

# Taller-Fundamentos

---

## Máquina de Turing Física — Suma y Resta en Unario

---

Representación física de una **máquina de Turing** que ejecuta **suma ( $A+B$ )** y **resta ( $A-B, A \geq B$ )** en **unario** usando Arduino, LEDs y un cabezal móvil.

### Idea general

- **Controlador:** Arduino Uno que implementa la lógica de transición (leer símbolo, escribir símbolo, mover L/R, cambiar de estado).
- **Cinta:** hilera de LEDs (LED ENCENDIDO = **1**, LED APAGADO = **\_**).
- **Cabezal:** carrito con LDR que “lee” el LED actual; la “escritura” la realiza el Arduino encendiendo/apagando ese LED.
- **Movimiento:** motor paso a paso 28BYJ-48 + driver ULN2003 (desplazamiento celda a celda).
- **Límites:** 2 limit switches para homing y extremos.
- **Interacción:** botones (Step, Reset, Modo Suma/Resta, Izq/Der) y LEDs de estado ( $q_0, q_1, q_2, \dots$ ).
- **Alimentación:** 5 V externa (motor + LEDs), GND común.

### Representación formal utilizada

- **Alfabeto de entrada:**  $\Sigma = \{ \text{1}, \text{_} \}$

### Propuesta de materiales

- Arduino Uno
- Protoboard + jumpers
- LEDs (10–15) + resistencias 220–330  $\Omega$
- LDR + resistencia 10 k $\Omega$
- 28BYJ-48 + ULN2003
- Riel (MGN12 u otro) + carrito
- 2 limit switches
- Botones/switches (4–6)
- LEDs extra para estados (4–6)
- Fuente 5 V externa
- Base de madera

### Propuesta de diseño (resumen)

- **Cinta:** fila de LEDs; cada LED es una celda (**1** = encendido, **\_** = apagado).
- **Cabezal:** carro con LDR que lee ON/OFF; escritura por Arduino sobre la celda actual.
- **Control y estados:** botones para Step/Reset/Modo; LEDs de estado para  $q_0, q_1, q_2, \dots$
- **Alimentación y montaje:** 5 V externa, protoboard y base rígida (madera).

# Fases del proyecto

## Fase 1 — Informe teórico

- Análisis teórico
- Definición del autómata: 7-tupla  $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$  y tablas de transición completas para **suma y resta ( $A \geq B$ )**

### Encargados

- Investigación de máquina de Turing: **Benjamín Cuello**
- Estados y 7-tupla: **Benjamín Salas**

## Fase 2 — Selección de materiales y diseño del sistema

- Selección y justificación de materiales y componentes.
- Diseño de la arquitectura física: cinta, movimiento del cabezal, lectura/escritura.
- Diagramas técnicos con dimensiones y funcionamiento de cada componente.

## Fase 3 — Construcción, programación y pruebas

- Ensamblaje del hardware y montaje mecánico.
- Programación de la lógica .
- Pruebas de funcionamiento y demostraciones controladas.

---- Actualizacion ----

## Simulación Digital — Máquina de Turing en Unity

Esta es la **versión digital interactiva** de la Máquina de Turing, programada completamente en **Unity (2022.3 LTS)**.

Esta simulación representa fielmente el funcionamiento del modelo físico, pero permite visualizar con mayor precisión los **estados, movimientos y escritura sobre la cinta**.

### Interfaz y controles principales

Botón / Control	Función
<b>Left / Right</b>	Mueve manualmente el cabezal una celda a la izquierda o derecha.
<b>Toggle</b>	Cambia el símbolo de la celda actual (ciclo: $B \rightarrow 1 \rightarrow 0 \rightarrow X \rightarrow B$ ).
<b>Step</b>	Ejecuta una única transición de estado (una “instrucción” de la tabla $\delta$ ).
<b>Run / Stop</b>	Inicia o detiene la ejecución automática continua.
<b>Speed Slider</b>	Controla la velocidad de ejecución durante el modo automático (más rápido hacia la derecha).

Botón / Control	Función
<b>Reset</b>	Limpia toda la cinta (deja todas las celdas en blanco <b>B</b> ) y vuelve el cabezal a la posición inicial.
<b>Suma / Resta</b>	Cambia entre las tablas de transición $\delta$ correspondientes a las operaciones de <b>suma (<math>A+B</math>) o resta (<math>A-B</math>)</b> .

## Representación de la cinta

- La **cinta** se compone de **23 celdas** (cubos), cada una representando un símbolo del alfabeto **{B, 1, 0, X}**.
- El **color del cubo** indica el símbolo almacenado en esa celda:

Color	Símbolo	Significado
Gris	<b>B</b>	Celda en blanco / sin valor.
Rojo	<b>1</b>	Unidad en unario (número).
Azul	<b>0</b>	Separador (solo en resta).
Naranja	<b>X</b>	Celda tachada o restada (valor eliminado).
Amarillo	—	Cabezal de lectura/escritura.

## Funcionamiento general

- Al iniciar, la máquina comienza en el estado inicial (**q0**) y con la cinta en blanco (**B...B**).
- El usuario puede escribir un número unario en la cinta usando **Toggle**, por ejemplo:
  - Suma: **1110111B...** → representa **3 + 3**.
  - Resta: **111011B...** → representa **3 - 2**.
- Al presionar **Step** o **Run**, el cabezal lee la celda actual, ejecuta la transición  $\delta$  correspondiente (según la tabla cargada) y:
  - Cambia de estado (**q0, q1, q2, ...**).
  - Escribe un nuevo símbolo (**1, X, B, etc.**).
  - Se mueve izquierda/derecha según la regla.
- El proceso continúa hasta llegar al **estado final qf**, donde la máquina se detiene automáticamente.
  - En suma: la cinta muestra la concatenación total de unos (**1**) equivalente a  $A+B$ .
  - En resta: las celdas resultan en **B** si el resultado es 0, o **1s** equivalentes a  $A-B$ .

## Indicadores visuales

- En la parte superior izquierda se muestra el **estado actual**:
 

**Estado: q3 (Resta) [Running]**

  - El texto entre paréntesis indica el modo activo (**Suma o Resta**).
  - El estado entre corchetes indica si la ejecución está **Running** o **Stopped**.

- El **cabezal** se desplaza visualmente entre las celdas, mostrando el proceso de lectura/escritura en tiempo real.
- 

## Lógica interna (Unity)

- **Cell.cs**: representa cada celda de la cinta, guarda su símbolo y actualiza el color correspondiente.
  - **HeadController.cs**: gestiona el movimiento del cabezal y las operaciones de lectura/escritura.
  - **TuringMachine.cs**: contiene las tablas  $\delta$  para **suma** y **resta**, el control del estado actual y el modo automático.
  - **UIController.cs**: conecta los botones y el texto del HUD (estado, botones, slider de velocidad, etc.).
-