

# Parcial 1

Desarrollo del proyecto

**Ana María Ardila Ariza**  
**Brayan Gomez Carmona**

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

## Índice

1. Análisis del problema y consideraciones para la alternativa de solución propuesta	2
2. Esquema donde describimos las tareas para el desarrollo del algoritmo.	2
3. Problemas de desarrollo que presentó.	2
4. Algoritmo implementado	3
5. consideraciones a tener en cuenta en la implementación.	6

## 1. Análisis del problema y consideraciones para la alternativa de solución propuesta

El problema busca usar nuestros conocimientos, habilidades y destrezas para resolver y crear algo de nuestra vida cotidiana como lo son los anuncios digitales. A lo largo de nuestros tiempos y en la era digital, ha habido diversas innovaciones en el sector publicitario y comercial, entre ellas los letreros con iluminación, el cual es nuestro objetivo para realizar pensando en esto como un proyecto de aplicación a la vida de nuestra carrera.

Para solucionar esto consideramos diversos factores, comodidad del usuario, eficiencia y la aplicación de nuestros conocimientos.

En primera instancia, la comodidad del usuario es fundamental ya que es a el que va dirigido este programa, por lo que el programa debe ser entendible y cómodo para que el usuario esté satisfecho.

En segunda instancia tenemos la eficiencia, creando un programa que funcione correctamente y con una eficiencia alta, a una velocidad buena y tratar de optimizarlo de la mejor manera posible.

Y, por último, la aplicación de nuestros conocimientos e investigación para desarrollar este proyecto de la mejor manera, cumpliendo el objetivo planteado, y teniendo muy en cuenta la segunda y primera instancia.

## 2. Esquema donde describimos las tareas para el desarrollo del algoritmo.

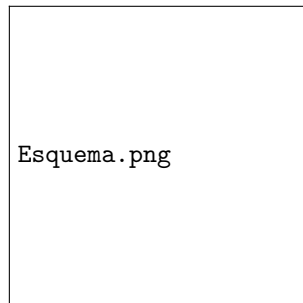


Figura 1: Esquema

## 3. Problemas de desarrollo que presentó.

Durante el desarrollo del algoritmo sobresalieron muchos problemas, los cuales hicieron que en múltiples momentos nos replanteáramos la lógica implementada e incluso el código en sí. Unos ejemplos de estos problemas fueron que al

tratar de usar la lógica de filas- columnas, no sabíamos como hacer que el usuario pudiera ingresar e imprimir en el letrero lo que entregara; otro problema fue la incógnita de si el código era lo suficientemente eficiente, y a la vez de fácil entendimiento, otro problema fue el de manejar el lenguaje c++ implementándolo en tinkercard, ya que el manejo de pines y la lógica de esta nos era ambigua aun antes de iniciar este proyecto. Estas y muchas mas dificultades se nos dieron a la hora de desarrollar este problema, las cuales tratamos de solucionar de la mejor forma para poder cumplir con nuestro objetivo.

## 4. Algoritmo implementado

```

const int data = 4;
const int latch = 5;
const int clock = 6;

int filas [8]={127,191,223,239,247,251,253,254};//Estas son las filas a barrer.
bool w;
int columnas [8] , sel , n, nFilas=0,c=0,*f,**p_matriz , nPat;

void *pedirdatos(){
    Serial.println("Cuantos_dibujos_desea_ingresar?:");
    while(Serial.available()==0){
        }

    nFilas=Serial.parseInt();

    p_matriz = new int*[nFilas]; //Memoria para las filas
    for(int i=0;i<nFilas;i++){
        p_matriz[i] = new int [8];
    }
    //Recoger los datos en la matriz
    for(c=0;c<nFilas;c++){
        Serial.print("Dibujo ");
        Serial.println(c+1);

        for(int j=0;j<8;j++){
            Serial.print("Secuencia ");
            Serial.println(j+1);
            while(Serial.available()==0){
                }
            n=Serial.parseInt();
            *((p_matriz+c)+j)= n;
        }
    }

```

```

    }
    int *p;
    p=new int [8]; //Me crea un arreglo para ingresarlo a la funcion imagen
    for (int k=0;k<8;k++){
        *(p+k)=*(*(p-matriz+c)+k);
    }
    f = p;
    delete [] p; // Elimina el arreglo de la memoria porque ya est en los chips
    imagen(f);
}

void verificar(){
    for (int i=0;i<8;i++){
        columnas[i]=255;
    }
}

bool GetBit( byte n, int pos)//Esta funcion me saca el bit en la posicion deseada
{
    int b = n >> pos ;
    b = b & 1 ;
    return b ;
}

void imagen(int col[]){ //Esta funcion recibe un arreglo con las posiciones y lo

    for (int x=0;x<8;x++){
        for (int i=0;i<8;i++){ //Este for escribe los 8 bits de cada columna.
            w=GetBit( col[x], i );
            if (w==true){
                digitalWrite(data,HIGH);
                digitalWrite(clock,HIGH);
                digitalWrite(clock,LOW);
            }

            else{
                digitalWrite(data,LOW);
                digitalWrite(clock,HIGH);
                digitalWrite(clock,LOW);
            }
        }
        for (int j=0;j<8;j++){ //Este for escribe los 8 bits de la fila y empuja a lo
            w=GetBit( filas[x], j );
            if (w==true){
                digitalWrite(data,HIGH);
                digitalWrite(clock,HIGH);
                digitalWrite(clock,LOW);
            }

            else{

```

```

        digitalWrite(data,LOW);
        digitalWrite(clock,HIGH);
        digitalWrite(clock,LOW);
    }

}
//Cuando ya todos los bits estan en su posicion activo la salida:
digitalWrite(latch,HIGH);
digitalWrite(latch,LOW);
}
}
int *patrones(int a){
    //Me crea un arreglo para ingresarlo a la funcion imagen
    int *p;
    p=new int [8];
    for(int k=0;k<8;k++){
        *(p+k)=*(*(p_matriz+a)+k);
    }
    return p;

    delete [] p;
}
void setup()
{
    pinMode(data, OUTPUT);
    pinMode(latch, OUTPUT);
    pinMode(clock, OUTPUT);
    Serial.begin(9600);
}
void loop(){//Imprime el men

    Serial.println("1._Verificar");
    Serial.println("2._Dibujar");
    Serial.println("3._mostrar_patrones");

    while(Serial.available() == 0){ //Mientras no se ingrese nada por el serial ej

        for(int i=0;i<nPat;i++){
            f=patrones(i);

            for(int k=0;k<20;k++){
                imagen(f);
            }
            delay(100);
        }
    }
}

```

```

sel=Serial.parseInt();
switch(sel){

  case 1:
    verificar(); //Me muestra que todos los leds funcionan
    imagen(columnas);
    break;

  case 2:

    pedirdatos(); //Aqui se ingresa el dibujo, dentro de la funcion se elije cua
    break;

  case 3:
    Serial.println("Cuantos_patrones_desea_ver:_");
    nPat=Serial.parseInt();
    break;

}

}

```

## 5. consideraciones a tener en cuenta en la implementación.

Para la correcta implementación del algoritmo, es necesario descargar el archivo de Excel adjuntado al repositorio de GitHub, porque esto facilita el ingreso de los datos ya que este programa recibe números en decimal y los convierte a binario, lo que hace el archivo de Excel automáticamente con la imagen o carácter que se quiere mostrar. Por último, se debe escoger una de las opciones en el menú que se imprime en la consola de tinkercard.