

# Obligatorio Infraestructura 2020

Francisco Cabanillas (231918) - Paula Hernandez (201886)

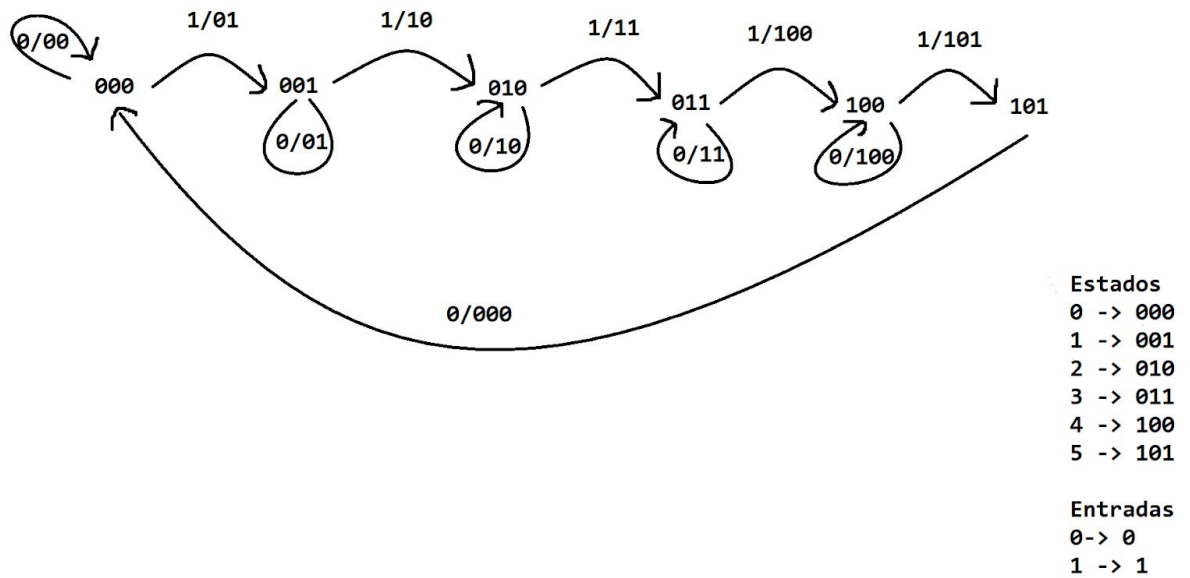
<b>Contador de 0 a 5 asincrónico, de uno en uno, ascendente, circular.</b>	<b>1</b>
<b>CODIFICADOR DE 8 ENTRADAS Y 3 SALIDAS</b>	<b>5</b>
<b>DECODIFICADOR</b>	<b>7</b>
<b>Simulador en JAVA</b>	<b>9</b>
Resultado de la opción 1 (Correr demo)	10
Pasos de la Opción 2	11
Resultado Opcion 2:	13
Resultado Opcion 3:	15
Nota:	15

1. a

A)

Contador de 0 a 5 asincrónico, de uno en uno, ascendente, circular.

Digitalizacion



Tablas

e	q2_n	q1_n	q0_n		q2 n+1	q1 n+1	q0 n+1	s2	s1	s0
0	0	0	0		0	0	0	0	0	0
0	0	0	1		0	0	1	0	0	1
0	0	1	0		0	1	0	0	1	0
0	0	1	1		0	1	1	0	1	1
0	1	0	0		1	0	0	1	0	0
0	1	0	1		1	0	1	1	0	1
0	1	1	0		X	X	X	X	X	X
0	1	1	1		X	X	X	X	X	X
1	0	0	0		0	0	1	0	0	1
1	0	0	1		0	1	0	0	1	0
1	0	1	0		0	1	1	0	1	1
1	0	1	1		1	0	0	1	0	0
1	1	0	0		1	0	1	1	0	1
1	1	0	1		0	0	0	0	0	0
1	1	1	0		X	X	X	X	X	X
1	1	1	1		X	X	X	X	X	X

q2 n+1				
e*q2_n				
q1_n*q0_n	00	01	11	10
00	0	1	1	0
01	0	1	0	0
11	0	X	X	1
10	0	X	X	0

Entrada:

Estado:

Salida:

e

q2q1q0

s2s1s0

q2 n+1

$$q2\ n+1 = (/e \cdot q2\_n) + (q2\_n \cdot /q0\_n) + (e \cdot q1\_n \cdot q0\_n)$$

q1 n+1				
e*q2_n				
q1_n*q0_n	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	X	X	0
10	1	X	X	1

q1 n+1

$$q1\ n+1 = (/e \cdot q1\_n) + (q1\_n \cdot /q0\_n) + (e \cdot /q2\_n \cdot /q1\_n \cdot q0\_n)$$

q0 n+1				
e*q2_n				
q1_n*q0_n	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	X	X	0
10	0	X	X	1

