PARCIAL 1:

Informe final del desarrollo del proyecto de
Informa2 S.A.S.

Juan Fernando Muñoz López
Carlos Daniel Lora Larios
Nicolás David Ruidiaz Durán

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Abril de 2021

Índice

1. Primeras impresiones del problema.

El arte de la programación radica su más pura esencia en usar la lógica para la construcción de soluciones eficientes ante problemáticas a las que nos enfrentamos en el día a día, es por eso que enfrentamos esta significativa actividad con la mejor actitud y más altas expectativas; para hacer que el amor por la programación y el desarrollo de hardware, crezca, consiguiendo en paralelo que nuestra vocación ingenieril esté más consolidada, para que sobre todo esta actividad nos deje un gran valor añadido en nuestro crecimiento como programadores, en liderazgo, compañerismo, paciencia, creatividad, y como no, en la lógica, para dar solución a problemas de cualquier ámbito teniendo como base ese importante sentido humano.

2. Desarrollo y análisis de la solución proupuesta:

Al analizar, pensar y discutir sobre el problema, hemos dado nuestras apreciaciones a la manera en la que vamos a atacar el problema. Puesto a que debemos llevar ligado el manejo del hardware a través de la plataforma Tinkercad y el desarrollo de software a través de la plataforma Qt, hemos optado en una primera instancia en desarrollar una propuesta de solución al problema planteado desde la plataforma Qt, mediante el lenguaje de programación C++. Esta propuesta consistía en hacer un menú iterativo en el que a partir de ingresar ciertos datos se brindaba un servicio diferente:

Puntos a considerar:

2.1. Consideraciones acerca del integrado 74HC595:

Antes de plantear una posible solución nos pusimos en la tarea de entender, e interactuar, y poner en funcionamiento proyectos en la plataforma Tinkercad en los que estuviera involucrado el integrado 74HC595; para poco a poco ir construyendo el circuito sobre el cual estaría estructurado nuestro programa.

Las conclusiones nos llevaron a determinar que la mejor opción era establecer una cadena de 8 integrados, a los que se le estará inyectando por medio de los pines digitales la información que nos permitirá hacer un desplazamiento de datos para lograr mostrar en una matriz de leds de tamaño 8x8 un patrón que determinará un usuario.

2.2. Acerca de la implementación en C++:

La estructura base de este fue hacer que un problema, fuera más sencillo de resolver dividiéndolo en unos más pequeños. Por lo que, a la hora de hacer la implementación de la solución en C++, decidimos hacer múltiples funciones que nos dieran una el servicio necesario para que las funciones que son requeridas "verificación, imagen y publik" fueran más a menas de desarrollar.

A partir de nuestras apreciaciones sobre el circuito que ya habíamos establecido en la plataforma Tinkercad, decidimos que la mejor forma en la que el usuario podría ingresar los datos y como el patrón esta determinado a partir de 8 filas con 8 elementos y sabiendo las limitaciones en la obtención de datos por medio del puerto serial, era pedir cada fila de la matriz, es decir, si de la primera fila solo el surgió desea prender los leds de las esquinas debe ingresar 1000001, por lo que un 1 es un led encendido o un 0 es un led apagado; al tomar los datos como enteros y sabiendo el proceso algorítmico para llevar esa información a la memoria si el usuario desea ingresar 00000011, este podría prescindir de ingresar los ceros y solo ingresar 11.

Esta misma lógica nos ayudó a llevar un símil a la plataforma Qt; para que emplear los siguientes aspectos:

-Crear un menú iterativo sobre el cuál usuario tiene la posibilidad de: verificar el funcionamiento de los leds, ingresar los datos para formar un patrón y mostrarlo en la matriz de leds, ingresar múltiples patrones y mostrarlos en ciertos intervalos en la matriz de leds.

-Para cada uno de estos aspectos creamos una función que pide la información y otra que la imprime. Lo que nos permitió llevar esta misma lógica a la plataforma Arduino, haciendo una especie de traducción principalmente de las funciones que imprimen los datos sobre la matriz de leds. Puesto que debíamos establecer el desplazamiento de los datos en un arreglo [8][8] a través de 8 integrados. A través del manejo de relojes y el puerto serial del Arduino. Y conseguir un asertivo funcionamiento del programa, además del uso de la memoria dinámica.

3. Esquema de desarrollo de actividades del algoritmo:

3.1. Tareas de electrónica de potencia:

En este caso lo que hicimos fue buscar ejemplos del mecanismo que usan estas tiendas comerciales para presentar las imagenes , asi que encontramos que podriamos usar una multiplexacion de leds que es un metodo que se emplea para trabajar con matrices de leds, en este caso de 8x8 conectando los catodos en

una misma fila y los anodos en una columna.

