



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey
Ingeniería Mecatrónica

Actividad: Final Project PLCs

Automatización Industrial

NOMBRE	MATRÍCULA
Carolina Zertuche Sepúlveda	A01720511
Rodrigo Rodas Barrera	A00827047

30/11/2021

Situación Problema:

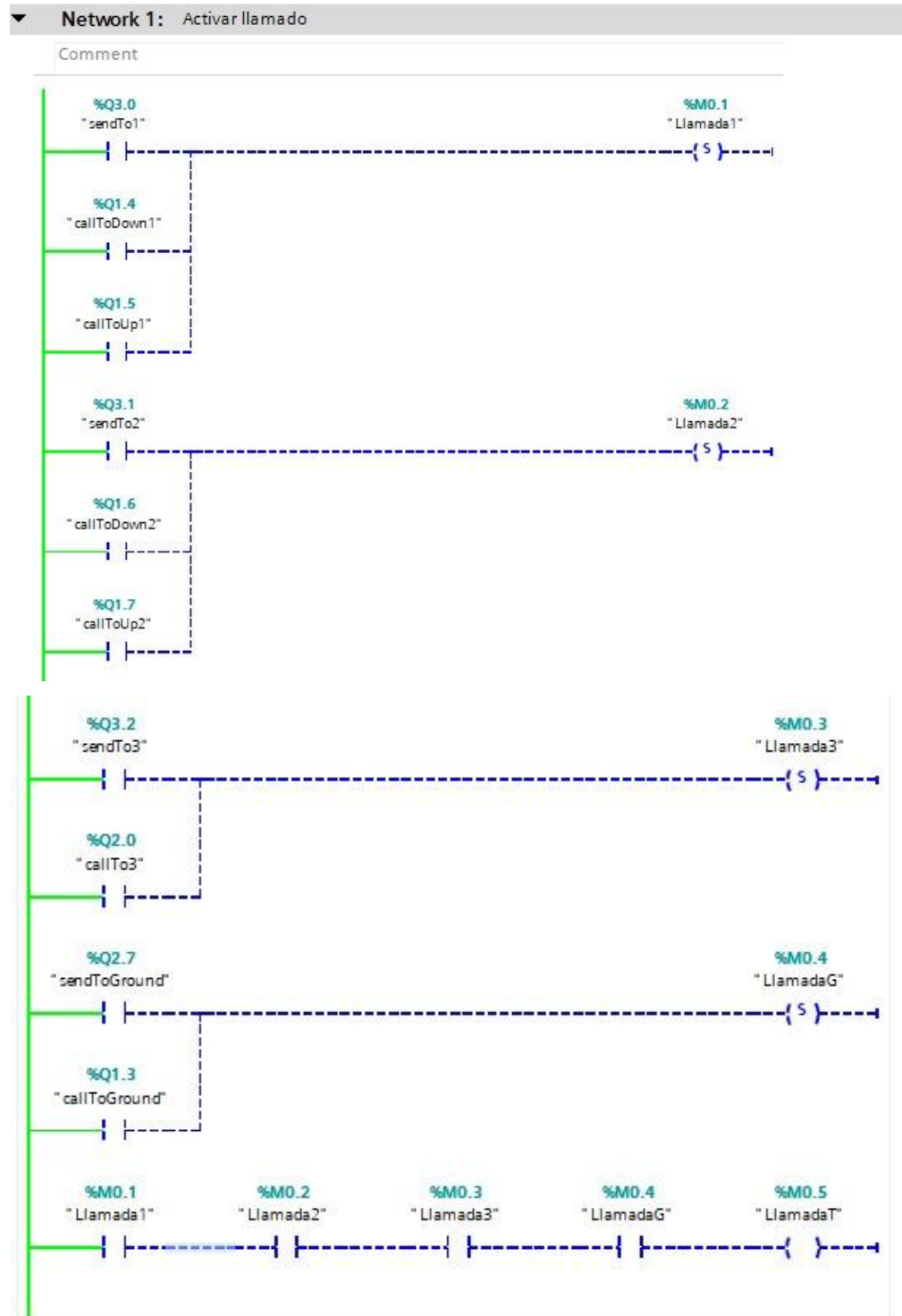
Con nuestro aprendizaje durante todo este periodo de clases se nos encargó realizar la siguiente simulación: Se nos presentó un elevador que consistía en una cabina que se puede mover a través de 1 planta baja y tres pisos. Cada piso contiene una puerta de apertura y botones de llamado del elevador con indicadores integrados. Igualmente el elevador tiene en la parte superior unas lámparas de colores que indican la dirección de movimiento y piso en el que se encuentra la cabina en cualquier momento. Finalmente este sistema cuenta con una caja de control que emula el panel de control interior de un elevador que permite seleccionar a qué piso un usuario quiere ir. Con el apoyo de los sensores, motores eléctricos y pistones que se presentan en este mecanismo se espera que utilizando una máquina virtual podamos hacer que el elevador cumpla con los siguientes requisitos:

- El elevador podrá ser llamado desde cualquier piso por los botones de llamada o los de envío.
- Cuando el elevador sea llamado o enviado, los indicadores de llamado o de envío deben de ser activados y permanecer encendidos hasta que el elevador llegue a su destino.
- Al llegar al piso, las puertas se deben de abrir automáticamente por 3 segundos y luego se deben de cerrar hasta que el elevador no sea enviado o llamado de nuevo.
- Si no existe ninguna petición, la cabina debe de mantenerse en reposo con las puertas cerradas.
- Los indicadores de dirección de movimiento deben indicar el último piso en donde estuvo el elevador hasta llegar a un nuevo piso.
- Igualmente, los indicadores de movimiento deben activarse siempre que esté atendida una posición.
- El elevador debe de moverse de su origen hacia el punto de destino a una velocidad normal pero debe de terminar su desplazamiento a una velocidad reducida.
- Si se activa el botón de emergencia, se debe parar el elevador y las puertas de todos los pisos se deben de abrir; al desactivar el paro el elevador puede volver a ser utilizado.
- Si el elevador se sobrecarga, debe de no moverse hasta que la condición desaparezca.

Con esto, podemos empezar la explicación de nuestro código de escalera realizado para el cumplimiento de este proyecto analizando cada network establecida:

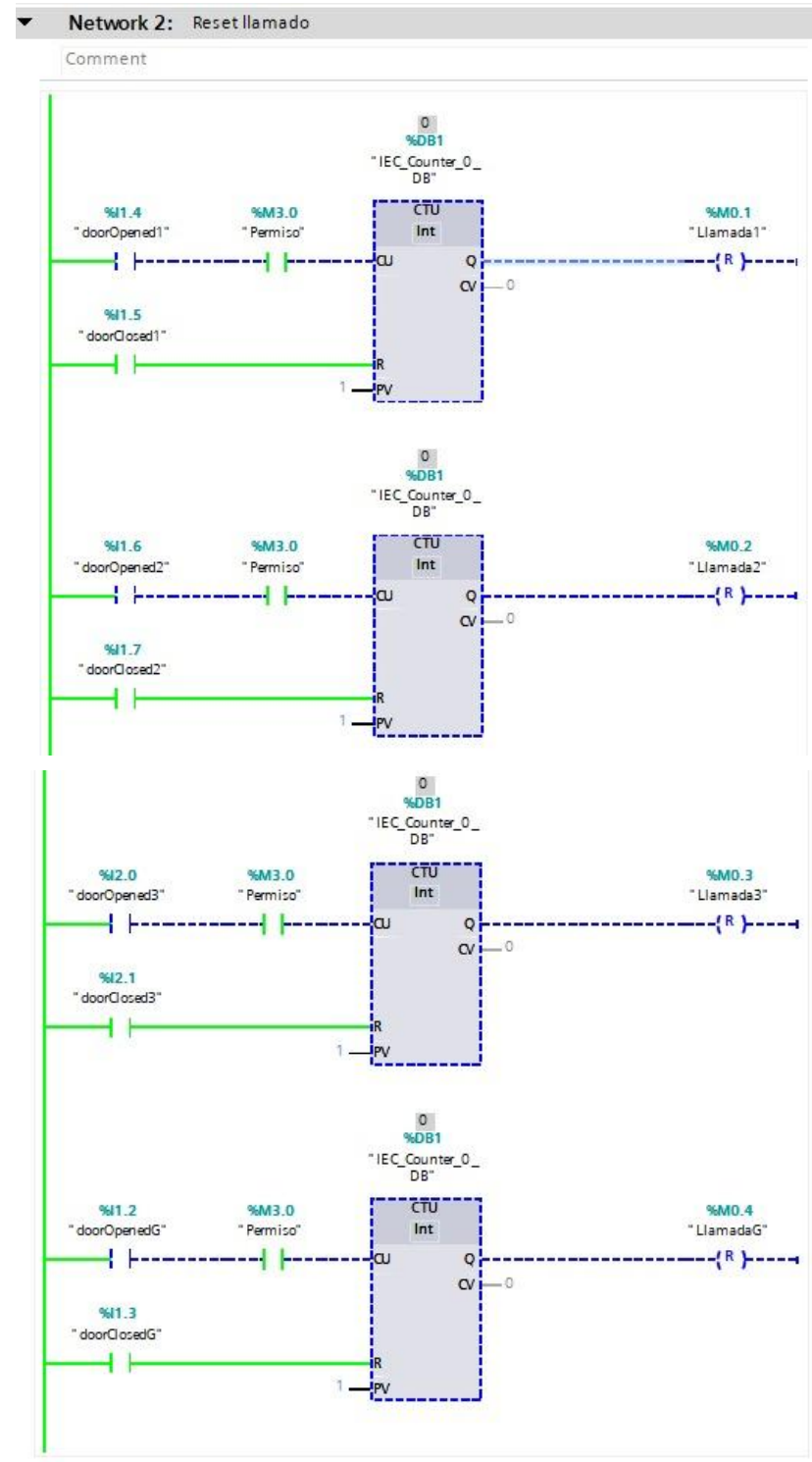
El **primer network** de nuestro sistema se encarga de realizar la llamada de la cabina del elevador ya sea por el panel interior o las flechas al lado de cada puerta. En sí el código está compuesto de un sistema de OR para cada piso, por ejemplo para el primer piso si se selecciona el botón de piso 1 o la flecha para bajar o la flecha para subir en ese piso, esta selección se guardará y activará su memoria de llamada designada que creamos. Esto es igual para cada piso, solo con la diferencia del piso de planta baja y el 3er piso presentan una flecha menos que los demás ya que al estar en estos pisos o no puedes bajar un piso más o no puedes subir un piso más. Este network finaliza con una línea donde presenta que si se

activan todas las memorias de llamada que establecimos, se guardará en una memoria que designamos como Llamada T que es la suma de todas las llamadas.



