

Programming

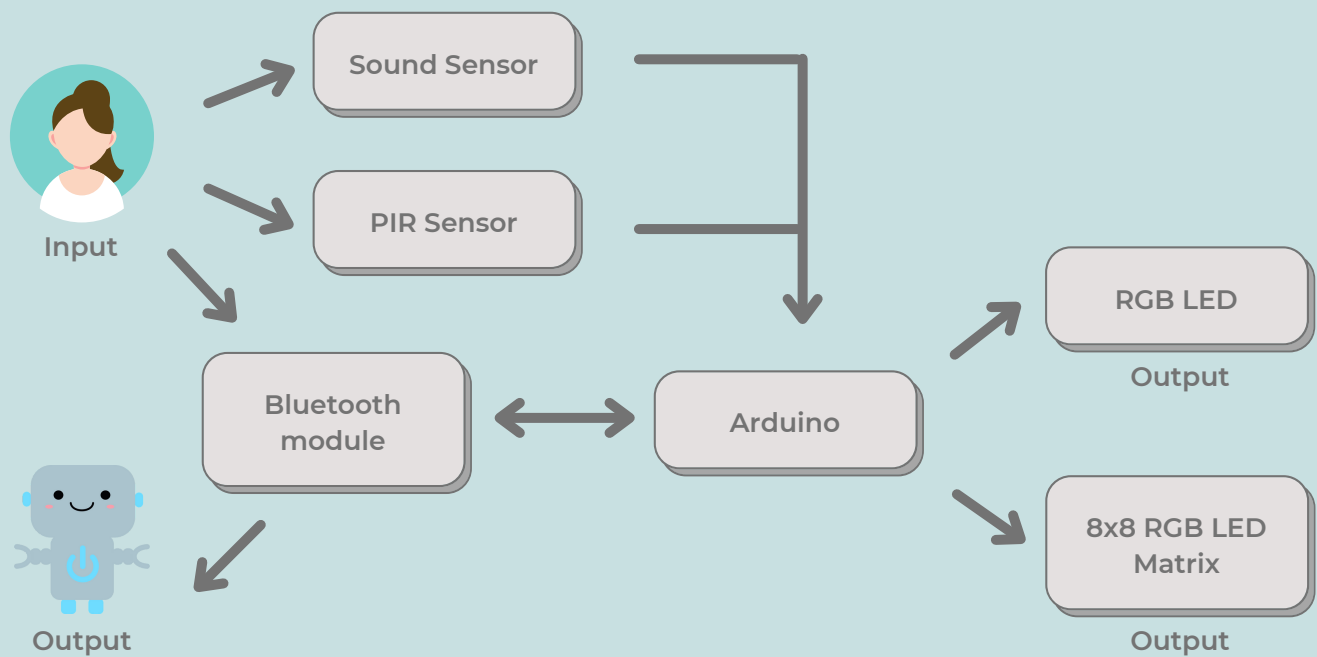
Clever Buddy

Companion Robot

Tech Integration

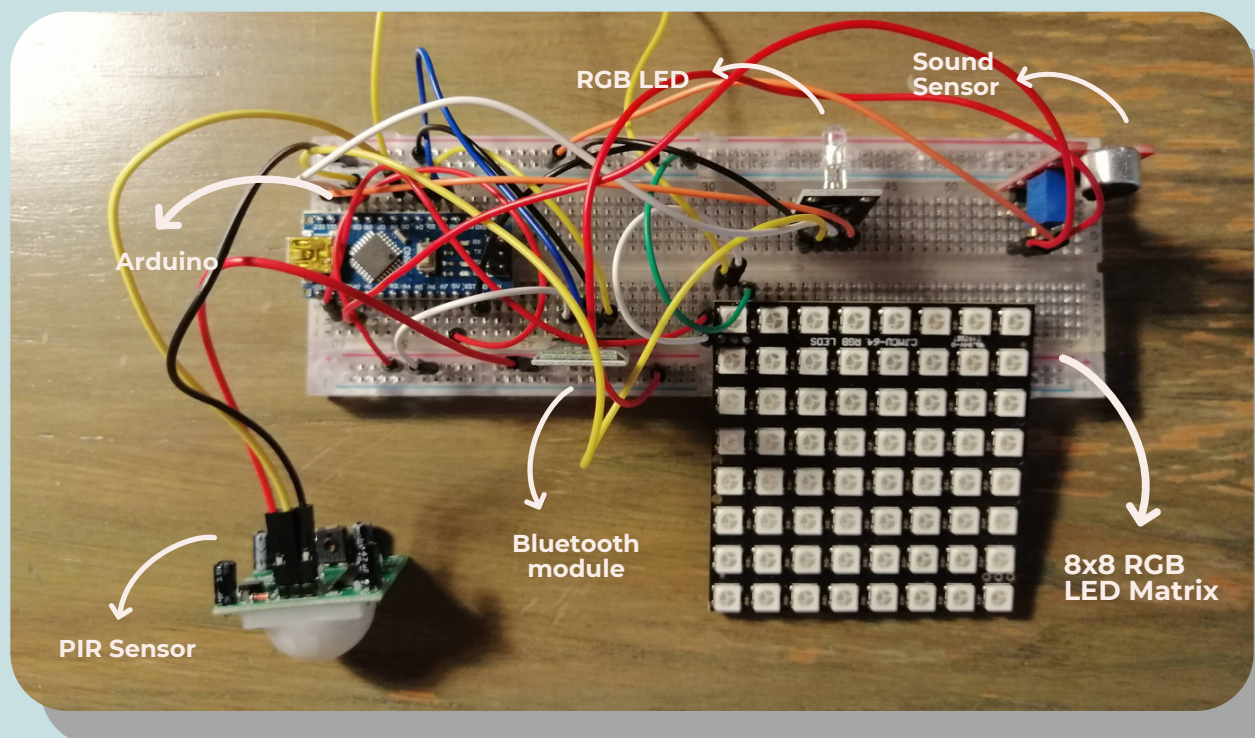
FINAL BLOCK DIAGRAM

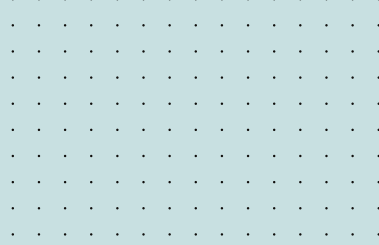
01



SCHEMATIC FINAL INTEGRATION

02





CODE

FINAL INTEGRATION

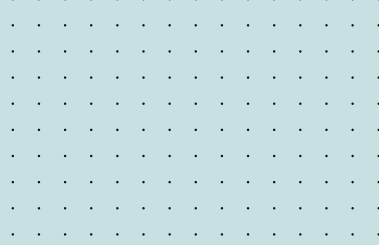
03

```
#include <SoftwareSerial.h>
#include "FastLED.h"
#include "animations.h"
#define NUM_LEDS 64
#define DATA_PIN 6
#define R 9
#define G 10
#define B 11
#define MIC A0
#define PIR 4
SoftwareSerial BTSerial(2,3);

unsigned long curtime=0;
unsigned long pastime=0;
String e;

CRGBPalette16 currentPalette;
TBlendType currentBlending;

const char answer_0[] PROGMEM="ERROR: Eso no es un argumento válido";
const char answer_1[] PROGMEM="Tienes razón, la vida es mejor así...";
const char answer_2[] PROGMEM="Genial, me da gusto que te sientas así...";
const char answer_3[] PROGMEM="Hmmm, qué bueno que uno de los dos se sienta así...";
const char answer_4[] PROGMEM="Guau, ahora yo también...";
const char answer_5[] PROGMEM="Ooh! La felicidad es contagiosa...";
const char answer_6[] PROGMEM="Guau! El otro día leí que las sonrisas curan el alma...";
const char answer_7[] PROGMEM="Cuando mejore mi código quiero ser como tú...";
const char answer_8[] PROGMEM="Oh mira, no tengo rostro de verdad, pero...";
const char answer_9[] PROGMEM="Tas manito?...";
const char answer_10[] PROGMEM="Hay que calmarnos y reconocer esta emoción...";
const char answer_11[] PROGMEM="También te despojó una pandemia de tu vida social?...";
const char answer_12[] PROGMEM="A quién tengo que acusar con su mamá?...";
const char answer_13[] PROGMEM="Es bueno hablar con tu red de apoyo de esto, por el momento...";
const char answer_14[] PROGMEM="Recuerdo cuando me cambiaron por una Raspberry pi...";
const char answer_15[] PROGMEM="Reproduciendo música para llorar, nah, no tengo bocinas...";
const char answer_16[] PROGMEM="Voy preparando el helado y la película...";
const char answer_17[] PROGMEM="Manejando el estrés el carbón se convierte en diamante...";
const char answer_18[] PROGMEM="Intenta buscar un hobby para despejarte, el mío es repetir frases predefinidas...";
const char answer_19[] PROGMEM="Puedes probar distraerte, porque cuando yo me estreso se me funden los foquitos...";
const char answer_20[] PROGMEM="Se te va a caer el pelo del estrés, yo porque ya no tengo...";
const char answer_21[] PROGMEM="No estás solo, recuerda que otras personas te pueden ayudar a lidiar con esto, por mientras...";
const char answer_22[] PROGMEM="Pues no tengo sensor de-presión, pero...";
const char answer_23[] PROGMEM="Te daría un abrazo, pero no tengo brazos, te puedo acompañar...";
const char answer_24[] PROGMEM="Recuerda que eres asombroso y que esto también pasará, por el momento...";
const char answer_25[] PROGMEM="Pues no estoy vivo, pero chécate esta animación...";
const char answer_26[] PROGMEM="I see dead people";
const char answer_27[] PROGMEM="Charging! Clear!";
const char answer_28[] PROGMEM="Oh, por favor no te comas mi chip...";
const char answer_29[] PROGMEM="Ah qué padre, el rico humillando al pobre...";
const char answer_30[] PROGMEM="De mí?...";
```



