

# **PARCIAL 1**

Informática II

**Juan Esteban Garcia Durango**  
**Jessica Valentina Gaviria Samboni**

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

# Índice

<b>1. Sección introductoria</b>	<b>2</b>
<b>2. Sección de contenido</b>	<b>2</b>
2.1. Análisis del problema y consideraciones para la alternativa de solución propuesta. . . . .	2
2.2. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo. . . . .	3
2.3. Algoritmo implementado. . . . .	3
2.4. Problemas de desarrollo que presentó. . . . .	3
2.5. Incluir código en el documento . . . . .	3
<b>3. Inclusión de imágenes</b>	<b>4</b>

## **1. Sección introductoria**

El siguiente trabajo contiene el proceso realizado en pro de el desarrollo del desafío planteado por la empresa Informa2 S.A.S..

## **2. Sección de contenido**

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

El primer planteamiento que nos trazamos respecto al problema propuesto, imprimir en una matriz de leds 8\*8 un mensaje ingresado por el usuario, fue el cómo reducir la cantidad de puertos digitales a emplear, y esto tuvo una solución viable en el hecho de poder conectar los integrados 74HC595 en serie, ya que de esta forma se emplean los mismos 3 puertos del arduino, y lo que se hace es que los bits ingresados se van desplazando, hasta llenar el registro.

Otros de los planeamientos previos fueron:

**1.** Decidir si los patrones serían predeterminados o diseñados por el usuario. Bajo la aparición de esta duda, decidimos darle la facilidad y oportunidad al usuario de que si decidía mostrar una letra en la matriz de leds, simplemente sería ingresar la letra mayúscula que quisiera. Pero de igual forma se le permitiría generar el mismo el patrón o patrones a su gusto. De lo anterior se generaron 2 alternativas:

**1.1** Manejo de los bits para poder representar la letra mayúscula predeterminada: Cada letra tendría su propio arreglo, que a la vez almacenaría subarreglos, en los cuales cada subarreglo contendría la información de los bits para cada columna de leds.

**1.2** Manejo de los bits para representar el patrón deseado por el usuario: Para este ítem lo que hicimos fue crear un arreglo bidimensional de [8][8], el cual tendría en todas sus entradas el valor cero inicialmente, pero que se vería modificado a 1, en caso de que el usuario quisiera emplear ese led en su patrón y por consiguiente encenderlo (estado 1 o HIGH).

**2.2. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.**

**2.3. Algoritmo implementado.**

**2.4. Problemas de desarrollo que presentó.**

Unos de los problemas que presentaron fueron: SE NOS BORRÓ EL CIRCUITO 1. En un primer momento no teníamos idea de cómo reducir los puertos digitales a usar, ya que originalmente tomamos la decisión de interconectar el SRCLK y el RCLK de los 8 integrados que usamos, para así solamente emplear un puerto, ya que el SRCLK y el RCLK funcionan bajo la misma idea de activación por flanco. Pero aún así nos seguían faltando pines, ya que pretendíamos conectar la entrada serial de cada integrado a un pin(8 en total). En vista de lo anterior y gracias a una investigación ardua, encontramos la forma de reducir los puertos a emplear.

**2.5. Incluir código en el documento**

A continuación, se presenta el código 2.5, que nos permite incluir en el informe partes de programa que requieran una explicación adicional.

```
// Programa desarrollado , compilado y ejecutado en https://www.onlinegd  
#include <iostream>
```

```
/*  
 * Esto es un comentario de varias lineas  
*/
```

```
// Comentario de una sola linea
```

```
#define N 10
```

```
using namespace std;
```

```
int main()  
{
```

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

```

        if( !( i % 2) )
            cout << "El valor de i es -> " << i << endl;
    }

    return 0;
}

//Resultado programa

/*
El valor de i es -> 0
El valor de i es -> 2
El valor de i es -> 4
El valor de i es -> 6
El valor de i es -> 8
*/

```

En la sección 3, se presentará como añadir ilustraciones al texto.

### 3. Inclusión de imágenes

En la Figura (1), se presenta el logo de C++ contenido en la carpeta images.



Figura 1: Logo de C++

Las secciones (1), (2) y (3) dependen del estilo del documento.

## Referencias