



Universidad de Antioquia  
Facultad de Ingenieria  
Informatica II

# Informe. Primer examen

Integrantes:

Juan Jose Baquero Arcila

Jose Alejandro Olivo Petit

Sebastian Bolivar Vanegas

Colombia, Medellín  
Febrero 2022

# Índice general

<b>1. Objetivos</b>	<b>2</b>
1.1. Objetivo general . . . . .	2
1.2. Objetivo específico . . . . .	2
<b>2. Resumen</b>	<b>3</b>
<b>3. Introducción</b>	<b>4</b>
<b>4. Marco Teorico</b>	<b>5</b>
<b>5. Análisis del problema</b>	<b>9</b>
<b>6. Marco experimental</b>	<b>11</b>
<b>7. Resultados</b>	<b>18</b>
<b>Referencias</b>	<b>19</b>

# Capítulo 1

## Objetivos

### 1.1. Objetivo general

Adquirir la capacidad para trabajar con Arduino e integrar la programación en C++, además de usar de manera adecuada las funciones de las plataformas explicadas en clase que permiten controlar el puerto serial y los puertos digitales.

### 1.2. Objetivo específico

- Desarrollar la capacidad de solución de problemas enfrentándonos a problemáticas de la vida cotidiana.
- Adquirir las destrezas y conocimientos fundamentales de la programación con C++, en donde resaltamos estructuras de programación (iterativas, secuenciales y de decisión), tipos de datos, apuntadores, arreglos, funciones y memoria dinámica.

## Capítulo 2

# Resumen

Se elaboro un sistema de descriptación, mediante el cual se recibe una cadena de números; será transformada al sistema binario para poder mandar los datos en serie, pasará al sistema que paraleliza la información y al de recepción. Luego serian llevados a una de las compuertas del sistema integrado, para salir paralelizados, posteriormente se verifica con la lógica respectiva y determinar el caso en que se ha encontrado la bandera y contar la cantidad de esta.

## Capítulo 3

# Introducción

El motivo principal del trabajo realizado es implementar un sistema de des-criptación que permitirá cifrar datos, en el cual la información viajará desde un sistema generador de información hacia un sistema de salida que se encargará de tomar las decisiones pertinentes. Para llevar a cabo con éxito este proceso se realizaron las siguientes técnicas:

El sistema generador de información recibirá un arreglo de números determinados para utilizar el dicho arreglo; La cadena será convertida al sistema binario para de esta manera poder mandar los datos en serie, después de que la cadena es convertida en el sistema que genera la información, esta pasa tanto al sistema que paraleliza los datos como al otro sistema de recepción; nos enfocamos en dos bloques; los cuales son, el sistema de recepción y el sistema de paralelización. Estos datos que se encuentran en serie serian llevados a una de las compuertas del sistema integrado, ya que entran en serie y salen paralelizados, el número será verificado con la lógica pertinente y dependiendo de la salida de la lógica, nos dirigimos al sistema de recepción para organizar el algoritmo. En caso de encontrar la bandera se retornará un verdadero y lo siguiente en este caso será contar la cantidad de banderas.