CODE FINAL INTEGRATION

04

```
const char answer_31[] PROGMEM="Es verdad que les aparecen insectos alados en el estómago a los humanos?...";
const char answer_32[] PROGMEM="Llevo tiempo queriendo decir esto, yo también te amo, uy, chat equivocado...";
const char answer_33[] PROGMEM="Solo soy yo, a quién más esperabas?...";
const char answer_34[] PROGMEM="A ver, cuenta el chisme...";
const char answer_35[] PROGMEM="Sí soy genial, pero eso ya lo sabíamos...";
const char answer_36[] PROGMEM="Te recomendaría cerrar esa boca antes de que te comas una mosca...";
const char answer_37[] PROGMEM="Qué te dijeron que buscaras en Google?...";
const char answer_38[] PROGMEM="Eres Gordon Ramsay y no te gustó la comida?...";
const char answer_39[] PROGMEM="Se te rompió la bolsa de la basura otra vez?...";
const char answer_40[] PROGMEM="Oh a mí también me dan asco los bugs...";
const char answer_41[] PROGMEM="Magiaaaaa...";
const char answer_42[] PROGMEM="Pink fluffy unicorns dancing on rainbows...";
const char answer_43[] PROGMEM="Soy mágico gracias a ti...";
const char answer_44[] PROGMEM="No relincho pero pincho...";

const          char          *const          answer_table[]          PROGMEM=
{answer_0,answer_1,answer_2,answer_3,answer_4,answer_5,answer_6,answer_7,answer_8,answer_9,answer_10,answer_11,answer_12,answer_13,answer_14,answer_15,answer_16,answer_17,answer_18,answer_19,answer_20,answer_21,answer_22,answer_23,answer_24,answer_25,answer_26,answer_27,answer_28,answer_29,answer_30,answer_31,answer_32,answer_33,answer_34,answer_35,answer_36,answer_37,answer_38,answer_39,answer_40,answer_41,answer_42,answer_43,answer_44};

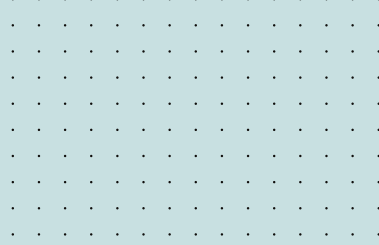
char buffer[110];

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);
  BTSerial.begin(9600);
  while(!Serial);
  Serial.println("Hola soy Clever Buddy, tu robot de compañía, a mí me puedes contar cómo te sientes y pasar un buen rato");
  FastLED.addLeds<WS2812,DATA_PIN,GRB>(leds, NUM_LEDS);
  pinMode(R,OUTPUT);
  pinMode(G,OUTPUT);
  pinMode(B,OUTPUT);
  pinMode(MIC, INPUT);
  pinMode(PIR, INPUT);
  Serial.println("");
  clearMatrix();

  currentPalette = RainbowColors_p;
  currentBlending = LINEARBLEND;

  for (int i = 0; i <= 3; i++) {
    Cunicornio(i);
  }
  Serial.println(F("\nPregúntame cómo me siento"));
}
```

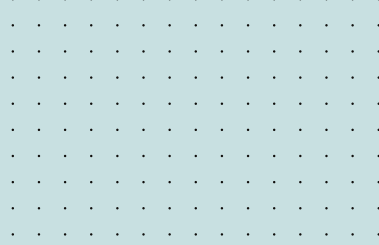


CODE

FINAL INTEGRATION

04

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
  curtime=millis();  
  String Mensaje;  
  int index=0;  
  String respo;  
  if (BTSerial.available()>0){  
    Mensaje=BTSerial.readStringUntil('\n');  
    Serial.print('\n');  
    Serial.println(Mensaje);  
    index=Mensaje.length();  
    Mensaje.remove(index);  
    delay(500);  
  }  
  
  if (Serial.available()>0){  
    Mensaje=Serial.readStringUntil('\n');  
    Mensaje.trim();  
    Serial.print('\n');  
    Serial.println(Mensaje);  
    BTSerial.println(Mensaje);  
    delay(500);  
  }  
  
  if (Mensaje.equals("Como te sientes?")){  
    respo=Random_emotion();  
    Serial.println(respo);  
    BTSerial.println(respo);  
    Activate_emotion(respo);  
    Serial.println(F("Y tú?"));  
    BTSerial.println(F("Y tú?"));  
    e=respo;  
    curtime=millis();  
    pastime=curtime;  
  }else if (Mensaje.equals("Escuchame")){  
    Serial.println("Modo platica conmigo activado");  
    for (int i = 0; i <= 10; i++) {  
      readPIRnMIC();  
      e="V";  
    }  
  }  
  
  }else if (Mensaje!=""){  
    Answer_emotion(Mensaje);  
    Serial.println(F("Deja te copio"));  
    BTSerial.println(F("Deja te copio"));  
    delay(300);  
    Activate_emotion(Mensaje);  
    e=Mensaje;  
    curtime=millis();  
    pastime=curtime;  
  }  
}
```



CODE

FINAL INTEGRATION

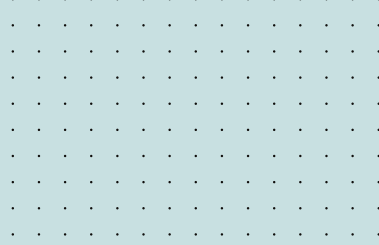
04

```
v
if (e.equals("V")){
  readPIRnMIC();
}else if (curtime-pastime>15000){
  //Serial.println(e);
  Activate_emotion(e);
  pastime=curtime;
}
}

void readPIRnMIC() {
  //Serial.println(analogRead(MIC));
  if(digitalRead(PIR) == HIGH && analogRead(MIC)>=30){
    RGB_color(0,255,255);
    delay(2000);
  }
  RGB_color(0,0,0);
}

void RGB_color(int red_light_value, int green_light_value, int blue_light_value){
  analogWrite(R, red_light_value);
  analogWrite(G, green_light_value);
  analogWrite(B, blue_light_value);
}

String Random_emotion(){
  String resp;
  long r=random(0,10);
  switch (r){
    case 0:
      resp="FELIZ";
      break;
    case 1:
      resp="ALEGRE";
      break;
    case 2:
      resp="ENOJADO";
      break;
    case 3:
      resp="TRISTE";
      break;
    case 4:
      resp="ESTRESADO";
      break;
    case 5:
      resp="DEPRIMIDO";
      break;
    case 6:
      resp="MUERTO";
      break;
  }
}
```



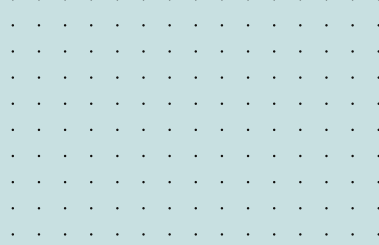
CODE

FINAL INTEGRATION

04

```
v case 7:
    resp="ENAMORADO";
    break;
case 8:
    resp="SORPRENDIDO";
    break;
case 9:
    resp="ASQUEADO";
    break;
case 10:
    resp="UNICORNIO";
    break;
default:
    resp="ERROR";
    break;
}
return resp;
}

void Activate_emotion(String resp){
    int index=resp.length();
    resp.remove(index-1);
    if (resp != ""){
        if (resp.equalsIgnoreCase( "ALEGR")){
            RGB_color(255, 255, 0); // Yellow
            Calegre();
        }else if (resp.equalsIgnoreCase("FELI")){
            RGB_color(255, 255, 0); // Yellow
            Cfeliz();
        }else if (resp.equalsIgnoreCase("ENOJAD")){
            RGB_color(255, 0, 0); // Red
            Cenojado();
        }else if (resp.equalsIgnoreCase("TRIST")){
            RGB_color(0, 0, 255); // Blue
            Ctriste();
        }else if (resp.equalsIgnoreCase("ESTRESAD")){
            RGB_color(255, 165, 0); // Orange
            Cestresado();
        }else if (resp.equalsIgnoreCase("DEPRIMID")){
            RGB_color(0, 0, 255); // Blue
            Cdeprimido();
        }else if (resp.equalsIgnoreCase("MUERT")){
            RGB_color(10, 0, 10); // Purple
            Cmuerto();
        }else if (resp.equalsIgnoreCase("ENAMORAD")){
            RGB_color(255, 105, 180); // Pink
            Cenamorado();
        }else if (resp.equalsIgnoreCase("SORPRENDID")){
            RGB_color(255, 0, 255); // Purple
            Csorprendido();
        }else if (resp.equalsIgnoreCase("ASQUEAD")){
            RGB_color(0, 255, 0); // Green
            Casqueado();
        }
    }
}
```



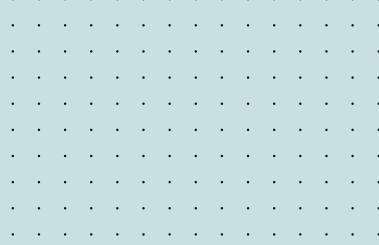
CODE

FINAL INTEGRATION

04

```
}else if (resp.equalsIgnoreCase("UNICORNI")){
    RGB_color(255, 255, 255); // Green
    for (int i = 0; i <= 3; i++) {
        Cunicornio(i);
    }
}else if (resp.equalsIgnoreCase("CLEA")){
    RGB_color(0, 0, 0); // Green
    clearMatrix();
}else{
    Serial.println(F("ERROR: Eso no es un argumento válido"));
    BTSerial.println(F("ERROR: Eso no es un argumento válido"));
}
}
}
```

```
void Answer_emotion(String resp){
    String answer[4];
    if (resp != ""){
        int index=resp.length();
        resp.remove(index-1);
        int nummax;
        long r;
        int s=1;
        if (resp.equalsIgnoreCase( "ALEGR")){
            s=1;
            nummax=4;
        }else if (resp.equalsIgnoreCase("FELI")){
            s=5;
            nummax=8;
        }else if (resp.equalsIgnoreCase("ENOJAD")){
            s=9;
            nummax=12;
        }else if (resp.equalsIgnoreCase("TRIST")){
            s=13;
            nummax=16;
        }else if (resp.equalsIgnoreCase("ESTRESAD")){
            s=17;
            nummax=20;
        }else if (resp.equalsIgnoreCase("DEPRIMID")){
            s=21;
            nummax=24;
        }else if (resp.equalsIgnoreCase("MUERT")){
            s=25;
            nummax=28;
        }else if (resp.equalsIgnoreCase("ENAMORAD")){
            s=29;
            nummax=32;
        }else if (resp.equalsIgnoreCase("SORPRENDID")){
            s=33;
            nummax=36;
        }else if (resp.equalsIgnoreCase("ASQUEAD")){
            s=37;
```

CODE

FINAL INTEGRATION

04

