**Universidad Politécnica de la Zona Metropolitana de Guadalajara**

**Carrera**: Mecatrónica

**Materia:** Controladores Lógicos Programables

**Maestro:** Carlos Enrique Moran Garabito

**Alumno:** Jessica Márquez Flores

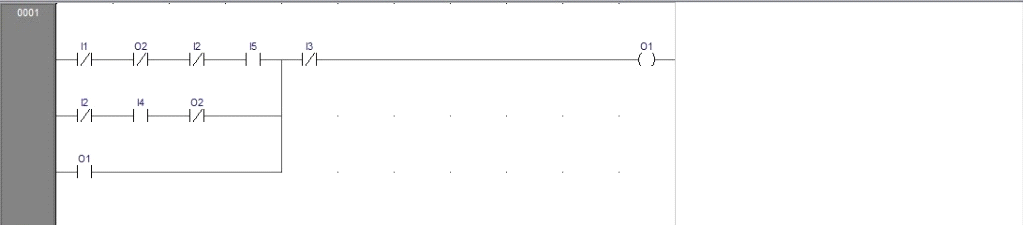
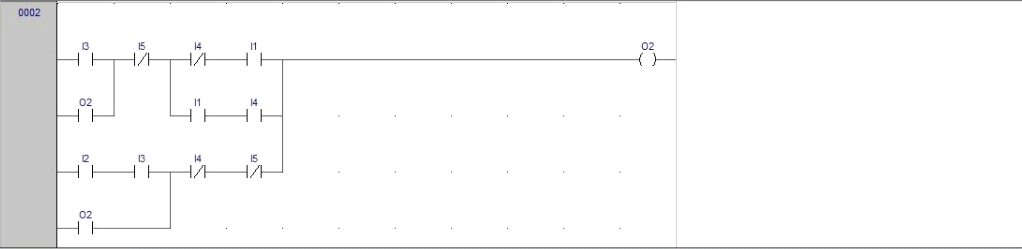
|  |
| --- |
| Contenido |
| Actividades 3  Practicas 4  Tareas 2 |

**Actividad 1**

Introducción:

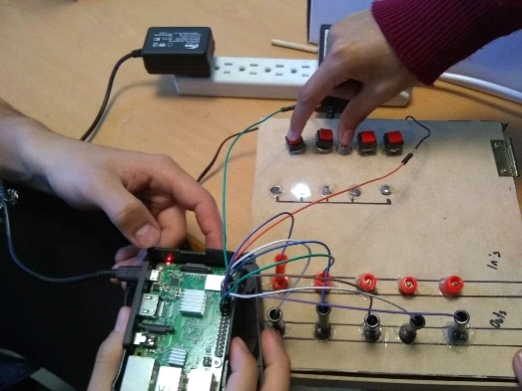
En esta actividad nos introduciremos al mundo de programación de un PLC por medio del “Diagrama de Escalera” utilizando recursos proporcionado por el docente los cuales serán Raspberry p 3 b+ un el programa para poder transferir todos los datos a esta, el cual será el Axel “LogicLab” utilizando el diagrama de escalera, y poder crear una serie de códigos transmitidos a través de botones no enclavados, y para lograr visualizar si el código funciona correctamente utilizaremos diodos LED.

Procedimiento:

Iniciamos encontrando como poder añadir nuevos interruptores y más líneas al programa para poder crear el código correcto, el cual es el siguiente:

Obteniendo 5 entradas y 2 salidas

Se tiene un cilindro en posición de (retraído) y con el sensor A activo. Cuando el operador presiona el botón P el cilindro sale hasta el sensor C, y si el botón deja de presionar regresa a home, pero si deja de presionar regresa a home.



Conclusión:

Se dificulto mucho la instalación de la Raspberry por la licencia, el ISO, etc... cuando se va desarrollando la actividad, comprendíamos que el diagrama de escalera se basaba en etapas y condiciones que hacían que la transición de etapa en etapa fuese posible, aun así, el diseño de un diagrama de escalera que funcione requiere de pruebas y errores, o al menos ese fue nuestro caso, para poner fin a esto, puedo decir que fue una buena “simulación de la transición de un pisto, de “A” a “C”.

**Actividad 2**

Línea de 4 latas

PROPOSITO:

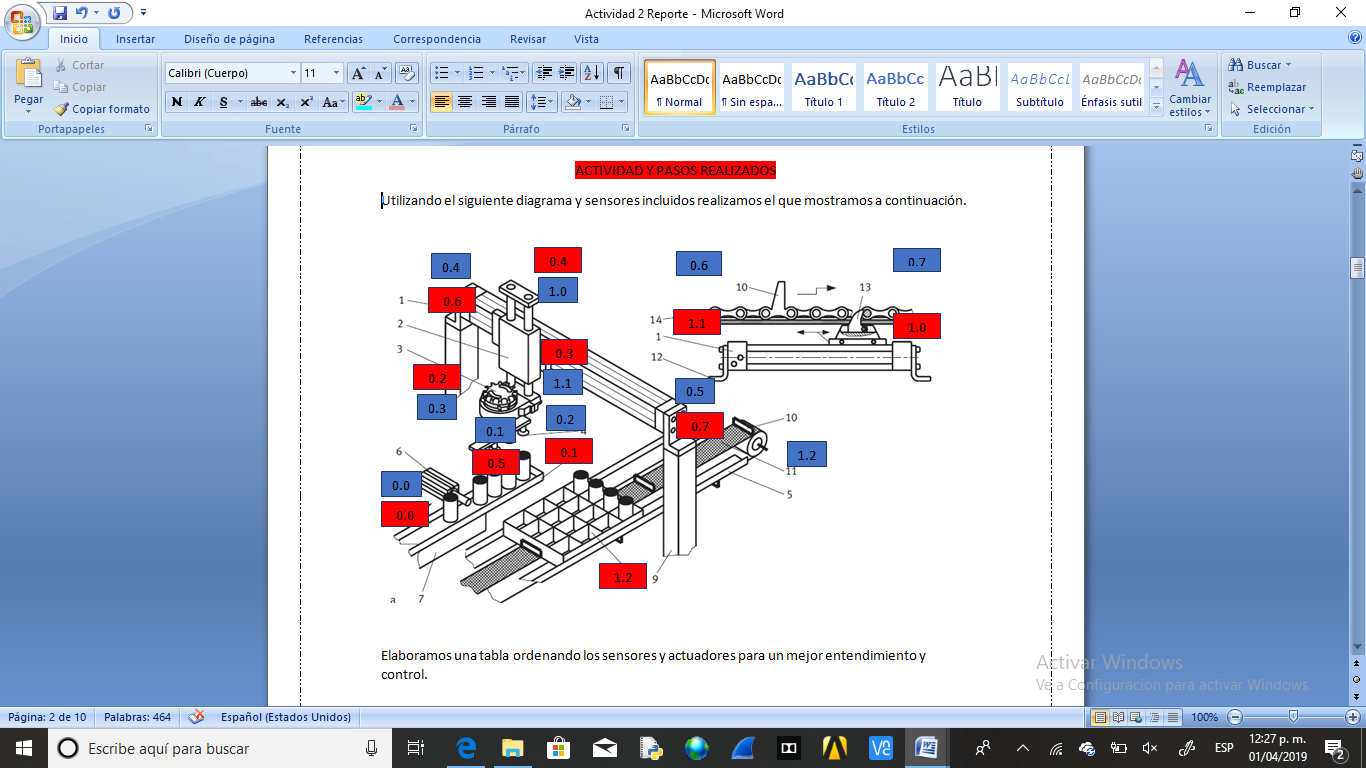
tiene como propósito brindarnos los conocimientos de cómo realizar un mapa de GRAFCET y pasarlo a un diagrama de escalera para posteriormente pasarlo a un PLC en este caso el LogicLab

INTRODUCCIÓN:

El Grafcet es un esquema o mapa para ver y diseñar el correcto funcionamiento de una maquina o línea de producción, donde separamos sensores y actuares con una misma nomenclatura como pueden ser 00, 0.1, 0.1, .etc.. esto dependiendo de el usuario, en este caso optamos por utilizar los mismos nombres para los sensores y actuares claro sabiendo que su función es diferente.

ACTIVIDAD Y PASOS REALIZADOS

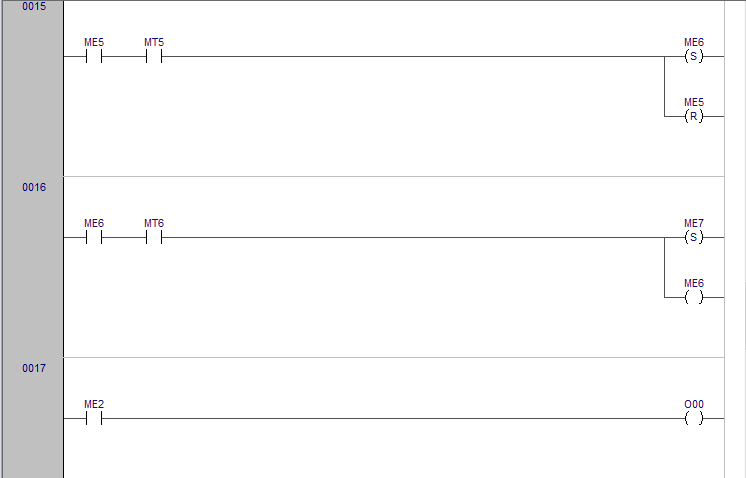
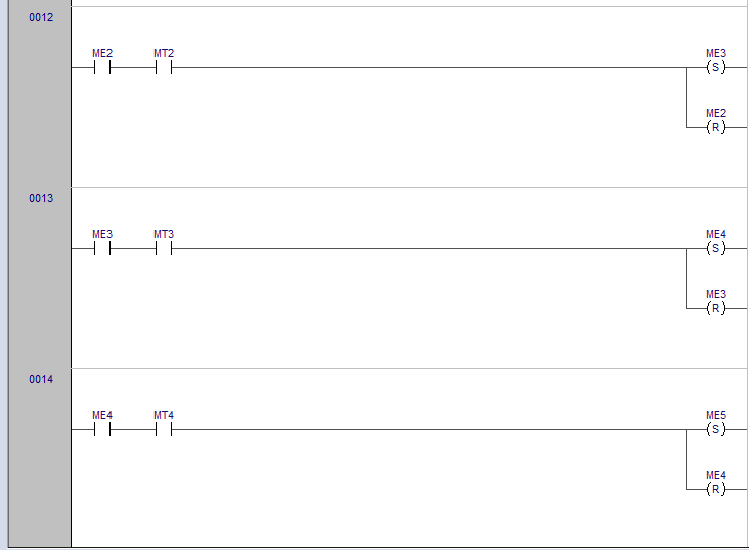
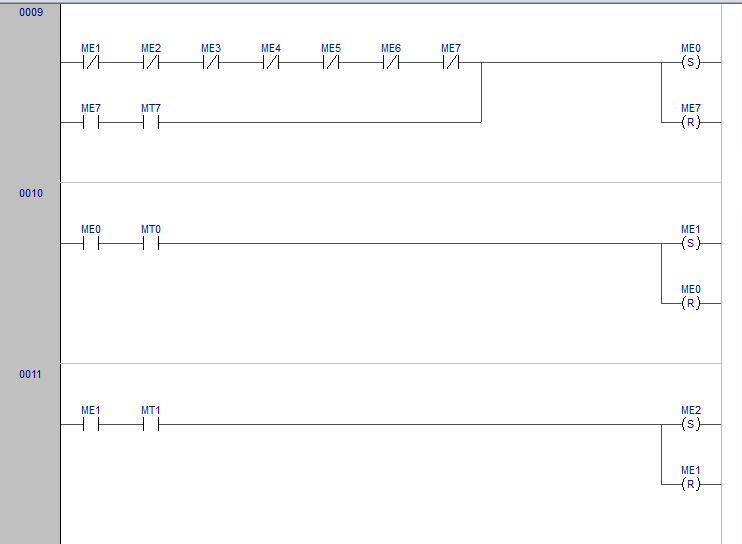
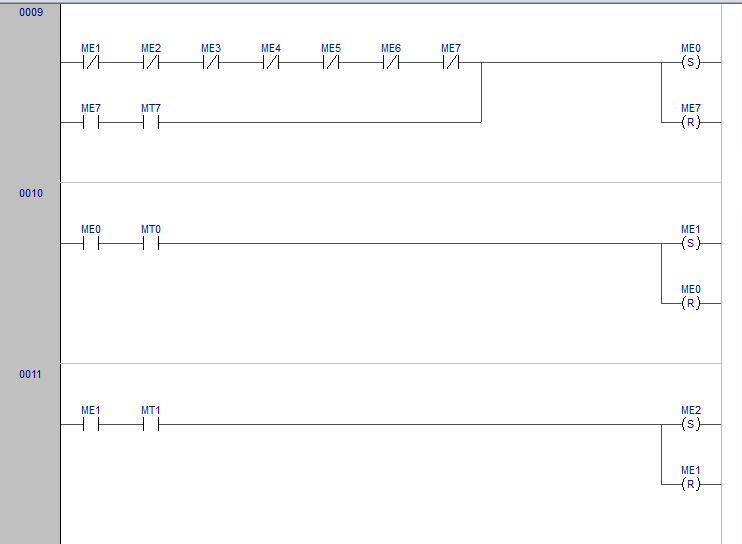
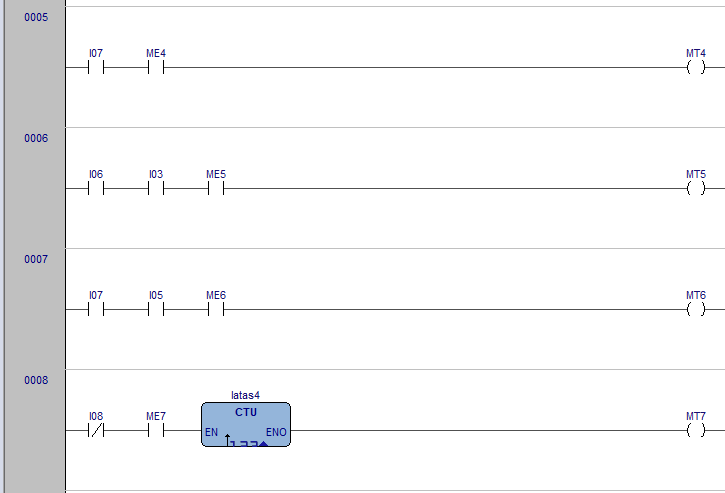
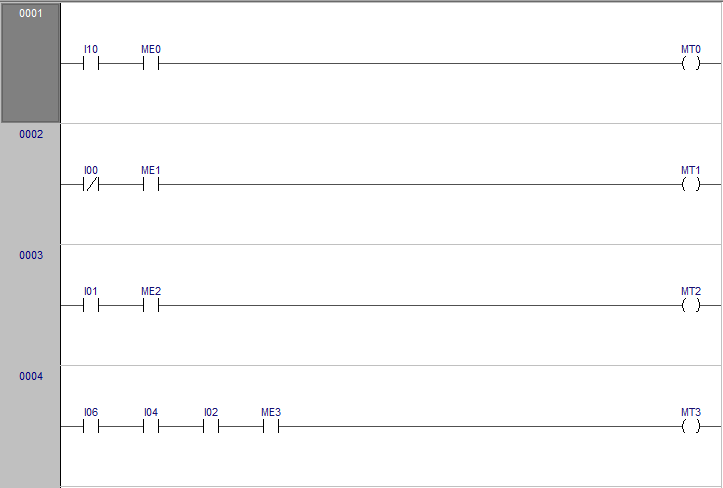
Utilizando el siguiente diagrama y sensores incluidos realizamos el que mostramos a continuación.



Elaboramos una tabla ordenando los sensores y actuadores para un mejor entendimiento y control.

|  |  |
| --- | --- |
| **SENSORES** | **ACTUADORES** |
| I00=CUATRO LATAS FULL | 00=CUATRO LATAS (TOPE) |
| I01=GIRO | 01=VACIO CHUPON |
| I02=GIRO | 02=GIRA 90 GRADOS |
| I03=BRAZO ABAJO | 03=REGRESO 90 GRADOS |
| I04=BRAZO ARRIBA | 04=MOVER DERECHA |
| I05=VACIO | 05=MOVER IZQUIERDA |
| I06=BRAZO IZQUIERDO | 06=MOVER POSICION 1 |
| I07=BRAZO DERECHO | 07=MOVER POSICION 2 |
| I10=POSICION INICIAL CAJA | 10=SUBIR BRAZO |

Una vez analizado y comprendido los sensores y actuadores basados en el diagrama pasamos las listas de estos a físico (hoja en blanco) para poder realizar el Grafcet de manera correcta y si surgía algún cambio o error poder modificarlo a tiempo para no cometer errores grandes.



Este diagrama es ya colocado el LogicLab, intentamos simularlo, pero no se puede simular ya que había un error en la actividad inicial que nos dio el maestro, y nos dijo que la dejáramos hasta el diagrama de escalera e cual es este paso.

Cabe destacar que tuvimos que poner un contador, ya que como mencionamos en las imágenes de Grafcet se ocupaba algo para que la línea regresara a home.

CONCLUSIÓN:

Resulto ser una actividad introductoria bastante simple, esencial para aprender a plasmar las ideas de un proceso y después desarrollarlas en un diagrama Ladder que hará exactamente lo que tú le ordenaste.

Por otra parte, la simulación es muy intuitiva dándote a entender muy bien los efectos que tienen el activar un sensor o combinarlo con otro.

**Actividad 3**

**Introducción:**

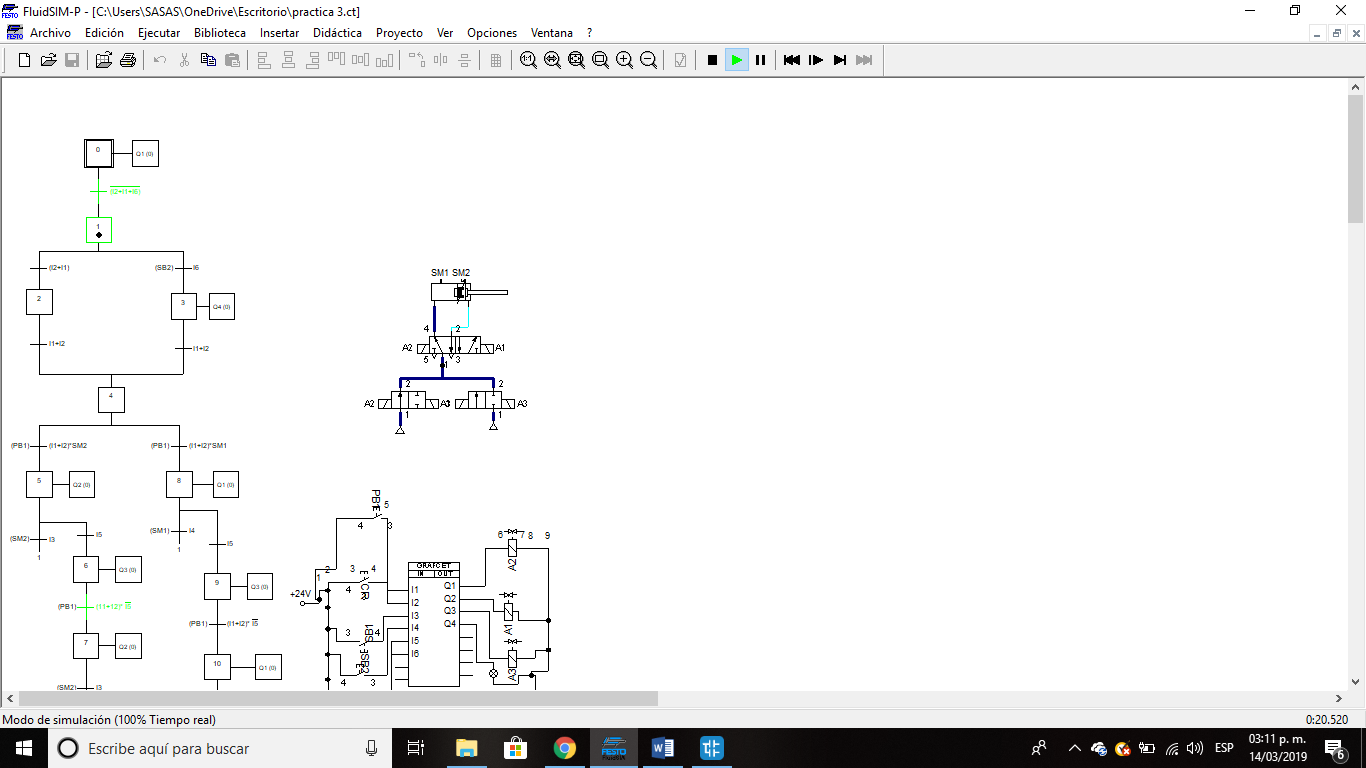
*Funcionamiento de un garaje eléctrico.*

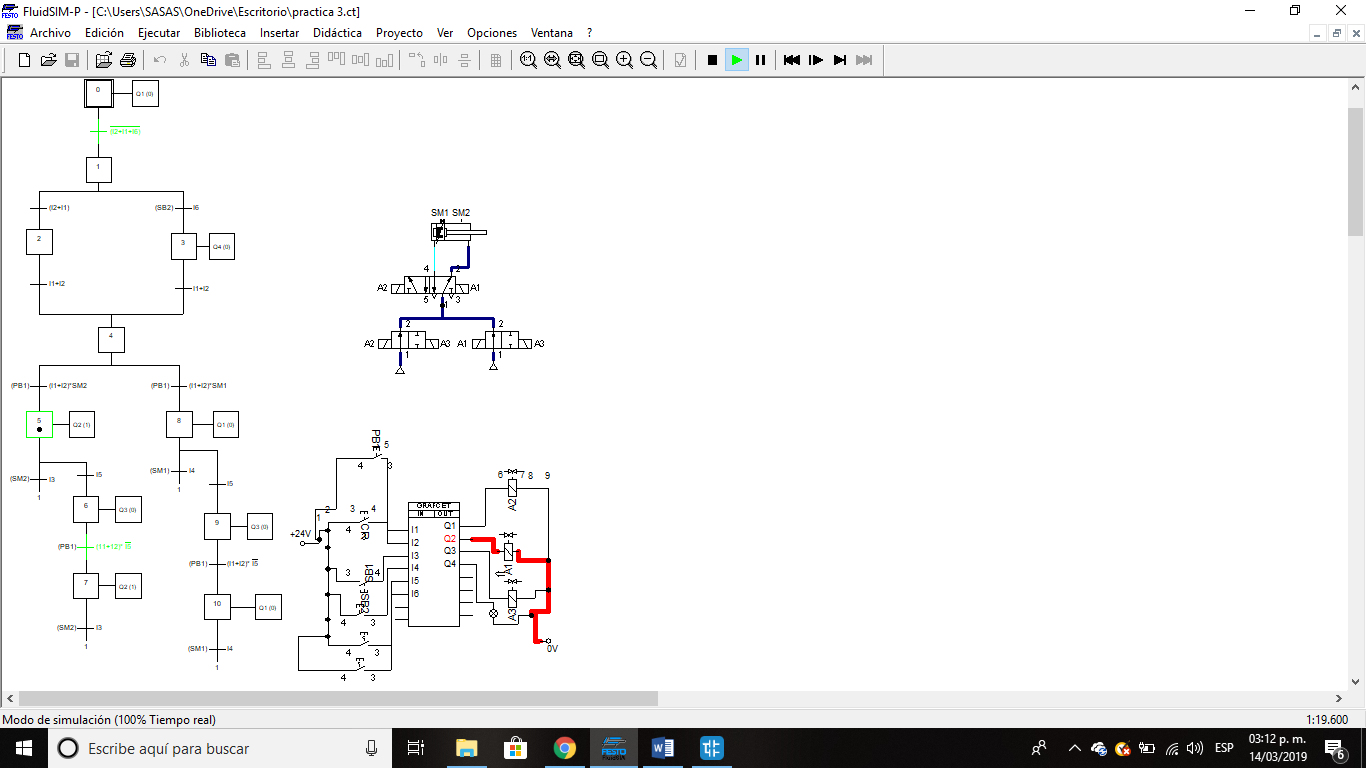
**Objetivo:**

*El garaje eléctrico debe de subir y bajar la puerta con sus respectivos limites, es decir, que se detenga el motor cuando la puerta ya está lo suficientemente cerrada y se detenga cuando la puerta ya esté totalmente cerrada. También se necesita que la puerta tenga un sensor que detenga el motor cuando detecte que algo obstruye su paso y de esta manera evitar algún accidente.*

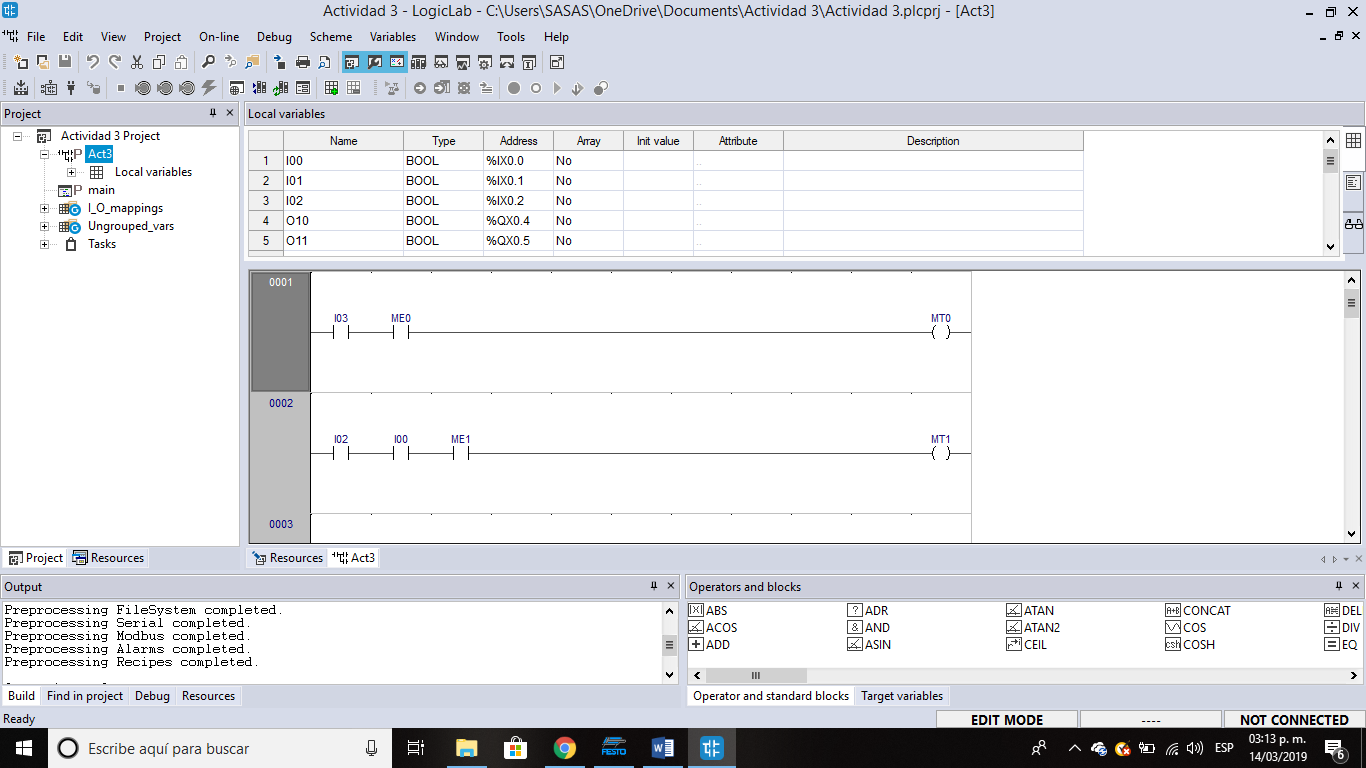
**Desarrollo:**

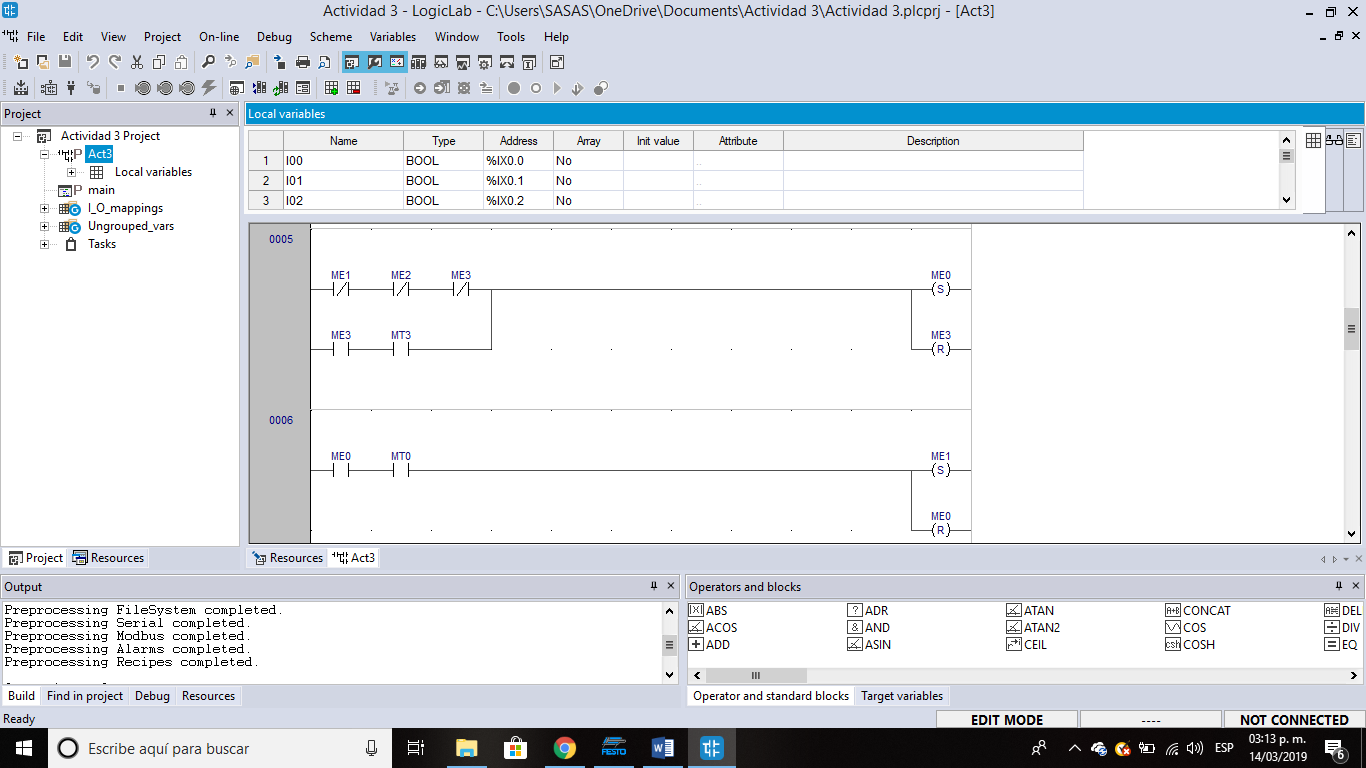
*La práctica deberá tomar inicio con el desarrollo del proceso en un Mapa GRAFCET, el cual ira a la par de un circuito del programa de FluidSim…*

*Posteriormente será simulado.*

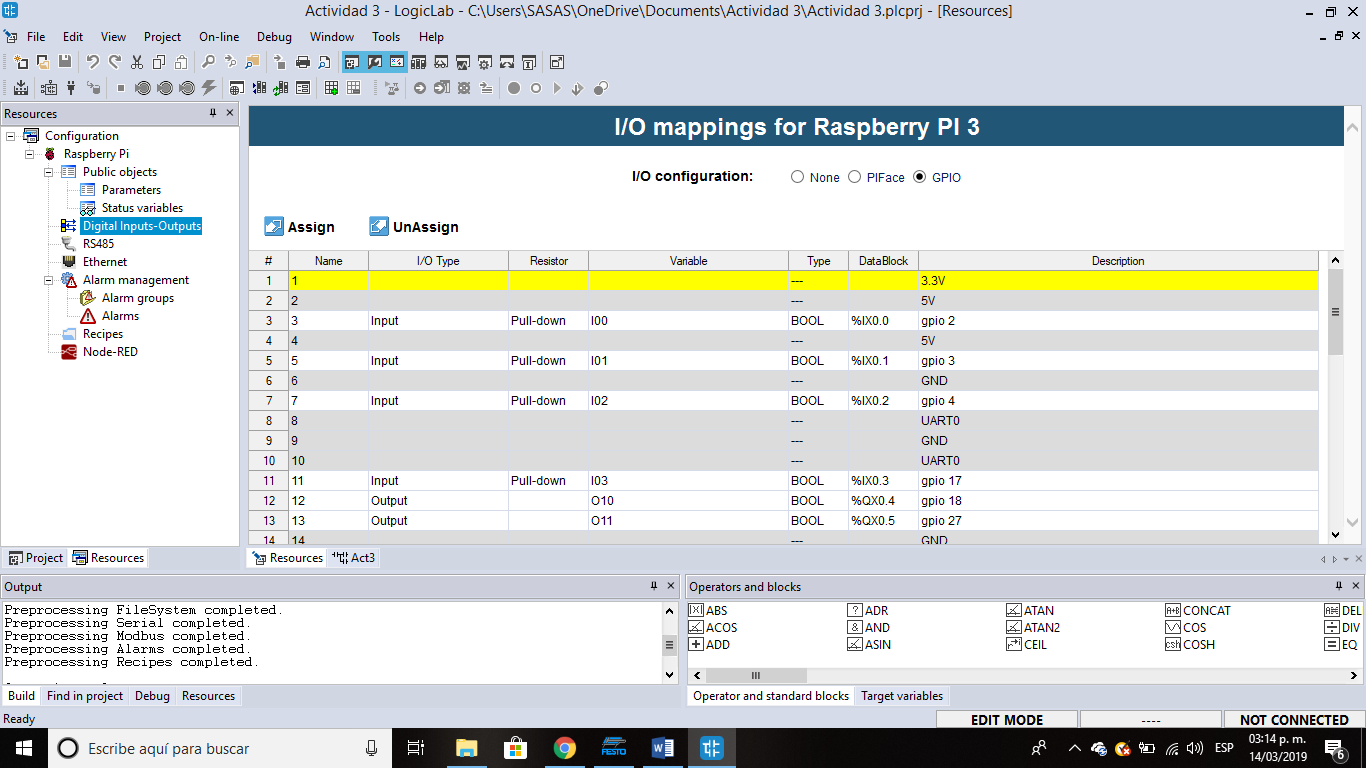


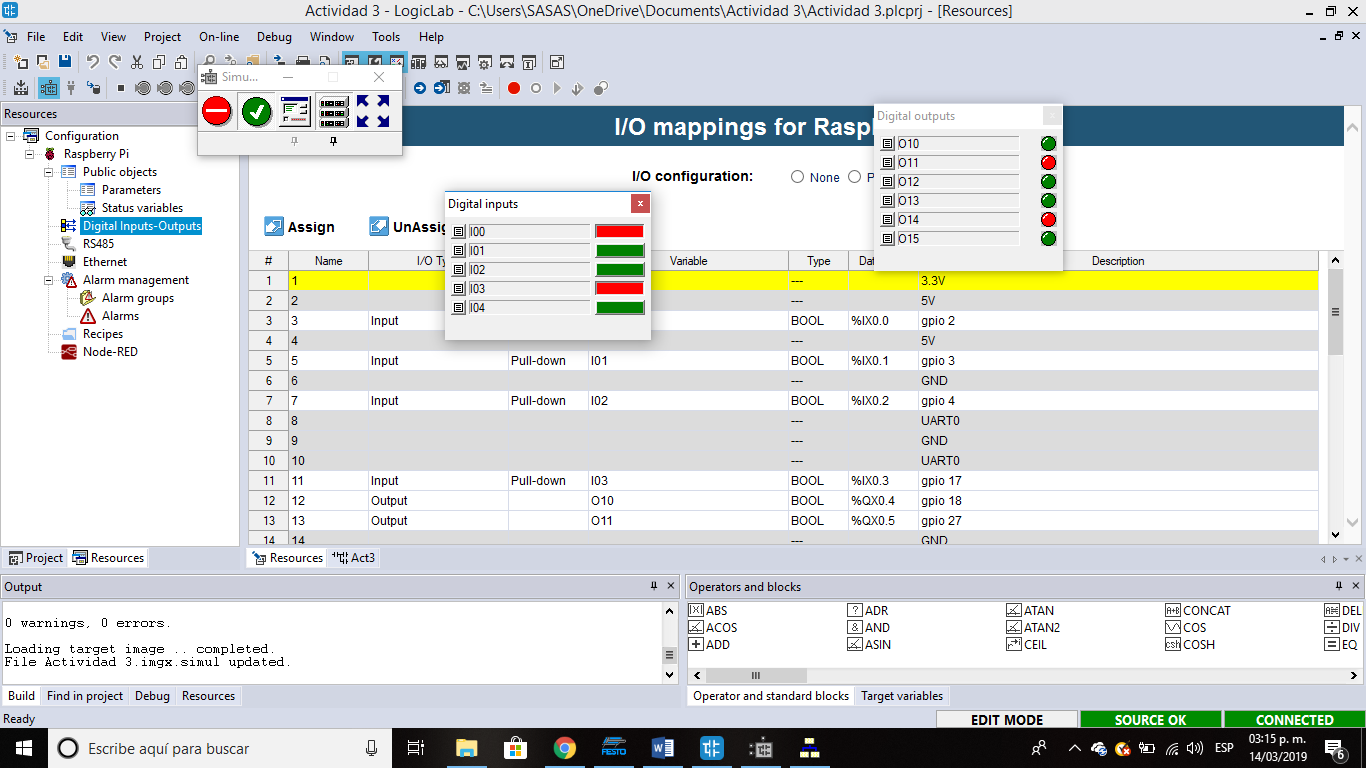
*Al ser simulado, procederemos a desarrollar un diagrama de escalera basado en nuestro mapa GRAFCET en LogicLab.*





*Finalmente simulamos estableciendo todas las entradas y salidas.*





**Conclusión:**

*Fue un proceso el cual debía tener como condición esencial el que no aplastara todo aquello que por causas del destino se encuentre debajo de la puerta.*

***Práctica 1***

Introducción:

Tenemos 3 objetivos principales, serán las consecuencias del presionar los primero 3 botones de nuestra caja de operaciones.

Al presionar el botón 1 los led’s deben encenderse del lado izquierdo hasta el derecho.

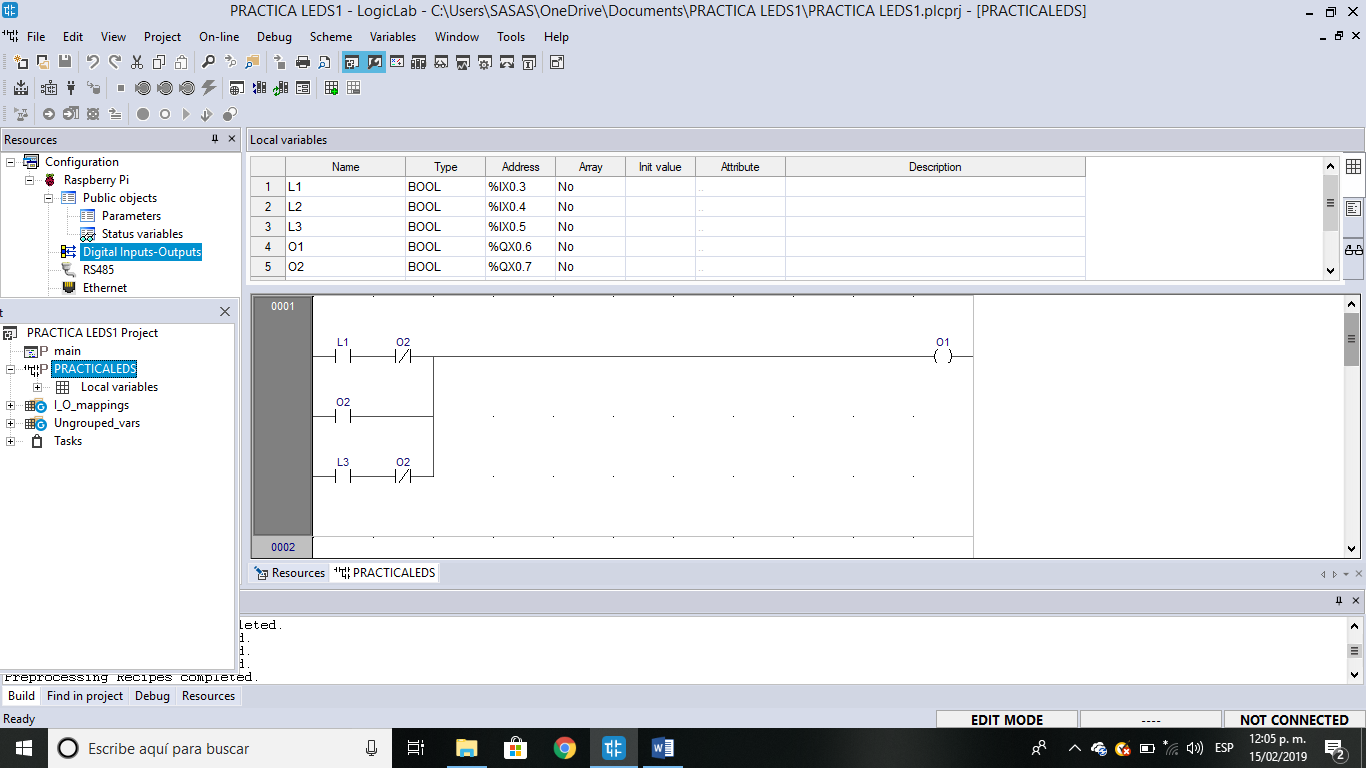
Al presionar el botón 2 los led’s deben encenderse del lado derecho hasta el izquierdo.

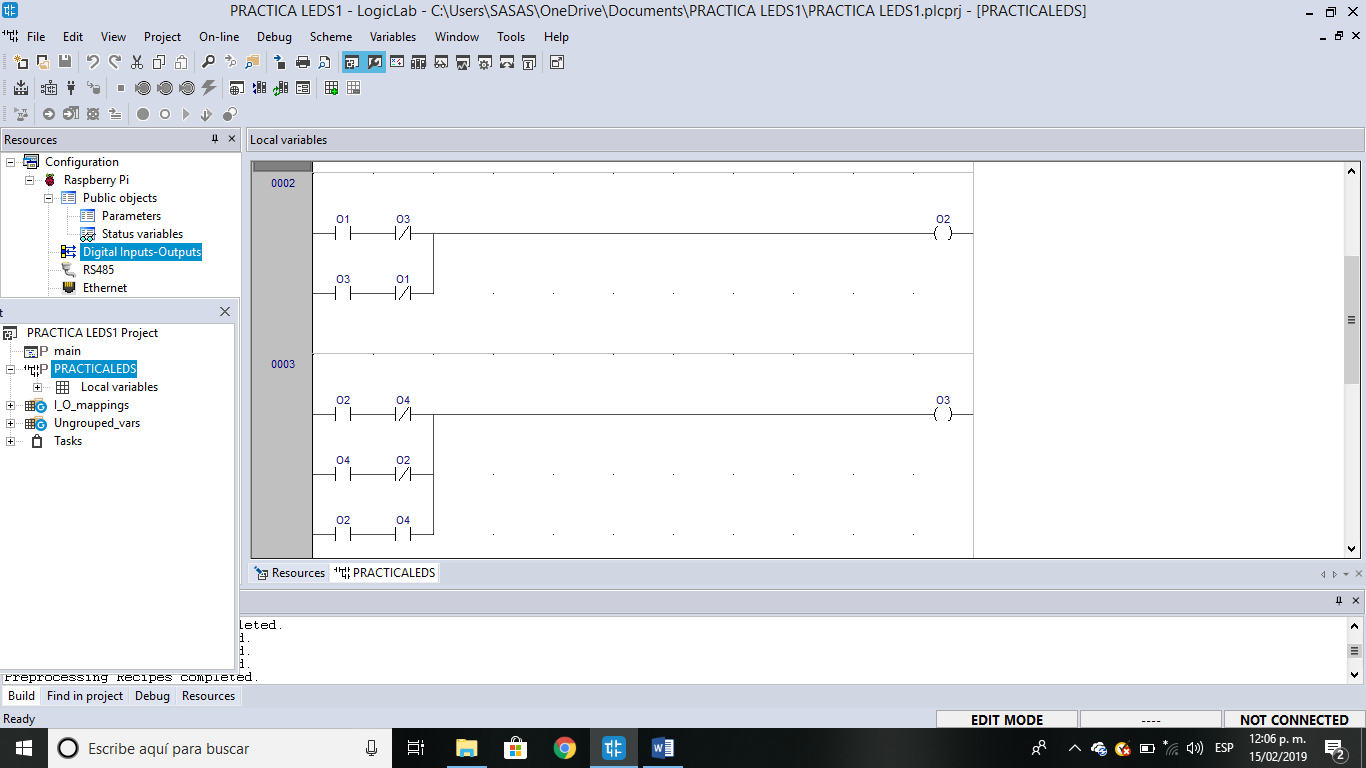
Al presionar el botón 3 los led’s deben encenderse de ambos lados en dirección al centro.

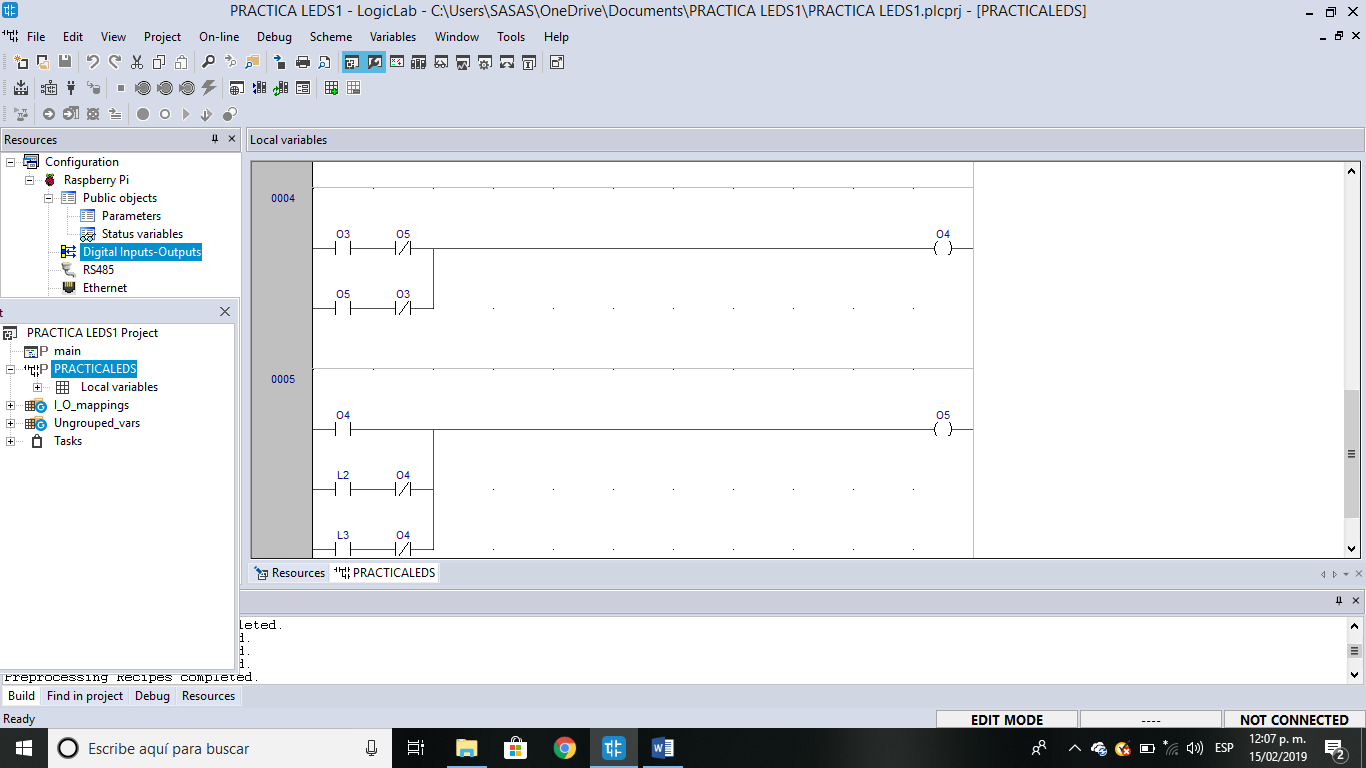
Esos son nuestros objetivos.

**Procedimiento:**

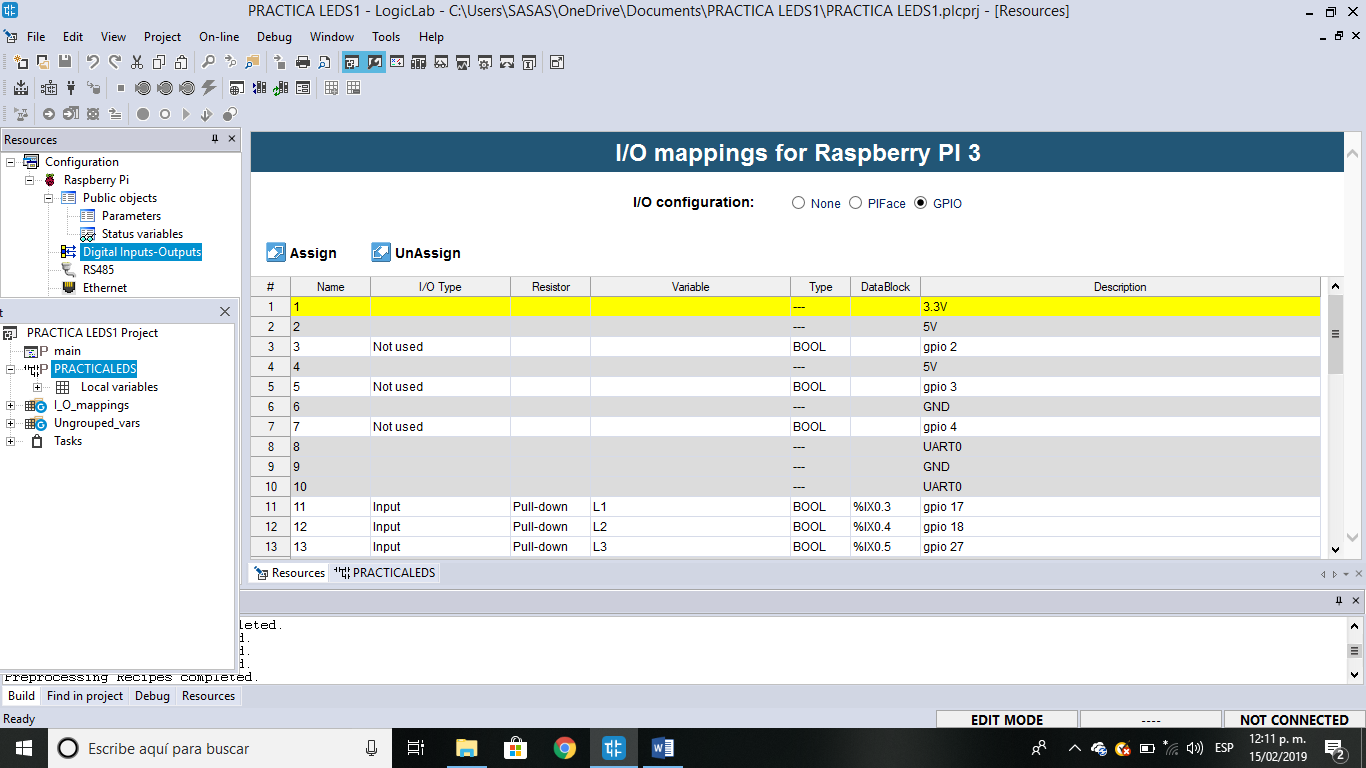
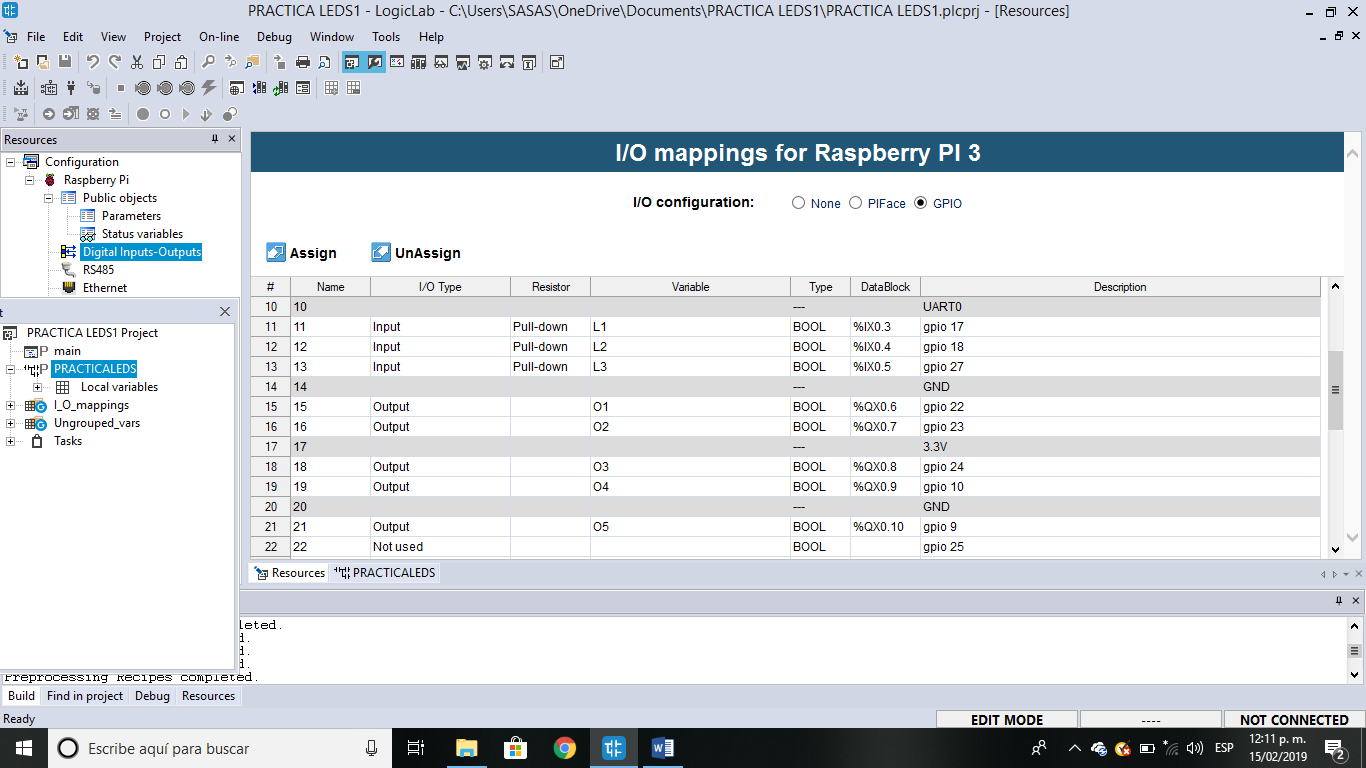
Utilizando las variables siguientes y el diagrama de escalera que se muestra a continuación, se pudo lograr que el objetivo de la practica se lograr, aunque de una manera un poco diferente a la que se creía en un principio.







Bien, ahora que ya está listo el diagrama, se puede pasar a la raspberry, ya saben, se conecta, compilas el código, lo descargas en la rasp, la rebooteas, y haces las conexiones necesarias de la raspberry, a la caja de control, siguiendo estas entradas y salidas:

Identificamos los pines establecidos en el código descargado, y conectamos todo y estos fueron los resultados.

Botón 1:



Botón 2:



Botón 3:



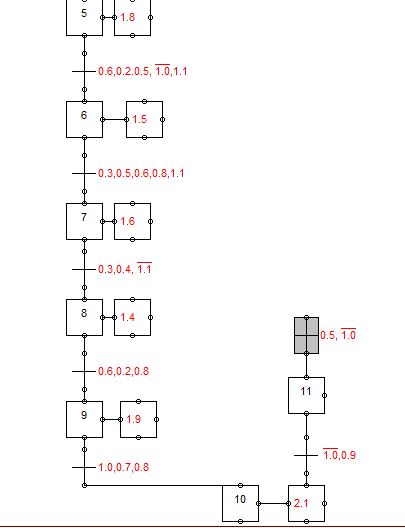
Conclusión:

Pude desarrollar un poco mas mis conocimientos acerca de cómo funciona un diagrama en escalera.

**Práctica 2**

Se nos pido realizar la simulación de una línea de producción con ciertos procesos, lo primero que realizamos fue un GRAFCET, para determinar cada actuador y sensor que necesitara la línea de producción.

Este fue el GRAFCET que realizamos.



Estos son los sensores y

actuadores que usamos:

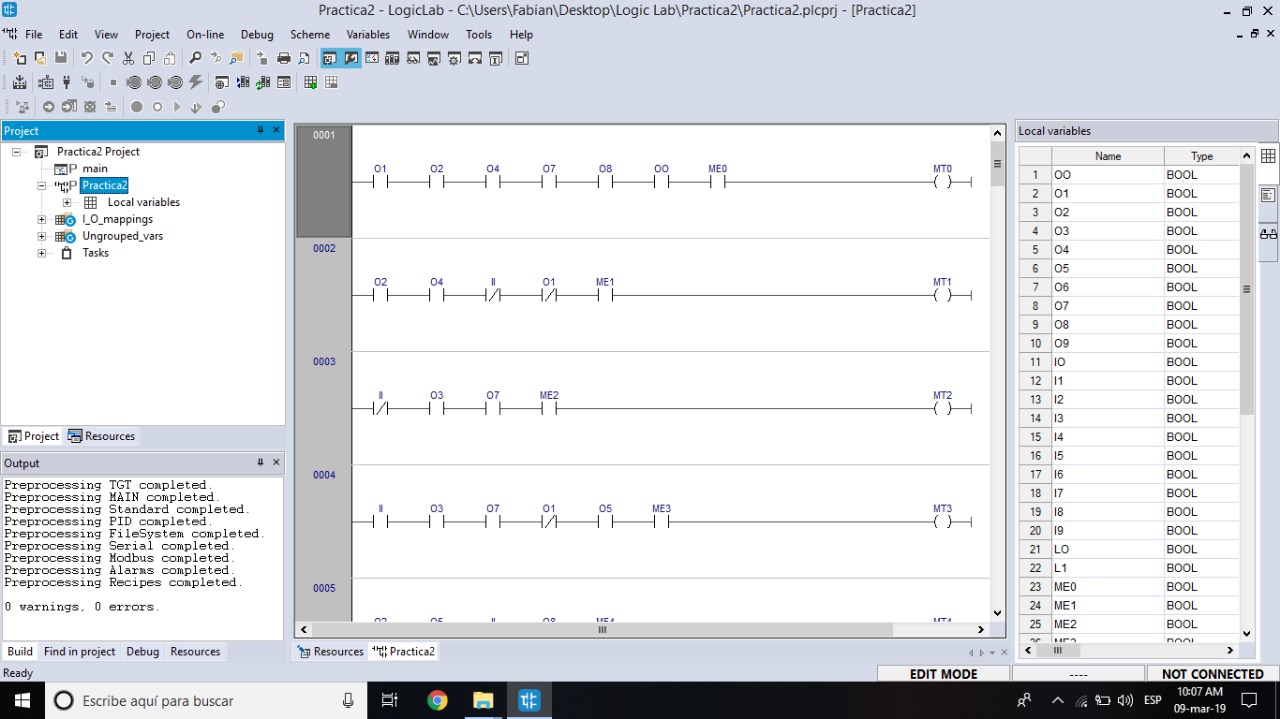
SENSORES

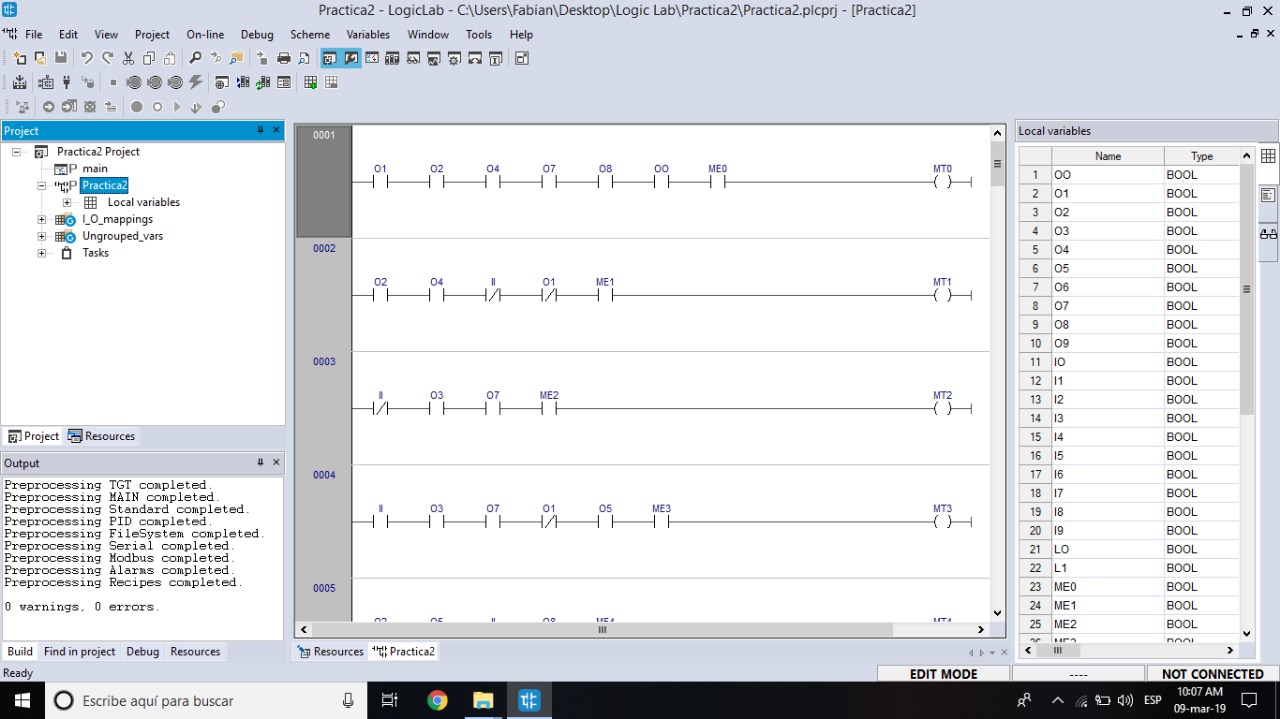
0.0 = sensor de pieza

* 1. = motor prendido
  2. = pistón arriba
  3. = pinza abierta
  4. = brazo izquierda
  5. = Brazo derecha
  6. = prensa arriba
  7. = sensor de pieza en la pinza

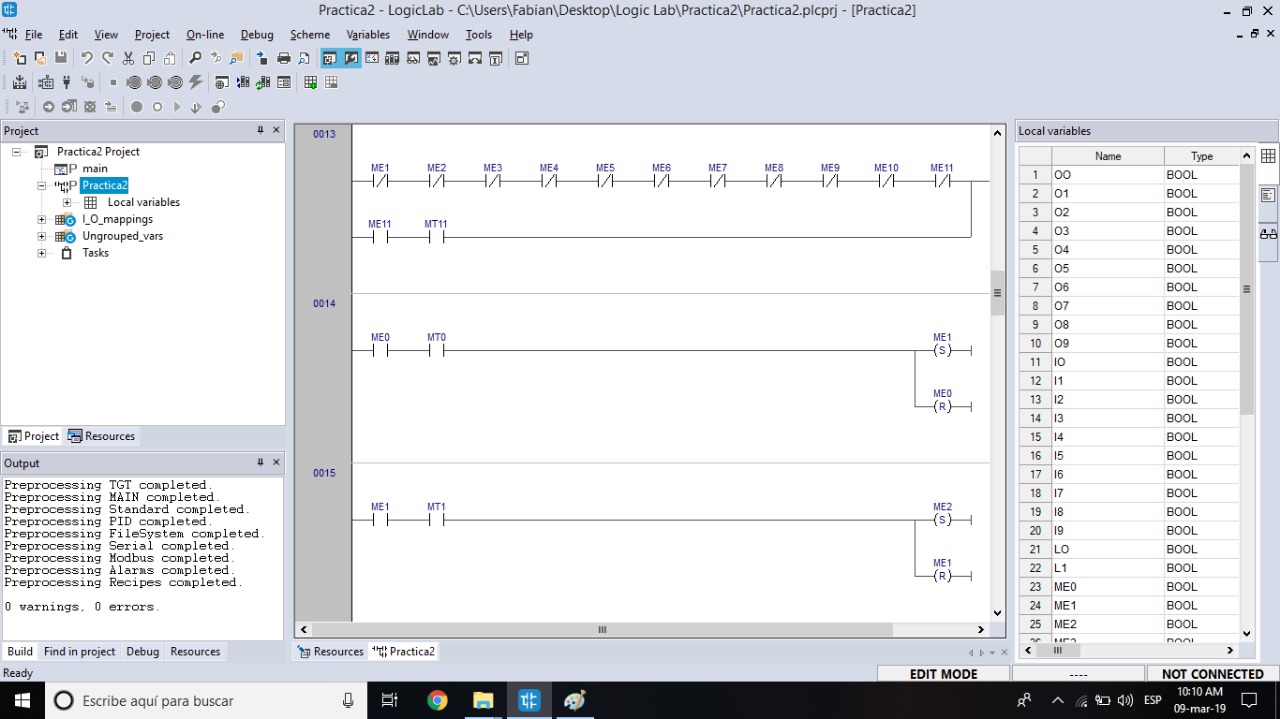
ACTUADORES

* 1. = prende motor
  2. = motor se apaga
  3. = sube pistón
  4. = se abre la pinza
  5. = brazo izquierda 90°
  6. = brazo derecha 90°
  7. = prensa sub

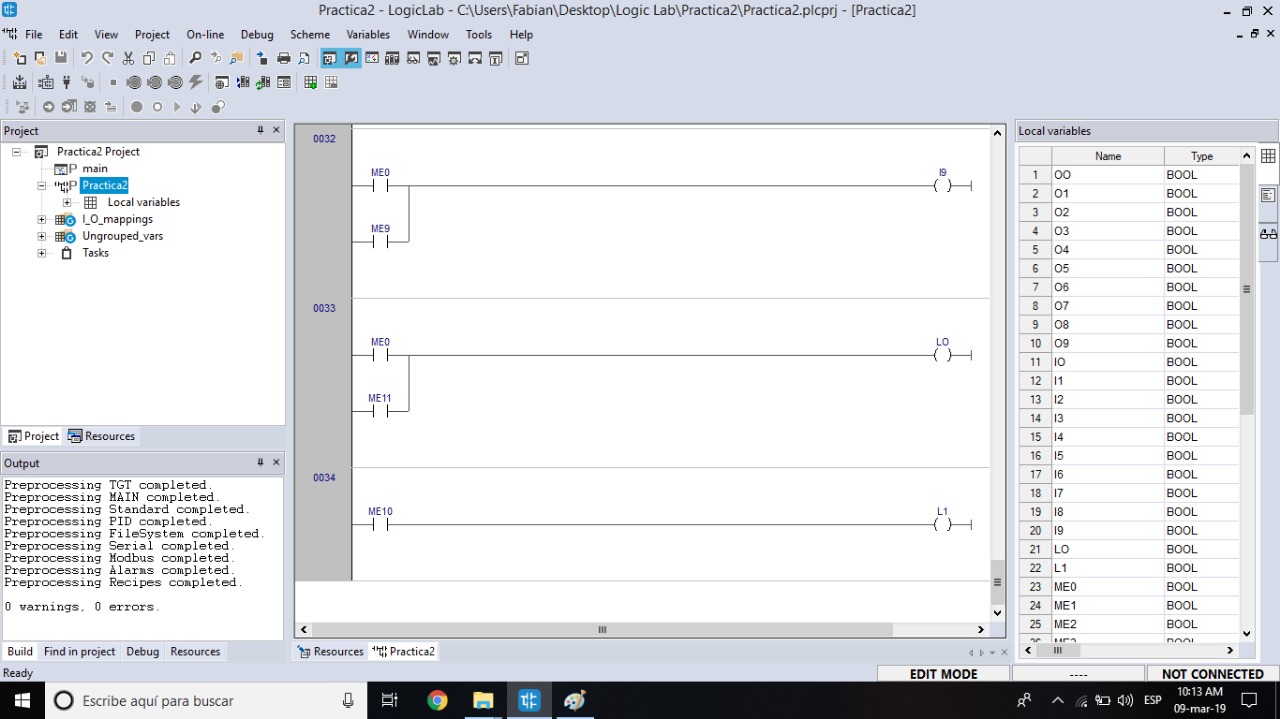


La línea roja indica las variables declaradas para poder

En esa parte de activa las memorias de estado para que realice las transiciones se pone que sensores usa cada memoria y así se activa la memoria de estado



Esta segunda parte es para activar de las memorias de estado las memorias de transición ósea que cada memoria de estado tiene que tener una trasmisión para poder empezar y tienes que desactivar una activar otra ya que no se puede tener más de una memoria de estado prendida se pone la condición que cuando todas las memorias de estado estén desactivadas se active la primera transición, la segunda parte solo se dice que cundo estén activas la memoria de estado cero y la uno se active la transición 2 y le ponemos un reset a la transición 1 y así con todas.



Por último se realiza esta condición para que regrese a home (la parte inicial)

Estando así se compila y se comienza a simular.

Conclusión.

Cuando desarrollas la habilidad de en método Grafcet, se hace posible el crear una línea de secuencias divididas entre señales de entrada y de salida, que al final de cuentas llegan a un objetivo en común.

**Práctica 3**

ENTRADA VISITANTE Y LOCAL

**INTRODUCCIIÓN**

En esta práctica se debe realizar una un circuito que consta en una entrada semiautomática para un condómino o fraccionamiento, ya que debe tener por lo menos 2 funciones de visitante y residente, mediante cada uno de estas opciones entre con una mayor facilidad al lugar. Debe ser realizado por medio de un mapa grafcet y construido y simulado en logiclab.

**Propósito**

Esta práctica se deben reforzar y reafirmar los conocimientos adquiridos previamente por las anteriores prácticas y actividades ya previamente realizadas, tomando en cuenta que se debe entregar en tiempo y en forma de manera correcta y funcional con un poco de ayuda del docente.

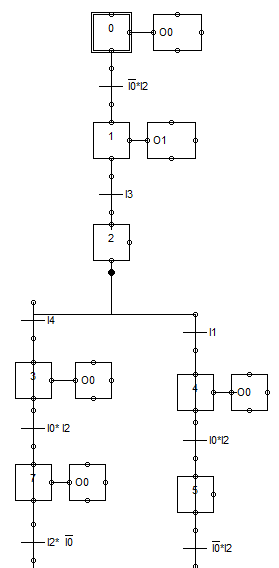
**Practica**

[](https://www.google.com.mx/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwi59dqylOvgAhVNXKwKHbYPCdkQjRx6BAgBEAU&url=https:/articulo.mercadolibre.com.mx/MLM-556298587-barrera-vehicular-de-control-de-acceso-pluma-de-45-metros-_JM&psig=AOvVaw0iiyXPLP-f4f00NNQU2boH&ust=1551880710486843)

|  |  |
| --- | --- |
| [IMG_256](https://www.google.com.mx/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjczt6nlOvgAhUBbK0KHY3CCzoQjRx6BAgBEAU&url=http:/wikifab.dimf.etsii.upm.es/wikifab/index.php/Sistema_de_Control_de_un_Aparcamiento_08001&psig=AOvVaw0iiyXPLP-f4f00NNQU2boH&ust=1551880710486843) | [IMG_256](https://www.google.com.mx/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj9-KL-levgAhVwmK0KHVVyDJ0QjRx6BAgBEAU&url=http:/itsmyblogmechatronics.blogspot.com/2015/04/car-park-barrierscoin-counters.html&psig=AOvVaw2CjaFyB-VdiV575Ue4S674&ust=1551881159203505) |

Lo primero que hicimos fue realizar el diagrama de GRAFCET el cual nosotros agregamos una modalidad de visitante y la forma que se solicita en la imagen, el GRAFCET que hicimos cuenta con 5 entradas y 2 salidas.

**Salidas:**

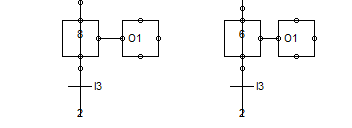
 - O0= barrera arriba,

- O1= barrera abajo

**Entradas:**

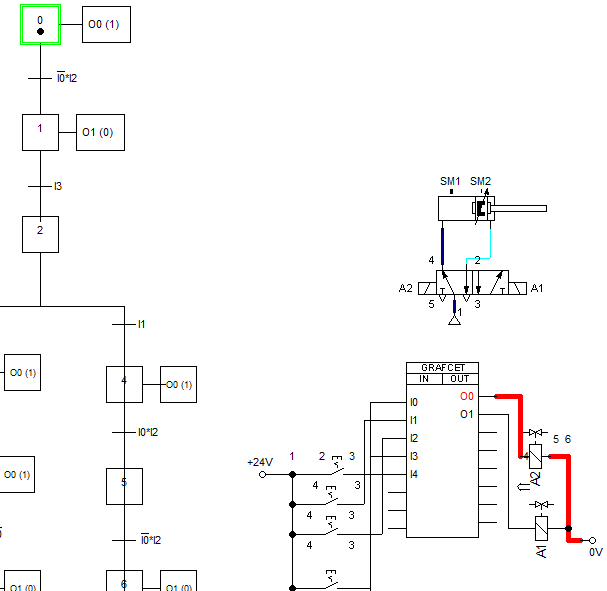
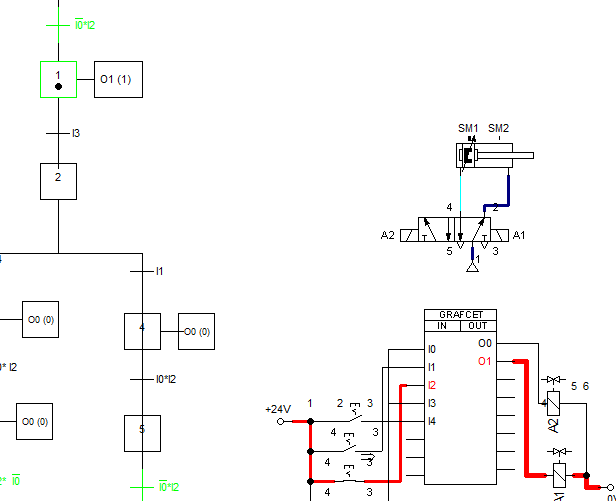
- I0 detector de masa

- I1 tag

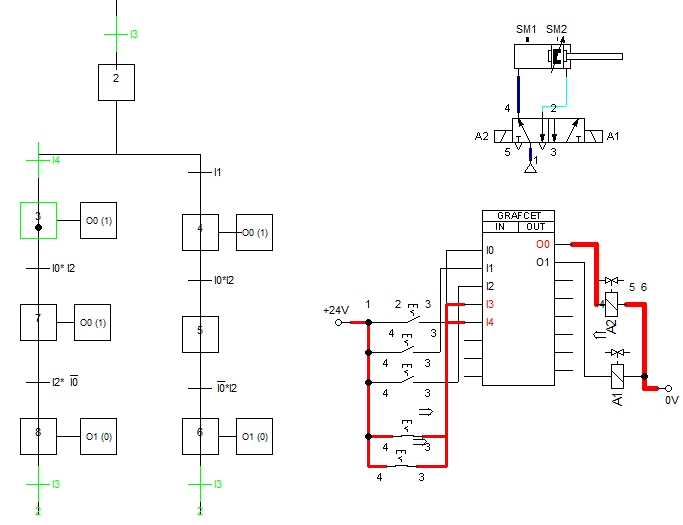
 - I2 = barrera arriba (sensor)

- I3 = barrera abajo (sensor)

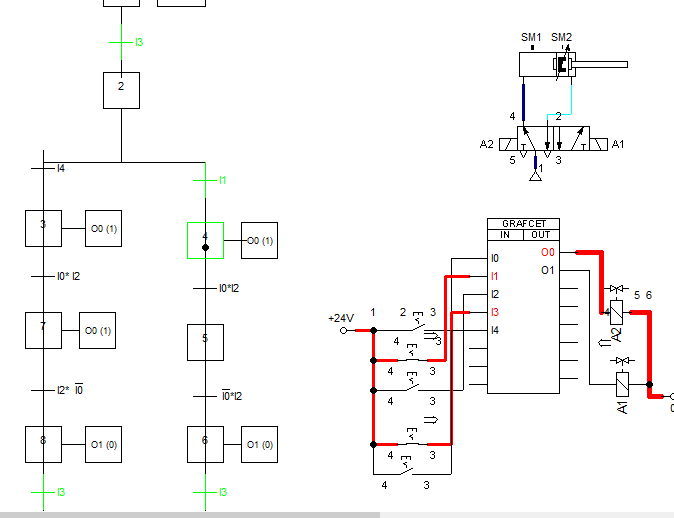
- I4 = visitante

La parte principal home inicia la barrera arriba y luego baja.

De ahí se realizan las dos opciones según lo que sea residente o invitado

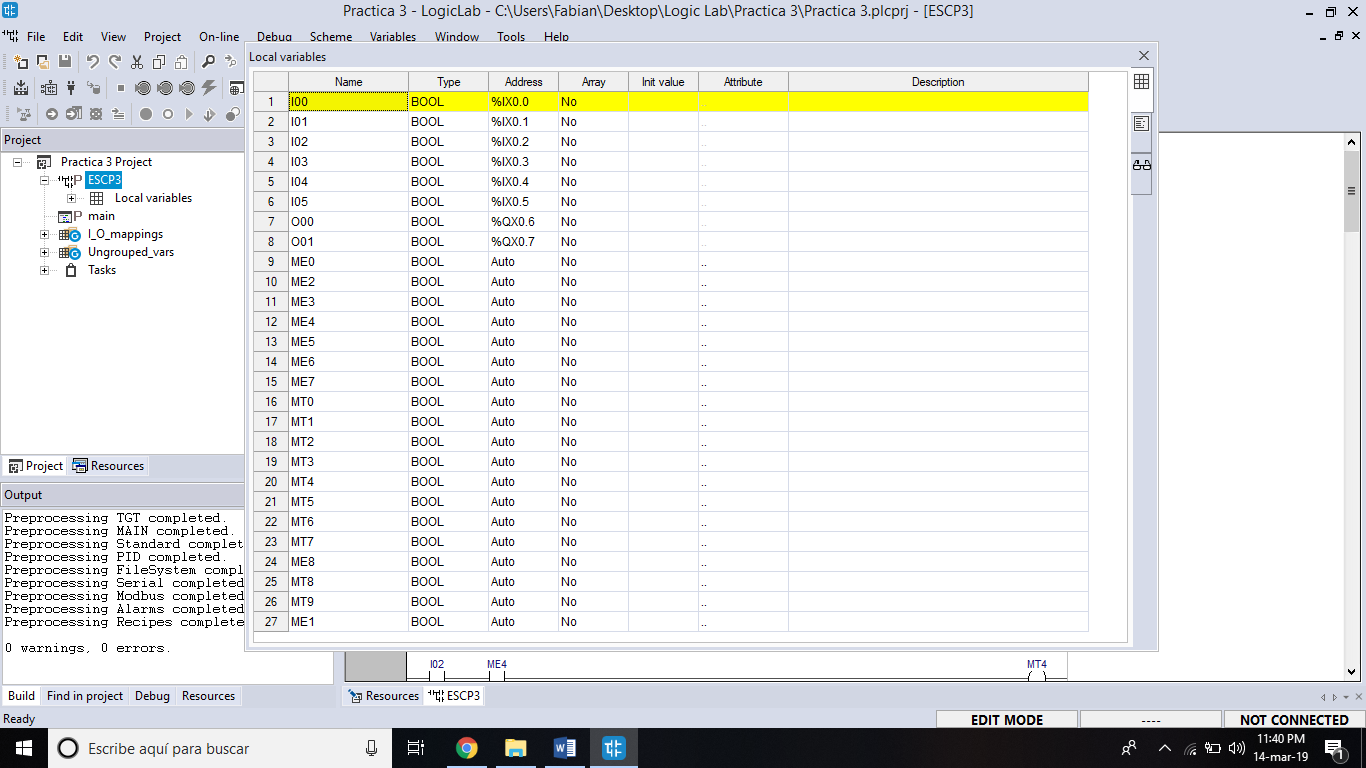
INVITADO

RESIDENTE

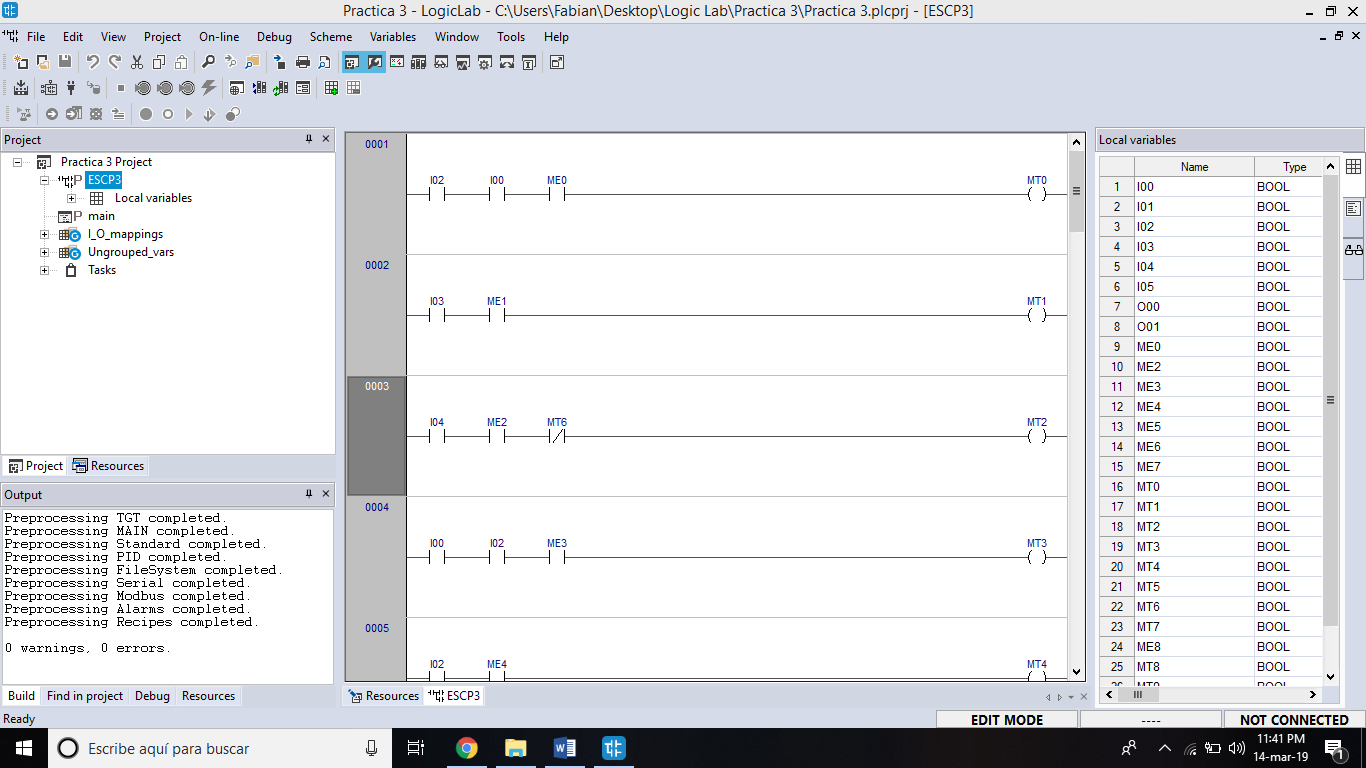


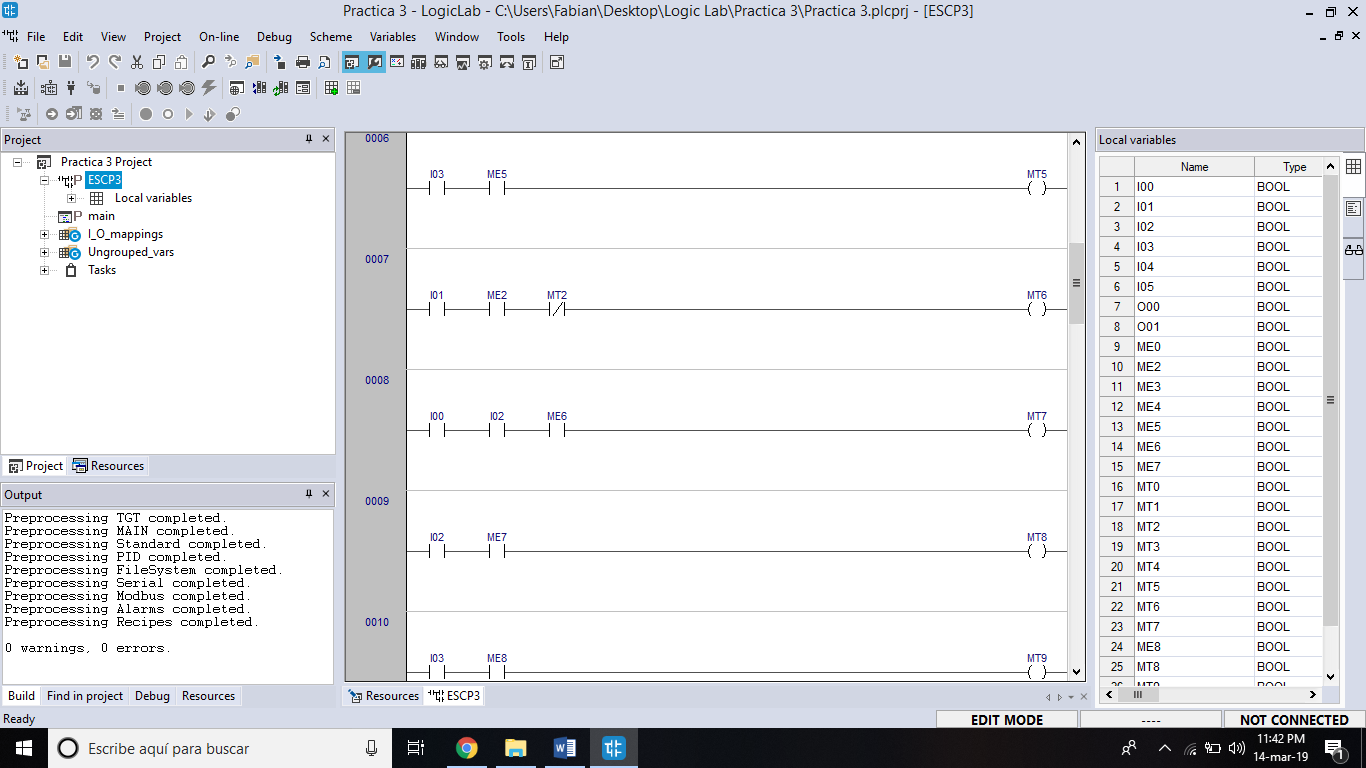
Una vez aprobado el mapa anterior por el docente procedemos al armando y simulación de este en el PLC logiclab

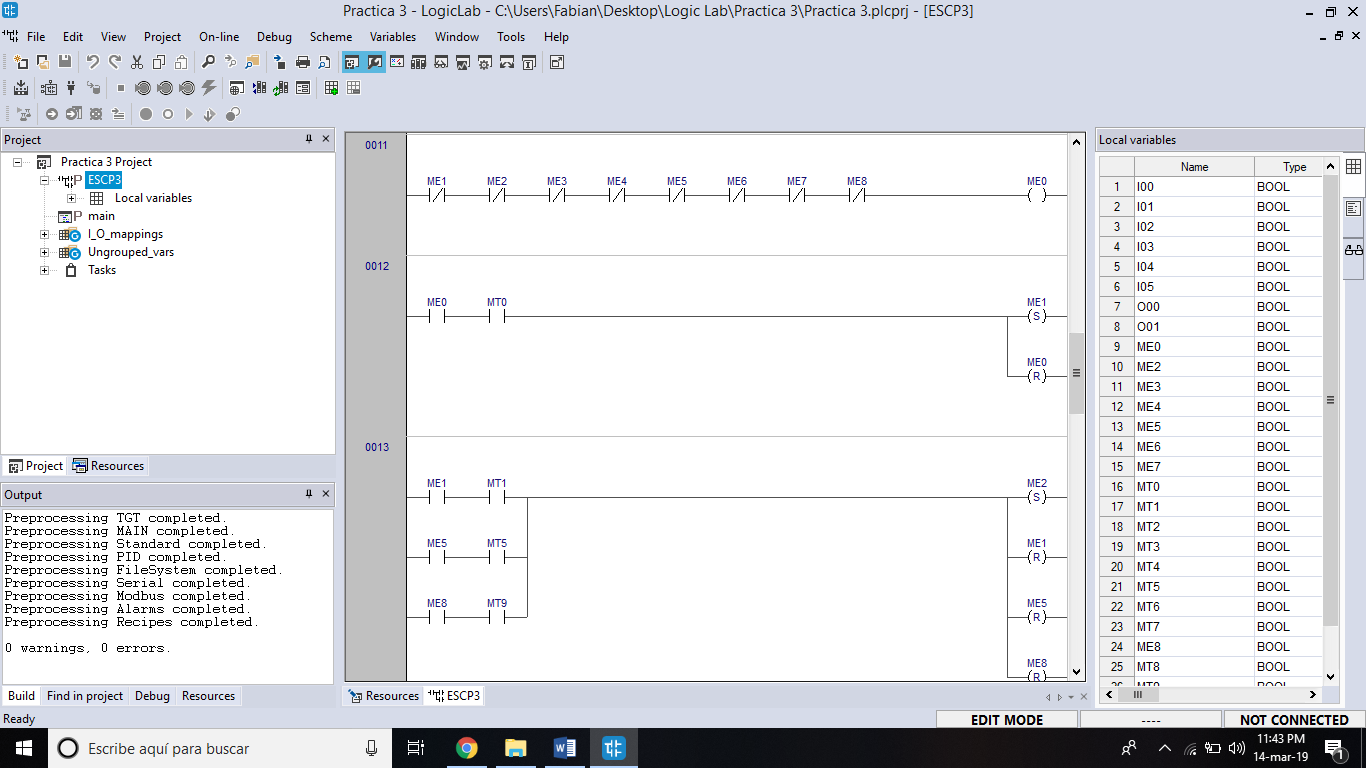
Declaramos entradas, Salidas y memorias de estado como de transición.

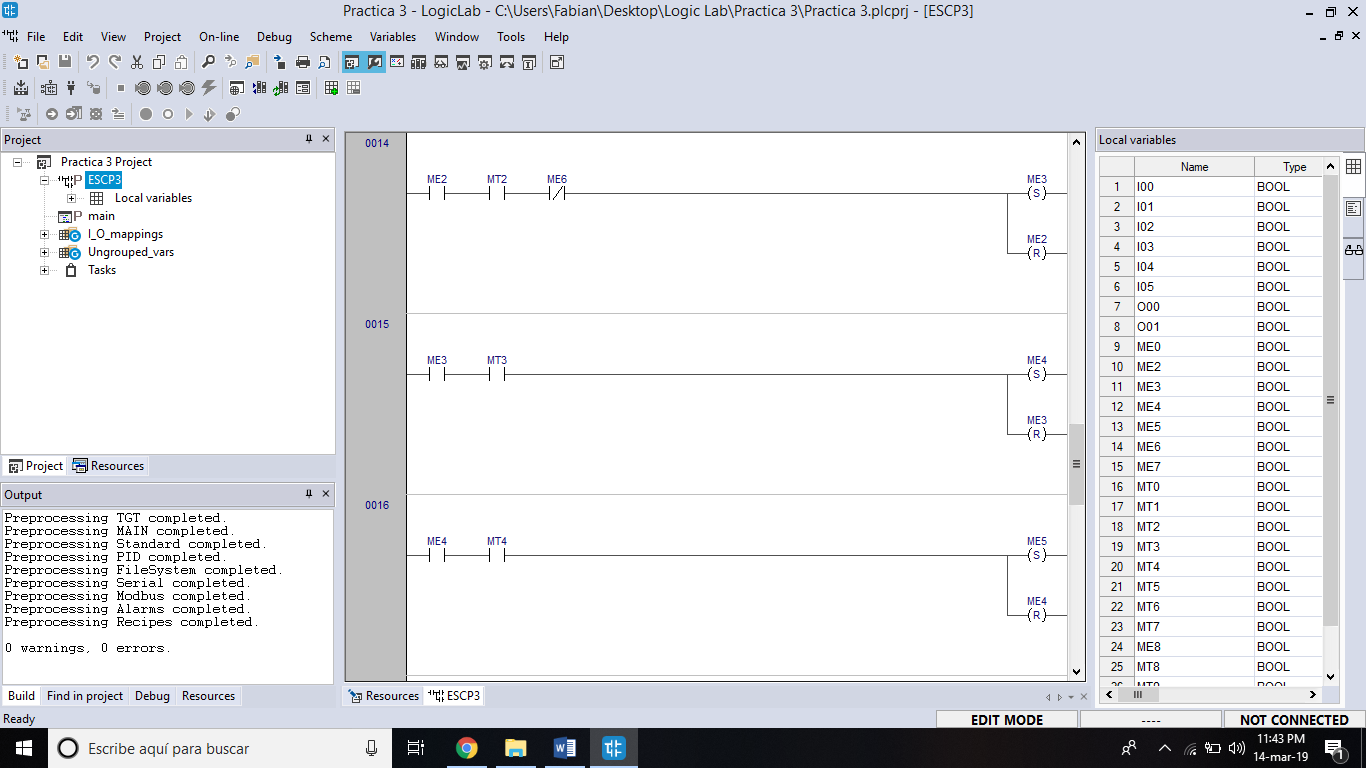


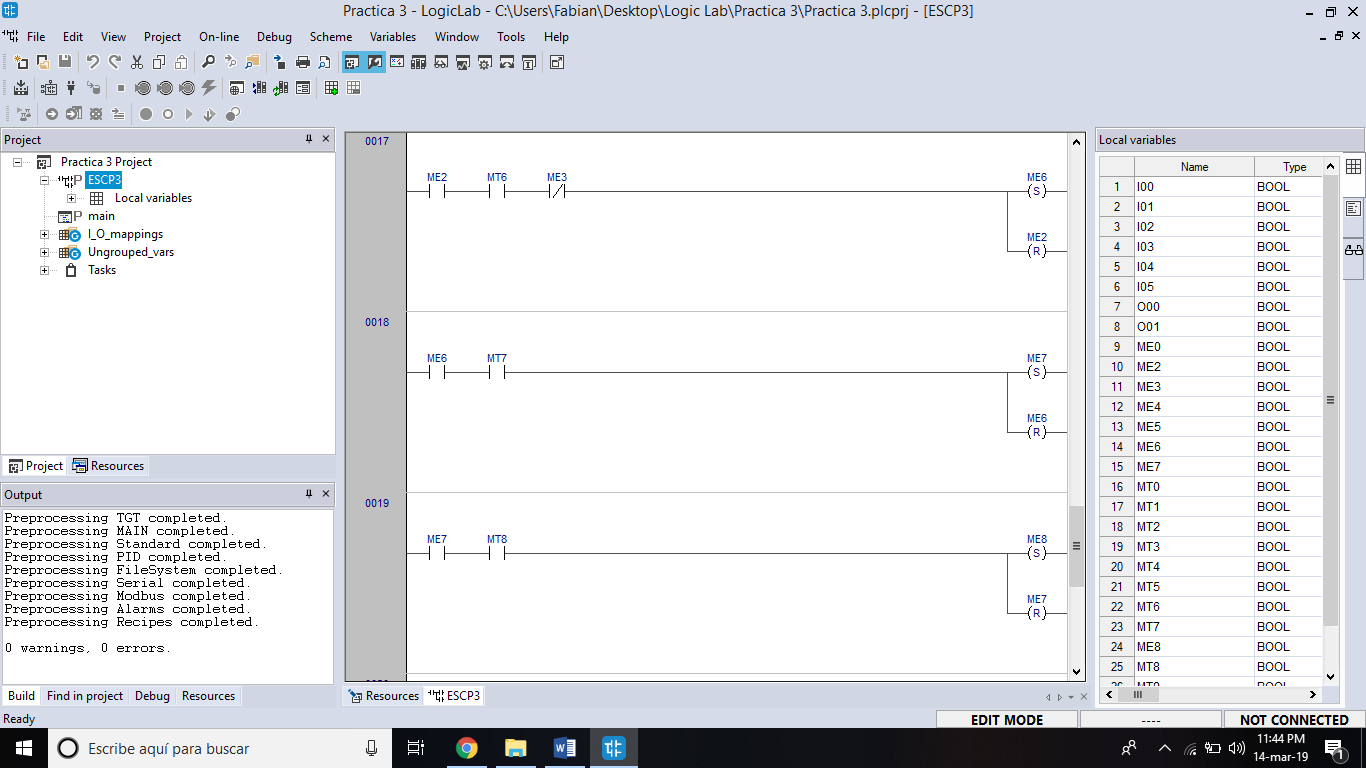
Y procedemos con el armado en el diagrama de escalera.

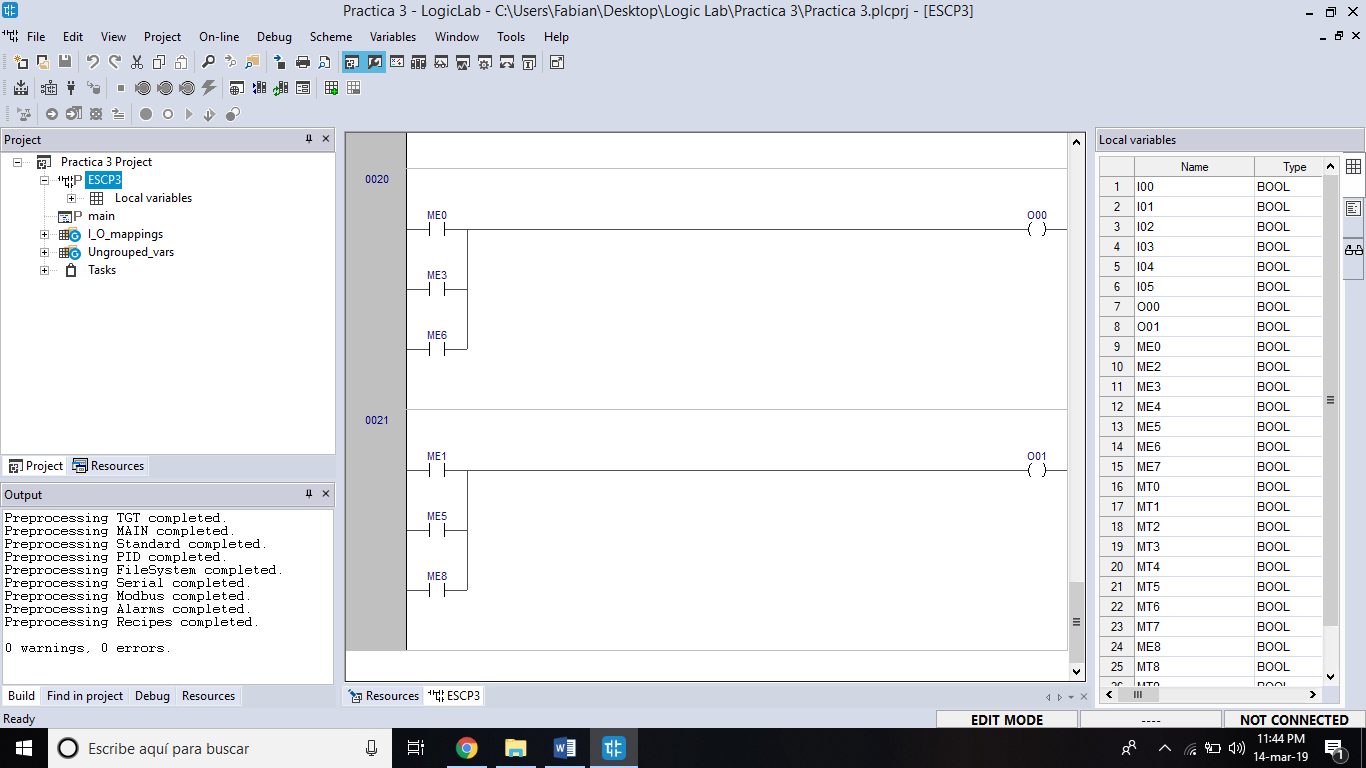












Conclusión:

En la industria nos funcionaria mucho ya que se usa mucho la programación en diagramas.

**Tarea1**

***Controladores Lógicos programables IEC 61131-3***

**Lenguaje escalera** (LD - Ladder Diagram), gráfico.

*El lenguaje Ladder, diagrama de contactos, o diagrama en escalera, es un lenguaje de programación gráfico muy popular dentro de los autómatas programables debido a que está basado en los esquemas eléctricos de control clásicos*

**Diagrama de bloques de funciones** (FBD - Function Block Diagram), gráfico.

*El diagrama de bloques de funciones, o Function Block Diagram (FBD) es un lenguaje gráfico para*[*controladores de lógica programable*](https://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable)*,​ que describe la función entre variables de entrada y variables de salida, misma que puede ser descrita como un conjunto de bloques. Las variables de entrada y salida están conectadas a bloques por líneas de conexión.*

*Las entradas y salidas de los bloques están conectadas mediante enlaces, los cuales pueden usarse para conectar dos puntos lógicos del diagrama, ya sea una variable de entrada con una entrada del bloque, una salida de un bloque con una entrada de otro bloque, o una salida de un bloque con una variable de salida.*

**Textoestructurado** (ST - Structured Text), textual.

*Structured Text es un lenguaje de marcas ligero creado para escribir textos de manera cómoda y rápida. Tiene la principal ventaja de que ese texto puede usarse para generar documentos equivalentes en HTML, TeX, docbook u otros lenguajes.*

*Actualmente se usa más reStructuredText, que es una revisión que mejora y amplía StructuredText.*

**Lista de instrucciones** (IL - InstructionList), textual.

*Es un lenguaje de bajo nivel y se asemeja bastante al lenguaje ensamblador. Las variables y las llamadas a funciones están definidas por elementos comunes del estándar IEC 61131-3, entonces varios lenguajes pueden ser usados en el mismo programa.*

*El control de programa se logra con «saltos» y llamadas a funciones (subrutinas con parámetros opcionales).*

*El formato de archivo se estandarizó a XML por PLCopen.*

**Bloques de función secuenciales (**SFC - SequentialFunction Chart), con elementos para organizar programas de computación paralela y secuencial.

Sequentialfunction chart (más conocido con las siglas SFC) es un lenguaje de programación gráfico empleado en controladores lógicos programables (PLCs). Es uno de los cinco lenguajes especificados en el estándar IEC 61131-3 y está basado en el método GRAFCET (a su vez basado en la Red de Petri)

Ejemplo de un programa sencillo en SFC.

El lenguaje es bastante útil para programar y controlar procesos que se dividen en etapas secuenciales. Los principales componentes del lenguaje son:

Etapas con acciones asociadas.

Transiciones con condiciones lógicas asociadas.

Enlaces entre etapas y transiciones.

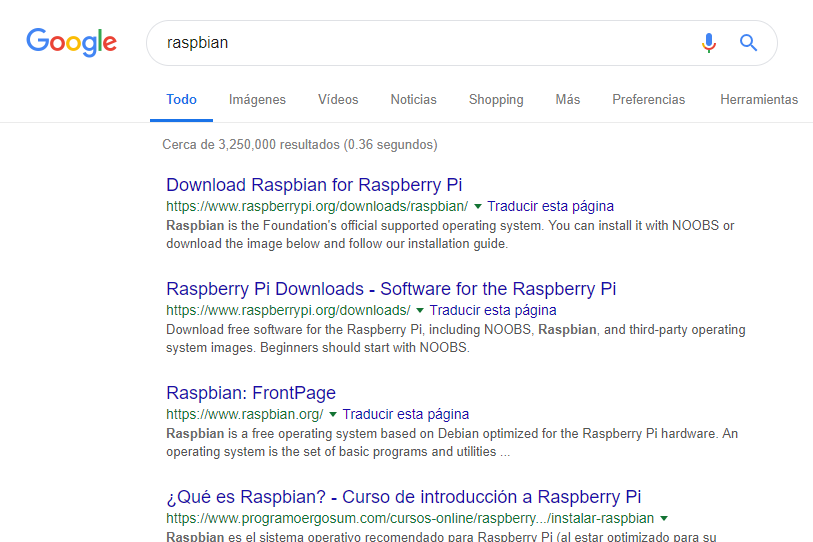
Las etapas en un programa en SFC pueden estar activas o inactivas. Las acciones de una etapa se ejecutan exclusivamente cuando dicha etapa está activa. Una etapa puede estar activa si dicha etapa es una etapa inicial especificada por el programador, o bien si se cumplieron las condiciones de una transición que lleva a dicha etapa. Las acciones asociadas a etapas pueden ser de muchos tipos, tales como activar o desactivar ciertas variables de salida. Es posible insertar acciones de un programa hecho en lenguaje Ladder dentro de un programa SFC (de hecho, comúnmente se hace esto para trabajar con variables de tipo entero).

**Tarea2**

**Instalación de “Rasbian”**

-Lo primero que realizamos fue conseguir una micro SD para hay instalar el “Rasbian” la memoria tiene que ser mayor de 4 GB ya que el programa pesa alrededor de 4 GB

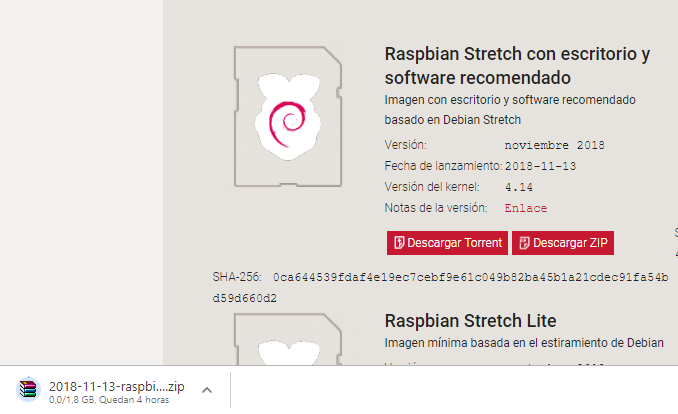
Para la micro sd la tienes que formatear con un programa en este caso usamos el SD CardFormatter para que los archivos del “Rasbian” se graben bien.

Buscamos en el navegador el programa “Rasbian” e ingresamos para poder instalarlo.

Ingresamos a la página irse a donde dice descargar, y escoger la opción de rasbian.

Al momento de seleccionar rasbian, seleccionamos la opción que dice:

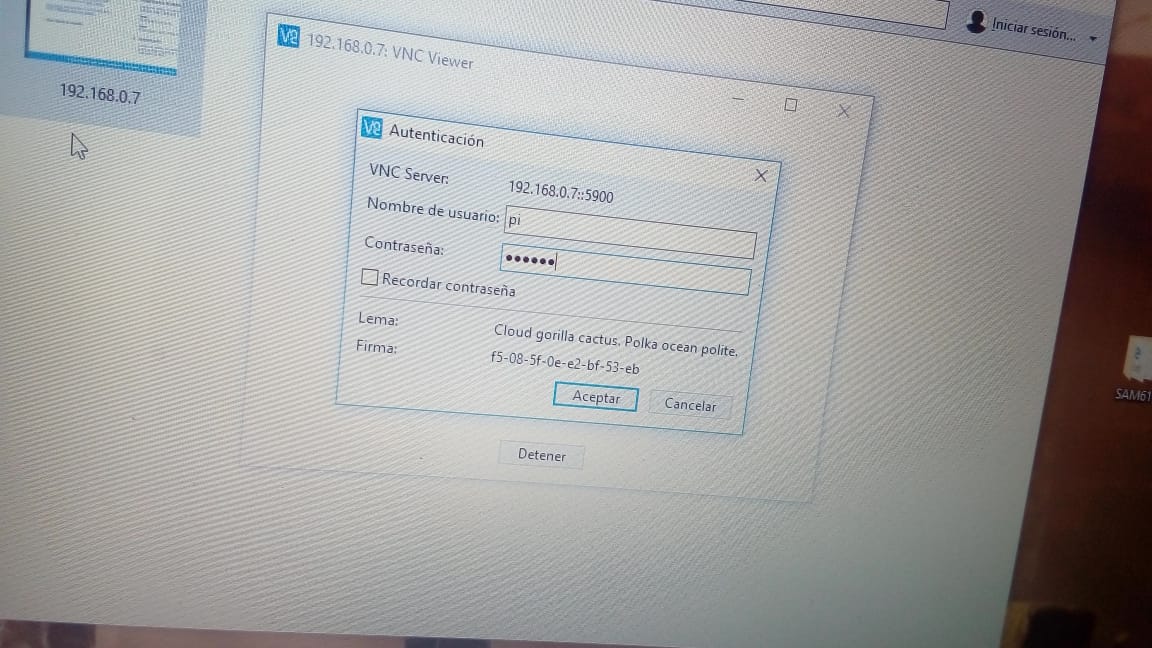
Raspbian Stretch con escritorio y software recomendado

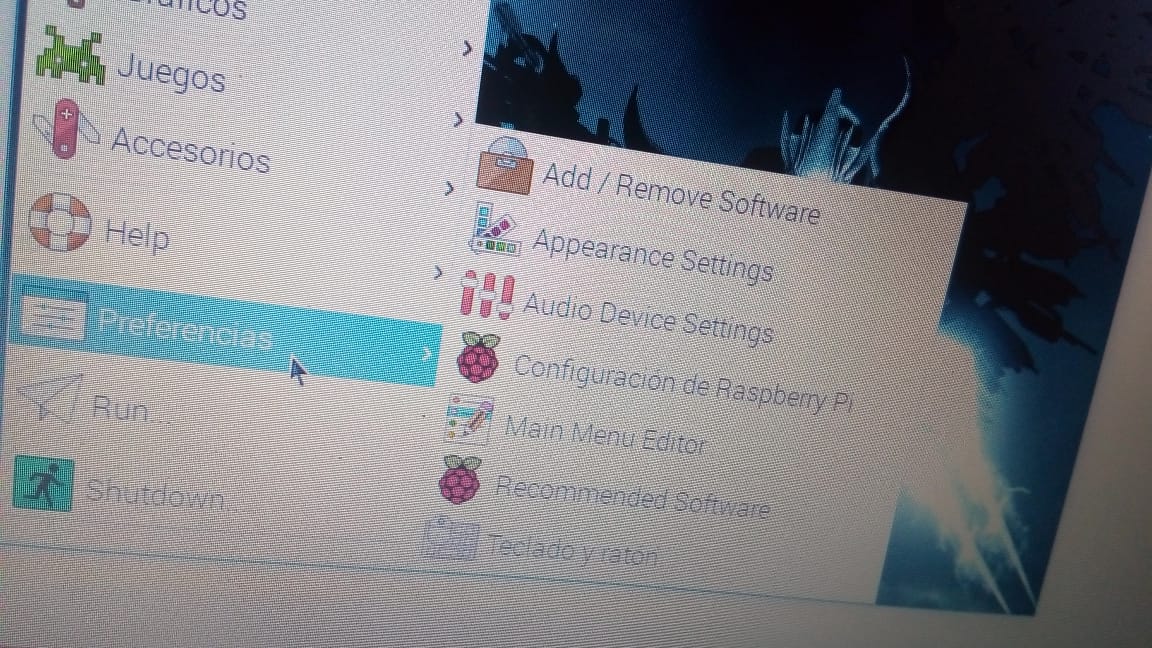
 Y descargar

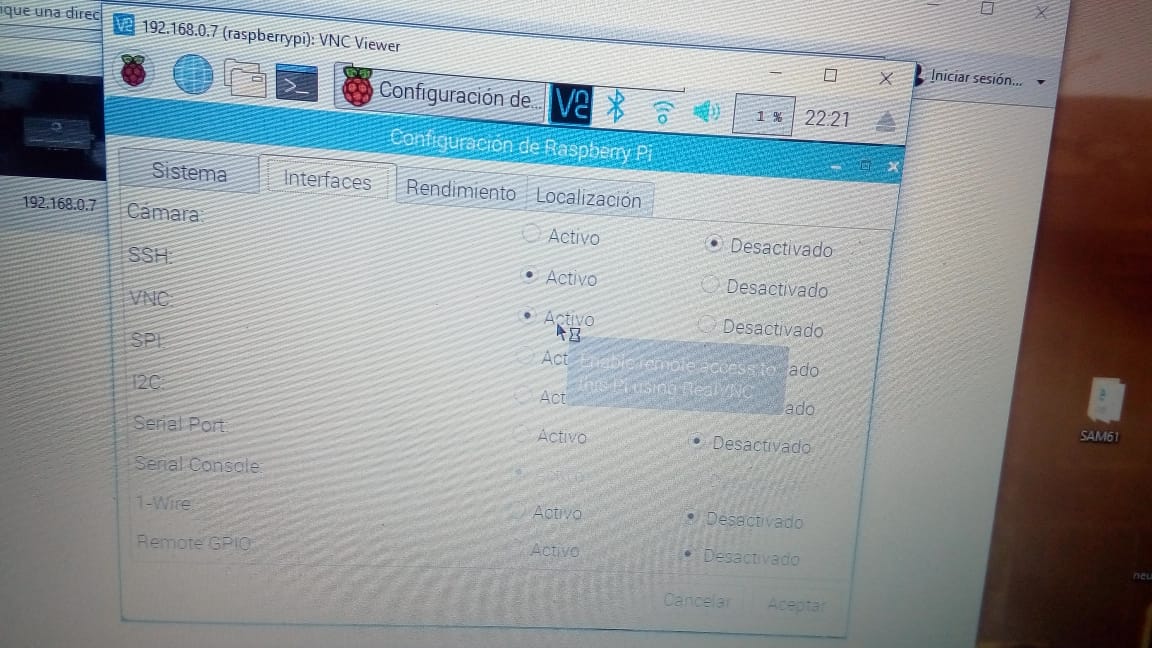
Teniendo el rasbian descargado colocamos la micro SD en la RASBERRY

conectamos la RB a un monitor por el puerto conectamos el teclado para poder manipular junto con el mouse una vez modificando el idioma te pide un usuario y una contraseña.

Te abre el menu de la Rasberry, cuando se abre el menú te vas a **“preferencias” “configuración de Rasberry”**

**“interfaz” y activar el VNC.** 

Configuración de la RASBERRY



Por último descargamos el programa VNC Viewer y conectamos con la dirección de la “RASBERRY” y ya se puede conectar a una pc.

