# **Practica 4: Control de elevador**



**CONTROLADORES LOGICOS PROGRAMABLES**

MECATRONICA 5°A

**MAESTRO**: MORAN GARABITO CARLOS

EDUARDO ROBLES VAZQUEZ

VICTOR GABRIEL TAPIA CASILLAS

**PROBLEMA:**

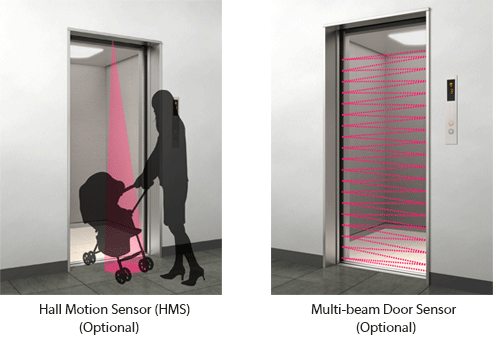
Diseñar un elevador de 2 pisos (planta baja, 1er piso y 2do piso) para ello debe tener un motor que funcione en 2 direcciones (arriba y abajo) teniendo en ambos sentidos 2 velocidades (alta y baja) además de un freno electro-mecánico (clutch), el cubo del elevador tendrá el sensor de puerta abierta y cerrada, el sensor de obstrucción de puerta (sensor de barrera), sensor de persona adentro [(sensor pirro eléctrico) si hay persona, se prende la luz], sensor de velocidad (para cuando va llegando al piso correspondiente, cambie a la velocidad 2) y sensor de llegada (cuando llegue, se detendrá el motor, y activará el freno), 1 botón para llamar el elevador funcionando de la siguiente manera:

1. Si no hay gente dentro del elevador, atenderá el llamado de inmediato
2. Si tiene gente y está desplazándose, y el piso de llamada está de paso, llegara a la llamada y luego seguirá la rutina, almacenando hasta 2 atenciones de llegada (memorizando a donde debe llegar), esto es, si la gente dentro del elevador, puso que quiere llegar al piso 2 y la llamada está en el piso 1, llegara por la gente del piso 1 y esta gente quiere ir a la planta baja, se moverá al piso 2, y después a la planta baja.
3. Si tiene gente y el piso no está de pasada, atenderá primero el lugar a donde debe llegar y luego se dirigirá al piso de llamada, esto es, si el elevador está en PB, y la gente va al piso 1 y la llamada llega del piso 2, el elevador llegara primero al piso 1 y después al piso 2.

También tendrá un sensor de falla eléctrica (si se va la luz, la planta de emergencia le mandara por medio de una entrada [bit] al PLC indicando que no hay luz y el PLC deberá llegar al piso más próximo, abrir la puerta, activar el freno y prender el foco de emergencia, quedando en esa posición hasta que la luz se haya reestablecido [el sensor de la planta de emergencia se apague]).

|  |  |
| --- | --- |
| IMG_256 | IMG_256 |
| IMG_256 | IMG_256 |

|  |  |
| --- | --- |
| IMG_256 | IMG_256 |



**OBJETIVO:**

Desarrollar un programa que controle un elevador de la forma indicada en el planteo del problema.

**DESARROLLO:**

Primero analizamos el problema propuesto y pasamos a definir nuestras salidas y sensores.

Sensores:

0.0 Puerta abierta  
0.1 Puerta cerrada  
0.2 Puerta obstruida  
0.3 Persona  
0.4 Velocidad  
0.5 Llegada  
0.6 Boton PB  
0.7 Boton P1  
0.8 Boton P2  
0.9 Falla eléctrica

0.10 Planta baja  
0.11 Planta 1  
0.12 Planta 2  
0.13 Boton E. PB  
0.14 Boton E. P1  
0.15 Boton E. P2  
0.16 Freno  
0.17 Motor arriba  
0.18 Motor abajo

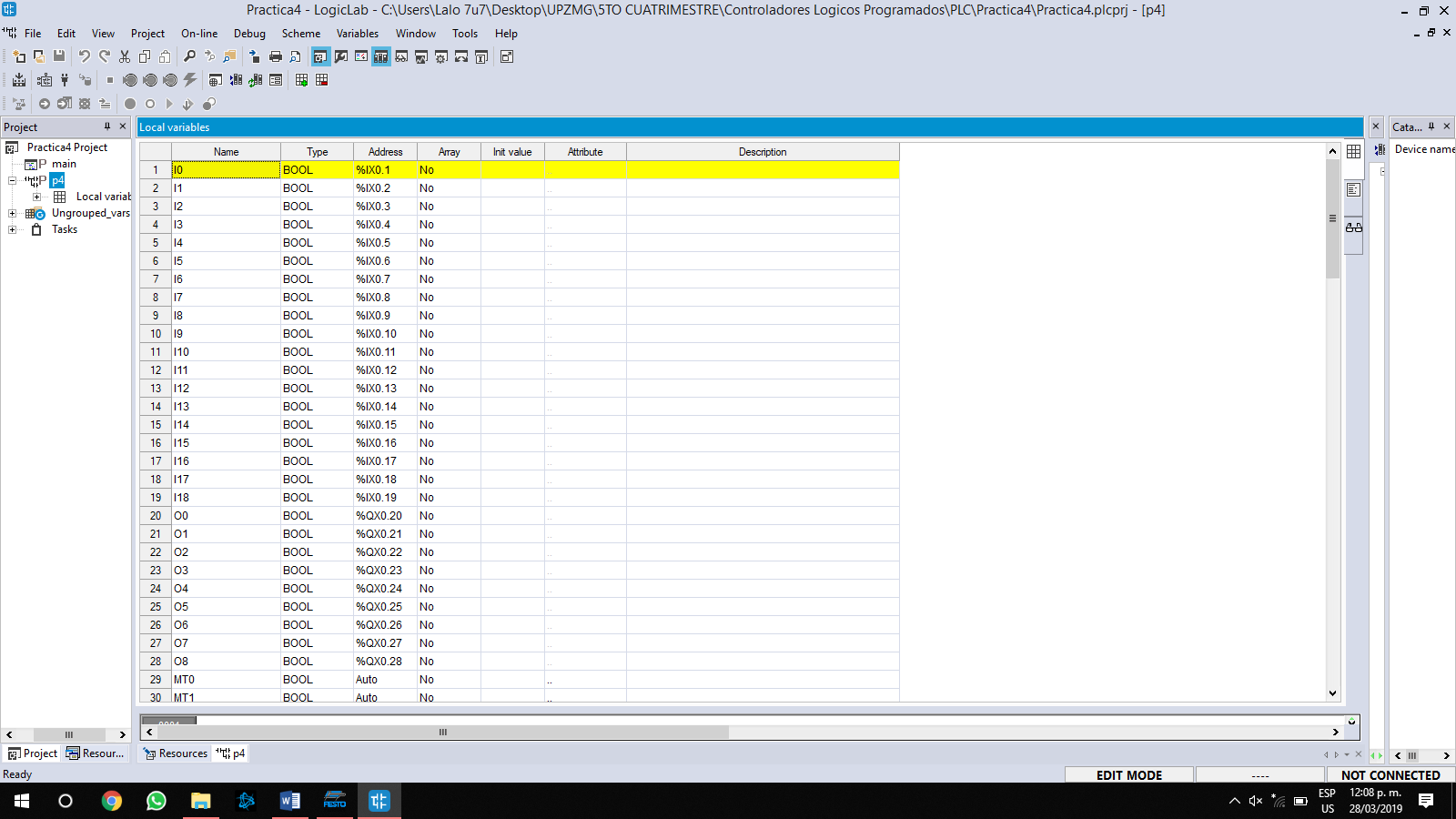
Actuadores

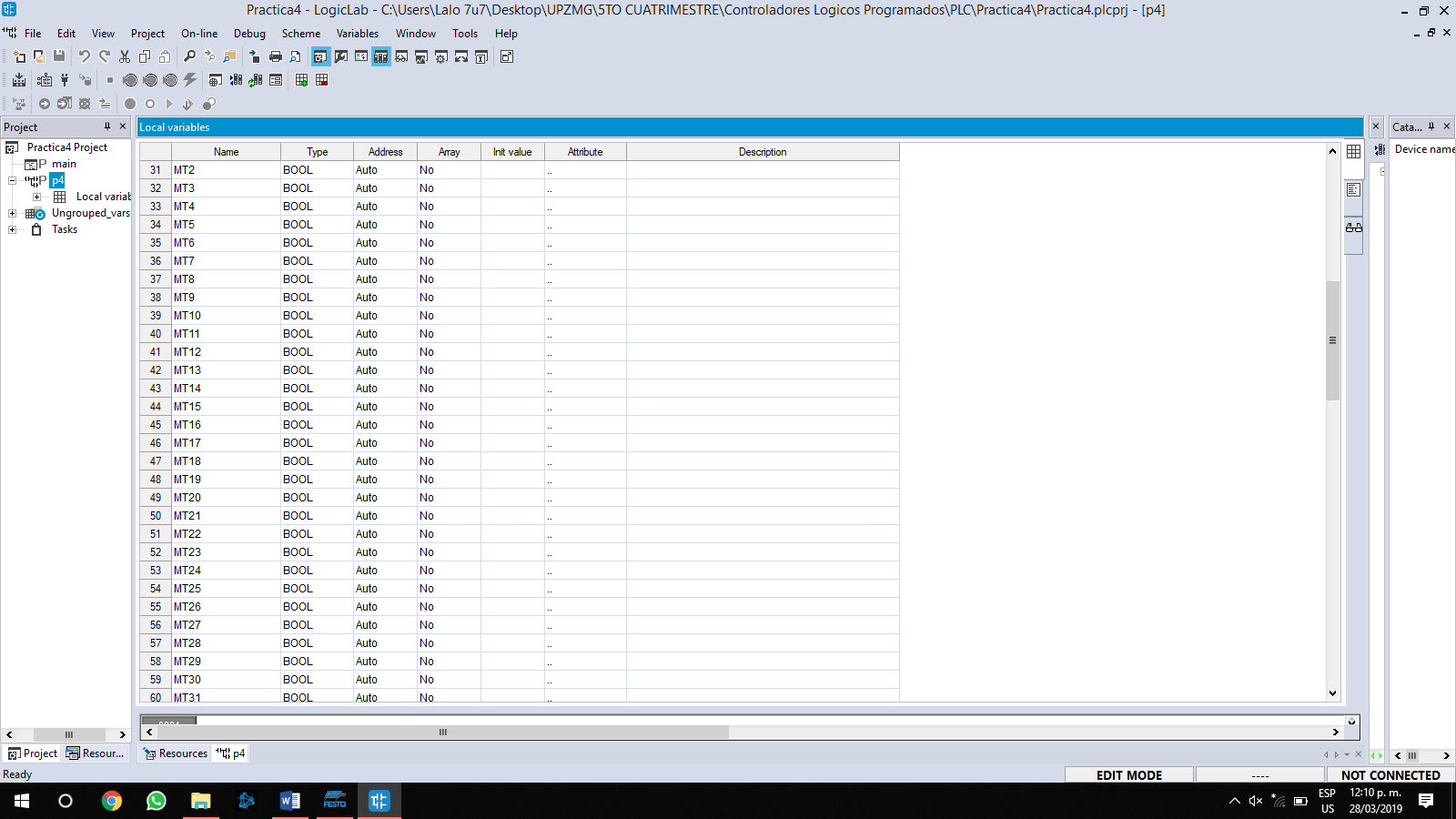
0.0 Motor arriba  
0.1 Motor abajo  
0.2 Velocidad alta  
0.3 Velocidad baja  
0.4 Freno  
0.5 Foco  
0.6 Puerta abierta  
0.7 Puerta cerrada  
0.8 Foco de emergencia

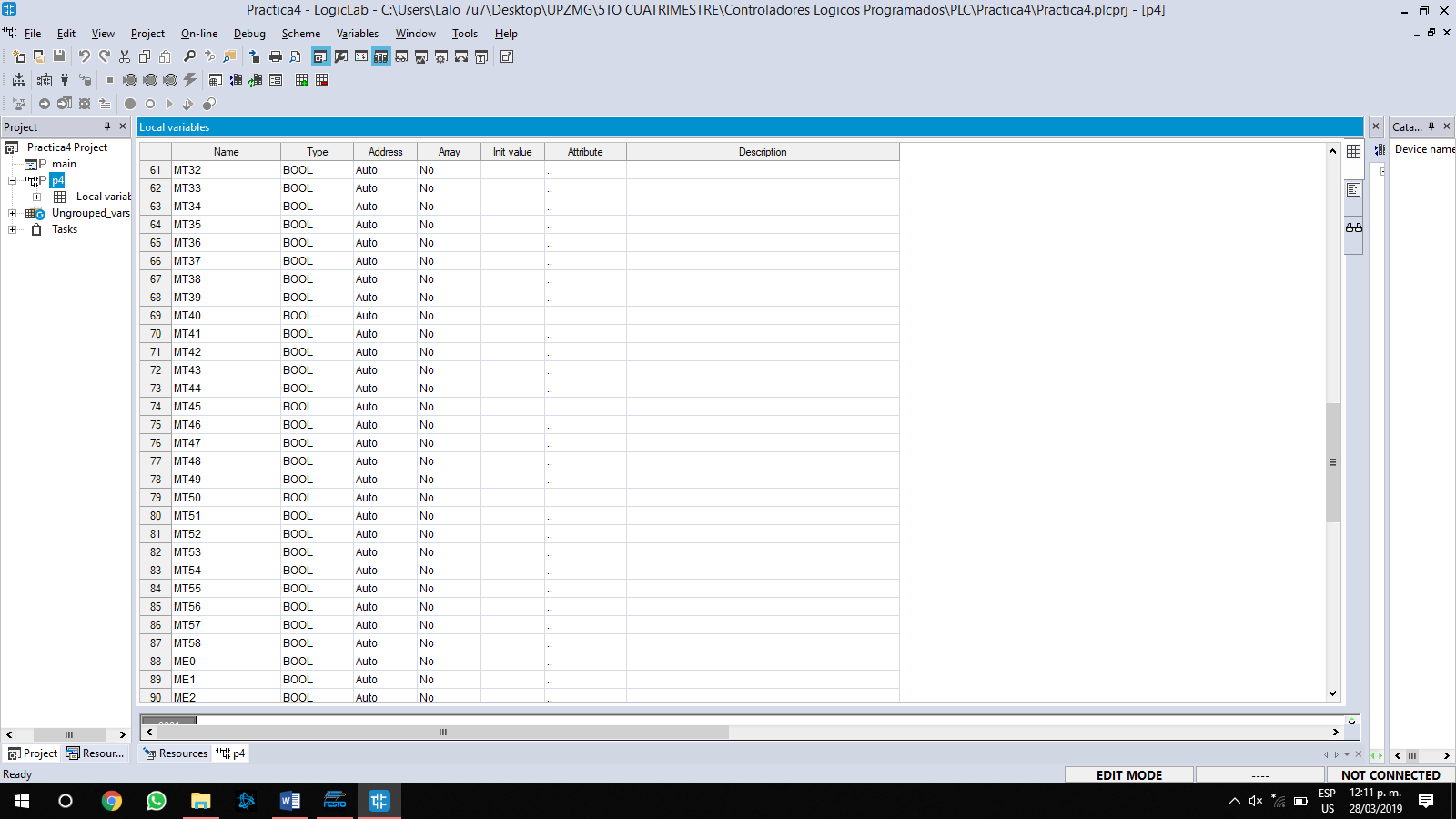
Ya que hemos definido las salidas y entradas pasamos a crear nuestro GRAFCET.

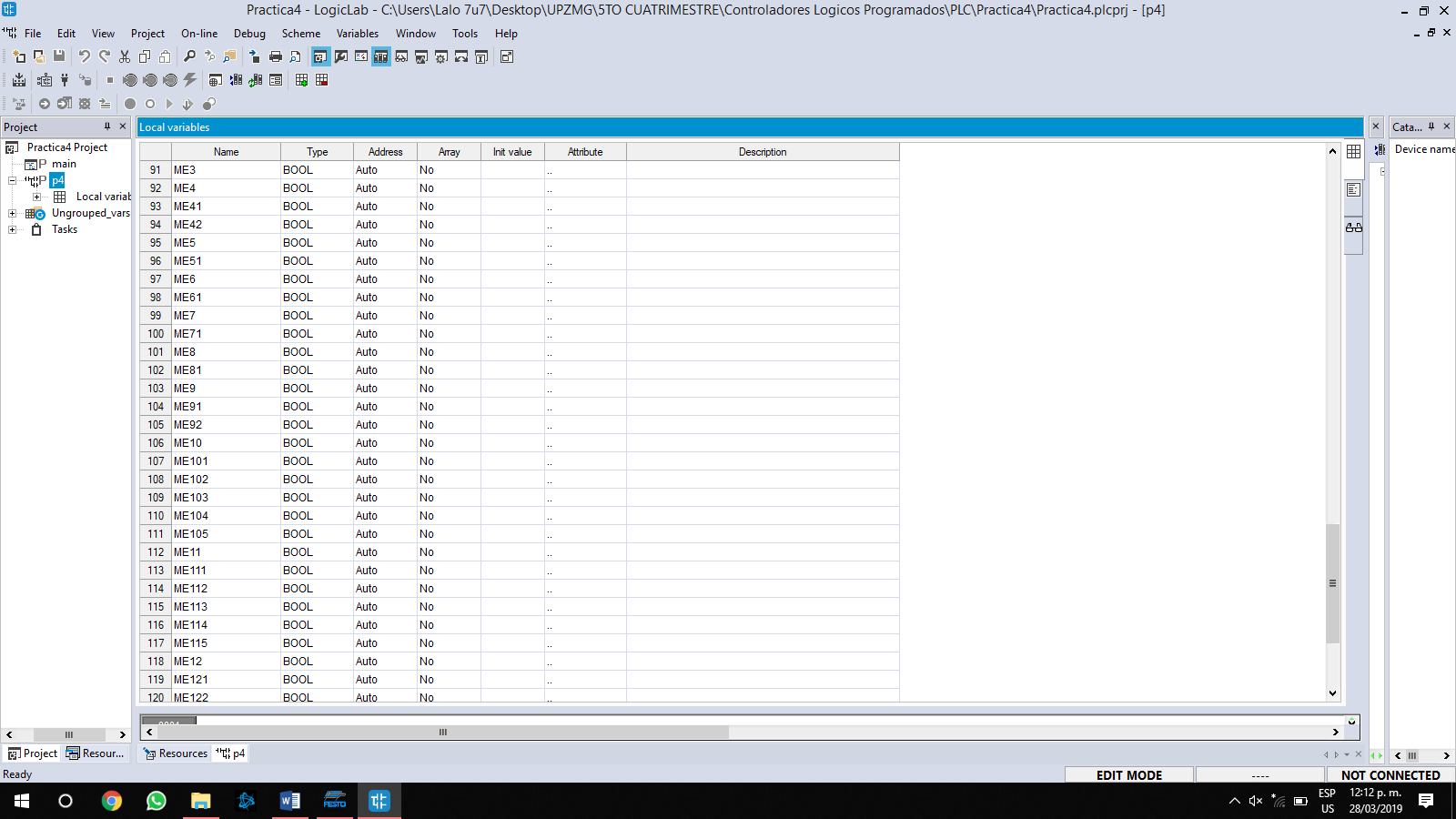


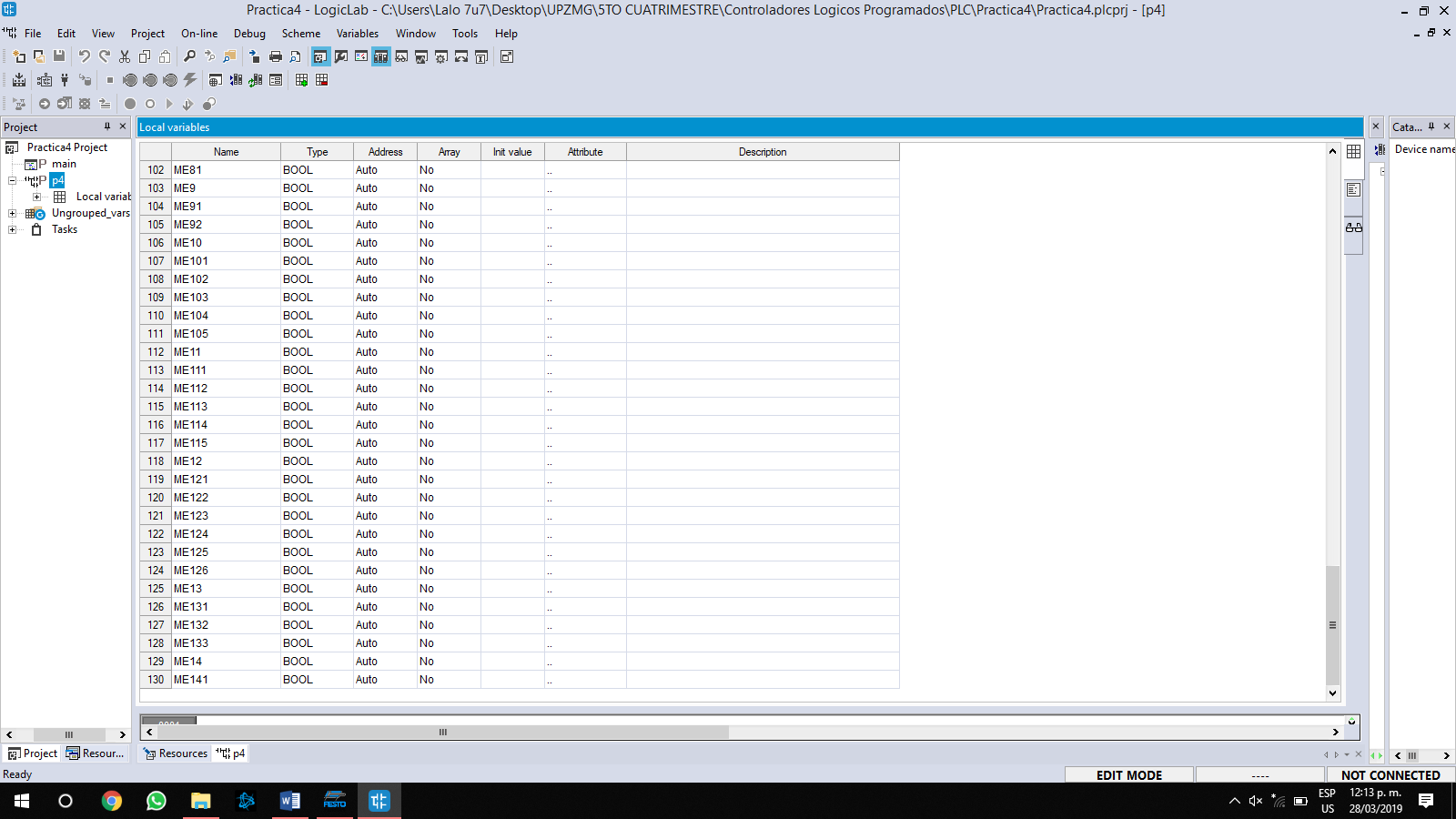
Posteriormente creamos el diagrama de escalera para el cual primero tendremos que dar de alta las variables que utilizaremos y su dirección correspondiente.

