

Taller #3: Circuitos Secuenciales

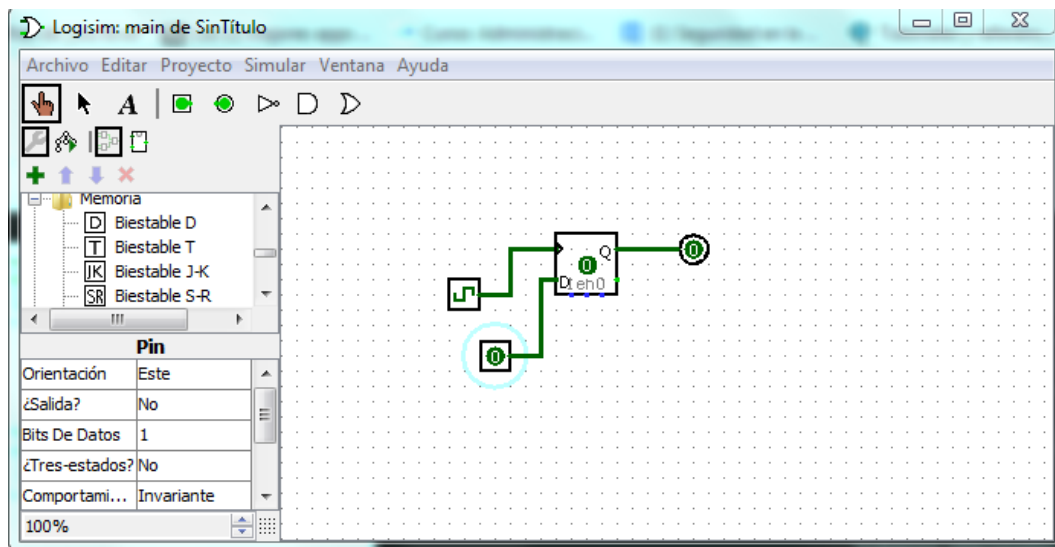
Un circuito secuencial es un circuito digital que implementa una o varias funciones de conmutación, con capacidad de memoria, es decir, la salida depende del estado en el que se encuentre el circuito. Sabiendo que los circuitos secuenciales dependen de un estado previo, necesitamos componentes capaces de almacenar data, estudiaremos en este taller lo que ofrece logisim para realizar este tipo de circuitos.

PARTE I

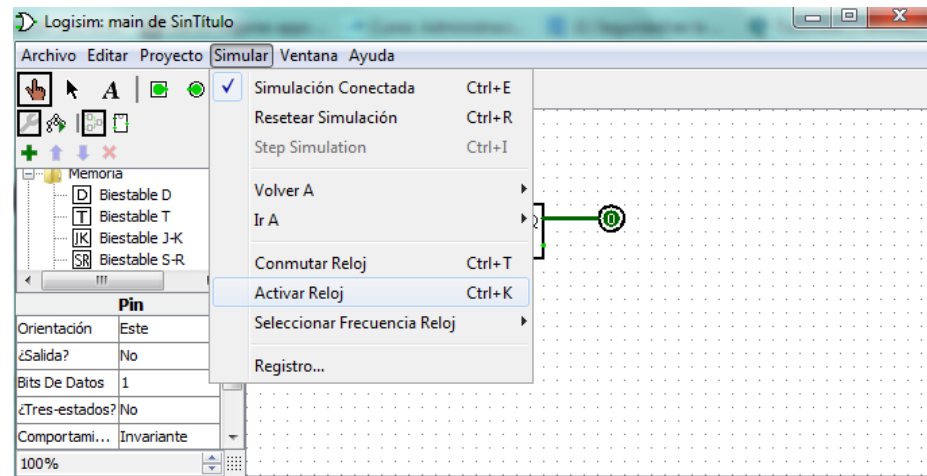
Estudiando los componentes Biestables.

