

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Procesos Tecnológicos e Industriales

Desarrollo tecnológico y generación de riqueza sustentable

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

Programa de Apoyo al Desarrollo Tecnológico de la Industria



**ITESO, Universidad
Jesuita de Guadalajara**

E06 Apoyo al Desarrollo Tecnológico de la Industria

Proyecto Material para capacitaciones en el uso de Arduino

PRESENTAN

Lic. en Ingeniería Mecánica. Sergio Hernández Reyes Retana

Lic. en Ingeniería Mecánica. Max Gerardo Toledo Rodríguez

Profesor PAP: David Manuel Ochoa González

Tlaquepaque, Jalisco, Mayo 2019

ÍNDICE

Contenido

Presentación Institucional de los Proyectos de Aplicación Profesional	2
Resumen	2
1. Introducción.....	3
1.1 Objetivos.....	3
1.2. Justificación.....	3
1.3 Antecedentes.....	4
1.4. Contexto	5
2. Desarrollo	6
2.1. Sustento teórico y metodológico	6
2.2. Planeación y seguimiento del proyecto	7
3. Resultados del trabajo profesional.....	8
4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto	8
5. Conclusiones.....	10
6. Bibliografía.....	10
Anexos.....	11
(A) Calendario	11
(B) Lluvia de ideas	11
(C) Análisis FODA de las ideas desarrolladas	12
(E) Factores a considerar del Proyecto Integrador	15
(F) Proyectos Propuestos para Arduino (Diagrmas de Talos Electronics, 2016):.....	16
(G) Glosario de términos:.....	26
(H) Archivos de programación.....	27
(I) Otras referencias útiles	37
(J) Catálogo de piezas necesarias:	38

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son una modalidad educativa del ITESO en la que el estudiante aplica sus saberes y competencias socio-profesionales para el desarrollo de un proyecto que plantea soluciones a problemas de entornos reales. Su espíritu está dirigido para que el estudiante ejerza su profesión mediante una perspectiva ética y socialmente responsable.

A través de las actividades realizadas en el PAP, se acreditan el servicio social y la opción terminal. Así, en este reporte se documentan las actividades que tuvieron lugar durante el desarrollo del proyecto, sus incidencias en el entorno, y las reflexiones y aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

El proyecto “Laboratorios de fabricación digital Nueva Santa María - ITESO para la inclusión en la industria 4.0 en zonas populares” tiene como objetivo crear y validar un modelo replicable para el desarrollo de las habilidades necesarias para trabajar dentro de las industrias que se están transformando con el modelo de industria 4.0 con el fin de que las personas que viven en zonas populares mejoren sus oportunidades de empleo o si son microempresarios o emprendedores puedan aumentar su competitividad. Este modelo integra la creación de laboratorios de fabricación digital y una metodología de capacitación —que comprende la fabricación digital, la automatización y la innovación— donde personas de comunidades en zonas populares que buscan mejorar sus habilidades técnicas para las industrias en transformación y aumentar la competitividad de microempresas y emprendedores locales colaboran con que estudiantes universitarios colaboran para este fin.

1. Introducción

1.1 Objetivos

Desarrollar y validar un modelo integral de operación de laboratorios de fabricación digital y de capacitación que pueda ser replicado en zonas populares para desarrollar las habilidades de sus miembros que busquen trabajar y competir en las industrias transformándose con el modelo de industria 4.0. 2. Desarrollo de una metodología de capacitación en habilidades necesarias para el modelo de la industria 4.0, incluyendo fabricación digital, automatización e innovación, enfocado en microempresas y emprendedores que busquen mejorar su competitividad, y personas interesadas en mejorar sus oportunidades de emplearse en empresas que requieran estas habilidades. 3. Implementar dos laboratorios de fabricación digital, uno localizado en la Nueva Santa María y otro en el ITESO, donde se llevarán a cabo las capacitaciones y se validará el modelo integral. 4. Crear un escenario de vinculación para que los estudiantes universitarios mejoren sus capacidades profesionales, en aplicación y transferencia de conocimientos, para aplicarlas en proyectos concretos de formación y mejora de competitividad usando fabricación digital con personas viviendo.

