

# Teoría de Autómatas y Lenguajes Formales

Unidad 1 — Sesión 6 (15-ene-2026): Expresiones regulares (RE) + verificación + micro-lab

Docente: Helder Octavio Fernández Guzmán

- Tema central: **Expresiones regulares (RE)** como **descripciones de lenguajes**.
- Meta práctica: **leer, construir y verificar** RE con casos de prueba.
- Herramientas:
  - **Automata Tutor: Words in RE** (verificación) y **RE to  $\epsilon$ -NFA** (demo).
- Bloque final: **Micro-lab** con evidencia mínima.

# Warm-up/calentamiento (prerrequisitos mínimos)

Periodo 1/6

- Alfabeto:  $\Sigma = \{0, 1\}$ .
- $\Sigma^*$ : todas las cadenas sobre  $\Sigma$  (incluye  $\varepsilon$ ).
- Lenguaje: **conjunto de cadenas**.
- Operaciones (intuición): **unión y concatenación**.

**Objetivo:** asegurar base común antes de RE.

# Warm-up/Calentamiento: 4 preguntas rápidas

Periodo 1/6

- 1 Si  $\Sigma = \{0, 1\}$ , ¿ $1010 \in \Sigma^*$ ? ¿y 2?
- 2 ¿ $\epsilon$  es una cadena? ¿cuál es su longitud?
- 3 Si  $L_1 = \{0, 00\}$  y  $L_2 = \{1\}$ , ¿cuál es  $L_1 \cdot L_2$ ?
- 4 Si  $L = \text{"cadenas que terminan en 1"}$ , da 2 ejemplos que pertenecen y 2 que no.

# Warm-up/Calentamiento: respuestas (para alinear conceptos)

Periodo 1/6

- (1)  $1010 \in \Sigma^*$  (usa solo símbolos de  $\Sigma$ ).  $2 \notin \Sigma^*$ .
- (2)  $\varepsilon$  sí es una cadena;  $|\varepsilon| = 0$ .
- (3)  $L_1 \cdot L_2 = \{01, 001\}$ .
- (4) Pertenecen: 1, 000001. No pertenecen:  $\varepsilon$ , 10.

**Nota:** un lenguaje es un conjunto (no un diagrama).

# Objetivos de aprendizaje (Sesión 6)

Periodo 1/6

Al final de la sesión, podrás:

- Interpretar una RE como **lenguaje** (decidir pertenencia de cadenas).
- Construir RE desde enunciados frecuentes (patrones típicos).
- Verificar RE con **casos de prueba** (no solo “a ojo”).
- Conectar RE con un  $\varepsilon$ -NFA equivalente (demo con herramienta).

- Unión:  $|$
- Concatenación: implícita (ej.: 01 significa 0 seguido de 1)
- Estrella de Kleene:  $*$  (cero o más repeticiones)
- Paréntesis:  $(\dots)$  para agrupar

**Convención:**  $\Sigma = \{0, 1\}$  y  $\Sigma^* \equiv (0|1)^*$ .

# Precedencia (evita la mayoría de errores)

Periodo 2/6

- Orden de precedencia:

$*$   $>$  concatenación  $>$   $|$

- Regla de oro: si dudas, **usa paréntesis**.

**ejercicio:** compara  $0|11^*$  vs  $(0|11)^*$  vs  $0|(11^*)$ .

# RE como lenguaje: “acepta” / “rechaza”

Periodo 2/6

Ejemplo:

$$R = (0|1)^*101(0|1)^*$$

- Acepta: 101, 1101011
- Rechaza:  $\varepsilon$ , 1001

**Idea:** una RE describe un **conjunto de cadenas**.

# Plantillas “de supervivencia” (para construir RE)

Periodo 3/6

- “contiene X”  $\rightarrow \Sigma^* X \Sigma^*$
- “empieza con X”  $\rightarrow X \Sigma^*$
- “termina en X”  $\rightarrow \Sigma^* X$
- “repite bloque B”  $\rightarrow (B)^*$
- “A o B”  $\rightarrow (A|B)$