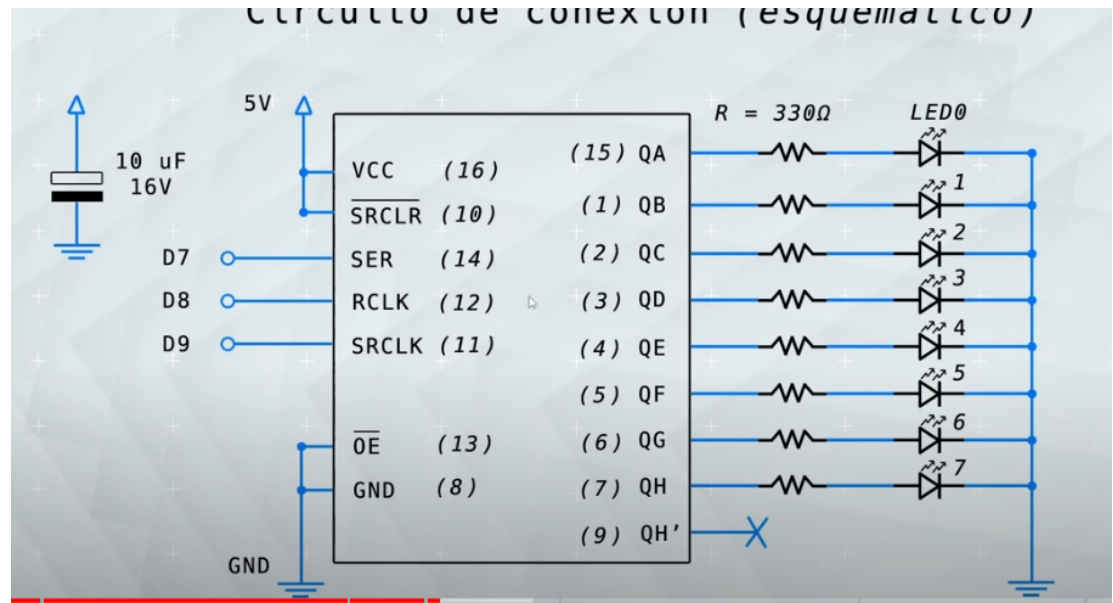
## Capítulo 4

# Marco Teorico

La paralelizacion es una forma de organizar cierto tipo de cosa, El paralelismo es una forma de computación en la cual varios cálculos pueden realizarse simultáneamente, La computación paralela es un tipo de arquitectura informática en la que varios procesadores ejecutan simultáneamente múltiples cálculos más pequeños desglosados de un problema complejo más grande en general. (Farestrepoca, 2016).

Para el funcionamiento del circuito integrado 74HC595, tuvimos que valernos de la documentación de Texas instrument para conocer el funcionamiento de los 16 puertos que tiene el integrado. En la siguiente imagen podremos apreciar el uso de de cada uno de los puertos con su respectivo nombre. Como los mas relevantes podemos encontrar los 8 puertos destinados para la paralelización de datos ubicados del 1 al 7 y el 15, también los puertos de voltaje y conexión a tierra y los tres puertos para el control del circuito. Una de las tres señales que controlan el funcionamiento es donde le entra la data de forma serial (SER), otra es usada para el registro de entrada del reloj (RCLK) en la cual ya organizo mis salidas con valores y una ultima señal(SRCLK) la cual se encarga de registrar la entrada del reloj del sistema. Con estos puertos podemos entender como seria el funcionamiento de este integrado, el cual es el de registro de desplazamiento. Este se da gracias a que recibe datos de forma serial para luego tenerlos separados simultáneamente en los diferentes puertos que tiene el circuito.

(intrument, 1982).



#### Resto de los integrados:

para la parte lógica, necesitamos usar tres tipos diferentes de circuitos integrados.

El primero sería el 74hc86 el cual contiene 4 compuertas XOR dentro de su esquema. Este tipo de compuertas funcionan con la lógica de que, si los bits a comparar son iguales, entonces la salida es cero. Si por el contrario los bits a comparar son diferentes, este dará como salida un 1 lógico. Este contiene 14 pines de los cuales tendremos 4 salidas de las compuertas.

El circuito integrado 74hc04 nos sirve para trabajar con la lógica de la compuerta inversora. Esta lógica como su nombre lo dice, invierte el bit que entra a la compuerta en su valor contrario. Si es alto pasa a bajo y viceversa. Este tiene 14 pines de los cuales 6 son para la lógica inversora.

El último circuito integrado a utilizar sería el 74hc08, en este se ve representada la lógica de las compuertas AND. Este tipo de comparación se basa en que los dos bits a comparar dan como resultado 1, si y solo si los dos bits comparados son unos; de resto el resultado sería cero. Este contiene 14 pines y 4 compuertas lógicas en su esquema. También hacemos uso de un led y una resistencia de 330 oh para poder verificar si la lógica está bien.

(INC., 2013b).

(INC., 2013c).

(INC., 2013a).

I2C es un puerto y protocolo de comunicación serial diseñado por Philips se-

miconductors a principios de los 80s, define la trama de datos y las conexiones físicas para transferir bits entre 2 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA y SCL. Además el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas con hasta velocidades de 100, 400 y 1000 kbits/s. También es conocido como IIC ó TWI- Two Wire Interface.

