

# Informe Parcial 1

Informática II

**Daniel Perez Gallego CC. 1193088770**  
**Miguel Serna Montoya CC. 1193129865**  
**Jorge Montaña Cisneros CC.**  
**1007327968**

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

# Índice

<b>1. Análisis del problema</b>	<b>1</b>
<b>2. Algoritmo Implementado</b>	<b>1</b>
<b>3. Problemas de desarrollo</b>	<b>11</b>
<b>4. Evolución del algoritmo</b>	<b>11</b>
4.1. Día 1 . . . . .	11
4.2. Día 2 . . . . .	12
4.3. Día 3 . . . . .	13
4.4. Día 4 . . . . .	13
4.5. Día 5 . . . . .	13
<b>5. Esquema</b>	<b>14</b>

## 1. Análisis del problema

El primer reto era evidente, debíamos conectar 64 luces LED a la mínima cantidad de pines posibles, claro esta, haciendo uso del dispositivo 74CH595 presentado en la clase, para si poder ampliar la cantidad de salidas digitales.

Terminado con eso, podríamos empezar a buscar métodos para encender ciertos LED y dejar los otros apagados y así poder formar figuras con la matriz 8x8, la opción mas lógica sería primero encender todos los LED y poner dicha solución en la función 'Verificación'. A partir de la solución usada en la función 'Imagen', la usaríamos en 'Publik' para que almacene los 3 caracteres deseados e imprima uno por uno.

Además de eso, debíamos hacer uso del serial para darle indicaciones al usuario y para que pudiese ingresar el numero, letra o carácter deseado y aclarar las restricciones en el manual de usuario y de este modo no generar errores.

## 2. Algoritmo Implementado

```
1 #define SER 2
2 #define SRCLK 3
3 #define RCLK 4
4 //Funcion para prender todos los leds
5 void verificacion();
6 //Funcion para mostrar el caracter en los leds
7 void imagen(char,int);
8 //Funcion para mostrar los patrones
```

```

9 void publik(char);
10 //Funcion para leer 8 bites de cada fila y mirar si estan
    disponibles
11 void desplazarbyte(uint8_t Pindato, uint8_t Pinreloj,
    uint8_t val);
12 short int opcion;
13 char letra;
14 void mostrarleds(int *Lista, int);
15 int ALL[] = {255,255,255,255,255,255,255,255};
16 float time = 4000.0;
17
18
19 void setup(){
20 //inicializamos el puerto serial
21   Serial.begin(9600);
22   //-----
23   //pin de salida
24   pinMode(SER, OUTPUT);
25   //pin de salida
26   pinMode(SRCLK, OUTPUT);
27   //pin de salida
28   pinMode(RCLK, OUTPUT);
29   Serial.println("Bienvend@. Que desea hacer?");
30   Serial.println("(1) Verificar que los leds funcionen");
31   Serial.println("(2) Mostrar un patron ingresado en los
    leds");
32   Serial.println("(3) Mostrar una secuencia de patrones
    en los leds");
33   Serial.println("(4) Salir");
34 }
35
36
37 void loop(){
38   //eleccion de usuario
39
40   //cuando se ingresa un dato se cumple la condicion
41   if(Serial.available()){
42     //ciclo en donde se recibe la opcion del usuario
43
44     opcion = Serial.read();
45     opcion = opcion-48;
46     //Opcion de verificacion
47     if(opcion == 1){
48       verificacion();
49       opcion = 0;
50     }

```

```

51 //Opcion de mostrar patron
52 else if(opcion == 2){
53     Serial.println("Por favor, ingrese la letra que sea ver en pantalla:");
54     while(true){
55         if(Serial.available()){
56             letra = Serial.read();
57             delay(1000);
58             imagen(letra,time);
59             Serial.println("\n\n\n\n\n\n\n\n\n");
60             break;
61         }
62     }
63     Serial.println("(1) Verificar que los leds funcionen");
64     Serial.println("(2) Mostrar un patron ingresado en los leds");
65     Serial.println("(3) Mostrar una secuencia de patrones en los leds");
66     Serial.println("(4) Salir");
67     opcion = 0;
68 }
69 //Mostrar secuencia de patrones
70 else if(opcion == 3){
71     publik();
72     opcion = 0;
73     Serial.println("\n\n\n\n\n\n\n\n\n");
74 }
75 else if(opcion == 4){
76     Serial.print("Adios bebe<3");
77     delay(400);
78     exit(0);
79 }
80 }while(opcion <1 && opcion >3);
81
82 }
83 //Definicion de funciones-----
84 void verificacion(){
85     for (int i=0; i<=8; i++) {
86
87         desplazarbyte(SER,SRCLK,~ALL[i]);
88         desplazarbyte(SER,SRCLK,128 >> i);
89         digitalWrite(RCLK, 1);
90         digitalWrite(RCLK, 0);
91     }
92 }