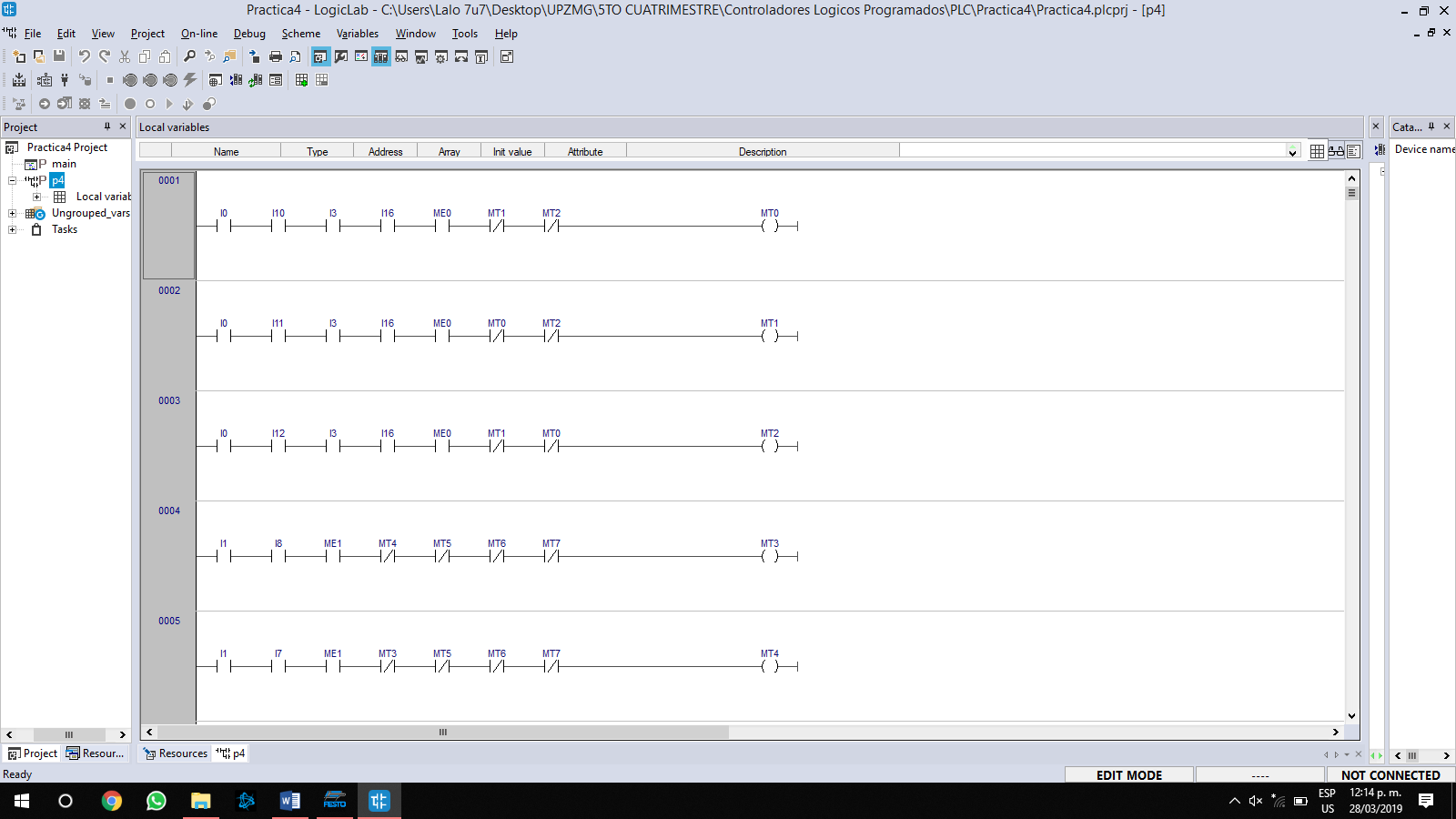




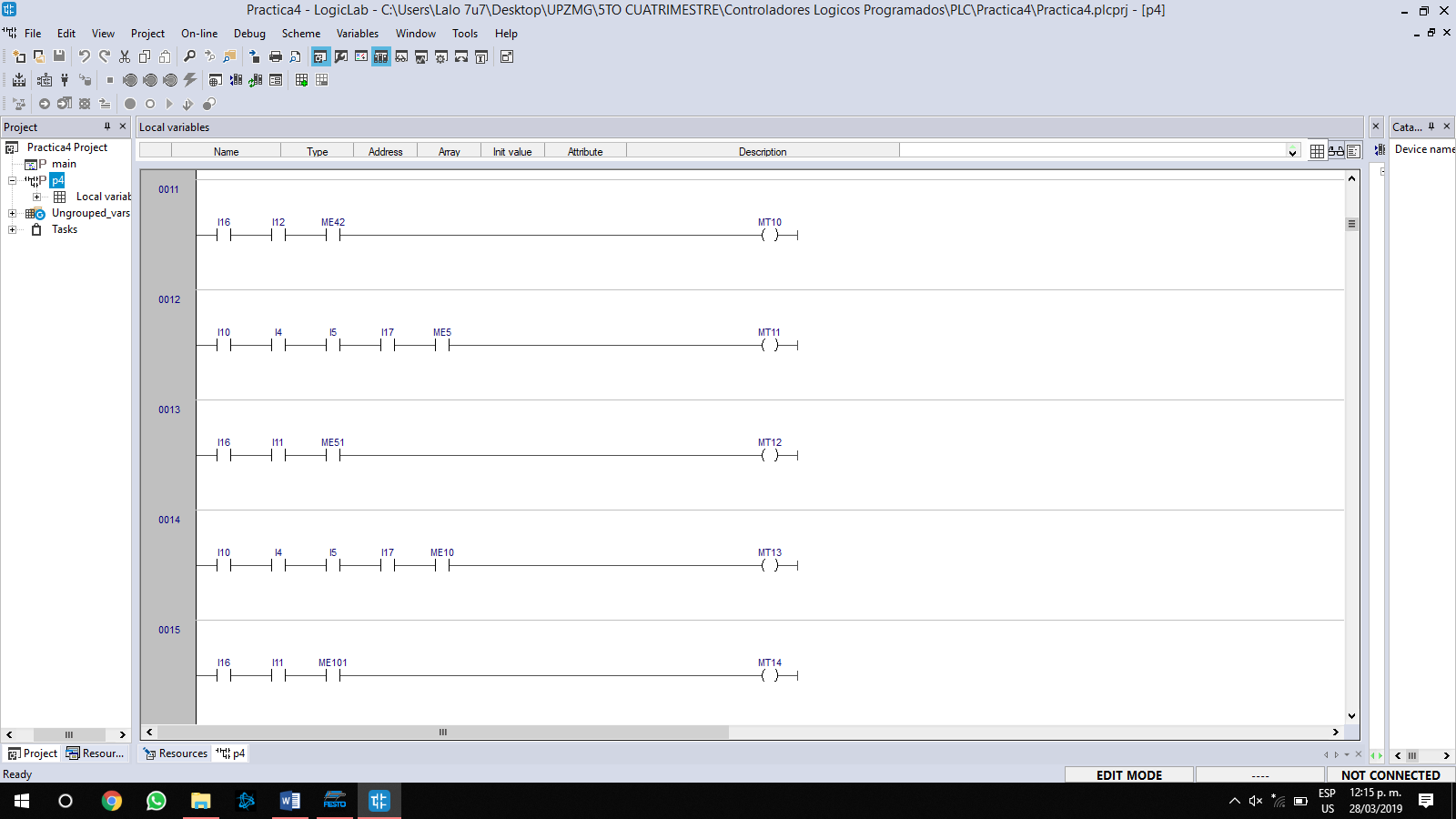




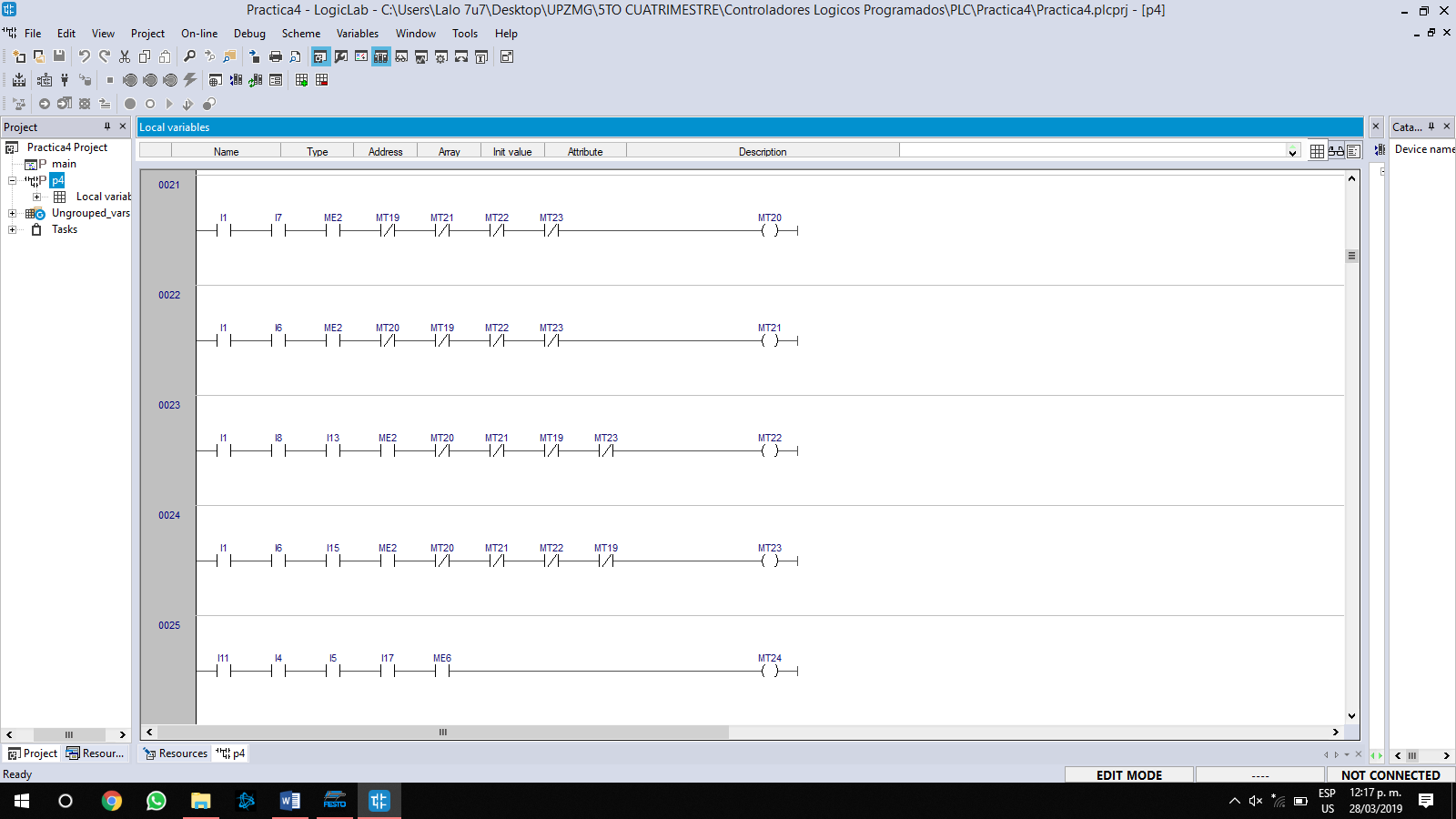
  
  
La primera parte de nuestro Ladder consta de las memorias de transición, las cuales son un conjunto de condiciones que necesitan estar activas para que pueda dar paso a la siguiente etapa.