**Consejo:** primero escribe el enunciado como piezas; luego traduce a RE.

# Ejemplo guiado 1: “contiene 101”

Periodo 3/6

- Lenguaje: cadenas binarias que contienen la subcadena 101.
- Construcción por plantilla:

$$\Sigma^*101\Sigma^* \equiv (0|1)^*101(0|1)^*$$

- Pruebas rápidas:
  - Acepta: 101, 0101
  - Rechaza: 10, 11111

## Ejemplo guiado 2: “termina en 00 o 11”

Periodo 3/6

- Lenguaje: cadenas que terminan en 00 o en 11.
- Construcción:

$$\Sigma^*(00|11) \equiv (0|1)^*(00|11)$$

- Pruebas rápidas:
  - Acepta: 00, 01011
  - Rechaza:  $\varepsilon$ , 1010

## Ejemplo semi-guiado 3: bloques 01 repetidos

Periodo 3/6

- Lenguaje: cadenas formadas por bloques 01 repetidos (incluye  $\varepsilon$ ).

- RE:

$(01)^*$

- Pruebas rápidas:

- Acepta:  $\varepsilon$ , 0101
- Rechaza: 0, 010

Checklist mínimo de pruebas (8 casos):

- 2 bordes:  $\varepsilon$  y el mínimo que debería aceptar.
- 2 “casi cumple”: falla por 1 símbolo.
- 2 casos medios.
- 2 casos largos / con repetición.

**Regla:** si aparece un contraejemplo, ajusta la RE (o tu interpretación del enunciado).

- ➊ Ingresar la RE.
- ➋ Probar una lista de cadenas (positivas y negativas).
- ➌ Si falla, ajustar RE o pruebas (buscar contraejemplos).
- ➍ Guardar evidencia mínima (según criterio del curso).

**Logro esperado:** RE correcta + pruebas coherentes.

## Demo breve: RE $\rightarrow$ $\varepsilon$ -NFA (equivalencia de lenguaje)

Periodo 4/6

- Convertiremos una RE a un  $\varepsilon$ -NFA usando Automata Tutor.
- Probaremos 4 cadenas (2 acepta / 2 rechaza) para confirmar:

$$L(\text{RE}) = L(\varepsilon\text{-NFA})$$

**Mensaje clave:** cambia la representación, no el lenguaje.

- Plataforma: **Moodle (Quiz)**
- Preguntas: **8**
- Tiempo: **10–12 min**
- Intentos: **1**
- Objetivo: detectar errores típicos antes de iniciar el micro-lab.

**Regla de uso:** si fallas en precedencia/paréntesis o en “contiene vs termina”, ajusta tu idea antes de iniciar el micro-lab.

# Micro-lab S6: construir y verificar (evidencia mínima)

Periodo 5/6

- Alfabeto:  $\Sigma = \{0, 1\}$ .
- Lenguaje: cadenas que
  - **empiezan** con 1, y
  - **contienen** la subcadena 00 (en cualquier posición).
- Evidencia mínima (entrega):
  - 1 RE final.
  - 2 8 casos de prueba (4 acepta / 4 rechaza) con justificación breve.
  - 3 Evidencia en Automata Tutor: **Words in RE**.

**Micro-lab:** 9:45–11:15 (trabajo del estudiante).

# Checklist del micro-lab (antes de validar)

Periodo 5/6

Antes de validar, confirma:

- 1 ¿Tu RE **obliga** a iniciar con 1?
- 2 ¿Tu RE permite 00 **en cualquier posición** (no solo al final)?
- 3 ¿Estás usando correctamente  $\Sigma^* = (0|1)^*$ ?
- 4 ¿Tus 8 pruebas incluyen bordes y “casi cumple”?

**Tip:** si una prueba falla, produce un contraejemplo y ajusta.

- Hoy:
  - leímos RE como lenguajes (acepta/rechaza),
  - construimos RE con plantillas típicas,
  - verificamos con casos de prueba y herramienta.
- Mañana (Sesión 7):
  - **propiedades de clausura** de lenguajes regulares,
  - práctica integrada (RE + justificación + verificación).