```

```

93 void imagen(char letra, float tiempo){
94     int *valores = new int[8];
95     if(letra >= 97 && letra <= 122){
96         letra -= 32;
97     }
98     //para la letra A
99     if(letra == 'A'){
100         int A[] = {0, 60, 102, 102, 126, 102, 102, 102};
101         valores = A;
102         mostrarleds(valores, tiempo);
103     }
104     //para letra B
105     else if(letra == 'B'){
106         int B[] = {248, 198, 198, 216, 216, 206, 198, 248};
107         valores = B;
108         mostrarleds(valores, tiempo);
109     }
110     //para letra C
111     else if(letra == 'C'){
112         int C[] = {0, 31, 127, 192, 192, 127, 31, 0};
113         valores = C;
114         mostrarleds(valores, tiempo);
115     }
116     //para letra D
117     else if(letra == 'D'){
118         int D[] = {248, 252, 198, 195, 195, 198, 252, 248};
119         valores = D;
120         mostrarleds(valores, tiempo);
121     }
122     //para letra E
123     else if(letra == 'E'){
124         int E[] = {255, 255, 224, 255, 255, 224, 255, 255};
125         valores = E;
126         mostrarleds(valores, tiempo);
127     }
128     //para letra F
129     else if(letra == 'F'){
130         int F[] = {126, 126, 96, 124, 124, 96, 96, 96};
131         valores = F;
132         mostrarleds(valores, tiempo);
133     }
134     //para letra G
135     else if(letra == 'G'){
136         int G[] = {62, 126, 192, 207, 207, 195, 127, 63};
137         valores = G;
138         mostrarleds(valores, tiempo);

```

```

139     }
140     //para letra H
141     else if(letra == 'H'){
142         int H[] = {195,195,195,255,255,195,195,195};
143         valores = H;
144         mostrarleds(valores,tiempo);
145     }
146     //para letra I
147     else if(letra == 'I'){
148         int I[] = {255,255,24,24,24,24,255,255};
149         valores = I;
150         mostrarleds(valores,tiempo);
151     }
152     //para letra J
153     else if(letra == 'J'){
154         int J[] = {255,24,24,24,152,216,120,48};
155         valores = J;
156         mostrarleds(valores,tiempo);
157     }
158     //para letra K
159     else if(letra == 'K'){
160         int K[] = {198,204,216,240,216,204,198,195};
161         valores = K;
162         mostrarleds(valores,tiempo);
163     }
164     //para letra L
165     else if(letra == 'L'){
166         int L[] = {48,48,48,48,48,48,62,62};
167         valores = L;
168         mostrarleds(valores,tiempo);
169     }
170     //para letra M
171     else if(letra == 'M'){
172         int M[] = {195,231,255,219,195,195,195,195};
173         valores = M;
174         mostrarleds(valores,tiempo);
175     }
176     //para letra N
177     else if(letra == 'N'){
178         int N[] = {227,227,243,219,219,207,199,199};
179         valores = N;
180         mostrarleds(valores,tiempo);
181     }
182     //para letra O
183     else if(letra == 'O'){
184         int O[] = {255,255,195,195,195,195,255,255};

