

# SENSORES CON ARDUINO EN TINKERCAD

Hugo Andrade, Marshall Flores Universidad de las Fuerzas Armadas “E.S.P.E” Ingeniería en electrónica y Telecomunicaciones

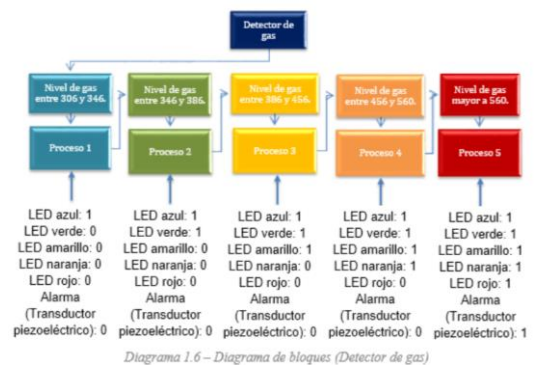
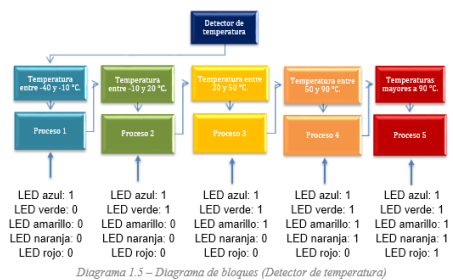
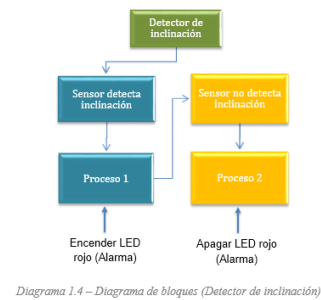
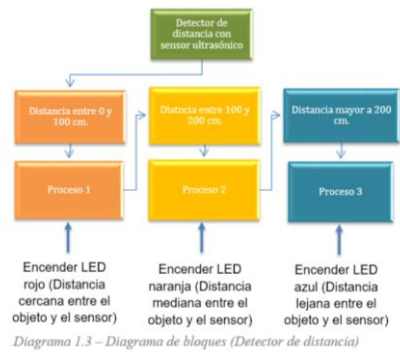
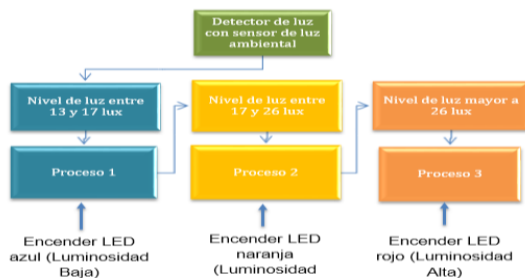
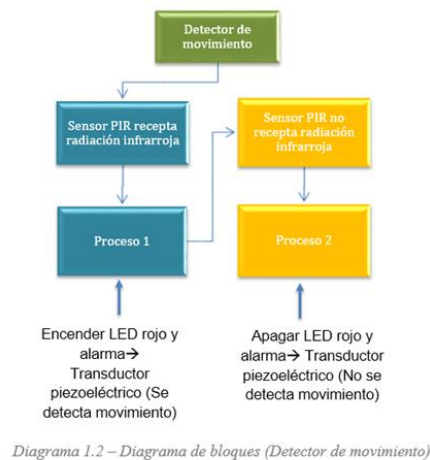
[hdandrad@espe.edu.ec](mailto:hdandrad@espe.edu.ec) , [jmflores15@espe.edu.ec](mailto:jmflores15@espe.edu.ec)

**Resumen:** La utilización de Arduino hoy en día es muy demandada en proyectos de electrónica, ya que al ser una plataforma libre y de código abierto, los usuarios pueden modificar los requerimientos a sus necesidades, debido a que incorpora un microcontrolador y un entorno de desarrollo que permiten que todo esto sea posible. En nuestro caso, al ser un entorno virtual en el cual vamos a trabajar, utilizaremos el Arduino UNO que incorpora Tinkercad, realizando la programación de un circuito mediante un entorno de desarrollo gráfico (Programación por bloques). El propósito de este trabajo de investigación es analizar los aspectos de hardware que ofrece un Arduino, haciendo énfasis en la entrada (sensores) y salida de información (indicadores LEDs, alarmas). Esto se determinará a partir de la fundamentación correspondiente a los componentes que conforma un Arduino, su funcionalidad y aplicándolo a diferentes circuitos, en los que se presente la operatividad y funcionamiento de algunos sensores que incorpora Tinkercad, tales como, el sensor de luz ambiental, sensor PIR, sensor de distancia, sensor de inclinación, sensor de temperatura y un sensor de gas.