Nos vimos tambien en la necesidad de buscar los valores de las resistencias necesarios para que funcionen los leds.

Por último verificamos que funcionara prendiendo todos los leds de la matriz.

3.2. Tareas de control de potencia:

En este caso la tarea más difícil a la que nos tuvimos que enfrentar fue la comprensión del funcionamiento del chip 7HC595, nvestigamos y nos guiamos de esquemas para conocer susentradas y saber cómo conectarlas para obtener las salidas que queríamos con los valores que ingresábamos a los pines del Arduino para así poder prender los leds.

Empezamos implementando los chips haciendo un modelo en el que utilizábamos dos de ellos para prender una fila de leds en la protoboard, aquí el reto fue comprender como conectar dos chips a través de cables para pasar la información entre ellos así que nos guiamos de varias fuentes y videos para saber cómo hacerlo.

A partir de aquí nos vimos en la tarea de plantear como íbamos a hacer el circuito.

Y se nos ocurrieron dos modelos para hacerlo:

- El primero era hacer la implementación de ocho chips, para que cada uno de ellos controle una fila de leds y así el usuario podría controlar los leds que se prenden en cada fila.

- El segundo era la implementación de solo 2 chips para que uno de ellos controle las filas y el otro las columnas.

En este punto nos dividimos el trabajo, 2 de nosotros trabajaron en el primero mientras el otro trabajaba en el segundo modelo, al final concluimos que el primer modelo parecía ser más accesible y aunque era más complejo en cuanto a las conexiones de los chips unos con otros, solo necesitaba 3 pines digitales del Arduino, así que nos decidimos por este.

Luego de esto cada uno investigo la mejor manera de ingresar los datos para almacenarlos y poder representar el patrón de 1 y 0 en la matriz de 8*8.

3.3. Implementación del código:

En este punto nos dividimos el trabajo, 2 de nosotros trabajaron en el primero mientras el otro trabajaba en el segundo modelo, al final concluimos que el primer modelo parecía ser más accesible y aunque era más complejo en cuanto a las conexiones de los chips unos con otros, solo necesitaba 3 pines digitales del Arduino, así que nos decidimos por este. Luego de esto cada uno investigo la mejor manera de ingresar los datos para almacenarlos y poder representar el patrón de 1 y 0 en la matriz de 8*8. Este punto nos lo dividimos en varias partes, uno de nosotros desarrollaba el menú de las funcionalidades que le daríamos al programa para que este controle el Arduino, el otro implementaría el uso de memoria dinámica mientras otro hacia las funciones imagen, verificación, entre otras para luego unir el trabajo y completar el proyecto.

Nos vimos en la tarea también de investigar cómo funciona la entrada por serial del Arduino ya que nos estaba presentando algunos problemas así que entre todos fuimos investigando e indagando para encontrar la mejor manera de utilizarlo y acoplarlo a lo que estábamos pensando para plasmar los patrones en los leds.

Una de las partes más difíciles a la que nos tuvimos que enfrentar fue la comprensión del funcionamiento de la memoria dinámica, ya que a pesar de que teníamos el concepto nos dimos cuenta de que teníamos que interactuar mucho con él , ya que en el código intentamos hacer varias cosas como tratar de pasar la memoria dinámica para un arreglo de 3 dimensiones como parámetro a una función o en cuanto a la sintaxis de la declaración de una memoria dinámica nos dimos cuenta que habían varias formas de hacerlo a través de la investigación.

4. Algoritmo implementado para la solución (Descriptivamente)

Procedimiento o pasos para la solución del problema:

4.1. Electrónica de potencia:

Son todos los pasos principales que tuvieron que ver con encender los leds:

- Realizar una multiplexación de leds con sus respectivas conexiones de cable a protoboard. La multiplexación es una técnica empleada para operar matrices de LEDs.

- Implementación de resistencias.

4.2. Control de potencia

- Implementación de Arduino.

- Implementación de los 8 chips en la protoboard .

- Conexión de los pines digitales a las respectivas entradas del chip 74hc595.

- Unir los chips a través de cables.

4.3. Código

El código se presentaría en este orden: **Opción 1:** Verificar que todos los leds prendan. A esta opción le corresponde la función que hace que se prendan todos los leds para verificar que estén bien.

Opción 2: Ingresar un patrón para mostrarlo en la matriz. Aquí el usuario ingresa por el monitor serial 8 cifras de unos y ceros por cada fila representada por un byte, donde 1 seria que el led se prenda y 0 que el led esté apagado.