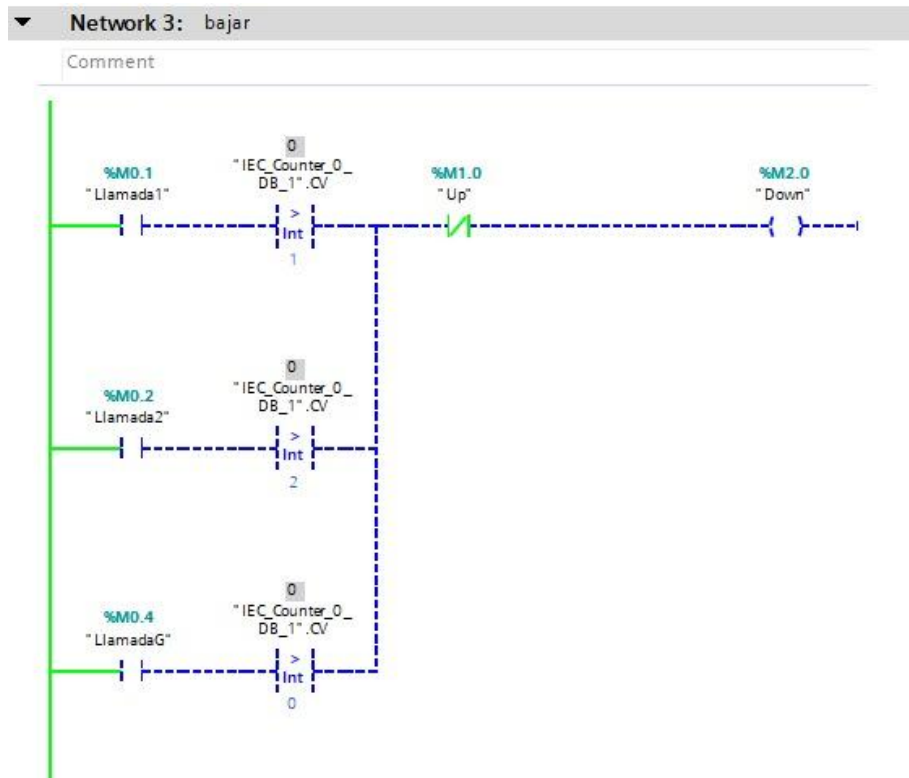
El **segundo network** se encarga de reiniciar el llamado que en el pasado network fue activado. Este sistema está compuesto de contadores para arriba, cada línea designada para una de las 4 llamadas. El código funciona que cuando el sensor que detecta que la puerta está abierta es activado e igual el permiso es activado el contador se irá de 0 a 1 y como 1 es

nuestro valor PV máximo, se dejará pasar la señal y la memoria de llamada será reiniciada. Cuando el sensor de la puerta cerrada es activado, el contador se va a reiniciar y se podrá hacer una llamada en ese piso otra vez.

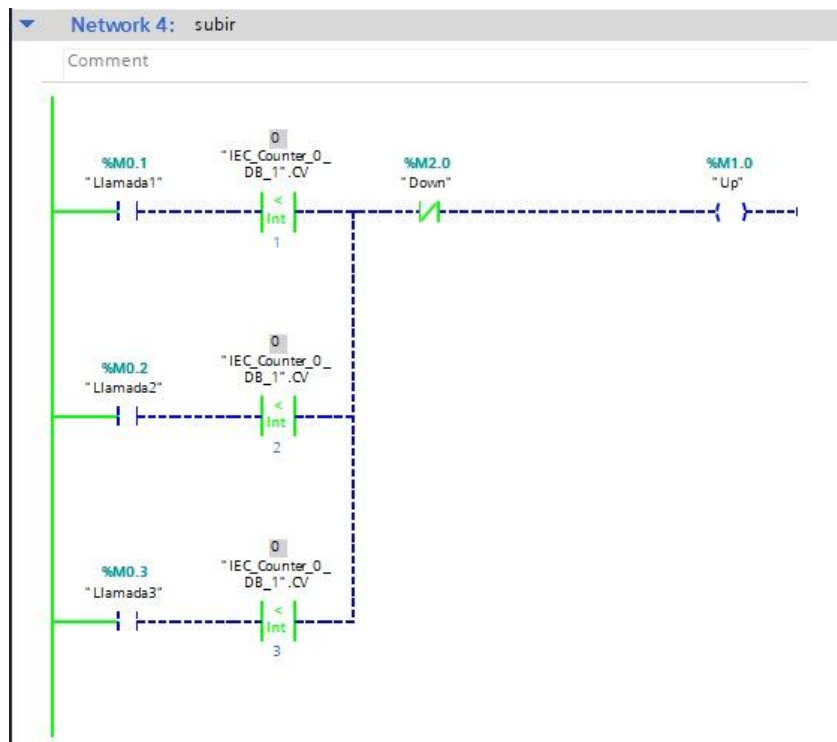


El **tercer network** ahora se encarga de mover la cabina del elevador hacia abajo; se presenta un sistema OR en donde se señala que si se activa la llamada 1, 2 o ground y cumple con la

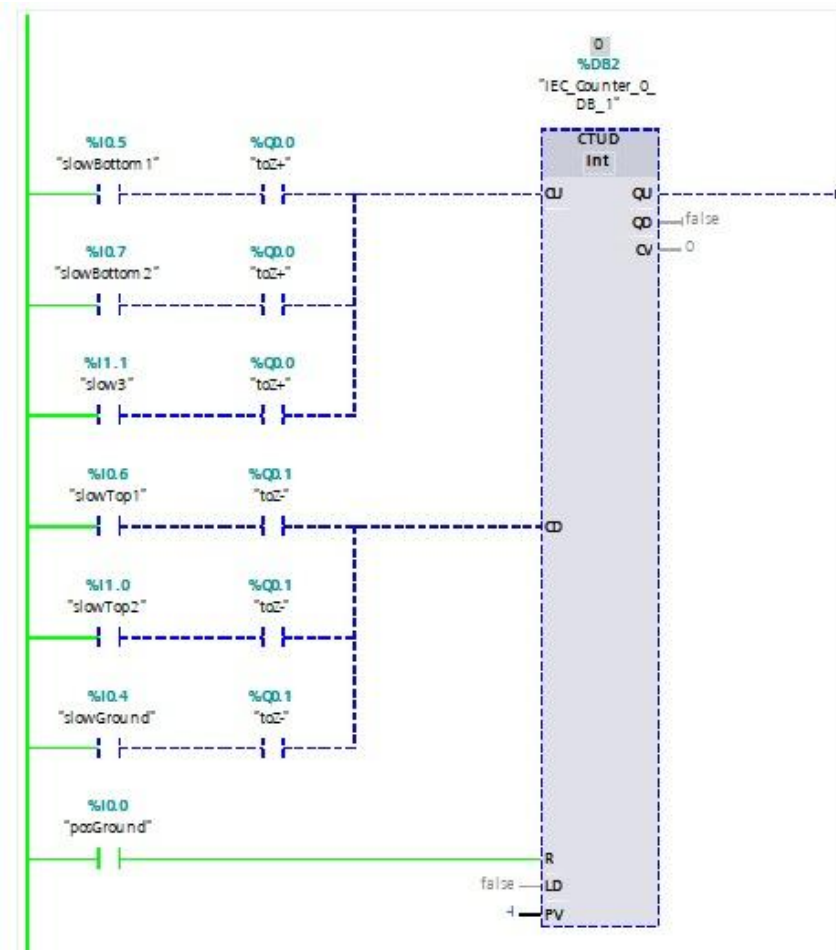
condición establecida aquí de el valor siendo más grande que el número presentado y en un counter que veremos en el quinto network, mientras que no se active una memoria que titulamos up, se mandará esta señal hacia una memoria llamada down; esto nos dejará poder bajar de cualquier piso arriba hacia uno de estos pisos.



