

Contenido

[Introducción 3](#_heading=h.fprugsrw75en)

[**Antecedentes 4**](#_heading=h.4rq6gb1dm3lf)

[● Microcontroladores 4](#_heading=h.ai188sf36fro)

[● Instrucciones de microcontroladores 7](#_heading=h.bzijdfv8jg3x)

[● Programación de microcontroladores 10](#_heading=h.posrb139jve8)

[5. Características y tipos de interfaces 11](#_heading=h.q51p6xz6z3q0)

[● Periféricos de entrada y salida en microcontroladores. 12](#_heading=h.k82tbh4yi2nt)

[● Sistemas Embebidos 16](#_heading=h.tiw94fa6iv9i)

[Conceptos Teóricos 20](#_heading=h.tfm5lopnyii7)

[● Motivación y Justificación 20](#_heading=h.toulju1zpqp)

[● Lista de Materiales 20](#_heading=h.kiy7s7n7l0w3)

[● Sensor de sensor de gas MQ6 21](#_heading=h.8ug18sgv789c)

[● Pantalla LCD 21](#_heading=h.57xk1e5c7dve)

[● Servomotor 22](#_heading=h.ixw7i5cbp475)

[● Protoboard 23](#_heading=h.djznipw8t5ek)

[● Arduino Uno 23](#_heading=h.7vjqga6tzn7u)

[● Leds 24](#_heading=h.pvibdmisq5sj)

[● Descripcion de la Placa Arduino y Código Fuente 25](#_heading=h.pcvwi9po9wua)

[● Código de Arduino 25](#_heading=h.u1zjve2fpoi8)

[● Interfaz Usuario-Arduino con App Inventor 27](#_heading=h.tgnlspaef6qo)

[Desarrollo 29](#_heading=h.j95d9vemhq67)

[● Delimitación de Alcances 29](#_heading=h.fwfdx0od6h7l)

[● Imagen de armado de circuito 30](#_heading=h.1es7t6f21nzb)

[**Conclusión 30**](#_heading=h.z73bz97e57mf)

[● Aporte Individual 30](#_heading=h.xkj9zcftw1h5)

[● Conclusion Individual 31](#_heading=h.dbr9jox7jdvl)

[● Conclusión General 32](#_heading=h.nf73y799bjzx)

[Fuentes de informacion 33](#_heading=h.r9yos4uy585q)

# Introducción

En la actualidad, la seguridad en el hogar es una prioridad, especialmente en situaciones donde fugas de gas pueden representar un riesgo grave para la salud y la integridad de las personas. Los accidentes causados por fugas de gas, como incendios o intoxicaciones, son evitables si se cuenta con sistemas de detección temprana. Este proyecto tiene como objetivo desarrollar un prototipo funcional que monitoree la presencia de gas licuado de petróleo (GLP) en un ambiente doméstico (maquetado) utilizando un sensor MQ6, un microcontrolador Arduino Uno y una aplicación para visualizar los datos en tiempo real.

El sistema propuesto integra componentes electrónicos como el sensor MQ6, capaz de detectar concentraciones peligrosas de gas, y actuadores como un servomotor (que simula la apertura de una ventana) y un LED (que actúa como alarma visual). La información recopilada por el sensor se procesa en el Arduino y se muestra en una pantalla LCD, mientras que una aplicación permite al usuario final visualizar los datos de manera remota, brindando alertas inmediatas en caso de una fuga. Además, el sistema está diseñado para ser escalable, permitiendo la integración de otros sensores o actuadores en futuras iteraciones, lo que amplía su potencial en aplicaciones de domótica y seguridad avanzada.

Este proyecto no solo demuestra la aplicación práctica de conceptos de arquitectura de computadoras, microcontroladores y sistemas embebidos, sino que también resalta la importancia de la tecnología en la prevención de accidentes domésticos. La combinación de hardware y software en este desarrollo refleja cómo la ingeniería puede ofrecer soluciones innovadoras a problemas cotidianos. A través de la implementación de este sistema, se busca contribuir a la seguridad en el hogar, combinando electrónica, programación y desarrollo de software para crear una solución accesible y eficiente.

El trabajo se estructura en tres etapas principales: la investigación y selección de componentes, el desarrollo del circuito y la programación del microcontrolador, y finalmente la integración con la aplicación para visualización de datos. Cada fase incluye evidencias fotográficas, diagramas y un reporte detallado que documenta el proceso y los resultados obtenidos. Adicionalmente, se abordarán los desafíos técnicos enfrentados durante la implementación, como la calibración del sensor y la optimización de la comunicación entre el Arduino y la aplicación, lo que enriquece el aprendizaje y la experiencia adquirida en el ámbito de los sistemas embebidos.

# Antecedentes

## **Microcontroladores**

#### **1. Definición de microcontrolador**

Un microcontrolador es un circuito integrado programable que contiene todos los componentes esenciales de una computadora en un solo chip. A diferencia de los microprocesadores que requieren componentes externos, los microcontroladores integran:

* Unidad Central de Procesamiento (CPU)
* Memoria (ROM/Flash para programa, RAM para datos)
* Periféricos de entrada/salida
* Interfaces de comunicación

Son diseñados para aplicaciones específicas de control y automatización, con capacidades de tiempo real y bajo consumo energético【Microchip Technology, s.f.】【Valdés & Pérez, 2018】.

#### **2. Tipos de microcontroladores**

**Por arquitectura:**

* **Arquitectura Harvard (ej: PIC de Microchip)**: Memorias separadas para instrucciones y datos, lo que permite una mayor velocidad de ejecución【Microchip Technology, s.f.】.
* **Arquitectura Von Neumann (ej: 8051)**: Memoria unificada para instrucciones y datos. Más simple pero potencialmente más lenta【Atmel Corporation, 2016】.
* **RISC (Computadora de Conjunto Reducido de Instrucciones, ej: ARM)**: Instrucciones simples y de ciclo único, lo que mejora la eficiencia energética【ARM Holdings, s.f.】.
* **CISC (Computadora de Conjunto Complejo de Instrucciones, ej: 8051)**: Instrucciones complejas y multi-ciclo, lo que permite menor código pero más complejidad【Atmel Corporation, 2016】.

**Por familias populares:**

* **PIC (Microchip Technology)**: Amplia gama de modelos, arquitectura Harvard, popular en aplicaciones industriales【Microchip Technology, s.f.】.
* **AVR (Atmel, ahora Microchip)**: Arquitectura RISC, usado en plataformas Arduino, y con un buen balance rendimiento/consumo【Atmel Corporation, 2016】.
* **ARM Cortex (Múltiples fabricantes)**: Arquitectura RISC avanzada, desde muy bajos a altos recursos, dominante en aplicaciones modernas【ARM Holdings, s.f.】.
* **ESP32/ESP8266 (Espressif)**: Incluyen WiFi/BT integrado, muy populares en IoT【Espressif Systems, 2023】.
* **8051 (Intel y otros)**: Arquitectura clásica, ampliamente usada en educación【Atmel Corporation, 2016】.