Opción 3: Mostrar una secuencia de patrones que el usuario elija, indicar el número de patrones y el tiempo de visualización entre cada patrón se le pide al usuario el número de patrones que desea ingresar y el tiempo de visualización entre ellos, después de imprimir los patrones le pregunta al usuario si quiere que esos patrones se impriman indefinidamente para ello se específica que caracteres o datos ingresar para cada caso.

Opción 4: Salir del programa. Si desea utilizar de nuevo el programa se debe reiniciar la simulación o el módulo del Arduino.

Nota: En caso de no ingresar en el menú alguno de los valores establecidos el programa mostrará un cartel que especifica que los datos ingresados son inválidos.Se deben ingresar los datos de la manera que dice el programa y el manual de usuario para que la expereiencia de uso se la mejor.

5. Problemáticas de desarrollo

En el desarrollo del problema contamos con múltiples problemas, puesto que para todos fue una novedad muy considerable el uso de la plataforma Arduino, aunque es verdad que el entorno de simulación Tinkercad brinda una forma intuitiva para su uso. En un principio fue una transición muy compleja de pasar de solo manejo de software a manejo de hardware y software, pero eso optamos en un principio en interactuar en la plataforma para disponernos a la solución del problema. También tuvimos muchas dificultades en encontrar un circuito estable, que funcionara con la lógica que teníamos pensada para la solución del

problema, nos costó al menos dos días encontrar la estructura del circuito que se adaptara a las necesidades. La problemática más importante fue el uso de la memoria dinámica, pues a partir de una continua investigación sobre el uso de arreglos en la misma, no pudimos establecer una forma de crear un arreglo de tres dimensiones en la memoria Heap, por lo que optamos por usar la memoria dinámica para guardar la matriz que ingresa el usuario en la función imagen.

6. Evolución del algoritmo y circuito

En cada sub-sección daremos con brevedad los aspectos que estamos teniendo en cuenta en la solcuión del problema planteado.

6.1. Aplicación en Arduino.

Para los tres esta transición al entorno de desarrollo Arduino no ha sido fácil, pero es verdad que nos estamos esforzando fielmente para nuestro aprendizaje en términos de aplicación en Hardware, y lograr que con el software se puedan solucionar problemas como el que se plantea en esta actividad.

Por ello, hemos estado interactuando y aprendiendo del funcionamiento del chip integrado llamado **74HC595**, el cual tiene como base principal el registro de corrimiento de información.

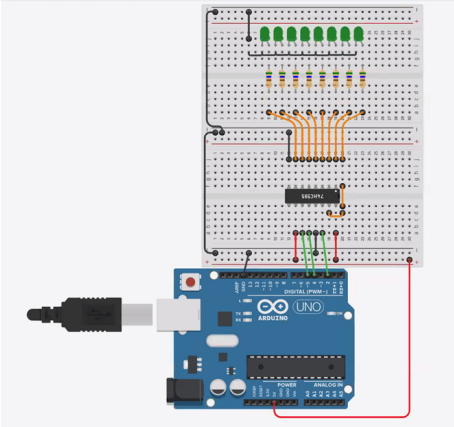

Figura 1: Arduino chip 74HC595.

El entendimiento de este chip es vital importancia para la solución del problema, por lo que adaptarnos al manejo del mismo, es el primer gran paso para conseguir hacer que 64 leds se enciendan o apaguen a nuestro deseo. Es por eso que poco a poco, estamos esclareciendo el camino para conseguir este significativo logro.

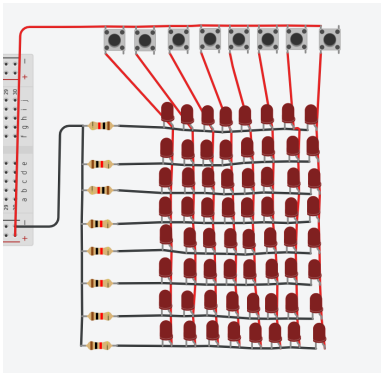

Figura 2: Matriz de leds 8x8 Off.

En la figura (2) se ejemplifica una tosca forma en la que tenemos pensada dar algún manejo a esa matriz leds. Y para que como se muestra en la figura (3) no solo podamos encender todos los leds, y si cada uno a nuestra voluntad, logrando mostrar los patrones que deseemos nosotros y principalmente los usuarios que usen nuestro proyecto. Pero consideramos que de este primer acercamiento vamos obtener grandes resultados.

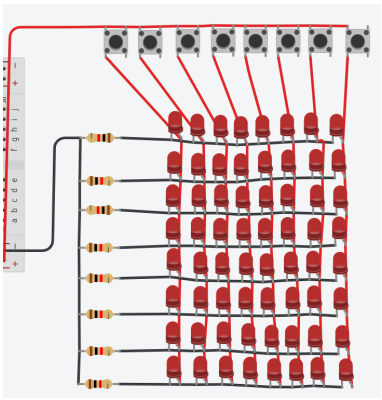

Figura 3: Matriz de leds 8x8 Off.