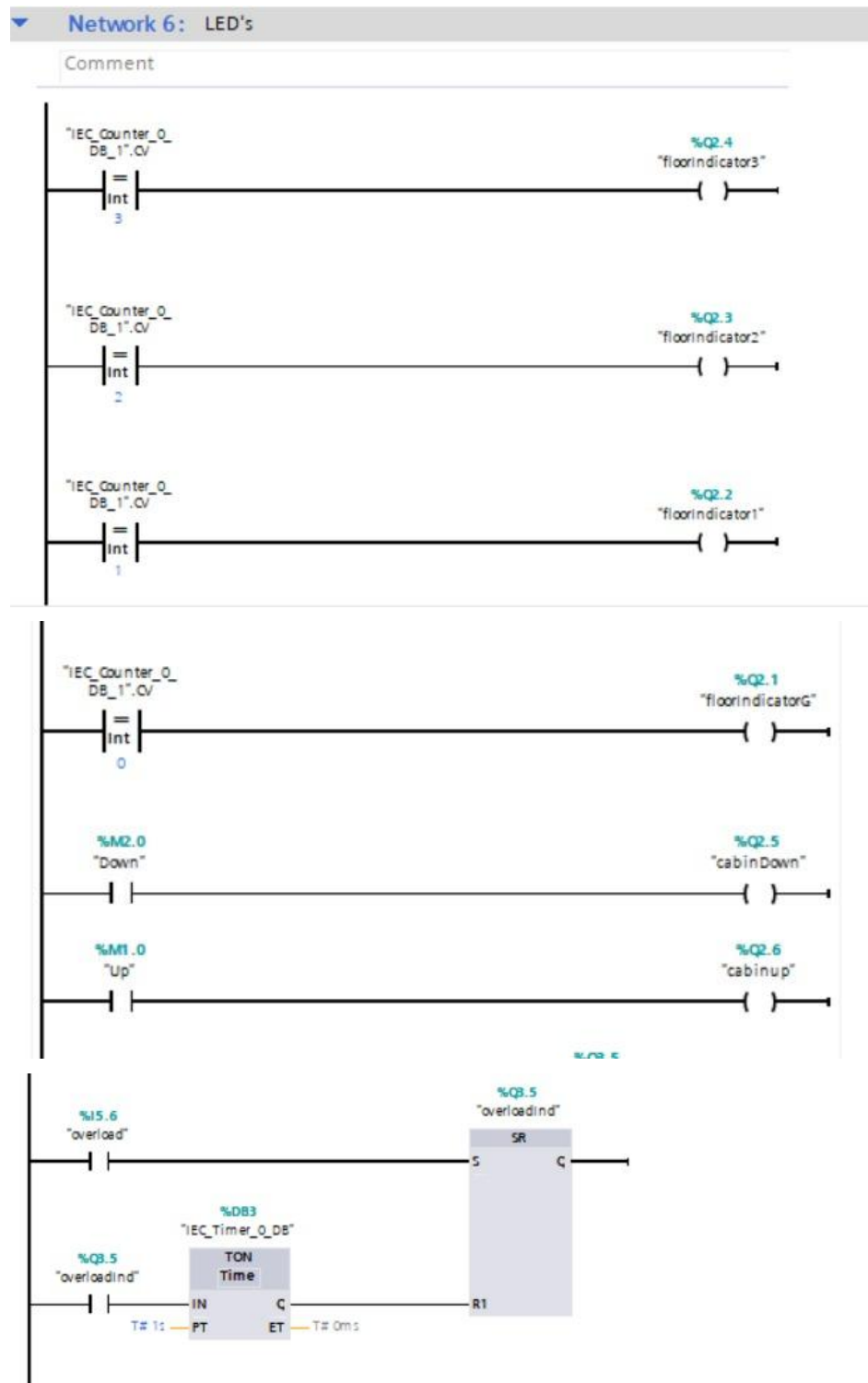
Este **cuarto network** hace lo mismo que en el anterior excepto que ahora lo hace para registrar el sistema cuando va a subir; si se activa la llamada 1, 2 o 3 (no se incluye ground por que no puedes subir a ground desde otro piso), y cumple con las condiciones del comparador establecidas aquí y en la base del quinto network, mientras que la memoria down no se encuentre activa, se registrará la señal en nuestra memoria Up.



El **quinto network** es donde nos encontramos con el counter que participa en los 2 networks anteriores de subida y bajada. Este counter es uno tipo up and down, en donde va a contar hacia arriba cuando se active alguna de las líneas de nuestro sistema OR; si se encuentra activo uno de los sensores mecánicos de slowBottom y se activa el motor de movimiento hacia arriba, se sumará un recuento al contador. Cuando el contador detecte que uno de los sensores mecánicos de slowTop se encuentra activo y el motor de movimiento hacia abajo es activado, se restará un valor del contador. Este contador sólo puede mantener un valor entre 0 a 4; cuando el elevador llega al último piso, el contador se va a reiniciar. En sí, este network es la razón por la que en el network 3 y 4 se presenta el timer y una función de comparación hacia cierto número por que al valor ser menor o mayor al número establecido, este podrá moverse a dicho piso por que o no se encuentra en ese piso ya o el sistema ya le estará diciendo que en base al número este va a necesitar moverse para arriba o hacia abajo para llegar a ese nuevo piso.



El **sexto network** trabaja con los indicadores de nuestro sistema; en base al counter del network anterior, si este es igual al número establecido, se encenderá el LED que cumpla con la condición. Las primeras 4 líneas son las que controlan los indicadores de piso en la parte superior; al cambiar de piso, el número de el contador va a cambiar entonces la luz también va a cambiar. Las siguientes 2 líneas controlan los LEDs que señalan si el elevador va a subir hacia arriba o hacia abajo que también se encuentran en la parte superior del elevador. Cuando la memoria de Up o la memoria de Down, las que se establecieron en los networks anteriores, se activará la luz designada. Finalmente terminamos con el sistema de overload en donde si el botón de overload es activado, el LED de overload será activado y solo se reiniciará después de 1 segundo.

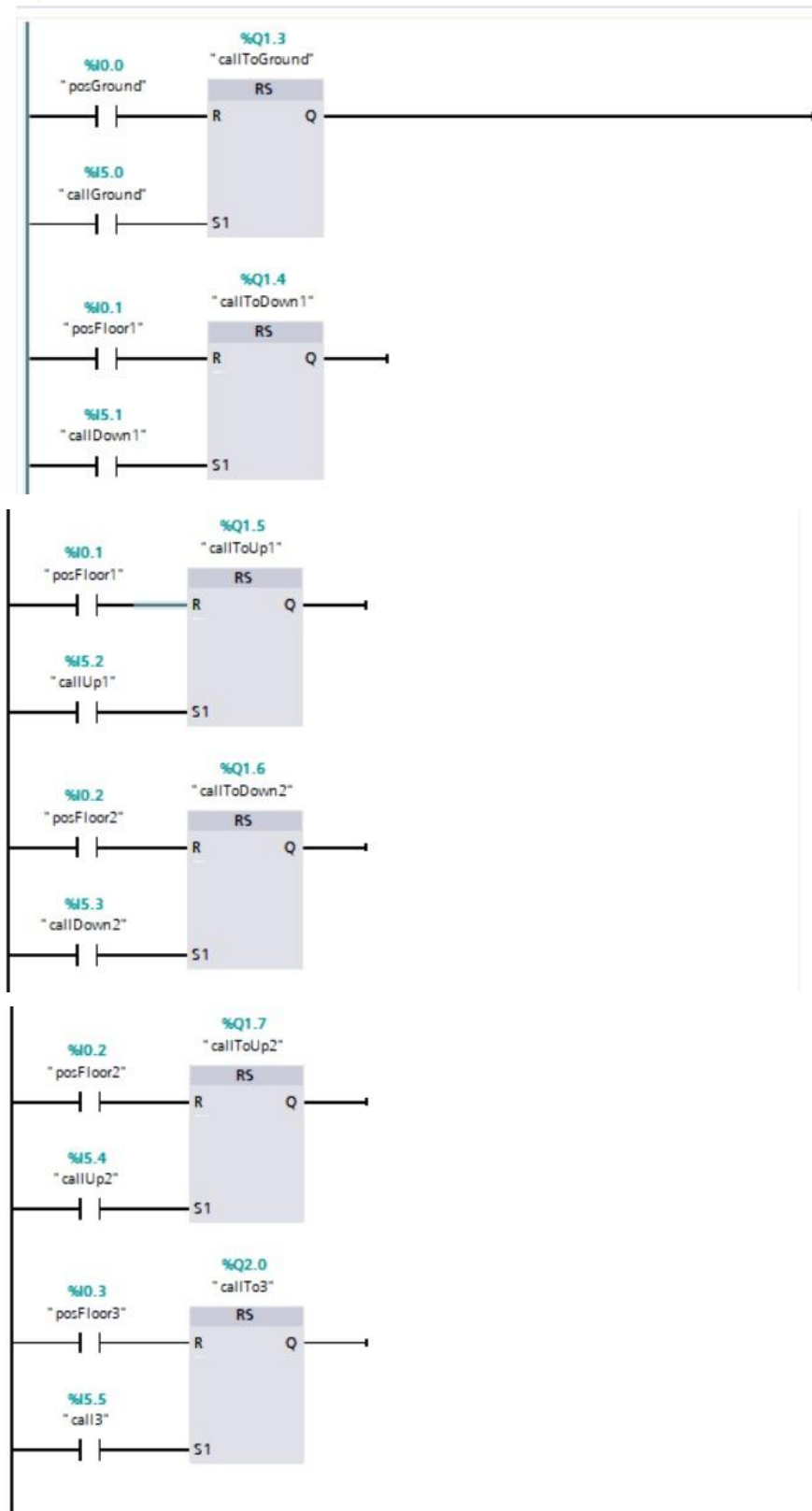


El **séptimo network** trabaja con los botones en el exterior del elevador; las flechas de llamada hacia arriba o abajo. En si las siguientes líneas repiten el mismo comando por lo que sólo analizaremos 2 para explicar el funcionamiento. La línea 2 y 3 hacen lo siguiente: cuando se presiona el botón de la flecha hacia abajo en el primer piso, se va a prender el LED callToDown1; este solo se podrá reiniciar cuando llegue el elevador a este primer piso. La tercera línea hace lo mismo excepto que ahora con el botón de la flecha hacia arriba en el primer piso; al presionar este, el led de esta flecha callToUp1 se encenderá y solo se apagará

