* ****ETAPA1: Cambiar clave de administrador****/
296 /* ****ETAPA1: Cambiar clave de administrador****/
297 // Definir clave de administrador
298 //UART
299 txmens_uart2(mAdmin);
300 //LCD
301 Nokia5110_OutString("Clave de");
302 Nokia5110_SetCursor(0,1);
303 Nokia5110_OutString("admin:");
304 Nokia5110_SetCursor(0,2);
305 Nokia5110_OutString("ADMIN");
306 Nokia5110_SetCursor(0,3);
307 Nokia5110_OutString(" 1. Cambiar");
308 Nokia5110_SetCursor(0,4);
309 Nokia5110_OutString(" 2. Mantener");
310
311 //MENU SELECCION ADMIN
312 confirmar=0;
313 while(confirmar!=0x0D){
314     opcion=0;
315     while((opcion != '1') && (opcion != '2')){
316         opcion=rxcar_uart2();
317     }
318     confirmar=rxcar_uart2();
319 }
320
321 if(opcion=='1'){ // 1. Cambiar clave de admin
322     nueva(&maestra[0],0);
323     //UART
324     txmens_uart2(mNuevoAdmin);
325     txmens_uart2(maestra);
326     //LCD
327     Nokia5110_Clear();
328     Nokia5110_OutString("Nueva clave");
329     Nokia5110_SetCursor(0,1);
330     Nokia5110_OutString("de admin:");
331     Nokia5110_SetCursor(0,2);
332     Nokia5110_OutString((char*)maestra);
333     // Confirmar cambio con parpadeo VERDE
334     Parpadeo(0x08);
335 }
336

```

```

337 // Inicializando claves
338 for( k=0; k < 11; k++ ){
339     for( i=0; i < 5; i++ ){
340         claves[k][i]=maestra[i];
341     }
342 }
343 // Inicializando usuario administrador
344 usuarios[10][0]='A';
345 usuarios[10][1]='d';
346 usuarios[10][2]='m';
347 usuarios[10][3]='i';
348 usuarios[10][4]='n';
349
350 while(1){
351 /* ****ETAPA2: Seleccion Menu de inicio**** */
352 /* ****ETAPA2: Seleccion Menu de inicio**** */
353 /* ****ETAPA2: Seleccion Menu de inicio**** */
354     txmens_uart2(mInicio);
355     Nokia5110_Clear();
356     Nokia5110_OutString("Opciones:");
357     Nokia5110_SetCursor(0,1);
358     Nokia5110_OutString("1. Ingresar ");
359     Nokia5110_SetCursor(0,2);
360     Nokia5110_OutString("2. Nuevo");
361     Nokia5110_SetCursor(0,3);
362     Nokia5110_OutString("usuario");
363     Nokia5110_SetCursor(0,4);
364     Nokia5110_OutString("3. Borrar");
365     Nokia5110_SetCursor(0,5);
366     Nokia5110_OutString("usuario");
367
368 //MENU SELECCION INICIO
369 confirmar=0;
370 while(confirmar!=0x0D){
371     opcion=confirmar;
372     while((opcion!='1')&&(opcion!='2')&&(opcion!='3')){
373         opcion=rxcar_uart2();
374     }
375     confirmar=rxcar_uart2();
376 }
377 /* ****ETAPA3: Ingreso de clave**** */
378 /* ****ETAPA3: Ingreso de clave**** */
379 /* ****ETAPA3: Ingreso de clave**** */
380 if(opcion=='1'){ //Opcion 1: Ingresar clave
381     //Solicitando clave
382     Nokia5110_Clear();
383     Nokia5110_OutString("Ingrese su ");
384     Nokia5110_SetCursor(0,1);
385     Nokia5110_OutString("Clave");
386     Nokia5110_SetCursor(0,2);
387     txmens_uart2(mClave);
388     //Ingresando
389     input(&ingreso[0],&longPalabra ,0,5);
390
391     //Reset de variables de verificacion
392     usuario=0;
393     for( k=0; k < 11; k++ ){
394         verificar[k]=1;
395
396         //Verificacion de clave
397         if(longPalabra>5){ usuario=0;}
398         else{
399             for( k=0; k < 11; k++ ){
400                 for( i=0; i < 5; i++ ){

```