1.2. Justificación

La implementación de las Tecnologías de Fabricación Digital (TFD) se ha dado en diferentes niveles según el tamaño de las empresas o negocios. Así, tecnologías que llevan años utilizándose en grandes industrias a ser utilizados también por micro y pequeñas empresas. Sin embargo, aunque por un lado este cambio se presenta como una oportunidad de crecimiento, representa un reto en cuanto a tener empleados capacitados en el uso de estas tecnologías.

Esta situación es todavía más crítica para las personas de zonas populares, debido al limitado acceso que tienen a la educación media superior. Así, estas personas corren el riesgo a quedar fuera del mercado laboral, al no tener las habilidades necesarias para el uso de las tecnológicas.

En este contexto este Proyecto de Aplicación Profesional busca facilitar el acceso a capacitación y experiencias de uso de TFD a los habitantes de las colonias que se encuentran en el Cerro del Cuatro, en particular de La Nueva Santa María.

1.3 Antecedentes

Aunque algunas TFD existen desde hace más de 50 años, los avances en la computación y la apertura de los mercados internacionales han facilitado recientemente el acceso ellas. Además, la masificación del acceso a Internet y la mayor disponibilidad de equipos de cómputo, permiten que estas sean utilizadas por personas sin una formación formal.

Esta combinación de factores ha permitido que estas TFD sean utilizadas para desarrollar y fabricar productos, artículos de decoración y artesanías por parte de infinidad de personas. Este proceso de creación personalizada contrasta con los procesos de fabricación impulsados por los modelos tradicionales de la industria, enfocados en la producción por en masa.

Así, el avance tecnológico ha hecho que los equipos utilizados sean cada vez más amigables y por lo tanto las habilidades necesarias para el diseño y fabricación usando estas TFD requieren cada vez menos preparación, volviéndolas cada vez más accesibles. A pesar de esto, tener experiencia en la utilización de TFD es un diferenciador en el momento de acceder a puestos de trabajo, pues a pesar de que la tecnología es cada vez más amigable el acceso a los equipos sigue siendo exclusivo para ciertos sectores de la sociedad.

Existen diversos proyectos que buscan permitir el acceso a TFD a grupos o personas que por su situación socioeconómica no lo tienen. En este sentido destaca la experiencia del Barrio de Analco, en Puebla, que ha contado con la colaboración de la Universidad Iberoamericana de aquella ciudad. Este proyecto busca capacitar y fomentar el uso de estas tecnologías entre los artesanos de este barrio, permitiéndoles facilitar algunos de sus procesos productivos. En un contexto diferente, la Universidad de Guadalajara está estableciendo una red de laboratorios donde se capacitan estudiantes de todos los estratos socioeconómicos en el uso de estas tecnologías (Núñez, 2016). Destaca también

la experiencia de los “ateneos de fabricación”, proyecto de la municipalidad de Barcelona el cual consiste en la construcción de laboratorios de fabricación digital en diversos barrios populares de esta ciudad con el objetivo de empoderar a los ciudadanos en el uso de diversas tecnologías (Besson, 2018).

1.4. Contexto

En este contexto, el proyecto del Laboratorio de Fabricación Digital Nueva Santa María surgió a mediados del 2018 como un espacio donde las personas que viven en la zona del Cerro del Cuatro pudieran tener acceso a TFD. El laboratorio fue equipado con recursos del ITESO y del Consejo Estatal de Ciencia y Tecnología (COECYTJAL) a través del PROSOFTAJ. Los objetivos de este laboratorio son multidimensionales:

- A nivel personal: se busca que las personas se capaciten en el uso de las TFD, participando en procesos comunitarios y creativos. Esta capacitación se da tanto en cursos y talleres como en el uso mismo de las tecnologías durante el desarrollo de proyectos personales y de la comunidad. Se busca que las habilidades desarrolladas en el laboratorio sean transferibles a otros espacios, como pueden ser laborales o de auto empleo.
- A nivel comunitario: se busca fortalecer las actividades productivas de la comunidad, añadiendo una herramienta al abanico de habilidades de las personas y/o proyectos de emprendimiento. Así, este laboratorio puede ser utilizado como herramienta de apoyo a las iniciativas productivas de las personas, familias o de la comunidad, ofreciendo una serie de servicios y productos para ayudar a posicionarse mejor en el mercado.
- De una manera más amplia: se busca también construir un espacio de creación y convivencia seguro que sea una opción para personas de todas las edades para participar en la comunidad.