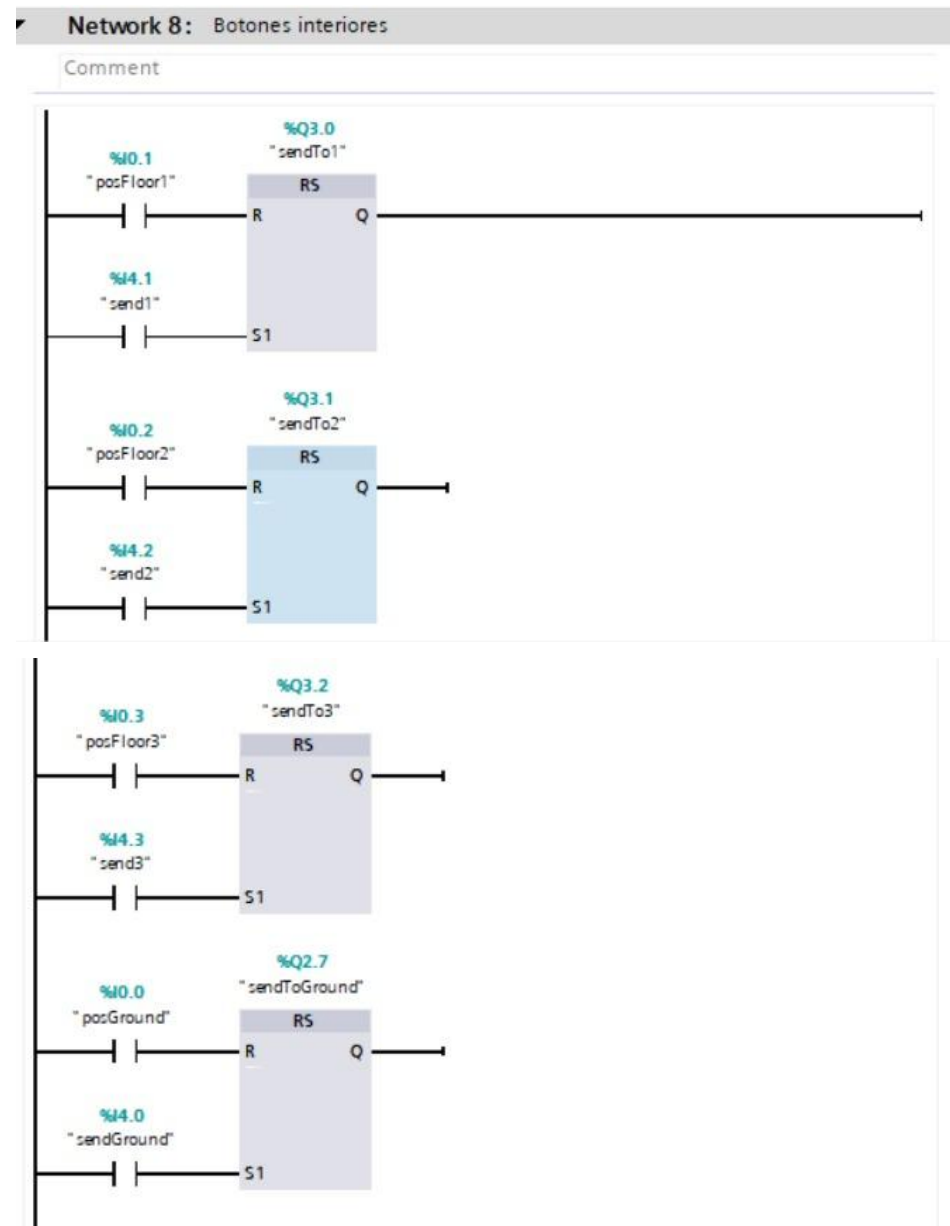
cuando el elevador llegue a este piso. Esto se repite para todos los botones de flecha en la parte exterior del elevador en nuestro sistema.

Network 7: Botones Exteriores

Comment



El **octavo network** hace lo mismo que el network anterior pero ahora para los botones dentro del elevador. Por ejemplo, al estar dentro del elevador y se selecciona ir al segundo piso, se mandará la señal y se prenderá el LED del botón de segundo piso. Esta luz se apagará cuando el elevador llegue al segundo piso. Esto se repite para todos los 4 pisos.

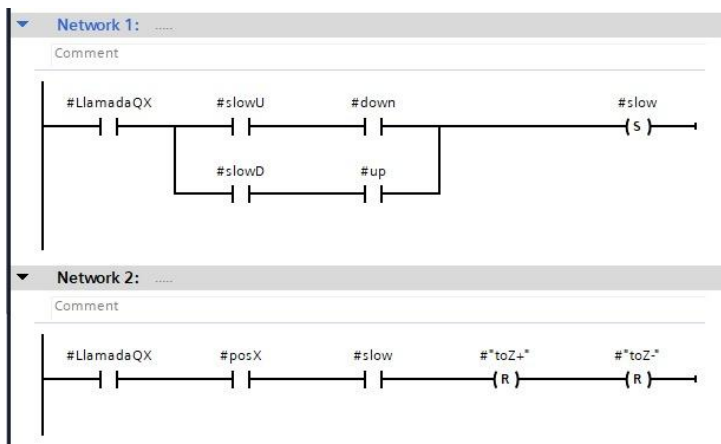


Del noveno hasta el décimo sexto network básicamente se realiza el mismo procedimiento, estos controlan la llegada o salida de los 4 pisos de nuestro sistema. Para hacerlo menos bultoso, desarrollamos unas funciones en otros 2 bloques aparte para permitir que este sea aplicable para todos. En estas nuevas funciones establecimos todas las variables que íbamos a necesitar como inputs e inOuts; de inputs tenemos todos los sensores y direccionales mientras que en inOuts tenemos a nuestros actuadores.

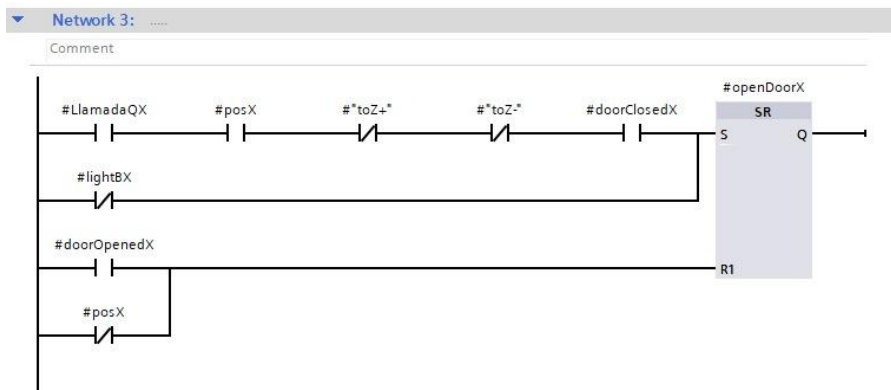
	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	▼ Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
2	■ LlamadaIX	Bool	false	Non-retain...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
3	■ posX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
4	■ doorOpenedX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
5	■ doorClosedX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
6	■ up	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
7	■ down	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
8	■ slowU	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
9	■ slowD	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
10	■ lightBX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
11	▼ Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
12	■ <Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
13	▼ InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
14	■ toZ+	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
15	■ toZ-	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
16	■ openDoorX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
17	■ closeDoorX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
18	■ slow	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
19	■ floorIndicatorX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
20	■ LlamadaQX	Bool	false	Non-retain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
21	▼ Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
22	▶ IEC_Timer_0_Instance	TON_TIME		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
23	▼ Temp				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
24	■ <Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
25	▼ Constant				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
26	■ <Add new>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Dentro de la primera nueva función, no ponemos valores específicos para que en nuestra función principal se nos permita utilizar el bloque para todos los pisos y no solo uno. La primera network empieza clasificando que si se activa la llamada designada y se activa o la variable de slow up que designemos y la memoria de down que establecimos o se activa la variable de slow down y la memoria de up, se va a activar el relevador slow.

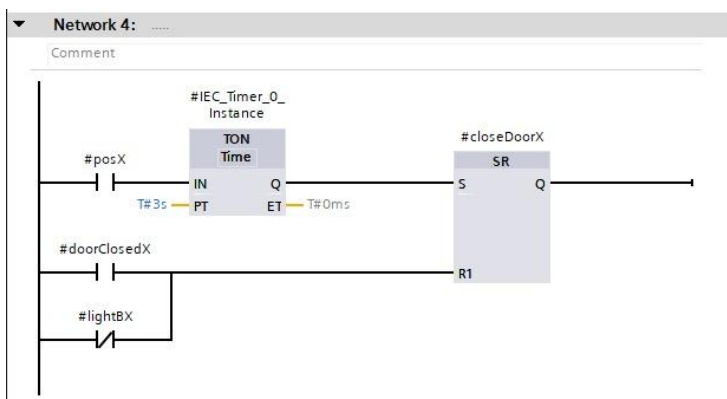
en el segundo network básicamente alinea que si se activa la llamada designada, en la posición designada y se activa el slow, se va a reiniciar el motor de movimiento hacia arriba y el motor de movimiento hacia abajo.



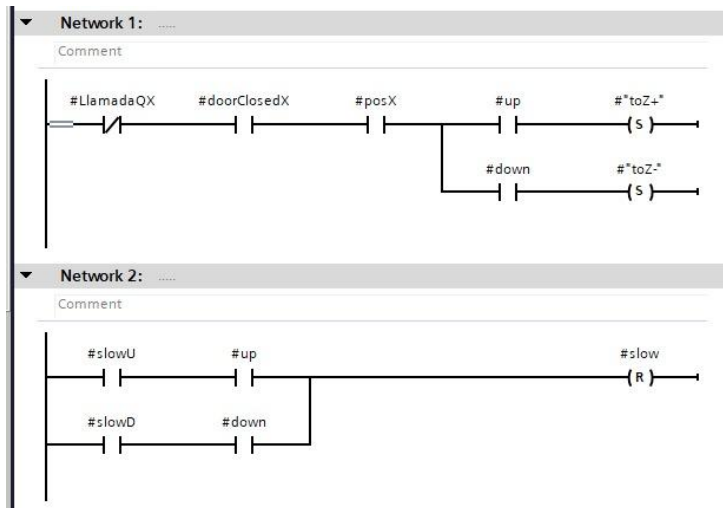
Continuando con el tercer network, este va a controlar si las puertas se van a abrir; empezamos con un sistema OR en donde la puerta se va a abrir o cuando se presente la llamada designada, se encuentre en la posición deseada, el motor para subir este apagado, el motor para bajar esta apagado, y la puerta designada se encuentre cerrada o el sensor de luz no se encuentre activo. Para que se reinicie el comando, la puerta debe de ya estar abierta o la posición designada debe de no estar activa.