- 1- El primero que estudiaremos es el **biestable D**, este se encuentra en *Memoria -> Biestable D*. Para que funcione al igual que los demás circuitos necesitamos una señal de reloj, que vamos a encontrar en *Wiring -> Reloj*. La entrada del reloj va a ir en el triángulito que se encuentra en el biestable.

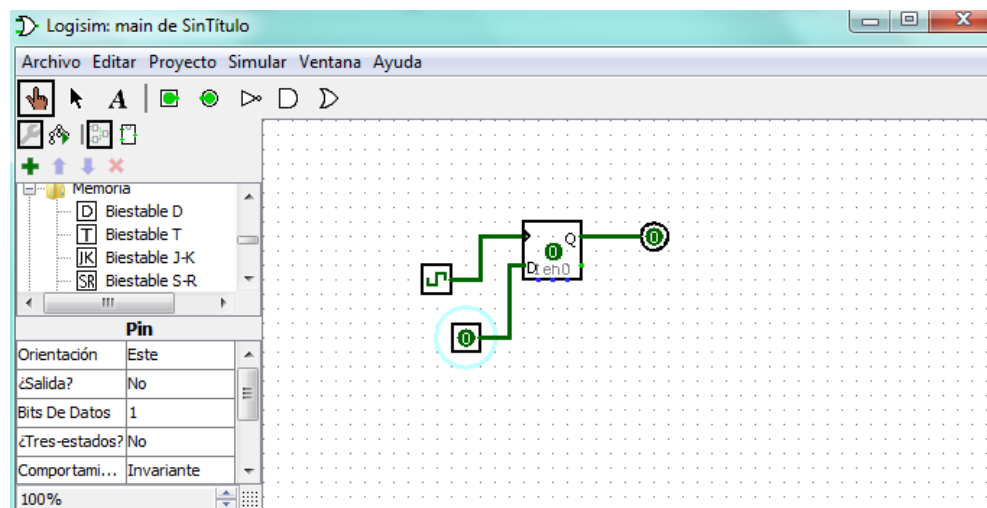
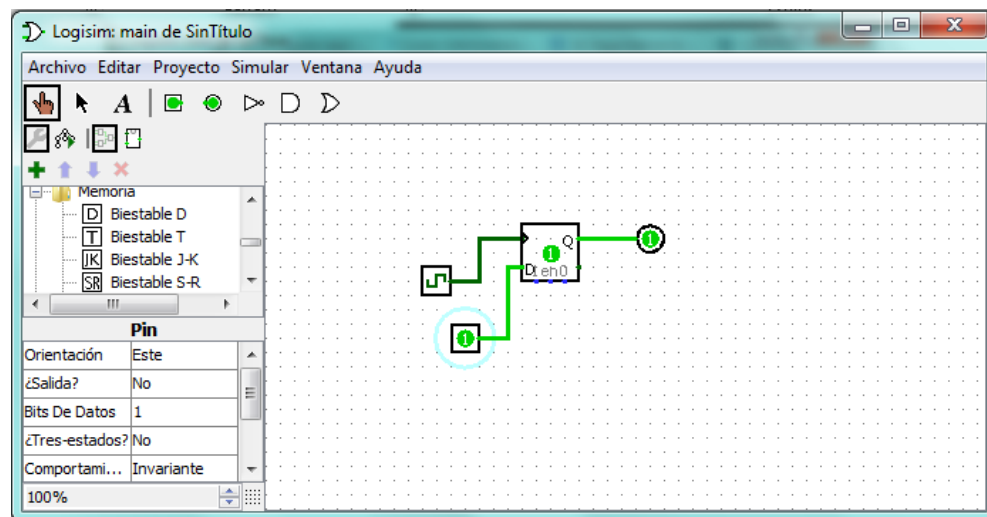
Recordemos que el funcionamiento del D es: cuando el reloj pasa de 0 a 1, el valor que almacena el biestable pasa a ser el valor de la entrada D (Data) en ese preciso instante.