q0 n+1

$$q0\ n+1 = /e \cdot q0\_n + e \cdot /q0\_n$$

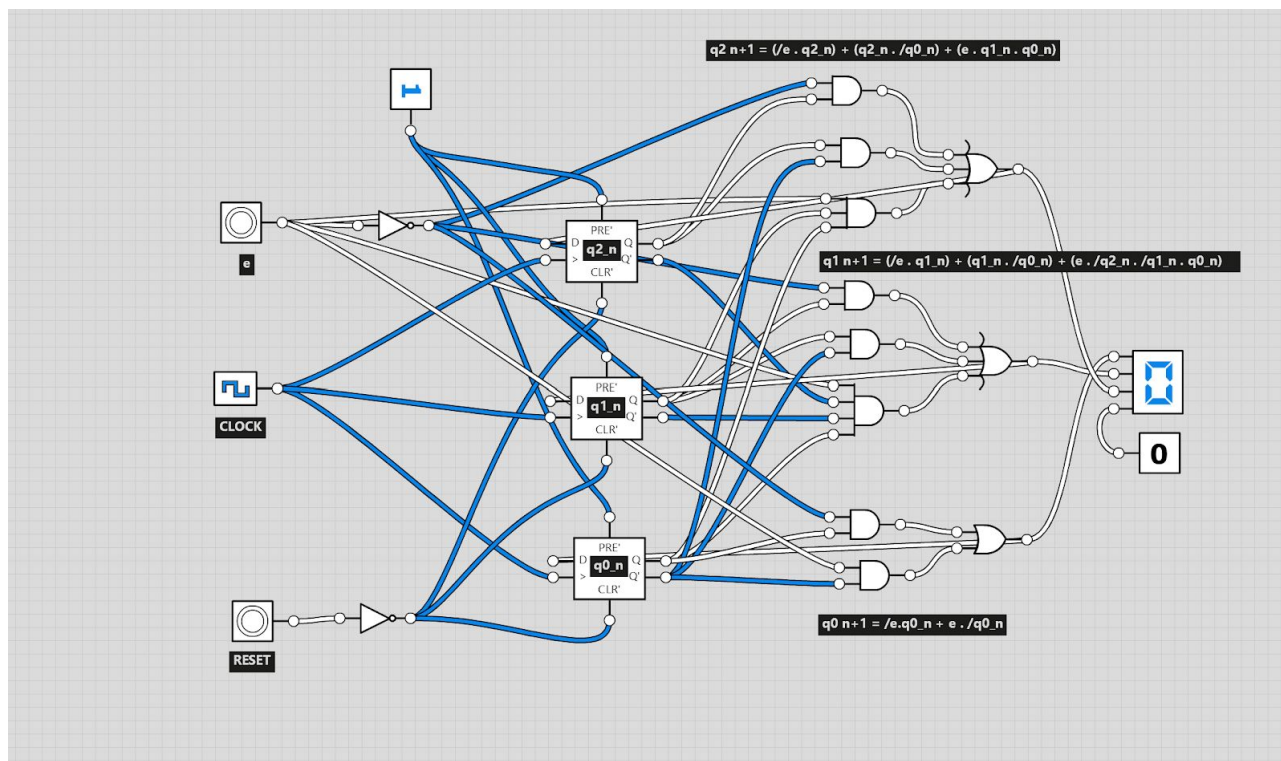
### Fórmulas concluidas

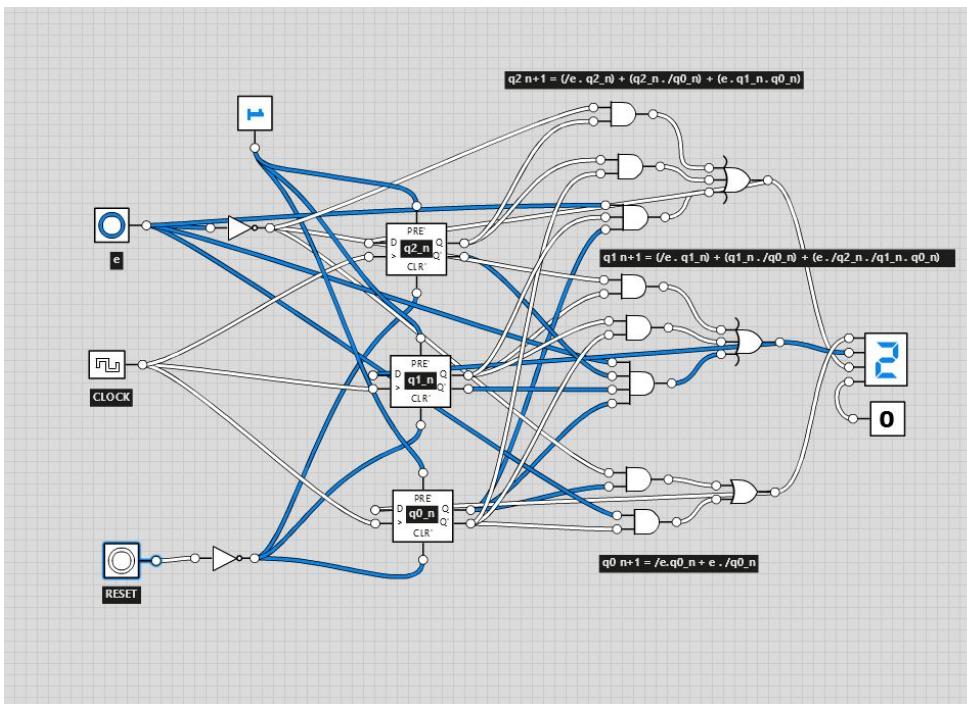
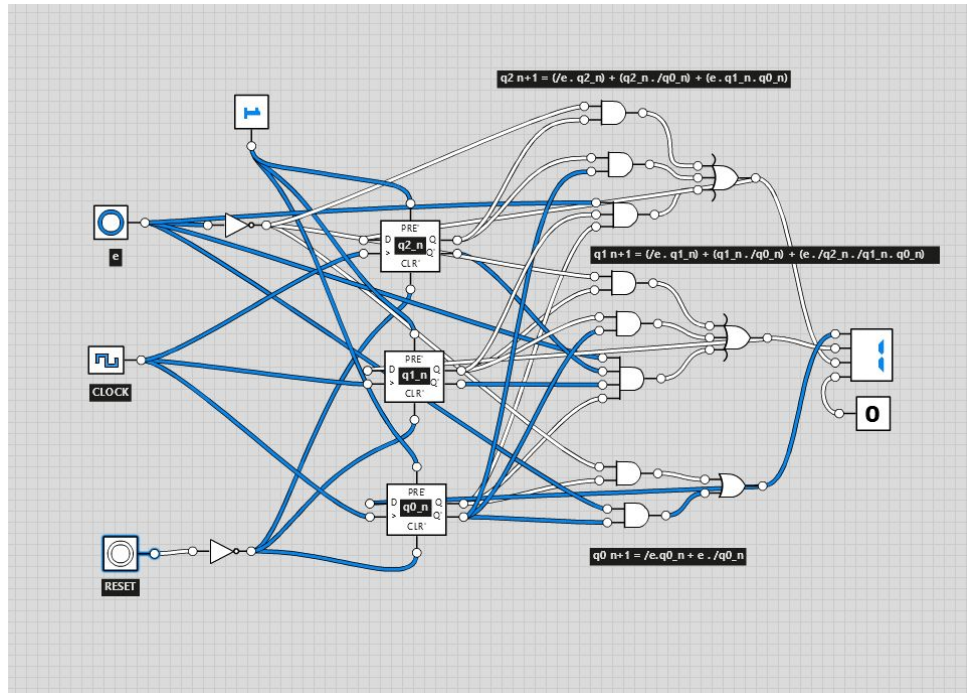
$$q0\ n+1 = /e \cdot q0\_n + e \cdot /q0\_n$$