El cuarto y último network de esta primera nueva función, se utiliza para controlar la cerrada de las puertas; cuando la cabina del elevador llegue a la posición designada, va a empezar un timer de 3 segundos y cuando pasen esos 3 segundos las puertas se van a cerrar. El sistema se va a reiniciar cuando la puerta ya esté cerrada o el sensor de luz designado no esté activado. En si el network 3 y 4 actúan al mismo tiempo excepto que la función de cerrar las puertas va a actuar 3 segundos después y por eso va a ayudar a cumplir con el requerimiento de mantener la puerta abierta por el tiempo designado. Esta primera nueva función será presentada como bloque 3 dentro de nuestra función principal.



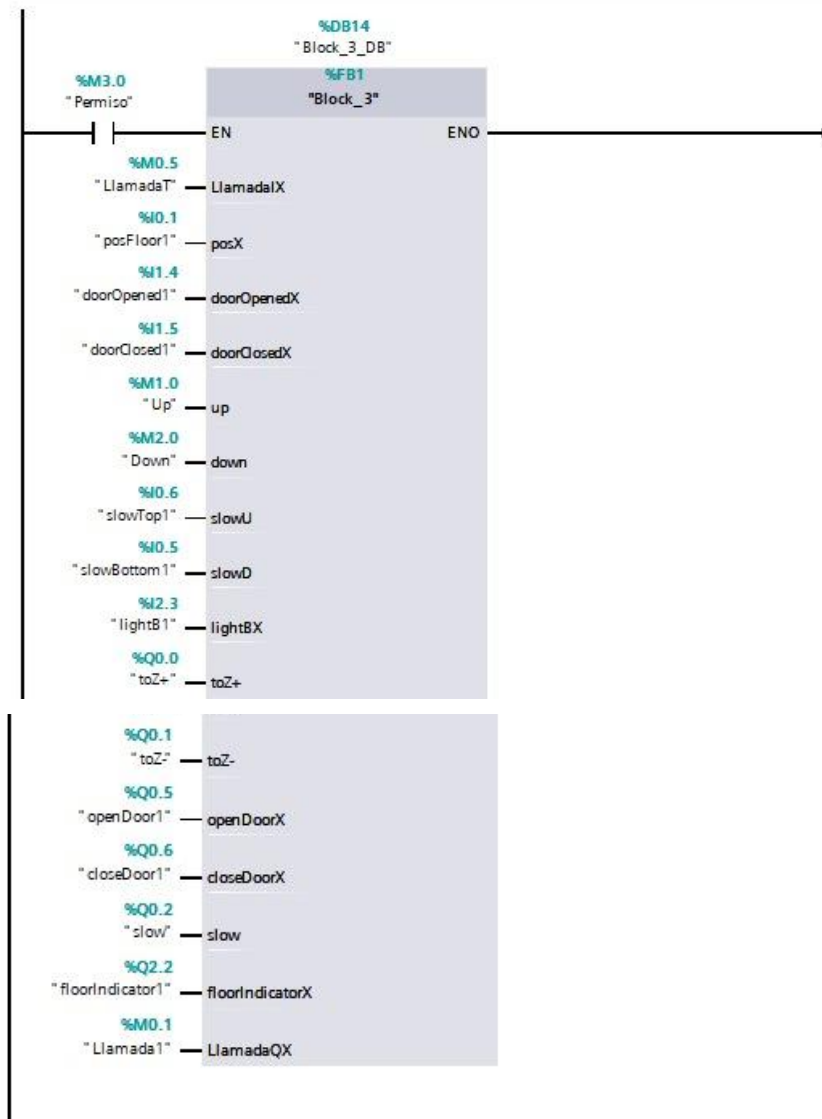
En la segunda nueva función esta se encarga de realizar la función de hacer salir la cabina del elevador desde el piso en el que se encuentra en ese momento. El primer network presenta que mientras la llamada designada se encuentre inactiva, las puertas estén cerradas, el elevador se encuentre en la posición designada o mientras se active la memoria up se va a activar el motor de subida o mientras se active la memoria de down se va a activar el motor de bajada. El segundo network presenta que cuando la variable de slowU y la memoria de up se encuentren activas o la variable en slowD y la memoria de down estén activas, se va a reiniciar el relevador slow. Esta segunda nueva función se presentará como bloque 4 dentro de nuestra función principal.



Ahora regresamos a la función principal en donde nuestra subfunción que creamos ahora es arreglada como los bloques 3 y 4. Los bloques 3 se encargan de realizar las llegadas que es el hacer la cabina llegar, reducir la velocidad al llegar y controlar el abrir y cerrar de las puertas mientras que el bloque 4 se encarga de las salidas que es checar que las puertas estén cerradas y que salga a una velocidad reducida de cada piso antes de regresar a su velocidad normal. Para explicar el acomodo, se hablara de él network 9 “Llegada a piso 1” y network piso 14 “Salida de piso 1”. Los dos networks mantienen las mismas variables pero las usan para otras funciones, el network 9 va a permitir que al llamar al elevador desde cualquier piso al primer piso, este llegue con velocidad reducida, pare, abra sus puertas por 3 segundos, vuelva a cerrar y se quede ahí hasta que reciba otro comando. El network 14 va asegurarse que el elevador tenga las puertas cerradas antes de mandar al elevador al siguiente piso designado empezando con una salida lenta y luego recuperando su velocidad. Estos dos bloques se repiten para cada piso como se pueden ver en nuestros networks.