Para activar el reloj vamos a ir al panel superior *Simular -> Activar Reloj* o simplemente haciendo CTRL + K.



Si la entrada D tiene uno, en el momento en el que el reloj pasa a una señal positiva (1), el biestable almacena el valor 1, si D vale 0 entonces se almacena es 0.



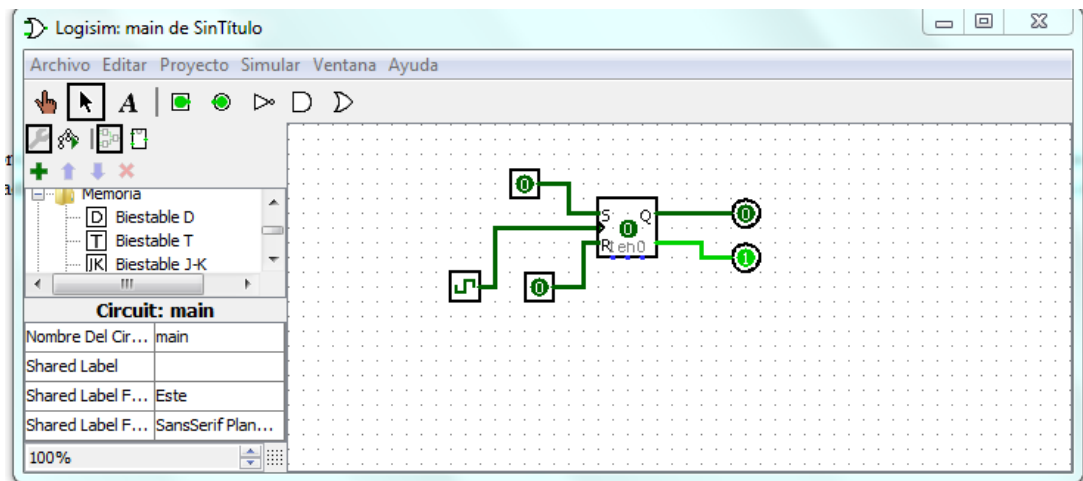
- 2- Para el **biestable SR** lo buscaremos en *Memoria* -> *Biestable SR* y recordemos que su funcionamiento es el siguiente:

Cuando el reloj pasa de 0 a 1, el valor que almacena el biestable permanece inalterado si R y S valen 0, pasa a ser 0 si la entrada R (Reset) vale 1 y pasa a ser 1 si la entrada S (Set) vale 1. El comportamiento no está definido si ambas entradas valen 1. Para este caso en logisim el valor del biestable permanece igual.

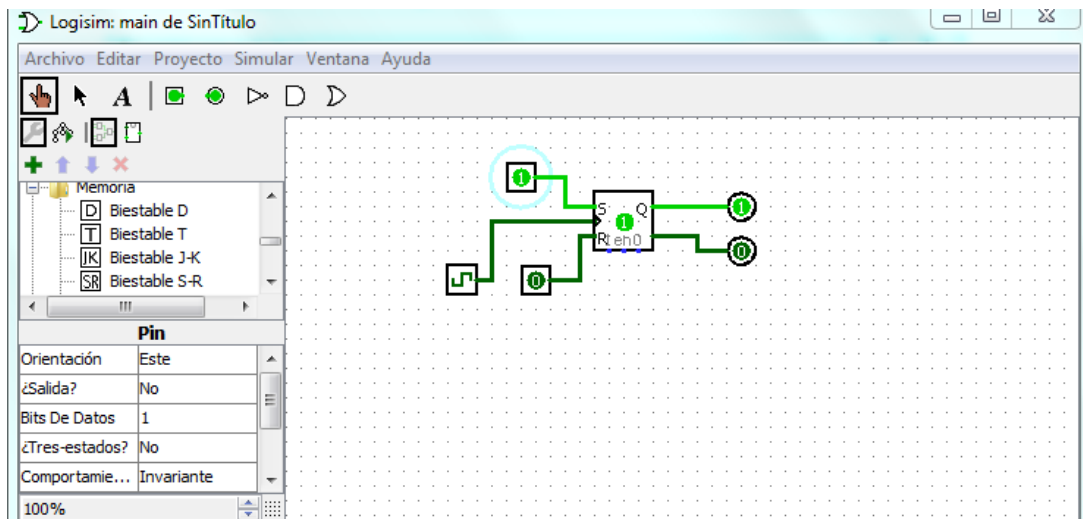
Lo conectamos de la siguiente manera, poniendo la entrada del reloj en el triángulito, las entradas S y R, y las salidas Q y Q complementado.