6.2. Unión de dos registros.

Con el fin de hacer mas claras nuestras ideas sobre el funcionamiento del chip **74HC595**, integramos dos registros de desplazamiento juntos uniendo los cables del reloj de la primera (SRCLK) y segunda etapa (RCLK) de registro entre sí y el pin SER llamado pindata por el que ingresamos la información, pasa en primera instancia a un registro y al completarse su capacidad de 8, y gracias al cable de color verde con el que unimos la salida del primer integrado a la entrada SER del segundo integrado, podemos ampliar las salidas del arduino a 16 usando solo 3 pines digitales.

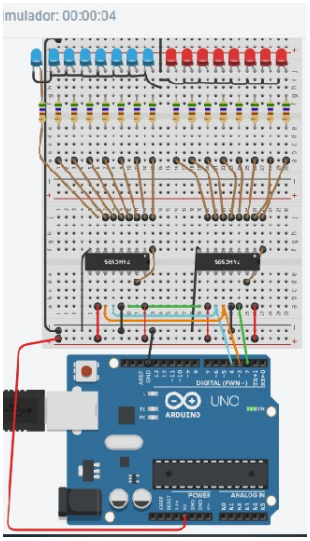

Figura 4: Manejo del chip **74HC595**.

Consideramos que es un gran avance pero ahora solo debemos poner nuestro objetivo en cuadriplicar el manejo de registros con el chip **74HC595**.

6.3. Cicrcuito con 9 integrado:

Con los análisis previos, hemos tomado la iniciativa de elaborar un circuito más elaborado y completo, sobre el cual hicimos múltiples pruebas, aunque los resultados no fueron los deseados, puesto que la lógica consistía en tener cuatro pines digitales que funcione para relojes dos para un chip y otros dos para los otros ocho chips. Pero nos dimos cuenta que el trabajo computacional para este proyecto es demasiado alto e innecesario por lo que estudiaremos la forma de modificar el circuito.

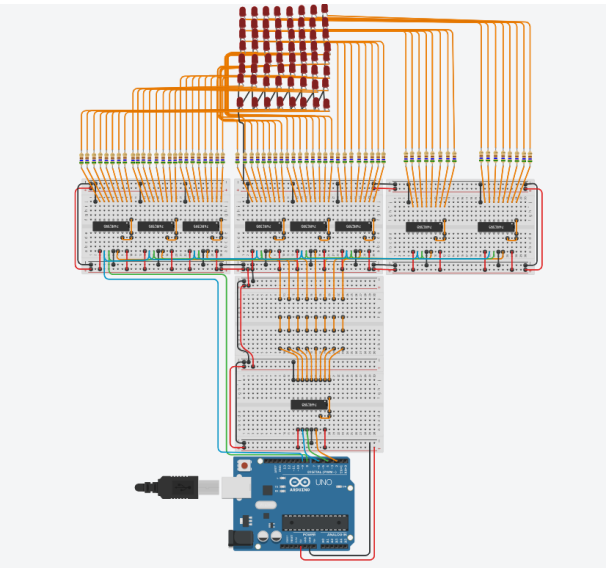

Figura 5: Circuito nueve chips **74HC595**.

```
//Apartir de este c digo hemos estado interactuando con el circuito presentado.

const int SER=2;
const int RCLK1=4;
const int SRCLK1=6;
const int RCLK2=8;
const int SRCLK2=9;

void encender(){
for(int i=0; i<64; i++)
{
  digitalWrite(SER, 1);

  digitalWrite(SRCLK1, 0);
  digitalWrite(SRCLK1, 1);
  digitalWrite(SRCLK1, 0);

  digitalWrite(RCLK1, 0);
  digitalWrite(RCLK1, 1);
  digitalWrite(RCLK1, 0);

  digitalWrite(SRCLK2, 0);
  digitalWrite(SRCLK2, 1);
  digitalWrite(SRCLK2, 0);

  digitalWrite(RCLK2, 0);
  digitalWrite(RCLK2, 1);
  digitalWrite(RCLK2, 0);
  }
  //delay(1000);

}
void apagar(){

  for(int i=0; i<64; i++)
{
  digitalWrite(SER, 0);

  digitalWrite(SRCLK1, 0);
  digitalWrite(SRCLK1, 1);
```

```
  digitalWrite( SRCLK1, 0);

  digitalWrite( RCLK1, 0);
  digitalWrite( RCLK1, 1);
  digitalWrite( RCLK1, 0);

  digitalWrite( SRCLK2, 0);
  digitalWrite( SRCLK2, 1);
  digitalWrite( SRCLK2, 0);

  digitalWrite( RCLK2, 0);
  digitalWrite( RCLK2, 1);
  digitalWrite( RCLK2, 0);
  }
}

void setup()
{
  pinMode(SER, OUTPUT);
  pinMode(RCLK1, OUTPUT);
  pinMode(SRCLK1, OUTPUT);
  pinMode(RCLK2, OUTPUT);
  pinMode(SRCLK2, OUTPUT);

  digitalWrite(SER, LOW);
  digitalWrite(RCLK1, LOW);
  digitalWrite(SRCLK1, LOW);
  digitalWrite(RCLK2, LOW);
  digitalWrite(SRCLK2, LOW);



}

void loop()
{

 encender();
 delay(1000);
 apagar();

}
```

