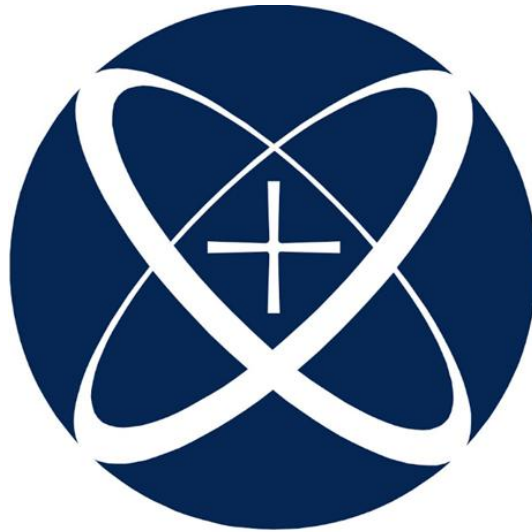


Instituto Tecnológico y de Estudios Superiores de Occidente, Tlaquepaque, Jalisco.

## **ACTIVIDAD\_1.3.0 DIAGRAMAS DE FLUJO Y MAQUINAS DE ESTADO FINITO**

*Simulación Matemática*  
*Alejandro Noel Hernández Gutiérrez*  
*V2025\_MAF2869A*



# ITESO

Universidad Jesuita  
de Guadalajara

Fernando Sacamo Ceniseros - Núm. Exp. 744604

FECHA ENTREGA: 04/06/2025

## DESCRIPCION PROCESO CREATIVO

El enfoque parte del análisis de caso de un automóvil estándar desglosando paso a paso el proceso de encendido, avance, cambios y todo aquel escenario posible como falla de batería o apagado del motor siendo presentado mediante un diagrama de flujo para una visualización más clara y estructurada

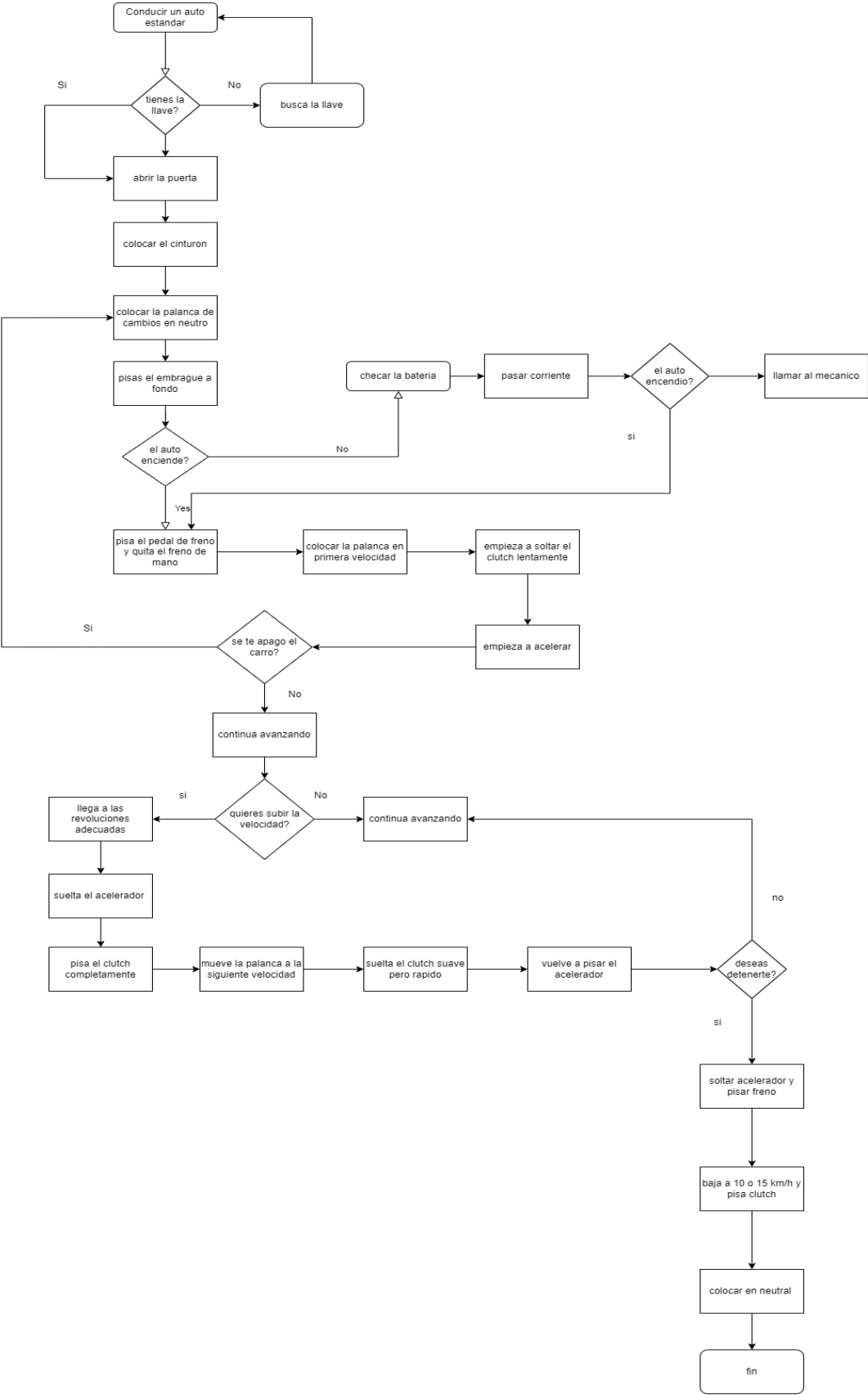
El desarrollo de la segunda actividad parte de una máquina de estados finitos aplicadas a un torniquete definiendo sus transiciones y estados, se implementó en Python con una estructura optimizada usando una tabla y ejemplificando de manera más clara en un diagrama de flujo.

