# Documento Explicativo para Sustentación: Módulo de Tramas

Este documento explica el funcionamiento del módulo de tramas implementado en Python para el proyecto final de Matemáticas Discretas II, orientado a la sustentación.

## 📄 Estructura General del Módulo

Este módulo simula una transmisión binaria mediante tramas, validada por una máquina de estados finitos (FSM). Permite:

1. Ingresar una lista de validación (valores de referencia para validar las tramas).
2. Ingresar tramas de 32 bits (entre 5 y 20).
3. Validar cada trama según condiciones lógicas.
4. Evaluar si la transmisión completa es correcta.

## 🔖 Representación Interna

* **Lista de validación**: lista de enteros (0 a 15), obtenidos a partir de 4 bits ingresados por el usuario.
* **Tramas**: cada trama es una cadena de 32 caracteres compuestos por ‘0’ o ‘1’.

## ⚙️ Funciones del Módulo

### 1. ingresar\_lista\_validacion()

* El usuario ingresa cuántos valores quiere (hasta 20).
* Cada valor se forma con 4 bits ingresados manualmente (uno por uno).
* Los bits se convierten a un entero con int(bits, 2).

### 2. ingresar\_tramas()

* El usuario indica cuántas tramas desea ingresar (entre 5 y 20).
* Cada trama se construye bit a bit, validando que solo sean ‘0’ o ‘1’.
* Se guardan como strings de 32 caracteres.

### 3. validar\_tramas(tramas, lista\_validacion)

* Para cada trama:
  + Se extraen los **bits 10 al 14** (es decir, posiciones 9 a 13).
  + Se convierte ese subgrupo en un número entero.
  + Se verifica:
    - Si es múltiplo de 3.
    - Si **ese número + el valor de la lista de validación** es múltiplo de 5.
* Devuelve una lista indicando para cada trama si fue “Válida” o “Inválida”.

### 4. evaluar\_transmision(resultados)

* Cuenta cuántas tramas fueron inválidas.
* Calcula el porcentaje de error:

error = (inválidas / total) \* 100

* Si el error es **menor al 20%**, la transmisión es correcta. De lo contrario, se considera fallida.

## 📈 Argumentos para la Sustentación

* “Diseñé una FSM sencilla donde cada trama pasa por lectura y validación.”
* “Extraigo los bits 10 a 14 como lo pide el enunciado para calcular el valor clave de cada trama.”
* “Sumo ese valor con la lista de validación porque eso representa una condición externa a la trama.”
* “Verifico si son múltiplos de 3 y 5 para decidir si cada trama es válida.”
* “La decisión final de transmisión correcta o incorrecta se basa en un porcentaje de error, lo cual simula control de calidad en sistemas reales.”

## 🌟 Conclusión

Este módulo cumple completamente la parte del proyecto relacionada con:

* FSM para validación binaria.
* Ingreso manual de datos bit a bit.
* Verificación estructurada y defendible.

Está escrito con funciones simples, código limpio y validaciones claras. Fácil de explicar en la sustentación y alineado con la rúbrica del curso.