**🎲 Dado electrónico con luces y sonido**

¿Te imaginas lanzar un dado sin tener que mover la mano? Con este proyecto podrás hacerlo: usaremos un **Arduino**, varias **luces LED** y un **zumbador** activo para simular el lanzamiento de un dado.

**🔧 ¿Qué materiales necesitamos?**

* Una placa **Arduino Uno** (el cerebro del proyecto).
* **7 luces LED** que serán los puntitos del dado.
* Un **botón** para “lanzar” el dado.
* Un **zumbador** que hará sonidos divertidos.
* Varios cables para conectar todo.

**🪄 ¿Cómo funciona?**

* Cada vez que presionas el **botón**, el dado empieza a “rodar”.
* Los LEDs se encienden rápidamente, mostrando diferentes combinaciones de puntitos, como si el dado estuviera girando en el aire.
* El **zumbador** hace un sonido que va cambiando de velocidad, para que parezca que el dado rueda cada vez más lento.
* Después de unos segundos, el dado se detiene y aparece la **cara final** (un número del 1 al 6).

**🎇 ¿Qué verás?**

* Si sale el número **1**, se encenderá solo un LED en el centro.
* Si sale el **2**, se prenderán dos LEDs en diagonal.
* El **3** tendrá tres puntitos, y así hasta el **6**, donde se encienden seis LEDs como en un dado real.
* El zumbador hará un **“pip” final** para avisar que ya tienes tu número.

**🤩 ¿Por qué es divertido?**

* Porque combina **luces y sonidos** que hacen sentir que el dado está vivo.
* Porque puedes jugar con tus amigos a cualquier juego de mesa sin un dado físico.
* ¡Y porque lo armas tú mismo!

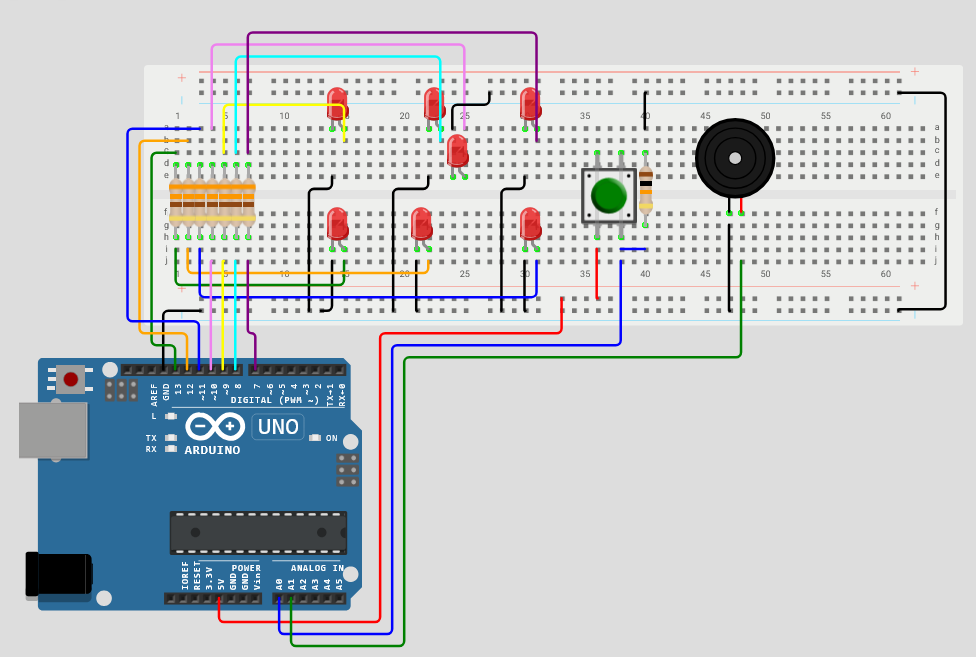
**🧰 Materiales (cosas que vas a necesitar)**

* 1 - Arduino Uno (la plaquita que es el cerebro).
* 7 – LEDs rojos (las lucecitas para los puntitos del dado).
* 1 - botón (para "lanzar" el dado).
* 1 - zumbador activo 5VDC, pequeño (para los sonidos).
* 30 - Cables macho a macho
* 1- Protoboard.
* 7 – resistencias 330 ohmios 1/4watts para los LEDs.
* 1 - resistencia 10Kohmios 1/4watt para el botón.