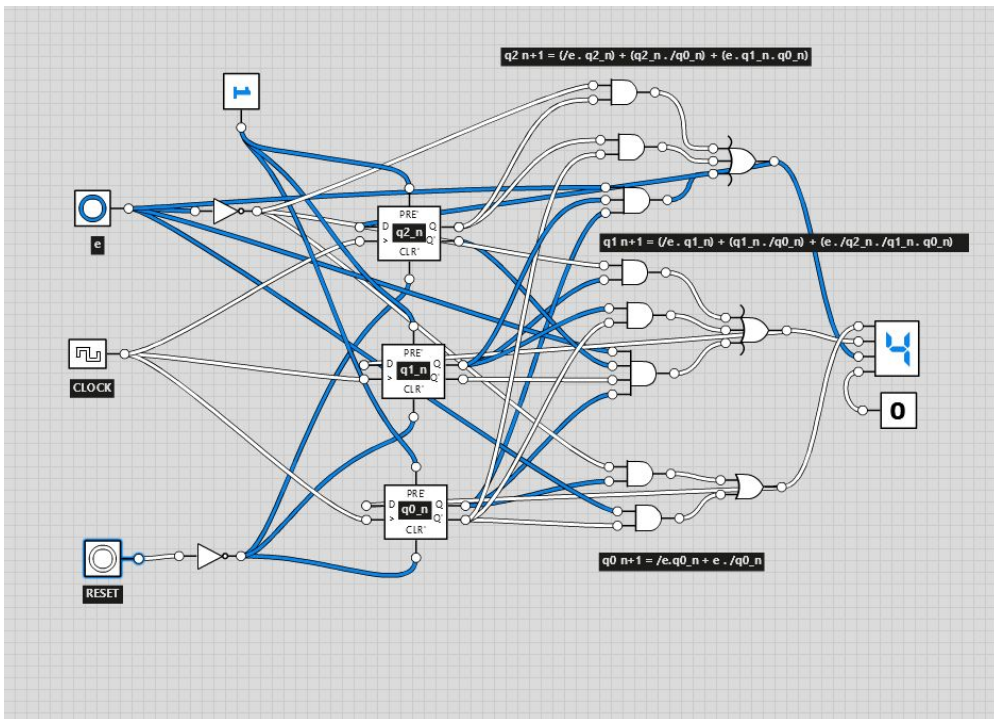
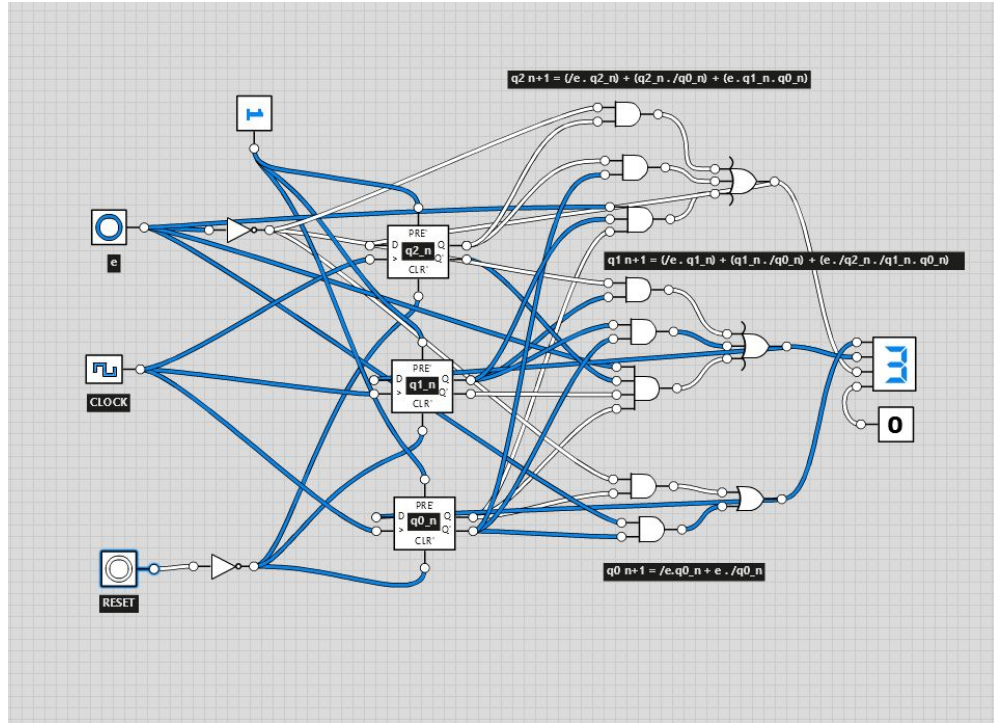
$$q1\ n+1 = (/e \cdot q1\_n) + (q1\_n \cdot /q0\_n) + (e \cdot /q2\_n \cdot /q1\_n \cdot q0\_n)$$