## Introducción

Implementar y analizar en base a la información de datos de entrada y salida de un Arduino, diferentes circuitos en los cuales se muestre el funcionamiento de los mismos, a través de la programación en un entorno gráfico y uso del Arduino en tinkercad, con el fin de entender el comportamiento del sistema de hardware que ofrece esta plataforma, junto con los sensores que serán utilizados para cada circuito.

## Diagramas



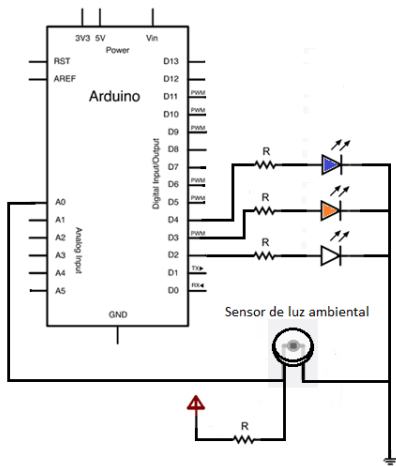


Diagrama 1.7 – Diagrama eléctrico – Detector de luz

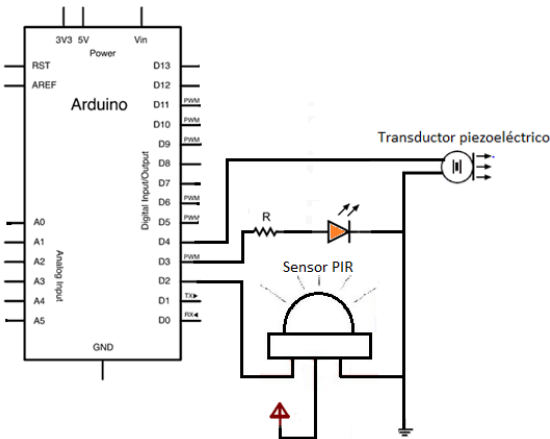


Diagrama 1.8 – Diagrama eléctrico – Detector de movimiento

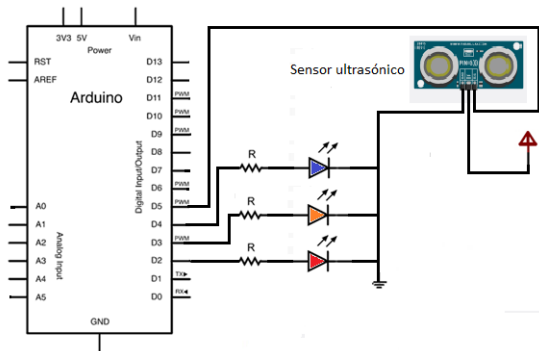


Diagrama 1.9 – Diagrama eléctrico – Detector de distancia

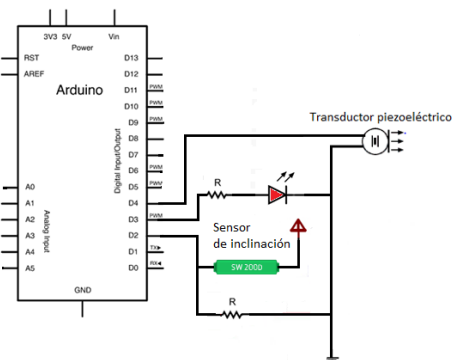


Diagrama 2.1 – Diagrama eléctrico – Detector de inclinación

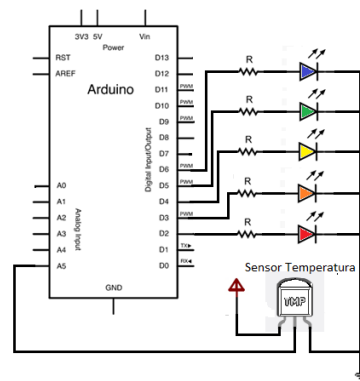


Diagrama 2.2 – Diagrama eléctrico – Detector de temperatura

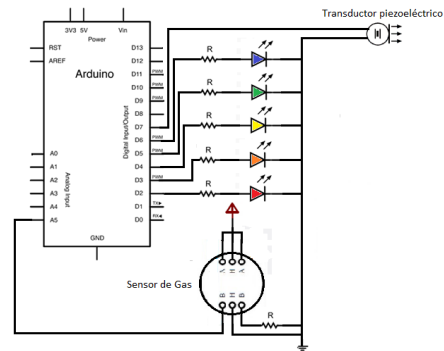


Diagrama 2.3 – Diagrama eléctrico – Detector de gas

## Lista de componentes

Componentes utilizados	
Software online TINKERCAD	Software en línea utilizado para realizar la simulación de un laboratorio virtual, cuya experiencia es muy similar a lo que se realizaría en un laboratorio físico. En este caso fue utilizado para representar el funcionamiento de un semáforo, mediante un Arduino, cuya programación se la elaboró mediante un entorno gráfico (programación por bloques).
Arduino	Es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. La placa en sí, está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos. En este caso, se lo programará mediante Tinkercad.
Protoboard	Placa de pruebas para realizar la conexión de todos los elementos que conforman un circuito
Resistencias	Componente electrónico utilizado para proteger a los demás elementos que conforman el circuito, oponiéndose al paso de la corriente
Diodos LED	Componente, cuya fuente de luz está constituida por un material semiconductor. En este caso es el equivalente a las luces de un semáforo. (Salidas)

Sensor de luz ambiental	Sensor diseñado para proporcionar una precisa y fiable determinación de la cantidad de luz en el ambiente en el que se encuentra.