Para iniciar la simulación hacemos **CTRL + K** y de esta manera activamos el reloj.

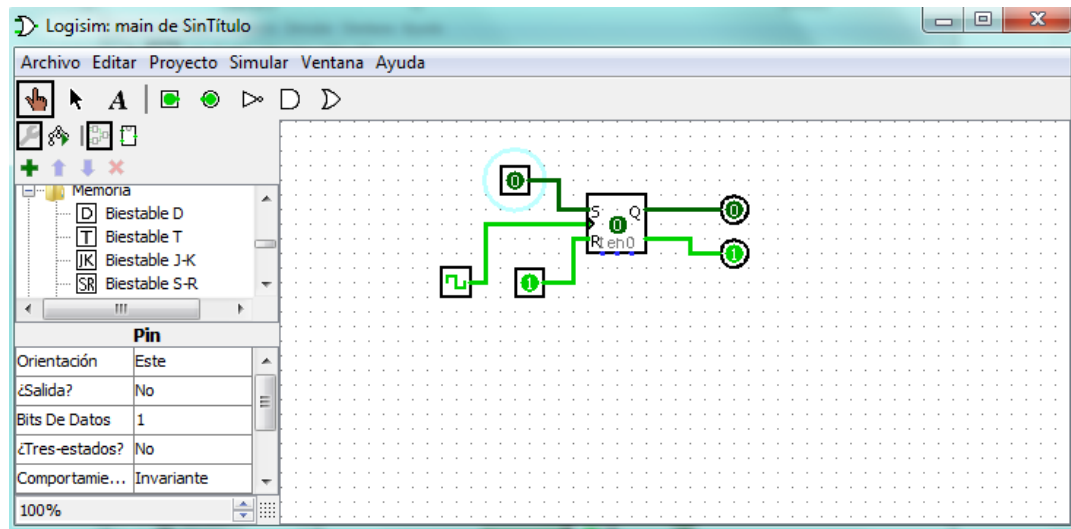
El funcionamiento sería el siguiente si ambas entradas son 0, el biestable SR va a almacenar el valor que tiene:



Si S pasa a 1, va a almacenar 1.



Ahora si S es 0 y R es 1, pasa a 0 el valor almacenado.



PARTE II

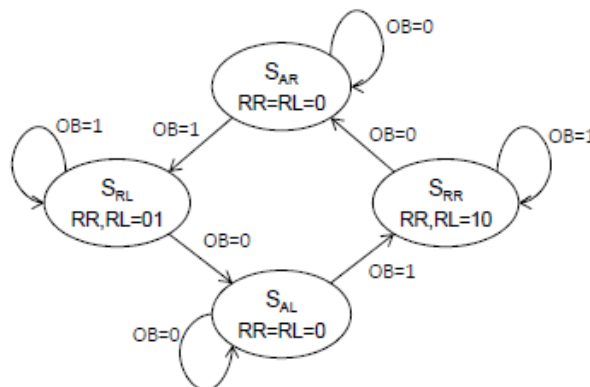
Realizando un circuito secuencial

1- Vamos a suponer que tenemos un robot-escoba que funciona de la siguiente manera:

- El robot tiene un sensor que, cuando detecta un obstáculo delante de él, genera una señal $OB=1$.
- El robot es capaz de avanzar recto o de girar 90° a derecha o izquierda (se para manualmente accionando un interruptor).
- El sistema que queremos describir debe decidir, en función de la entrada OB , si el robot debe avanzar, girar a la derecha o girar a la izquierda según la regla siguiente:

“Cuando el robot detecta un obstáculo va girando a la derecha hasta que deja de detectar dicho obstáculo, momento en el que comienza a avanzar en línea recta. La siguiente vez que detecta un obstáculo, el robot gira en sentido contrario a cómo lo hizo anteriormente, es decir, si antes giró a la derecha ahora girará a la izquierda y viceversa.”

2- Lo primero que haremos es analizar el circuito y construiremos una máquina de moore, que resulta de la siguiente manera:



Dónde SAR es avanzar hacia adelante recordando que la última vez que giro fue a la derecha, SAL es avanzar hacia adelante y la última vez que giro fue a la izquierda, SRR es que se encuentra girando a la derecha y SRL es que está girando hacia la izquierda.

RR es girar a la derecha y RL girar a la izquierda.

- 3- Ahora tomando en cuenta la máquina de moore que construimos podemos realizar la tabla de estados y la tabla de salidas para poder realizar el circuito secuencial.

Tabla de estados:

Estado Actual	Entrada OB	Estado Siguiente
SAR	0	SAR
SAR	1	SRL
SRR	0	SAR
SRR	1	SRR
SAL	0	SAL
SAL	1	SRR
SRL	0	SAL
SRL	1	SRL

Tabla de salidas:

Estado Actual	Salidas: RR RL
SAR	0 0
SRR	1 0
SAL	0 0
SRL	0 1

- 4- Para poder realizar el circuito necesitamos codificar los estados, ya que un circuito implementa 0s y 1s, como tenemos 4 estados podemos codificarlo con 2 bits que llamaremos q0 y q1.

Estado	q1 q0
SAR	0 0
SRR	0 1
SAL	1 0
SRL	1 1

- 5- Ahora lo que haremos será la tabla de estados cambiarla por una tabla de estados binaria, sustituyendo los valores de los estados en la misma, y quedaría de la siguiente manera:

	q1	q0	OB	q1'	q0'
SAR	0	0	0	0	0
SAR	0	0	1	1	1
SRR	0	1	0	0	0
SRR	0	1	1	0	1
SAL	1	0	0	1	0
SAL	1	0	1	0	1
SRL	1	1	0	1	0
SRL	1	1	1	1	1

Y hacemos lo mismo para la tabla de salidas, sustituimos los estados por su valor binario.

	q1	q0	RR	RL
SAR	0	0	0	0
SRR	0	1	1	0
SAL	1	0	0	0
SRL	1	1	0	1

- 6- Ahora hacemos la función booleana asociada a las tablas de estados y de salidas. Y tendríamos las siguientes funciones booleanas para poder diseñar el circuito.

$$q1' = \overline{q1} \cdot \overline{q0} \cdot OB + q1 \cdot \overline{OB} + q1 \cdot q0$$