```

401         if(ingreso[i]!=claves[k][i]){
402             verificar[k]=0;}}
403         if(verificar[k]==1){
404             usuario=k+1;} // Asignando clave a usuario+1 cuya clave corresponda
405         }
406     }
407
408 /* ****ETAPA4: Dar/Negar Acceso*****/
409 /* ****ETAPA4: Dar/Negar Acceso*****/
410 /* ****ETAPA4: Dar/Negar Acceso*****/
411     if(usuario>=1){ //Caso clave valida para algun usuario
412         //UART
413         txmens_uart2(mFin1);
414         txmens_uart2(usuarios[usuario-1]);
415         //LCD
416         Nokia5110_Clear();
417         Nokia5110_OutString("Bienvenido");
418         Nokia5110_SetCursor(0,1);
419         Nokia5110_OutString((char*)usuarios[usuario-1]);
420         color=0x08; //Parpadeo final Verde
421     }else{ //Caso clave invalida
422         //UART
423         txmens_uart2(mFin2);
424         //LCD
425         Nokia5110_Clear();
426         Nokia5110_OutString("Clave");
427         Nokia5110_SetCursor(0,1);
428         Nokia5110_OutString("Incorrecta");
429         color=0x02; //Parpadeo final Rojo
430     }
431
432 /* ****ETAPA5: Registro nuevo usuario (Espacio disponible)****/
433 /* ****ETAPA5: Registro nuevo usuario (Espacio disponible)****/
434 /* ****ETAPA5: Registro nuevo usuario (Espacio disponible)****/
435     }
436     if(opcion=='2'){//Opcion 2: Registrar nuevo usuario
437         //Ingresar clave maestra
438         valida=input_maestra(maestra);
439
440         if(valida==1){ //Si clave maestra correcta
441             if(ind<=9){ //Menos de 10 usuarios guardados
442                 do{//USUARIO
443                     nueva(&usuarios[ind][0],1);
444
445                     //Resetear verificacion
446                     usuario=0;
447                     for( k=0; k < 11; k++ ){verificar[k]=1;}
448
449                     //Verificar si coincide con alguno ya existente
450                     for( k=0; k < 11; k++ ){
451                         if(k!=ind){
452                             for( i=0; i < 20; i++ ){ //Evaluar que coincidan sus
453                                 caracteres
454                                 if(usuarios[ind][i]!=usuarios[k][i]){
455                                     verificar[k]=0;}}
456                                 if(verificar[k]==1){usuario=k+1;}}
457                         }
458                     if(usuario!=0){
459                         txmens_uart2(mExistente);
460                         Nokia5110_Clear();
461                         Nokia5110_OutString("Usuario");
462                         Nokia5110_SetCursor(0,1);
463

```

```

464     Nokia5110_OutString("ya existe");
465     Parpadeo(0x02);
466     }
467 }while(usuario!=0); // Si coincide con algun usuario seguira en el
bucle
468
469 do{ //CLAVE
470     nueva(&claves[ind][0],0);
471
472     // Resetear verificacion
473     usuario=0;
474     for( k=0; k < 11; k++ ){
475         verificar[k]=1;
476     // Verificar si coincide con alguno ya existente
477     for( k=0; k < 11; k++ ){
478         if(k!=ind){
479             for( i=0; i < 5; i++ ){ // Evaluar que coincidan sus caracteres
480                 if(claves[ind][i]!=claves[k][i]){
481                     verificar[k]=0;
482                 }
483                 if(verificar[k]==1){usuario=k+1;}
484             }
485         }
486         if(usuario!=0){
487             txmens_uart2(mExistente);
488             Nokia5110_Clear();
489             Nokia5110_OutString("Clave");
490             Nokia5110_SetCursor(0,1);
491             Nokia5110_OutString("ya existe");
492             Parpadeo(0x02);
493         }
494     }while(usuario!=0); // Si coincide con algun usuario seguira en el bucle
495
496 //UART
497 txmens_uart2(mUsuarioR);
498 txmens_uart2(usuarios[ind]);
499 //LCD
500 Nokia5110_Clear();
501 Nokia5110_OutString("Se ha");
502 Nokia5110_SetCursor(0,1);
503 Nokia5110_OutString("Registrado");
504 Nokia5110_SetCursor(0,2);
505 Nokia5110_OutString((char*)usuarios[ind]);
506 color=0x08; // Parpadeo final Verde
507 ind++;
508 }
509
510 /******ETAPA6: Registro nuevo usuario (Memoria Llena)*****/
511 /******ETAPA6: Registro nuevo usuario (Memoria Llena)*****/
512 /******ETAPA6: Registro nuevo usuario (Memoria Llena)*****/
513 else{ // Notificar que arreglos estan llenos
514     //UART
515     txmens_uart2(mLleno);
516     //LCD
517     Nokia5110_Clear();
518     Nokia5110_OutString("Memoria");
519     Nokia5110_SetCursor(0,1);
520     Nokia5110_OutString("lена");
521     color=0x02; // Parpadeo final Rojo
522 }
523
524
525 /******ETAPA6: Registro nuevo usuario (Memoria Llena)*****/

```