La creación de un espacio así representa un reto multidimensional: primeramente se busca que sea un espacio de formación, donde se promuevan los procesos creativos y las

soluciones novedosas; se busca, además, que los usuarios del laboratorio se apropien cada vez más del espacio y que este pueda funcionar como un espacio para crear nuevas relaciones o fortalecer relaciones existentes de los miembros de la comunidad; finalmente, se busca que el laboratorio sea auto sustentable económicamente, ya sea mediante donaciones, venta de productos o servicios o el cobro de capacitaciones y/o suscripciones.

De esta manera, los desarrollos presentados en este reporte buscan atender estos retos de una manera puntual, proponiendo soluciones que permitan a la comunidad fortalecer el laboratorio.

2. Desarrollo

2.1. Sustento teórico y metodológico

Para poder realizar este proyecto fue necesaria una gran organización por parte de todos los involucrados, debido a que al principio del proyecto se definió un objetivo común que era realizar unas capacitaciones para el FabLab que posteriormente se cambió a un apoyo a las capacitaciones para lo cual se realizaron métodos de evaluación, mantenimiento y sustentación del FabLab. Una vez terminado este proceso el grupo de trabajo se dividió para que existieran diferentes objetivos y poder realizar el mayor trabajo posible. El equipo encargado de mantener el proyecto sostenible económicamente desarrollo un método por el cual se pudiera generar dinero manteniendo el objetivo inicial del FabLab que era mantener las capacitaciones gratuitas para todos sus participantes. Otro equipo se dedicó a realizar un proyecto integrador que complementara el FabLab y las capacitaciones fueran más ricas en conocimientos para los participantes. Para realizar estos dos objetivos los distintos equipos hicieron modelos calendarizados con objetivos pequeños para tener la facilidad de llegar al objetivo final. Se definieron las actividades de mayor importancia por medio de una lluvia de ideas por parte de todos los participante y posteriormente un diagrama de Gant de las actividades.

2.2. Planeación y seguimiento del proyecto

- Descripción del proyecto

Los alumnos de ingeniería mecánica asistieron al FabLab original para poder realizar un análisis de lo necesario para poder mejorar y complementar el proyecto. Previo a definir actividades de mejora y complementación se asistió al programa como apoyo a las capacitaciones durante cinco semanas seguidas para poder tener una idea concreta de lo que eran todas las sesiones que se ofrecían a todos los participantes. En estas cinco sesiones solo se aportó una ayuda por parte de los alumnos para posteriormente definir las actividades.

- Plan de trabajo

Se definieron los equipos de trabajo y el grupo encargado del proyecto integrador definió estas las actividades que serían necesarias para poder realizar el proyecto integrador y poder incluirlo al FabLab.

- Investigación de diseños
- Selección de diseños
- Elaboración de guía de diseño
- Adquisición de materiales
- Fabricación de piezas
- Ensamble
- Programación
- Documentación
- Presentación

- Desarrollo de propuesta de mejora

Se propuso realizar el proyecto integrador para ofrecer un mayor conocimiento a los participantes del FabLab. Este proyecto integrador permite una introducción y capacitación mucho más alta a la industria 4.0 lo cual provoca una ventaja competitiva respecto a ofertas laborales y desarrollo personal para los participantes. Siendo un programa para niños, jóvenes y adultos; el proyecto integrador, abre las puertas a sus

participantes de poder obtener con mayor facilidad desde un trabajo o poder llegar a especializarse con el tema de la industria 4.0.

3. Resultados del trabajo profesional

El resultado más importante o como un mayor impacto fue el de un proyecto integrador que va de la mano con el proyecto inicial a la introducción e incorporación de las personas que asisten al programa para fomentar y educar sobre la industria 4.0. El proyecto integrador es un manual con una serie de ejercicios o lecciones para poder enseñar a las personas el uso de un Arduino y su alcance. El manual cuenta con una serie de clases y el costo que tendría para la institución poder llevar a cabo este proyecto integrador. No es fácil conocer el impacto que podrá tener debido a que el desarrollo o implementación del mismo será más adelante funcionando como una herramienta mucho más avanzada para los participantes del programa. El resultado estimado es una serie de conocimientos mucho más amplia en relación a lo que fue el planteamiento inicial del proyecto.