```

```

185     valores = 0;
186     mostrarleds(valores, tiempo);
187 }
188 //para letra P
189 else if(letra == 'P'){
190     int P[] = {255,227,227,227,255,254,224,224};
191     valores = P;
192     mostrarleds(valores, tiempo);
193 }
194 //para letra Q
195 else if(letra == 'Q'){
196     int Q[] = {0,60,66,66,66,70,62,1};
197     valores = Q;
198     mostrarleds(valores, tiempo);
199 }
200 //para letra R
201 else if(letra == 'R'){
202     int R[] = {255,255,195,223,206,204,198,195};
203     valores = R;
204     mostrarleds(valores, tiempo);
205 }
206 //para letra S
207 else if(letra == 'S'){
208     int S[] = {255,255,192,255,255,3,255,255};
209     valores = S;
210     mostrarleds(valores, tiempo);
211 }
212 //para letra T
213 else if(letra == 'T'){
214     int T[] = {255,255,255,60,60,60,60,60};
215     valores = T;
216     mostrarleds(valores, tiempo);
217 }
218 //para letra U
219 else if(letra == 'U'){
220     int U[] = {195,195,195,195,195,195,255,255};
221     valores = U;
222     mostrarleds(valores, tiempo);
223 }
224 //para letra V
225 else if(letra == 'V'){
226     int V[] = {129,195,195,102,102,60,60,24};
227     valores = V;
228     mostrarleds(valores, tiempo);
229 }
230 //para letra W

```

```

231 else if(letra == 'W'){
232     int W[] = {129,153,153,153,153,153,153,126};
233     valores = W;
234     mostrarleds(valores,tiempo);
235 }
236 //para letra X
237 else if(letra == 'X'){
238     int X[] = {129,195,102,60,60,102,195,129};
239     valores = X;
240     mostrarleds(valores,tiempo);
241 }
242 //para letra Y
243 else if(letra == 'Y'){
244     int Y[] = {195,102,60,24,24,24,24,24};
245     valores = Y;
246     mostrarleds(valores,tiempo);
247 }
248 //para letra Z
249 else if(letra == 'Z'){
250     int Z[] = {255,255,6,12,24,48,255,255};
251     valores = Z;
252     mostrarleds(valores,tiempo);
253 }
254 //para numero 0
255 else if(letra == '0'){
256     int cero[] = {60,126,102,102,102,102,126,60};
257     valores = cero;
258     mostrarleds(valores,tiempo);
259 }
260
261 //para numero 1
262 else if(letra == '1'){
263     int uno[] = {24,56,88,24,24,24,24,24};
264     valores = uno;
265     mostrarleds(valores,tiempo);
266 }
267
268
269 //para numero 2
270 else if(letra == '2'){
271     int dos[] = {60,126,102,12,24,48,126,126};
272     valores = dos;
273     mostrarleds(valores,tiempo);
274 }
275
276 //para numero 3

```



```

277     else if(letra == '3'){
278         int tres[] = {60,126,102,14,14,102,126,60};
279         valores = tres;
280         mostrarleds(valores,tiempo);
281     }
282
283     //para numero 4
284     else if(letra == '4'){
285         int cuatro[] = {14,30,54,102,255,255,6,6};
286         valores = cuatro;
287         mostrarleds(valores,tiempo);
288     }
289
290     //para numero 5
291     else if(letra == '5'){
292         int cinco[] = {254,254,192,252,254,14,254,252};
293         valores = cinco;
294         mostrarleds(valores,tiempo);
295     }
296
297     //para numero 6
298     else if(letra == '6'){
299         int seis[] = {30,62,112,252,254,198,254,124};
300         valores = seis;
301         mostrarleds(valores,tiempo);
302     }
303
304
305     //para numero 7
306     else if(letra == '7'){
307         int siete[] = {255,255,6,12,24,48,96,192};
308         valores = siete;
309         mostrarleds(valores,tiempo);
310     }
311
312     //para numero 8
313     else if(letra == '8'){
314         int ocho[] = {60,66,66,60,66,66,60,0};
315         valores = ocho;
316         mostrarleds(valores,tiempo);
317     }
318
319     //para numero 9
320     else if(letra == '9'){
321         int nueve[] = {60,126,102,126,58,6,6,6};
322         valores = nueve;