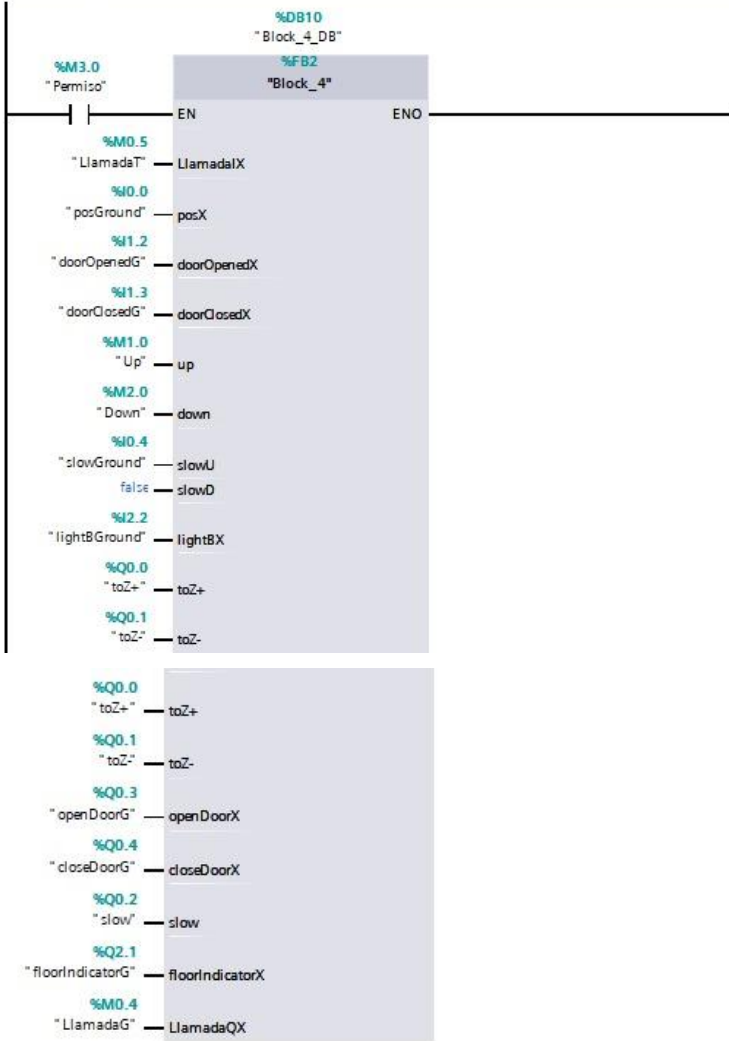
Network 9: Llegar a Piso 1

Comment



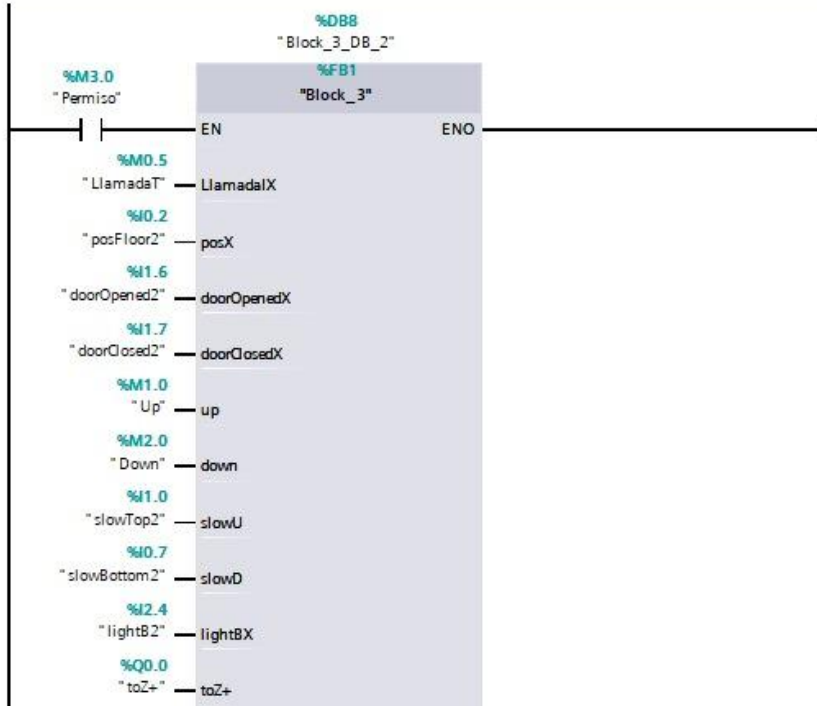
▼ Network 10: Salir Ground

Comment



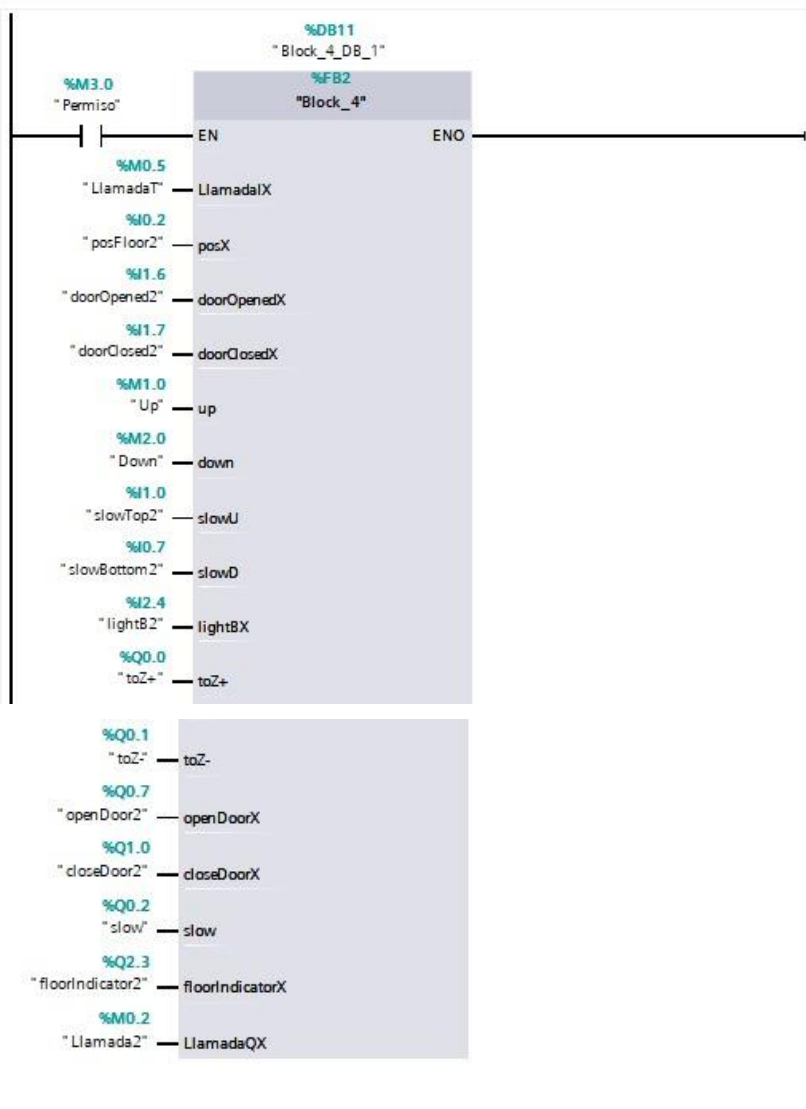
▼ Network 11: Llegar Piso 2

Comment



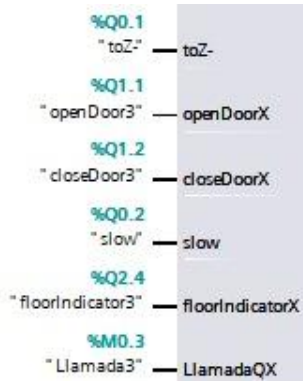
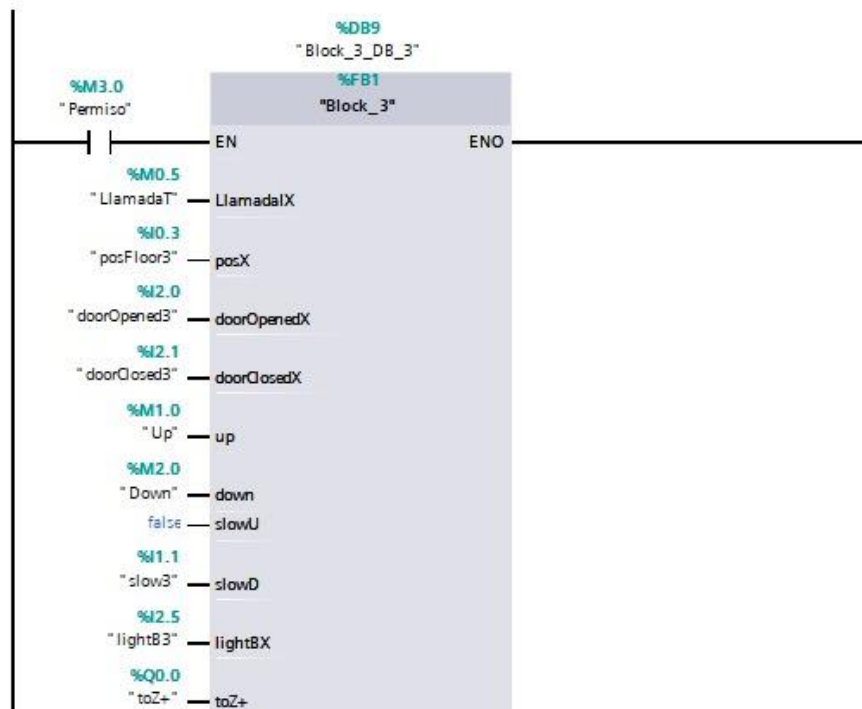
▼ Network 12: Salir Piso 2