#### **Descripción de las señales:**

o **SCL (System Clock)** Es la línea de los pulsos de reloj que sincronizan el sistema.

o **SDA (System Data)** Es la línea por la que se mueven los datos entre los dispositivos.

o **GND (Masa)** Común de la interconexión entre todos los dispositivos enchufados al bus.

**Las líneas SDA y SCL:** Son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistores "pull- up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

**El MAESTRO I2C:** se encarga de controlar al cable de reloj, por sus siglas en inglés llamada SCL – Serial CLock. Además el MAESTRO se encarga de iniciar y parar la comunicación. La información binaria serial se envía sólo por la línea o cable de datos seriales, en inglés se llama SDA – Serial DAta. Dos Maestros no pueden hacer uso de un mismo puerto I2C. Puede funcionar de dos maneras, como maestro-transmisor o maestro-receptor. Sus funciones principales son:

- Iniciar la comunicación – S
- Enviar 7 bits de dirección – ADDR
- Generar 1 bit de Lectura ó Escritura – R/W
- Enviar 8 bits de dirección de memoria
- Transmitir 8 bits de datos –
- Confirmar la recepción de datos – ACK – ACKnowledged
- Generar confirmación de No-recepción, NACK – No-ACKnowledged
- Finalizar la comunicación

El ESCLAVO I2C, generalmente suele ser un sensor. Este elemento suministra de la información de interés al MAESTRO. Puede actuar de dos formas: esclavo-transmisor ó esclavo-receptor. Un dispositivo I2C esclavo, no puede generar a la señal SCL. Sus funciones principales son:

- Enviar información en paquetes de 8 bits.



- Enviar confirmaciones de recepción, llamadas ACK (Carletti, 2017).

## Capítulo 5

# Análisis del problema

### **Transmisión:**

En la primera parte de la transmisión, primero recibiremos el arreglo de números enteros proporcionados para nuestro equipo. Para esta parte se creará un algoritmo probado en Qt que reciba cadenas de cualquier tamaño y sea capaz de poder operar con ellas, para luego proceder a hacer la determinada transformación del número de decimal a binario para así poder proporcionar la data que el sistema completo requiere. También en esta parte hay una señal de reloj, que junto a la señal de data que es el número convertido a binario se le pasa al siguiente sistema que paraleliza los datos y al sistema de recepción de información.

### **Paralelización y descriptación:**

En esta parte del sistema, usaremos ya la parte del circuito integrado para poder utilizar la data de forma eficiente, paralelizándola de forma que la tengamos en cada uno de los puertos de nuestro circuito integrado. Cuando ya tengamos nuestra data organizada de la manera que queremos gracias a nuestro circuito, pasaremos a la parte de la lógica del sistema de descriptación que dependerá de la bandera que nos haya tocado y de lo que tengamos que hacer después de tener la bandera. Para la parte de recibir la bandera usaremos la lógica combinatorial de las compuertas más conocidas (and,or,xor) para poder darle lógica a nuestra bandera y empezar la descriptación de los datos y para la parte del procesamiento de los datos podemos usar el reloj del sistema para poder decidir que números del arreglo de la cadena son los que necesitamos.

### **Recepción:**

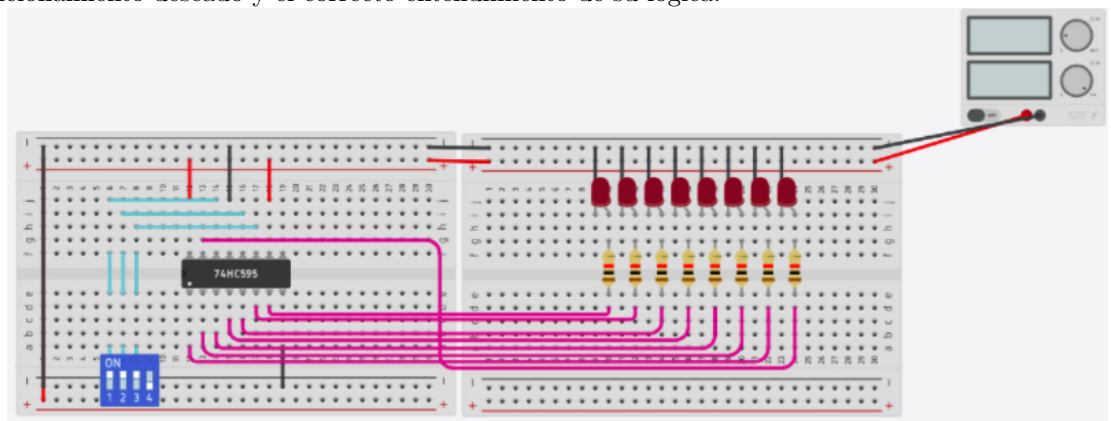
Después de haber hecho válida la lógica de la bandera en nuestro arreglo de números, procederemos a decirle al sistema de recepción que debe hacer con los datos que le están llegando en serie desde el sistema que genera la información, entonces gracias a esta lógica él puede saber cuáles son los que necesita y así guardarlos para luego poderlos pasar al sistema de visualización y mandarle los