$$q2\ n+1 = (/e \cdot q2\_n) + (q2\_n \cdot /q0\_n) + (e \cdot q1\_n \cdot q0\_n)$$

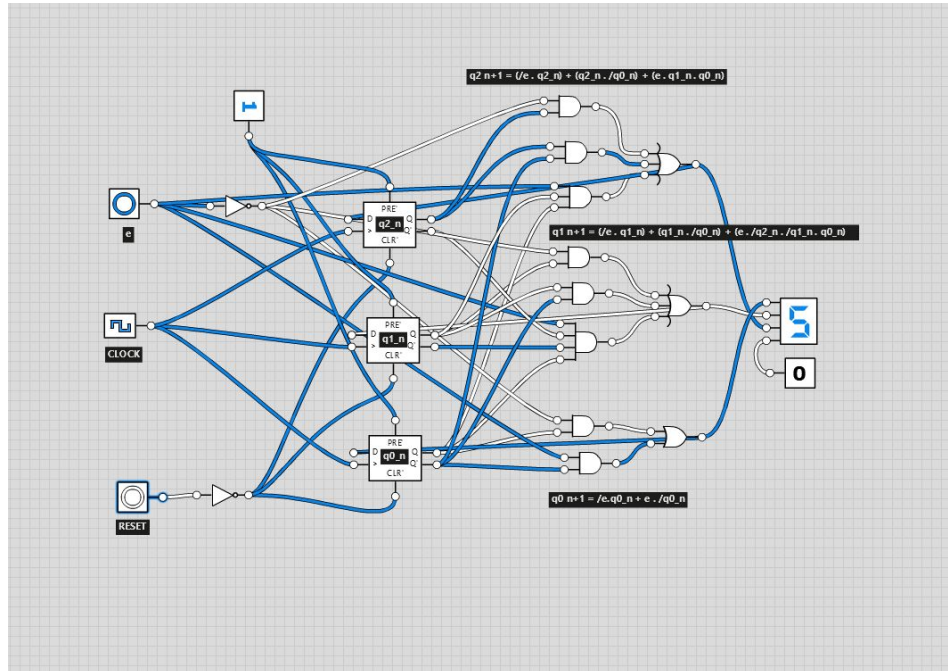
### Circuito Final







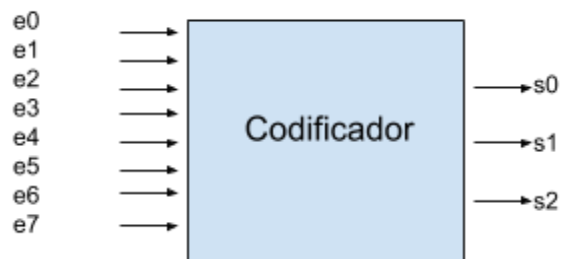




B

## CODIFICADOR DE 8 ENTRADAS Y 3 SALIDAS

Esquema:



El código implícito elegido es:

Entradas/Salida	A2	A1	A0
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E4	1	0	0
E5	1	0	1
E6	1	1	0
E7	1	1	1

Activando un interruptor a la vez:

En la tabla de verdad, descartamos los casos que no sirven mostrando las únicas combinaciones de entradas posibles:

e7	e6	e5	e4	e3	e2	e1	e0	s2	s1	s0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Entonces, por ejemplo, s2 se activa encendiendo e4, e5, e6 o e7..