Comment



▼ Network 13: Llegar Piso 3

Comment



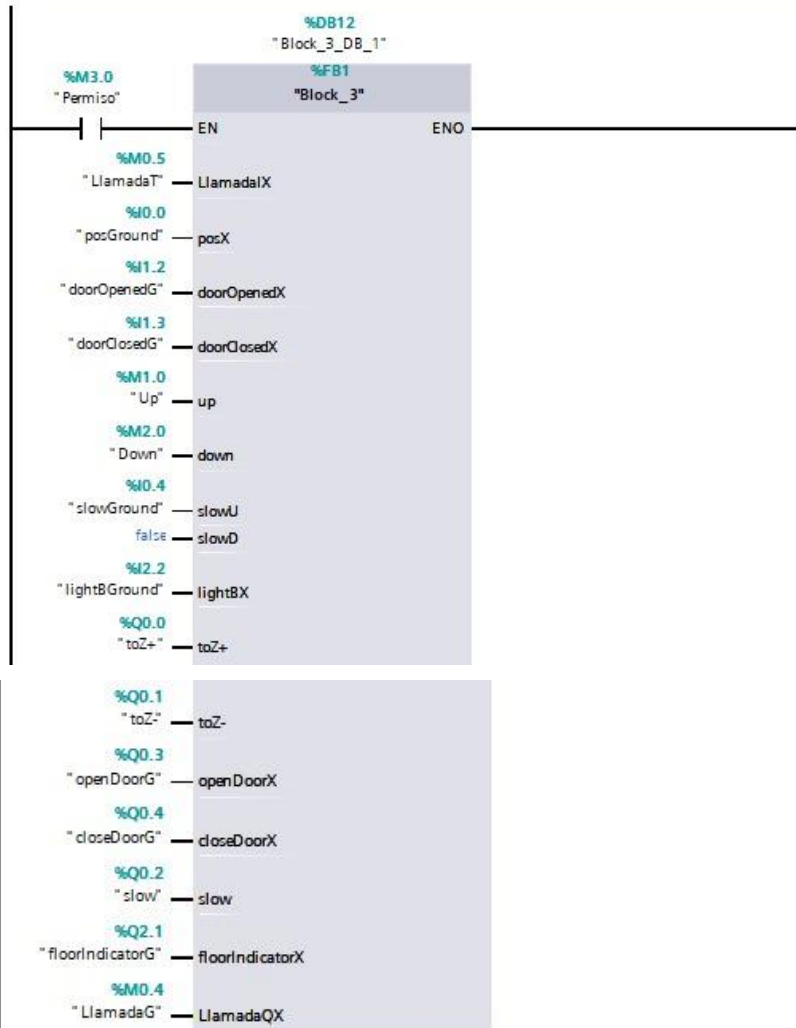
▼ Network 14: Salir Piso 1

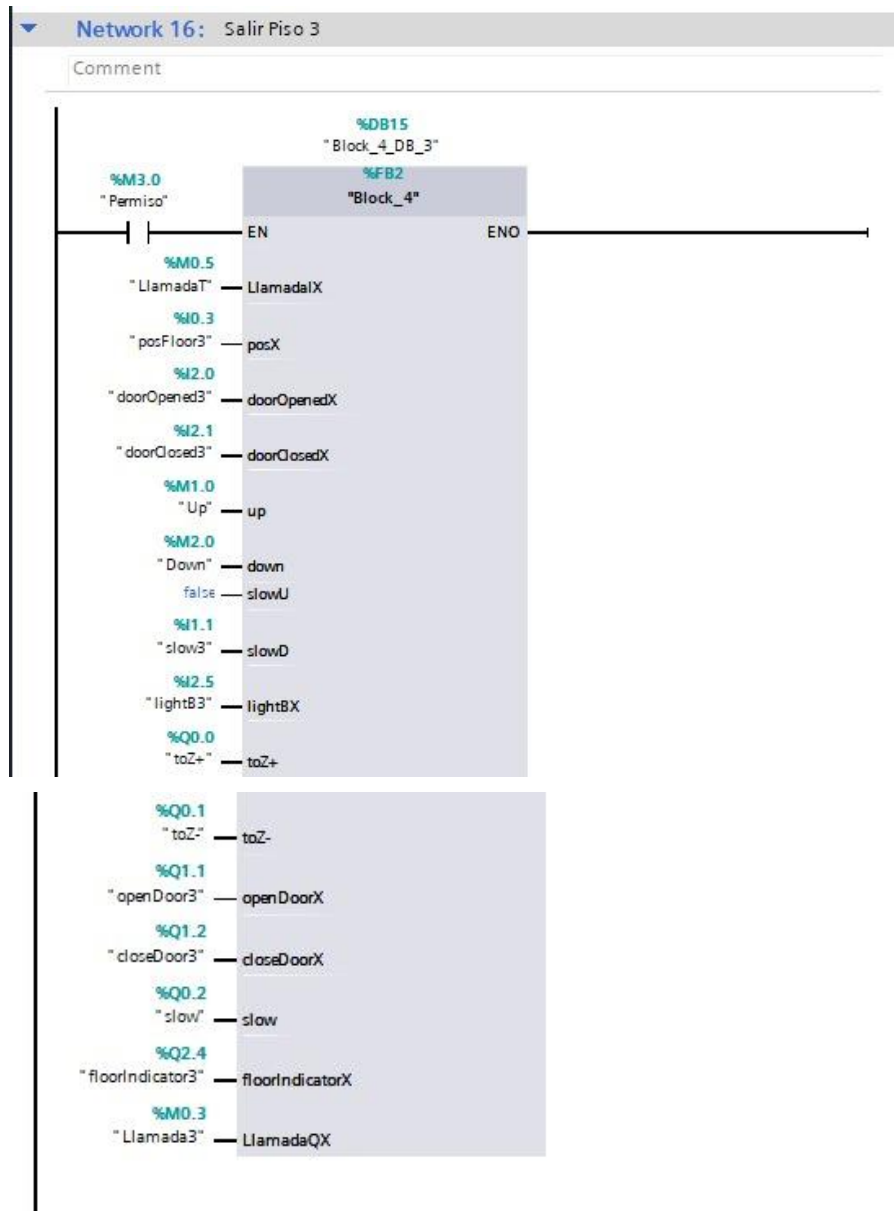
Comment



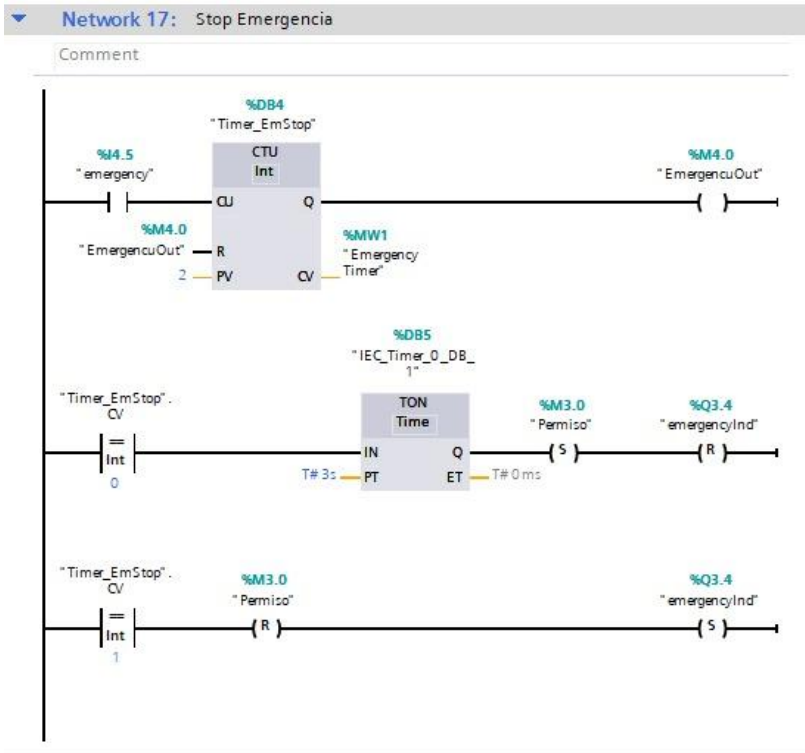
▼ Network 15: Llegar Ground

Comment

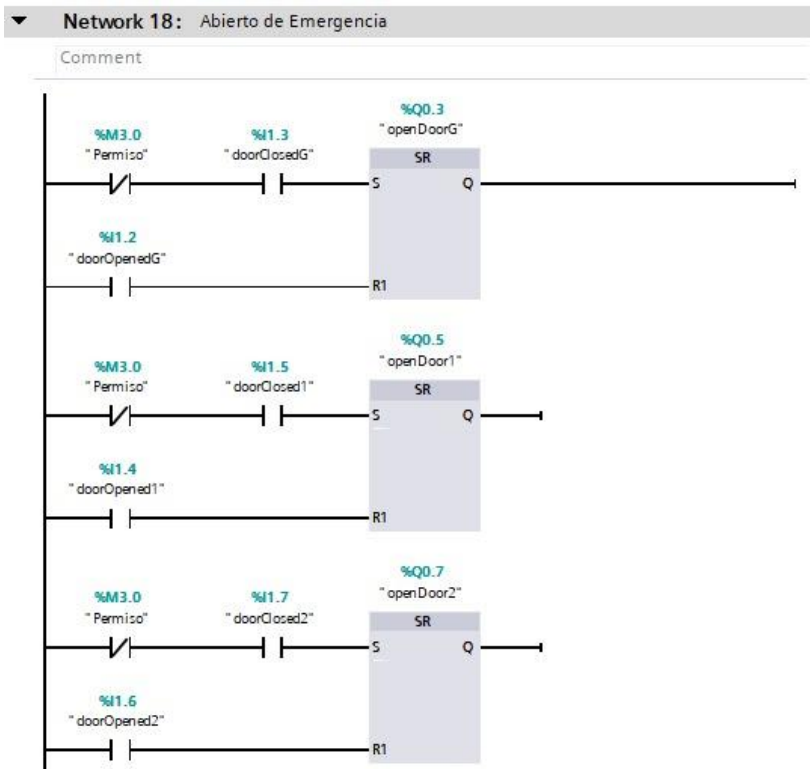


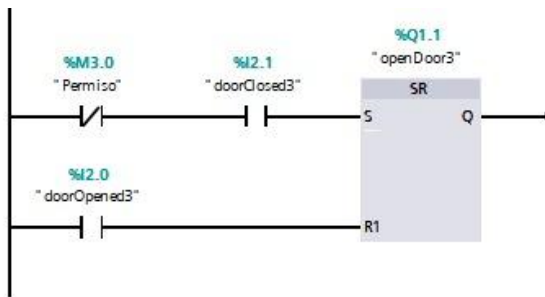


Para el **décimo séptimo network** , ahora empieza a tocar el tema del paro de emergencia; en este network empezamos solo con el sistema para ir al elevador. Si el botón de emergencias es presionado, este se registrará en counter empezando desde el “EmergencyTime” establecido que es una memoria que creamos. Esto se guardará en otra memoria titulada EmergencyOut; cuando se presione el botón dos veces, el contador va a ser que el EmergencyOut se reinicie. Continuando con las siguientes partes, si el valor del contador es igual a 0 se esperará 3 segundos hasta que mande una señal para activar nuestro permiso y reiniciar el indicador de emergencia. Si el valor del contador es igual a 1 este va a reiniciar el permiso y va a activar el indicador de emergencia.

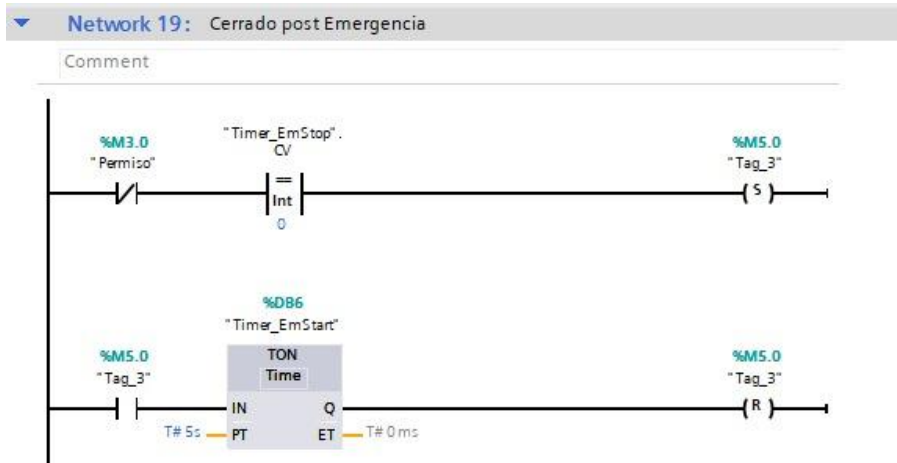


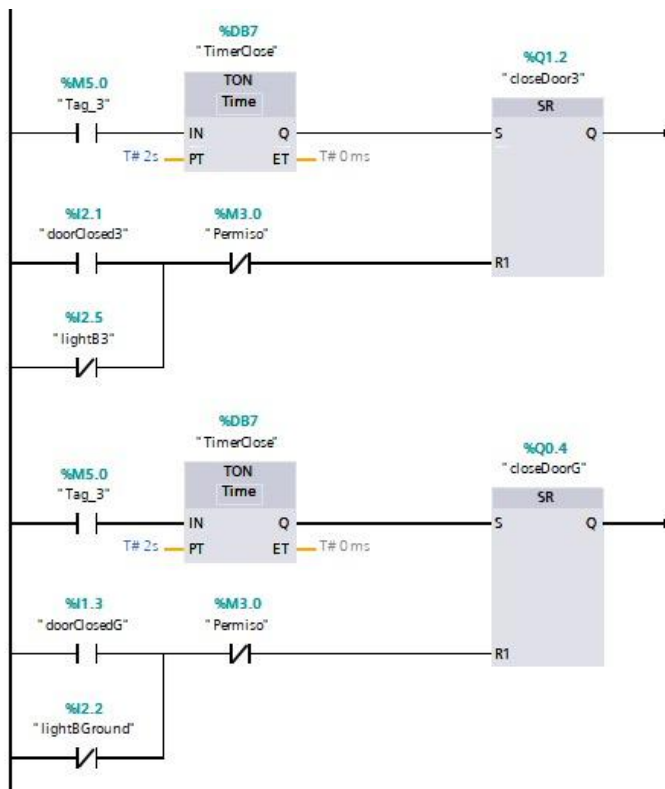
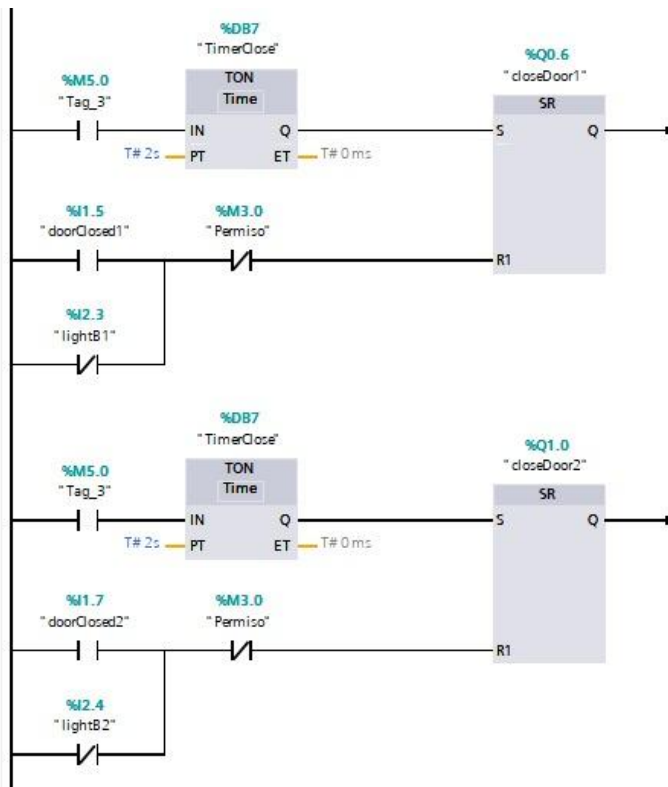
Continuando con el **décimo octavo network**, este se enfoca en el proceso de abrir las puertas cuando se activa el botón de emergencia; las 4 partes del código hacen lo mismo pero designado a cada diferente piso. Lo que sucede es que mientras el permiso no se encuentre activo, y la puerta se encuentre cerrada, se va a poder activar el comando de abrir la puerta; el sistema se va a reiniciar cuando ya se detecte que la puerta ya está abierta.





El **décimo noveno network** ahora trabaja en cerrar las puertas cuando ya se acabó la emergencia. Mientras que el permiso se mantenga inactivo, y el timer que se encuentra en el network 17 sea igual a 0, se va a activar nuestra memoria que planteamos. Si esta memoria se encuentra activa, va a empezar un temporizador TON de 5 segundos y al terminar se va a reiniciar esta memoria. Aquí es donde sigue el control de la puerta; igual que en otros de nuestros networks, esta función se repite 4 veces para satisfacer la necesidad de la puerta de cada piso. En si dependiendo de el número de el piso, cuando se active la memoria que mencionamos anteriormente, empezará a correr un temporizador TON de 2 segundos que al terminar mandará la señal a la función de cerrar puerta del piso designado. Para reiniciar el sistema o la puerta designada debe de estar cerrada y el permiso desactivado o la banda de luz debe de estar desactivada y la banda de luz también desactivada. Esto se hace en todas las puertas para cerrarlas.

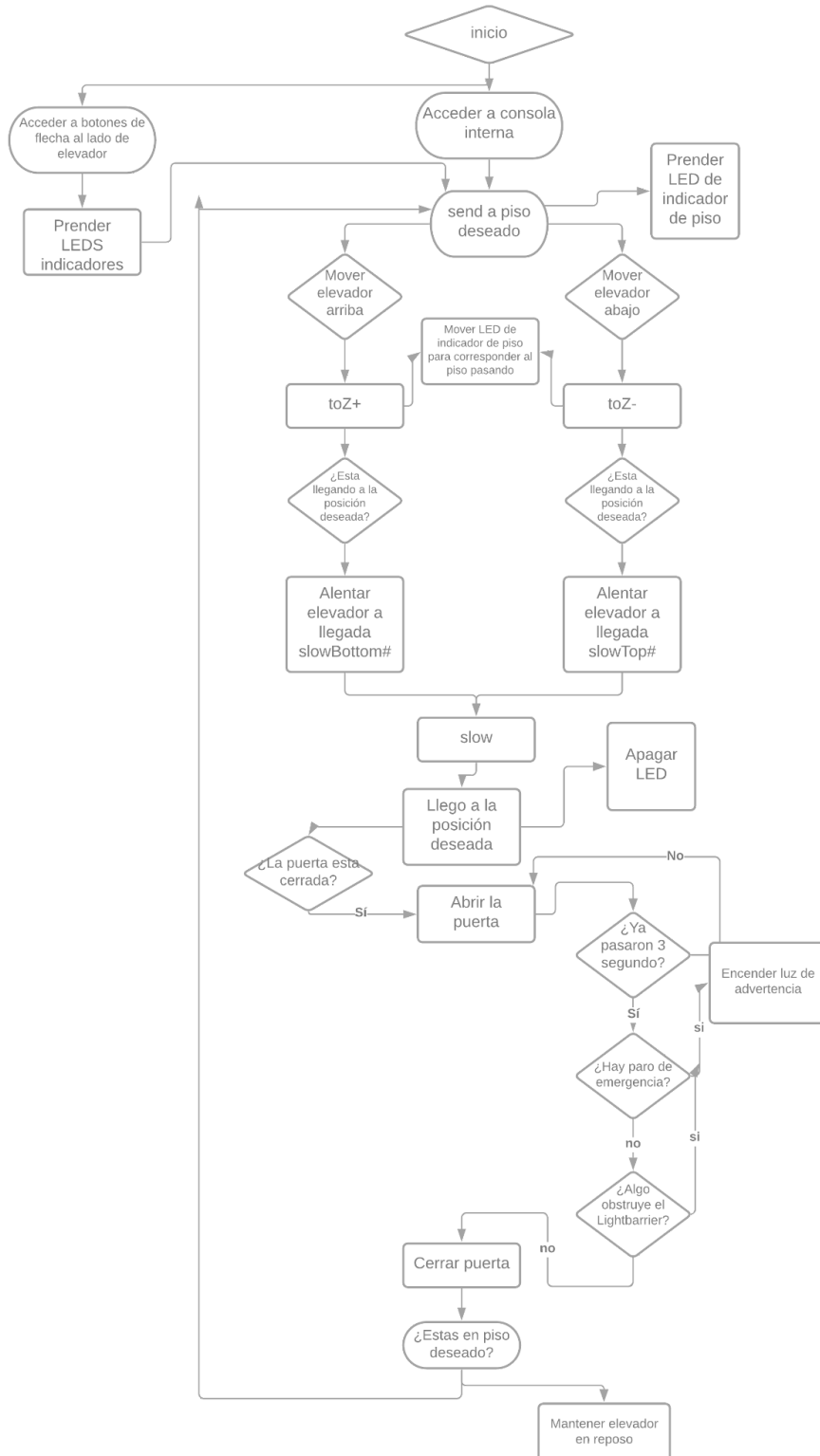




Con esto se termina el desarrollo de nuestro programa para hacer el elevador funcionar; a continuación se anexa un video de la demostración de el funcionamiento de el elevador en nuestra prueba de finalización de este proyecto:

<https://www.youtube.com/watch?v=zg7icX6n70U>

Diagrama de flujo del elevador:



Conclusión

Después de la realización de este proyecto, nos es posible entender de mejor manera el funcionamiento de los PLC's cómo técnica de control y su amplio rango de uso, tanto para la industria como para maquinarias que se usan en nuestro día a día. La programación de escalera, además de ser fácil de entender, presenta varias características que permiten desarrollar resultados complejos y avanzados. Esta también nos permitió observar de distinta manera la relación entre cada una de las variables que se deben de tomar en cuenta al momento de realizar una acción. Como en nuestro caso eran los permisos relacionados a que no se presentara ninguna emergencia o alerta, que cambiara de piso si las puertas estuvieran cerradas y demás.

Cabe mencionar que, durante la realización de este proyecto se presentó una complicación durante la práctica de otro proyecto, generando así un corto circuito en la computadora en donde se realizaba nuestro primer intento y dando pérdida no sólo al avance del proyecto, sino también a una computadora relativamente nueva.

Pero a pesar de este inconveniente, gracias al amplio conocimiento que teníamos sobre este tema, nos fue posible solucionar la problemática e incluso comprender de mejor manera ciertos aspectos, que puede que, aún no quedaran completamente claros; y el desarrollo de nuevos conceptos que nos permitieron completar la actividad.