## ACTIVIDAD 1

Crear un diagrama de flujo que represente el proceso de conducir un auto estándar. Considera los siguientes pasos:

- Abrir la puerta
- Colocar el cinturón
- Encender el auto
- Escenario en que no enciende el auto y se pasa batería
- Escenario en que enciende el auto
- Subir velocidad
- Detener el auto

DIAGRAMA:



## ACTIVIDAD 2

Implementar una Máquina de Estados Finitos (FSM) en Python que simule el funcionamiento de un torniquete con los siguientes estados y transiciones:

- Abstrae el proceso del funcionamiento de un torniquete en un diagrama de flujo.
- Estado inicial: "Bloqueado"
- Transiciones:
  - "Echar moneda" (1) -> Estado pasa a "Desbloqueado"
  - "Empujar" (2) en estado "Bloqueado" -> Sin efecto
  - "Empujar" (2) en estado "Desbloqueado" -> Estado pasa a "Bloqueado"
  - "Echar moneda" (1) en estado "Desbloqueado" -> Sin efecto
- La FSM debe ejecutarse en un bucle, mostrar el estado actual y solicitar entrada del usuario.

### CODIGO PYTHON

Estado inicial	entrada	Estado alcanzado	Salida
bloqueado	moneda	Desbloqueado	Desbloquea el torniquete
bloqueado	empujar	Bloqueado	Sin efecto
desbloqueado	moneda	Desbloqueado	Sin efecto
desbloqueado	empujar	bloqueado	Requiere moneda

### CODIGO:

```
class Torniquete:
    def __init__(self):
        self.estado = "Bloqueado"

    def procesar_entrada(self, entrada):
        if entrada == 1:
            if self.estado == "Bloqueado":
```

```
        self.estado = "Desbloqueado"
        print("Moneda aceptada. Torniquete desbloqueado.")
    else:
        print("Ya está desbloqueado. No necesitas otra moneda.")
    elif entrada == 2:
        if self.estado == "Bloqueado":
            print("Torniquete bloqueado. No puedes pasar.")
        else:
            self.estado = "Bloqueado"
            print("Pasaste. Torniquete bloqueado nuevamente.")
    else:
        print("Entrada inválida. Usa 1 para echar moneda o 2 para
empujar.")

def ejecutar(self):
    while True:
        print(f"\nEstado actual: {self.estado}")
        try:
            entrada = int(input("¿Qué deseas hacer? (1 = Echar moneda, 2 =
Empujar): "))
            self.procesar_entrada(entrada)
        except ValueError:
            print("Por favor ingresa un número válido.")

torniquete = Torniquete()
torniquete.ejecutar()
```

PANTALLA DE EJECUCION

```
Estado actual: Bloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 2
Torniquete bloqueado. No puedes pasar.

Estado actual: Bloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 1
Moneda aceptada. Torniquete desbloqueado.

Estado actual: Desbloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 1
Ya está desbloqueado. No necesitas otra moneda.

Estado actual: Desbloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 2
Pasaste. Torniquete bloqueado nuevamente.

Estado actual: Bloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 2
Torniquete bloqueado. No puedes pasar.

Estado actual: Bloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 1
Moneda aceptada. Torniquete desbloqueado.

Estado actual: Desbloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): 3
Entrada inválida. Usa 1 para echar moneda o 2 para empujar.

Estado actual: Desbloqueado
¿Qué deseas hacer? (1 = Echar moneda, 2 = Empujar): s
Por favor ingresa un número válido.
```

DIAGRAMA