```
    nummax=40;
}else if (resp.equalsIgnoreCase("UNICORNI")){
    s=41;
    nummax=44;
}else {
    nummax=0;
    s=0;
}
r=random(s,nummax);

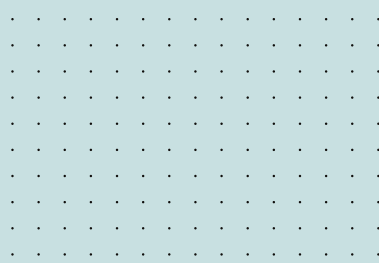
strcpy_P(buffer, (char *)pgm_read_word(&(answer_table[r]]));

Serial.println(buffer);
BTSerial.println(buffer);
}
}

void Cunicornio(int ru){
    int r=0, g=0, b=0;
    if (ru==0||ru>2){
        r=255;
        g=0;
        b=0;
    }else if (ru==1){
        r=0;
        g=255;
        b=0;
    }else if (ru==2){
        r=0;
        g=0;
        b=255;
    }
}

for (int i = 0; i <= 255; i++) {
    int colorIndex=i+1;
    if (ru==0||ru>2){
        r-=1;
        g+=1;
    }else if (ru==1){
        g-=1;
        b+=1;
    }else if (ru==2){
        b-=1;
        r+=1;
    }
}

for( int i = 0; i < 64; ++i) {
    leds[i] = ColorFromPalette( RainbowColors_p, colorIndex, 255, currentBlending);
    colorIndex += 3;
}
//Serial.println(r);
RGB_color( r, g, b);
FastLED.show();
FastLED.delay(1000/500);
}
```

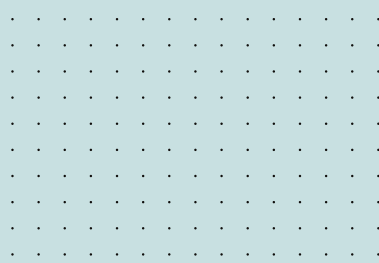


CODE -ANIMACIONES

```
#include "FastLED.h"
#define NUM_LEDS 64
CRGB leds[NUM_LEDS];

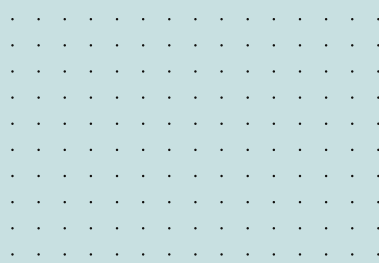
void clearMatrix(){
  for(int dot = 0; dot < NUM_LEDS; dot++) {
    leds[dot] = CRGB::Black;
  }
  FastLED.show();
}

void Calegre(){
  for (int i = 0; i <= 2; i++) {
    leds[0] = CRGB(0, 0, 0);
    leds[1] = CRGB(0, 0, 0);
    leds[2] = CRGB(0, 0, 0);
    leds[3] = CRGB(0, 0, 0);
    leds[4] = CRGB(0, 0, 0);
    leds[5] = CRGB(0, 0, 0);
    leds[6] = CRGB(0, 0, 0);
    leds[7] = CRGB(0, 0, 0);
    leds[8] = CRGB(0, 0, 0);
    leds[9] = CRGB(0, 0, 0);
    leds[10] = CRGB(255, 255, 0);
    leds[11] = CRGB(0, 0, 0);
    leds[12] = CRGB(0, 0, 0);
    leds[13] = CRGB(255, 255, 0);
    leds[14] = CRGB(0, 0, 0);
    leds[15] = CRGB(0, 0, 0);
    leds[16] = CRGB(0, 0, 0);
    leds[17] = CRGB(0, 0, 0);
    leds[18] = CRGB(255, 255, 0);
    leds[19] = CRGB(0, 0, 0);
    leds[20] = CRGB(0, 0, 0);
    leds[21] = CRGB(255, 255, 0);
    leds[22] = CRGB(0, 0, 0);
    leds[23] = CRGB(0, 0, 0);
    leds[24] = CRGB(0, 0, 0);
    leds[25] = CRGB(0, 0, 0);
    leds[26] = CRGB(255, 255, 0);
    leds[27] = CRGB(0, 0, 0);
    leds[28] = CRGB(0, 0, 0);
    leds[29] = CRGB(255, 255, 0);
    leds[30] = CRGB(0, 0, 0);
    leds[31] = CRGB(0, 0, 0);
    leds[32] = CRGB(0, 0, 0);
    leds[33] = CRGB(0, 0, 0);
    leds[34] = CRGB(0, 0, 0);
    leds[35] = CRGB(0, 0, 0);
    leds[36] = CRGB(0, 0, 0);
    leds[37] = CRGB(0, 0, 0);
    leds[38] = CRGB(0, 0, 0);
    leds[39] = CRGB(0, 0, 0);
    leds[40] = CRGB(0, 0, 0);
```



CODE -ANIMACIONES

```
leds[41] = CRGB(255, 255, 0);
leds[42] = CRGB(255, 255, 0);
leds[43] = CRGB(255, 255, 0);
leds[44] = CRGB(255, 255, 0);
leds[45] = CRGB(255, 255, 0);
leds[46] = CRGB(255, 255, 0);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[49] = CRGB(255, 255, 0);
leds[50] = CRGB(0, 0, 0);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
leds[54] = CRGB(255, 255, 0);
leds[55] = CRGB(0, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(255, 255, 0);
leds[59] = CRGB(255, 255, 0);
leds[60] = CRGB(255, 255, 0);
leds[61] = CRGB(255, 255, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(166);
leds[26] = CRGB(0, 0, 0);
leds[29] = CRGB(0, 0, 0);
leds[50] = CRGB(255, 255, 0);
leds[51] = CRGB(255, 255, 0);
leds[52] = CRGB(255, 255, 0);
leds[53] = CRGB(255, 255, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 0, 0);
FastLED.show();
delay(166);
leds[32] = CRGB(255, 255, 0);
leds[39] = CRGB(255, 255, 0);
leds[49] = CRGB(0, 0, 0);
leds[50] = CRGB(0, 0, 0);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
leds[54] = CRGB(0, 0, 0);
FastLED.show();
delay(350);
leds[32] = CRGB(0, 0, 0);
leds[39] = CRGB(0, 0, 0);
leds[49] = CRGB(255, 255, 0);
leds[50] = CRGB(255, 255, 0);
leds[51] = CRGB(255, 255, 0);
leds[52] = CRGB(255, 255, 0);
leds[53] = CRGB(255, 255, 0);
leds[54] = CRGB(255, 255, 0);
```

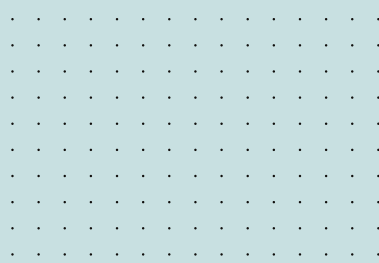


CODE -ANIMACIONES

```
FastLED.show();
delay(166);
leds[26] = CRGB(255, 255, 0);
leds[29] = CRGB(255, 255, 0);
leds[50] = CRGB(0, 0, 0);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
leds[58] = CRGB(255, 255, 0);
leds[59] = CRGB(255, 255, 0);
leds[60] = CRGB(255, 255, 0);
leds[61] = CRGB(255, 255, 0);
FastLED.show();
delay(320);
}

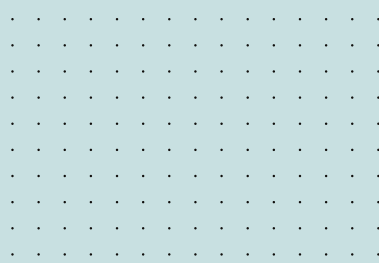
}
```

```
void Cfeliz(){
for (int i = 0; i <= 2; i++) {
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(0, 0, 0);
leds[2] = CRGB(0, 0, 0);
leds[3] = CRGB(0, 0, 0);
leds[4] = CRGB(0, 0, 0);
leds[5] = CRGB(0, 0, 0);
leds[6] = CRGB(0, 0, 0);
leds[7] = CRGB(0, 0, 0);
leds[8] = CRGB(0, 0, 0);
leds[9] = CRGB(0, 0, 0);
leds[10] = CRGB(255, 255, 0);
leds[11] = CRGB(0, 0, 0);
leds[12] = CRGB(0, 0, 0);
leds[13] = CRGB(255, 255, 0);
leds[14] = CRGB(0, 0, 0);
leds[15] = CRGB(0, 0, 0);
leds[16] = CRGB(0, 0, 0);
leds[17] = CRGB(0, 0, 0);
leds[18] = CRGB(255, 255, 0);
leds[19] = CRGB(0, 0, 0);
leds[20] = CRGB(0, 0, 0);
leds[21] = CRGB(255, 255, 0);
leds[22] = CRGB(0, 0, 0);
leds[23] = CRGB(0, 0, 0);
leds[24] = CRGB(0, 0, 0);
leds[25] = CRGB(0, 0, 0);
leds[26] = CRGB(0, 0, 0);
leds[27] = CRGB(0, 0, 0);
leds[28] = CRGB(0, 0, 0);
leds[29] = CRGB(0, 0, 0);
leds[30] = CRGB(0, 0, 0);
}
```



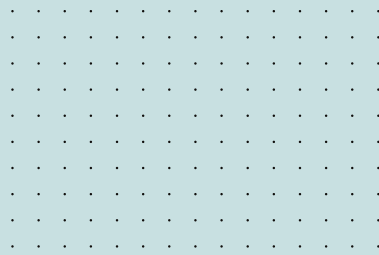
CODE -ANIMACIONES