#### **3. Componentes y características de microcontroladores**

**Componentes principales:**

* **CPU**: Ejecuta instrucciones del programa.
* **Memoria**:  
  + **Flash/ROM**: Almacena el programa.
  + **RAM**: Almacena datos temporales.
  + **EEPROM**: Almacenamiento persistente.
* **Periféricos**:  
  + **GPIO**: Pines de propósito general.
  + **ADC**: Convierte señales analógicas.
  + **PWM**: Control de potencia.
  + **Temporizadores/Contadores**.
* **Interfaces de comunicación**:  
  + **UART**: Comunicación serial asíncrona.
  + **SPI**: Interfaz serial de alta velocidad.
  + **I2C**: Comunicación entre circuitos integrados.
* **Sistema de interrupciones**: Para manejo de eventos.

**Características clave:**

* Bajo consumo: Ideal para dispositivos portátiles.
* Tiempo real: Respuesta predecible a eventos.
* Integración: Todos los componentes en un chip.
* Flexibilidad: Programable para diversas aplicaciones.
* Robustez: Diseñado para entornos industriales.
* Costo efectivo: Económico en producción masiva【Microchip Technology, s.f.】【Valdés & Pérez, 2018】.

#### **4. Proceso de selección de microcontroladores**

**Definir requerimientos del proyecto**:

* Tipo de procesamiento necesario.
* Velocidad de operación requerida.
* Cantidad y tipo de entradas/salidas.

**Evaluar necesidades de memoria**:

* Tamaño del programa (Flash).
* Datos temporales (RAM).
* Almacenamiento persistente (EEPROM).

**Identificar periféricos necesarios**:

* ADC (para sensores analógicos).
* PWM (para control de motores).
* Interfaces de comunicación (UART, I2C, SPI).

**Considerar restricciones de energía**:

* Consumo en operación normal.
* Modos de bajo consumo disponibles.
* Fuente de alimentación disponible.

**Evaluar entorno de desarrollo**:

* Disponibilidad de compiladores.
* Herramientas de depuración.
* Soporte de la comunidad.

**Analizar costos**:

* Costo del microcontrolador.
* Costo de componentes adicionales.
* Costo de herramientas de desarrollo.

**Verificar disponibilidad**:

* Tiempos de entrega.
* Opciones de proveedores.
* Perspectiva a largo plazo【Valdés & Pérez, 2018】.

## **Instrucciones de microcontroladores**

#### **1. Sintaxis de instrucciones**

La sintaxis básica sigue generalmente este formato:

ETIQUETA: MNEMÓNICO OPERANDO1, OPERANDO2 ; COMENTARIO

Ejemplo en AVR:

INICIO: LDI R16, 0x25 ; Carga el valor hexadecimal 25 en el registro R16

ADD R1, R2 ; Suma los contenidos de R1 y R2, resultado en R1

**Elementos clave:**

* **Etiqueta**: Opcional, marca una posición en el código.
* **Mnemónico**: Nombre de la instrucción (ej: MOV, ADD).
* **Operandos**: Datos sobre los que opera la instrucción.
* **Comentario**: Explicación del código (después de ;)【Kernighan & Ritchie, 1988】.

#### **2. Estructura de las instrucciones**

Las instrucciones se pueden clasificar por su función:

* **Instrucciones de transferencia de datos**:  
  + MOV: Mover datos entre registros.
  + LDI: Cargar inmediato en registro.
  + LD/ST: Cargar/almacenar desde/hacia memoria.
* **Instrucciones aritméticas**:  
  + ADD/ADC: Suma (con/sin acarreo).
  + SUB/SBC: Resta (con/sin acarreo).
  + MUL: Multiplicación.
  + INC/DEC: Incremento/Decremento.
* **Instrucciones lógicas**:  
  + AND/OR/XOR: Operaciones lógicas.
  + COM: Complemento.
  + NEG: Negación.
* **Instrucciones de control de flujo**:  
  + RJMP: Salto relativo.
  + CALL/RET: Llamada/retorno de subrutina.
  + BRxx: Bifurcaciones condicionales.
* **Instrucciones de manejo de bits**:  
  + SETB/CLRB: Poner a 1/0 un bit.
  + SBIC/SBIS: Saltar si bit está claro/establecido【Atmel Corporation, 2016】.

#### **3. Características y tipos de instrucciones**

**Características principales:**

* Longitud: Pueden ser de 8, 16 o 32 bits.
* Ciclos de ejecución: Desde 1 hasta varios ciclos de reloj.
* Modos de direccionamiento:  
  + Inmediato: El operando es un valor constante.
  + Directo: Acceso directo a memoria.
  + Indirecto: Acceso mediante puntero.
  + Relativo: Para saltos.

**Tipos según complejidad:**

* Instrucciones de un solo ciclo: Ejecución rápida.
* Instrucciones multi-ciclo: Para operaciones complejas.

**Instrucciones de control especiales**:

* SLEEP: Modo de bajo consumo.
* WDT: Control del watchdog timer.
* SEI/CLI: Habilitar/deshabilitar interrupciones【Atmel Corporation, 2016】.

## **Programación de microcontroladores**

#### **1. Interfaces de microcontroladores**

Las interfaces son los medios por los cuales un microcontrolador (MCU) se comunica con otros dispositivos, como sensores, memorias, pantallas, módulos de comunicación, etc. Pueden ser digitales, analógicas o mixtas, y se implementan mediante protocolos de comunicación o conexiones físicas【Philips Semiconductors, 2003】【Fundación Raspberry Pi, 2023】.

#### **2. Conceptos de señal analógica y digital**

* **Señal analógica**: Representación continua en el tiempo. Puede tomar infinitos valores en un rango. Ejemplos: Temperatura, sonido, luminosidad. Requiere ADC para ser procesada por microcontroladores.
* **Señal digital**: Representación discreta en el tiempo. Solo toma valores definidos (generalmente 0 y 1). Ejemplos: Interruptores, LEDs, comunicación serial. Nativa para procesamiento en microcontroladores【Valdés & Pérez, 2018】.

3. **Características de sistemas digitales y analógicos**

| **Característica** | **Sistemas Digitales** | **Sistemas Analógicos** |
| --- | --- | --- |
| **Representación** | Discreta (0/1) | Continua |
| **Inmunidad al ruido** | Alta | Baja |
| **Precisión** | Limitada por resolución | Limitada por componentes |
| **Procesamiento** | Flexible (programable) | Específico (circuitos fijos) |
| **Reproducibilidad** | Exacta | Con variaciones |
| **Complejidad diseño** | En software | En hardware |
| **Consumo energético** | Depende de frecuencia | Generalmente constante |

#### **4. Concepto de interfaz de microcontroladores**

Una interfaz es el conjunto de hardware y software que permite la comunicación entre el microcontrolador y dispositivos externos. Sus funciones principales son:

* **Adaptación de niveles:** Ajustar voltajes entre dispositivos**.**
* **Conversión de señales**: Analógico-digital o viceversa.
* **Aislamiento:** Protección eléctrica.
* **Protocolo de comunicación:** Establecer reglas de intercambio de datos【González, 2020】.