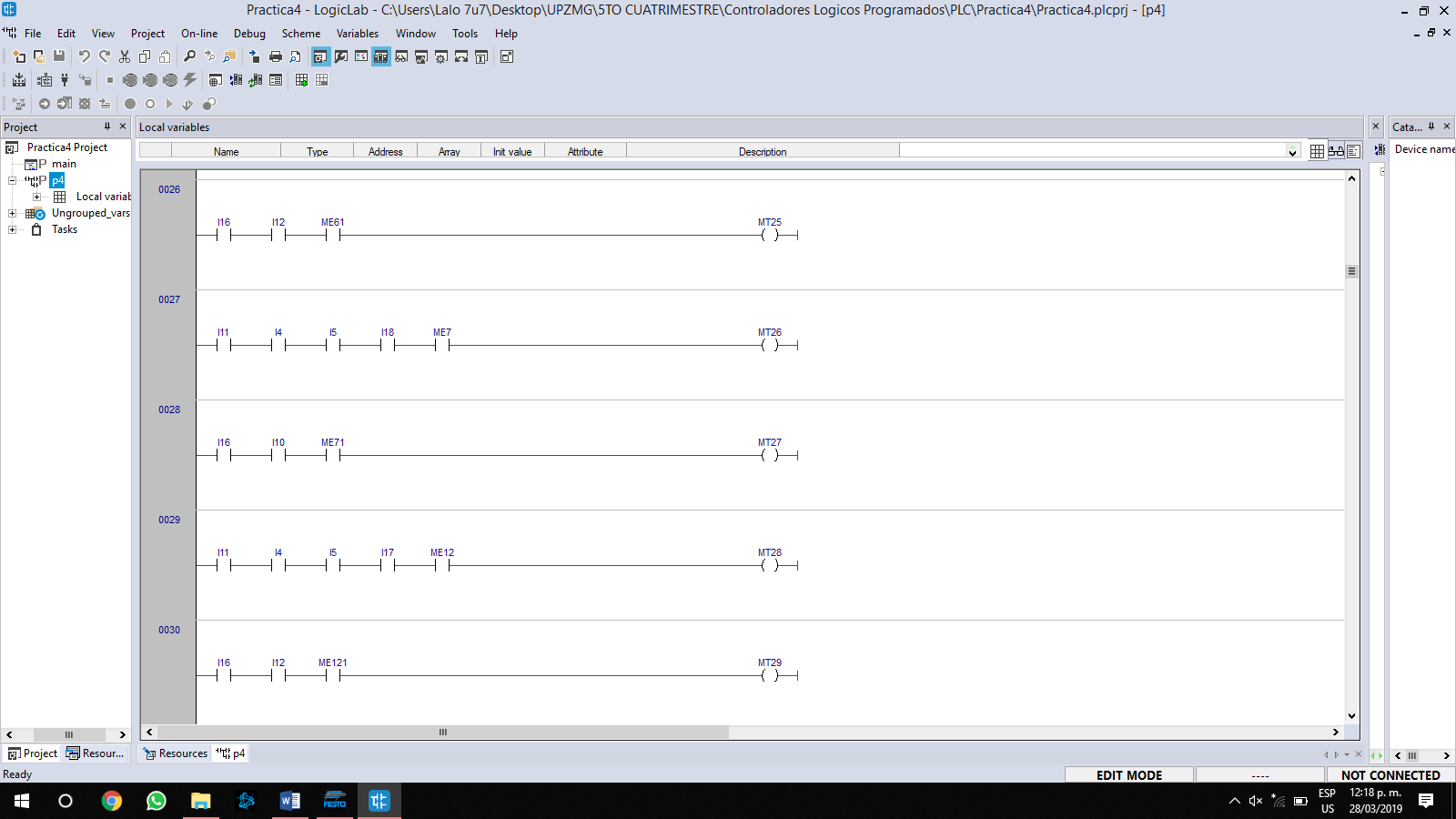


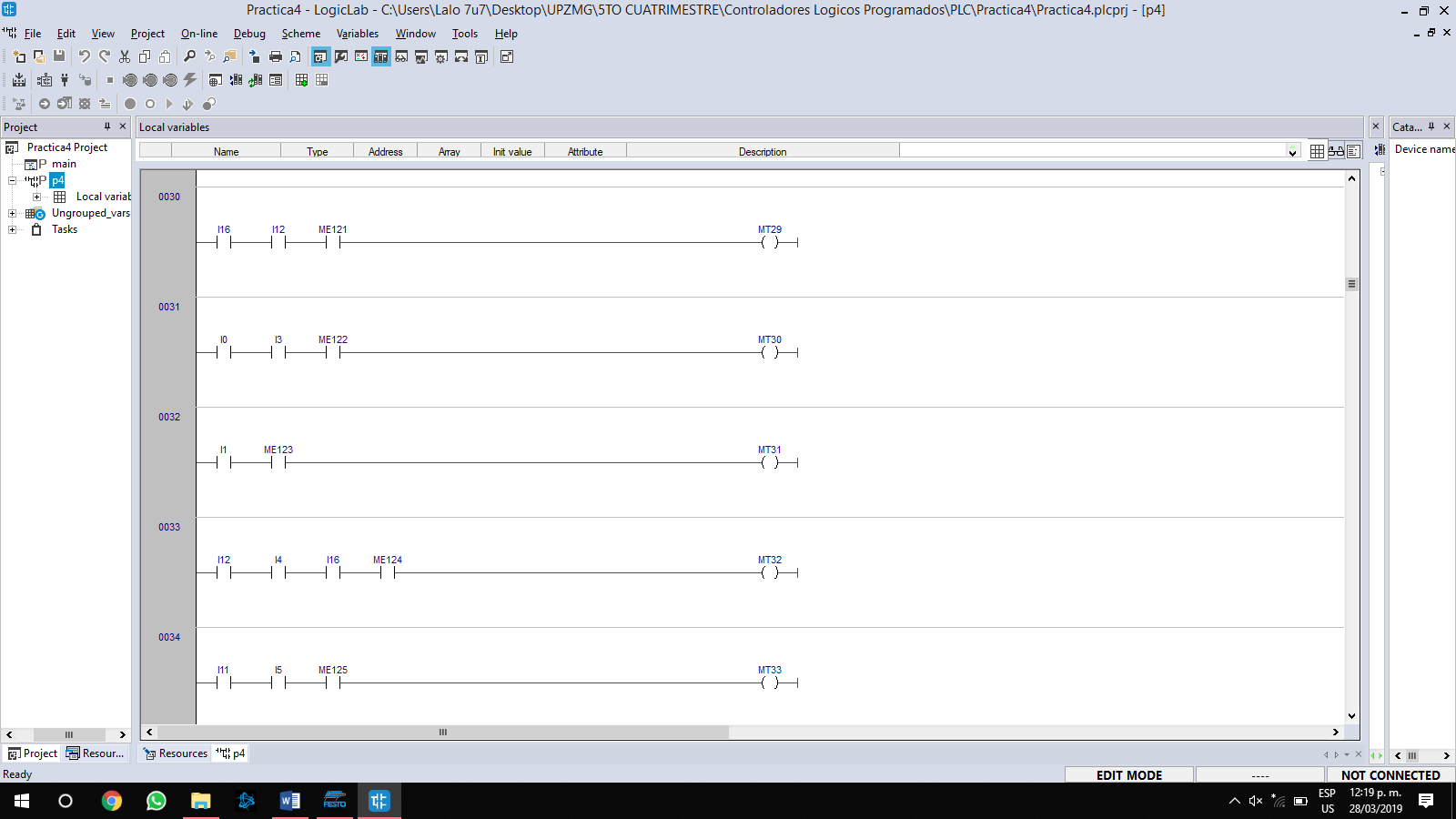


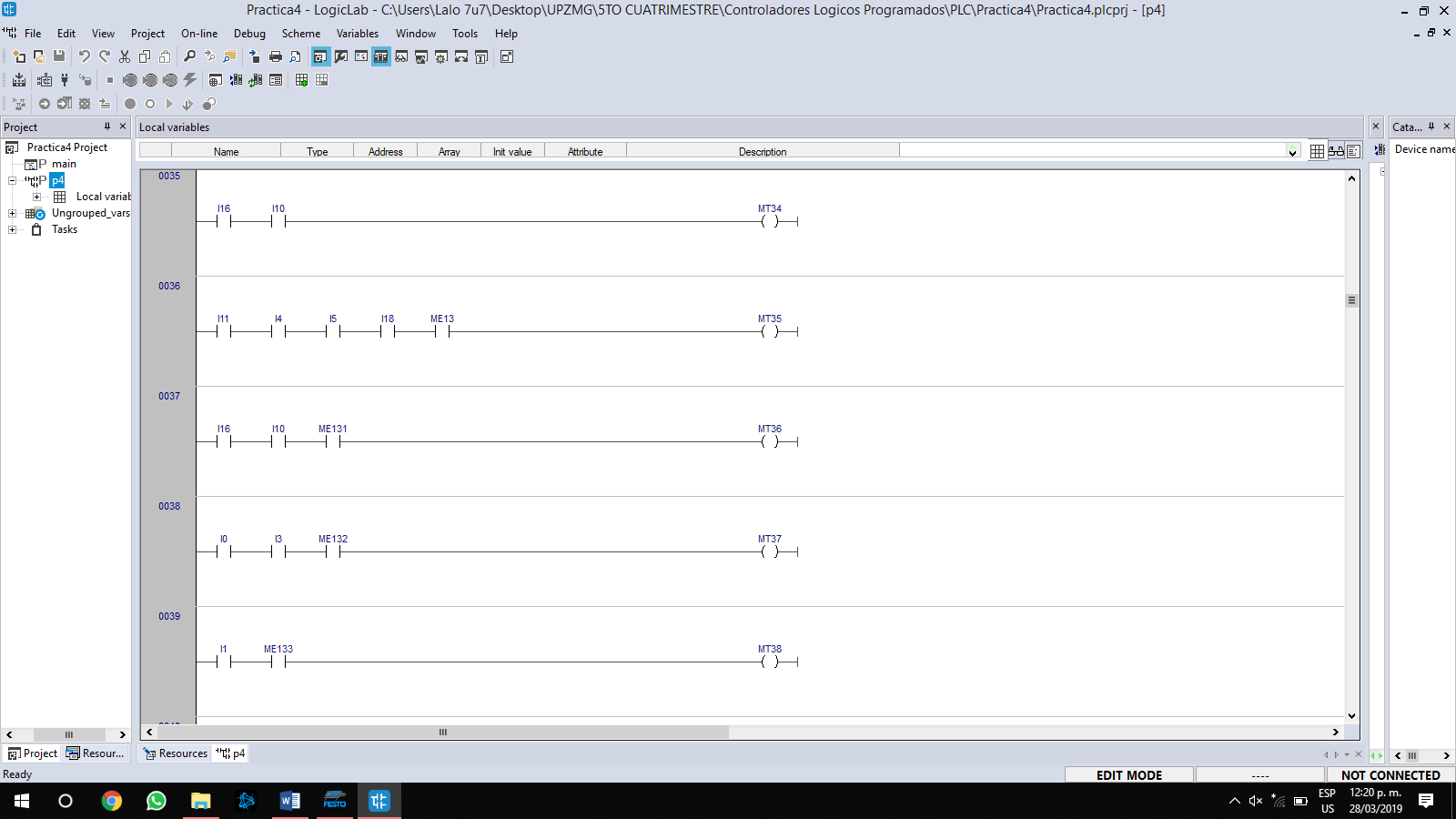


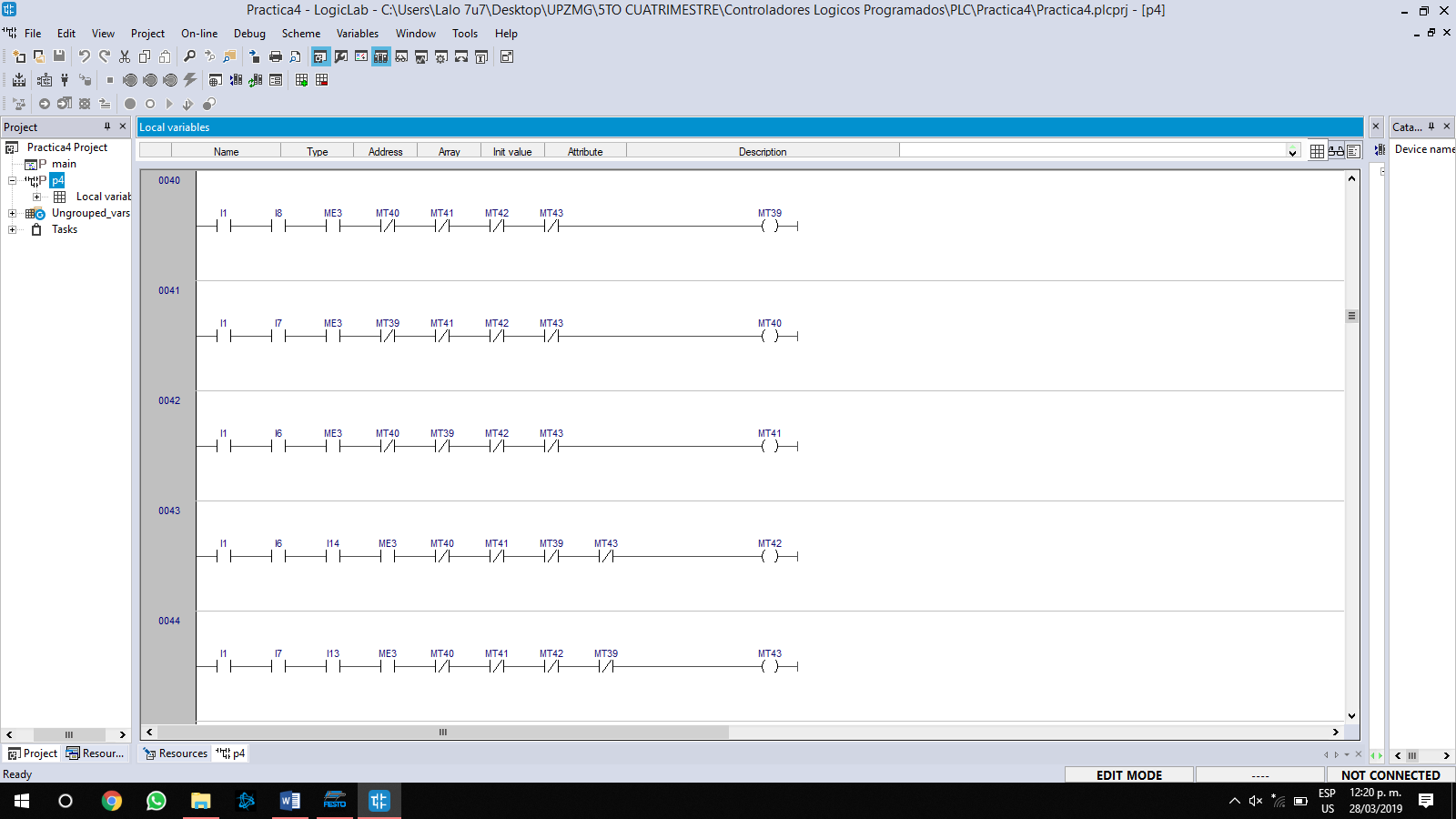


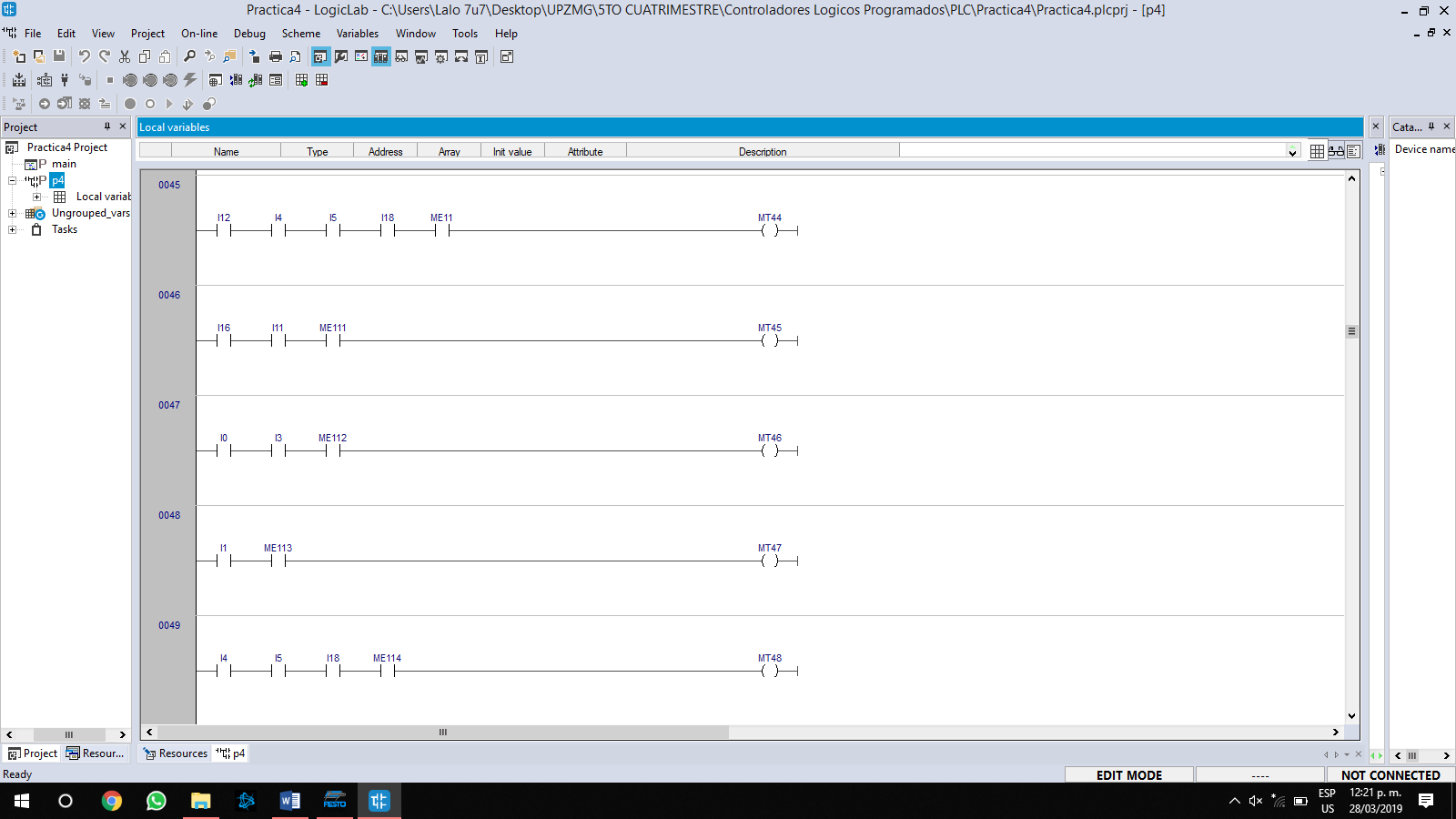


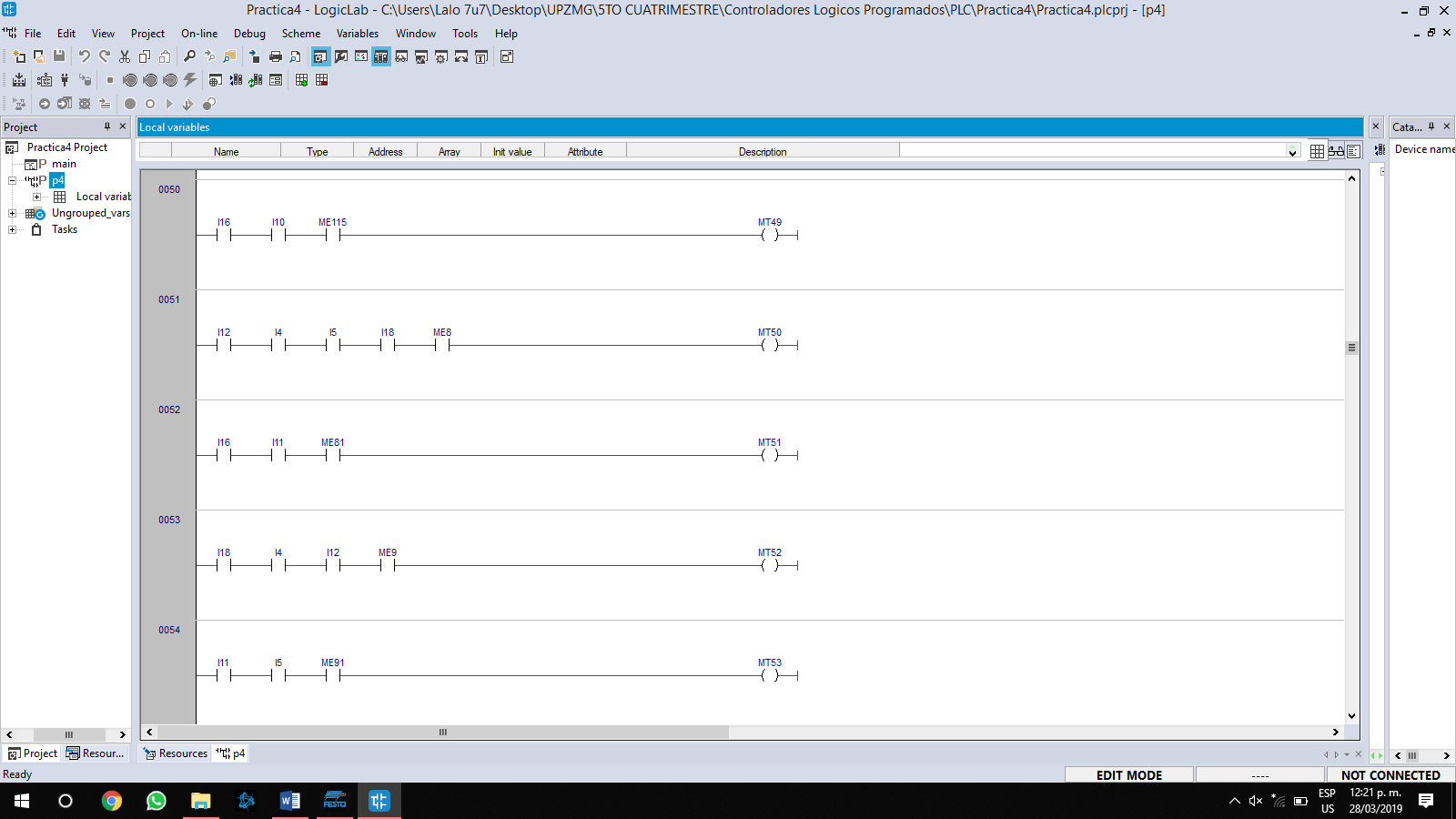


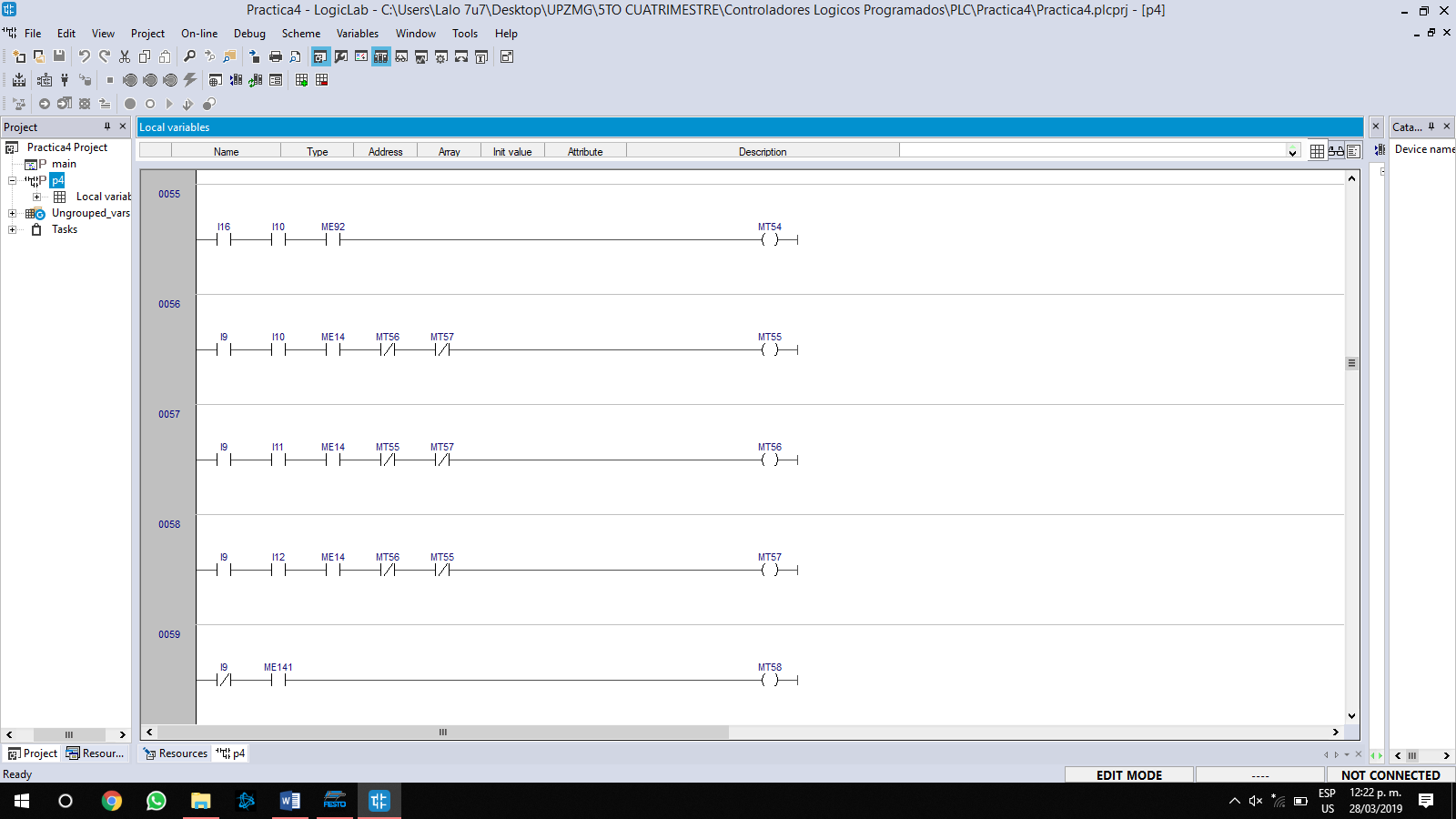