4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto

- Aprendizajes profesionales

En este proyecto se desarrollaron o mejor dicho se fortalecieron dos habilidades muy importantes. La primera fue el conocimiento alcance de un Arduino, debido a que esto no es precisamente de ingeniería mecánica pero puede ser utilizado como una herramienta y para poder hacer un curso sobre el tema se tiene que dominar lo que se está impartiendo en el mismo. Por lo cual se considera un gran aprendizaje al igual que el tema coaching, ya que un alumno está acostumbra a su proceso de aprendizaje y la asesoría a demás compañeros con procesos similares así que el método de compartir estos aprendizajes con personas cuyos procesos son muy diferente o que tienen menos conocimientos sobre el tema.

Aprendizajes sociales

El FabLab es un proyecto que puede ser de alto impacto, dependiendo la publicidad y seguimiento que se le dé. Puede transformar el alcance que tiene la industria 4.0 en México si se capacita de manera correcta a las personas participantes. Debido a que la industria manufacturera es una de las principales fuentes de empleos en México y la industria 4.0 es algo que está por detonar. Es imposible medir el apoyo que da este proyecto a las personas que asisten ya que es necesario de voluntad de los participantes para seguirse preparando y buscar opciones que complementen sus conocimientos aprendidos en el proyecto. Sin embargo, las personas de la comunidad del cerro del cuatro que asisten pueden desarrollar una capacidad de para incorporarse a la industria 4.0 con facilidad.

- Aprendizajes éticos

Cuando se empezó el proyecto no existía la idea de un proyecto integrador que pudiera complementar el FabLab. Este proyecto abre un mundo de puertas a los participantes y por la zona en la que está el FabLab, las personas de bajos recursos que asisten a las capacitaciones pueden usar la oportunidad para mejorar su estado laboral y mejorar el estado de la comunidad del cerro del cuatro. Los alumnos se llevan el aprender a compartir sus conocimientos adquiridos en la universidad con la comunidad, sabiendo que este conocimiento le podrá ayudar a una persona, su familia y a toda la comunidad.

- Aprendizajes en lo personal

Para poder desarrollar el proyecto integrador fue necesaria una investigación exhaustiva de programación de Arduinos y de los posibles proyectos que se pueden lograr con este equipo. Debido a que dentro de la ingeniería mecánica no existe una materia como tal de programación en el lenguaje c++ fue necesario aprender a programar en dicho lenguaje. La evaluación de los distintos proyectos que se llevaron a la tarea tanto de programar como elaborar en físico también fue todo una investigación para poder entender la manera de elaboración de la tarjeta. El principal aprendizaje fue el poder plasmar los

conocimientos adquiridos en un método de enseñanza que puedan entender los participantes del FabLab del cerro del cuatro.

5. Conclusiones

Se empezó a formular la propuesta de trabajar con Arduinos y protoboards para poder motivar a una audiencia en dirección de la electrónica como un complemento de las habilidades aprendidas en las capacitaciones pasadas. Creemos firmemente que motivando el autoaprendizaje y combinándolo con la creatividad se puede motivar lo suficiente para generar un impacto positivo entre los participantes. Así, creemos que creando procedimientos con información detallada (con una lista de materiales, un procedimiento y una demostración gráfica) puede ayudar a que las personas sean capaces de emprender sus propios proyectos. Sin embargo, existe una parte del proyecto que requiere de ciertas líneas de programación, las cuales consideramos que requieren ser impartidas en una sección completamente diferente de capacitaciones. En este sentido nosotros nos limitamos a colocar el código de fuente abierta de una liga proporcionada acorde al proyecto.

6. Bibliografía

Besson, R. (2018). Les «Ateneus de Fabricació» barcelonais et les «Laboratorios ciudadanos» madrilènes. Une nouvelle approche de l'innovation urbaine? *Géographie, Économie, Société*, 20(1), 113–141.

Fablab Analco | FabLabs. (6 abril de 2018). Recuperado de <https://www.fablabs.io/labs/fablabanalco>

Núñez, M. E. C. (2016). La virtualización de la educación superior en América Latina: entre tendencias y paradigmas. *Revista de Educación a Distancia*, (48).