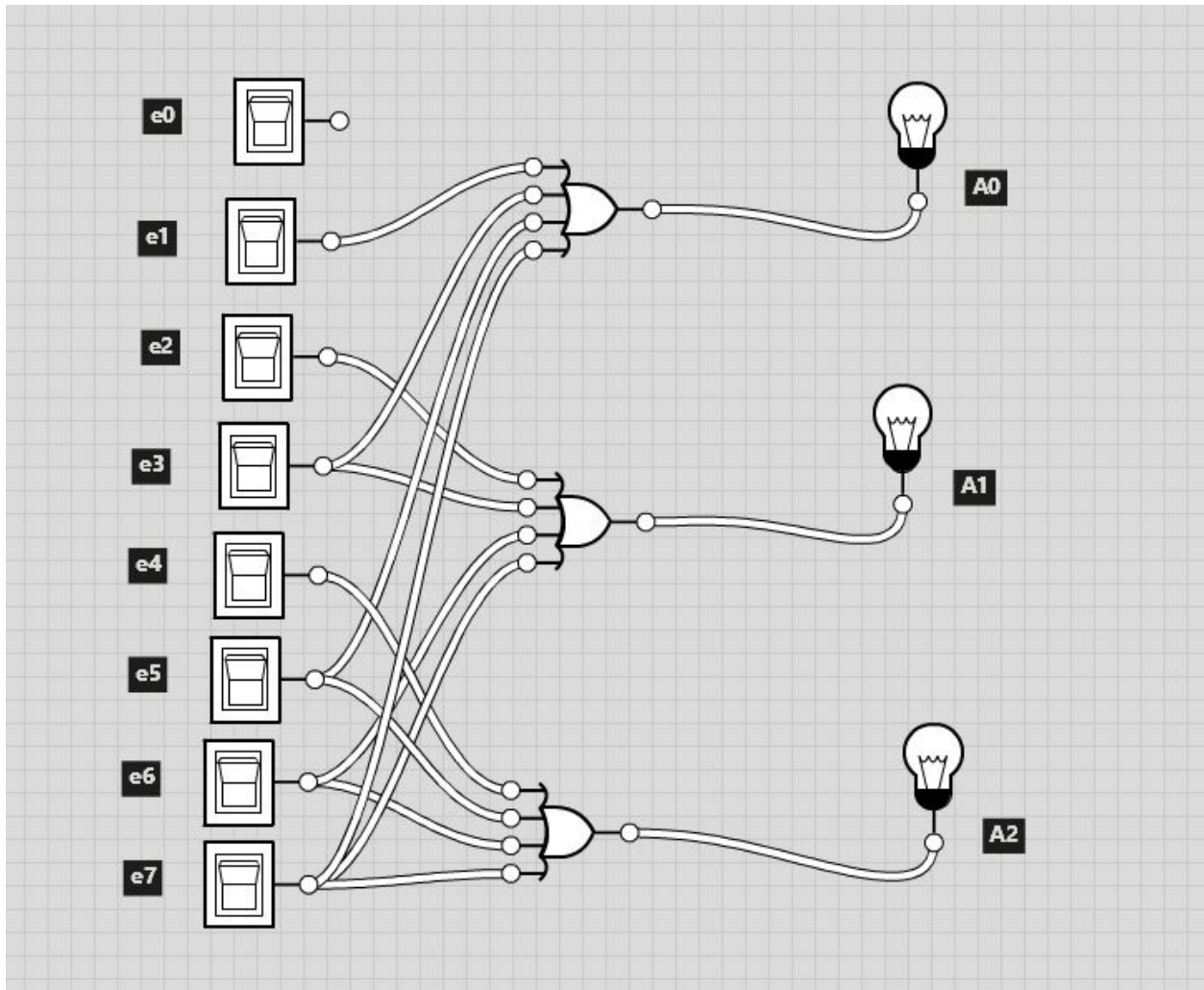
Se deducen las siguientes expresiones lógicas para las salidas:

$$s2 = e4 + e5 + e6 + e7$$

$$s1 = e2 + e3 + e6 + e7$$

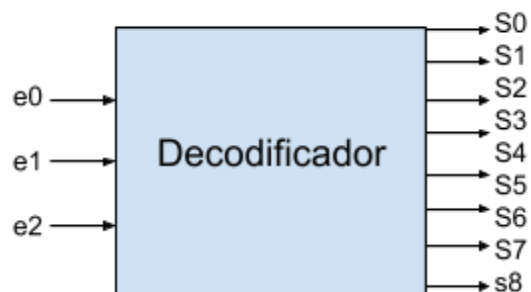
$$s0 = e1 + e3 + e5 + e7$$

### Codificador representado en Logic.ly



### DECODIFICADOR

Mismo código implícito  
3 entradas, 8 salidas  
Esquema:



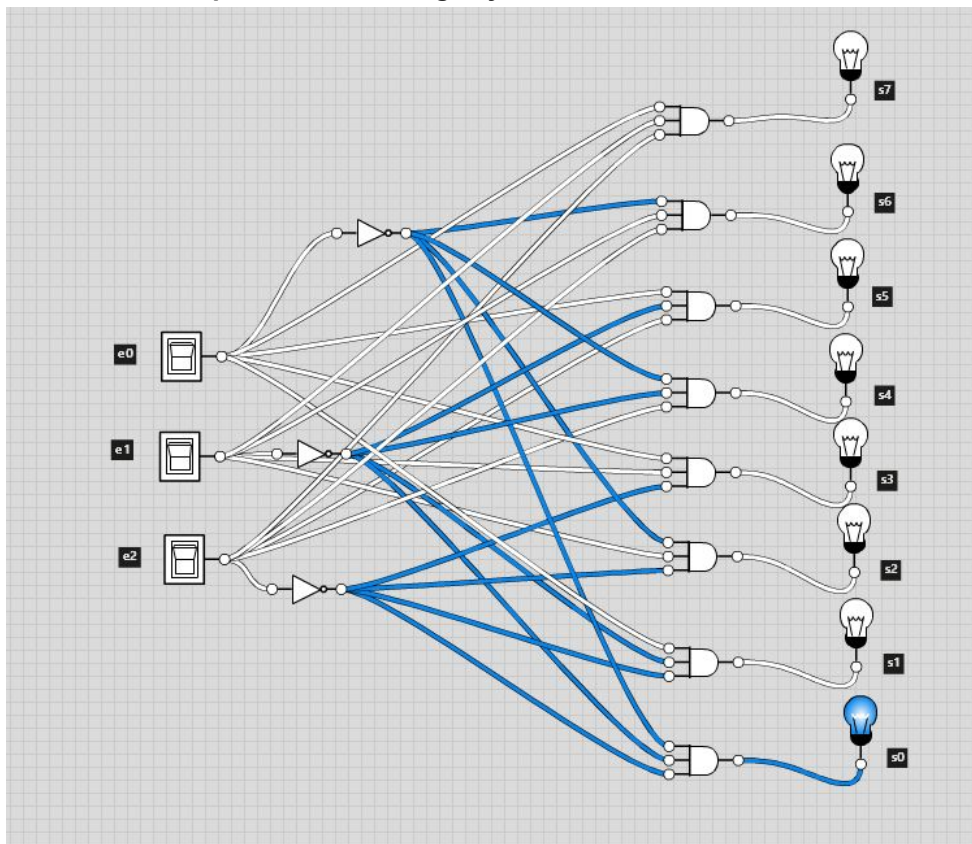


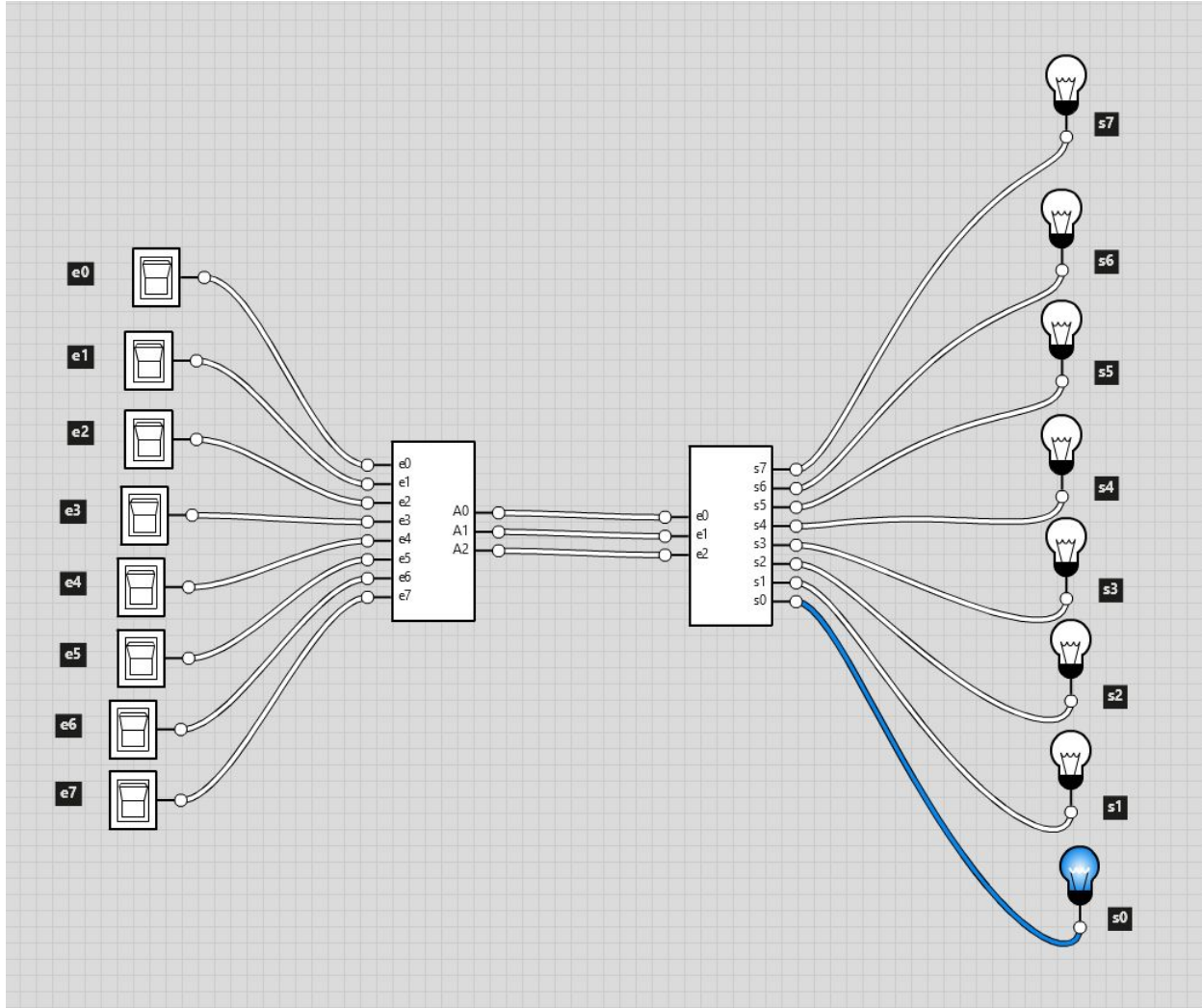
e2	e1	e0	s7	s6	s5	s4	s3	s2	s1	s0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Se deducen las siguientes expresiones lógicas para las salidas, por comprensión utilizando las formas canónicas:

$s7 = e2.e1.e0$   
 $s6 = e2.e1./e0$   
 $s5 = e2./e1.e0$   
 $s4 = e2./e1./e0$   
 $s3 = /e2.e1.e0$   
 $s2 = /e2.e1./e0$   
 $s1 = /e2./e1.e0$   
 $s0 = /e2./e1./e0$

**Decodificador representado en Logic.ly**





2)

## Simulador en JAVA

Para este simulador de reparto de CPU y control de permisos se decidió realizar toda en consola dados los problemas que tuvimos a la hora de querer utilizar la herramienta “NetBeans” (utilizada previamente en Programación 2), por lo tanto utilizamos el IDE “Visual Studio Code” en el cual decidimos hacer un menu principal donde se selecciona 1 de 3 opciones a la vez:

- Correr Demo (Esto va a correr una serie de datos previamente ingresados en el sistema).
- Ingresar 3 recursos, 3 usuarios y el Quantum total para correr en la demo. (En esta opción se le permitirá al usuario ingresar 3 nombres de recursos, 3 nombres de

usuarios, los permisos de estos usuarios sobre dichos recursos y un ciclo máximo por proceso antes de retirarte la CPU y entregarla al siguiente).

- Finalizar (Esta tercera opción es para finalizar la aplicación).

```
---- Simulacion Obligatorio Infraestructura ----

----- Menu -----
Opcion 1: Correr demo
Opcion 2: Ingresar 3 recursos, 3 usuarios y el Quantum total para correr en la demo
Opcion 3: Finalizar
Ingrese un valor (1/2/3) dependiendo su eleccion: █
```

## Resultado de la opción 1 (Correr demo)

```
Ingrese un valor (1/2/3) dependiendo su eleccion: 1
ACCESO DENEGADO: El usuario Pepe solicito usar el programa 1 y no cuenta con los permisos necesarios.
--- Correr programa de proceso 1 ---
ERROR: El proceso 1 intento utilizar el recurso (Impresora) sin solicitarlo previamente.
ERROR: El proceso 1 intento utilizar el recurso (Calculadora) sin solicitarlo previamente.
ACCESO DENEGADO: El proceso 1 del usuario Juan solicitó acceso al recurso Impresora y fue denegado por falta de permisos, se corta la ejecución del proceso.
--- Correr programa de proceso 2 ---
Pedir Pantalla
Usar Pantalla
Devolver Pantalla
Pedir Impresora
Usar Impresora
Devolver Impresora
Proceso 2 perdio CPU por timeOut en posicion 5
```

```
--- Correr programa de proceso 3 ---  
ACCESO DENEGADO: El proceso 3 del usuario Maria solicitó acceso al recurso Calculadora y fue denegado por falta de permisos, se corta la ejecución del proceso.  
--- Correr programa de proceso 5 ---  
Pedir Pantalla  
Usar Pantalla  
Devolver Pantalla  
  
Proceso 5 se ejecuto completamente.  
  
--- Correr programa de proceso 6 ---  
ACCESO DENEGADO: El proceso 6 del usuario Juan solicitó acceso al recurso Calculadora y fue denegado por falta de permisos, se corta la ejecución del proceso.  
--- Correr programa de proceso 2 ---  
Pedir Pantalla  
Usar Pantalla  
Devolver Pantalla  
  
Proceso 2 se ejecuto completamente.  
  
-- El array procesos listos tiene 0 procesos. --  
-- El array procesos finalizados tiene 5 procesos. --  
-- El array procesos bloqueados quedo con 0 procesos. --
```

## Pasos de la Opción 2

```
Ingrese un valor (1/2/3) dependiendo su eleccion: 2  
  
Ingrese el nombre del recurso 1/3: Parlante  
  
El recurso 'Parlante' fue añadido exitosamente.  
  
Ingrese el nombre del recurso 2/3: PuertoUSB  
  
El recurso 'PuertoUSB' fue añadido exitosamente.  
  
Ingrese el nombre del recurso 3/3: Camara  
  
El recurso 'Camara' fue añadido exitosamente.
```

```
Ingrese el nombre del usuario 1/3: Paula  
El usuario 'Paula' fue añadido exitosamente.  
  
Ingrese el nombre del usuario 2/3: Francisco  
El usuario 'Francisco' fue añadido exitosamente.  
  
Ingrese el nombre del usuario 3/3: Angel  
El usuario 'Angel' fue añadido exitosamente.
```

```
Ingrese un valor para representar el quantum (ciclo total por proceso): 9
```

```
Ingrese 1 o 0 para darle permisos al usuario Paula sobre el recurso Parlante: 1  
Ingrese 1 o 0 para darle permisos al usuario Paula sobre el recurso PuertoUSB: 1  
Ingrese 1 o 0 para darle permisos al usuario Paula sobre el recurso Camara: 0  
Ingrese 1 o 0 para darle permisos al usuario Francisco sobre el recurso Parlante: 0  
Ingrese 1 o 0 para darle permisos al usuario Francisco sobre el recurso PuertoUSB: 1  
Ingrese 1 o 0 para darle permisos al usuario Francisco sobre el recurso Camara: 0  
Ingrese 1 o 0 para darle permisos al usuario Angel sobre el recurso Parlante: 1  
Ingrese 1 o 0 para darle permisos al usuario Angel sobre el recurso PuertoUSB: 1  
Ingrese 1 o 0 para darle permisos al usuario Angel sobre el recurso Camara: 1  
  
Permisos agregados exitosamente.
```

Aclaración: Para los permisos de los usuarios sobre los programas se asume que la mejor opción era utilizar los arrays de permiso de la demo dada su complejidad de implementación en consola. Permitiendo únicamente ingresar los permisos de los usuarios sobre los recursos, dado que ambos datos fueron agregados por ellos.