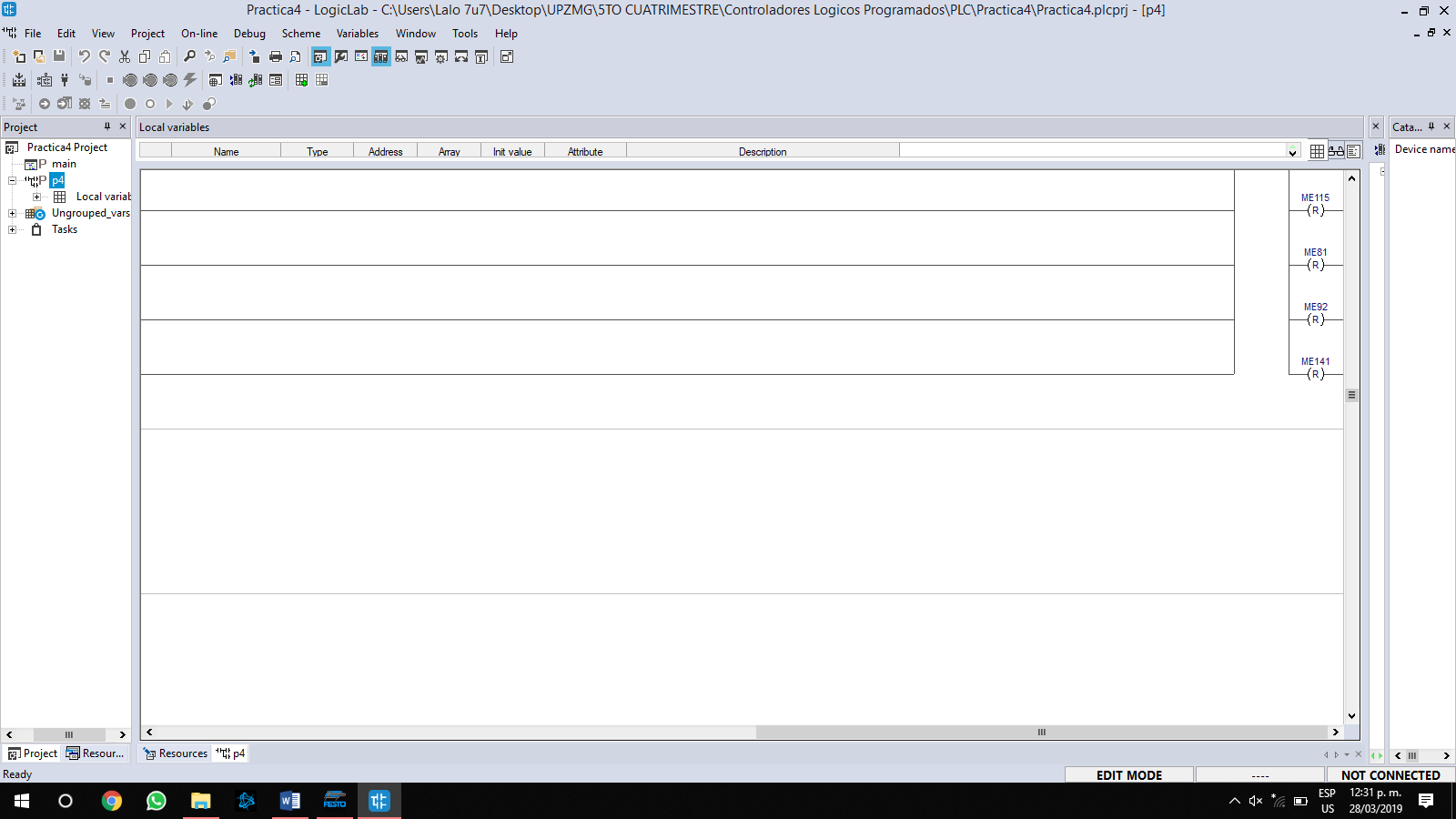
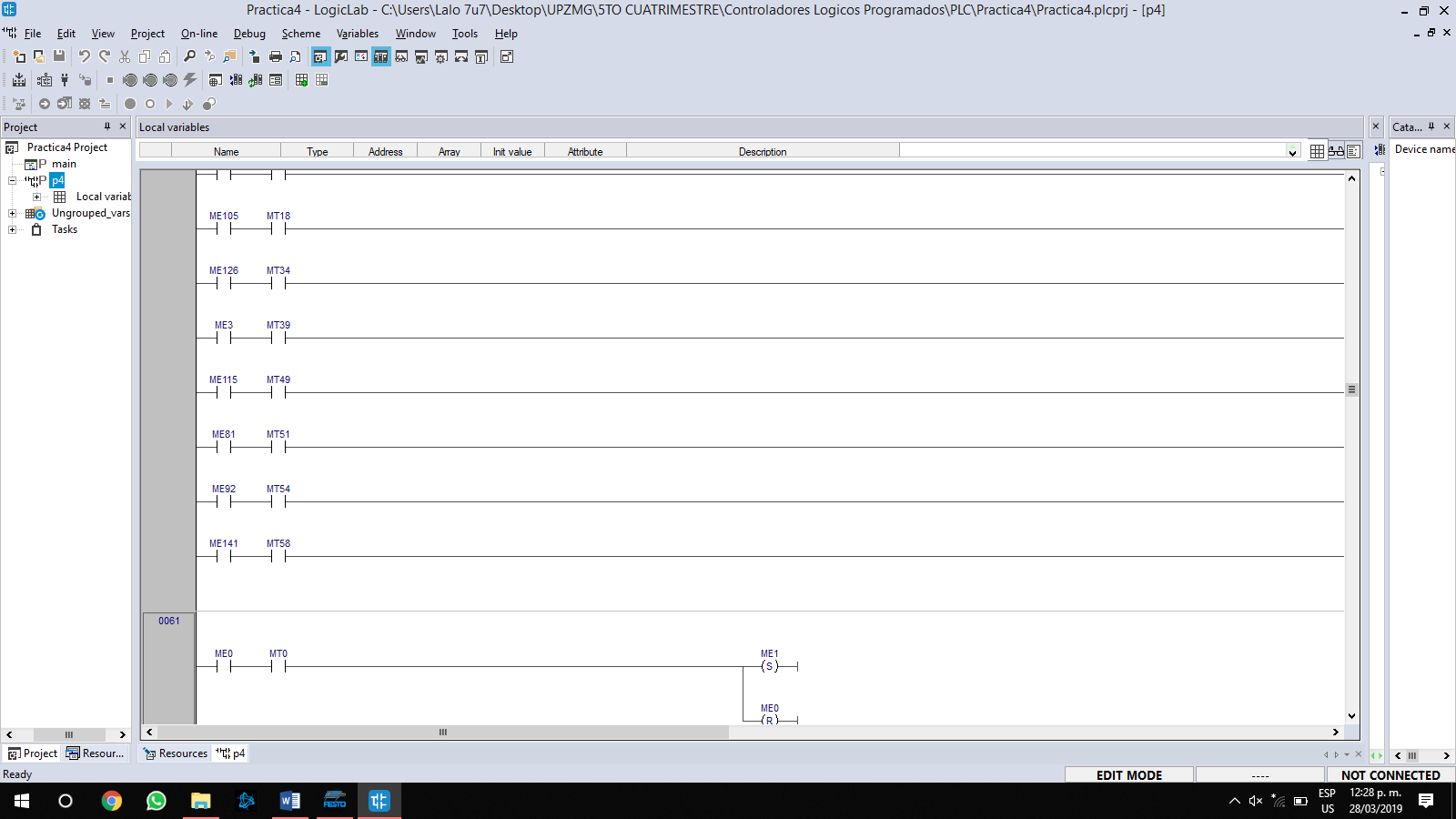
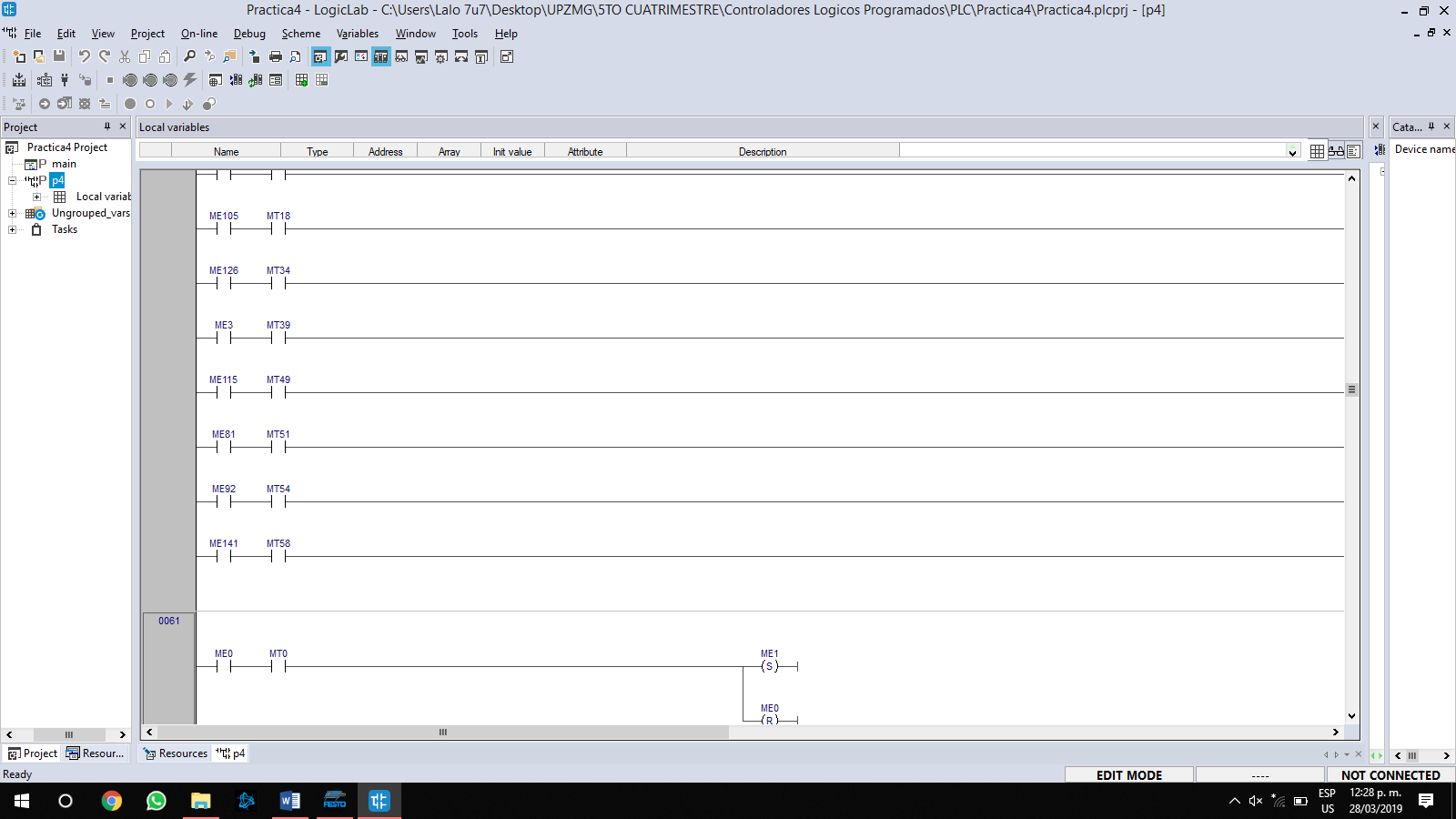
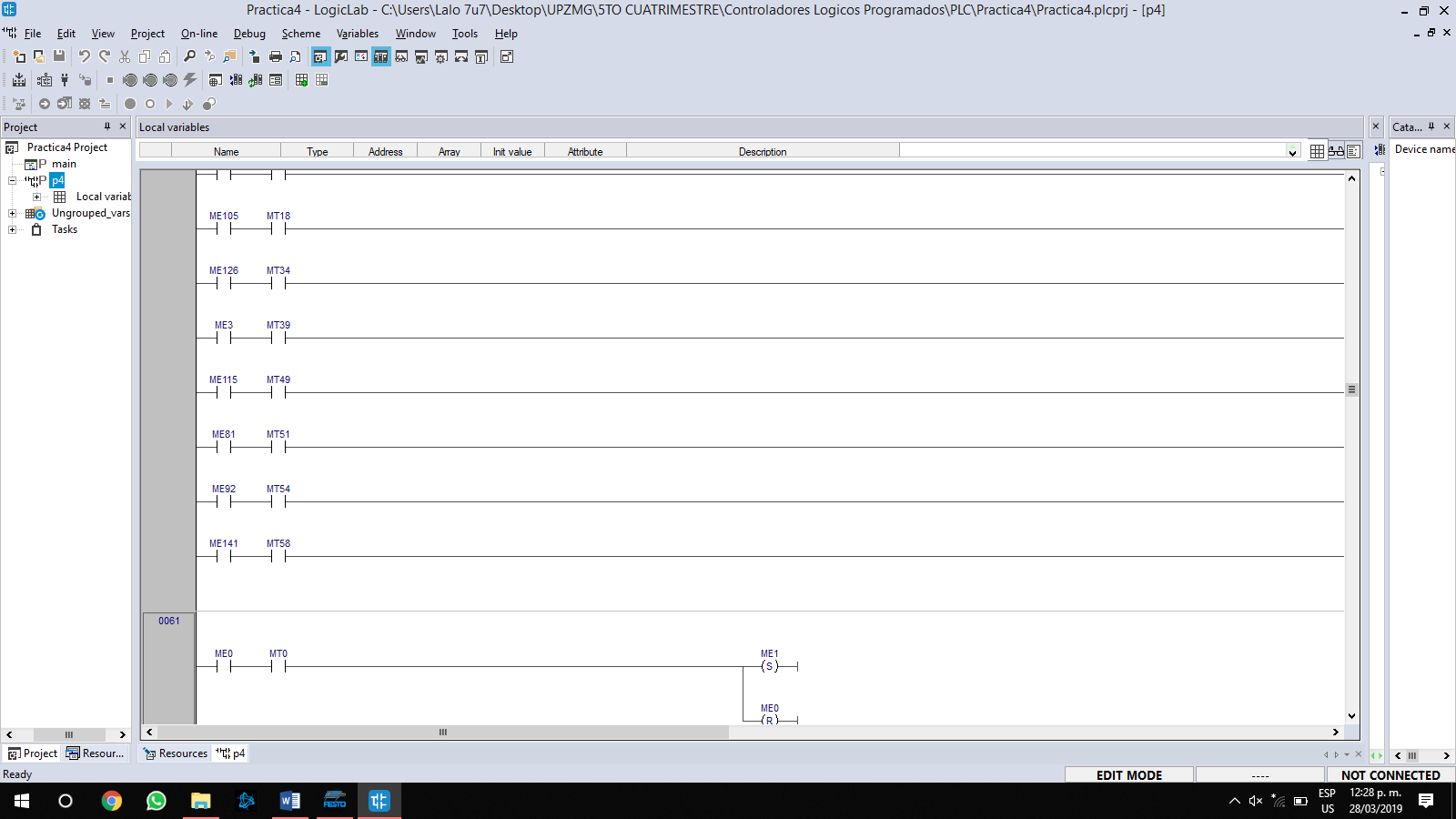
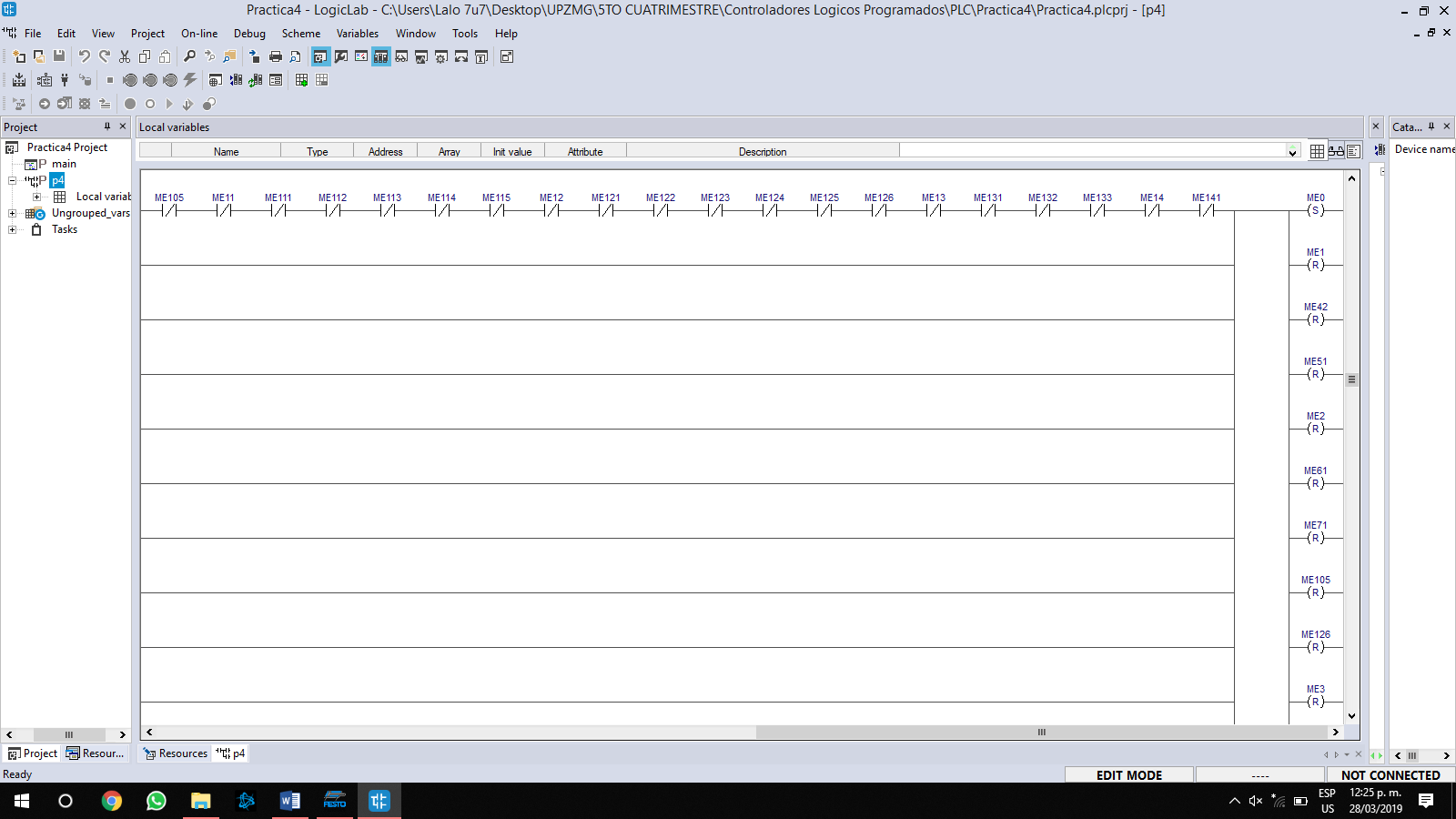
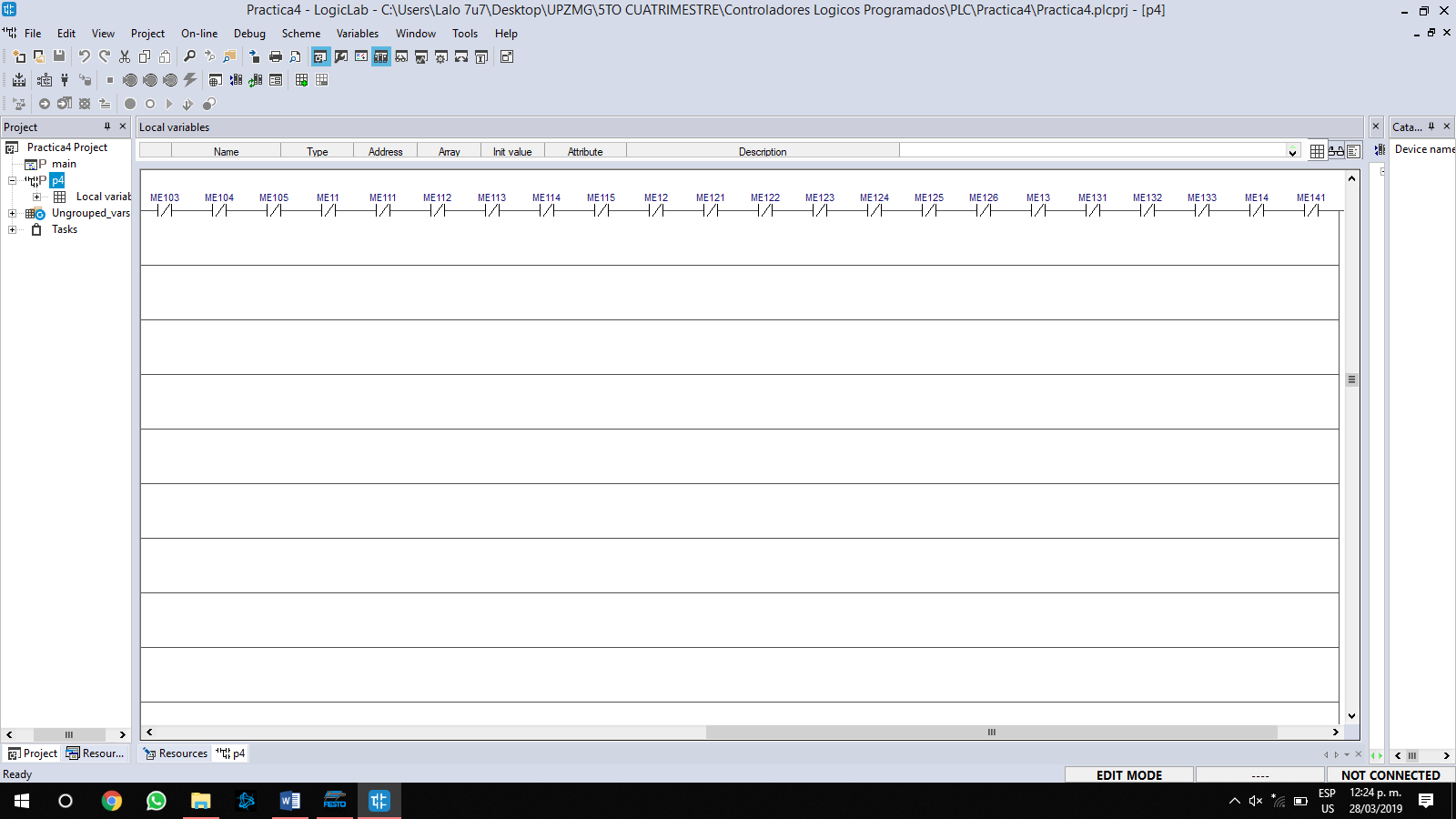
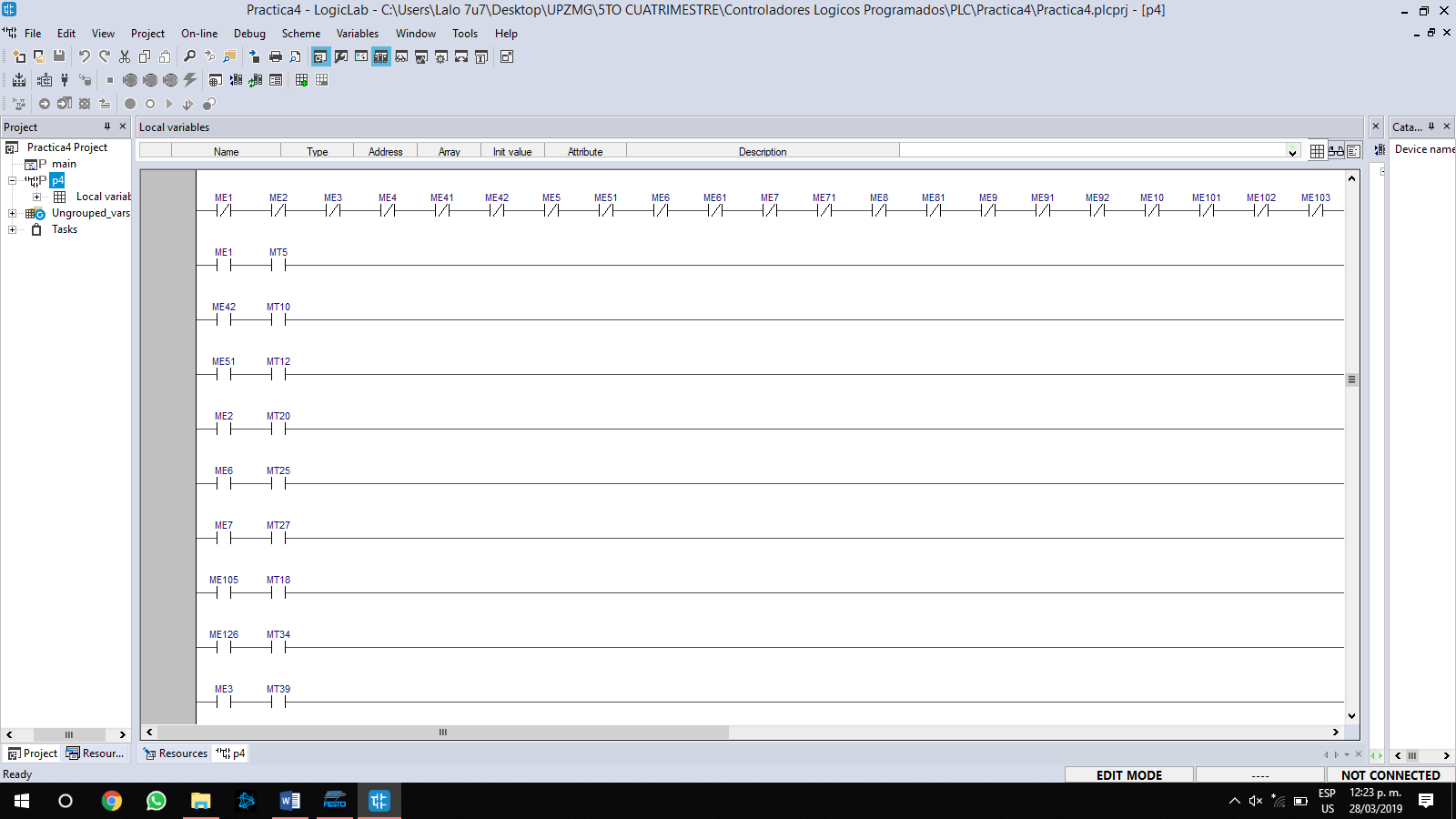
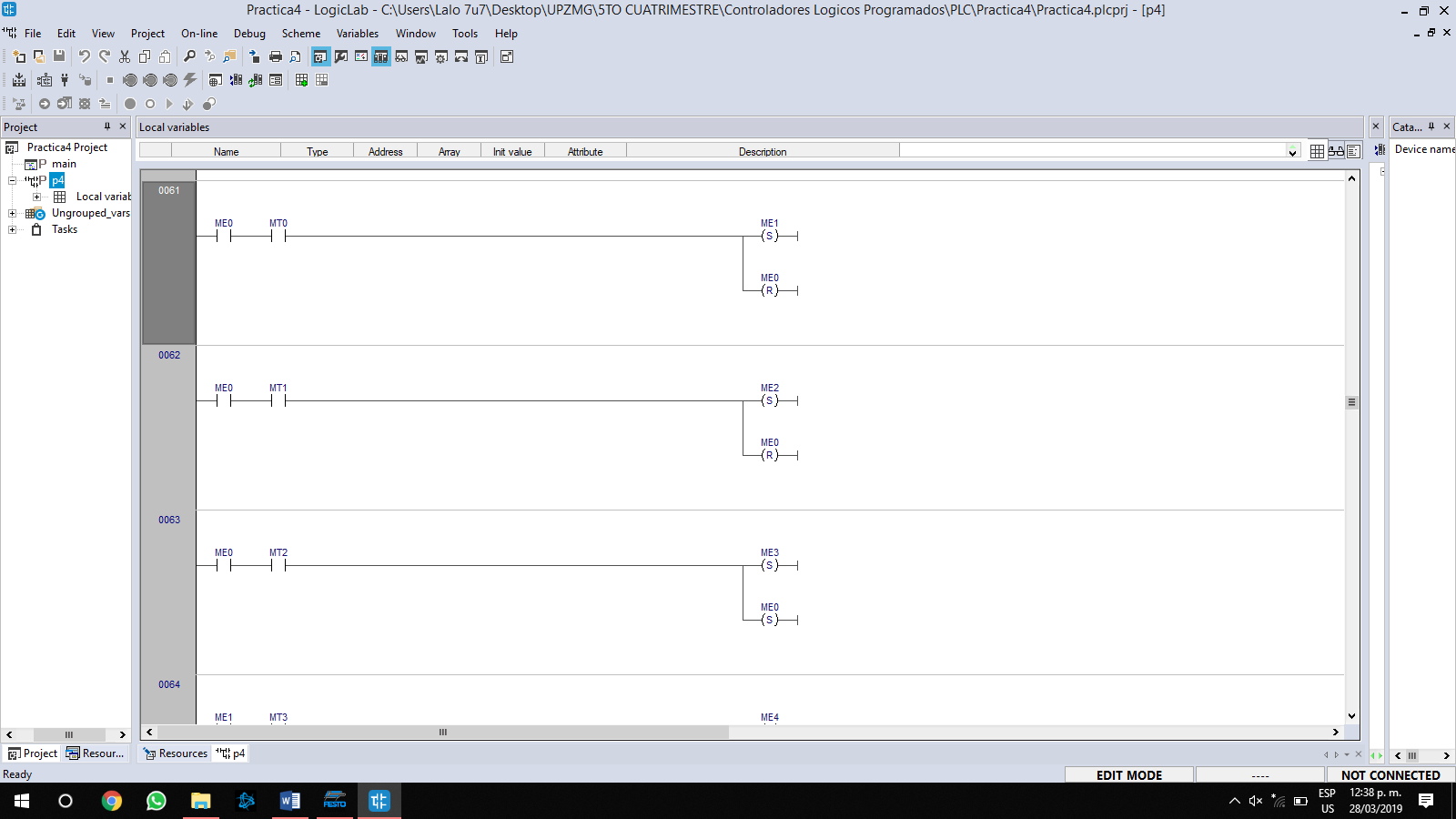