datos clasificados al pc2.

## Capítulo 6

# Marco experimental

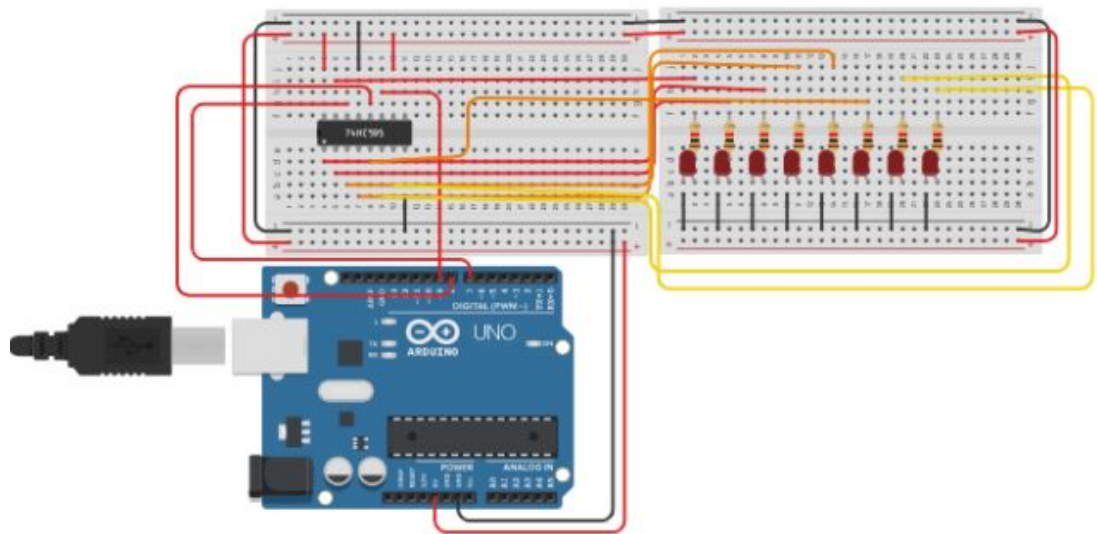
Para poder aprender a utilizar el circuito integrado que me paraleliza los datos, tendremos que hacer un pequeño diseño para dar cuenta de su funcionamiento y el uso de todas sus compuertas. Para este cometido utilizamos el simulador de circuitos tinkercad, en el crearemos un circuito con un switch, una fuente de voltaje, 8 leds con su respectiva resistencia de 330 y una placa de pruebas (protoboard). Aquí procederemos a conectar las compuertas del integrado según su documentación ya anexada. En este daremos uso de las tres compuertas más importantes para darle control al circuito, estas son la compuerta de la data (14) la cual nos envía información de forma serial, la SRCLK(11) la cual me mueve los registros que entran de forma serial hacia las otras compuertas y el RCLK(12) que cuando es uno lleva a la salida lo que este guardado en las compuertas que guardan el dato paralelizado. A la hora de realizarlo se tuvo inconvenientes con la manera de utilizar estas tres compuertas de control del circuito, pero después de investigar y experimentar un poco más, se llegó al funcionamiento deseado y el correcto entendimiento de su lógica.



Link del Tinkercad:

<https://www.tinkercad.com/things/7cgRHLXIbp0>

Ya que sabemos usar el circuito integrado de forma sencilla, procedemos a utilizarlo con las herramientas que nos reúnen hoy aquí. así que ahora en tinkercad daremos uso de la tarjeta Arduino uno dada por el simulador para aprender a utilizar los dos componentes al mismo tiempo. El uso de la tarjeta Arduino nos facilita mucho el uso de las compuertas de control, ya que dentro de esta tarjeta podemos dar uso de código de Arduino que está basado en c++ y con este poder darle la lógica que el 74hc595 necesita para poder paralelizar los datos de forma correcta. Procedemos a enumerar los puertos que necesitamos del Arduino con pinMode y de ahí utilizamos la función SHIFTOUT para poder usar la lógica del control del integrado. También damos uso de 8 leds con sus resistencias para verificar que si estamos haciendo el registro de desplazamiento que pensábamos. Todo funciona como es debido.



**Link del Tinkercad:**

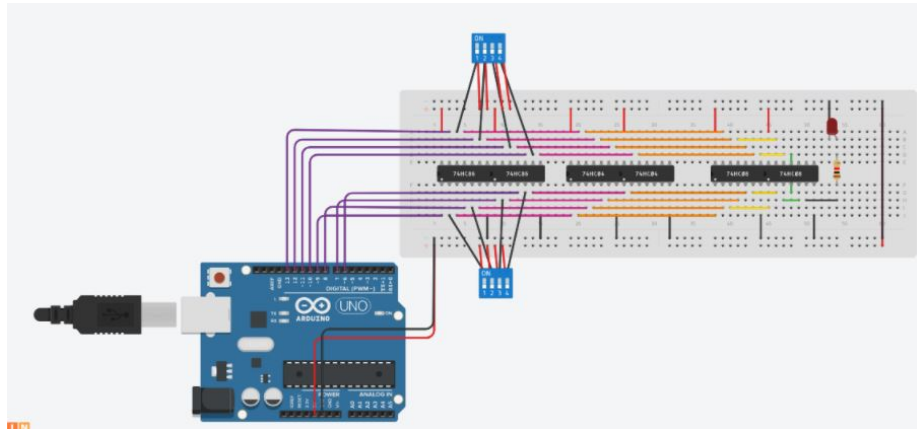
<https://www.tinkercad.com/things/cKVCXVzINTh-glorious-kup/editel>

Ahora para realizar la parte de descryptación procedemos a usar diseños ya utilizados anteriormente y le agregamos lo que nos hace falta para que pueda hacer su parte en el sistema. Para esto utilizamos el circuito de la tarjeta de Arduino y el circuito integrado y le agregamos 2 74hc06, 2 74hc04, 2 74hc08 y el uso de dos switches de 4 para simular un octeto de bits.

Ya después de tener los datos paralelizados, estos se comparan en las compuertas XOR con los valores por defecto que utilizamos en nuestros switches. Como salida esperada de estos integrados podemos pensar en que, si todos los bits paralelizados son iguales a los bits dejados por defecto por los swtiches, tendremos ceros en todas las salidas, de ahí con 2 circuitos 74HC04 procedemos a invertir los 8 posibles ceros que teníamos de la lógica anterior para volvernos a un uno y de ahí usar el circuito 74HC08 para usar la lógica AND por si algún

bit es diferente de 1. Si alguno era diferente de 1 era que el número no era igual a la bandera deseada, en caso contrario significa que es la bandera deseada y encenderíamos un led por su valor lógico de 1.