## Resultado Opcion 2:

```
ACCESO DENEGADO: El usuario Francisco solicito usar el programa 1 y no cuenta con los permisos necesarios.
--- Correr programa de proceso 1 ---
ERROR: El proceso 1 intento utilizar el recurso (Parlante) sin solicitarlo previamente.
ERROR: El proceso 1 intento utilizar el recurso (PuertoUSB) sin solicitarlo previamente.
Pedir Parlante

Proceso 1 perdio CPU por timeOut en posicion 2

--- Correr programa de proceso 2 ---
ACCESO DENEGADO: El proceso 2 del usuario Francisco solicito acceso al recurso Camara y fue denegado por falta de permisos, se corta la ejecución del proceso.
--- Correr programa de proceso 3 ---
Pedir PuertoUSB
Usar PuertoUSB
ERROR: El proceso 3 intento utilizar el recurso (Camara) sin solicitarlo previamente.

Proceso 3 perdio CPU por timeOut en posicion 2

--- Correr programa de proceso 5 ---
Pedir Camara
Usar Camara
Devolver Camara
ALERTA: El proceso 5 no pudo completar la tarea Pedir Parlante porque el recurso se encuentra en uso.
--- Correr programa de proceso 6 ---
ERROR: El proceso 6 intento utilizar el recurso (PuertoUSB) sin solicitarlo previamente.
ACCESO DENEGADO: El proceso 6 del usuario Paula solicito acceso al recurso Camara y fue denegado por falta de permisos, se corta la ejecución del proceso.
```

```
--- Correr programa de proceso 1 ---  
  
Usar Parlante  
  
Usar Parlante  
  
Usar Parlante  
  
Proceso 1 perdio CPU por timeOut en posicion 2  
  
--- Correr programa de proceso 3 ---  
  
Devolver PuertoUSB  
  
Pedir Camara  
  
Usar Camara  
  
Devolver Camara  
  
Proceso 3 se ejecuto completamente.  
  
--- Correr programa de proceso 1 ---  
  
Usar Parlante  
  
Devolver Parlante  
  
Pedir PuertoUSB  
  
Usar PuertoUSB  
  
Proceso 1 perdio CPU por timeOut en posicion 3  
  
--- Correr programa de proceso 5 ---  
  
Pedir Camara  
  
Usar Camara  
  
Devolver Camara  
  
Pedir Parlante  
  
Proceso 5 perdio CPU por timeOut en posicion 6
```

```
--- Correr programa de proceso 1 ---  
Devolver PuertoUSB  
  
Proceso 1 se ejecuto completamente.  
  
--- Correr programa de proceso 5 ---  
ERROR: El proceso 5 intento utilizar el recurso (Camara) sin solicitarlo previamente.  
ERROR: El proceso 5 intento utilizar el recurso (Camara) sin solicitarlo previamente.  
  
Proceso 5 se ejecuto completamente.  
  
-- El array procesos listos tiene 0 procesos. --  
-- El array procesos finalizados tiene 5 procesos. --  
-- El array procesos bloqueados quedo con 0 procesos. --
```


### Resultado Opcion 3:

```
Ingrese un valor (1/2/3) dependiendo su eleccion: 3  
La aplicacion fue finalizada.  
PS C:\Users\franc\Desktop\OBL-INF\obligInf> █
```

### Nota:

Para ejecutar la aplicación adjunta en la entrega se recomienda el uso de Visual Studio Code, con las extensiones:

- Java Extension Pack (incluye Debugger for Java, Language Support for Java, Java Test Runner, Maven for Java, project Manager for Java y Visual Studio IntelliCode).



## Java Extension Pack

vscjava.vscode-java-pack Preview

Microsoft | 5,955,805 | ★★★★★ | Repository | License | v0.10.0


Popular extensions for Java development and more.

Disable Uninstall This extension is enabled globally.


This extension is recommended based on the files you recently opened. Ignore Recommendation

[Details](#) [Feature Contributions](#) [Changelog](#)


### Extension Pack (6)




**Language Support for Java(T...**  
Java Linting, Intellisense, format...  
Red Hat




**Debugger for Java**  
A lightweight Java debugger fo...  
Microsoft




**Java Test Runner**  
Run and debug JUnit or TestNG...  
Microsoft



**Maven for Java**  
Manage Maven projects, execut...  
Microsoft



**Project Manager for Java**  
Manage Java projects in Visual ...  
Microsoft



**Visual Studio IntelliCode**  
AI-assisted development  
Microsoft

## Java Extension Pack

Java Extension Pack is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out [Java in VS Code](#) to get started.

### Extensions Included

By installing Java Extension Pack, the following extensions are installed: