🎮 Juego de Simón con Arduino

Proyecto educativo de electrónica interactiva con LEDs, relés y pantalla LCD

🎉 Descripción General del Proyecto

Este proyecto consiste en replicar el clásico **"Juego de Simón"** usando una placa **Arduino UNO**, relés y LEDs de 12 V, botones y una pantalla LCD. El objetivo del juego es memorizar y repetir una secuencia de luces y sonidos generada por el sistema. A medida que el jugador acierta, la secuencia se vuelve más larga y desafiante.

El montaje se realiza **sin protoboard**, utilizando **módulos de relé de 12 V** para encender LEDs de mayor potencia, lo que lo convierte en un proyecto ideal para exhibiciones, entornos más robustos o integraciones en cajas personalizadas.

🧰 Materiales Necesarios

🔌 Electrónicos

- 1 placa **Arduino UNO**
- 4 **Relés de 12 V** (módulo de 4 canales optoacoplado)
- 4 **LEDs de 12 V** (con resistencia integrada)
- 4 **Botones pulsadores NO** (uno por LED)
- 1 **Botón START**
- 1 **Botón RESET**
- 1 **Buzzer activo (5 V)**
- 1 **Pantalla LCD 16x4 con módulo I2C**
- 1 **Fuente de alimentación 12 V** (mínimo 1 A)
- Cables dupont macho-hembra / hembra-hembra
- Cables de alimentación

🧳 Opcionales

- Caja de montaje o base de madera/plástico
- Etiquetas para identificar botones y LEDs
- Interruptor general
- Ventilador o disipador (si se utiliza en exteriores o largos periodos)

🔄 Esquema de Conexiones

🧠 Arduino UNO (pines digitales):

Componente	Pin Arduino	Notas
Relé LED 1	D2	IN1 del módulo de relé
Relé LED 2	D3	IN2
Relé LED 3	D4	IN3
Relé LED 4	D5	IN4
Botón 1	D6	INPUT_PULLUP
Botón 2	D7	INPUT_PULLUP
Botón 3	D8	INPUT_PULLUP
Botón 4	D9	INPUT_PULLUP
Botón START	D10	INPUT_PULLUP
Botón RESET	D11	INPUT_PULLUP
Buzzer activo	D12	tone()

⌨️ LCD I2C:

Línea I2C	Pin Arduino
SDA	A4
SCL	A5
VCC/GND	5V / GND

💡 Conexión LED 12 V a relé (por LED):

- Ánodo LED → +12 V
- Cátodo LED → COM del relé
- NO del relé → GND de 12 V

  ⚠️ *Requiere que GND Arduino y GND de la fuente de 12 V estén conectados*

⚖️ Código del Proyecto en Arduino

📦 Librerías usadas:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

⚠️ Advertencias:

- Verifica que tu pantalla LCD sea **16x4** y que la dirección I2C sea correcta (`0x27` o `0x3F`).

- El código utiliza `INPUT_PULLUP`, por lo tanto, los botones deben ir conectados entre **pin digital y GND**.

📜 Código completo:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define NUM_COLORES 4
#define MAX_SECUENCIA 100

const int ledPins[NUM_COLORES] = {2, 3, 4, 5};
const int botonPins[NUM_COLORES] = {6, 7, 8, 9};
const int pinStart = 10;
const int pinReset = 11;
const int buzzerPin = 12;
const int tonos[NUM_COLORES] = {261, 329, 392, 523};

LiquidCrystal_I2C lcd(0x27, 16, 4);

int secuencia[MAX_SECUENCIA];
int nivelActual = 0;
int pasoJugador = 0;
bool enJuego = false;

void setup() {
  for (int i = 0; i < NUM_COLORES; i++) {
    pinMode(ledPins[i], OUTPUT);
    pinMode(botonPins[i], INPUT_PULLUP);
  }
  pinMode(pinStart, INPUT_PULLUP);
  pinMode(pinReset, INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Pulsa START para");
  lcd.setCursor(0, 1);
  lcd.print("comenzar el juego");

  randomSeed(analogRead(A0));
}

void loop() {
  if (digitalRead(pinReset) == LOW) {
    reiniciarJuego();
    delay(300);
  }
```

```
  if (!enJuego && digitalRead(pinStart) == LOW) {
    delay(300);
    iniciarJuego();
  }
  if (enJuego) {
    for (int i = 0; i < NUM_COLORES; i++) {
      if (digitalRead(botonPins[i]) == LOW) {
        reproducirNota(i);
        comprobarRespuesta(i);
        while (digitalRead(botonPins[i]) == LOW);
        delay(200);
      }
    }
  }
}

void iniciarJuego() {
  enJuego = true;
  nivelActual = 1;
  pasoJugador = 0;
  generarSecuencia();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Nivel 1");
  delay(1000);
  reproducirSecuencia();
}

void reiniciarJuego() {
  enJuego = false;
  nivelActual = 0;
  pasoJugador = 0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Pulsa START para");
  lcd.setCursor(0, 1);
  lcd.print("comenzar el juego");
}

void generarSecuencia() {
  for (int i = 0; i < MAX_SECUENCIA; i++) {
    secuencia[i] = random(NUM_COLORES);
  }
}

void reproducirSecuencia() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Mostrando secuencia");
```