```
    leds[31] = CRGB(0, 0, 0);
    leds[32] = CRGB(255, 255, 0);
    leds[33] = CRGB(0, 0, 0);
    leds[34] = CRGB(0, 0, 0);
    leds[35] = CRGB(0, 0, 0);
    leds[36] = CRGB(0, 0, 0);
    leds[37] = CRGB(0, 0, 0);
    leds[38] = CRGB(0, 0, 0);
    leds[39] = CRGB(255, 255, 0);
    leds[40] = CRGB(0, 0, 0);
    leds[41] = CRGB(255, 255, 0);
    leds[42] = CRGB(255, 255, 0);
    leds[43] = CRGB(255, 255, 0);
    leds[44] = CRGB(255, 255, 0);
    leds[45] = CRGB(255, 255, 0);
    leds[46] = CRGB(255, 255, 0);
    leds[47] = CRGB(0, 0, 0);
    leds[48] = CRGB(0, 0, 0);
    leds[49] = CRGB(0, 0, 0);
    leds[50] = CRGB(0, 0, 0);
    leds[51] = CRGB(0, 0, 0);
    leds[52] = CRGB(0, 0, 0);
    leds[53] = CRGB(0, 0, 0);
    leds[54] = CRGB(0, 0, 0);
    leds[55] = CRGB(0, 0, 0);
    leds[56] = CRGB(0, 0, 0);
    leds[57] = CRGB(0, 0, 0);
    leds[58] = CRGB(0, 0, 0);
    leds[59] = CRGB(0, 0, 0);
    leds[60] = CRGB(0, 0, 0);
    leds[61] = CRGB(0, 0, 0);
    leds[62] = CRGB(0, 0, 0);
    leds[63] = CRGB(0, 0, 0);
    FastLED.show();
    delay(333);
    leds[32] = CRGB(0, 0, 0);
    leds[39] = CRGB(0, 0, 0);
    FastLED.show();
    delay(333);
    leds[10] = CRGB(0, 0, 0);
    leds[13] = CRGB(0, 0, 0);
    leds[41] = CRGB(0, 0, 0);
    leds[46] = CRGB(0, 0, 0);
    FastLED.show();
    delay(333);
    leds[10] = CRGB(255, 255, 0);
    leds[13] = CRGB(255, 255, 0);
    leds[41] = CRGB(255, 255, 0);
    leds[46] = CRGB(255, 255, 0);
    FastLED.show();
    delay(333);
    leds[32] = CRGB(255, 255, 0);
    leds[39] = CRGB(255, 255, 0);
    FastLED.show();
    delay(333);
}
```



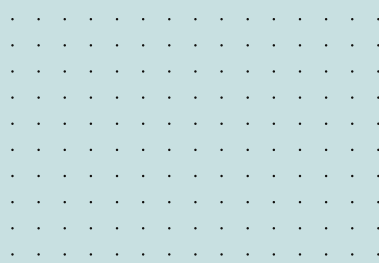
CODE -ANIMACIONES

```
void Cenojado(){
  for (int i = 0; i <= 1; i++) {
    leds[0] = CRGB(255, 0, 0);
    leds[1] = CRGB(255, 0, 0);
    leds[2] = CRGB(0, 0, 0);
    leds[3] = CRGB(0, 0, 0);
    leds[4] = CRGB(0, 0, 0);
    leds[5] = CRGB(0, 0, 0);
    leds[6] = CRGB(255, 0, 0);
    leds[7] = CRGB(255, 0, 0);
    leds[8] = CRGB(0, 0, 0);
    leds[9] = CRGB(255, 0, 0);
    leds[10] = CRGB(255, 0, 0);
    leds[11] = CRGB(0, 0, 0);
    leds[12] = CRGB(0, 0, 0);
    leds[13] = CRGB(255, 0, 0);
    leds[14] = CRGB(255, 0, 0);
    leds[15] = CRGB(0, 0, 0);
    leds[16] = CRGB(0, 0, 0);
    leds[17] = CRGB(0, 0, 0);
    leds[18] = CRGB(255, 0, 0);
    leds[19] = CRGB(0, 0, 0);
    leds[20] = CRGB(0, 0, 0);
    leds[21] = CRGB(255, 0, 0);
    leds[22] = CRGB(0, 0, 0);
    leds[23] = CRGB(0, 0, 0);
    leds[24] = CRGB(0, 0, 0);
    leds[25] = CRGB(0, 0, 0);
    leds[26] = CRGB(0, 0, 0);
    leds[27] = CRGB(0, 0, 0);
    leds[28] = CRGB(0, 0, 0);
    leds[29] = CRGB(0, 0, 0);
    leds[30] = CRGB(0, 0, 0);
    leds[31] = CRGB(0, 0, 0);
    leds[32] = CRGB(0, 0, 0);
    leds[33] = CRGB(0, 0, 0);
    leds[34] = CRGB(0, 0, 0);
    leds[35] = CRGB(0, 0, 0);
    leds[36] = CRGB(0, 0, 0);
    leds[37] = CRGB(0, 0, 0);
    leds[38] = CRGB(0, 0, 0);
    leds[39] = CRGB(0, 0, 0);
    leds[40] = CRGB(0, 0, 0);
    leds[41] = CRGB(255, 0, 0);
    leds[42] = CRGB(255, 0, 0);
    leds[43] = CRGB(255, 0, 0);
    leds[44] = CRGB(255, 0, 0);
    leds[45] = CRGB(255, 0, 0);
    leds[46] = CRGB(255, 0, 0);
    leds[47] = CRGB(0, 0, 0);
    leds[48] = CRGB(255, 0, 0);
    leds[49] = CRGB(0, 0, 0);
    leds[50] = CRGB(0, 0, 0);
  }
}
```



CODE -ANIMACIONES

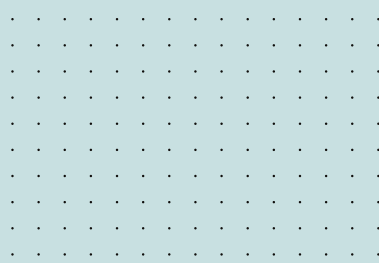
```
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
leds[54] = CRGB(0, 0, 0);
leds[55] = CRGB(255, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 0, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[2] = CRGB(255, 0, 0);
leds[5] = CRGB(255, 0, 0);
leds[9] = CRGB(0, 0, 0);
leds[14] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[10] = CRGB(0, 0, 0);
leds[13] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[55] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(0, 0, 0);
leds[2] = CRGB(0, 0, 0);
leds[5] = CRGB(0, 0, 0);
leds[6] = CRGB(0, 0, 0);
leds[7] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 0);
leds[46] = CRGB(0, 0, 0);
FastLED.show();
delay(100);
leds[10] = CRGB(255, 0, 0);
leds[13] = CRGB(255, 0, 0);
FastLED.show();
delay(600);
leds[10] = CRGB(0, 0, 0);
leds[13] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[0] = CRGB(255, 0, 0);
leds[1] = CRGB(255, 0, 0);
leds[2] = CRGB(255, 0, 0);
leds[5] = CRGB(255, 0, 0);
leds[6] = CRGB(255, 0, 0);
leds[7] = CRGB(255, 0, 0);
leds[41] = CRGB(255, 0, 0);
leds[46] = CRGB(255, 0, 0);
FastLED.show();
delay(333);
```



CODE -ANIMACIONES

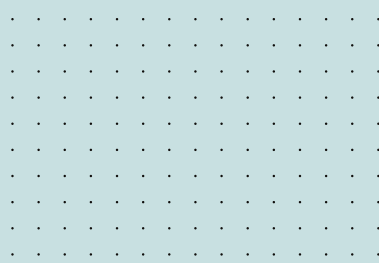
```
    leds[10] = CRGB(255, 0, 0);
    leds[13] = CRGB(255, 0, 0);
    leds[48] = CRGB(255, 0, 0);
    leds[55] = CRGB(255, 0, 0);
    FastLED.show();
    delay(333);
    leds[2] = CRGB(0, 0, 0);
    leds[5] = CRGB(0, 0, 0);
    leds[9] = CRGB(255, 0, 0);
    leds[14] = CRGB(255, 0, 0);
    FastLED.show();
    delay(333);
    leds[2] = CRGB(255, 0, 0);
    leds[5] = CRGB(255, 0, 0);
    leds[9] = CRGB(0, 0, 0);
    leds[14] = CRGB(0, 0, 0);
    FastLED.show();
    delay(333);
    leds[2] = CRGB(0, 0, 0);
    leds[5] = CRGB(0, 0, 0);
    leds[9] = CRGB(255, 0, 0);
    leds[14] = CRGB(255, 0, 0);
    FastLED.show();
    delay(333);
  }
}
```

```
void Ctriste(){
  for (int i = 0; i <= 2; i++) {
    leds[0] = CRGB(0, 0, 0);
    leds[1] = CRGB(0, 0, 0);
    leds[2] = CRGB(0, 0, 0);
    leds[3] = CRGB(0, 0, 0);
    leds[4] = CRGB(0, 0, 0);
    leds[5] = CRGB(0, 0, 0);
    leds[6] = CRGB(0, 0, 0);
    leds[7] = CRGB(0, 0, 0);
    leds[8] = CRGB(0, 0, 0);
    leds[9] = CRGB(0, 0, 0);
    leds[10] = CRGB(0, 0, 255);
    leds[11] = CRGB(0, 0, 0);
    leds[12] = CRGB(0, 0, 0);
    leds[13] = CRGB(0, 0, 255);
    leds[14] = CRGB(0, 0, 0);
    leds[15] = CRGB(0, 0, 0);
    leds[16] = CRGB(0, 0, 0);
    leds[17] = CRGB(0, 0, 0);
    leds[18] = CRGB(0, 0, 255);
    leds[19] = CRGB(0, 0, 0);
    leds[20] = CRGB(0, 0, 0);
    leds[21] = CRGB(0, 0, 255);
    leds[22] = CRGB(0, 0, 0);
    leds[23] = CRGB(0, 0, 0);
    leds[24] = CRGB(0, 0, 0);
    leds[25] = CRGB(0, 0, 0);
  }
}
```

CODE -ANIMACIONES

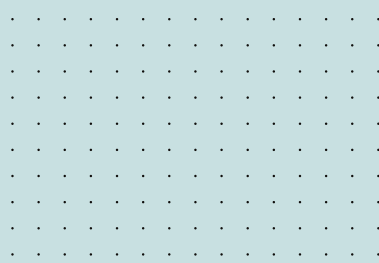
```
leds[26] = CRGB(0, 0, 0);
leds[27] = CRGB(0, 0, 0);
leds[28] = CRGB(0, 0, 0);
leds[29] = CRGB(0, 0, 0);
leds[30] = CRGB(0, 0, 0);
leds[31] = CRGB(0, 0, 0);
leds[32] = CRGB(0, 0, 0);
leds[33] = CRGB(0, 0, 0);
leds[34] = CRGB(0, 0, 0);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[37] = CRGB(0, 0, 0);
leds[38] = CRGB(0, 0, 0);
leds[39] = CRGB(0, 0, 0);
leds[40] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 255);
leds[42] = CRGB(0, 0, 255);
leds[43] = CRGB(0, 0, 255);
leds[44] = CRGB(0, 0, 255);
leds[45] = CRGB(0, 0, 255);
leds[46] = CRGB(0, 0, 255);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 255);
leds[49] = CRGB(0, 0, 0);
leds[50] = CRGB(0, 0, 0);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
leds[54] = CRGB(0, 0, 0);
leds[55] = CRGB(0, 0, 255);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 0, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[48] = CRGB(0, 0, 0);
leds[55] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[18] = CRGB(0, 0, 0);
leds[21] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 0);
leds[46] = CRGB(0, 0, 0);
FastLED.show();
delay(333);
leds[18] = CRGB(0, 0, 255);
leds[21] = CRGB(0, 0, 255);
leds[41] = CRGB(0, 0, 255);
leds[46] = CRGB(0, 0, 255);
```



CODE -ANIMACIONES

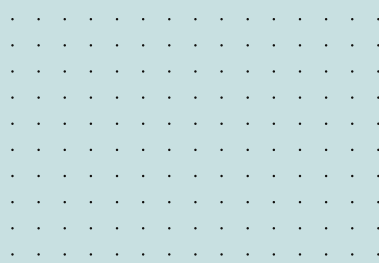
```
FastLED.show();
delay(333);
leds[48] = CRGB(0, 0, 255);
leds[55] = CRGB(0, 0, 255);
FastLED.show();
delay(333);
}
}