6.4. Programa implemenetado en C++

Como lo habiamos mencioando con aterioridad decidimos plantear una solución a la problemática con el código que se muestra a continuación.

```
#include <iostream>

using namespace std;
void imprimirleds(int asientos[8][8]);
void ingresardatos(int matriz[8][8]);
void ingresardatos2(int matriz[][8][8], int n);
void imprimirleds2(int asientos[][8][8], int n);
void publik(void);

int main()
{
    int inicializar[8][8]={{1, 1 ,1 ,1, 1, 1, 1, 1}, {1, 1 ,1 ,1, 1, 1, 1, 1}, {
    int borrar[8][8]={{0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0}, {0, 0,
    int matriz[8][8]={};
    int N;


     bool band=true;

    cout << "-----------------------------------------------------------"<<endl;
    cout << "|      BIENVENIDO AL GRAFICADOR DE LEDS 8X8      |"<<endl;

         while(band){

        cout << "-----------------------------------------------------------"<<endl;
        cout << "|                    MENU.                  |"<<endl;
        cout << "-----------------------------------------------------------"<<endl;
        cout << "1. Verificar funcionamiento de leds. "<<endl;
        cout << "2. Ingresar un patron para mostrarlo en la matriz. "<<endl;
        cout << "3. Ingresar varios patrones para ser mostrado en
        distintos intervalos de tiempo que debe ingresar. "<<endl;
         cout << "4. Salir del programa. "<<endl;


         cout << "Ingresa una opcion: ";
         cin >> N;
         \\\
\\\
\\\
\\\
\\\
```

```
    switch (N) {

    case 1:
    cout << "------------------------------------------------------------"<<endl;
    cout << "       VERIFICACION DE LA MATRIZ DE LEDS."<<endl;
    cout << "------------------------------------------------------------"<<endl;

         imprimirleds(inicializar);
         break;
    case 2:
    cout << "------------------------------------------------------------"<<endl;
    cout << "INGRESA TU PATRON PARA MOSTRARLO EN LA MATRIZ DE LEDS."<<endl;
    cout << "------------------------------------------------------------"<<endl;
    ingresardatos(matriz);
    imprimirleds(matriz);


         break;
    case 3:
    cout << "-----------------------------------------"<<endl;
    cout << "INGRESA LOS PATRONES QUE DESEES Y EL INTERVALO PARAMOSTRARLOS"<
    cout << "-----------------------------------------"<<endl;

         publik();


         break;

case 4:
cout << "------------------------------------------------------------"<<endl;
cout << "       GRACIAS POR USAR NUESTRO SERVICIO."<<endl;
cout << "------------------------------------------------------------"<<endl;


        band =false;
         break;




    default:
    cout << "------------------------------------------------------------"<<endl;
    cout << "                DATOS INVALIDOS"<<endl;
   cout << "------------------------------------------------------------"<<endl;
```

```
     }
     }
    return 0;
}


void publik(void){

    int N, n=0;

    cout << "Ingrese el numero de patrones: ";
    cin >> N;

    cout << "Ingrese el tiempo en (segundos) entre un patron y otro: ";
    cin >> n;

    int publik[N][8][8]={};

    for(int i=0; i<N; i++){

          ingresardatos2(publik, i);


    }
    for (int j=0; j<N; j++){
        for (int k=1; k<=n; k++){

        cout << k << " segundo."<<endl;

        }
        imprimirleds2(publik, j);
    }

}
void ingresardatos2(int matriz[][8][8], int n){


  bool band=true;
  cout << "Ingrese cada fila de su patron, donde un 0 es
  un led apagado, y un 1 es un led prendido, ingrese un Byte
  representando una fila."<<endl;

  unsigned long int byte;
  int bit, A=1, k=0;
```

```
  while(band){


    cout <<A<< ". Ingrese un Byte: "<<endl;
    cin >> byte;
    cout <<endl;

    for (int i=7; i>=0 ;i--){
            bit=byte%10;


              matriz[n][k][i]=bit;
              byte/=10;


            }

    if(A==8){
    band=false;
    }
    A++;
    k++;

  }


}

void ingresardatos(int matriz[8][8]){


  bool band=true;
  cout << "Ingrese cada fila de su patron, donde un 0 es un led
  apagado, y un 1 es un led prendido, ingrese un Byte representando
  una fila."<<endl;

  unsigned long int byte;
  int bit, A=1, k=0;

  while(band){


    cout <<A<< ". Ingrese un Byte: "<<endl;
    cin >> byte;
    cout <<endl;
```

```
    for (int i=7; i>=0 ;i--){
            bit=byte%10;



              matriz[k][i]=bit;
              byte/=10;


            }

    if(A==8){
    band=false;
    }
    A++;
    k++;

  }


}

void imprimirleds(int asientos[8][8]){


              for (int k=1; k<=17; k++){

                    if (k!=17){

                    cout << "- ";

                    }
                    else{

                       cout << "-"<<endl;

                    }
                }

              for (int j=0; j<8; j++){


                    for (int i=0; i<8; i++){

                          if (i==0 ){
```

```
                    cout << "| "<<asientos[j][i]<<" | ";

                    }
                    else{
                        if(i==7){

                         cout <<  asientos[j][i]<<" | "<<endl;


                        }
                        else{

                            cout <<  asientos[j][i]<<" | ";

                        }

                    }
                }

            for (int k=1; k<=17; k++){

                    if (k!=17){

                    cout << "- ";

                    }
                    else{

                      cout << "-"<<endl;

                    }
                }

        }

}

void imprimirleds2(int asientos[][8][8], int n){


            for (int k=1; k<=17; k++){

                    if (k!=17){

                    cout << "- ";
```

```
            }
            else{

              cout << "-"<<endl;

            }
        }

    for (int j=0; j<8; j++){


        for (int i=0; i<8; i++){

                if (i==0 ){

                cout << "| "<<asientos[n][j][i]<<" | ";

                }
                else{
                    if(i==7){

                     cout <<  asientos[n][j][i]<<" | "<<endl;


                    }
                    else{

                        cout <<  asientos[n][j][i]<<" | ";

                    }

                }
        }

        for (int k=1; k<=17; k++){

                if (k!=17){

                cout << "- ";

                }
                else{

                  cout << "-"<<endl;

                }
```

```
                    }
                }
}
```