Ya como teníamos funcionando el circuito integrado, el añadido de la lógica con compuertas se hizo más sencillo por tener ya los bits deseados en los puertos del integrado que paraleliza. De aquí se verifico con los switches por defecto que representaban nuestra bandera si es la misma que la entrada por el 74HC595 con el encendido de un led.



```

1 // bloque de descriptacion
2 void setup()
3 {
4   pinMode(13, OUTPUT);
5   pinMode(12, OUTPUT);
6   pinMode(11, OUTPUT);
7   pinMode(10, OUTPUT);
8   pinMode(9, OUTPUT);
9   pinMode(8, OUTPUT);
10  pinMode(7, OUTPUT);
11  pinMode(6, OUTPUT);
12  digitalWrite(13,0);
13  digitalWrite(12,0);
14  digitalWrite(11,0);
15  digitalWrite(10,0);
16  digitalWrite(9,0);
17  digitalWrite(8,0);
18  digitalWrite(7,0);
19  digitalWrite(6,1);
20 }
21
22 void loop()
23 {
24
25 }

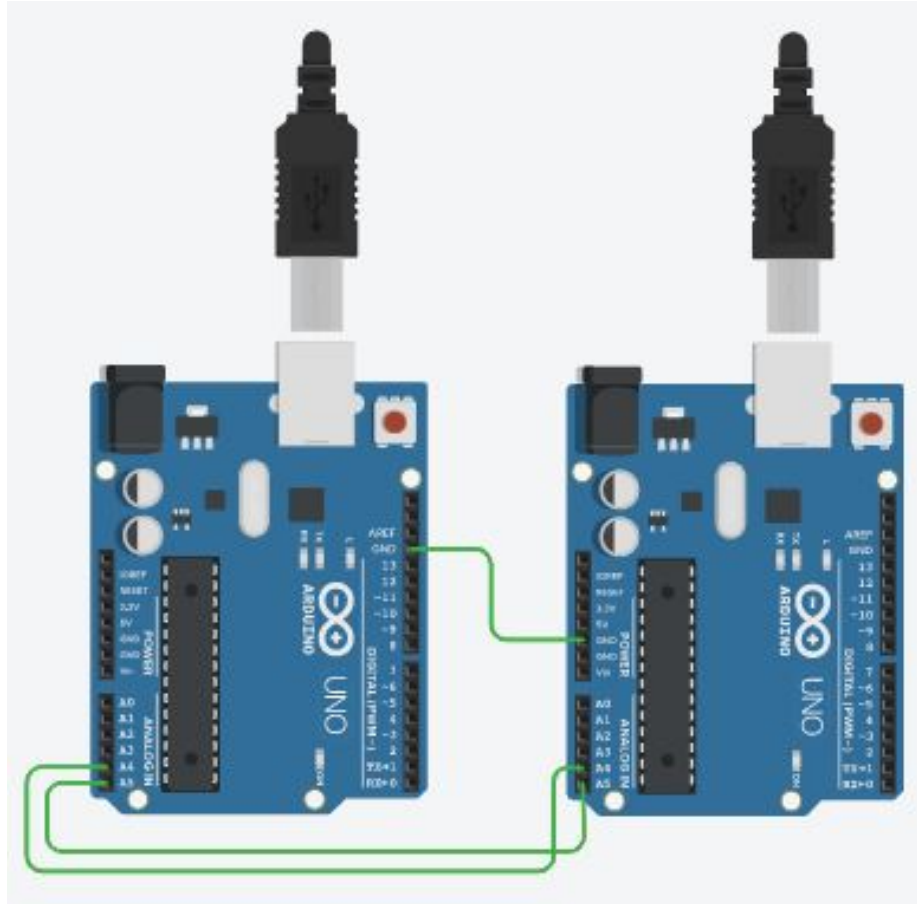
```

En esta parte hacemos uso de la plataforma tinkercad para probar el funcionamiento del circuito integrado. Primero lo probaremos con pulsadores y leds para ver su funcionamiento de forma manual.

Después de este lo probaremos utilizando el Arduino con el circuito integrado, conectando los diferentes pines a los led. Esto se muestra en los links siguientes:

<https://www.tinkercad.com/things/cKVCXVzINTg-lorious-kup/editel?tenant=circuits>  
Pruebas de funcionamiento de la transmisión serial.