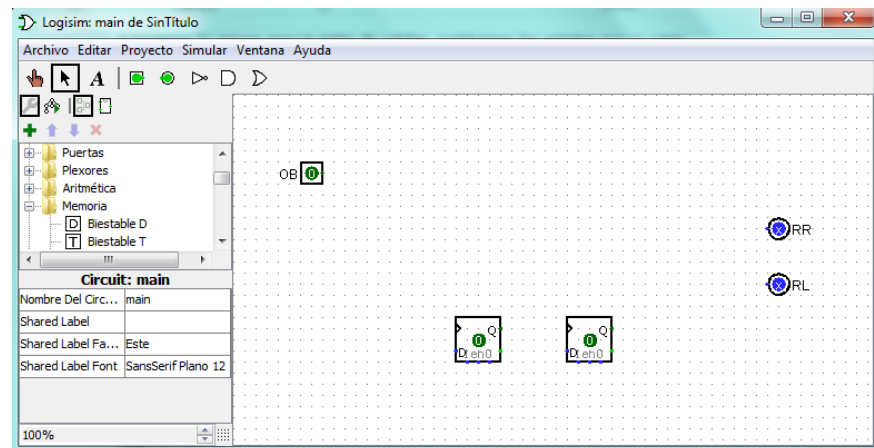
$$q0' = OB$$

$$RR = \overline{q1} \cdot q0$$

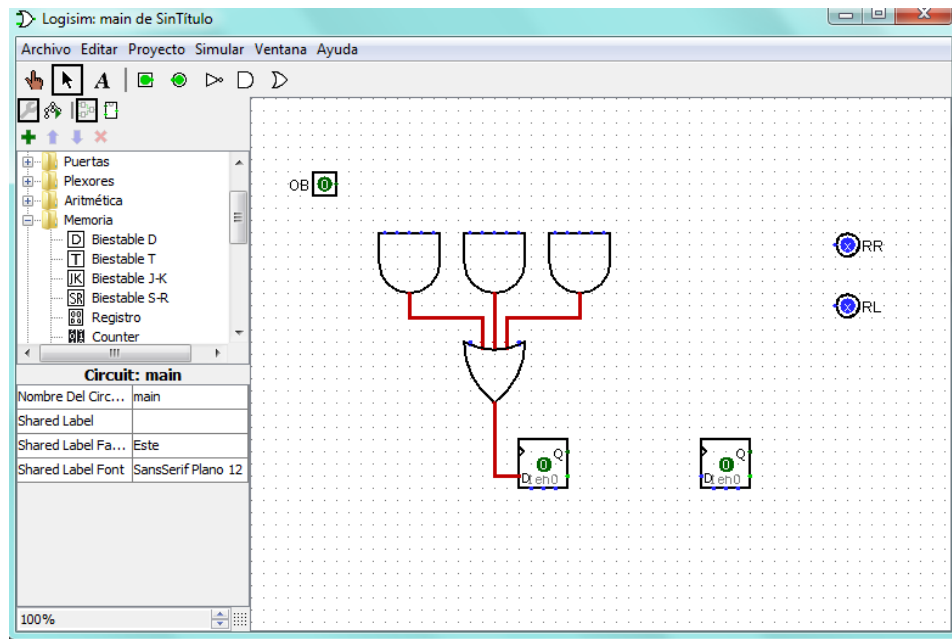
$$RL = q1 \cdot q0$$

Debido a que tenemos 4 estados vamos a necesitar dos componentes biestables para poder simular el comportamiento de q1 y q0, para este caso utilizaremos el componente biestable D, por lo tanto la entrada D del primer biestable será q0', y del segundo será q1'.