### **5. Características y tipos de interfaces**

**Características principales:**

* Velocidad de transmisión
* Distancia máxima
* Número de dispositivos conectables
* Sincronización (síncrona/asíncrona)
* Topología (punto a punto, bus, red)

**Tipos principales:**

* **GPIO (General Purpose Input/Output)**: Pines configurables como entrada o salida. Niveles lógicos (0/1). Uso más básico y versátil【Valdés & Pérez, 2018】.
* **UART (Universal Asynchronous Receiver-Transmitter)**: Comunicación serial asíncrona, con dos hilos (TX y RX) y velocidades estándar (9600, 115200 baudios)【Philips Semiconductors, 2003】.
* **SPI (Serial Peripheral Interface)**: Comunicación serial síncrona, full-duplex (transmisión simultánea), con 4 líneas (SCK, MOSI, MISO, SS) y alta velocidad【Philips Semiconductors, 2003】.
* **I2C (Inter-Integrated Circuit)**: Comunicación serial síncrona con dos hilos (SDA y SCL), direccionamiento de múltiples dispositivos y velocidad moderada【Philips Semiconductors, 2003】.
* **USB (Universal Serial Bus)**: Comunicación serial de alta velocidad, con un protocolo complejo y requiere un controlador dedicado【Philips Semiconductors, 2003】.
* **Ethernet**: Conexión de red que requiere un stack TCP/IP para aplicaciones de red【Valdés & Pérez, 2018】.
* **PWM (Pulse Width Modulation)**: No es un protocolo de comunicación, sino que se usa para el control de potencia mediante pulsos, y es utilizado en motores, LEDs, etc【Valdés & Pérez, 2018】.
* **ADC/DAC**: Conversión analógico-digital y viceversa, usada para interfaz con sensores analógicos【Texas Instruments, 2021】.

Cada tipo de interfaz tiene sus ventajas y se selecciona según los requerimientos específicos del proyecto en cuanto a velocidad, distancia, número de dispositivos y complejidad de implementación【Valdés & Pérez, 2018】

## **Periféricos de entrada y salida en microcontroladores.**

#### **1. Definición de periférico de entrada/salida**

Los periféricos de entrada y salida (E/S) son dispositivos que permiten al microcontrolador interactuar con el entorno externo. Se clasifican en:

* **Periféricos de entrada:** Captan información del mundo físico y la convierten en señales eléctricas que el microcontrolador puede procesar (ej: sensores, teclados)【Valdés & Pérez, 2018】.
* **Periféricos de salida:** Reciben señales del microcontrolador y las transforman en acciones físicas (ej: motores, displays)【Valdés & Pérez, 2018】.

Estos periféricos pueden ser internos (integrados en el chip del microcontrolador) o externos (conectados mediante interfaces)【Valdés & Pérez, 2018】.

#### **2. Características y tipos de periféricos**

**Características principales:**

* **Niveles de voltaje:** Compatibles con el microcontrolador (3.3V, 5V)**.**
* **Consumo de energía**: Adecuado a la capacidad del sistema.
* **Velocidad de operación**: Coincidente con los requerimientos.
* **Interfaz de conexión**: Digital, analógica o protocolo específico.
* **Robustez:** Resistencia a condiciones ambientales【Valdés & Pérez, 2018】【González, 2020】.

**Tipos de periféricos de entrada:**

* **Sensores:**
  + Temperatura (DS18B20, LM35)
  + Humedad (DHT11, DHT22)
  + Proximidad (HC-SR04 ultrasonico)
  + Acelerómetros (MPU6050)
  + Pulsómetros (señales ECG)
* **Dispositivos de interfaz humana:**
  + Teclados matriciales
  + Botones pulsadores
  + Pantallas táctiles
  + Micrófonos
* **Interfaces de comunicación:**
  + Lectores RFID
  + Módulos Bluetooth/WiFi
  + Receptores infrarrojos【Valdés & Pérez, 2018】.

**Tipos de periféricos de salida:**

* **Actuadores:**
  + Motores (DC, paso a paso, servomotores)
  + Relés (control de cargas eléctricas)
  + Solenoides
  + Bombas
* **Indicadores:**
  + LEDs y displays (7 segmentos, LCD, OLED)
  + Buzzer/altavoces
  + Pantallas gráficas
* **Interfaces de comunicación:**
  + Módulos de transmisión inalámbrica
  + Interfaces de red【González, 2020】【Valdés & Pérez, 2018】.

#### **3. Proceso de conexión de interfaces E/S**

Identificación de requerimientos:

* Determinar tipo de señal (analógica/digital).
* Verificar niveles de voltaje.
* Calcular consumo de corriente【Valdés & Pérez, 2018】**.**

Preparación de circuitos de interfaz:

* Divisores de voltaje para adaptación de niveles.
* Circuitos de polarización para sensores.
* Drivers para actuadores (transistores, H-bridge).
* Protecciones (diodos, optoacopladores)【Valdés & Pérez, 2018】.

Conexión física:

* Asignación de pines del microcontrolador.
* Conexión de tierra común.
* Cableado siguiendo buenas prácticas EMI【González, 2020】.

Configuración del microcontrolador:

* Establecer dirección de pines (entrada/salida).
* Configurar periféricos internos (ADC, PWM).
* Habilitar resistencias pull-up/pull-down si es necesario【Valdés & Pérez, 2018】**.**

Pruebas iniciales:

* Verificar niveles de señal.
* Comprobar respuesta básica.
* Medir consumo energético【Valdés & Pérez, 2018】.

**4. Proceso de programación de microcontroladores**

Configuración inicial:

* Establecer frecuencia de reloj.
* Inicializar periféricos internos.
* Configurar interrupciones【Valdés & Pérez, 2018】.

**R**utinas de manejo de E/S:  
 Ejemplo en C para AVR:

DDRB |= (1 << PB0); // Configura PB0 como salida

PORTB |= (1 << PB0); // Enciende PB0

**Lectura de entradas:**

if (PIND & (1 << PD2)) { // Lee PD2

// Acción si entrada es alta

}

**Manejo de periféricos avanzados**:

Configuración de ADC:

ADMUX = (1 << REFS0); // Voltaje referencia AVcc

ADCSRA = (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1); // Habilitar ADC, prescaler 64

**Implementación de protocolos**:

* Comunicación I2C, SPI, UART.
* Protocolos específicos de sensores【Valdés & Pérez, 2018】.

**Optimización**:

* Reducción de consumo energético.
* Mejora de tiempos de respuesta.
* Manejo eficiente de memoria【Valdés & Pérez, 2018】.

## **Sistemas Embebidos**

#### **1. Definiciones fundamentales**

Un **sistema embebido** es un sistema computacional dedicado diseñado para realizar funciones específicas dentro de un sistema mayor. Sus características incluyen:

* Propósito específico
* Restricciones de recursos
* Operación en tiempo real
* Alta confiabilidad【Valdés & Pérez, 2018】【González, 2020】.

Un **sensor** es un dispositivo que detecta cambios en el entorno físico y los convierte en señales eléctricas. Ejemplos incluyen sensores de temperatura, movimiento y humedad【González, 2020】.

Un **actuador** convierte señales eléctricas en acción física, como motores, relés y válvulas solenoides【Valdés & Pérez, 2018】.

#### **2. Características y tipos**

**Sistemas embebidos**:

* Características:  
  + Restricciones de tamaño y peso
  + Bajo consumo energético
  + Operación continua
  + Tolerancia a fallos
* Tipos:  
  + **Por complejidad**: Sistemas pequeños (8-bit microcontroladores), sistemas medianos (32-bit microcontroladores), sistemas complejos (SoC con Linux embebido).
  + **Por aplicación**: Sistemas de control industrial, dispositivos médicos, automoción, electrónica de consumo【Valdés & Pérez, 2018】.