```
  for (int i = 0; i < nivelActual; i++) {
    int color = secuencia[i];
    digitalWrite(ledPins[color], HIGH);
    tone(buzzerPin, tonos[color], 300);
    delay(400);
    digitalWrite(ledPins[color], LOW);
    noTone(buzzerPin);
    delay(200);
  }
  pasoJugador = 0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Tu turno");
  lcd.setCursor(0, 1);
  lcd.print("Nivel ");
  lcd.print(nivelActual);
}

void reproducirNota(int color) {
  digitalWrite(ledPins[color], HIGH);
  tone(buzzerPin, tonos[color], 300);
  delay(300);
  digitalWrite(ledPins[color], LOW);
  noTone(buzzerPin);
}

void comprobarRespuesta(int botonPulsado) {
  if (botonPulsado == secuencia[pasoJugador]) {
    pasoJugador++;
    if (pasoJugador == nivelActual) {
      nivelActual++;
      if (nivelActual > MAX_SECUENCIA) nivelActual = MAX_SECUENCIA;
      delay(500);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Nivel ");
      lcd.print(nivelActual);
      delay(1000);
      reproducirSecuencia();
    }
  } else {
    tonoError();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("\xC1Error!");
    lcd.setCursor(0, 1);
    lcd.print("Nivel alcanzado:");
    lcd.setCursor(0, 2);
    lcd.print(nivelActual);
    enJuego = false;
```

```
  }
}

void tonoError() {
  for (int i = 0; i < 3; i++) {
    tone(buzzerPin, 150, 200);
    delay(300);
    noTone(buzzerPin);
  }
}
```

📅 Funcionamiento del Juego (Manual de Uso)

1. 🔌 **Encender el sistema**
   - La pantalla muestra: *“Pulsa START para comenzar”*
2. ▶️ **Inicio del juego**
   - Al pulsar el botón START, se genera una secuencia aleatoria y se muestra el nivel.
3. 💡 **Reproducción de secuencia**
   - Se iluminan los LEDs uno a uno con sonidos, según el nivel actual.
4. 🎮 **Turno del jugador**
   - El jugador debe repetir la secuencia pulsando los botones correctos.
5. ✅ **Avance o** ❌ **error**
   - Si acierta: se sube un nivel.
   - Si falla: suena el buzzer de error y se muestra el nivel alcanzado.
6. 🔄 **Reiniciar juego**
   - Pulsar RESET para volver al inicio en cualquier momento.

🔍 Ajustes y Personalización

Parámetro	Cómo cambiarlo	Efecto
Velocidad de reproducción	Ajustar `delay(400)` en el bucle	Aumenta o reduce dificultad
Número máximo de niveles	Cambiar `MAX_SECUENCIA`	Alarga o acorta el juego
Tonos de cada LED	Editar array `tonos[]`	Cambia la melodía
Mensajes LCD	Editables en `lcd.print()`	Personalización

Parámetro	Cómo cambiarlo	Efecto
		educativa
Combinación de colores	Personaliza cables y LEDs	Mayor atractivo visual

---

🔧 Montaje Físico

- Monta los botones y LEDs en un panel frontal con etiquetas para facilitar su uso.
- Fija los módulos de relé con tornillos o adhesivo en una base firme y aislada.
- Usa canaletas, fundas o bridas para ordenar los cables y evitar enredos o fallos.
- Instala todo dentro de una caja o carcasa transparente o impresa en 3D.
- Asegúrate de que el puerto USB del Arduino quede accesible para programar o actualizar.
- Deja espacio para ventilación si se usará durante largos periodos o en ambientes cerrados.

🚀 Siguientes Pasos

- Añadir niveles de dificultad que reduzcan los tiempos de respuesta con cada nivel superado.
- Incluir LEDs RGB para variar los colores de forma dinámica según el nivel o estado del juego.
- Guardar el récord más alto en la EEPROM para conservarlo tras reiniciar el Arduino.
- Implementar animaciones o efectos especiales en la pantalla LCD.
- Incorporar música de victoria o derrota con una melodía más elaborada.
- Crear una versión con pantalla táctil o pantalla TFT gráfica para hacerla más interactiva.

📌 Consideraciones Finales

- Comparte siempre GND entre Arduino y la fuente de 12 V para evitar problemas de referencia.
- Verifica el estado por defecto de los relés: algunos podrían activarse al iniciar si no se configura correctamente.
- Si el módulo de relé usa un pin JD-VCC separado, asegúrate de puentearlo con VCC o alimentar ambos según tu modelo.

- No dejes los LEDs de 12 V encendidos de forma continua sin supervisión o disipador, ya que podrían sobrecalentarse.
- Antes de cerrar la carcasa final, realiza todas las pruebas de funcionamiento en frío para evitar desmontajes posteriores.

🧳 Autores y Créditos

- 👨‍🏫 **Formador**: Víctor Pardo
- 🏛️ **Entidad organizadora**: EuskalEncounter 33
- 🗓️ **Fecha del taller**: 25/07/2025
- 📍 **Versión del documento**: Julio 2025