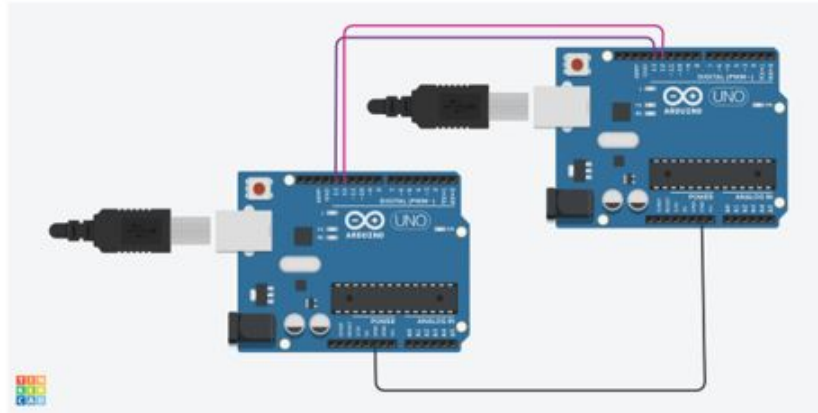
Para realizar la comunicación entre dos Arduinos probamos diferentes métodos, comenzamos utilizando el protocolo de I2C el cual requiere de 3 conexiones , en este caso la conexión entre los puertos A4 siendo el SDA la línea que transfiere los datos, además la conexión entre los puertos A5 serian el SCL funcionando como el reloj y la tierra, y el uso de la librería Wire.h para poder manejarlo, al implementarlo la transmisión de datos funciono de forma satisfactoria, luego necesitábamos un contador para contar la cantidad de veces que encontrábamos la bandera pero no logramos encontrar una forma de mantener un contador por todo el tiempo de ejecución durante cada envió de datos al Arduino esclavo, por lo tanto descartamos su uso.



**link de tinkercad:** <https://www.tinkercad.com/things/jMFM9EehBc>

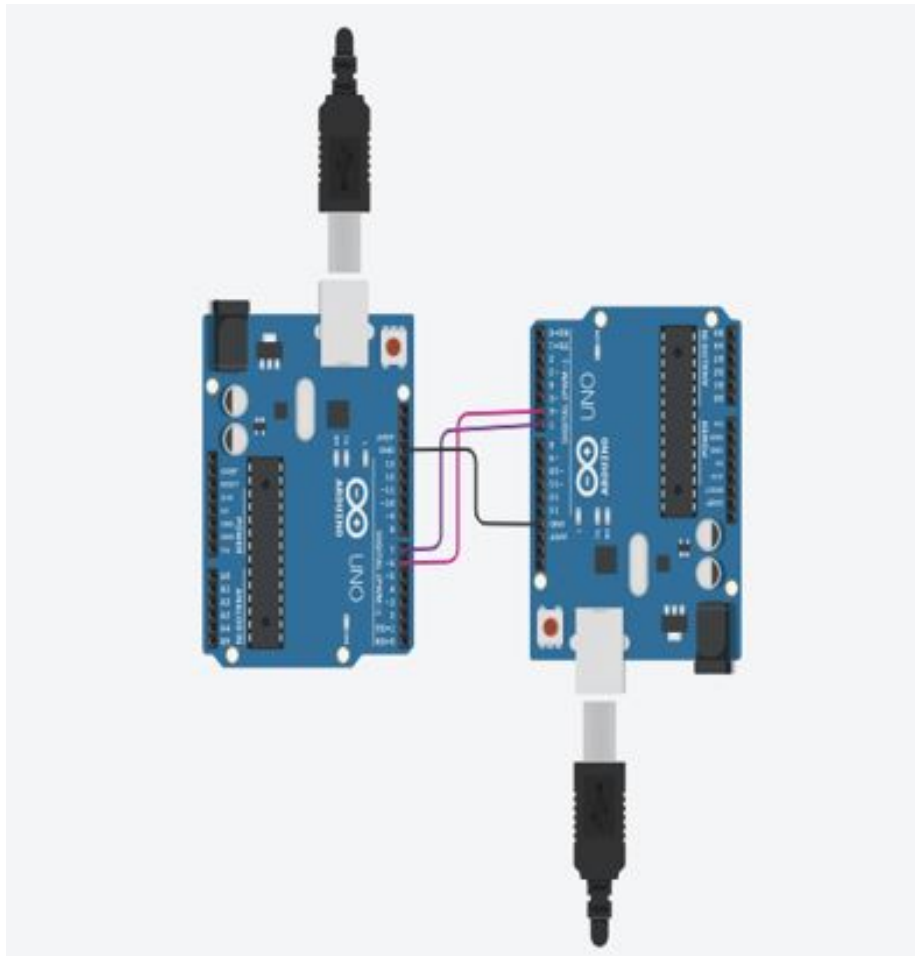
Luego como segunda opción para comunicar los Arduino realizamos dos conexiones entre puertos digitales en este caso utilizando el puerto 12 para enviar una señal indicando cuando comenzar a captar datos además utilizando el puerto 13 para enviar los datos, utilizando las funciones `digitalWrite()` y `digitalRead()` para enviar y recibir cada bit de información la transmisión de datos resulto satisfactoria pero descartamos el uso de esta conexión debido a la dificultad que presentaba la sincronización entre el sistema de generación de información y el sistema de recepción al implementarlo al sistema completo.