**Sensores**:

* Características:  
  + Rango de medición
  + Precisión
  + Tiempo de respuesta
  + Consumo energético
* Tipos:  
  + **Físicos**: Temperatura, presión, humedad.
  + **Mecánicos**: Acelerómetros, giroscopios, sensores de fuerza.
  + **Ópticos**: Fotodiodos, sensores infrarrojos, cámaras【González, 2020】.

**Actuadores**:

* Características:  
  + Fuerza/par motor
  + Velocidad de respuesta
  + Precisión de posicionamiento
  + Consumo energético
* Tipos:  
  + **Motores**: DC, paso a paso, servomotores.
  + **Electromecánicos**: Relés, solenoides.
  + **Neumáticos/hidráulicos**: Válvulas, cilindros【Valdés & Pérez, 2018】【González, 2020】.

#### **3. Plataformas de sistemas embebidos**

**Plataformas de bajo costo**:

* Arduino (AVR)
* ESP8266/ESP32
* Raspberry Pi Pico (RP2040)

**Plataformas de alto rendimiento**:

* Raspberry Pi
* BeagleBone
* NVIDIA Jetson

**Plataformas industriales**:

* STM32
* PIC32
* Texas Instruments MSP430

**Plataformas especializadas**:

* Adafruit Feather
* Teensy
* Micro:bit【Fundación Raspberry Pi, 2023】【Valdés & Pérez, 2018】.

#### **4. Aplicaciones de microcontroladores**

#### **Robótica**

* **Aplicaciones**:  
  + Control de motores y servos
  + Procesamiento de datos de sensores
  + Toma de decisiones autónomas
  + Comunicación con otros sistemas
* **Integración**:  
  + Ejemplo: control de servo con PWM【Valdés & Pérez, 2018】.

#### **Domótica**

* **Aplicaciones**:  
  + Control de iluminación automático
  + Gestión de climatización
  + Seguridad del hogar
  + Automatización de electrodomésticos
* **Integración**:  
  + Ejemplo: control de calefacción basado en temperatura【Valdés & Pérez, 2018】.

#### **Internet de las Cosas (IoT)**

* **Aplicaciones**:  
  + Monitoreo remoto de variables
  + Control a distancia de dispositivos
  + Análisis de datos en la nube
  + Sistemas predictivos
* **Integración**:  
  + Ejemplo: envío de datos a la nube con ESP32【Espressif Systems, 2023】.

#### **Aplicaciones y relación con tu proyecto**

* **Domótica**: Seguridad automática, respuesta ante emergencias, monitoreo remoto.
* **IoT**: Dispositivo conectado a red, notificaciones remotas, posible integración con otros sistemas.
* **Robótica**: Uso de actuadores controlados, sistema autónomo que toma decisiones, integración en tu proyecto (sensor de entrada crítica, microcontrolador evalúa riesgo, actuadores mitigan peligro, comunicación con usuarios remotos, automatización sin intervención humana)【Valdés & Pérez, 2018】【Espressif Systems, 2023】.

# Conceptos Teóricos

## **Motivación y Justificación**

La seguridad en el hogar es prioritaria ante los riesgos de fugas de gas que pueden causar incendios o intoxicaciones. Este proyecto surge para ofrecer una solución accesible mediante tecnología, combinando un sensor MQ6 con Arduino para detectar concentraciones peligrosas y activar alarmas visuales y mecánicas. La integración con una aplicación móvil permite monitoreo remoto, brindando protección continua incluso cuando los residentes no están presentes.

Más allá de su aplicación práctica, el desarrollo demuestra cómo conceptos de arquitectura de computadoras y sistemas embebidos pueden resolver problemas reales. Su diseño modular permite adaptarse a diferentes entornos y escalarse con otros sensores, destacando el potencial de la tecnología para mejorar la calidad de vida. La propuesta no solo alerta sobre peligros, sino que actúa automáticamente para mitigarlos, ofreciendo un enfoque innovador frente a soluciones comerciales existentes.

## **Lista de Materiales**

* Sensor de gas MQ6
* Pantalla LCD
* Protoboard
* Servomotor o motor a pasos
* Arduino Uno
* Cables
* Led

## **Sensor de sensor de gas MQ6**

El MQ-6 Detector de Gas LP es un sensor electrónico diseñado para detectar la presencia de gases inflamables como gas licuado de petróleo (GLP), butano, propano y otros hidrocarburos en el aire. Funciona mediante un componente sensible de óxido metálico que varía su resistencia según la concentración del gas, generando una señal analógica que puede ser interpretada por un microcontrolador

Sus principales usos y aplicaciones se encuentran en **sistemas de detección de fugas de gas domésticos e industriales**, **alarmas de seguridad**, **monitoreo ambiental**, y **proyectos de automatización del hogar**, donde se requiere una respuesta rápida ante concentraciones peligrosas de gas. El sensor ofrece una salida analógica proporcional a la concentración del gas, y en muchos módulos también una salida digital configurable mediante un potenciómetro. Es compatible con una amplia gama de tarjetas de desarrollo como **Arduino** (UNO, Mega, Nano), **ESP32**, **ESP8266**, **STM32**, y **Raspberry Pi**, lo que facilita su integración en sistemas embebidos para crear soluciones seguras y eficientes tanto en entornos educativos como profesionales.

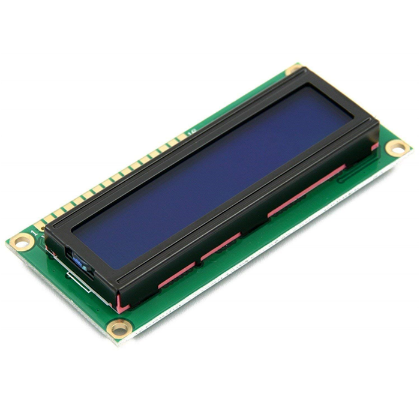
**Funcionamiento del Módulo MQ-6.**

El sensor propiamente se encuentra encerrado en dos capas de malla de acero inoxidable que asegura que el elemento calentador interno no cause una explosión dado que en su ambiente de trabajo puede haber presencia de gases inflamables, además filtra las partículas suspendidas para que solo gases accedan a la cámara. Dentro, se encuentra una bobina de níquel-cromo para formar el sistema de calefacción y un revestimiento de dióxido de estaño (que es sensible a gases combustibles) forma el sistema de detección.

Una vez se calienta el dióxido de estaño, absorbe el oxígenoo (del aire limpio) en su superficie, atrayendo electrones del dióxido de estaño y dificultando el flujo de corriente. En presencia de gases, la densidad de oxígeno absorbido disminuye liberando a los electrones, permitiendo que la corriente fluya con mayor libertad por el sensor.

## **Pantalla LCD**

Una pantalla LCD incluye una capa delgada de material de cristal líquido intercalada entre dos electrodos sobre sustratos de vidrio, con dos polarizadores en cada lado. **Un polarizador es un filtro óptico que deja pasar las ondas de luz de una polarización específica mientras bloquea las ondas de luz de otras polarizaciones.** Los electrodos deben ser transparentes, por lo que el material más popular es ITO (óxido de indio y estaño).