Sensor PIR	Componente electrónico diseñado para detectar cambios en la radiación infrarroja recibida. Trabajan en su totalidad para la detección de la energía emitida por otros objetos.
Sensor ultrasónico	Componente electrónico que mide la distancia existente entre un objeto y el sensor, mediante el uso de ondas ultrasónicas
Sensor de inclinación	Sensor que mide sin el ángulo de inclinación de un objeto en relación con la gravitación de la Tierra
Sensor de temperatura	Sensor utilizado para hacer funcionar ciertos circuitos o componentes dependiendo de la temperatura del ambiente en el que se encuentra.
Sensor de gas	Sensores usados para prevenir la exposición a gases combustibles y gases tóxicos.
Transductor piezoeléctrico	Dispositivos muy apropiados para tomar datos de vibración a alta frecuencia (Utilizado como señal de alarma).

## EXPLICACIÓN DEL CÓDIGO FUENTE

### Detector de luz ambiental

El funcionamiento del programa básicamente consiste de estructuras condicionales “If-Else” anidadas, en las que se postulan las diferentes condiciones para que los LEDs funcionen como indicadores de la existencia de bajas, medianas y altas cantidades de luz. (Figura 2.4). Por otro lado, la primera línea de código (expresada en bloques), nos permite asignar el pin analógico A0 a la variable creada “BrilloLuz”. La segunda línea de código, nos permite imprimir en un monitor, los valores o niveles de luminosidad que el sensor capta.



Figura 2.4. Secuencia de bloques - Operatividad de un detector de luz ambiental.

#### - Alta luminosidad

La primera estructura condicional nos estipula que, si la variable BrilloLuz se encuentra en el rango de 13 y 17, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.4 – Llave de color rojo), en la que únicamente el pin digital 2 (LED blanco) se encontrará en estado lógico ALTO. Esto quiere decir que existen altos niveles de luz captados por el sensor de luz ambiental en un determinado lugar.

#### - Mediana luminosidad

La segunda estructura condicional nos estipula que, si la variable BrilloLuz se encuentra en el rango de 17 y 26, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.4 – Llave de color azul), en la que únicamente el pin digital 3 (LED naranja) se encontrará en estado lógico ALTO. Esto quiere decir que existen medianos niveles de luz captados por el sensor de luz ambiental en un determinado lugar.

#### - Baja luminosidad

Si no se cumple con ninguna de las dos estructuras condicionales anteriores, se asume que la variable BrilloLuz se encontrará en el rango de 26 en adelante, entonces el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.4 – Llave de color naranja), en la que únicamente el pin digital 4 (LED azul) se encontrará en estado lógico ALTO. Esto quiere decir que existen bajos niveles de luz captados por el sensor de luz ambiental en un determinado lugar.

### Detector de movimiento

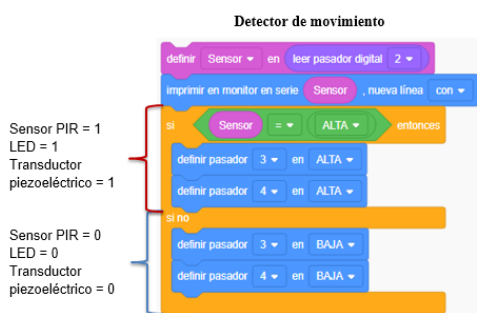


Figura 2.5. Secuencia de bloques - Operatividad de un detector de movimiento.

#### - Existencia de Movimiento

La primera estructura condicional (Ilustrada en la figura 2.5 – Llave de color rojo) nos estipula que, si la variable Sensor se encuentra en estado lógico ALTO, el programa ejecutará la secuencia de bloques, en la que el pin digital 3 (LED) y el pin digital 4 (Transductor piezoeléctrico) se encontrarán en estado lógico ALTO. Esto quiere decir que el sensor recibió niveles de radiación infrarroja, lo cual implica que existió movimiento en un área específica. Para indicar dicho suceso, hemos considerado encender el LED y el transductor piezoeléctrico en señal de alarma.

#### - No existe movimiento

Si no se cumple dicha condición se ejecutará la secuencia de bloques (Ilustrada en la figura 2.5 – Llave de color azul), en la que el pin digital 3

(LED) y el pin digital 4 (Transductor piezoeléctrico) se encontrarán en estado lógico BAJO. Esto quiere decir que el sensor no recibió niveles de radiación infrarroja, lo cual implica que no existió movimiento en un área específica.

### Detector de distancia



Figura 2.6. Secuencia de bloques - Operatividad de un detector de distancia.

#### - Corta Distancia

La primera estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de distancia entre 0 y 100 cm, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.6 – Llave de color rojo), en la que únicamente el pin digital 2 (LED rojo) se encontrará en estado lógico ALTO. Esto quiere decir que la distancia entre el objeto y el sensor ultrasónico es corta. Para lo cual hemos encendido un LED rojo en señal de alarma.

#### - Mediana Distancia

La segunda estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de distancia entre 100 y 200 cm, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.6 – Llave de color azul), en la que únicamente el pin digital 3 (LED naranja) se encontrará en estado lógico ALTO. Esto quiere decir que la distancia entre el objeto y el sensor ultrasónico es mediana. Para lo cual hemos encendido un LED naranja para indicar dicha distancia.

#### - Larga Distancia

Si no se cumple con ninguna de las dos estructuras condicionales anteriores, se asume que la variable Sensor se encontrará en el rango de 200 cm. en adelante, entonces el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.6 – Llave de color naranja), en la que únicamente el pin digital 4 (LED azul) se encontrará en estado lógico ALTO. Esto quiere decir que la distancia entre el objeto y el sensor ultrasónico es lejana. Para lo cual hemos encendido un LED azul para indicar dicha distancia.

### Detector de inclinación



Figura 2.7. Secuencia de bloques - Operatividad de un detector de inclinación.

#### - Existencia de inclinación

La estructura condicional (Ilustrada en la figura 2.7 – Llave de color rojo) nos estipula que, si la variable Sensor se encuentra en estado lógico ALTO, el programa ejecutará la secuencia de bloques, en la que el pin digital 3 (LED) se encontrará en estado lógico ALTO. Esto quiere decir que existió inclinación de un objeto y esto fue captado por el sensor, expresándolo como una señal digital equivalente a 1 lógico. Para indicar dicho suceso, hemos considerado encender un LED.

#### - No existe movimiento

Si no se cumple dicha condición se ejecutará la secuencia de bloques (Ilustrada en la figura 2.5 – Llave de color azul), en la que el pin digital 3

(LED se encontrarán en estado lógico BAJO. Esto quiere decir que no existió inclinación alguna del objeto y esto fue captado por el sensor, expresándolo como una señal digital equivalente a 0 lógico. Para indicar dicho suceso, hemos considerado apagar el LED.

**Detector de temperatura**



Figura 2.8. Secuencia de bloques - Operatividad de un detector de temperatura.

**- Temperaturas entre -40°C y -10°C**

La primera estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de temperatura entre -40 y -10°C, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.8 – Llave de color azul), en la que únicamente el pin digital 6 (LED azul) se encontrará en estado lógico ALTO. Esto quiere decir que la temperatura captada por el sensor es baja. Para lo cual hemos encendido un LED azul para representar dicha característica.

**- Temperaturas entre -10 °C y 20°C**

La segunda estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de temperatura entre -10 y 20°C, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.8 – Llave de color verde), en la que los pines digitales 6 y 5 (LED azul y verde respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que la temperatura captada por el sensor aumentó en relación al caso anterior. Para lo cual hemos encendido los LEDs azul y verde para representar dicha característica.

**- Temperaturas entre 20 °C y 50°C**

La tercera estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de temperatura entre -20 y 50°C, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.8 – Llave de color amarillo), en la que los pines digitales 6, 5 y 4 (LEDs azul, verde y amarillo respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que la temperatura captada por el sensor aumentó en relación a los dos casos anteriores. Para lo cual hemos encendido los LEDs azul, verde y amarillo para representar dicha característica.

**- Temperaturas entre 50 °C y 90°C**

La cuarta estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de temperatura entre 50 y 90°C, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.8 – Llave de color naranja), en la que los pines digitales 6, 5, 4 y 3 (LEDs azul, verde, amarillo y naranja respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que la temperatura captada por el sensor aumentó en relación a los tres casos anteriores. Para lo cual hemos

encendido los LEDs azul, verde, amarillo y naranja para representar dicha característica.

**- Temperaturas mayores a 90 °C**

Si no se cumple ninguna de las estructuras condicionales anteriores, se asume que la temperatura estará con valores mayores a 90°C, entonces el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.8 – Llave de color rojo), en la que los pines digitales 6, 5, 4, 3 y 2 (LEDs azul, verde, amarillo, naranja y rojo respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que la temperatura captada por el sensor aumentó en relación a los cuatro casos anteriores, siendo los niveles más altos de temperatura. Para lo cual hemos encendido todos los LEDs y así representar dicha característica. En otras palabras, los Leds indican de manera gráfica la escala de temperatura que es captada por el sensor.

**Sensor de gas**

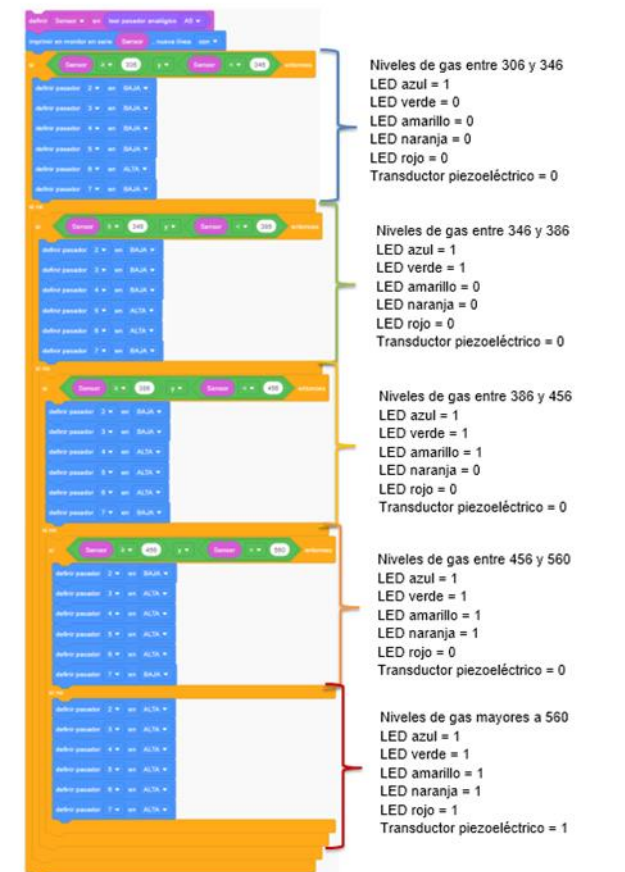


Figura 2.9. Secuencia de bloques - Operatividad de un detector de gas.

**- Nivel de gas entre 306 y 346**

La primera estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de nivel de gas entre 306 y 346, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.9 – Llave de color azul), en la que únicamente el pin digital 6 (LED azul) se encontrará en estado lógico ALTO. Esto quiere decir que los niveles de gas captados por el sensor son bajos. Para lo cual hemos encendido un LED azul, representando así dicha característica.

**- Nivel de gas entre 346 y 386**

La segunda estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de nivel de gas entre 346 y 386, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.9 – Llave de color verde), en la que los pines digitales 6 y 5 (LEDs azul y verde respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que los niveles de gas captados por el sensor aumentaron en relación al caso anterior. Para lo cual hemos encendido los LEDs azul y verde, representando así dicha característica.

**- Nivel de gas entre 386 y 456**

La tercera estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de nivel de gas entre 386 y 456, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.9 – Llave de color amarillo), en la que los pines digitales 6, 5 y 4 (LEDs azul, verde y



amarillo respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que los niveles de gas captados por el sensor aumentaron en relación a los dos casos anteriores. Para lo cual hemos encendido los LEDs azul, verde y amarillo, representando así dicha característica.

#### **Nivel de gas entre 456 y 560**

La cuarta estructura condicional nos estipula que, si la variable Sensor se encuentra en el rango de nivel de gas entre 456 y 560, el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.9 – Llave de color naranja), en la que los pines digitales 6, 5, 4 y 3 (LEDs azul, verde, amarillo y naranja respectivamente) se encontrarán en estado lógico ALTO. Esto quiere decir que los niveles de gas captados por el sensor aumentaron en relación a los tres casos anteriores. Para lo cual hemos encendido los LEDs azul, verde, amarillo y naranja, representando así dicha característica.

#### **Niveles de gas mayores a 560**

Si no se cumple ninguna de las estructuras condicionales anteriores, se asume que los niveles de gas están por encima de 560, entonces el programa ejecutará la secuencia de bloques (Ilustrada en la figura 2.9 – Llave de color rojo), en la que los pines digitales 6, 5, 4, 3 y 2 (LEDs azul, verde, amarillo, naranja y rojo respectivamente) se encontrarán en estado lógico ALTO, al igual que el pin digital 7 (Transductor piezoeléctrico). Esto quiere decir que los niveles de gas captados por el sensor aumentaron en relación a los cuatro casos anteriores, siendo niveles altos de gas captados por el sensor. Para lo cual hemos encendido todos los LEDs y el transductor piezoeléctrico que representa una señal de alerta o alarma por altas cantidades de gas situadas en un lugar determinado.

Una de las ventajas de programar un Arduino en Tinkercad, es que además de elaborar el programa mediante bloques podemos visualizar el mismo código de manera “textual”. Y de esta manera tener dos opciones de realización para el programador, dependiendo de cual le resulte más intuitiva. A continuación, mostramos el código que se generó de manera automática en Tinkercad (Figura 1.4).

### **CONCLUSIONES**

-La implementación de los seis circuitos, cuya operatividad se basa en la función de cada sensor utilizado, se lo ha realizado de tal manera que los datos de entrada y salida que se utilizaron se relacionen entre sí por medio de las instrucciones que se establecieron en el programa elaborado en un entorno gráfico, en el cual se implementaron principalmente estructuras condicionales para cumplir el funcionamiento de dichos circuitos, dependiendo del estado lógico o el rango de valores en el que se encuentren las entradas (Sensores) para que sus salidas (LEDs y transductores piezoeléctricos) se comporten como señal de alarma o una escala que obedezca a los parámetros del nivel de luz, temperatura y gas existente en un determinado lugar, también la existencia o no de movimiento e inclinación de determinados objetos y la distancia existente entre un objeto con respecto al sensor. Claramente todo ello depende del sensor con el que se esté trabajando.

-La familiarización con la funcionalidad de cada componente que conforma a un Arduino (Hardware), en este caso el Arduino UNO, fueron muy importantes para lograr realizar nuestro circuito y entender cómo interactúan, mediante el programa para con los componentes del circuito implementado. Básicamente el cerebro de esta plataforma electrónica es el microcontrolador ATMEGA 328, el cual contiene un total de 28 pines o puertos, y 14 de ellos (correspondientes a los pines digitales) hemos utilizado para la entrada y salida de información, es decir, los correspondientes a los sensores PIR, de distancia (ultrasónico), y de inclinación y LED's respectivamente. Además de ello también se utilizaron los pines de entrada analógicos para leer los datos captados por los sensores de luz ambiental, temperatura y gas. Hay que tener en cuenta que el reloj oscilador del Arduino es fundamental, ya que gracias a él las instrucciones se realizan con respecto a los tiempos que, en este caso, se han impuesto por default en las instrucciones del programa. Y cuya información se guarda en cada una de las memorias que incorpora esta plataforma (SRAM, EEPROM, FLASH).

-La realización del programa e implementación del circuito se la realizó en base a todo lo investigado, lo cual tiene que ver con las funcionalidades que ofrece el microcontrolador Arduino, principalmente con lo correspondiente a su Hardware y el funcionamiento de cada sensor utilizado para cada circuito que se ha implementado. Y es muy importante saber que la utilización de Arduino hoy en día es muy demandada en proyectos de electrónica, ya que tiene varias ventajas como, por ejemplo:

su bajo coste, y su versatilidad, ya que los usuarios pueden modificar los requerimientos a sus necesidades, además que, en nuestro caso, la programación se la realizó en Tinkercad con un entorno de desarrollo gráfico, lo que implica que es muy intuitiva y fácil de utilizar para cualquier tipo de usuario, incluso para los aprendices. De hecho, al momento de realizar la investigación con los artículos de los autores investigados, podemos darnos cuenta de la gran cantidad de aplicaciones que tiene un Arduino, junto con varios sensores y que de cierta manera se relaciona con nuestra investigación en cuestión.

### **RECOMENDACIONES**

Se recomienda realizar la investigación única y exclusivamente de los temas a ser tratados durante la ejecución y realización del informe que a su vez sustenten los fundamentos para la elaboración del algoritmo y su programación.

Implementar y declarar correctamente las variables necesarias del programa, de tal manera que permitan realizar el código de programación sin que exista ningún tipo de error al momento de ejecutar y compilar el programa.

Si no existe familiarización con la programación mediante interfaz textual, recomendamos realizar el algoritmo de programación en una interfaz gráfica, pues una de las ventajas que tiene Tinkercad es que podemos visualizar las dos interfaces al mismo tiempo. Y de esta manera comprender de manera intuitiva la realización del programa. Incluso, exportar el código generado, para implementarlo en un Arduino físico.

### **BIBLIOGRAFIA**

[1] Pereira Pires, R., Gracioli, G., Wanner, L. y Augusto Medeiros Frohlich, A. Evaluación de un algoritmo de ubicación basado en RSSI para redes de sensores inalámbricos, 16 de junio de 2011 Disponible:

<https://ieeexplore.ieee.org/document/5876427>

[2] González, R., Pérez, JM, Vela, O. y De Burgos, E. Comparación de rendimiento de un detector de semiconductores CZT de gran volumen y un detector de centelleo 28 de agosto de 2006, Disponible:

<https://ieeexplore.ieee.org/document/1684119>

[3] Yusof, HHM, Harun, SW, Dimiyati, K., Bora, T., Sterckx, K., Mohammed, WS y Dutta, J. Sensores de humedad integrados de bajo costo basados en nanobarras de óxido de zinc para la plataforma Arduino", 1 de abril de 2019 Disponible:

<https://ieeexplore.ieee.org/document/8574955>

Anónimo (20 abril, 2020) Sensor de inclinación obtenido de:

<https://descubrearduino.com/tilt-switch-o-sensor-de-inclinacion-que-es-y-para-que-sirve/>

Anónimo (6 de Marzo, 2019) Sensor de Temperatura obtenido de:

<https://srcsl.com/tipos-sensores-temperatura/>

330ohms (11 julio, 2016) Sensor de Gas recuperado de:

<https://blog.330ohms.com/2016/07/11/como-funcionan-los-sensores-de-gas/>