```

```

323     mostrarleds(valores, tiempo);
324 }
325
326 //para emoji1
327 else if(letra == '!'){
328     int cara1[] = {0,36,36,0,66,66,60,0};
329     valores = cara1;
330     mostrarleds(valores, tiempo);
331 }
332
333 //para emoji2
334 else if(letra == '"'){
335     int cara2[] = {165,165,165,231,0,129,90,36};
336     valores = cara2;
337     mostrarleds(valores, tiempo);
338 }
339
340 //para emoji3
341 else if(letra == '#'){
342     int cara3[] = {102,102,0,126,66,66,126,0};
343     valores = cara3;
344     mostrarleds(valores, tiempo);
345 }
346
347 delete[] valores;
348 }
349 void desplazarbyte(uint8_t Pindato, uint8_t Pinreloj,
350     uint8_t val){
351     uint8_t i = 0;
352     for (i = 0; i < 8; i++) {
353         digitalWrite(Pindato, !(val & (1 << i)));
354         digitalWrite(Pinreloj, HIGH);
355         digitalWrite(Pinreloj, LOW);
356     }
357 }
358 void mostrarleds(int *Lista, float tiempo){
359     int contador = 0;
360     contador = 0;
361     int aux[1];
362     while(true){
363         for (int i=0; i <= 8; i++)
364             { desplazarbyte(SER, SRCLK, ~(* (Lista+i)));
365               desplazarbyte(SER, SRCLK, 128 >> i);
366               digitalWrite(RCLK, 1);
367               digitalWrite(RCLK, 0);

```

```

368     }
369     delay(15);
370     contador = contador + 2;
371     if(contador > tiempo/100){
372         break;
373     }
374 }
375 }
376 }
377 void publik() {
378     char charpatron[]="";
379     int Npatron;
380     float Dpatron;
381     Serial.println("Ingrese_la_cantidad_de_patrones_que_
382                     desea_visualizar");
383     while(true) {
384         if(Serial.available()) {
385             Npatron = Serial.parseInt();
386             delay(200);
387             break;
388         }
389     }
390     Serial.println("Ingrese_el_tiempo_de_visualizacion_en_
391                     segundos_entre_cada_patron:_");
392     while(true) {
393         if(Serial.available()) {
394             Dpatron = Serial.parseFloat();
395             delay(200);
396             break;
397         }
398     }
399     Dpatron = Dpatron*1000;
400     Serial.println("Ingrese_La_secuencia_de_patrones:_");
401     int largo = 0;
402     while(largo < Npatron) {
403         if(Serial.available()) {
404             for(int i=0; i<=Npatron; i++) {
405                 charpatron[i]=Serial.read();
406                 imagen(charpatron[i], Dpatron);
407                 delay(400);
408             }
409             break;
410         }
411     }
412     delay(200);
413 }

```

412 |  
413 }

### 3. Problemas de desarrollo

El principal de los problemas que se nos presento fue sobre como íbamos a indicar los valores que los LED debían tener para formar cada letra, al final decidimos realizarlo de la forma mas obvia y lógica, simplemente dando los valores en binario a cada fila para formar dicho carácter, por ejemplo, una fila completamente encendida seria '255', lo que es el equivalente en decimal a '11111111' y con esa misma lógica podríamos 'dibujar' las matrices fila por fia de cada letra.

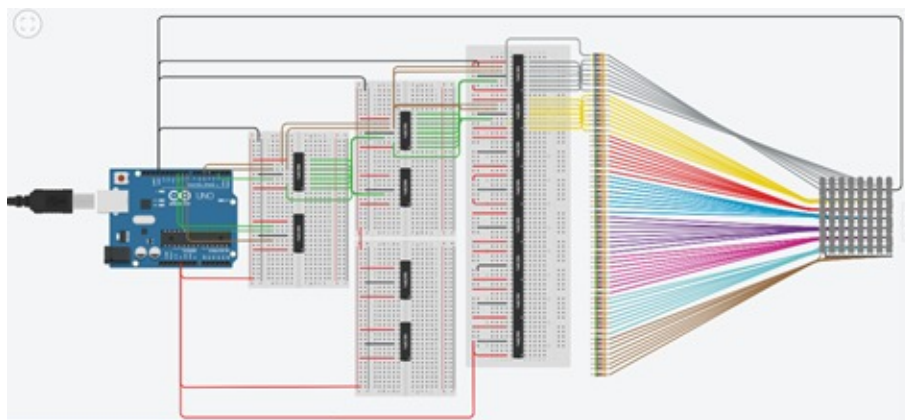
El hecho de que debíamos tener presente el arreglo de todas las letras, números y emojis, se convirtió en un problema para poder reconocer y validar todo lo que podía ingresar el usuario, además que la complejidad y lo extenso que se convirtió del algoritmo, el circuito y de que estuviéramos los 3 trabajando simultáneamente daban cavida a retrasos por la lentitud de la página y de que si uno de los miembros borraba o agregaba algo, los otros dos no podían ver el cambio sino unos 10 segundos después.