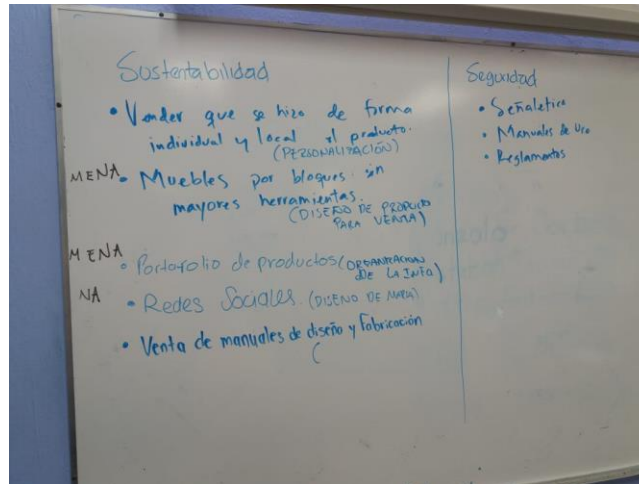
Talos Electronics. (2016). Tutorial Kit 136 Elementos. 2019, de Talos Electronics Sitio web: <https://www.taloselectronics.com/blogs/tutoriales/kit-136-elementos>

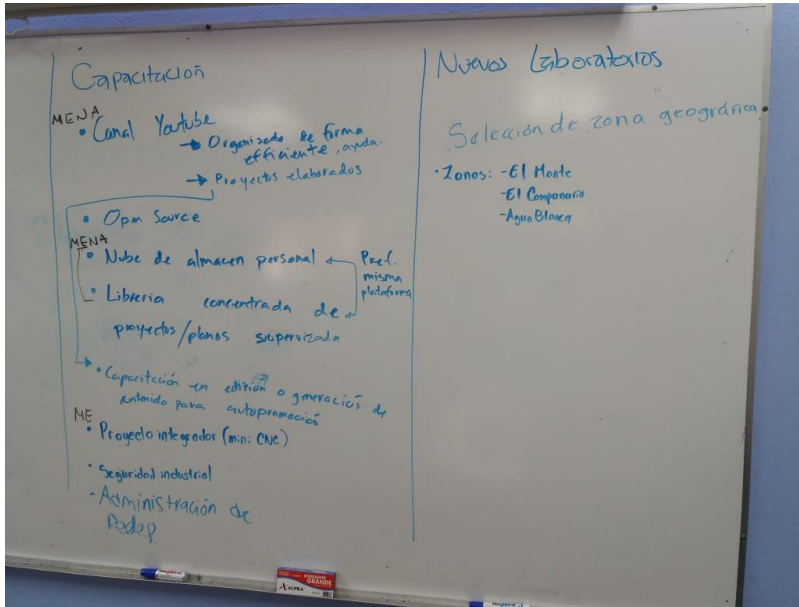
Anexos

(A) Calendario

PROYECTO INTEGRADOR							
	Semana 9	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15
Etapa 1	Investigación de Diseños						
	Selección del Diseño						
	Elaboración de Guía de Procedimiento						
Etapa 2	Adquisición de Materiales						
	Fabricación de Piezas						
	Ensamble						
Etapa 3	Programación						
	Documentación						
	Presentación						

(B) Lluvia de ideas





(C) Análisis FODA de las ideas desarrolladas

I.- YouTube (Material de capacitación en video)

<p>Fortalezas</p> <ul style="list-style-type: none"> • Accesibilidad asegurada • Capacidades de edición • Conocimiento de software/tecnología • Mientras haya capacitaciones se cumple la función 	<p>Debilidades</p> <ul style="list-style-type: none"> • Limitado debido al acceso a internet • Edición de video • Manejo de redes sociales • Seguimiento en el futuro y administración del mismo
<p>Oportunidades</p> <ul style="list-style-type: none"> • Acceso en el templo • Visibilidad y acceso más allá de la comunidad • Patrocinio • Canal de venta / Promoción 	<p>Amenazas</p> <ul style="list-style-type: none"> • No lograr que sea realmente accesible • Problema legal • Comentarios Negativos

II.- Desarrollo de productos (Muebles por bloques sin mayores herramientas y portafolio de productos)