**Como la pantalla LCD no puede emitir luz por sí misma, normalmente se coloca una luz de fondo detrás de la pantalla LCD para poder verla en un entorno oscuro.** Las fuentes de luz para la retroiluminación pueden ser LED (diodo emisor de luz) o CCFL (lámparas fluorescentes de cátodo frío). La retroiluminación LED es la más popular. Por supuesto, si desea tener una pantalla a color, se puede convertir una capa de filtro de color en una celda LCD. El filtro de color consta de color RGB. También puede agregar un panel táctil frente a una pantalla LCD.

La primera tecnología de panel LCD en producción en masa se llama TN (Twisted Nematic). **El principio detrás de las pantallas LCD es que cuando no se aplica un campo eléctrico a las moléculas de cristal líquido, las moléculas giran 90 grados en la celda LCD.** Cuando la luz de la luz ambiental o de la luz de fondo pasa a través del primer polarizador, la luz se polariza y se retuerce con la capa molecular de cristal líquido. Cuando llega al segundo polarizador, se bloquea. El espectador ve que la pantalla es negra.

Cuando se aplica un campo eléctrico a las moléculas de cristal líquido, se desenroscan.  **Cuando la luz polarizada llega a la capa de moléculas de cristal líquido, la luz pasa directamente sin torcerse.** Cuando llegue al segundo polarizador, también pasará, el espectador verá que la pantalla está brillante.

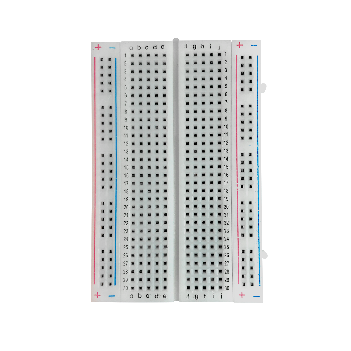
Debido a que la tecnología LCD utiliza campos eléctricos en lugar de corriente eléctrica (paso de electrones), tiene un bajo consumo de energía.

## **Servomotor**

Un servomotor es un motor eléctrico que incorpora un juego de engranes junto con un circuito electrónico de control, con la finalidad de poder controlar el giro de dicho motor y así poder elegir siempre la posición deseada. Normalmente estos dispositivos tienen un rango de operación que esta limitado a la hora de su construcción, por lo general trabajan con 180°. Aun que cabe mencionar es fácil modificar esta limitante para que pueda dar giros completos, ya que por lo general los engranes solo tienen un tope que es el que limita el giro.

Los servomotores funcionan a través de una señal eléctrica que tenemos que modificar según la posición en la que queremos girar nuestro motor. A estas modificaciones se le conoce como PWM por sus siglas en ingles **Pulse Wide Modulation** o **Modulación por ancho de pulso**. Todos los servos tienen tres cables uno de Voltaje (4.8v a 6v), uno para tierra (Gnd) y el tercero que se utiliza para recibir la señal modulada que se utiliza para controlar el giro.

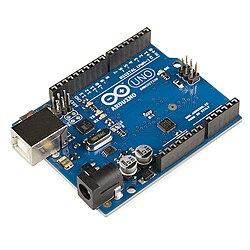
## **Protoboard**

Una Protoboard es un instrumento que permite probar el diseño de un circuito sin la necesidad de soldar o desoldar componentes. Las conexiones en una Protoboard se hacen con solo insertar los componentes lo que permite armar y modificar circuitos con mayor velocidad.

Normalmente estas placas son usadas para realizar pruebas experimentales. Si la prueba resulta satisfactoria el circuito se construye de una forma más permanente para evitar el riesgo de que algún componente pueda desconectarse. En caso de que la prueba no sea satisfactoria, puede modificarse el circuito fácilmente.

Las Protoboards tienen tres partes: el canal central, las pistas, y los buses. En el canal central, ubicado en la parte media, se conectan los circuitos integrados para mantener aislados los pines de ambos lados del circuito integrado. Los buses se encuentran el los lados de la Protoboard, y generalmente se emplean para conectar la tierra del circuito y su voltajes de alimentación. La mayoría de las veces los buses están indicados con franjas color negro o azul para indicar el bus de tierra, y con franjas color rojo para indicar el bus de voltaje positivo. El resto de los orificios de la Protoboard pertenecen a las pistas. Como se mencionó anteriormente, las pistas están separadas por filas. Las filas están indicadas con números y las columnas están indicadas con letras

## **Arduino Uno**

Arduino es una plataforma de desarrollo electrónico de código abierto que ha revolucionado el mundo de la tecnología educativa, la automatización y la creación de prototipos. Esta plataforma está compuesta por una combinación de hardware y software de libre uso, lo que significa que cualquier persona puede utilizarla, modificarla y adaptarla a sus necesidades sin restricciones. El hardware está representado por distintas placas electrónicas, como el popular Arduino Uno, que contienen un microcontrolador programable, pines de entrada y salida, y conexiones para alimentar dispositivos externos. Por su parte, el software se basa en el entorno de desarrollo Arduino IDE, una aplicación intuitiva que permite escribir código en un lenguaje basado en C/C++ y cargarlo fácilmente a la placa a través de un cable USB. Cada programa de Arduino se estructura alrededor de dos funciones esenciales: setup(), que configura el dispositivo al inicio, y loop(), que ejecuta de forma continua las instrucciones definidas mientras el sistema esté encendido.

Gracias a su diseño sencillo y su bajo costo, Arduino se ha convertido en una herramienta ampliamente utilizada tanto en el ámbito educativo como profesional. Permite a los usuarios crear dispositivos que interactúan con el entorno mediante sensores, como los de luz, temperatura, sonido o movimiento, y elementos de salida como luces LED, motores, pantallas y relés. Esto hace posible la realización de proyectos de automatización del hogar (domótica), sistemas de riego automático, robots móviles, estaciones meteorológicas, alarmas de seguridad, entre muchos otros. Además, Arduino cuenta con una comunidad global muy activa que comparte constantemente tutoriales, librerías y ejemplos, lo que facilita el aprendizaje y fomenta la colaboración entre desarrolladores de todo el mundo.

## **Leds**

Un LED (acrónimo de Light-Emitting Diode, o diodo emisor de luz) es un dispositivo semiconductor que, al recibir una corriente eléctrica en una unión p-n, genera luz mediante el fenómeno de la electroluminiscencia. Está constituido por un pequeño chip encapsulado en materiales protectores, y se activa con una corriente continua que fluye desde el ánodo al cátodo. La luz emitida puede variar en color (desde infrarrojo hasta ultravioleta) según el tipo de material semiconductor utilizado, ya que la longitud de onda depende de la brecha energética de la unión p-n. Existen diferentes variantes como los LEDs SMD (Surface-Mount Device), diseñados para montarse directamente sobre placas de circuito impreso y, en versiones RGB, para ofrecer múltiples colores mediante combinaciones de celdas independientes. También se encuentran los más robustos LEDs DIP (Dual In-Line Package), especialmente utilizados en pantallas exteriores, valorados por su alta resistencia al impacto y brillo intenso. Entre sus ventajas destacan su eficiencia energética (muchos lúmenes por vatio), una vida útil que puede alcanzar entre 80 000 y 120 000 horas, encendido instantáneo, tamaño compacto y alta durabilidad. Gracias a estas características, los LEDs se emplean en una amplia variedad de aplicaciones: iluminación general y decorativa, pantallas (LCD retroiluminadas o gigantes LED), indicadores electrónicos, comunicación óptica (como los mandos a distancia o enlaces de fibra), sistemas de señalización, módulos informativos y en dispositivos de estado sólido (SSL). 

