

# Investigación: comandos y funcionalidades. C++

## Ejercicio 1: Encender un LED

Comandos utilizados:

- `pinMode(ledPin, OUTPUT)`: Configura el pin especificado (`ledPin`) como salida.
- `digitalWrite(ledPin, HIGH)`: Enciende el dispositivo conectado al pin especificado (`ledPin`).

## Ejercicio 2: Parpadeo de un LED

Comandos utilizados:

- `pinMode(ledPin, OUTPUT)`: Configura el pin especificado (`ledPin`) como salida.
- `digitalWrite(ledPin, HIGH)`: Enciende el dispositivo conectado al pin especificado (`ledPin`).
- `digitalWrite(ledPin, LOW)`: Apaga el dispositivo conectado al pin especificado (`ledPin`).
- `delay(blinkInterval / 2)`: Introduce un retardo en milisegundos (la mitad del intervalo de parpadeo).

## Ejercicio 3: Secuencia de LEDs

Comandos utilizados:

- `pinMode(ledPins[i], OUTPUT)`: Configura el pin especificado (`ledPins[i]`) como salida.
- `digitalWrite(ledPins[i], HIGH)`: Enciende el dispositivo conectado al pin especificado (`ledPins[i]`).
- `digitalWrite(ledPins[i], LOW)`: Apaga el dispositivo conectado al pin especificado (`ledPins[i]`).
- `delay(delayTime)`: Introduce un retardo en milisegundos (tiempo de retardo).

## **Ejercicio 4: Control de LED con botón**

Comandos utilizados:

- `pinMode(ledPin, OUTPUT)`: Configura el pin especificado (`ledPin`) como salida.
- `pinMode(buttonPin, INPUT)`: Configura el pin especificado (`buttonPin`) como entrada.
- `digitalRead(buttonPin)`: Lee el estado del pin especificado (`buttonPin`) y devuelve un valor HIGH (presionado) o LOW (no presionado).
- `digitalWrite(ledPin, HIGH)`: Enciende el dispositivo conectado al pin especificado (`ledPin`).
- `digitalWrite(ledPin, LOW)`: Apaga el dispositivo conectado al pin especificado (`ledPin`).

### **Explicación adicional:**

- Los comandos `pinMode` y `digitalWrite` son funciones básicas de Arduino para configurar y controlar pines de entrada/salida.
- La función `delay` permite introducir un retardo en la ejecución del programa, lo que es fundamental para crear efectos de parpadeo o secuencias temporizadas.
- La lectura del estado del botón con `digitalRead` permite controlar el comportamiento del LED en función de la interacción del usuario.