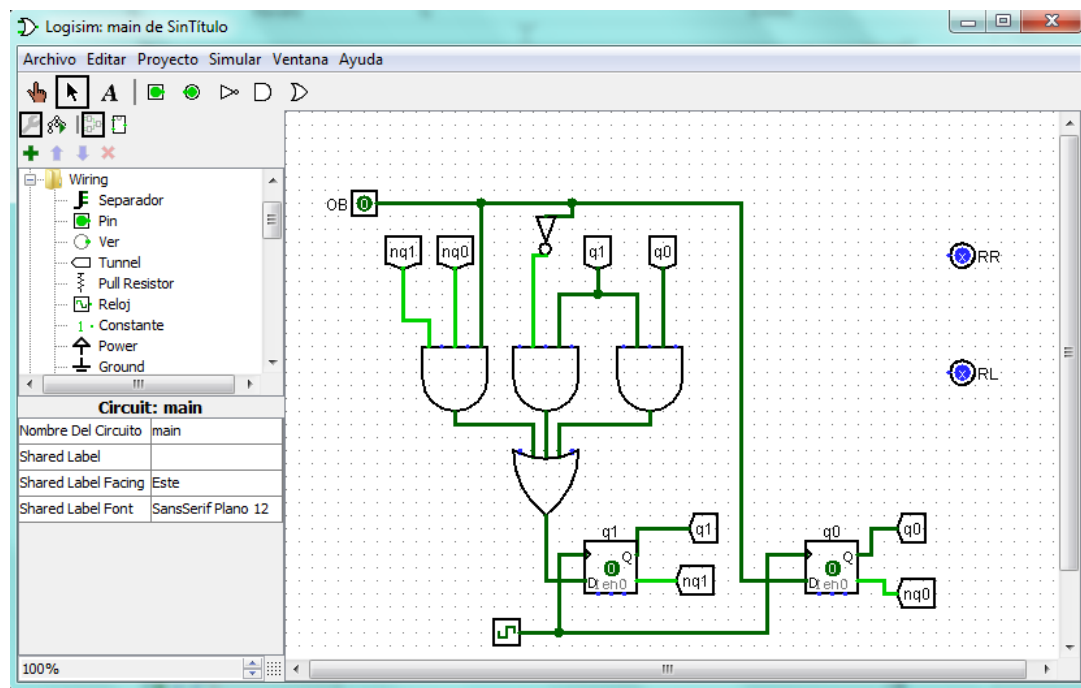
Para realizar este circuito en logisim, vamos a poner los principales componentes que necesitamos, estos son la entrada OB, las salidas RR y RL y los dos biestables D.



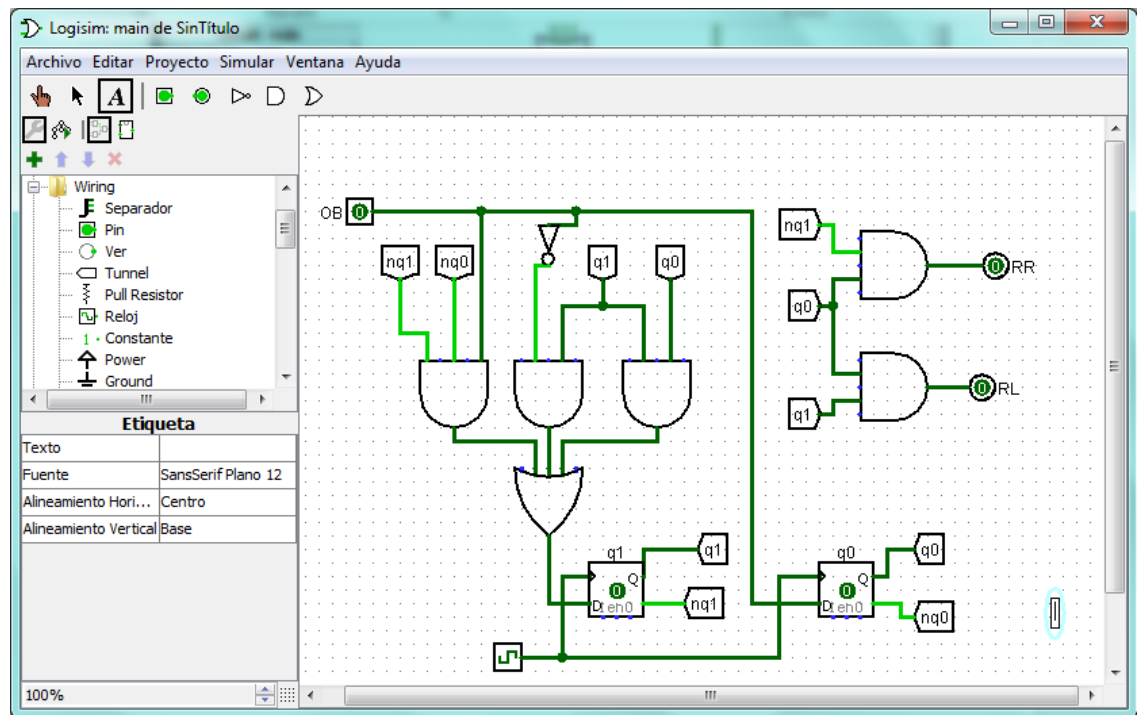
Para el primer biestable tenemos que la entrada D es una suma de tres multiplicaciones por lo tanto, necesitamos tres AND y un OR que es la entrada de D.