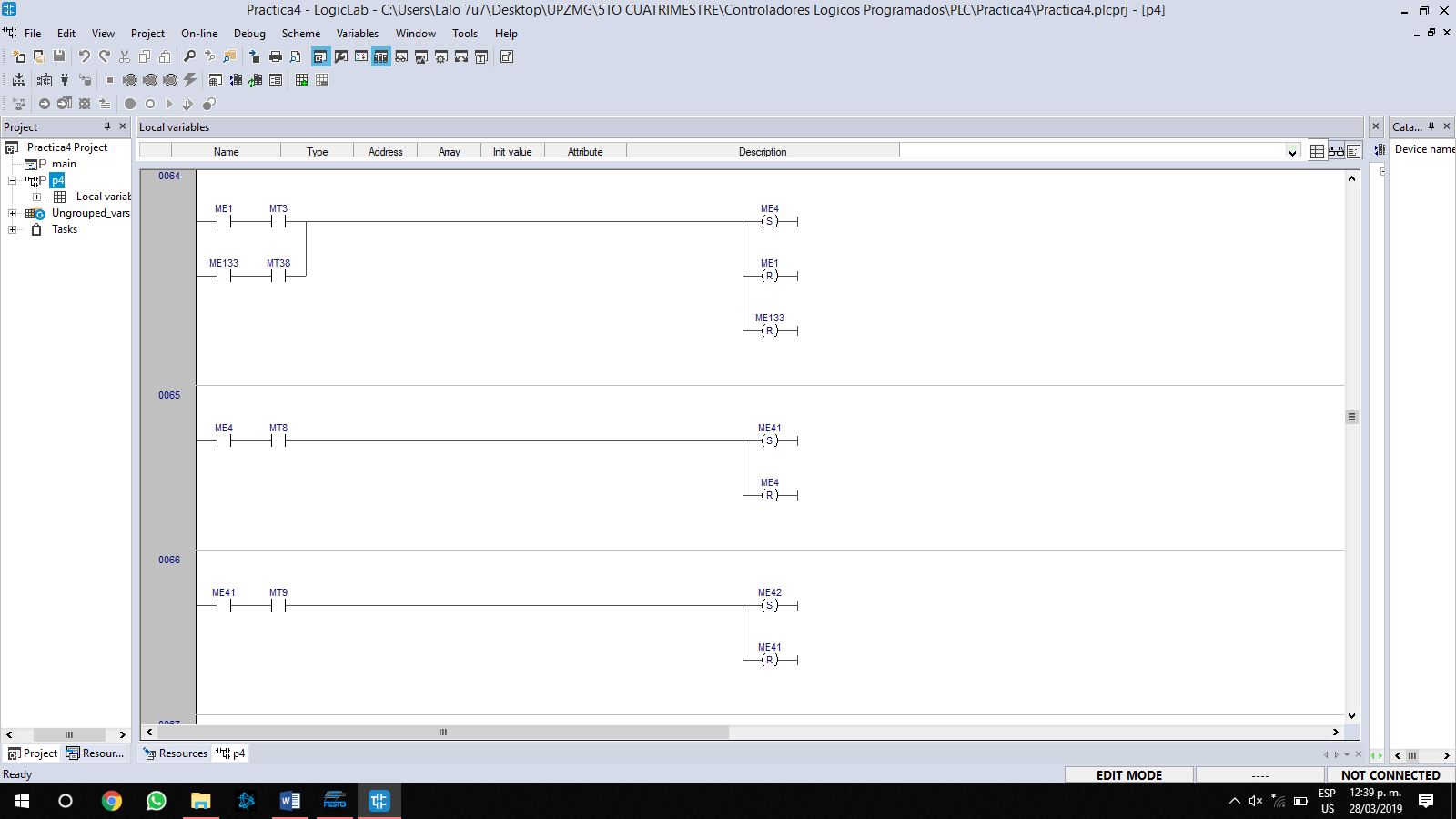


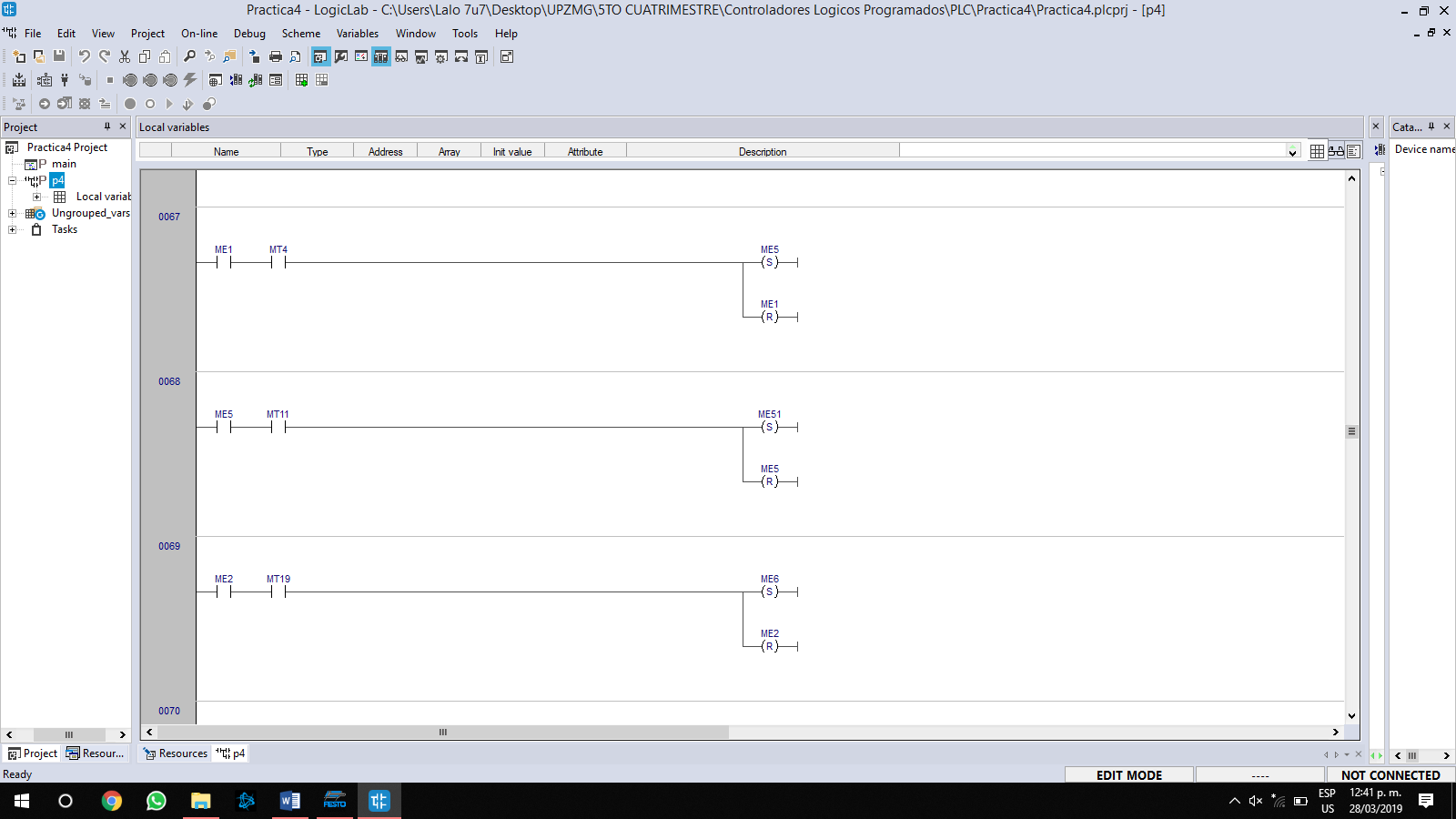


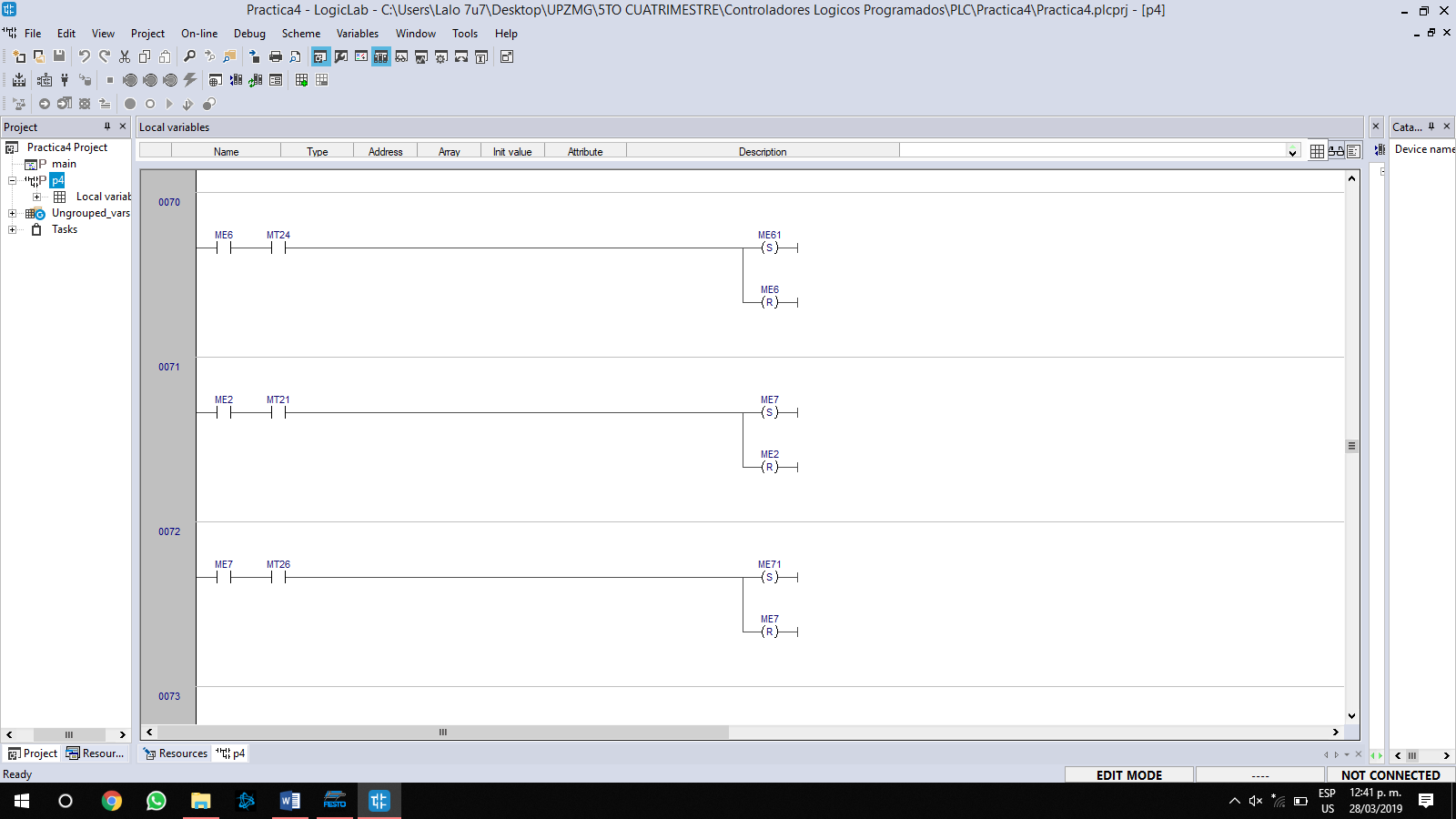


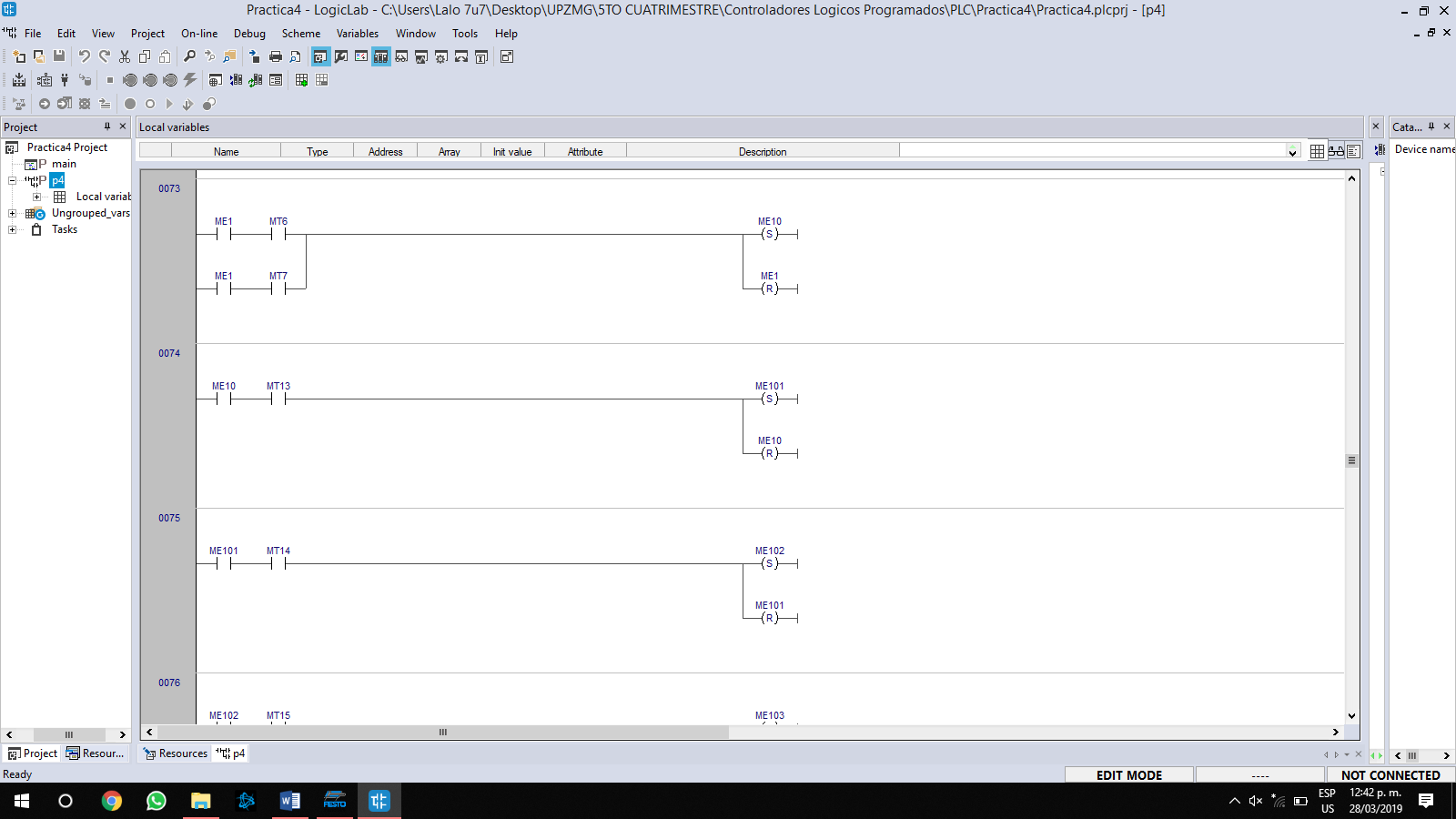


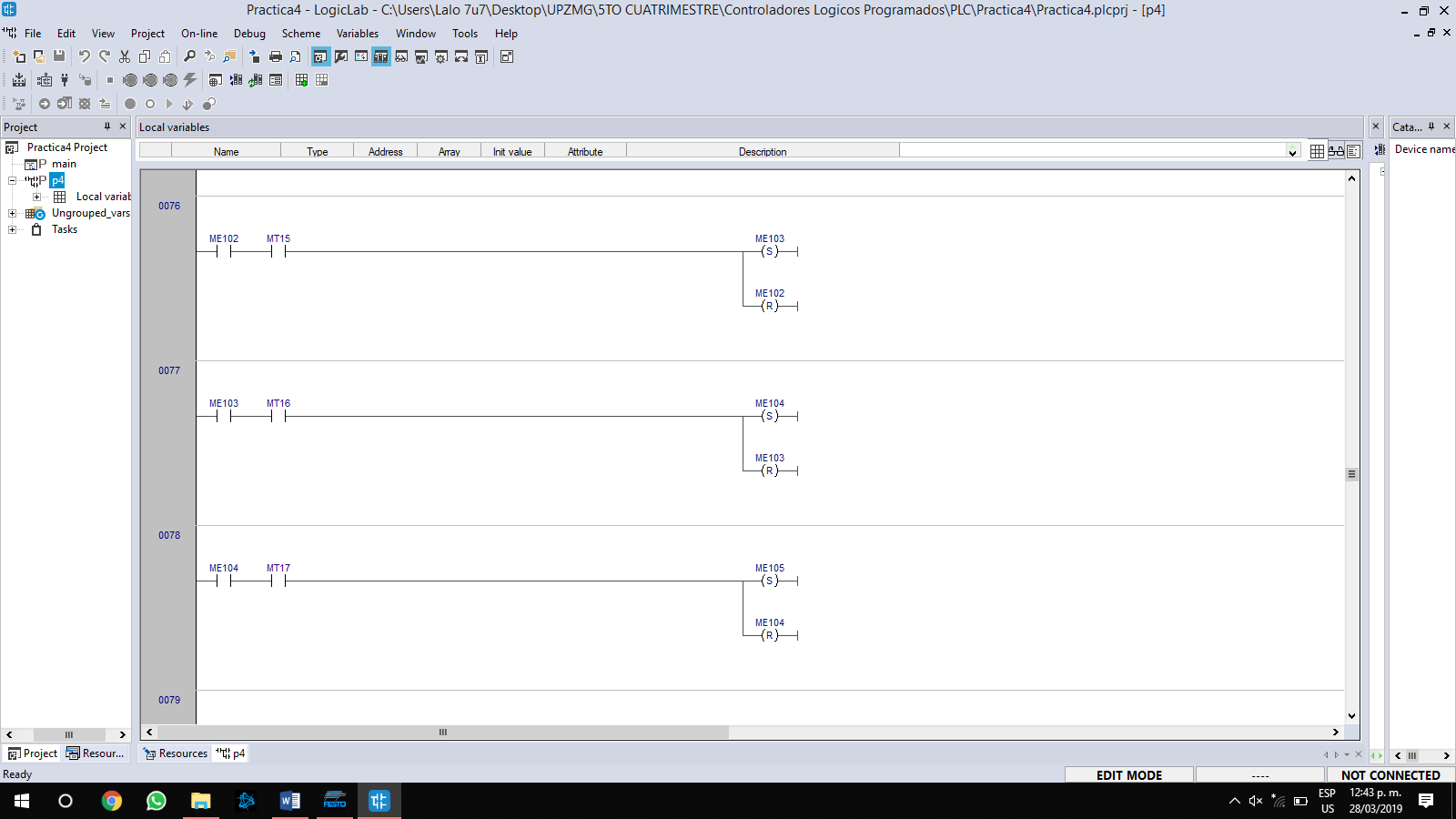
Aquí pasamos a la parte de las memorias de estado donde hay que activar o desactivar para pasar al siguiente estado  
  