Que nos ayudó como guía en la implementación de la solución en la plataforma Arduino sobre el circuito final que desarrollamos.

6.5. Cicrcuito final e implemetanción de la solución

Al interactuar con nuestro circuito con nueve integrados analizamos que era mejor optar por manejar 8 integrados en vez de uno, puesto que era mejor hacer el desplazamiento la primera fila del patrón desde el último integrado (en relación de derecha a izquierda) hasta el primer integrado para de esta manera ir completando cada integrado con respectivos datos en concordancia a fila que se ingresa, y por último activar el reloj de salida, para mostrar el reloj en la matriz de leds.

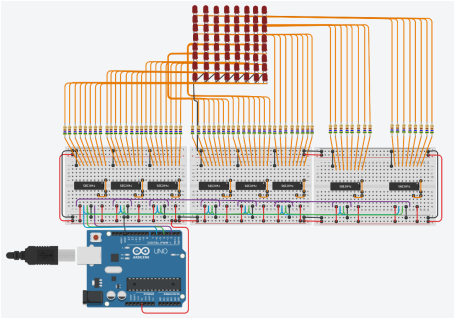

Figura 6: Circuito Final

A continuación se presentará el código implementado en la plataforma de arduino: (Está adpatado para la correcta visualización en el informe)

```
//-------------------------------------------------
//Puertos digitales a usar.
//-------------------------------------------------

const int SER=2;//Entrada de datos
const int RCLK=4;//Reloj 2(salida)
const int SRCLK=6;//Reloj 1(entrada)
//-------------------------------------------------
//
//-------------------------------------------------


//-------------------------------------------------
//Prototipo de las funciones implementadas.
//-------------------------------------------------

void publik(void);
void apagar(void);
void ingresardatos(int **matriz);
void ingresardatos2(int matriz[][8][8], int n);
void verificacion(void);
void imagen1(int **matriz);
void imagen2(int matriz[][8][8], int n);

//-------------------------------------------------
//Fin de prototipo de las funciones implementadas.
//-------------------------------------------------


void setup()
{
  pinMode(SER, OUTPUT);
  pinMode(RCLK, OUTPUT);
  pinMode(SRCLK, OUTPUT);
  Serial.begin(9600);

  int **pt=new int *[8];
  for (int i=0; i<8; i++){
            pt[i]=new int   [8];

        }
        for (int j=0; j<8; j++){
            for (int k=0; k<8; k++){
                pt[j][k]=0;

            }
```

```
    }

//-------------------------------------------------
//          INICIO DEL PROGRAMA.
//-------------------------------------------------
int N;
bool band=true;


Serial.println("-------------------------------------------------------------");
Serial.println("|      BIENVENIDO AL GRAFICADOR DE LEDS 8X8      |");

    while(band){

    Serial.println("-------------------------------------------------------------");
    Serial.println("|                    MENU.                    |");
    Serial.println("-------------------------------------------------------------");

    Serial.println("1. Verificar funcionamiento de leds. ");
    Serial.println("2. Ingresar un patron para mostrarlo en la matriz. ");
    Serial.println("3. Ingresar varios patrones para ser mostrado en distin
    Serial.println("4. Salir del programa. ");


    Serial.println("Ingresa una opcion: ");

    while(Serial.available()==0){}
    N=Serial.parseInt();

    switch (N) {

    case 1:
        Serial.println("-------------------------------------------------------
        Serial.println( "       VERIFICACION DE LA MATRIZ DE LEDS.");
        Serial.println( "-------------------------------------------------------

        //encenderM();
        verificacion();
        delay(10000);
        apagar();

        break;
    case 2:
        Serial.println( "-------------------------------------------------------
        Serial.println( "INGRESA TU PATRON PARA MOSTRARLO EN LA MATRIZ DE L