```

526 /* *****ETAPA7: Clave maestra incorrecta*****/
527 /* ****ETAPA8: Borrar Usuario*****/
528 else{
529     txmens_uart2(mFin2); //Caso Clave maestra incorrecta
530     //LCD
531     Nokia5110_Clear();
532     Nokia5110_OutString("Clave");
533     Nokia5110_SetCursor(0,1);
534     Nokia5110_OutString("incorrecta");
535     color=0x02; //Parpadeo final Rojo
536 }
537 }
538
539 /* ****ETAPA8: Borrar Usuario*****/
540 /* ****ETAPA8: Borrar Usuario*****/
541 /* ****ETAPA8: Borrar Usuario*****/
542 if(opcion=='3'){//Opcion 3: Borrar usuario
543     valida=input_maestra(maestra);
544
545     if(valida==1){ //Si clave maestra correcta
546         do{
547             color=0x08;
548             txmens_uart2(mUsuarioS);
549             //LCD
550             Nokia5110_Clear();
551             Nokia5110_OutString("Usuario a");
552             Nokia5110_SetCursor(0,1);
553             Nokia5110_OutString("borrar:");
554             Nokia5110_SetCursor(0,2);
555             //Ingreso
556             input(&ingreso[0],&longUsuario ,1 ,20 );
557
558             //Resetear verificacion
559             usuario=0;
560             for( k=0; k < 10; k++ ){ verificar[k]=1;}
561
562             //Verificar
563             if((longUsuario<1)|| (longUsuario>20)){usuario=0;}
564             else{
565                 for( k=0; k < 10; k++ ){
566                     for( i=0; i < 20; i++ ){//Evaluar que coincidan sus caracteres
567                         if(ingreso[i]!=usuarios[k][i]){
568                             verificar[k]=0;
569                         }
570                         if(verificar[k]==1){usuario=k+1;}
571                     }
572                 }
573
574             if(usuario==0){ //Usuario invalido
575                 //LCD
576                 Nokia5110_Clear();
577                 Nokia5110_OutString("Usuario");
578                 Nokia5110_SetCursor(0,1);
579                 Nokia5110_OutString("invalido");
580                 //UART
581                 txmens_uart2(mUsuarioF);
582                 Parpadeo(0x02);}
583             }while(usuario==0);
584
585             //Remplazando credenciales de usuario seleccionado
586             txmens_uart2(mUsuarioB);
587             txmens_uart2(usuarios[usuario-1]);
588             //LCD
589             Nokia5110_Clear();

```

```

590 Nokia5110_OutString("Borrando");
591 Nokia5110_SetCursor(0,1);
592 Nokia5110_OutString((char*)usuarios[usuario-1]);
593
594 for(i=(usuario-1);i<9;i++){
595     for(k=0;k<20;k++){
596         if(k<5){claves[i][k]=claves[i+1][k];}
597         usuarios[i][k]=usuarios[i+1][k];
598     }
599 }
600 // Cambiando la ultima clave por la de admin otra vez
601 for( k=0; k < 20; k++){
602     if(k<5){claves[9][k]=maestra[k];}
603     usuarios[9][k]=0;
604 }
605 ind--; // Retrocediendo una posicion en los registros
606 } else {
607     txmens_uart2(mFin2); // Caso Clave maestra incorrecta
608     //LCD
609     Nokia5110_Clear();
610     Nokia5110_OutString("Clave");
611     Nokia5110_SetCursor(0,1);
612     Nokia5110_OutString("incorrecta");
613     color=0x02;
614 }
615 }
616 // Parpadeo final
617 Parpadeo(color);
618 }
619 }
620 /* *****/
621 /* *****FIN DE PROGRAMA***** */
622 /* *****/

```

Listing 1: Control de Acceso