Se nos presentaros algunos errores de conexion de los LED, llegando al punto de que se formaban las figuras, pero al revés.

### 4. Evolución del algoritmo

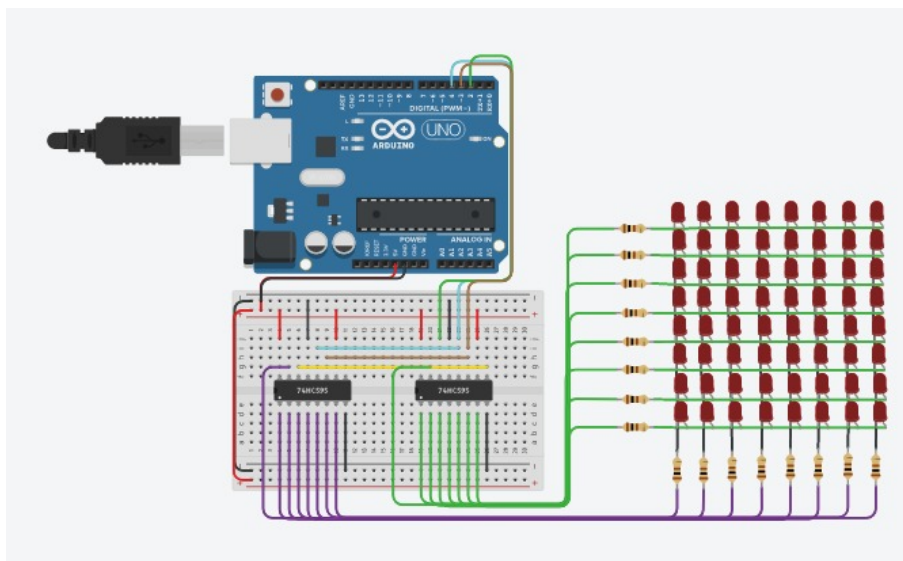
#### 4.1. Día 1

Dia 1, 17 de abril. Implementamos un circuito en base a las conexiones mostradas en la clase del sábado, junto a la matriz 8x8 de los LED



## 4.2. Día 2

Cambiamos significativamente el circuito, con el propósito principal de reducir la cantidad de pines usados lo máximo posible y logramos idear uno que solo hace uso de 3 pines, además de librarnos de posibles errores por tanta conexión. Para el algoritmo inicializamos la estructura que íbamos a seguir, las salidas del SER, SRCLK, RCLK y también el puerto serial. Inclusive, realizamos la primera función 'Verificacion' con éxito, dándole valores en binario como habíamos acordado antes.



Para la función verificación, que será la base para el resto de funciones, recorrimos las 8 filas de los LED y le asignamos su valor en binario, que en este caso todos bombillos deberían estar encendidos, dentro del arreglo 'ALL' utilizando la función desplazarbyte para leer los bits de cada fila y comprobar si están disponibles

```

96 void verificacion() {
97     for (int i=0; i<=8; i++) {
98
99         desplazarbyte(SER, SRCLK, ~ALL[i]);
100        desplazarbyte(SER, SRCLK, 128 >> i);
101
102        digitalWrite(RCLK, 1);
103        digitalWrite(RCLK, 0);
104    }
105 }

```

### 4.3. Día 3

Finalmente, avanzamos hasta implementar la función publik a un punto aceptable, con sus respectivas restricciones dadas en la guía. Definimos las funciones para cada letra del abecedario, número decimal y para los emojis dentro de la función imagen para simplificar el trabajo y ser un poco más eficientes, además de agregarle punteros y arreglos

```

313 void publik(){
314     char charpatron[]="";
315     int Npatron, Dpatron;
316     Serial.println("Ingrese la cantidad de patrones que desea visualizar");
317     while(true){
318         if(Serial.available()){
319             Npatron = Serial.parseInt();
320             delay(200);
321             break;
322         }
323     }
324     Serial.println("Ingrese el tiempo de visualizacion en segundos entre cada patron : ");
325     while(true){
326         if(Serial.available()){
327             Dpatron = Serial.parseInt();
328             delay(200);
329             break;
330         }
331     }
332     Dpatron = Dpatron*1000;
333     Serial.println("Ingrese La secuencia de patrones: ");
334     int largo = 0;
335     while(largo < Npatron){
336         if(Serial.available()){
337             for(int i=0; i<=Npatron; i++){
338                 charpatron[i]=Serial.read();
339                 imagen(charpatron[i], Dpatron);
340                 delay(400);

```

### 4.4. Día 4

Realizamos los arreglos finales al código e implementamos memoria dinámica, reestructuración de datos y punteros.

