

# **Informe del parcial 1**

Informática 2

**Francis David Roa Bernal**  
**David Correa Ochoa**

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

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Ejecución del parcial</b>	<b>2</b>
2.1. Analisis de problemas y Consideraciones . . . . .	2
2.2. Esquema de tareas . . . . .	2
2.3. Algoritmo . . . . .	3
2.4. Problemas de desarrollo . . . . .	3
2.5. Evolucion del algoritmo y consideraciones para la implementación	3
<b>3. Inclusión de imágenes</b>	<b>4</b>

## 1. Introducción

Se requiere de un programa en lenguaje C++ que por medio de la plataforma Arduino Uno, registros de desplazamiento 74HC595 y la plataforma TINKER-CAD para mostrar patrones ingresados por el usuario mediante el monitor serial en un arreglo de 8X8 LEDs.

## 2. Ejecución del parcial

### 2.1. Analisis de problemas y Consideraciones

Cumplir con el objetivo propuesto en la introduccion depende de que los siguientes problemas sean resueltos

- El programa es muy pesado para el microcontrolador del arduino.
- No se sabe la cantidad de patrones que el usuario desea mostrar
- Lectura de datos recibidos en formato ascii por el monitor serial
- El usuario no sabe que patron está formando
- No se sabe si todos los LEDs están funcionando correctamente
- Los componentes pueden estar dañados
- Ordenar y cablear el arreglo de LEDs

Para lograr cumplir con el objetivo, se propusieron algunas ideas que al final no se implementaron en la versión final y usaron algunas herramientas que hicieron más facil encontrar errores en el codigo

- Uso de un transistor NPN bjt a modo de switch para hacer que el arreglo de LEDs muestre el patrón solo cuando los 74hc595 hallan terminado de recibir los datos del patron
- Usar un arreglo de matrices para guardar los patrones ingresados por el usuario, es decir, punteros triple.
- El uso del recurso <https://create.arduino.cc/editor> fue de gran utilidad al buscar y corregir errores
- Se usó el entorno Qt creator para la etapa inicial del codigo y la creacion de las funciones

### 2.2. Esquema de tareas

Las tareas mostradas en la imagen anterior son consultas sobre temas de la clase que no quedaron del todo claros

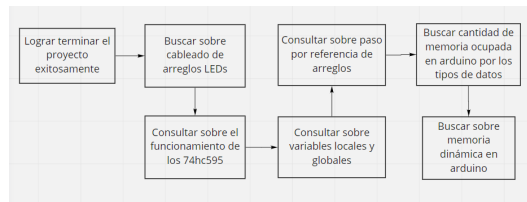


Figura 1: Esquema de tareas

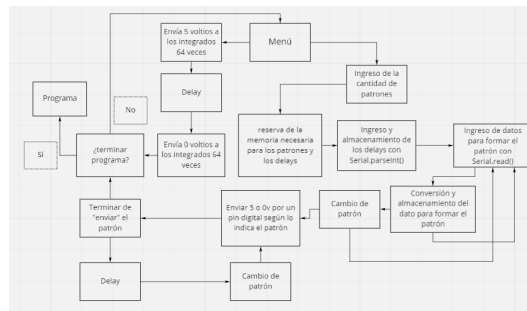


Figura 2: algoritmo implementado

## 2.3. Algoritmo

## 2.4. Problemas de desarrollo

Durante el desarrollo del código implementado en la versión final del proyecto, nos encontramos con diferentes problemas que mencionamos a continuación

- Al intentar trabajar con punteros triples y memoria dinámica nos encontramos con el programa corria infinitamente, por lo que el compilador no avanzaba
- Al intentar simular con transistores no obtuvimos el resultado deseado anteriormente expuesto
- Al enviar datos por el monitor serial y recibirlos con el `Serial.read()` notamos que guardaba los datos en formato ascii
- Al enviar datos por el monitor serial y recibirlos con el `Serial.parseInt()` notamos una gran ralentización en la ejecución del código

## 2.5. Evolucion del algoritmo y consideraciones para la implementación

Al implementar el algoritmo en la plataforma TINKERCAD tuvimos en cuenta las siguientes consideraciones

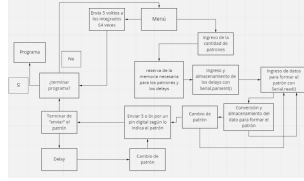


Figura 3: primer algoritmo

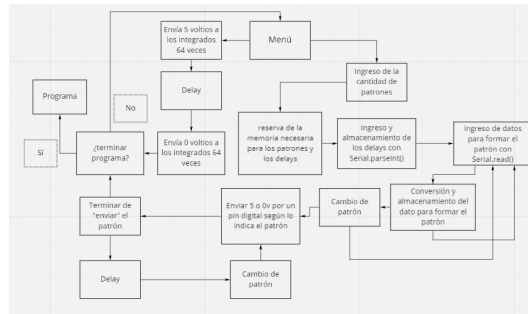


Figura 4: algoritmo implementado

- cambiar los cout por Serial.print() y Serial.println()
- cambiar los cin por Serial.read() y Serial.parseInt() según el punto del código en el que se fueran a implementar

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

### 3. Inclusión de imágenes

En la Figura (4), se presenta el logo de C++ contenido en la carpeta images. Las secciones (1), (2) y (3) dependen del estilo del documento.