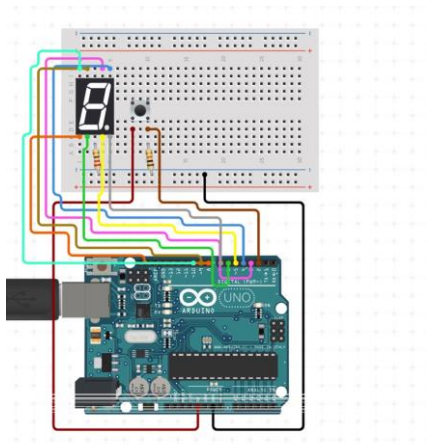


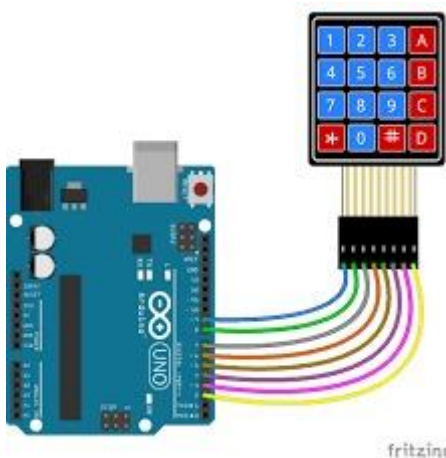
Leo,orlando y brisa



Código 1: Control de un display de 7 segmentos con números aleatorios

Explicación detallada: Este código controla un display de 7 segmentos, un componente común en electrónica que permite mostrar números del 0 al 9. Utiliza un pulsador para alternar entre dos estados: la generación continua de números aleatorios y la pausa para mostrar un único número. Estructura y funcionamiento: Inicialización: El array `numero[]` almacena las combinaciones binarias necesarias para encender los segmentos específicos del display. Por ejemplo, para mostrar el número 1, solo se deben activar los segmentos correspondientes, y el valor binario se establece como `0b0110000`. Configuración (setup): Los pines conectados al display (3 a 10) se configuran como salidas. El pulsador conectado al pin 2 se configura como entrada. Generación de números: En el bucle principal (loop), se usa `random(1, 7)` para generar números aleatorios entre 1 y 6. Si el pulsador está presionado, el estado cambia: el programa alterna entre detener y reanudar la generación de números. Control del display: Dependiendo del número generado, los segmentos correspondientes se activan mediante un bucle que asigna los valores del array `numero[]` a los pines de salida. Usos prácticos: Simulación de un dado electrónico. Aplicaciones educativas para enseñar conceptos básicos de electrónica digital y control de displays.

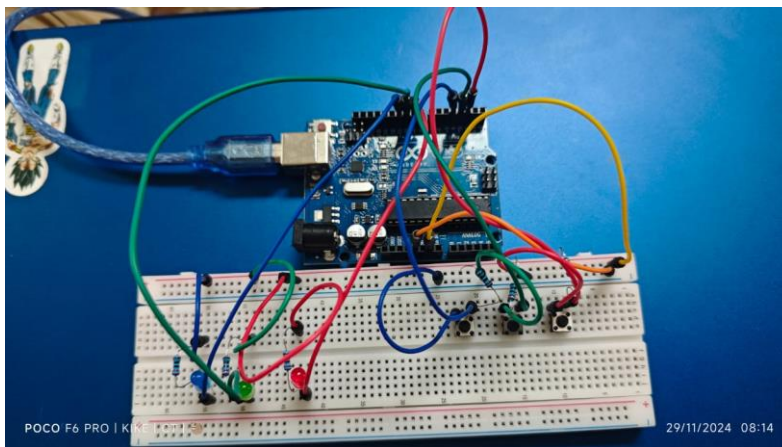
Dani,miguel y omar



Código 3: Teclado matricial para ingresar contraseñas

Explicación detallada: Este programa permite ingresar una contraseña usando un teclado matricial. La contraseña ingresada se compara con una clave maestra predefinida. Si es correcta, se enciende un LED verde; si no, se enciende un LED rojo. Estructura y funcionamiento: Inicialización: Se define el mapeo de teclas en el array `keys`. Los pines para las filas y columnas del teclado están especificados en `pinFilas` y `pinColumnas`. Configuración (`setup`): Se inicializan los LEDs como salidas. Se establece la comunicación con el monitor serial para depuración. Lógica principal (`loop`): Cada vez que se presiona una tecla, se almacena en el array `CLAVE`. Cuando se ingresan 6 caracteres, el programa compara la clave ingresada con la clave maestra usando `strcmp()`. Acciones según el resultado: Si la clave es correcta: Se enciende el LED verde por 2 segundos. Si es incorrecta: Se enciende el LED rojo por 2 segundos. Usos prácticos: Sistemas básicos de control de acceso. Simulación de dispositivos de seguridad.