```

```
            Serial.println( "-----------------------------------------------------------------

            ingresardatos(pt);
            imagen1(pt);
            delay(10000);
            delete [] pt;
            apagar();


            break;
        case 3:
            Serial.println( "-----------------------------------------------------------------
            Serial.println( "INGRESA LOS PATRONES QUE DESEES Y EL INTERVALO PAR
            Serial.println( "-----------------------------------------------------------------

            publik();


            break;

        case 4:
            Serial.println( "-----------------------------------------------------------------"
            Serial.println( "        GRACIAS POR USAR NUESTRO SERVICIO.");
            Serial.println( "-----------------------------------------------------------------"


            band =false;
            break;
        default:
            Serial.println( "-----------------------------------------------------------------"
            Serial.println( "                DATOS INVALIDOS");
            Serial.println( "-----------------------------------------------------------------"



    }
    }

    //-------------------------------------------
    // FIN DEL PROGRAMA.
    //-------------------------------------------
}

void loop()
{
```

```
}

//-------------------------------------------
//Definicion de las funciones del programa.
//-------------------------------------------

//Funcio que una vez se le ingrese una arreglo 8x8 vac o , pide datos
void ingresardatos(int **matriz){


  bool band=true;
  Serial.println("Ingrese cada fila de su patron, donde un 0 es un led apagado, 


  unsigned long int byte;
  int bit, A=1, k=0;

  while(band){



    Serial.print(A);
    Serial.println(". Ingrese un Byte: ");
    while(Serial.available()==0){}
    byte=Serial.parseInt();
    Serial.println(byte);

    Serial.println("");

    for (int i=7; i>=0 ;i--){
          bit=byte%10;



      matriz[k][i]=bit;

      byte/=10;


          }

    if(A==8){
    band=false;
    }
    A++;
    k++;
```

```
  }
  digitalWrite( RCLK, 0);
  digitalWrite( RCLK, 1);
  digitalWrite( RCLK, 0);



}
void ingresardatos2(int matriz[][8][8], int n){


  bool band=true;
  Serial.println("Ingrese cada fila de su patron, donde un 0 es un led apagado, 

   unsigned long int byte;

  int bit, A=1, k=0;

  while(band){


    Serial.print(A);
    Serial.println(". Ingrese un Byte: ");
    while(Serial.available()==0){}
    byte=Serial.parseInt();
    Serial.println(byte);


    Serial.println("");

    for (int i=7; i>=0 ;i--){
            bit=(byte)%10;


              matriz[n][k][i]=bit;
              (byte)/=10;


            }

    if(A==8){
    band=false;
    }
    A++;
    k++;
```

```
  }


}
void publik(void){

  unsigned int N, n;

  Serial.print("Ingrese el numero de patrones: ");
  while(Serial.available()==0){}
  N=Serial.parseInt();
  Serial.println(N);

  Serial.print("Ingrese el tiempo en (segundos) entre un patron y otro: ");
  while(Serial.available()==0){}
  n=Serial.parseInt();
  Serial.println(n);

  int publik[N][8][8]= {};


    for(int i=0; i<N; i++){
      Serial.print("Patron: ");
      Serial.println(i+1);

          ingresardatos2(publik, i);


    }
  for (int j=0; j<N; j++){

        imagen2(publik, j);
        //apagar();
        delay(n*1000);
    }
   apagar();


   Serial.println("Desea imprimir los patrones indefinidamente hasta ingresar el
   Serial.println("Ingrese Y, para SI, cualquier otro caracter para NO");