## **Descripción de la Placa Arduino**

La placa Arduino Uno empleada en este proyecto constituye el núcleo del sistema de control. Se trata de una plataforma de desarrollo basada en el microcontrolador ATmega328P, que opera a una frecuencia de 16 MHz con un voltaje de trabajo estándar de 5V. La placa incluye:

* 14 pines digitales (6 con capacidad PWM)
* 6 entradas analógicas
* 32 KB de memoria Flash para almacenamiento de programas
* 2 KB de SRAM y 1 KB de EEPROM
* Interfaz USB para programación y comunicación serial

La arquitectura de la placa permite:

1. Lectura precisa de señales analógicas del sensor MQ6
2. Control simultáneo de múltiples actuadores (servomotor, LED, LCD)
3. Procesamiento en tiempo real de los datos del sensor
4. Comunicación bidireccional con la aplicación móvil

El diseño abierto de Arduino facilita la implementación de librerías específicas como:

* LiquidCrystal.h para manejo de pantallas
* Servo.h para control preciso de motores
* SoftwareSerial para comunicación adicional

La placa se alimenta mediante USB o fuente externa (7-12V DC), consumiendo menos de 50mA en operación normal, lo que la hace ideal para sistemas de monitoreo continuo. Su robustez y amplia comunidad de soporte la convierten en la opción óptima para prototipos educativos y aplicaciones reales.

## **Interfaz Usuario-Arduino con App Inventor**

Para la interacción con el usuario, se implementó una aplicación móvil desarrollada en MIT App Inventor que se comunica con el Arduino mediante módulo Bluetooth HC-05. La solución integra:

Funcionalidades clave:

* Visualización en tiempo real de los niveles de gas detectados
* Alertas inmediatas cuando se supera el umbral de seguridad
* Envío automático de SMS a números predefinidos en caso de emergencia
* Historial gráfico de las mediciones
* Botón para controlar el cierre de la ventana

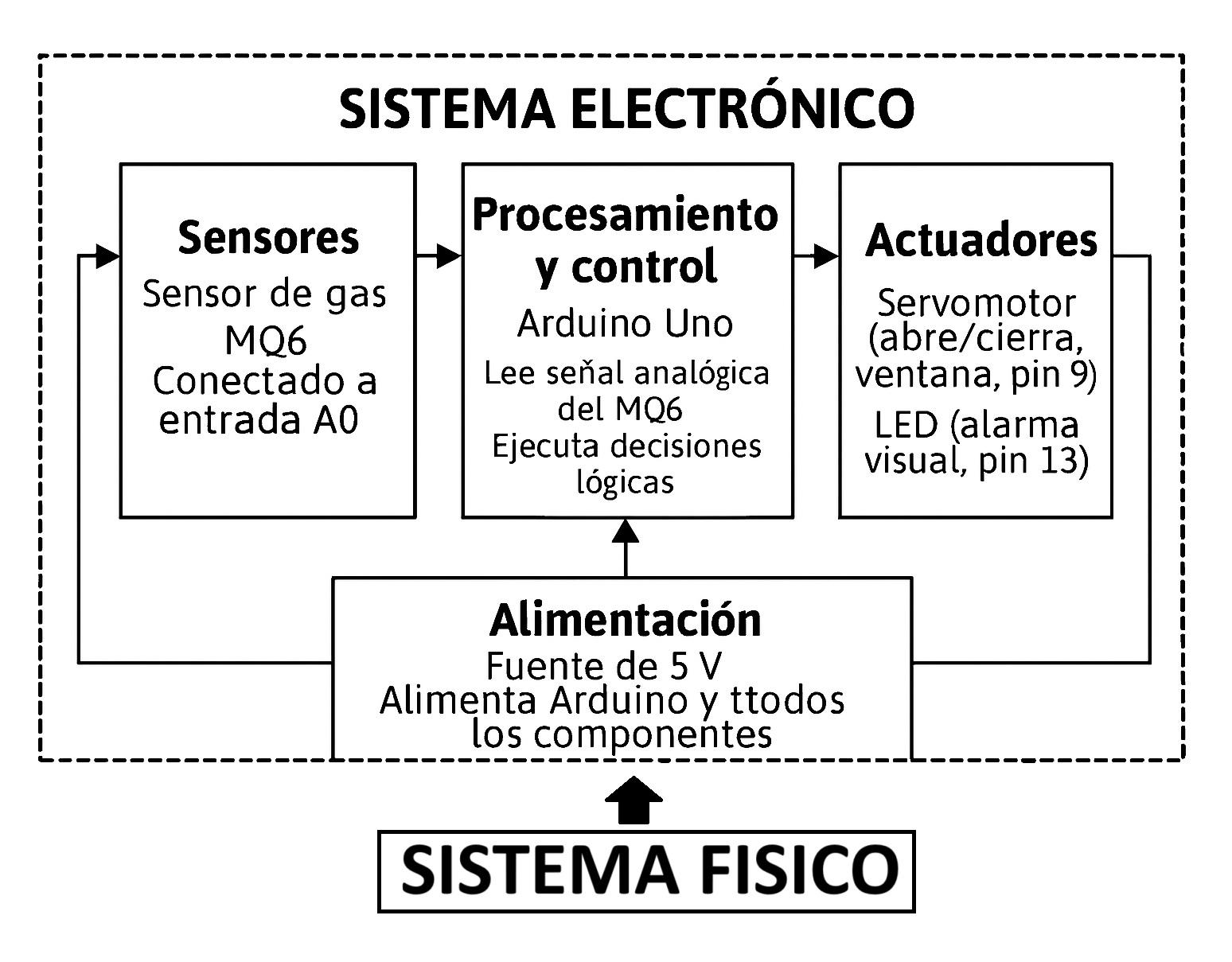
Configuración técnica:

* Protocolo de comunicación serial a 9600 baudios
* Diseño de interfaz intuitiva con indicadores visuales
* Integración con servicios de mensajería para notificaciones
* Compatibilidad con dispositivos Android

La aplicación procesa los datos recibidos del Arduino y activa los protocolos de alerta correspondientes, combinando una respuesta local (notificaciones en el dispositivo) con acciones remotas (envío de SMS), garantizando una doble capa de seguridad para el usuario.

* **Justificación de la selección de MIT App Inventor**

Se eligió MIT App Inventor como plataforma para desarrollar la interfaz usuario-Arduino debido a su combinación ideal de simplicidad y funcionalidad para proyectos de prototipado. Su entorno de programación visual basado en bloques permite crear aplicaciones Android funcionales sin necesidad de dominar lenguajes de programación móvil avanzados, lo que lo hace perfecto para implementaciones educativas y proyectos con plazos ajustados. La plataforma ofrece compatibilidad nativa con módulos Bluetooth como el HC-05, facilitando la comunicación bidireccional con Arduino para recibir datos del sensor MQ6 y enviar comandos. Además, incluye componentes listos para usar, como el envío de SMS y notificaciones push, esenciales para el sistema de alertas ante fugas de gas. A diferencia de otras opciones como Blynk (con limitaciones en su versión gratuita) o el desarrollo nativo (que requiere mayor complejidad técnica), App Inventor proporciona todas las herramientas necesarias de forma gratuita y accesible, sin sacrificar las capacidades críticas del proyecto: monitoreo en tiempo real, registro de datos y notificaciones inmediatas vía mensajes de texto y la aplicación móvil. Esta combinación de características lo convierte en la solución óptima para integrar Arduino con una interfaz amigable y funcional para el usuario final.