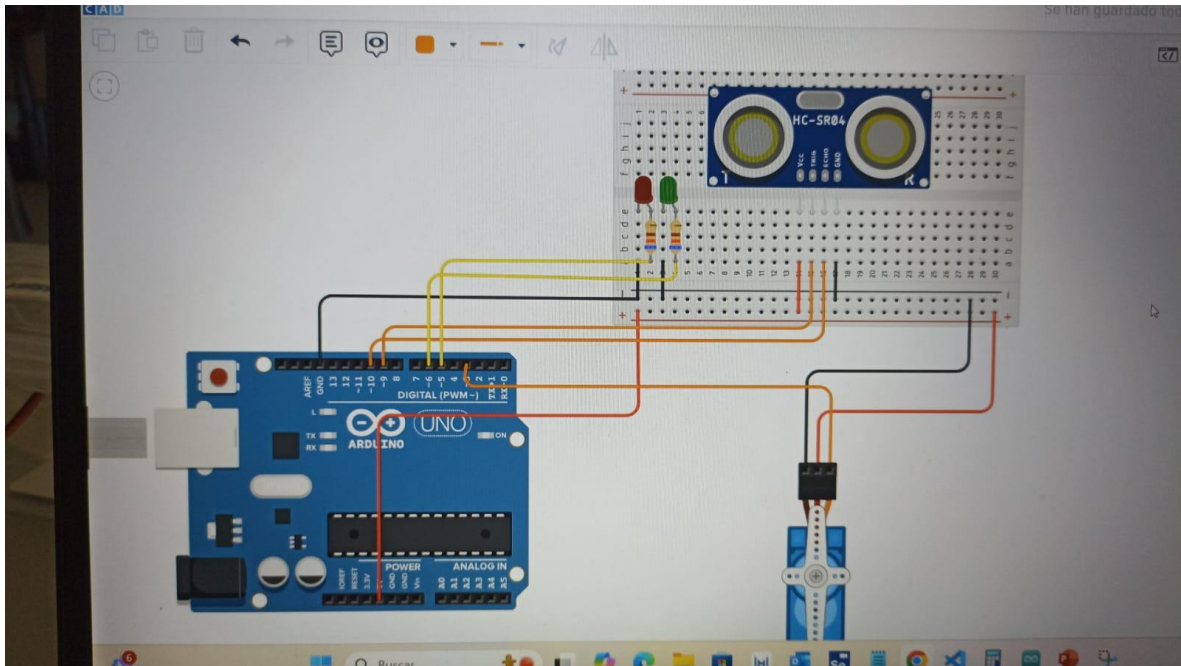
Alejandro ,kike,y alex



Código 4: Control de un LED RGB con botones

Explicación detallada: Este código controla un LED RGB utilizando tres botones. Cada botón activa un color primario: rojo, verde o azul. Estructura y funcionamiento: Inicialización: Los pines del LED RGB están conectados a salidas PWM. Los botones están conectados a entradas digitales. Lógica principal (`loop`): Se verifica el estado de cada botón. Según el botón presionado, la función `encenderColor()` establece los valores de los canales rojo, verde y azul del LED RGB. Usos prácticos: Control básico de iluminación decorativa. Aprendizaje de LEDs RGB y mezcla de colores.