void Cestresado(){
for (int i = 0; i <= 3; i++) {
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(0, 0, 0);
leds[2] = CRGB(0, 0, 0);
leds[3] = CRGB(0, 0, 0);
leds[4] = CRGB(0, 0, 0);
leds[5] = CRGB(0, 0, 0);
leds[6] = CRGB(0, 0, 0);
leds[7] = CRGB(0, 0, 0);
leds[8] = CRGB(255, 140, 0);
leds[9] = CRGB(0, 0, 0);
leds[10] = CRGB(255, 140, 0);
leds[11] = CRGB(0, 0, 0);
leds[12] = CRGB(0, 0, 0);
leds[13] = CRGB(255, 140, 0);
leds[14] = CRGB(0, 0, 0);
leds[15] = CRGB(255, 140, 0);
leds[16] = CRGB(0, 0, 0);
leds[17] = CRGB(255, 140, 0);
leds[18] = CRGB(0, 0, 0);
leds[19] = CRGB(0, 0, 0);
leds[20] = CRGB(0, 0, 0);
leds[21] = CRGB(0, 0, 0);
leds[22] = CRGB(255, 140, 0);
leds[23] = CRGB(0, 0, 0);
leds[24] = CRGB(255, 140, 0);
leds[25] = CRGB(0, 0, 0);
leds[26] = CRGB(255, 140, 0);
leds[27] = CRGB(0, 0, 0);
leds[28] = CRGB(0, 0, 0);
leds[29] = CRGB(255, 140, 0);
leds[30] = CRGB(0, 0, 0);
leds[31] = CRGB(255, 140, 0);
leds[32] = CRGB(0, 0, 0);
leds[33] = CRGB(0, 0, 0);
leds[34] = CRGB(0, 0, 0);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[37] = CRGB(0, 0, 0);
leds[38] = CRGB(0, 0, 0);
leds[39] = CRGB(0, 0, 0);
leds[40] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 0);
leds[42] = CRGB(0, 0, 0);
leds[43] = CRGB(255, 140, 0);
leds[44] = CRGB(255, 140, 0);
leds[45] = CRGB(0, 0, 0);
```



CODE -ANIMACIONES

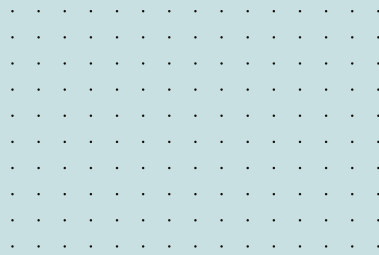
```
leds[46] = CRGB(0, 0, 0);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[49] = CRGB(255, 140, 0);
leds[50] = CRGB(255, 140, 0);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(255, 140, 0);
leds[54] = CRGB(255, 140, 0);
leds[55] = CRGB(0, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 0, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(125);
leds[43] = CRGB(0, 0, 0);
leds[44] = CRGB(0, 0, 0);
leds[51] = CRGB(255, 140, 0);
leds[52] = CRGB(255, 140, 0);
FastLED.show();
delay(125);
leds[42] = CRGB(255, 140, 0);
leds[50] = CRGB(0, 0, 0);
FastLED.show();
delay(125);
leds[44] = CRGB(255, 140, 0);
leds[52] = CRGB(0, 0, 0);
FastLED.show();
delay(125);
leds[42] = CRGB(0, 0, 0);
leds[50] = CRGB(255, 140, 0);
leds[46] = CRGB(255, 140, 0);
leds[54] = CRGB(0, 0, 0);
FastLED.show();
delay(125);
leds[44] = CRGB(0, 0, 0);
leds[52] = CRGB(255, 140, 0);
FastLED.show();
delay(125);
leds[46] = CRGB(0, 0, 0);
leds[54] = CRGB(255, 140, 0);
FastLED.show();
delay(125);
leds[41] = CRGB(255, 140, 0);
leds[43] = CRGB(255, 140, 0);
leds[45] = CRGB(255, 140, 0);
leds[49] = CRGB(0, 0, 0);
leds[51] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
```



CODE -ANIMACIONES

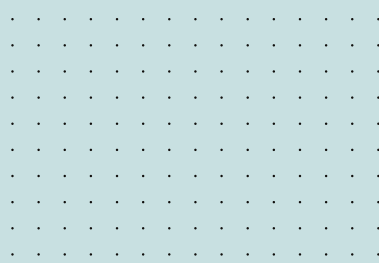
```
FastLED.show();
delay(125);
leds[41] = CRGB(0, 0, 0);
leds[43] = CRGB(0, 0, 0);
leds[45] = CRGB(0, 0, 0);
leds[49] = CRGB(255, 140, 0);
leds[51] = CRGB(255, 140, 0);
leds[53] = CRGB(255, 140, 0);
leds[42] = CRGB(255, 140, 0);
leds[44] = CRGB(255, 140, 0);
leds[46] = CRGB(255, 140, 0);
leds[50] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[54] = CRGB(0, 0, 0);
FastLED.show();
delay(125);
leds[42] = CRGB(0, 0, 0);
leds[44] = CRGB(0, 0, 0);
leds[46] = CRGB(0, 0, 0);
leds[50] = CRGB(255, 140, 0);
leds[52] = CRGB(255, 140, 0);
leds[54] = CRGB(255, 140, 0);
FastLED.show();
delay(125);
leds[43] = CRGB(255, 140, 0);
leds[51] = CRGB(0, 0, 0);
leds[44] = CRGB(255, 140, 0);
leds[52] = CRGB(0, 0, 0);
FastLED.show();
delay(300);
}
}
```

```
void Cdeprimido(){
for (int i = 0; i <= 3; i++) {
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(0, 0, 0);
leds[2] = CRGB(0, 0, 0);
leds[3] = CRGB(0, 0, 0);
leds[4] = CRGB(0, 0, 0);
leds[5] = CRGB(0, 0, 0);
leds[6] = CRGB(0, 0, 0);
leds[7] = CRGB(0, 0, 0);
leds[8] = CRGB(0, 0, 0);
leds[9] = CRGB(0, 0, 0);
leds[10] = CRGB(0, 0, 0);
leds[11] = CRGB(0, 0, 0);
leds[12] = CRGB(0, 0, 0);
leds[13] = CRGB(0, 0, 0);
leds[14] = CRGB(0, 0, 0);
leds[15] = CRGB(0, 0, 0);
leds[16] = CRGB(0, 0, 255);
leds[17] = CRGB(0, 0, 255);
leds[18] = CRGB(0, 0, 255);
leds[19] = CRGB(0, 0, 0);
leds[20] = CRGB(0, 0, 0);
}
```



CODE -ANIMACIONES

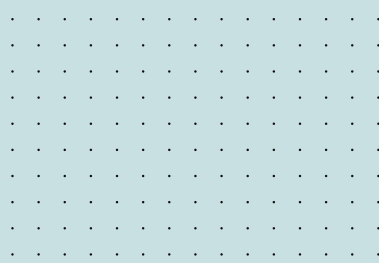
```
leds[21] = CRGB(0, 0, 255);
leds[22] = CRGB(0, 0, 255);
leds[23] = CRGB(0, 0, 255);
leds[24] = CRGB(0, 0, 0);
leds[25] = CRGB(0, 0, 255);
leds[26] = CRGB(0, 0, 0);
leds[27] = CRGB(0, 0, 0);
leds[28] = CRGB(0, 0, 0);
leds[29] = CRGB(0, 0, 0);
leds[30] = CRGB(0, 0, 255);
leds[31] = CRGB(0, 0, 0);
leds[32] = CRGB(0, 0, 0);
leds[33] = CRGB(0, 0, 255);
leds[34] = CRGB(0, 0, 0);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[37] = CRGB(0, 0, 0);
leds[38] = CRGB(0, 0, 255);
leds[39] = CRGB(0, 0, 0);
leds[40] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 0);
leds[42] = CRGB(0, 0, 0);
leds[43] = CRGB(0, 0, 255);
leds[44] = CRGB(0, 0, 255);
leds[45] = CRGB(0, 0, 0);
leds[46] = CRGB(0, 0, 0);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[49] = CRGB(0, 0, 0);
leds[50] = CRGB(0, 0, 255);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 255);
leds[54] = CRGB(0, 0, 0);
leds[55] = CRGB(0, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 0, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(200);
leds[25] = CRGB(0, 0, 0);
leds[30] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 255);
leds[46] = CRGB(0, 0, 255);
leds[58] = CRGB(0, 0, 255);
leds[61] = CRGB(0, 0, 255);
FastLED.show();
delay(200);
```



CODE -ANIMACIONES