### 4.5. Día 5

Se rediseñaron los caracteres para imprimir en el arreglo 8x8

## 5. Esquema

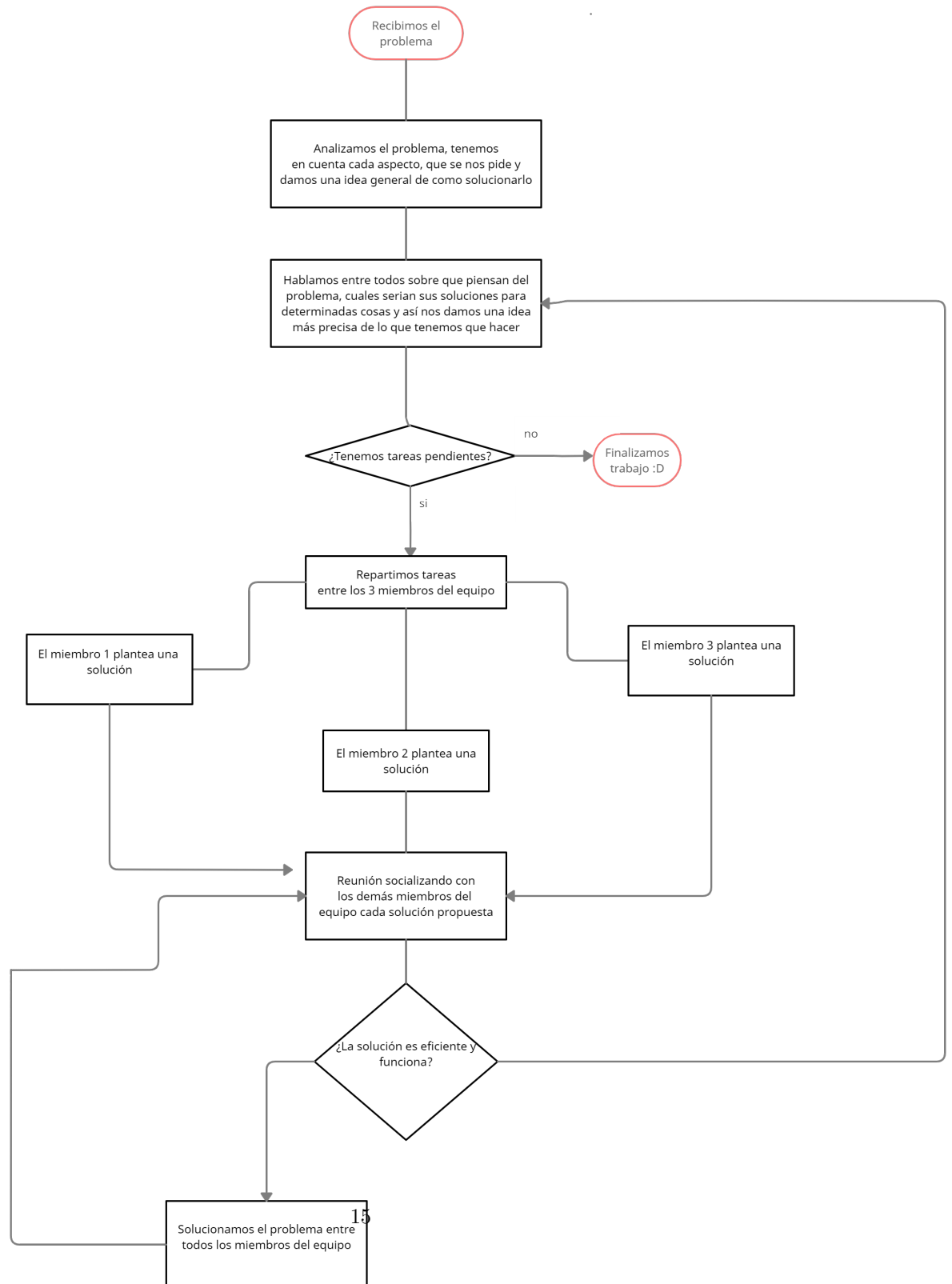


Figura 1: diagrama