* **Diagrama de Bloques**
* **Descripción del diagrama Electrónico**

Esquema:

1. Sensores: MQ6 conectado a entrada analógica A0.
2. Procesamiento: Arduino Uno (interpreta señales).
3. Actuadores:
   * Servomotor (pin 9) → Ventana.
   * LED (pin 13) → Alarma.
4. Alimentación: Fuente de 5V para Arduino y componentes.

# Desarrollo

## **Delimitación de Alcances**

Alcance Deseado:

* Sistema de detección de gas con sensor MQ6 y Arduino Uno.
* Visualización de datos en pantalla LCD y aplicación móvil.
* Activación de alarmas (LED y servomotor) al detectar gas.
* Interfaz de usuario intuitiva para monitoreo remoto.

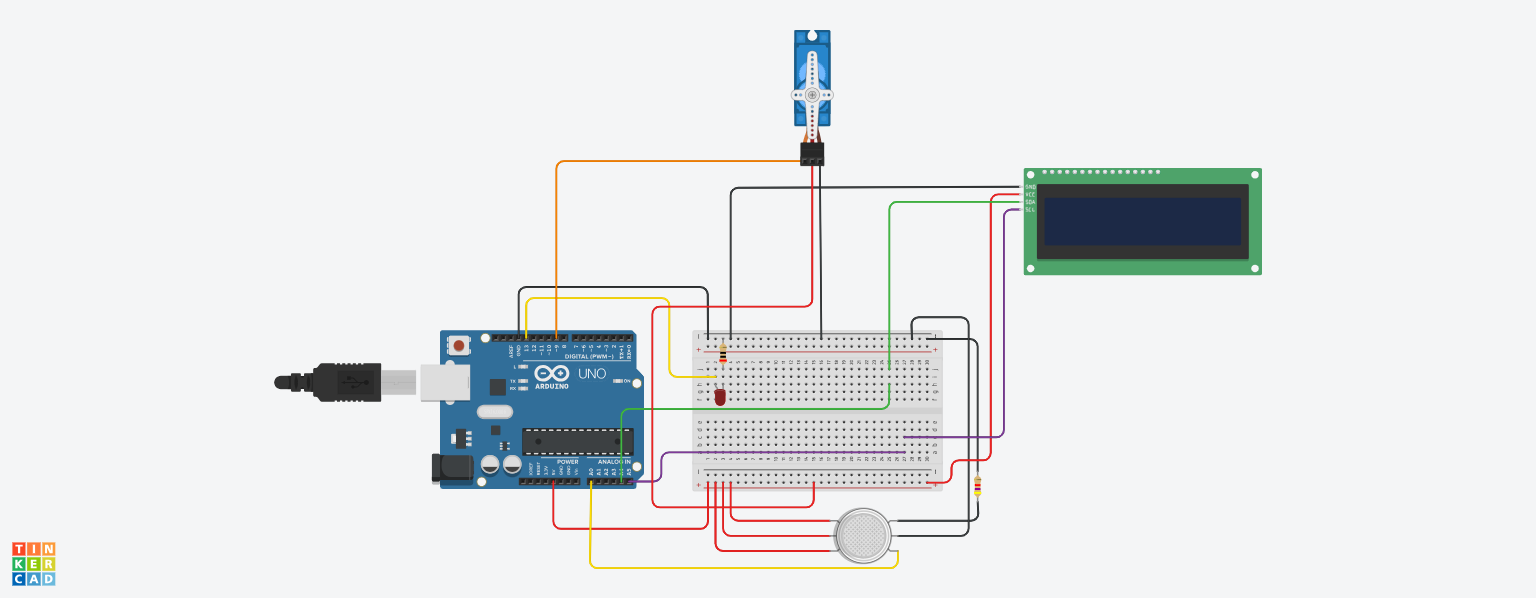
Alcance Real:

* Detección básica de gas con umbral ajustable (sin calibración profesional).
* Aplicación móvil con funcionalidad básica (sin historial avanzado ni base de datos).
* Prototipo funcional en maqueta (no escala industrial).
* Comunicación Arduino-App vía cable/USB (no inalámbrica en esta fase).

Limitaciones:

* Precisión del sensor afectada por condiciones ambientales.
* La app no envía notificaciones push (solo muestra datos en tiempo real).
* Servomotor simula apertura de ventana, pero no incluye mecanismo real.

## **Imagen de armado de circuito**



## **Código fuente**

El código fuente realizado para el funcionamiento de este proyecto se encuentra adjunta en una carpeta de drive denominada “Arduino y sensor” dentro de la misma carpeta del proyecto, el código que se encuentra en esa carpeta ayuda a l funcionamiento del sensor, e indica al servomotor y al led a que se enciendan para un correcto funcionamiento del proyecto

[Codigo Fuente](https://docs.google.com/document/d/1lEpqUQPSRfbqSxmsmmqMU65-XKXsLDlmlh1hY7jOycg/edit?usp=sharing)

## **Aplicación móvil**

El archivo tipo .aia la que se uso para este proyecto se encuentra adjunto en una carpeta denominada “Aplicación móvil” dentro del mismo archivo el proyecto el cual contiene el archivo que proporciona la aplicación la cual su función es conectarse a el módulo bluetooth y posteriormente muestra la información de que cantidad esta detectando el sensor de gas además contiene un botón que nos ayuda al cerrado de la ventana

<https://drive.google.com/file/d/1fT67Y2CKu_XITv_gH3jfQeO-blbKnYP8/view?usp=sharing>

# Conclusión

## **Aporte Individual**

Palmer

Palmer desempeñó un rol fundamental en la estructuración técnica y documental del proyecto. Su trabajo comenzó con la revisión exhaustiva de hojas de datos y estándares técnicos, asegurando que todos los componentes cumpliesen con los requerimientos del diseño. Además, lideró la elaboración de la documentación formal, redactando informes detallados y organizando la información para facilitar su consulta. En el ámbito técnico, se encargó del análisis de los datasheets de los sensores, determinando sus parámetros críticos para una implementación óptima. También tuvo a su cargo la selección de los lenguajes de programación, evaluando las ventajas de AppInventor y Arduino para garantizar la eficiencia del sistema. Palmer gestionó la compra y verificación de materiales, asegurándose de que cada componente fuese compatible y estuviese disponible a tiempo. En la fase de implementación, realizó las conexiones eléctricas y participó activamente en el armado físico del prototipo, optimizando su diseño para mejorar funcionalidad y estética. Durante las pruebas, fue clave en la detección de errores, calibración de sensores y ajustes de hardware, además de colaborar en la preparación de la presentación final. Su enfoque meticuloso y técnico aseguró que el proyecto cumpliera con altos estándares de calidad.