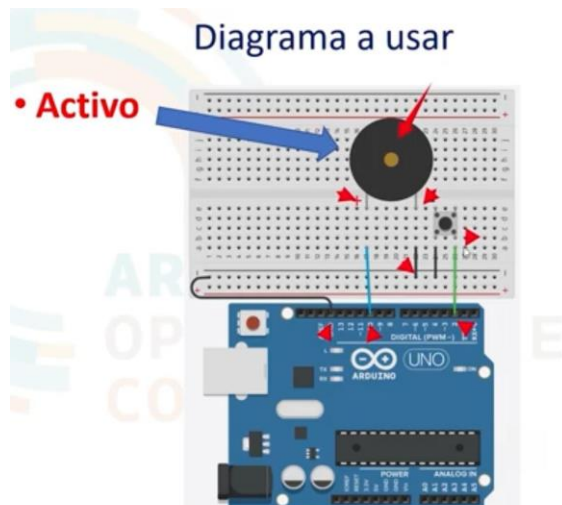
Iris,liz y dalia



Código 5: Sistema con sensor ultrasónico y servomotor

Explicación detallada: Este código utiliza un sensor ultrasónico para detectar la proximidad de un objeto y controla un servomotor que simula abrir y cerrar una puerta. Estructura y funcionamiento: Sensor ultrasónico: El sensor mide la distancia al objeto calculando el tiempo que tarda un pulso de sonido en regresar. Si la distancia medida es menor a 5 cm, se considera que un objeto está cerca. Control del servomotor: Si un objeto está cerca: Se enciende un LED verde y el servomotor mueve la puerta para abrirla. Después de 3 segundos, el servomotor cierra la puerta y se enciende un LED rojo. Usos prácticos: Simulación de puertas automáticas. Aplicaciones en sistemas de acceso sin contacto.

Fer,yahir bryan



BUZZER

Un buzzer es un pequeño dispositivo que emite sonidos o tonos cuando se le suministra corriente eléctrica. En un proyecto con Arduino, es ideal para generar alertas sonoras, tonos musicales, o cualquier tipo de señal auditiva.

Resumen para usar un buzzer en Arduino:

Tipos de buzzer:

Pasivo: Necesita señales PWM para producir diferentes tonos (controlados mediante la función `tone()`).

Activo: Emite un sonido fijo cuando recibe corriente (más fácil de usar).

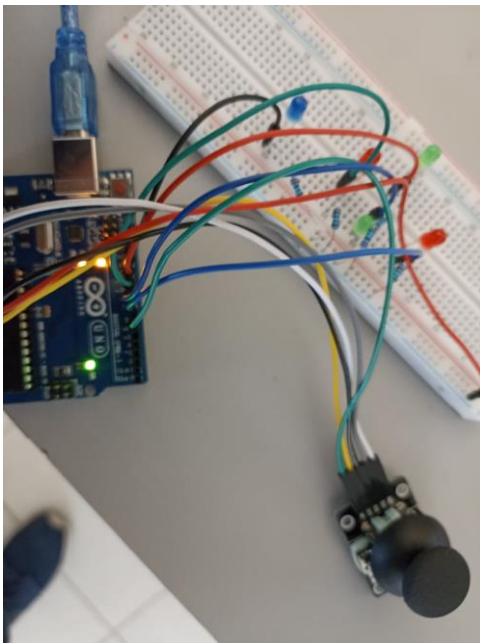
Conexión:

Conecta el pin positivo del buzzer (marcado con un "+" o el lado más largo) a un pin digital de Arduino (por ejemplo, D9).

Conecta el pin negativo al GND de Arduino.

(Opcional) Usa una resistencia para limitar la corriente si es necesario.

EMMA Y JOSUE



Un **joystick** se puede usar en Arduino para controlar la intensidad o estado de varios LEDs según la posición en sus ejes **X** e **Y**. Los joysticks analógicos tienen dos potenciómetros internos que proporcionan valores de salida para los ejes, y a menudo incluyen un botón integrado (usualmente activado al presionarlo).

Conexión básica del joystick y LEDs:

Joystick:

Eje X (VRx): Conectar a un pin analógico (por ejemplo, A0).

Eje Y (VRy): Conectar a otro pin analógico (por ejemplo, A1).

GND: Conectar al GND de Arduino.

VCC: Conectar a 5V de Arduino.

Botón (SW): Conectar a un pin digital (opcional).

LEDs:

Conecta un LED con resistencia (220 ohm) a cada pin digital que desees controlar (por ejemplo, D2, D3, D4, D5).

GND de los LEDs al GND de Arduino.