**⚠️ Antes de comenzar: seguridad**

* Trabaja sobre una mesa limpia.
* Pide ayuda a un adulto para conectar cosas a la placa si no estás seguro.
* No conectes la placa al USB con las manos mojadas.

## ARMADO DEL PROTOBOARD



## 🔩 Paso 1 — Coloca las luces (LEDs)

1. Inserta los 7 LEDs en el protoboard.
2. Recuerda que cada LED tiene una pata larga y otra corta: la pata **larga** es el "lado positivo" y la corta es el "lado negativo". Si una luz no enciende, prueba a darle la vuelta.

## 🔌 Paso 2 — Conecta los cables y resistencias

* Cada LED lleva un cable que llega a una resistencia de 330 ohmios y de cada resistencia va a un pin del Arduino uno.
* El botón tendrá dos cables: uno que va al pin del botón y otro que va a tierra (GND).
* El zumbador tendrá un cable al pin del zumbador y otro a tierra (GND).

Si no sabes qué es GND o cómo poner una resistencia, pide ayuda a un adulto. Lo importante ahora es entender dónde va cada cosa.

## ▶️ Paso 3 — Conecta el Arduino a la computadora

* Usa el cable USB para conectar el Arduino a la computadora.
* Pide a un adulto que te ayude a subir (cargar) el programa si aún no lo hiciste.
* PROGRAMA EN LA PARTE FINAL DEL TEXTO

## 🧪 Paso 4 — Prueba tu dado

1. Presiona el botón.
2. Verás que las luces comienzan a cambiar rápido como si el dado rodara.
3. Escucharás el zumbador que suena rápido y luego se hace más lento.
4. Después de unos segundos el dado se detiene y aparece la cara final (1 a 6).

Si algo no funciona: revisa que los cables estén bien puestos, que los LEDs no estén al revés o que la placa esté conectada al USB.

## 🧪 Paso 5 — Puedes comprobar el funcionamiento en el siguiente enlace que es un simulador de proyectos

* <https://wokwi.com/projects/441194051910993921>

**PROGRAMA PARA CARGAR AL ARDUINO UNO**

// Definición de pines para los LEDs

const int LED\_TOP\_LEFT = 7;

const int LED\_TOP\_CENTER = 8;

const int LED\_TOP\_RIGHT = 9;

const int LED\_CENTER = 10;

const int LED\_BOTTOM\_LEFT = 11;

const int LED\_BOTTOM\_CENTER = 12;

const int LED\_BOTTOM\_RIGHT = 13;

// Pin del botón y buzzer

const int BUTTON\_PIN = A0;

const int BUZZER\_PIN = A1;

// Variables para el control del botón

int lastButtonState = LOW;

unsigned long lastDebounceTime = 0;

unsigned long debounceDelay = 50;

// Variables para la animación

bool rolling = false;

unsigned long rollStartTime = 0;

unsigned long currentDelay = 80; // Velocidad inicial más rápida

int currentFace = 1;

void setup() {

  // Configurar pines de LEDs como salida

  pinMode(LED\_TOP\_LEFT, OUTPUT);

  pinMode(LED\_TOP\_CENTER, OUTPUT);

  pinMode(LED\_TOP\_RIGHT, OUTPUT);

  pinMode(LED\_CENTER, OUTPUT);

  pinMode(LED\_BOTTOM\_LEFT, OUTPUT);

  pinMode(LED\_BOTTOM\_CENTER, OUTPUT);

  pinMode(LED\_BOTTOM\_RIGHT, OUTPUT);

  // Configurar pin del botón como entrada y buzzer como salida

  pinMode(BUTTON\_PIN, INPUT);

  pinMode(BUZZER\_PIN, OUTPUT);

  // Apagar todos los LEDs inicialmente

  clearAllLEDs();

  // Iniciar comunicación serial para debugging (opcional)

**Serial**.begin(9600);

  // Semilla aleatoria basada en una entrada analógica flotante

  randomSeed(analogRead(A2));

}

void loop() {

  int buttonState = digitalRead(BUTTON\_PIN);