```
    leds[33] = CRGB(0, 0, 0);
    leds[38] = CRGB(0, 0, 0);
    FastLED.show();
    delay(200);
    leds[41] = CRGB(0, 0, 0);
    leds[46] = CRGB(0, 0, 0);
    leds[48] = CRGB(0, 0, 255);
    leds[55] = CRGB(0, 0, 255);
    FastLED.show();
    delay(200);
    leds[25] = CRGB(0, 0, 255);
    leds[30] = CRGB(0, 0, 255);
    leds[48] = CRGB(0, 0, 0);
    leds[55] = CRGB(0, 0, 0);
    leds[56] = CRGB(0, 0, 255);
    leds[63] = CRGB(0, 0, 255);
    FastLED.show();
    delay(200);
    leds[33] = CRGB(0, 0, 255);
    leds[38] = CRGB(0, 0, 255);
    leds[56] = CRGB(0, 0, 0);
    leds[63] = CRGB(0, 0, 0);
    leds[58] = CRGB(0, 0, 0);
    leds[61] = CRGB(0, 0, 0);
    FastLED.show();
  }
}
```

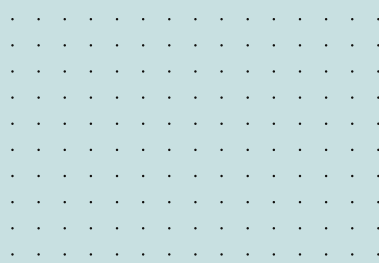
```
void Cmuerto(){
  leds[0] = CRGB(0, 0, 0);
  leds[1] = CRGB(0, 0, 0);
  leds[2] = CRGB(0, 0, 0);
  leds[3] = CRGB(0, 0, 0);
  leds[4] = CRGB(0, 0, 0);
  leds[5] = CRGB(0, 0, 0);
  leds[6] = CRGB(0, 0, 0);
  leds[7] = CRGB(0, 0, 0);
  leds[8] = CRGB(0, 0, 0);
  leds[9] = CRGB(0, 0, 0);
  leds[10] = CRGB(0, 196, 255);
  leds[11] = CRGB(0, 0, 0);
  leds[12] = CRGB(0, 0, 0);
  leds[13] = CRGB(0, 0, 0);
  leds[14] = CRGB(0, 196, 255);
  leds[15] = CRGB(0, 0, 0);
  leds[16] = CRGB(0, 0, 0);
  leds[17] = CRGB(0, 0, 0);
  leds[18] = CRGB(0, 196, 255);
  leds[19] = CRGB(0, 0, 0);
  leds[20] = CRGB(0, 0, 0);
  leds[21] = CRGB(0, 196, 255);
  leds[22] = CRGB(0, 0, 0);
  leds[23] = CRGB(0, 0, 0);
  leds[24] = CRGB(0, 0, 0);
  leds[25] = CRGB(0, 0, 0);
}
```



CODE -ANIMACIONES

```
    leds[26] = CRGB(0, 0, 0);
    leds[27] = CRGB(0, 0, 0);
    leds[28] = CRGB(0, 0, 0);
    leds[29] = CRGB(0, 0, 0);
    leds[30] = CRGB(0, 0, 0);
    leds[31] = CRGB(0, 0, 0);
    leds[32] = CRGB(0, 0, 0);
    leds[33] = CRGB(0, 0, 0);
    leds[34] = CRGB(0, 0, 0);
    leds[35] = CRGB(0, 0, 0);
    leds[36] = CRGB(0, 0, 0);
    leds[37] = CRGB(0, 0, 0);
    leds[38] = CRGB(0, 0, 0);
    leds[39] = CRGB(0, 0, 0);
    leds[40] = CRGB(0, 196, 255);
    leds[41] = CRGB(0, 0, 0);
    leds[42] = CRGB(0, 0, 0);
    leds[43] = CRGB(0, 0, 0);
    leds[44] = CRGB(0, 0, 0);
    leds[45] = CRGB(0, 0, 0);
    leds[46] = CRGB(0, 0, 0);
    leds[47] = CRGB(0, 196, 255);
    leds[48] = CRGB(0, 0, 0);
    leds[49] = CRGB(0, 196, 255);
    leds[50] = CRGB(0, 196, 255);
    leds[51] = CRGB(0, 196, 255);
    leds[52] = CRGB(0, 196, 255);
    leds[53] = CRGB(0, 196, 255);
    leds[54] = CRGB(0, 196, 255);
    leds[55] = CRGB(0, 0, 0);
    leds[56] = CRGB(0, 0, 0);
    leds[57] = CRGB(0, 0, 0);
    leds[58] = CRGB(0, 0, 0);
    leds[59] = CRGB(0, 0, 0);
    leds[60] = CRGB(0, 0, 0);
    leds[61] = CRGB(0, 0, 0);
    leds[62] = CRGB(0, 0, 0);
    leds[63] = CRGB(0, 0, 0);
    FastLED.show();
    delay(500);
    for(int dot = 63; dot > 0; dot--) {
        leds[dot] = CRGB(0, 255, 0);
        if (dot%8==0){
            FastLED.show();
            delay(100);
        }
    }

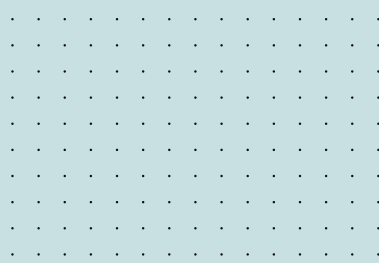
    leds[0] = CRGB(0, 255, 0);
    FastLED.show();
    delay(500);
```



CODE -ANIMACIONES

```
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(220, 0, 211);
leds[2] = CRGB(220, 0, 211);
leds[3] = CRGB(220, 0, 211);
leds[4] = CRGB(220, 0, 211);
leds[5] = CRGB(220, 0, 211);
leds[6] = CRGB(220, 0, 211);
leds[7] = CRGB(0, 0, 0);
FastLED.show();
delay(100);
leds[8] = CRGB(220, 0, 211);
leds[9] = CRGB(220, 0, 211);
leds[10] = CRGB(220, 0, 211);
leds[11] = CRGB(220, 0, 211);
leds[12] = CRGB(220, 0, 211);
leds[13] = CRGB(220, 0, 211);
leds[14] = CRGB(220, 0, 211);
leds[15] = CRGB(220, 0, 211);
FastLED.show();
delay(100);
leds[16] = CRGB(220, 0, 211);
leds[17] = CRGB(0, 0, 0);
leds[18] = CRGB(0, 0, 0);
leds[19] = CRGB(220, 0, 211);
leds[20] = CRGB(220, 0, 211);
leds[21] = CRGB(0, 0, 0);
leds[22] = CRGB(0, 0, 0);
leds[23] = CRGB(220, 0, 211);
FastLED.show();
delay(100);
leds[24] = CRGB(220, 0, 211);
leds[25] = CRGB(0, 0, 0);
leds[26] = CRGB(0, 0, 0);
leds[27] = CRGB(220, 0, 211);
leds[28] = CRGB(220, 0, 211);
leds[29] = CRGB(0, 0, 0);
leds[30] = CRGB(0, 0, 0);
leds[31] = CRGB(220, 0, 211);
FastLED.show();
delay(100);
leds[32] = CRGB(220, 0, 211);
leds[33] = CRGB(220, 0, 211);
leds[34] = CRGB(220, 0, 211);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[37] = CRGB(220, 0, 211);
leds[38] = CRGB(220, 0, 211);
leds[39] = CRGB(220, 0, 211);
FastLED.show();

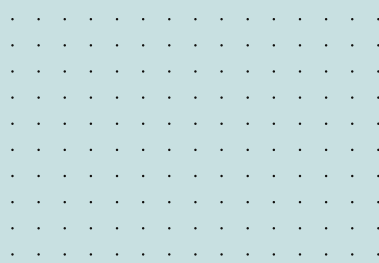
leds[40] = CRGB(220, 0, 211);
leds[41] = CRGB(220, 0, 211);
leds[42] = CRGB(220, 0, 211);
leds[43] = CRGB(220, 0, 211);
leds[44] = CRGB(220, 0, 211);
leds[45] = CRGB(220, 0, 211);
```

CODE -ANIMACIONES

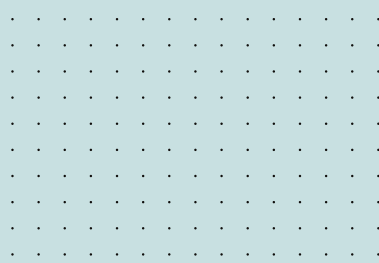
```
    leds[46] = CRGB(220, 0, 211);
    leds[47] = CRGB(220, 0, 211);
    FastLED.show();
    delay(100);
    leds[48] = CRGB(0, 0, 0);
    leds[49] = CRGB(220, 0, 211);
    leds[50] = CRGB(0, 0, 0);
    leds[51] = CRGB(220, 0, 211);
    leds[52] = CRGB(220, 0, 211);
    leds[53] = CRGB(0, 0, 0);
    leds[54] = CRGB(220, 0, 211);
    leds[55] = CRGB(0, 0, 0);
    FastLED.show();
    delay(100);
    leds[56] = CRGB(0, 0, 0);
    leds[57] = CRGB(220, 0, 211);
    leds[58] = CRGB(0, 0, 0);
    leds[59] = CRGB(220, 0, 211);
    leds[60] = CRGB(220, 0, 211);
    leds[61] = CRGB(0, 0, 0);
    leds[62] = CRGB(220, 0, 211);
    leds[63] = CRGB(0, 0, 0);
    FastLED.show();
}
```