Carlos

Carlos aportó al proyecto una combinación de habilidades técnicas y organizativas, destacándose en la planificación y desarrollo del software. Inició su participación con la revisión de información técnica y materiales de referencia, extrayendo datos clave para fundamentar las decisiones del equipo. Elaboró el listado preliminar de materiales, priorizando componentes esenciales y verificando su disponibilidad. Una de sus contribuciones más significativas fue la creación del cronograma definitivo, asignando tiempos realistas a cada etapa y ajustándose según los avances y desafíos encontrados. En el ámbito de la programación, Carlos lideró el desarrollo de la aplicación en AppInventor, diseñando una interfaz intuitiva y garantizando la comunicación efectiva con el hardware mediante Bluetooth/WiFi. También participó en la selección de los lenguajes de programación, asegurando que las opciones elegidas fueran viables y eficaces. Durante la fase de construcción, colaboró en el armado físico y maquetado, sugiriendo mejoras para facilitar el mantenimiento del prototipo. En las pruebas, ejecutó evaluaciones funcionales, documentó resultados y optimizó el código para mejorar el rendimiento del sistema. Su trabajo en la calibración y programación del Arduino fue crucial para afinar la precisión de los sensores. Finalmente, contribuyó a la presentación del proyecto, preparando demostraciones y verificando que todos los entregables cumpliesen con los requisitos establecidos. Su enfoque organizado y resolutivo fue clave para el éxito del trabajo en equipo.

## **Conclusion Individual**

Palmer  
Este proyecto representó una valiosa oportunidad para aplicar conocimientos técnicos en un desarrollo con impacto práctico. Mi principal contribución se centró en garantizar el correcto funcionamiento del hardware: desde la selección de componentes hasta la implementación física del circuito. El proceso de calibración del sensor MQ6 fue particularmente revelador, mostrándome la importancia de comprender a profundidad las especificaciones técnicas para obtener mediciones confiables.

Uno de los aprendizajes más significativos fue comprobar cómo la documentación detallada facilita tanto el desarrollo como la replicabilidad del proyecto. Cada ajuste en las conexiones eléctricas o en la disposición de componentes demostró que pequeños detalles pueden marcar la diferencia en el rendimiento final del sistema.

Carlos  
Mi participación en el proyecto me permitió desarrollar habilidades clave en integración de sistemas. El mayor reto -y logro- fue establecer una comunicación efectiva entre el Arduino y la aplicación móvil. Implementar la interfaz en App Inventor requirió múltiples iteraciones, pero el resultado final validó el esfuerzo invertido.

Este trabajo reforzó mi comprensión sobre la importancia de diseñar soluciones pensando en el usuario final. La creación de alertas visuales claras y la simplificación de la interfaz fueron aspectos cruciales que mejoraron sustancialmente la utilidad práctica del sistema. También valoré la necesidad de planificación flexible, adaptándonos a retos técnicos inesperados sin perder de vista los objetivos principales.

## **Conclusión General**

El desarrollo de este detector de gas demostró cómo la combinación de hardware accesible y software adecuado puede generar soluciones tecnológicas con aplicaciones reales. El proyecto cumplió satisfactoriamente con su objetivo principal: crear un sistema funcional de detección temprana, integrando componentes electrónicos básicos con una interfaz amigable.

Los principales aprendizajes obtenidos fueron:

1. La importancia de la calibración precisa en sistemas de sensado
2. La necesidad de pruebas iterativas para garantizar confiabilidad
3. El valor de documentar metódicamente cada etapa del desarrollo

Como trabajo futuro, el prototipo ofrece oportunidades interesantes de mejora, particularmente en cuanto a estabilidad de la comunicación inalámbrica y ampliación de funcionalidades. Este proyecto no solo representó un ejercicio académico, sino que además evidenció el potencial de soluciones tecnológicas accesibles para abordar problemas de seguridad cotidianos.

# Fuentes de informacion

Antecedentes

* ARM Holdings. (s.f.). *Manual de referencia técnica de la serie ARM Cortex-M*. [https://developer.arm.com](https://developer.arm.com/)
* Atmel Corporation. (2016). *Introducción a los microcontroladores AVR: Programación e interfaz*. Springer.
* Espressif Systems. (2023). *Manual de referencia técnica del ESP32*. [https://www.espressif.com](https://www.espressif.com/)
* Kernighan, B. W., & Ritchie, D. M. (1988). *El lenguaje de programación C* (2a ed.). Prentice Hall.
* Microchip Technology. (s.f.). *Microcontroladores PIC*. [https://www.microchip.com](https://www.microchip.com/)
* MIT App Inventor. (2023). *Documentación oficial*. [http://appinventor.mit.edu](http://appinventor.mit.edu/)
* Orient Display. (2021). *Tecnología LCD explicada*. [https://www.orientdisplay.com](https://www.orientdisplay.com/)
* Philips Semiconductors. (2003). *Especificación y manual de usuario del bus I²C* (UM10204).
* Fundación Raspberry Pi. (2023). *Hoja técnica de Raspberry Pi Pico*. [https://www.raspberrypi.com](https://www.raspberrypi.com/)
* STMicroelectronics. (2022). *Manual de referencia STM32F1xx* (Doc ID RM0008).
* Texas Instruments. (2021). *Conceptos básicos de conversión analógico-digital* (Informe de aplicación SLAA588).
* Zhengzhou Winsen Electronics Technology Co., Ltd. (2022). \*Hoja técnica del sensor de gas MQ-6\*. [https://www.winsen-sensor.com](https://www.winsen-sensor.com/)
* Valdés, F., & Pérez, M. (2018). *Sistemas embebidos: Fundamentos y aplicaciones con microcontroladores PIC y ARM* (2a ed.). Alfaomega.
* González, J. A. (2020). *Electrónica aplicada a sistemas embebidos: Sensores, actuadores y comunicación IoT.* Marcombo.
* Martínez, R., & López, E. (2019). *Diseño de sistemas digitales con microcontroladores: De la teoría a la práctica.* McGraw-Hill.

*Proyecto*

* *MQ-6 Detector de Gas LP - UNIT Electronics*. (2025, July 12). UNIT Electronics. https://uelectronics.com/producto/mq-6-detector-de-gas-lpg/?srsltid=AfmBOorM8nJI3LniTqRmJvxMNxTDf9tPUu3gzHdnTg\_\_XIVwylKpMAq8
* ‌ *Circuit design Amazing Waasa-Blad - Tinkercad*. (2025). Tinkercad. https://www.tinkercad.com/things/kNrsqoQedV9/editel?returnTo=%2Fdashboard
* ‌ *¿Qué es una pantalla LCD? Tecnología LCD y tipos de pantalla | Orientar pantalla*. (2021, November 24). Orient Display. <https://www.orientdisplay.com/es/knowledge-base/lcd-basics/what-is-lcd-liquid-crystal-display/>
* 330ohms. (2024, July 30). *330ohms*. 330ohms. https://www.330ohms.com/blogs/blog/protoboards?srsltid=AfmBOoqA8y--yGhCn6Wq88mlP-Gp0RjbIEpsUhv4wCYr6C9-vTrI7ere
* ‌Yúbal Fernández. (2025, June 25). *Qué es Arduino, cómo funciona y qué puedes hacer con uno*. Xataka.com; Xataka. https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno
* *¿Qué es un led? Tipos de Led, aplicaciones y usos.* (2018, August 2). Visual Led. https://visualled.com/glosario/que-es-un-led/
* ‌

‌