  // Detectar flanco de subida del botón con debounce

  if (buttonState == HIGH && lastButtonState == LOW &&

      millis() - lastDebounceTime > debounceDelay) {

    lastDebounceTime = millis();

    if (!rolling) {

      // Iniciar animación de lanzamiento

      rolling = true;

      rollStartTime = millis();

      currentDelay = 80; // Velocidad inicial más rápida

**Serial**.println("Iniciando lanzamiento...");

    }

  }

  lastButtonState = buttonState;

  if (rolling) {

    // Animación de lanzamiento en curso

    unsigned long currentTime = millis();

    unsigned long elapsedTime = currentTime - rollStartTime;

    // Generar sonido con el buzzer (tono que varía con la velocidad)

    int toneFreq = map(constrain(currentDelay, 80, 500), 80, 500, 800, 300);

    tone(BUZZER\_PIN, toneFreq, 50);

    // Mostrar cara aleatoria

    showRandomFace();

    // Reducir gradualmente la velocidad (animación de 3 segundos)

    if (elapsedTime > 2000) {

      // Último segundo: reducir velocidad más rápido

      currentDelay = map(elapsedTime, 2000, 3000, 200, 500);

    } else if (elapsedTime > 1000) {

      // Segundo intermedio: transición de velocidad

      currentDelay = map(elapsedTime, 1000, 2000, 100, 200);

    }

    // Detener la animación después de 3 segundos

    if (elapsedTime > 2500) {

      rolling = false;

      noTone(BUZZER\_PIN); // Detener sonido

      // Sonido de resultado

      tone(BUZZER\_PIN, 600, 200);

      // Mostrar cara final aleatoria

      currentFace = random(1, 7);

      showFace(currentFace);

**Serial**.print("Cara final: ");

**Serial**.println(currentFace);

    }

    delay(currentDelay);

  }

}

void showRandomFace() {

  int randomFace = random(1, 7);

  showFace(randomFace);

}

void showFace(int face) {

  // Apagar todos los LEDs primero

  clearAllLEDs();

  // Encender LEDs según la cara del dado

  switch(face) {

    case 1:

      digitalWrite(LED\_CENTER, HIGH);

      break;

    case 2:

      digitalWrite(LED\_TOP\_LEFT, HIGH);

      digitalWrite(LED\_BOTTOM\_RIGHT, HIGH);

      break;

    case 3:

      digitalWrite(LED\_TOP\_LEFT, HIGH);

      digitalWrite(LED\_CENTER, HIGH);

      digitalWrite(LED\_BOTTOM\_RIGHT, HIGH);

      break;

    case 4:

      digitalWrite(LED\_TOP\_LEFT, HIGH);

      digitalWrite(LED\_TOP\_RIGHT, HIGH);

      digitalWrite(LED\_BOTTOM\_LEFT, HIGH);

      digitalWrite(LED\_BOTTOM\_RIGHT, HIGH);

      break;

    case 5:

      digitalWrite(LED\_TOP\_LEFT, HIGH);

      digitalWrite(LED\_TOP\_RIGHT, HIGH);

      digitalWrite(LED\_CENTER, HIGH);

      digitalWrite(LED\_BOTTOM\_LEFT, HIGH);

      digitalWrite(LED\_BOTTOM\_RIGHT, HIGH);

      break;

    case 6:

      digitalWrite(LED\_TOP\_LEFT, HIGH);

      digitalWrite(LED\_TOP\_CENTER, HIGH);

      digitalWrite(LED\_TOP\_RIGHT, HIGH);

      digitalWrite(LED\_BOTTOM\_LEFT, HIGH);

      digitalWrite(LED\_BOTTOM\_CENTER, HIGH);

      digitalWrite(LED\_BOTTOM\_RIGHT, HIGH);

      break;

  }

}

void clearAllLEDs() {

  digitalWrite(LED\_TOP\_LEFT, LOW);

  digitalWrite(LED\_TOP\_CENTER, LOW);

  digitalWrite(LED\_TOP\_RIGHT, LOW);

  digitalWrite(LED\_CENTER, LOW);

  digitalWrite(LED\_BOTTOM\_LEFT, LOW);

  digitalWrite(LED\_BOTTOM\_CENTER, LOW);

  digitalWrite(LED\_BOTTOM\_RIGHT, LOW);

}