  


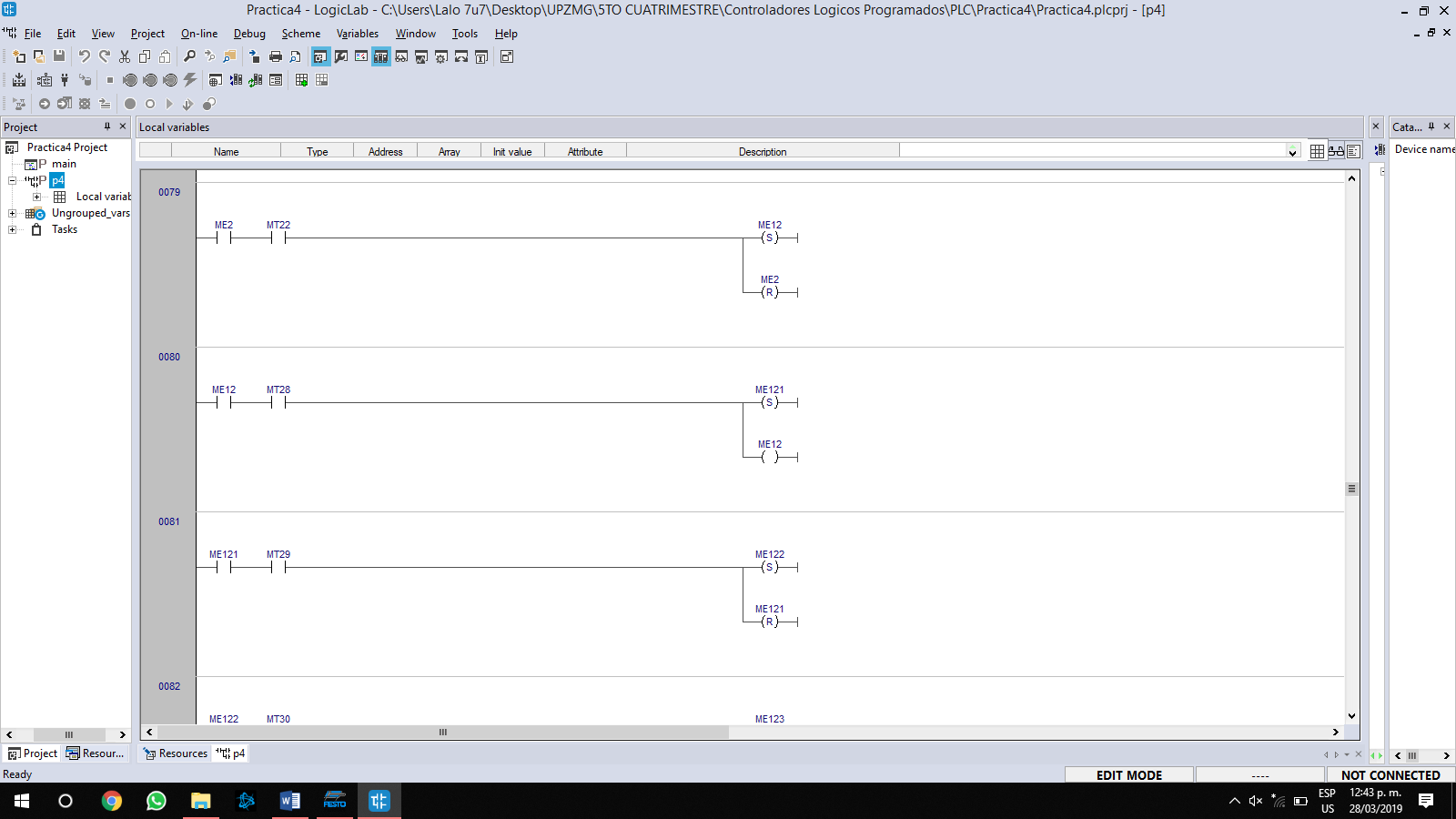


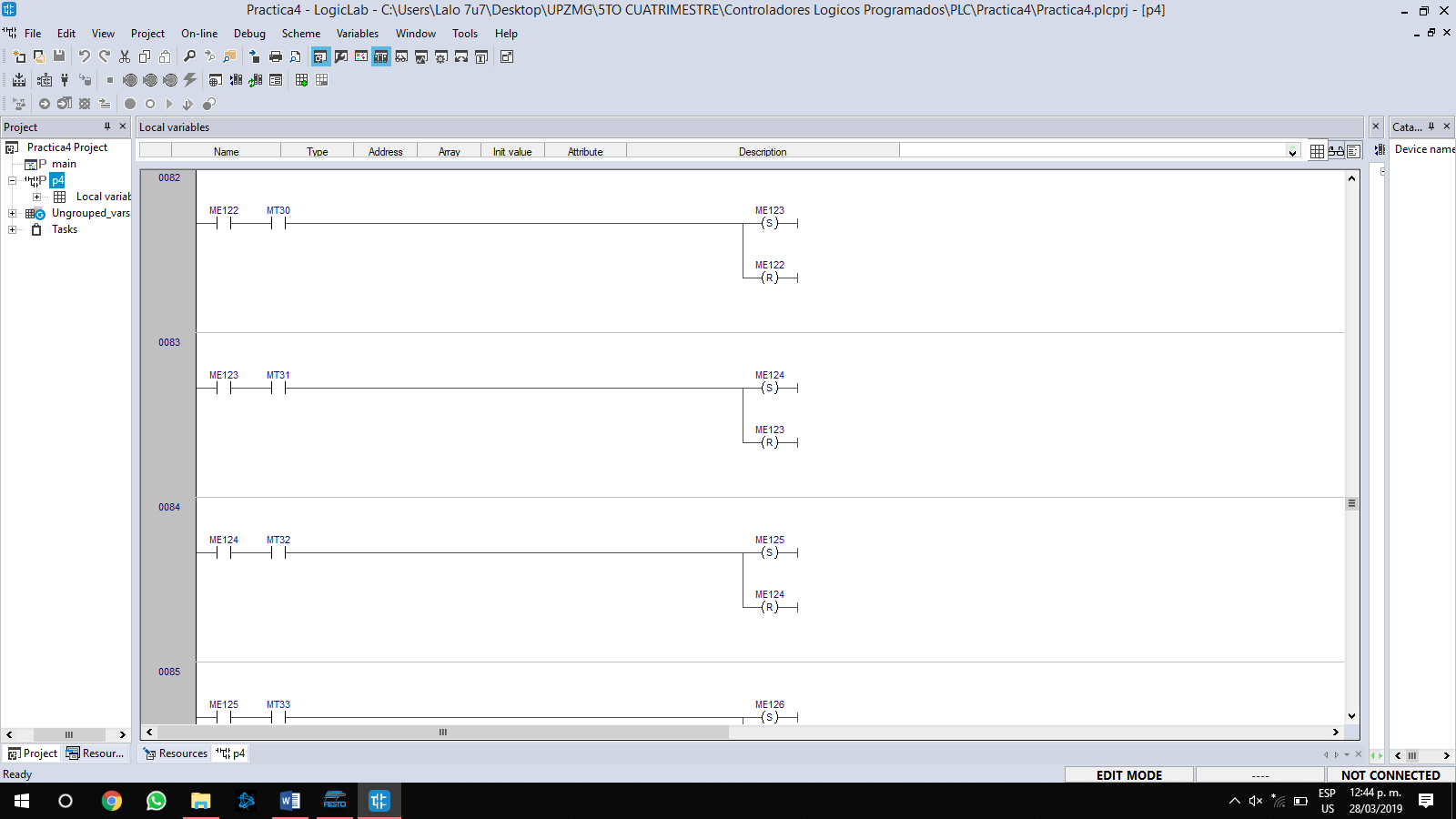


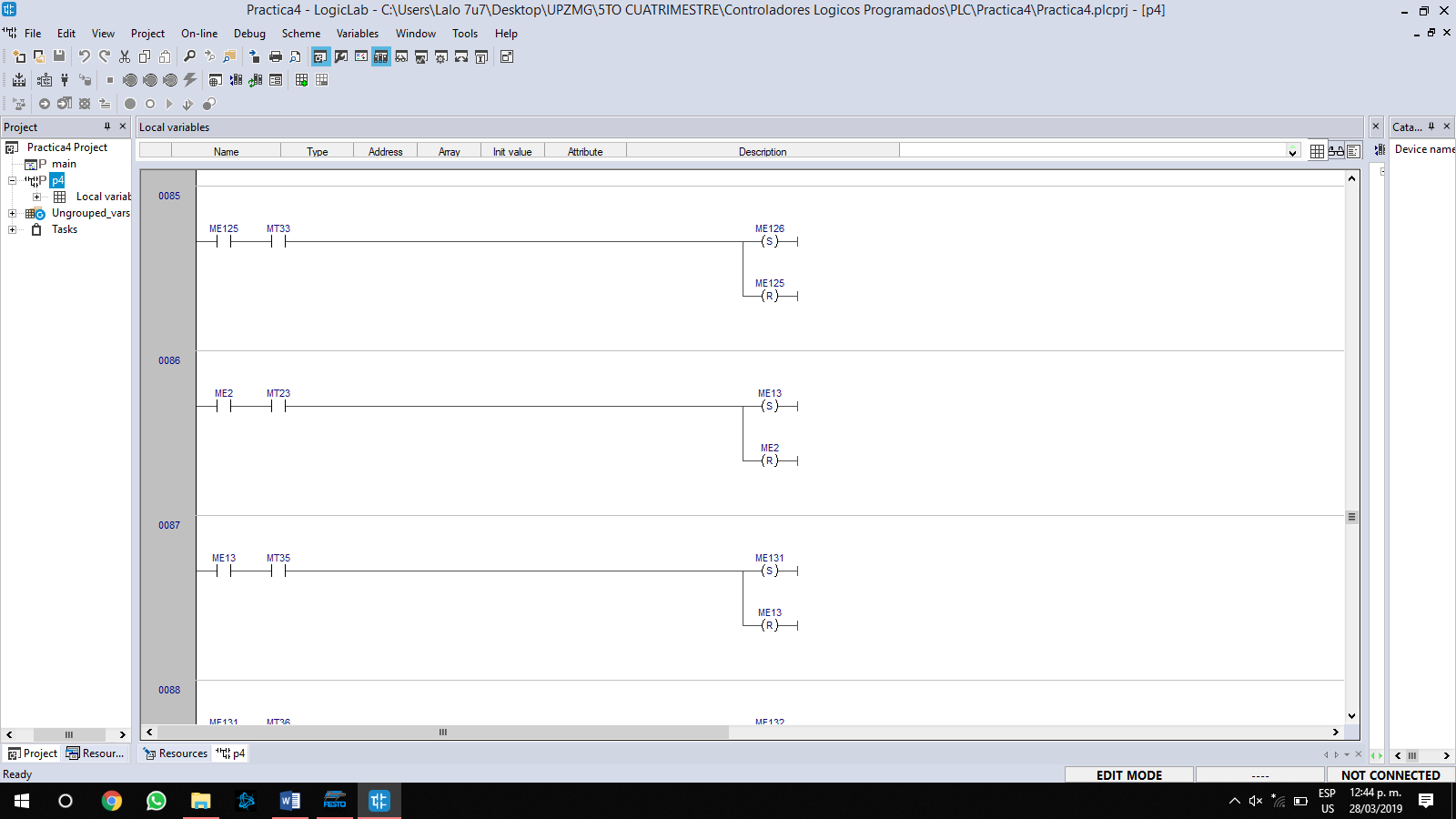


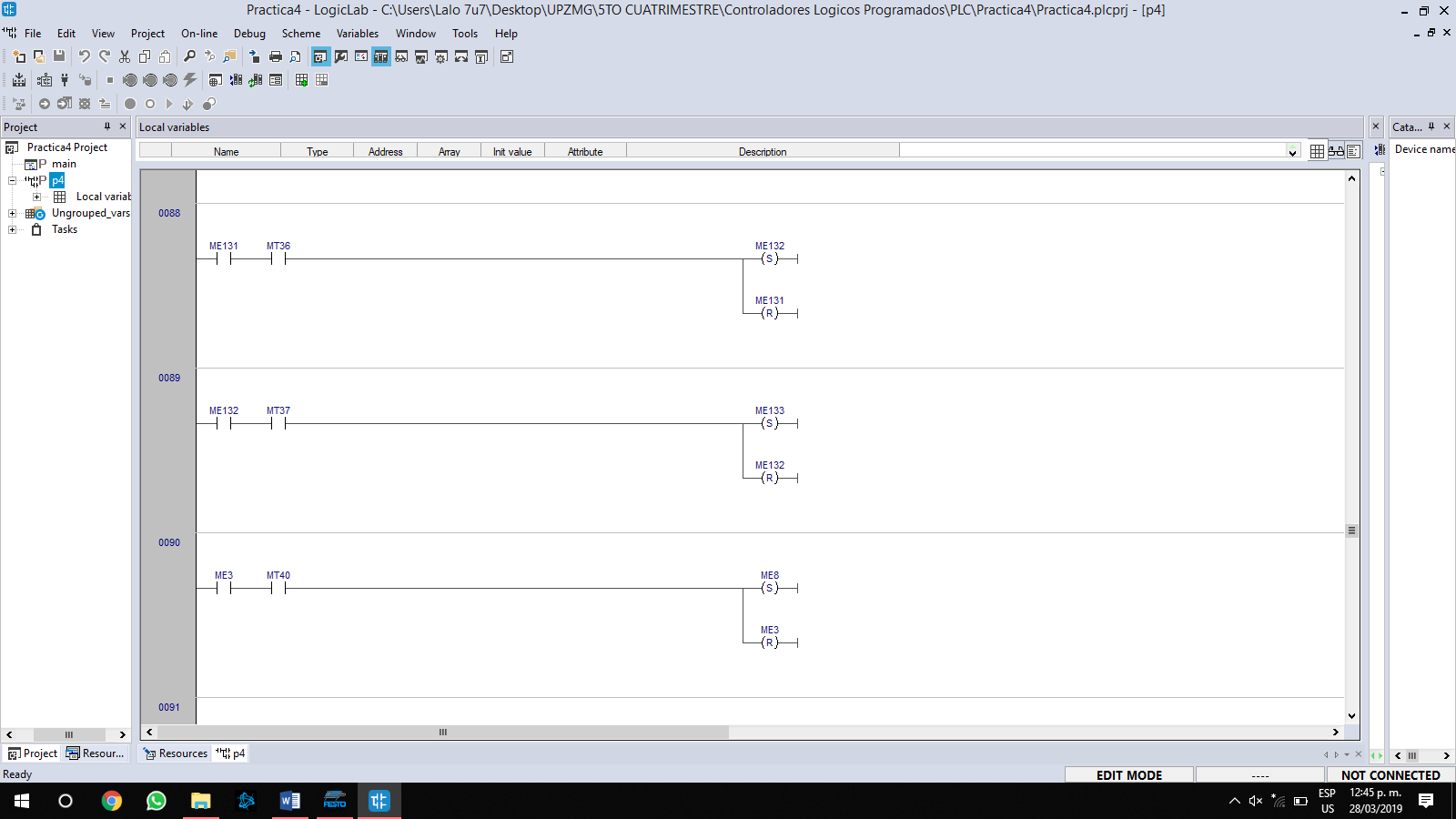


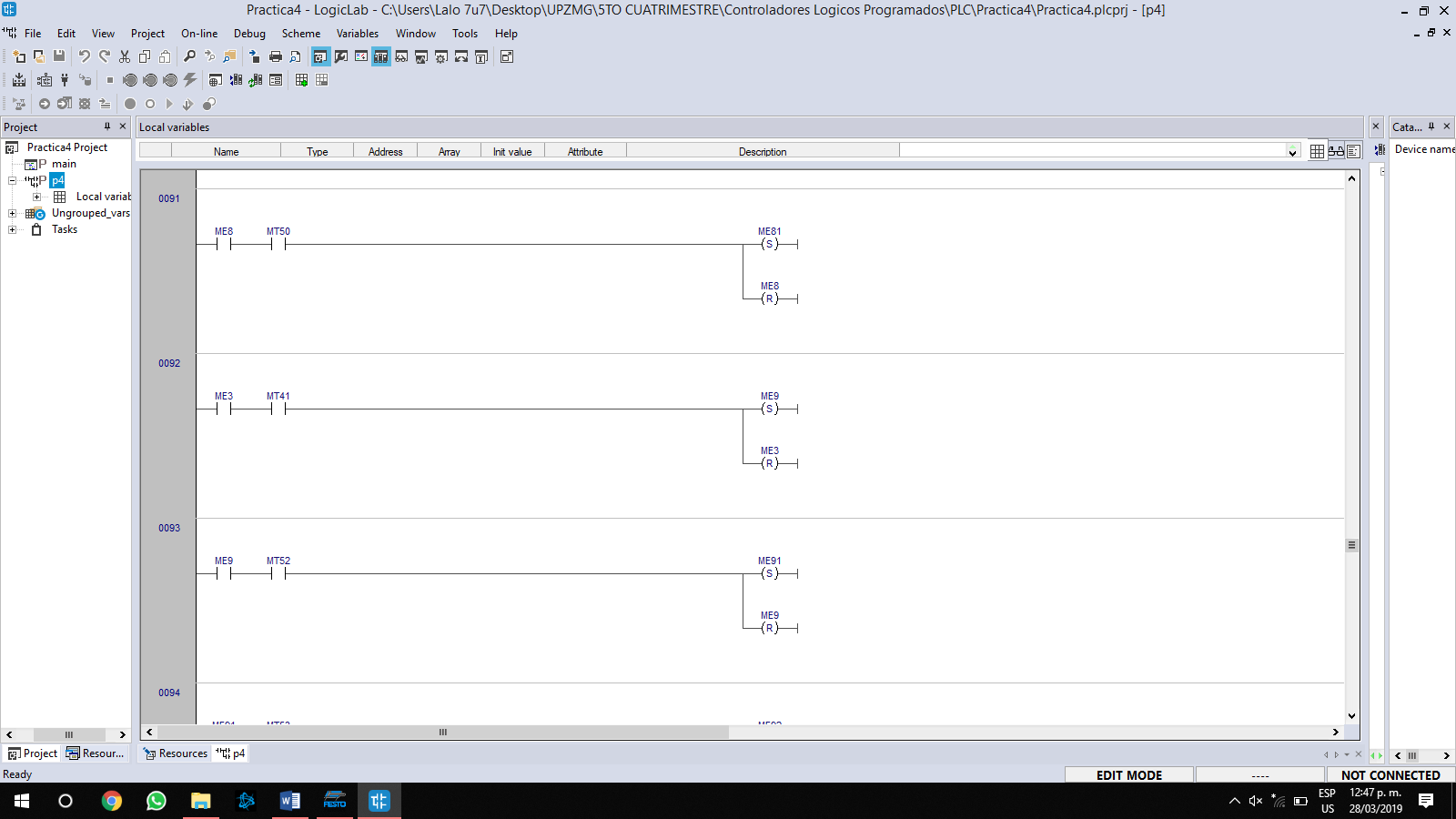


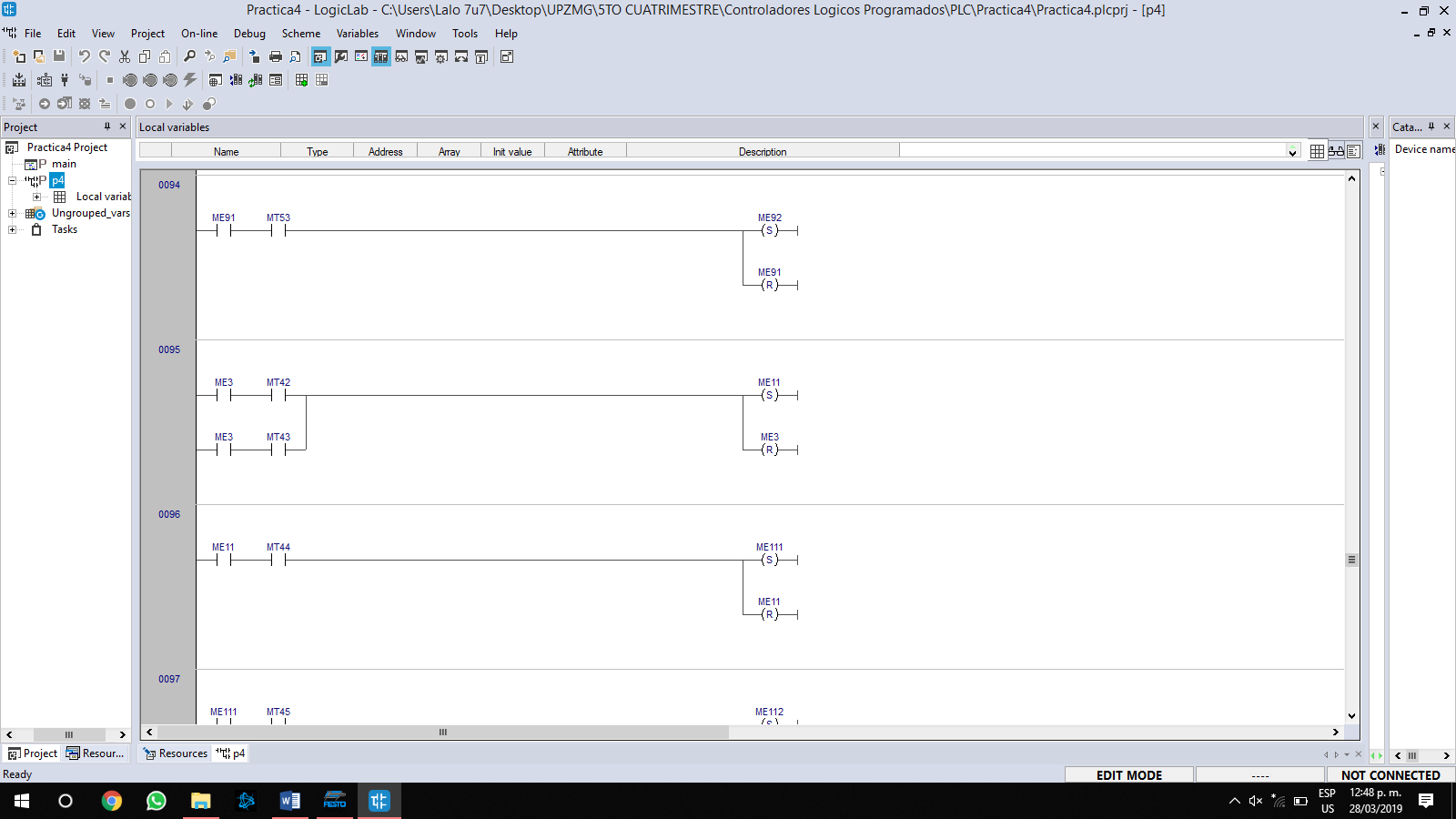


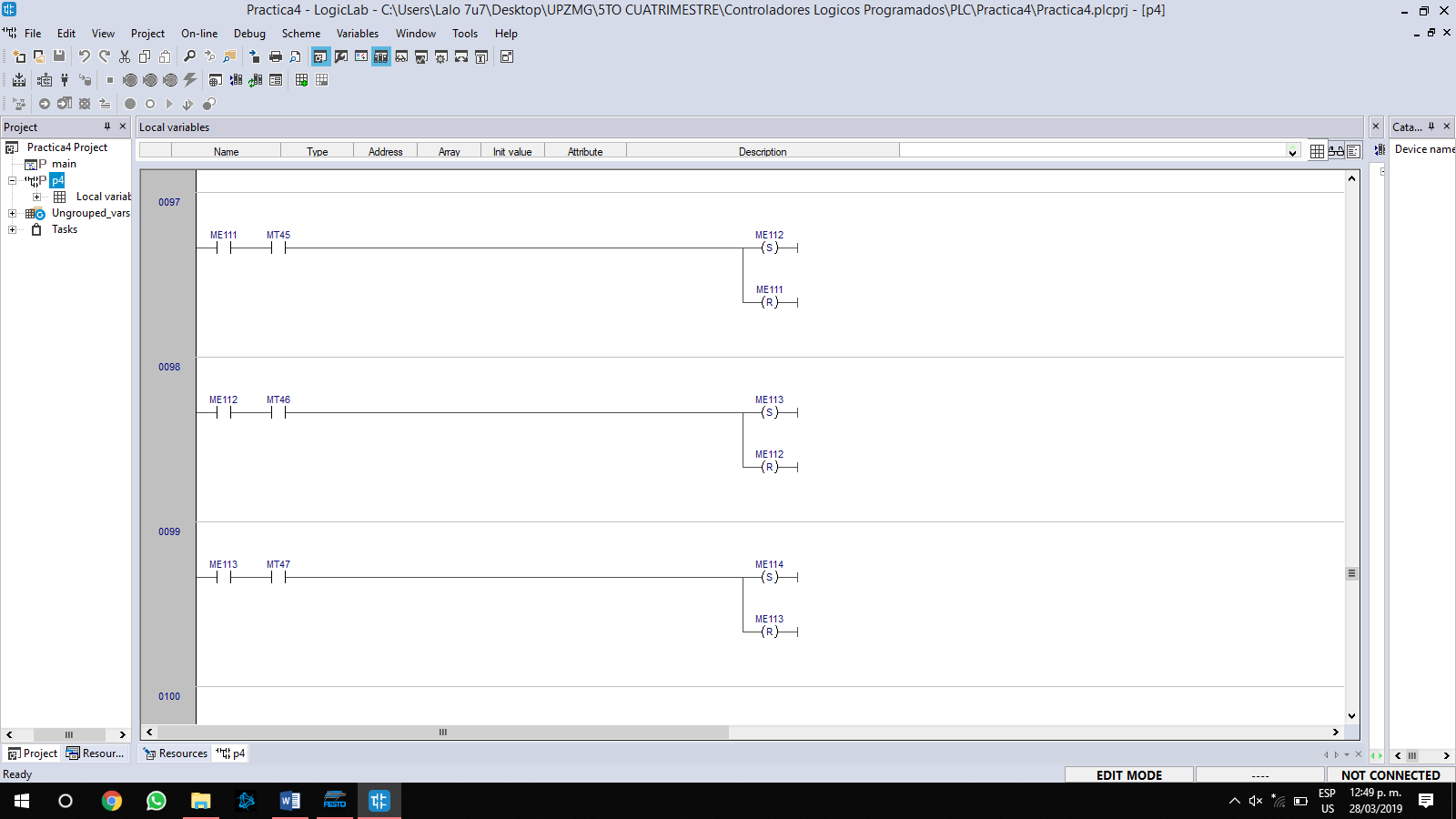


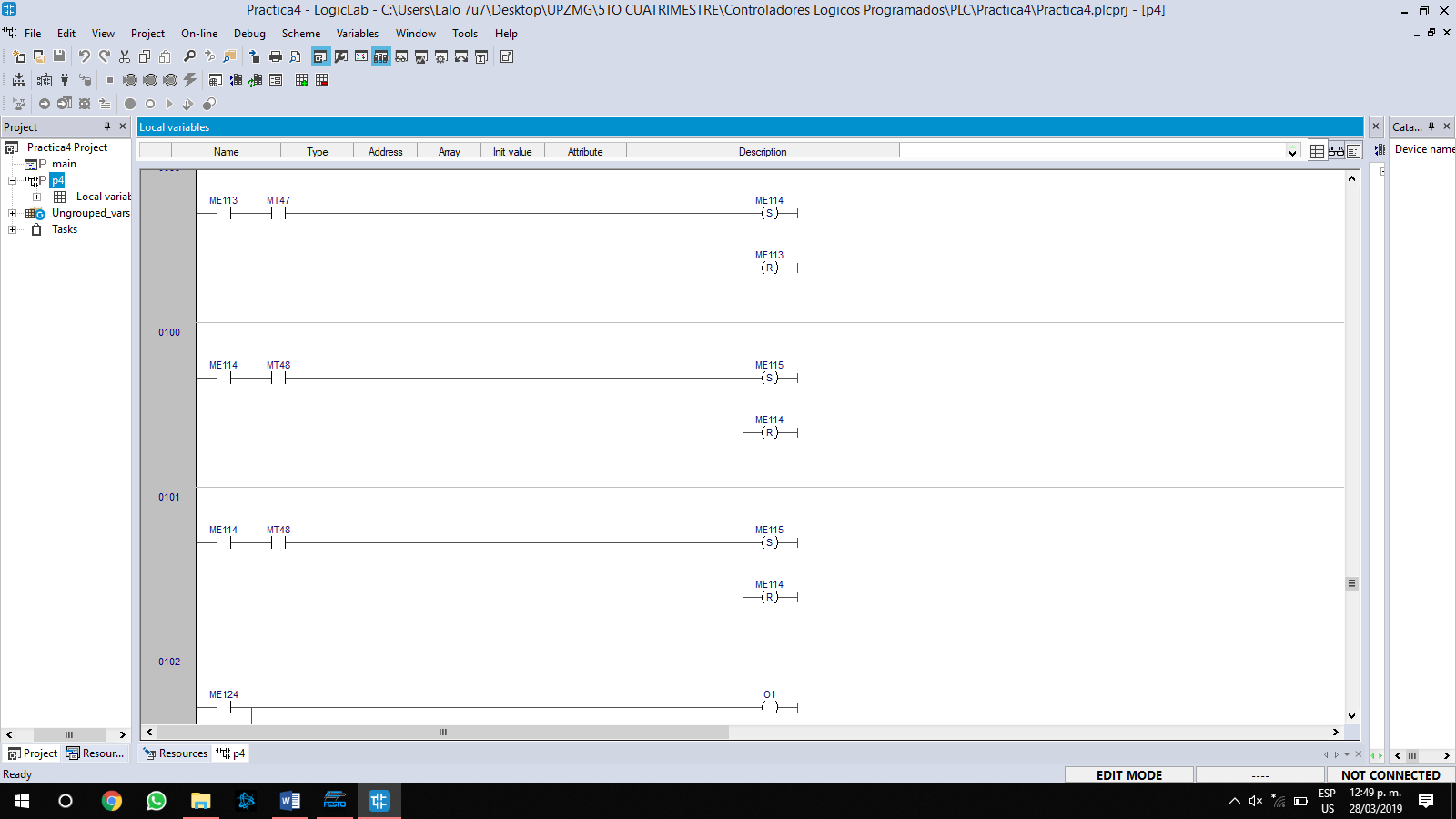


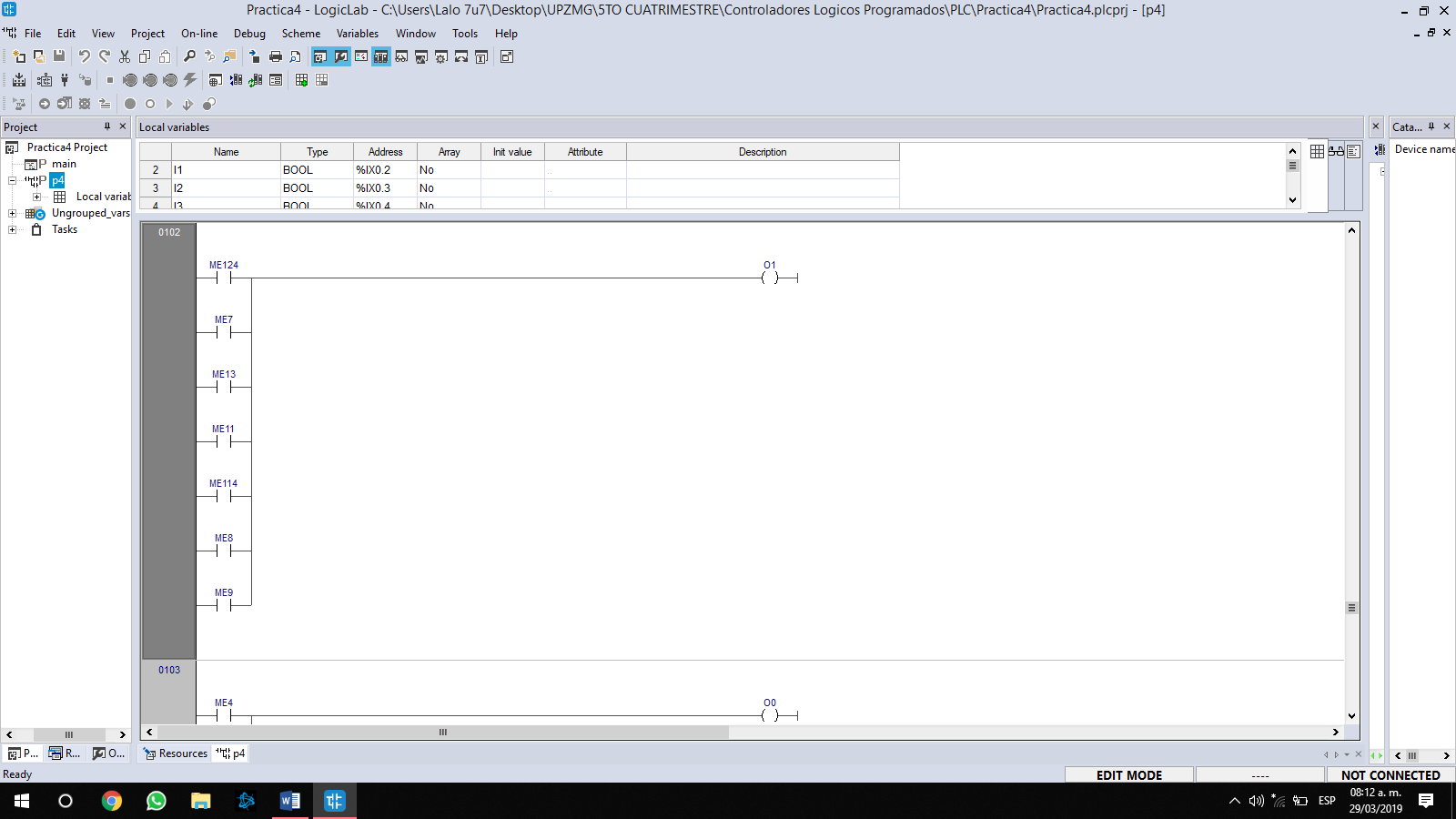


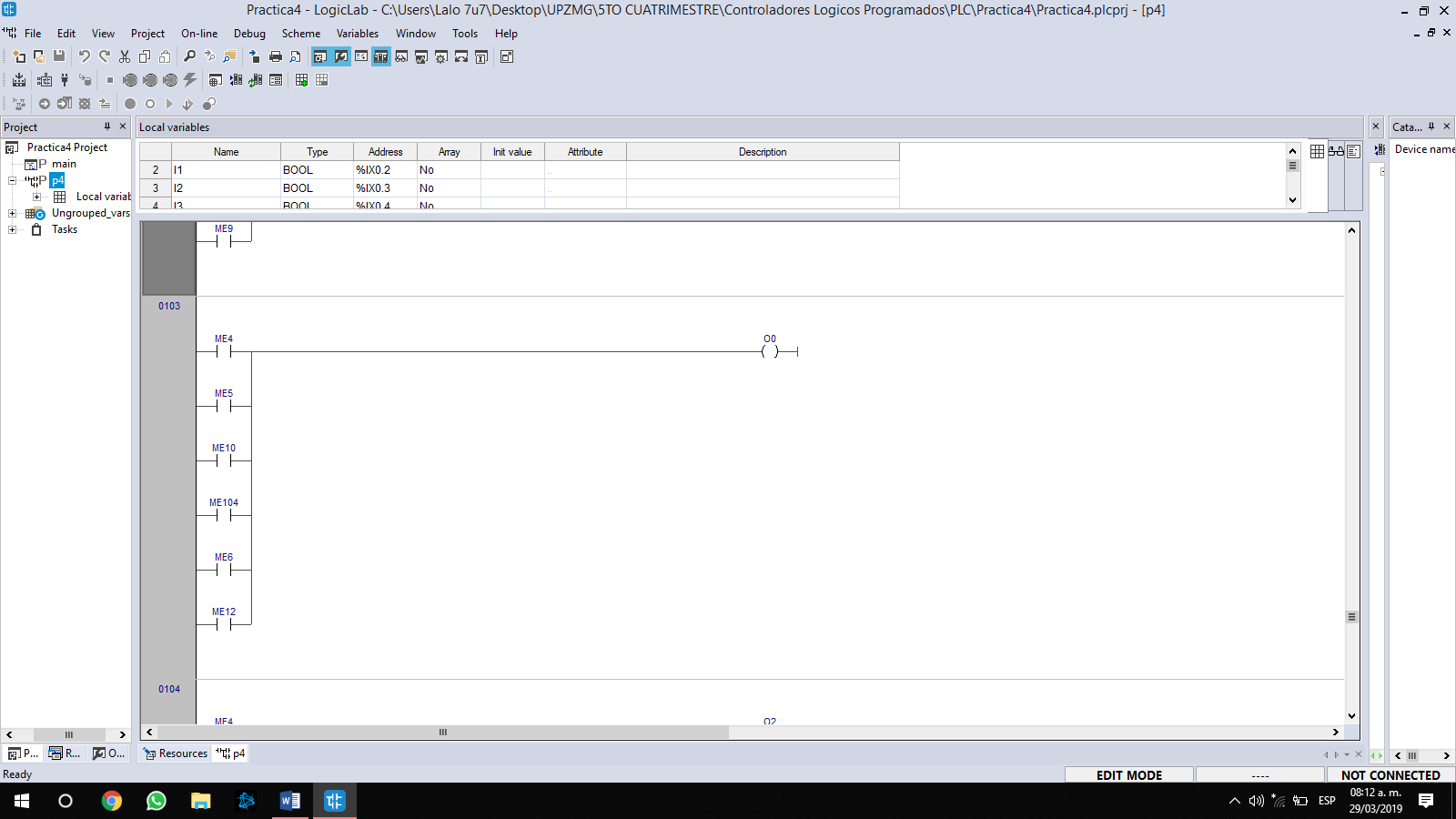


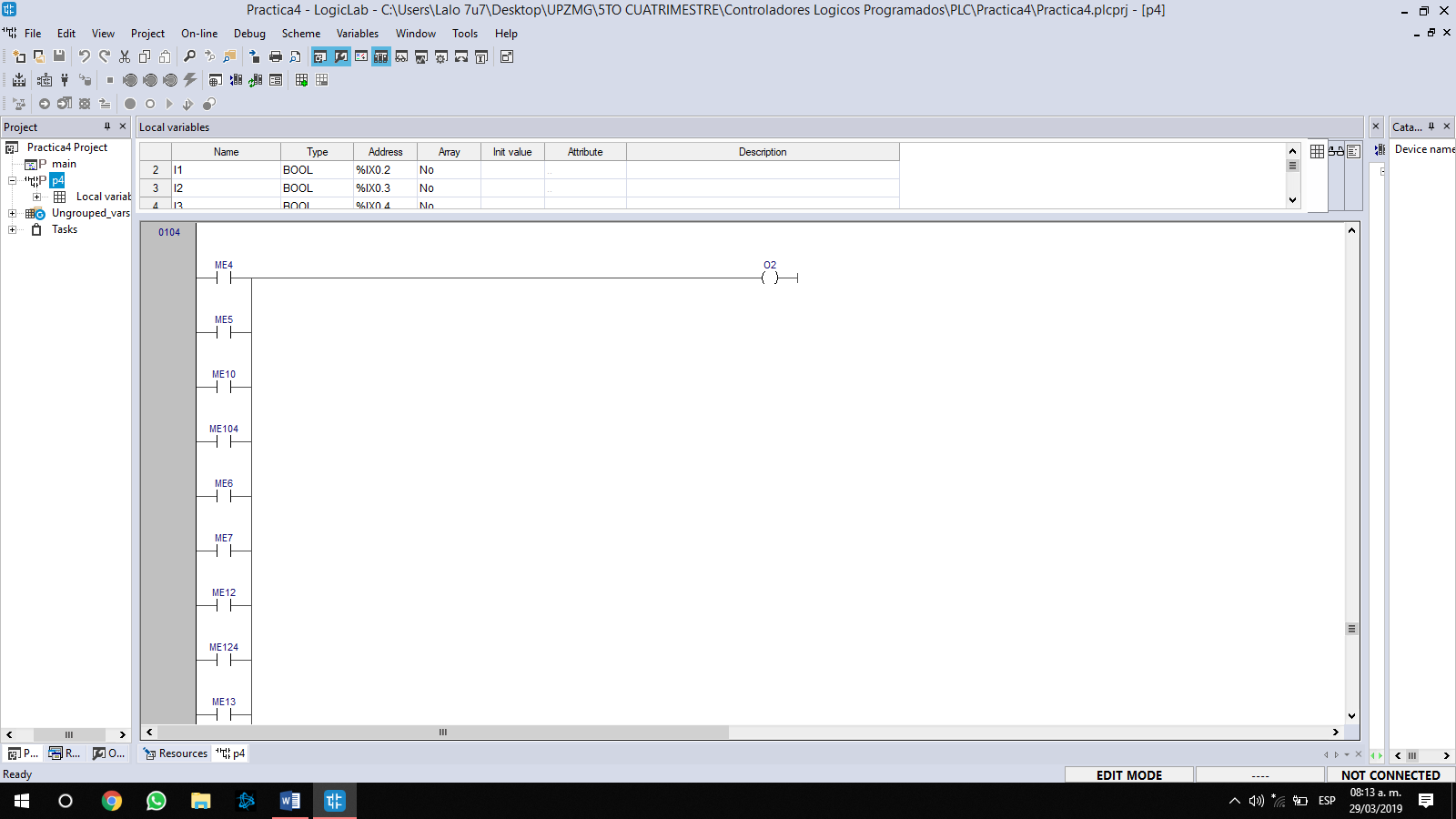


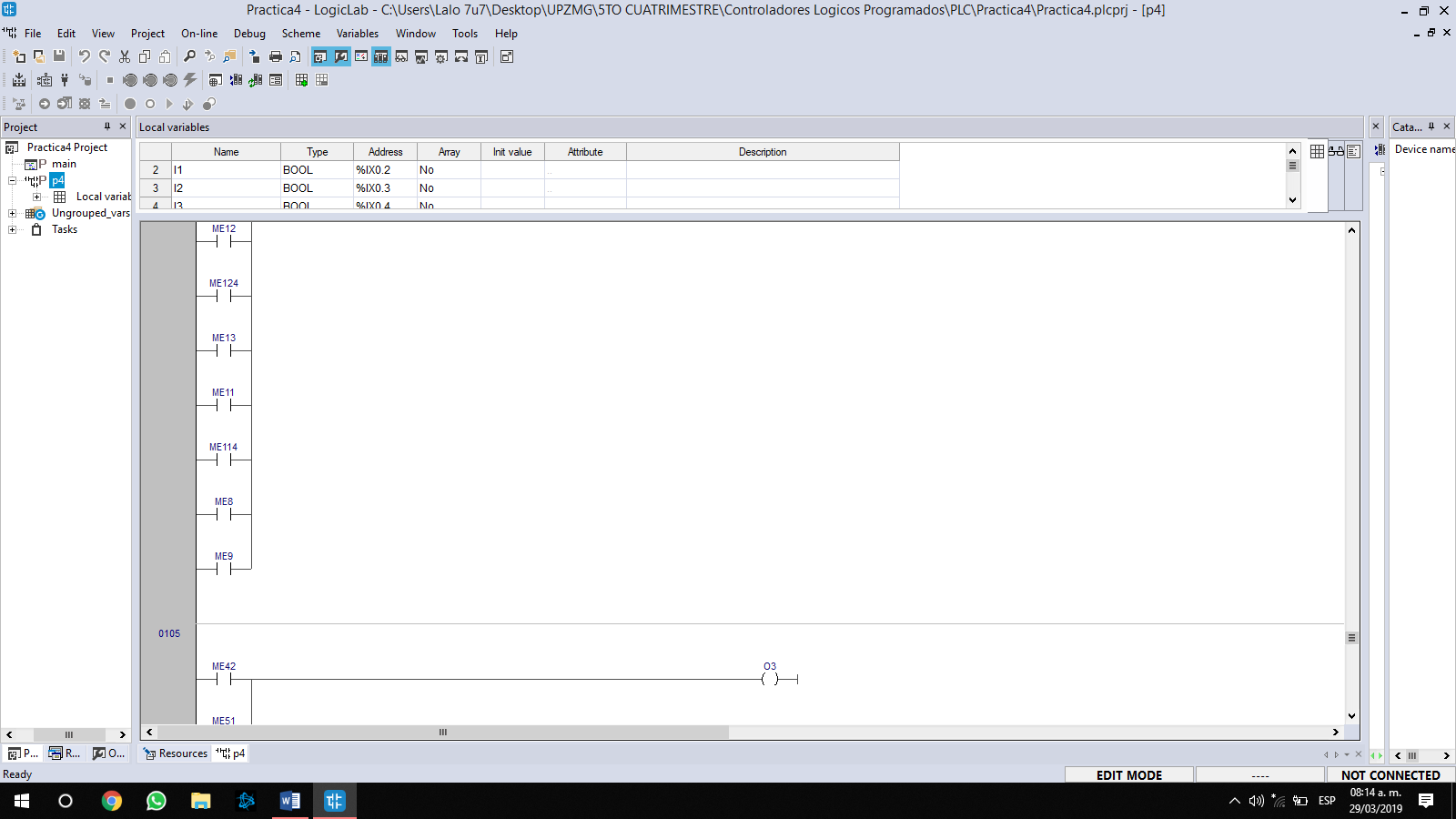