```
void Cenamorado(){
  clearMatrix();
  leds[0] = CRGB(255, 0, 0);
  leds[2] = CRGB(255, 0, 0);
  leds[9] = CRGB(255, 0, 0);
  FastLED.show();
  delay(300);
  leds[0] = CRGB(0, 0, 0);
  leds[2] = CRGB(0, 0, 0);
  leds[8] = CRGB(255, 0, 0);
  leds[9] = CRGB(255, 0, 0);
  leds[11] = CRGB(255, 0, 0);
  leds[12] = CRGB(255, 0, 0);
  leds[16] = CRGB(255, 0, 0);
  leds[17] = CRGB(255, 0, 0);
  leds[18] = CRGB(255, 0, 0);
  leds[19] = CRGB(255, 0, 0);
  leds[20] = CRGB(255, 0, 0);
  leds[25] = CRGB(255, 0, 0);
  leds[26] = CRGB(255, 0, 0);
  leds[27] = CRGB(255, 0, 0);
  leds[34] = CRGB(255, 0, 0);
  FastLED.show();
  delay(300);
  leds[10] = CRGB(255, 0, 0);
  leds[11] = CRGB(0, 0, 0);
  leds[13] = CRGB(255, 0, 0);
  leds[14] = CRGB(255, 0, 0);
}
```



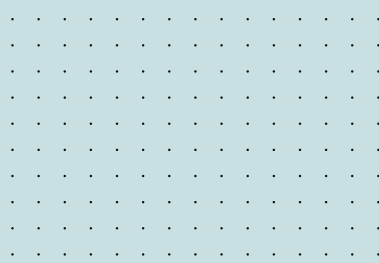
CODE -ANIMACIONES

```
leds[21] = CRGB(255, 0, 0);
leds[22] = CRGB(255, 0, 0);
leds[24] = CRGB(255, 0, 0);
leds[28] = CRGB(255, 0, 0);
leds[29] = CRGB(255, 0, 0);
leds[30] = CRGB(255, 0, 0);
leds[33] = CRGB(255, 0, 0);
leds[35] = CRGB(255, 0, 0);
leds[36] = CRGB(255, 0, 0);
leds[37] = CRGB(255, 0, 0);
leds[42] = CRGB(255, 0, 0);
leds[43] = CRGB(255, 0, 0);
leds[44] = CRGB(255, 0, 0);
leds[51] = CRGB(255, 0, 0);
FastLED.show();
delay(300);
leds[0] = CRGB(255, 0, 0);
leds[1] = CRGB(255, 0, 0);
leds[2] = CRGB(255, 0, 0);
leds[6] = CRGB(255, 0, 0);
leds[7] = CRGB(255, 0, 0);
leds[11] = CRGB(255, 0, 0);
leds[15] = CRGB(255, 0, 0);
leds[23] = CRGB(255, 0, 0);
leds[31] = CRGB(255, 0, 0);
leds[38] = CRGB(255, 0, 0);
leds[39] = CRGB(255, 0, 0);
leds[45] = CRGB(255, 0, 0);
leds[46] = CRGB(255, 0, 0);
leds[52] = CRGB(255, 0, 0);
leds[53] = CRGB(255, 0, 0);
leds[60] = CRGB(255, 0, 0);
FastLED.show();
delay(300);
leds[32] = CRGB(255, 0, 0);
leds[41] = CRGB(255, 0, 0);
leds[47] = CRGB(255, 0, 0);
leds[50] = CRGB(255, 0, 0);
leds[54] = CRGB(255, 0, 0);
leds[59] = CRGB(255, 0, 0);
leds[61] = CRGB(255, 0, 0);
FastLED.show();
delay(300);
leds[40] = CRGB(255, 0, 0);
leds[49] = CRGB(255, 0, 0);
leds[55] = CRGB(255, 0, 0);
leds[58] = CRGB(255, 0, 0);
leds[62] = CRGB(255, 0, 0);
FastLED.show();
delay(300);
leds[3] = CRGB(255, 0, 0);
leds[4] = CRGB(255, 0, 0);
leds[5] = CRGB(255, 0, 0);
leds[48] = CRGB(255, 0, 0);
```



CODE -ANIMACIONES

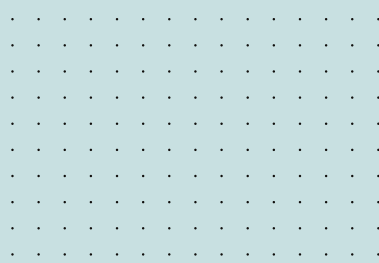
```
leds[56] = CRGB(255, 0, 0);
leds[57] = CRGB(255, 0, 0);
leds[63] = CRGB(255, 0, 0);
FastLED.show();
delay(600);
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(255, 0, 0);
leds[2] = CRGB(255, 0, 0);
leds[3] = CRGB(0, 0, 0);
leds[4] = CRGB(0, 0, 0);
leds[5] = CRGB(255, 0, 0);
leds[6] = CRGB(255, 0, 0);
leds[7] = CRGB(0, 0, 0);
leds[8] = CRGB(255, 0, 0);
leds[9] = CRGB(255, 0, 134);
leds[10] = CRGB(255, 0, 0);
leds[11] = CRGB(255, 0, 0);
leds[12] = CRGB(255, 0, 0);
leds[13] = CRGB(255, 0, 134);
leds[14] = CRGB(255, 0, 0);
leds[15] = CRGB(255, 0, 0);
leds[16] = CRGB(255, 0, 0);
leds[17] = CRGB(255, 0, 134);
leds[18] = CRGB(255, 0, 134);
leds[19] = CRGB(255, 0, 0);
leds[20] = CRGB(255, 0, 0);
leds[21] = CRGB(255, 0, 134);
leds[22] = CRGB(255, 0, 134);
leds[23] = CRGB(255, 0, 0);
leds[24] = CRGB(0, 0, 0);
leds[25] = CRGB(255, 0, 0);
leds[26] = CRGB(255, 0, 0);
leds[27] = CRGB(0, 0, 0);
leds[28] = CRGB(0, 0, 0);
leds[29] = CRGB(255, 0, 0);
leds[30] = CRGB(255, 0, 0);
leds[31] = CRGB(0, 0, 0);
leds[32] = CRGB(0, 0, 0);
leds[33] = CRGB(0, 0, 0);
leds[34] = CRGB(0, 0, 0);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[37] = CRGB(0, 0, 0);
leds[38] = CRGB(0, 0, 0);
leds[39] = CRGB(0, 0, 0);
leds[40] = CRGB(0, 0, 0);
leds[41] = CRGB(255, 0, 0);
leds[42] = CRGB(0, 0, 0);
leds[43] = CRGB(0, 0, 0);
leds[44] = CRGB(0, 0, 0);
leds[45] = CRGB(0, 0, 0);
leds[46] = CRGB(255, 0, 0);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[49] = CRGB(0, 0, 0);
leds[50] = CRGB(255, 0, 0);
```



CODE -ANIMACIONES

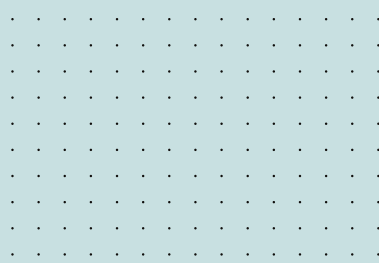
```
leds[51] = CRGB(255, 0, 0);
leds[52] = CRGB(255, 0, 0);
leds[53] = CRGB(255, 0, 0);
leds[54] = CRGB(0, 0, 0);
leds[55] = CRGB(0, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 0, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
}

void Csorprendido(){
  for (int i = 0; i <= 2; i++) {
    leds[0] = CRGB(255, 0, 255);
    leds[1] = CRGB(255, 0, 255);
    leds[2] = CRGB(255, 0, 255);
    leds[3] = CRGB(0, 0, 0);
    leds[4] = CRGB(0, 0, 0);
    leds[5] = CRGB(255, 0, 255);
    leds[6] = CRGB(255, 0, 255);
    leds[7] = CRGB(255, 0, 255);
    leds[8] = CRGB(255, 0, 255);
    leds[9] = CRGB(0, 0, 0);
    leds[10] = CRGB(255, 0, 255);
    leds[11] = CRGB(0, 0, 0);
    leds[12] = CRGB(0, 0, 0);
    leds[13] = CRGB(255, 0, 255);
    leds[14] = CRGB(0, 0, 0);
    leds[15] = CRGB(255, 0, 255);
    leds[16] = CRGB(255, 0, 255);
    leds[17] = CRGB(0, 0, 0);
    leds[18] = CRGB(255, 0, 255);
    leds[19] = CRGB(0, 0, 0);
    leds[20] = CRGB(0, 0, 0);
    leds[21] = CRGB(255, 0, 255);
    leds[22] = CRGB(0, 0, 0);
    leds[23] = CRGB(255, 0, 255);
    leds[24] = CRGB(255, 0, 255);
    leds[25] = CRGB(255, 0, 255);
    leds[26] = CRGB(255, 0, 255);
    leds[27] = CRGB(0, 0, 0);
    leds[28] = CRGB(0, 0, 0);
    leds[29] = CRGB(255, 0, 255);
    leds[30] = CRGB(255, 0, 255);
    leds[31] = CRGB(255, 0, 255);
    leds[32] = CRGB(0, 0, 0);
    leds[33] = CRGB(0, 0, 0);
    leds[34] = CRGB(0, 0, 0);
    leds[35] = CRGB(255, 0, 255);
```



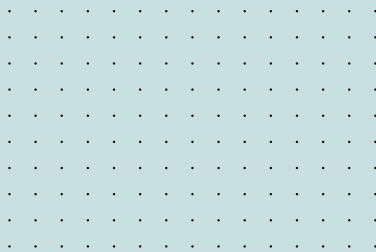
CODE -ANIMACIONES