**link de tinkercad:** <https://www.tinkercad.com/things/giZ0Pa93G0c>

por ultimo comunicamos a ambos Arduinos utilizando los puertos digitales y la librería `SoftwareSerial.h` que nos permite con el uso de dos puertos digitales imitar mediante software la funcionalidad del protocolo UART en este caso utilizando el puerto digital 6 y el puerto digital 7 como RX y TX alternándose en cada Arduino , dicho sistema nos permite enviar de manera satisfactoria bytes de información al mismo tiempo por el contrario este sistema cuenta con algunas limitaciones, como la imposibilidad de conectar cualquier pin digital de esta forma.

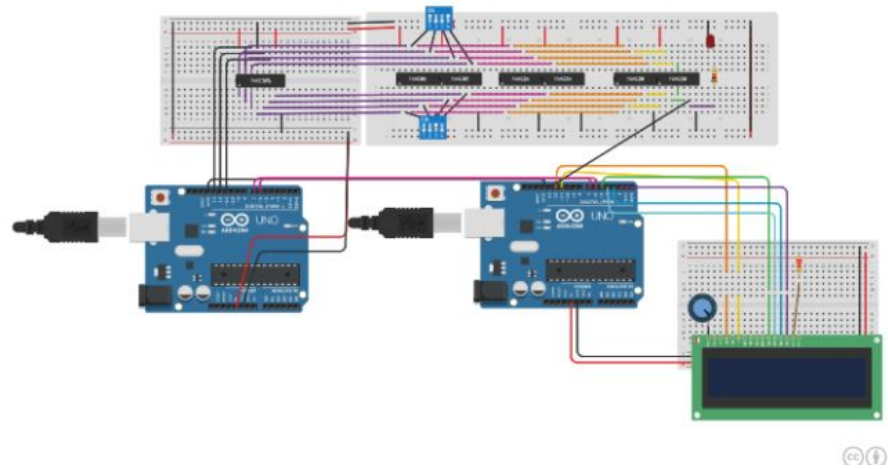


link de tinkercad: <https://www.tinkercad.com/things/jxmiq3WkNGb>

## Capítulo 7

# Resultados

En esta sección se va a consolidar el esquema del sistema concreto con su respectiva dirección URL para hacer las verificaciones respectivas.



link de tinkercad: <https://www.tinkercad.com/things/3KFSm5g8POe>

# Referencias

- Carletti, E. J. (2017, 19 de Julio). *Comunicación-bus i2c*. Publicación de un blog. Descargado de <https://blogthinkbig.com/ventajas-latex-editar-documentos>
- Farestrepoca. (2016, 12 de Mayo). *Programación paralela*. Publicación de un blog. Descargado de [http://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela\\_teoría/index.html#eleven](http://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela_teoría/index.html#eleven)
- INC., D. (2013a, 23 de Enero). *74hc08*. Publicación de un blog. Descargado de <https://www.diodes.com/assets/Datasheets/74HC08.pdf>
- INC., D. (2013b, 21 de Enero). *74hc86*. Publicación de un blog. Descargado de <https://www.diodes.com/assets/Datasheets/74HC86.pdf>
- INC., D. (2013c, 23 de Enero). *74hc86*. Publicación de un blog. Descargado de <https://www.diodes.com/assets/Datasheets/74HC04.pdf>
- intrument, T. (1982, 22 de Diciembre). *Hoja de referencia circuito integrado*. Publicación de un blog. Descargado de [https://www.ti.com/lit/ds/symlink/sn74hc595.pdf?ts=1645444457612&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/sn74hc595.pdf?ts=1645444457612&ref_url=https%253A%252F%252Fwww.google.com%252F)