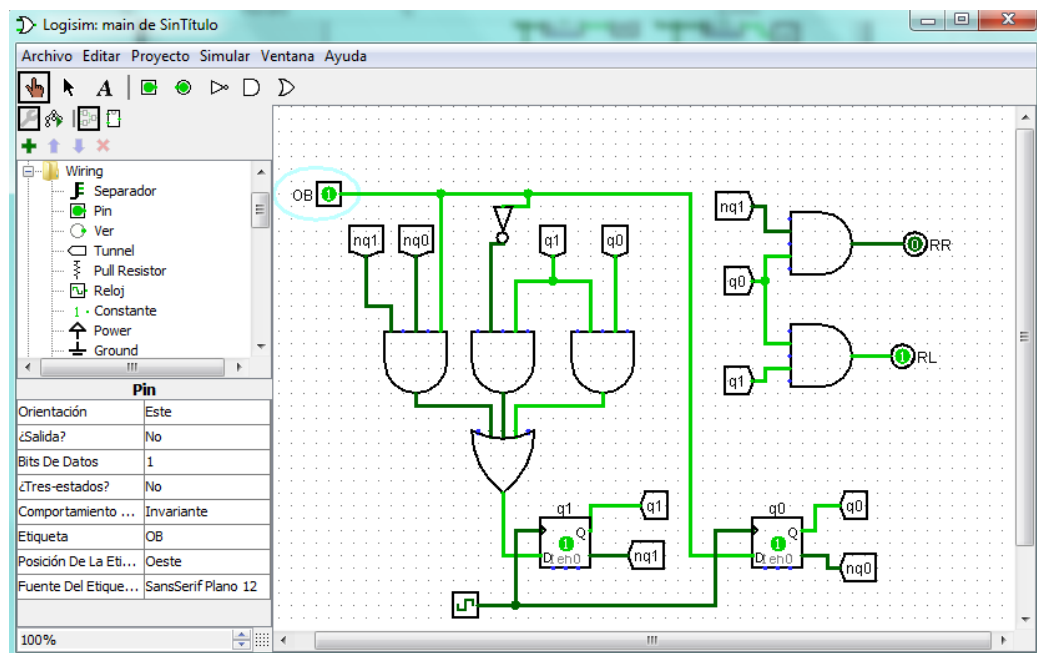
Para trabajar organizados colocaremos los nombres de los biestables para poder identificarlos y utilizaremos túneles para poder tener el circuito limpio y ordenado. Los que tienen de prefijo una n significa que es el complemento, de una vez colocamos la entrada D del q0 que es OB, y colocamos la entrada D del biestable q1 así como se realiza en los circuitos combinacionales.



Para finalizar realizaremos las salidas RR y RL, con sus respectivas entradas q1 y q0, en este caso también hacemos uso de los tuneles para ser organizados y el circuito completo quedaría de la siguiente forma:



Para probarlo accionamos el reloj con CTRL+K y jugamos con la entrada OB, nos podemos dar cuenta que corresponde a la tabla de estados, ya que se encuentra en un principio $q1=q0=0$, al decir que $OB=1$, $q1$ y $q0$ pasan a almacenar 1 y la salida es el estado SRL que corresponde a $RR, RL=0, 1$



Y de la manera anterior podemos probar todos los casos del robot escoba, en la imagen de abajo se ve el cambio cuando está en el estado 1 1 y OB es 0, $q1 = 0$ y $q0 = 0$ es decir es el estado SAL donde RR y RL valen 0.