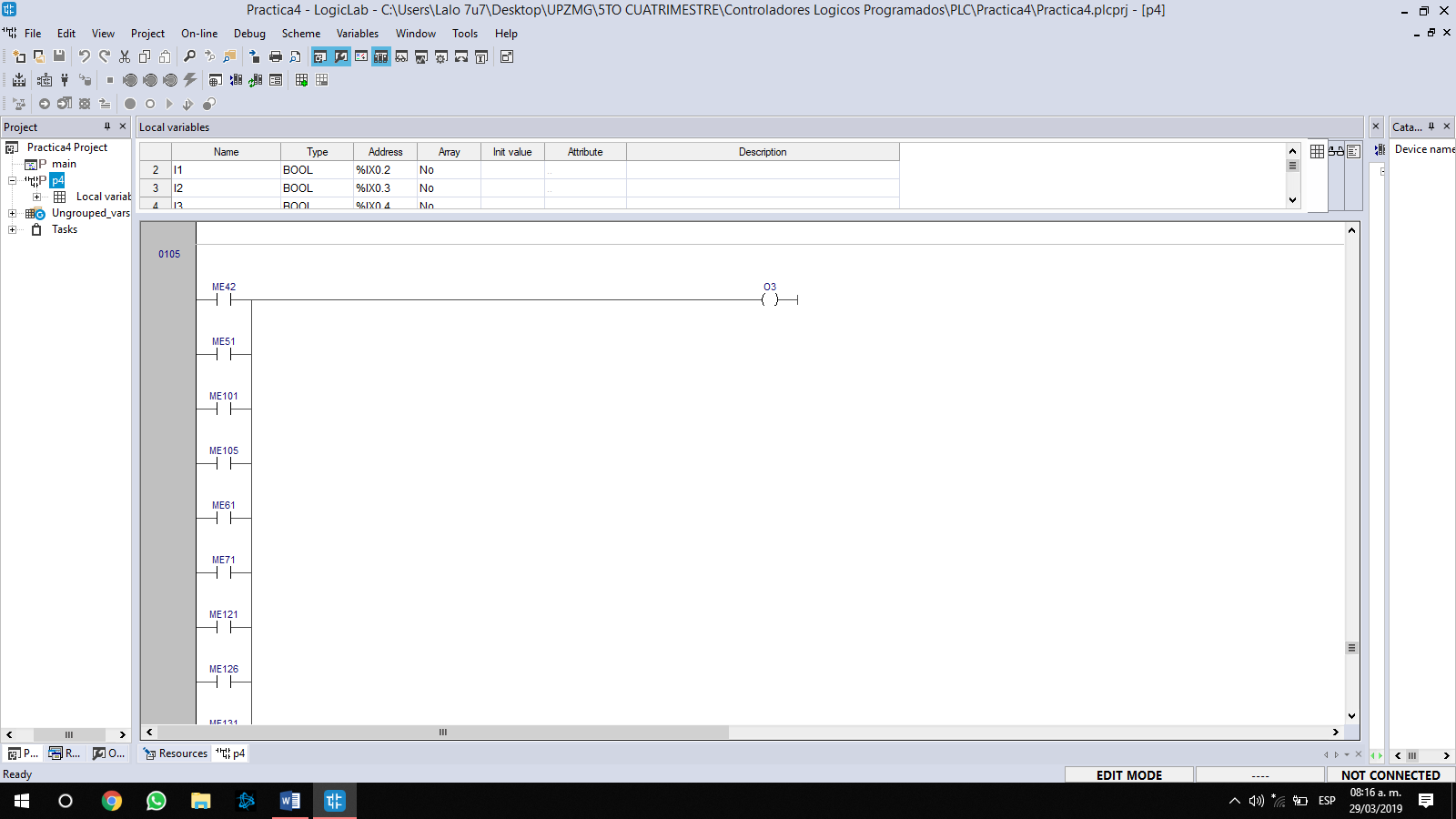


En las acciones declaramos que memorias de estados activan dicha salida.  


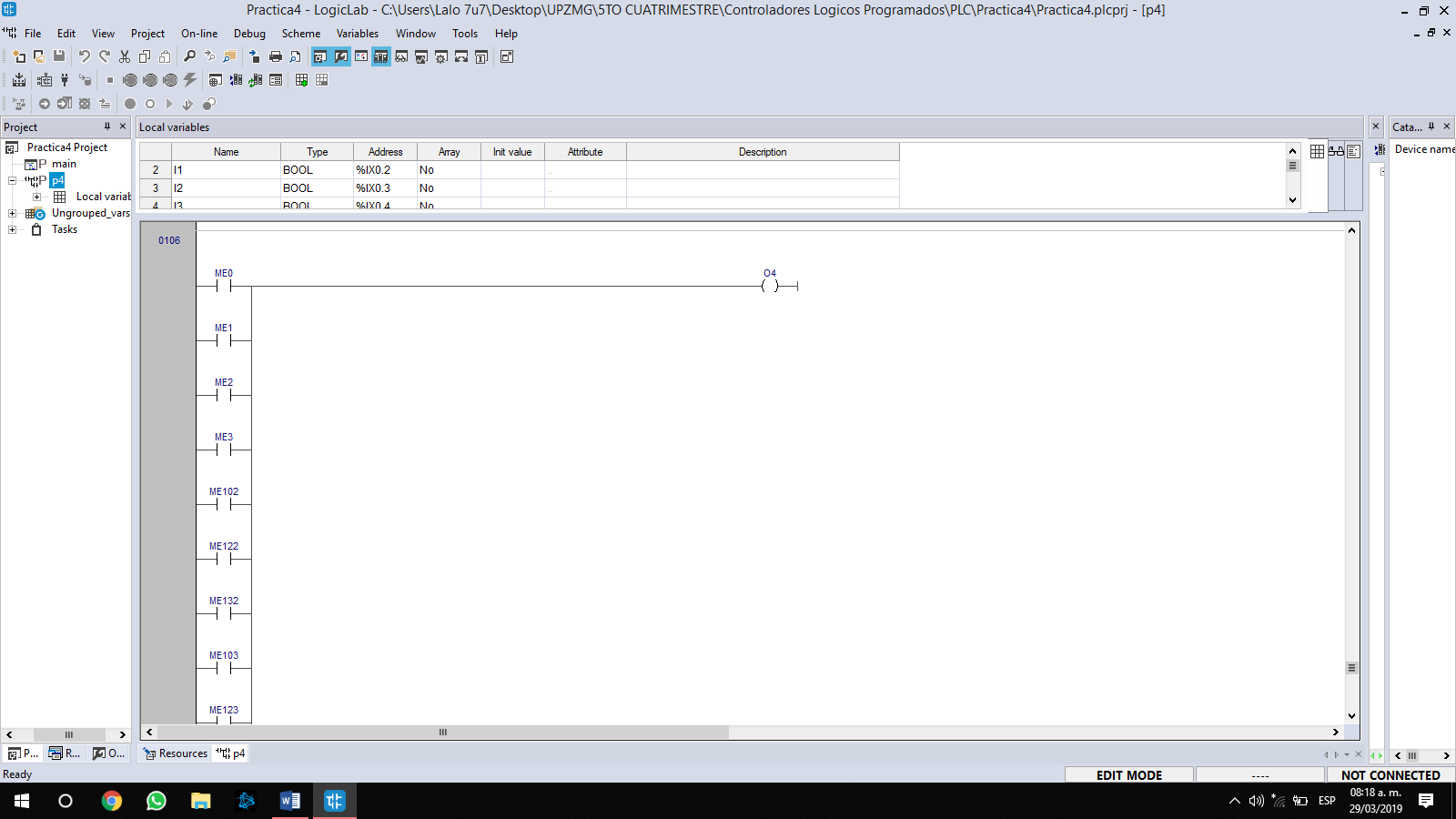


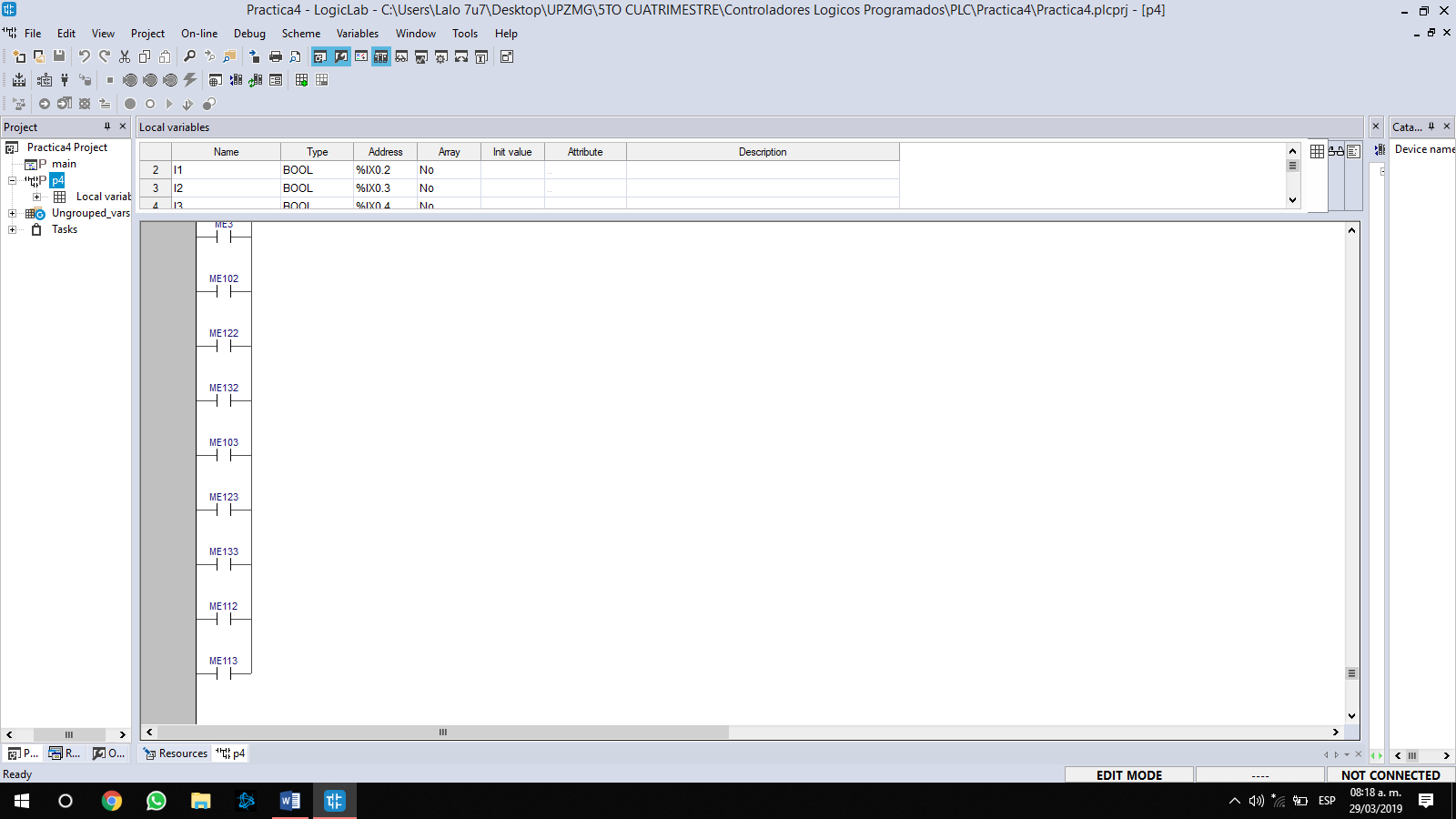


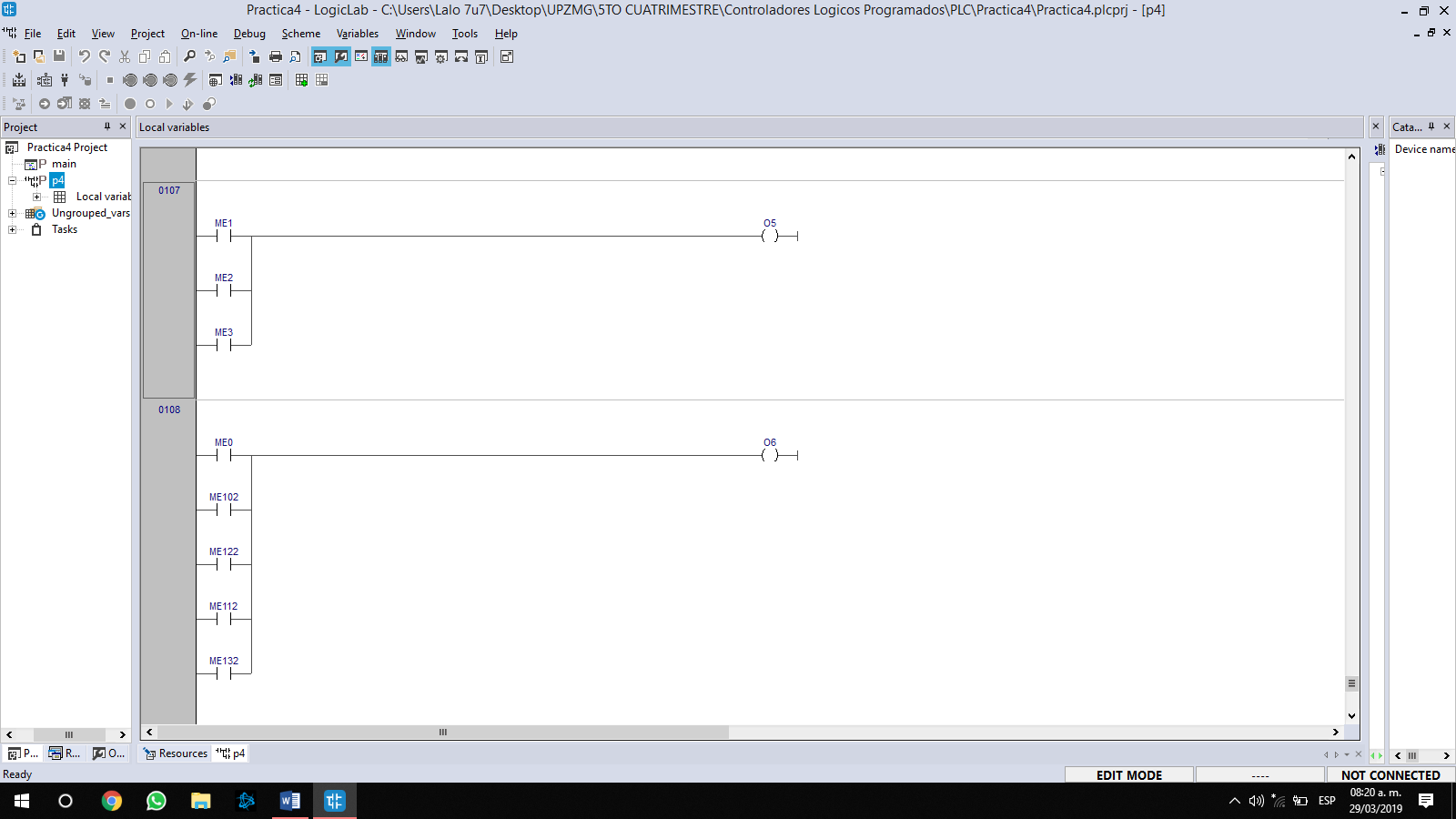










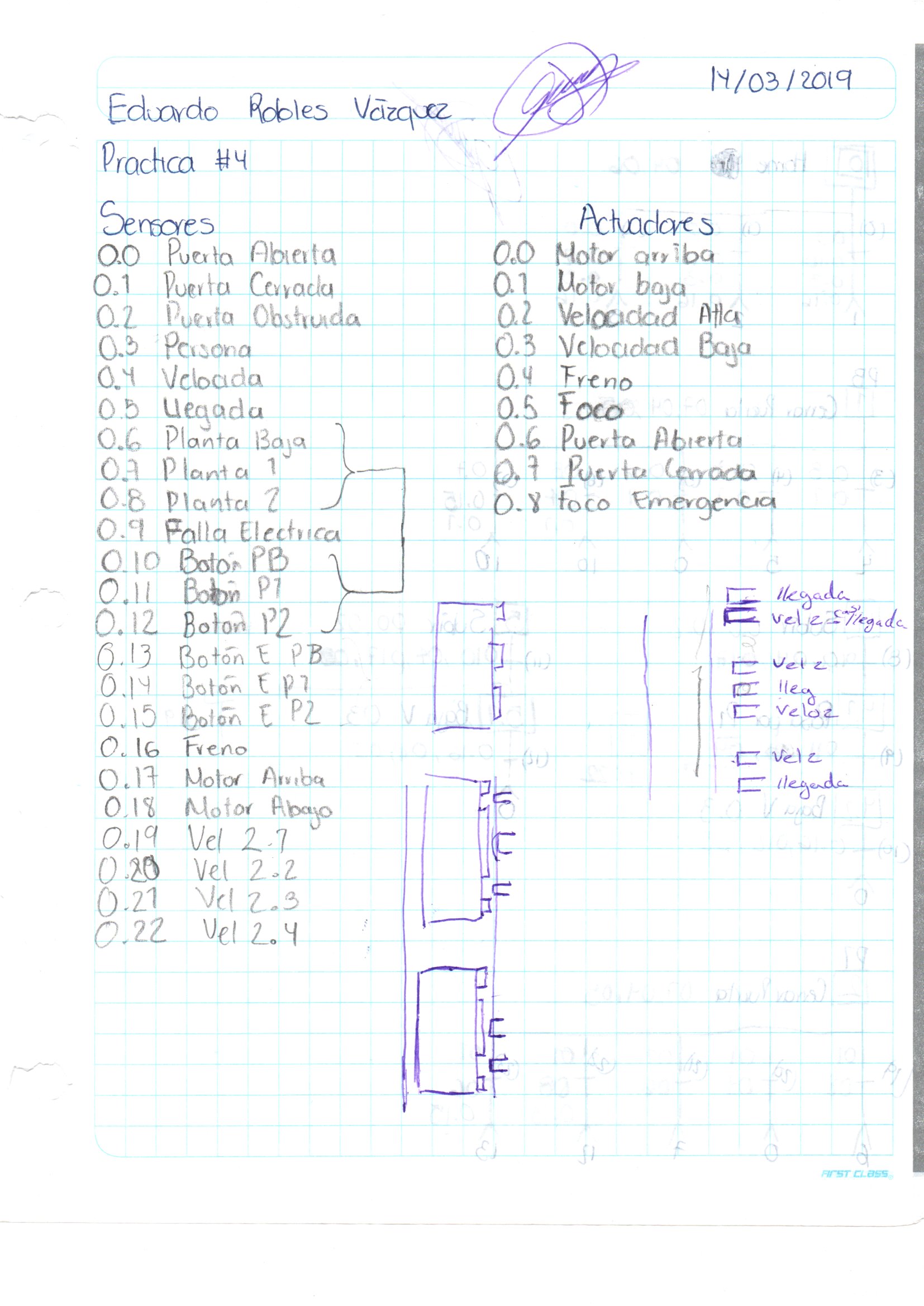


**CONCLUSIONES:**

**Víctor Tapia**: Ha sido la práctica más difícil, aplicamos todo lo aprendido hasta la fecha, el GRAFCET consto de varias subdivisiones conectadas entre sí para cumplir las condiciones propuestas en problema y el diagrama ladder resulto ser mucho más extenso de lo esperado, inclusive tuvo ciertos errores en la simulación debido al extenso número de variables con el cual este contaba.

**Eduardo Robles Vázquez:** Esta práctica ha sido todo un reto para nosotros por la complejidad que representa, en cuestión de conocimientos lo más difícil fue el diseño del elevador y todas sus funciones, lejos de eso el convertir el GRAFCET a programación Ladder no represento mucha dificultad puesto que es un tema que ya dominamos bien.

**FIRMA DE ENTREGA**

****