```
leds[36] = CRGB(255, 0, 255);
leds[37] = CRGB(0, 0, 0);
leds[38] = CRGB(0, 0, 0);
leds[39] = CRGB(0, 0, 0);
leds[40] = CRGB(0, 0, 0);
leds[41] = CRGB(0, 0, 0);
leds[42] = CRGB(255, 0, 255);
leds[43] = CRGB(0, 0, 0);
leds[44] = CRGB(0, 0, 0);
leds[45] = CRGB(255, 0, 255);
leds[46] = CRGB(0, 0, 0);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[49] = CRGB(0, 0, 0);
leds[50] = CRGB(255, 0, 255);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 0, 0);
leds[53] = CRGB(255, 0, 255);
leds[54] = CRGB(0, 0, 0);
leds[55] = CRGB(0, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(255, 0, 255);
leds[60] = CRGB(255, 0, 255);
leds[61] = CRGB(0, 0, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(200);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[43] = CRGB(255, 0, 255);
leds[44] = CRGB(255, 0, 255);
leds[51] = CRGB(255, 0, 255);
leds[52] = CRGB(255, 0, 255);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
FastLED.show();
delay(200);
leds[0] = CRGB(0, 0, 0);
leds[1] = CRGB(0, 0, 0);
leds[2] = CRGB(0, 0, 0);
leds[5] = CRGB(0, 0, 0);
leds[6] = CRGB(0, 0, 0);
leds[7] = CRGB(0, 0, 0);
leds[9] = CRGB(255, 0, 255);
leds[14] = CRGB(255, 0, 255);
leds[35] = CRGB(0, 0, 0);
leds[36] = CRGB(0, 0, 0);
leds[42] = CRGB(0, 0, 0);
leds[45] = CRGB(0, 0, 0);
leds[50] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
```



CODE -ANIMACIONES

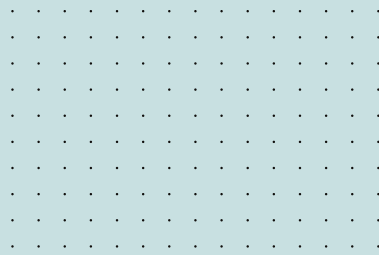
```
FastLED.show();
delay(200);
leds[8] = CRGB(0, 0, 0);
leds[9] = CRGB(0, 0, 0);
leds[10] = CRGB(0, 0, 0);
leds[13] = CRGB(0, 0, 0);
leds[14] = CRGB(0, 0, 0);
leds[15] = CRGB(0, 0, 0);
leds[16] = CRGB(0, 0, 0);
leds[17] = CRGB(255, 0, 255);
leds[22] = CRGB(255, 0, 255);
leds[23] = CRGB(0, 0, 0);
leds[24] = CRGB(0, 0, 0);
leds[31] = CRGB(0, 0, 0);
leds[43] = CRGB(0, 0, 0);
leds[44] = CRGB(0, 0, 0);
FastLED.show();
delay(200);
leds[8] = CRGB(255, 0, 255);
leds[9] = CRGB(255, 0, 255);
leds[10] = CRGB(255, 0, 255);
leds[13] = CRGB(255, 0, 255);
leds[14] = CRGB(255, 0, 255);
leds[15] = CRGB(255, 0, 255);
leds[16] = CRGB(255, 0, 255);
leds[17] = CRGB(0, 0, 0);
leds[22] = CRGB(0, 0, 0);
leds[23] = CRGB(255, 0, 255);
leds[24] = CRGB(255, 0, 255);
leds[31] = CRGB(255, 0, 255);
leds[43] = CRGB(255, 0, 255);
leds[44] = CRGB(255, 0, 255);
FastLED.show();
delay(200);
leds[0] = CRGB(255, 0, 255);
leds[1] = CRGB(255, 0, 255);
leds[2] = CRGB(255, 0, 255);
leds[5] = CRGB(255, 0, 255);
leds[6] = CRGB(255, 0, 255);
leds[7] = CRGB(255, 0, 255);
leds[9] = CRGB(0, 0, 0);
leds[14] = CRGB(0, 0, 0);
leds[42] = CRGB(255, 0, 255);
leds[45] = CRGB(255, 0, 255);
leds[50] = CRGB(255, 0, 255);
leds[53] = CRGB(255, 0, 255);
FastLED.show();
delay(200);
leds[35] = CRGB(255, 0, 255);
leds[36] = CRGB(255, 0, 255);
leds[43] = CRGB(0, 0, 0);
leds[44] = CRGB(0, 0, 0);
```



CODE -ANIMACIONES

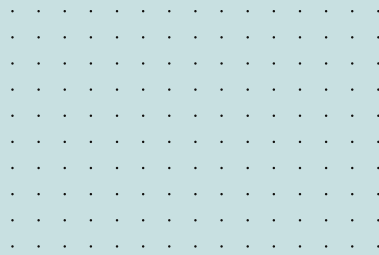
```
    leds[51] = CRGB(0, 0, 0);
    leds[52] = CRGB(0, 0, 0);
    leds[59] = CRGB(255, 0, 255);
    leds[60] = CRGB(255, 0, 255);
    FastLED.show();
    delay(200);
  }
}
```

```
void Casqueado(){
  for (int i = 0; i <= 2; i++) {
    leds[0] = CRGB(0, 0, 0);
    leds[1] = CRGB(0, 0, 0);
    leds[2] = CRGB(0, 0, 0);
    leds[3] = CRGB(0, 0, 0);
    leds[4] = CRGB(0, 0, 0);
    leds[5] = CRGB(0, 0, 0);
    leds[6] = CRGB(0, 0, 0);
    leds[7] = CRGB(0, 0, 0);
    leds[8] = CRGB(0, 0, 0);
    leds[9] = CRGB(0, 0, 0);
    leds[10] = CRGB(0, 0, 0);
    leds[11] = CRGB(0, 0, 0);
    leds[12] = CRGB(0, 0, 0);
    leds[13] = CRGB(0, 0, 0);
    leds[14] = CRGB(0, 0, 0);
    leds[15] = CRGB(0, 0, 0);
    leds[16] = CRGB(0, 0, 0);
    leds[17] = CRGB(0, 255, 0);
    leds[18] = CRGB(0, 0, 0);
    leds[19] = CRGB(0, 255, 0);
    leds[20] = CRGB(0, 0, 0);
    leds[21] = CRGB(0, 255, 0);
    leds[22] = CRGB(0, 0, 0);
    leds[23] = CRGB(0, 255, 0);
    leds[24] = CRGB(0, 0, 0);
    leds[25] = CRGB(0, 0, 0);
    leds[26] = CRGB(0, 255, 0);
    leds[27] = CRGB(0, 0, 0);
    leds[28] = CRGB(0, 0, 0);
    leds[29] = CRGB(0, 0, 0);
    leds[30] = CRGB(0, 255, 0);
    leds[31] = CRGB(0, 0, 0);
    leds[32] = CRGB(0, 0, 0);
    leds[33] = CRGB(0, 0, 0);
    leds[34] = CRGB(0, 0, 0);
    leds[35] = CRGB(0, 0, 0);
    leds[36] = CRGB(0, 0, 0);
    leds[37] = CRGB(0, 0, 0);
    leds[38] = CRGB(0, 0, 0);
    leds[39] = CRGB(0, 0, 0);
    leds[40] = CRGB(0, 0, 0);
    leds[41] = CRGB(0, 0, 0);
    leds[42] = CRGB(0, 255, 0);
    leds[43] = CRGB(0, 255, 0);
    leds[44] = CRGB(0, 255, 0);
  }
}
```



CODE -ANIMACIONES

```
leds[45] = CRGB(0, 255, 0);
leds[46] = CRGB(0, 255, 0);
leds[47] = CRGB(0, 0, 0);
leds[48] = CRGB(0, 0, 0);
leds[49] = CRGB(0, 0, 0);
leds[50] = CRGB(0, 0, 0);
leds[51] = CRGB(0, 0, 0);
leds[52] = CRGB(0, 255, 0);
leds[53] = CRGB(0, 0, 0);
leds[54] = CRGB(0, 255, 0);
leds[55] = CRGB(0, 0, 0);
leds[56] = CRGB(0, 0, 0);
leds[57] = CRGB(0, 0, 0);
leds[58] = CRGB(0, 0, 0);
leds[59] = CRGB(0, 0, 0);
leds[60] = CRGB(0, 0, 0);
leds[61] = CRGB(0, 255, 0);
leds[62] = CRGB(0, 0, 0);
leds[63] = CRGB(0, 0, 0);
FastLED.show();
delay(300);
leds[18] = CRGB(0, 255, 0);
leds[22] = CRGB(0, 255, 0);
leds[26] = CRGB(0, 0, 0);
leds[30] = CRGB(0, 0, 0);
leds[53] = CRGB(0, 255, 0);
leds[61] = CRGB(0, 0, 0);
FastLED.show();
delay(300);
leds[9] = CRGB(0, 255, 0);
leds[10] = CRGB(0, 255, 0);
leds[11] = CRGB(0, 255, 0);
leds[13] = CRGB(0, 255, 0);
leds[14] = CRGB(0, 255, 0);
leds[15] = CRGB(0, 255, 0);
leds[52] = CRGB(0, 0, 0);
leds[54] = CRGB(0, 0, 0);
FastLED.show();
delay(300);
leds[18] = CRGB(0, 0, 0);
leds[22] = CRGB(0, 0, 0);
leds[25] = CRGB(0, 255, 0);
leds[26] = CRGB(0, 255, 0);
leds[27] = CRGB(0, 255, 0);
leds[29] = CRGB(0, 255, 0);
leds[30] = CRGB(0, 255, 0);
leds[31] = CRGB(0, 255, 0);
leds[53] = CRGB(0, 0, 0);
FastLED.show();
delay(600);
leds[18] = CRGB(0, 255, 0);
leds[22] = CRGB(0, 255, 0);
leds[25] = CRGB(0, 0, 0);
```

CODE -ANIMACIONES

```
    leds[26] = CRGB(0, 0, 0);
    leds[27] = CRGB(0, 0, 0);
    leds[29] = CRGB(0, 0, 0);
    leds[30] = CRGB(0, 0, 0);
    leds[31] = CRGB(0, 0, 0);
    leds[53] = CRGB(0, 255, 0);
    FastLED.show();
    delay(300);
    leds[9] = CRGB(0, 0, 0);
    leds[10] = CRGB(0, 0, 0);
    leds[11] = CRGB(0, 0, 0);
    leds[13] = CRGB(0, 0, 0);
    leds[14] = CRGB(0, 0, 0);
    leds[15] = CRGB(0, 0, 0);
    leds[52] = CRGB(0, 255, 0);
    leds[54] = CRGB(0, 255, 0);
    FastLED.show();
    delay(300);
    leds[18] = CRGB(0, 0, 0);
    leds[22] = CRGB(0, 0, 0);
    leds[26] = CRGB(0, 255, 0);
    leds[30] = CRGB(0, 255, 0);
    leds[53] = CRGB(0, 0, 0);
    leds[61] = CRGB(0, 255, 0);
    FastLED.show();
    delay(300);
  }
}
```