   char letra;
   int  bit2=0;
   char band;
```

```
    while(Serial.available()==0){}
    letra=Serial.read();
    if(letra=='Y'){
     Serial.println("Imprimiendo patrones...");
    Serial.println("Ingrese X, para dejar de mostrar patrones.");

     while(band!='X'){
         band=Serial.read();


       for (int k=0; k<N &&  band!='X'; k++){
         band=Serial.read();
         imagen2(publik, k);

         delay(n*1000);

       }




      }

     }
   else{

   Serial.println("Entrando al menu...");
   Serial.println("");

   }
   //delete [] publik;
    apagar();

}

void apagar(void){

 int tamanio=8;

   for (int i=0;i<tamanio;i++){//filas
     for (int j=0;j<tamanio;j++){//columnas
       digitalWrite( SER, 0);

       digitalWrite( SRCLK, 0);
       digitalWrite( SRCLK, 1);
       digitalWrite( SRCLK, 0);
```

```
    }
  }

  digitalWrite( RCLK, 0);
  digitalWrite( RCLK, 1);
  digitalWrite( RCLK, 0);

  Serial.println("leds-Off");
  Serial.println("");

}

void verificacion(void){
  int tamanio=8;
  for (int i=0;i<tamanio;i++){//filas
    for (int j=0;j<tamanio;j++){//columnas
      digitalWrite( SER, 1);

      digitalWrite( SRCLK, 0);
      digitalWrite( SRCLK, 1);
      digitalWrite( SRCLK, 0);


    }
  }
      digitalWrite( RCLK, 0);
      digitalWrite( RCLK, 1);
      digitalWrite( RCLK, 0);



  Serial.println("Pruebas finalizadas: Leds encendidos");
  Serial.println("");
}

void imagen1(int **matriz){
  int tamanio=8;
  int bit=0;
  unsigned long int byte=0;
  for (int i=0;i<tamanio;i++){//filas

    for (int j=0;j<tamanio;j++){//columnas
     bit=matriz[i][j];
```

```
        if (bit == 1){

          digitalWrite( SER, 1);

          digitalWrite( SRCLK, 0);
          digitalWrite( SRCLK, 1);
          digitalWrite( SRCLK, 0);




        }
        else if(bit==0){
          digitalWrite( SER, 0);

          digitalWrite( SRCLK, 0);
          digitalWrite( SRCLK, 1);
          digitalWrite( SRCLK, 0);



        }

      }


    }
    //Serial.println("Imprimiendo patrones...");
    digitalWrite( RCLK, 0);
    digitalWrite( RCLK, 1);
    digitalWrite( RCLK, 0);
    Serial.println("");



}
void imagen2(int matriz[][8][8], int n){
  int tamanio=8;
  int bit=0;
  unsigned long int byte=0;
  for (int i=0;i<tamanio;i++){//filas

    for (int j=0;j<tamanio;j++){//columnas
     bit=matriz[n][i][j];
      if (bit == 1){
```

```
        digitalWrite( SER, 1);

        digitalWrite( SRCLK, 0);
        digitalWrite( SRCLK, 1);
        digitalWrite( SRCLK, 0);




      }
      else if(bit==0){
        digitalWrite( SER, 0);

        digitalWrite( SRCLK, 0);
        digitalWrite( SRCLK, 1);
        digitalWrite( SRCLK, 0);



      }

    }

  }

  //Serial.println("Imprimiendo patrones...");
  digitalWrite( RCLK, 0);
  digitalWrite( RCLK, 1);
  digitalWrite( RCLK, 0);
  Serial.println("");

}
//--------------------------------------------------
//Definicion de las funciones del programa.
//--------------------------------------------------
```

6.6. Sobre la implementación en arduino:

Procuramos hacer de la manera más intuitiva la interfaz gráfica de nuestro programa, para que a través del monitor Serial de Arduino, se pueda interactuar de la manera adecuada al programa. Además, teniendo en cuenta las limitaciones en términos de memoria de esta plataforma, también procuramos implementar la memoria dinámicas en la función "imagen" para que los patrones sean guardados en la memoria Heap y hacer que la memoria Stack esté más liviano para un mejor funcionamiento.