

Aquí tienes la **documentación completa y estructurada** para el taller de Arduino titulado “**Juego de los Topos**”, diseñada para ser clara, didáctica y replicable por participantes. Puedes usarla como material en PDF, anexo o guía en clase.

Juego de los Topos con Arduino UNO

Taller práctico de electrónica y programación

Descripción General

En este taller aprenderás a construir un **juego interactivo tipo “Whack-a-Mole”** (Juego de los Topos) utilizando Arduino UNO, LEDs, botones y una pantalla LCD. El objetivo del juego es reaccionar rápidamente pulsando el botón correcto cuando un LED (topo) se enciende. A medida que pasa el tiempo, la dificultad aumenta, ¡poniendo a prueba tus reflejos!

Los participantes desarrollarán habilidades en:

- Programación de Arduino con `millis()`
 - Control de entradas y salidas digitales
 - Uso de pantallas LCD I2C
 - Diseño de lógica de juego
-

Materiales Necesarios

Electrónicos

- 1 placa **Arduino UNO**
- 6 **LEDs** (rojo, verde, azul, etc.)
- 6 **resistencias de 220 Ω** para los LEDs
- 6 **botones pulsadores**
- 6 **resistencias de 10k Ω** (si no usas INPUT_PULLUP)
- 1 **Pantalla LCD 16x4 con módulo I2C**
- 1 **botón START**
- 1 **botón RESET**
- Cables macho-macho / macho-hembra
- 1 **protoboard**

Opcionales / soporte

- Placa base o cartón para el montaje
 - Cinta adhesiva, bridas, impresiones 3D
 - Caja de proyecto para estética
-

Esquema de Conexiones

Componente	Pin Arduino	Detalles
LED 1	D2	Con resistencia 220 Ω
LED 2	D3	
LED 3	D4	
LED 4	D5	
LED 5	D6	
LED 6	D7	
Botón 1	D8	INPUT_PULLUP
Botón 2	D9	
Botón 3	D10	
Botón 4	D11	
Botón 5	D12	
Botón 6	A0	
Botón START	A1	
Botón RESET	A2	
LCD SDA	A4	Interfaz I2C (0x27)
LCD SCL	A5	

⚠ IMPORTANTE: Usa resistencias para los LEDs y si no usas INPUT_PULLUP, añade resistencias de 10k Ω entre cada botón y GND.

Código del Proyecto

- Utiliza la librería: LiquidCrystal_I2C
- La dirección I2C típica del LCD es 0x27 (puede variar: comprueba con un escáner I2C si no funciona).
- El código está completamente comentado y estructurado en secciones.

 **Puedes encontrar el código completo aquí** (enlace a GitHub o PDF compartido)



Funcionamiento del Juego

1. Al encender, la pantalla LCD muestra el mensaje de bienvenida.
 2. El jugador pulsa el botón **START**.
 3. Comienza una cuenta atrás de 60 segundos.
 4. Se enciende aleatoriamente un LED (representando un topo).
 5. El jugador debe pulsar el botón asociado al LED encendido antes de que se apague.
 6. Si acierta, suma **1 punto**.
 7. Si pulsa mal o no pulsa a tiempo, suma **1 fallo**.
 8. Cada pocos segundos, la duración del LED encendido disminuye, aumentando la dificultad.
 9. Al acabar el tiempo, se muestra el resultado final en la pantalla.
 10. El jugador puede pulsar **RESET** para comenzar otra vez.
-



Ajustes y Personalización

- Cambia la duración inicial del LED (`moleDuration`)
- Modifica los umbrales de dificultad:

```
if (elapsed > 40000) moleDuration = 800;  
else if (elapsed > 20000) moleDuration = 1200;  
else moleDuration = 2000;
```

- Puedes usar menos LEDs (ajustando los arrays).
 - Agrega un buzzer para efectos de sonido (opcional).
 - Cambia mensajes en la pantalla LCD.
-



Montaje Físico

Sugerencias:

- Usa una **tabla de madera/cartón** para colocar LEDs y botones como si fueran topos.
 - Coloca los botones **debajo de los LEDs**, como si los aplastaras.
 - Fija los componentes con cinta de doble cara, silicona caliente o soportes 3D.
 - Opcional: agrega una carcasa impresa o hecha a mano para estética.
-

Siguintes pasos

- Añadir sonidos con un **buzzer activo**.
 - Guardar la mejor puntuación con **EEPROM**.
 - Crear **niveles de dificultad** con LEDs de diferentes colores.
 - Usar una **pantalla OLED o TFT** para mostrar gráficos.
 - Crear una **versión multijugador**.
 - Medir el tiempo de reacción.
-

Consideraciones Finales

- Verifica los pines si usas una placa distinta a Arduino UNO.
 - Si la LCD no muestra texto, revisa:
 - Dirección I2C
 - Conexiones SDA/SCL
 - Contraste del potenciómetro (si aplica)
 - Usa `Serial.print()` para depurar si algo no funciona.
 - Evita usar `delay()` en el bucle principal: el juego usa `millis()` para ser no bloqueante.
-

Autores y Créditos

Taller de Introducción a Arduino - Juego de los Topos Organizado por:
EuskalEncounter 33 y Desarrollado por: Víctor Pardo Fecha: [25/07/2025]