Fortalezas <ul style="list-style-type: none"> • 5 Ing. Mecánicos • Uso de Software de diseño • Disponibilidad de maquinas • Administración (Materiales, proceso, etc) hecha por nosotros • Análisis económico 	Debilidades <ul style="list-style-type: none"> • Seguimiento después del PAP
Oportunidades <ul style="list-style-type: none"> • Proyección • Publicidad al proyecto • \$\$\$ 	Amenazas <ul style="list-style-type: none"> • Falta de interés (Ventas) • Plagio o piratería

III.- Redes Sociales

Fortalezas <ul style="list-style-type: none"> • Apoyo institucional 	Debilidades <ul style="list-style-type: none"> • Falta de conocimiento •
Oportunidades <ul style="list-style-type: none"> • Es gratis • Visibilidad al proyecto • Herramientas muy accesibles • Tienda 	Amenazas <ul style="list-style-type: none"> • Falta de contenido • Continuidad

IV.- Librería (Uso de internet para obtener y almacenar diseños y/o proyectos)

Fortalezas <ul style="list-style-type: none"> • Control de lo que se sube • Conocemos y dominamos las plataformas • División Lab/Personal 	Debilidades <ul style="list-style-type: none"> • Límite de capacidad
Oportunidades <ul style="list-style-type: none"> • Herramientas libres y gratuitas 	Amenazas <ul style="list-style-type: none"> • Guardar cosas que no sirvan • Acceso a internet • Mal uso de Internet

V.- Proyecto Integrador

Fortalezas <ul style="list-style-type: none"> • Ingenieros Mecánicos • Producir materiales para dar continuidad • Diseño para la máquina que tenemos 	Debilidades <ul style="list-style-type: none"> • Nivel previo de conocimiento • Costo de materiales • Estudio de algunas tecnologías
Oportunidades <ul style="list-style-type: none"> • Dar una visión de más tecnología • Aprender • Motivación • Vender como taller \$\$ • Venderlo como producto y meterlo a portafolio 	Amenazas <ul style="list-style-type: none"> • Que tome mucho tiempo • Perder el foco del proyecto • Menos interesados

VI.- Habilitar Horarios

Fortalezas <ul style="list-style-type: none"> • 5 Ing. Mecánicos • Posibilidad de armar ventanas • Comunicación (Whatsapp) • Hacer junto con otras actividades 	Debilidades <ul style="list-style-type: none"> • Tiempos • Software
Oportunidades <ul style="list-style-type: none"> • Flujo de gente • Tiempo extra • Enfoque personalizado 	Amenazas <ul style="list-style-type: none"> • Inseguridad • Poca gente

(E) Factores a considerar del Proyecto Integrador

Factores Importantes:

- Costo
- Factibilidad
- Piloto

Posibles ejemplos en el mercado:

1. Ventilador cortando aspas con el router programable.
2. Lámparas que cambian de color
3. Scribit
4. Shaper Origin

Propuestas Avanzadas

1. Motor a pasos controlado mediante joystick y una banda.
2. Cubo de 64 Leds
3. Juego de contacto niño chiquito
4. Sensor de Alarma
5. Semáforo
6. Pantalla de Leds gigante

Prerrequisitos:

1. Entender cómo funciona un protoboard
2. Entender cómo funciona un Arduino
3. Entender cómo se introduce programación al Arduino

(F) Proyectos Propuestos para Arduino (Diagramas de Talos Electronics, 2016):

1. Zumbador Activo/ Pasivo

Un zumbador activo es capaz de producir un zumbido cuando se alimenta con 5V, el cual tiene diversas utilidades como alarma, el tiempo que dura encendido y apagado es programable.

Un zumbador pasivo o altavoz, son dispositivos que permiten convertir una señal eléctrica a una onda de sonido. Sin embargo, no disponen de electrónica interna por ende tenemos que proporcionar una señal eléctrica para conseguir el sonido deseado. Al proporcionar y controlar la señal eléctrica del zumbador pasivo y los altavoces tenemos la ventaja de poder variar el tono emitido modificando la señal aplicada y así podemos crear diferentes tonalidades.

Los materiales requeridos son los siguientes:

- Arduino
- Protoboard
- Zumbador activo/pasivo
- Cables

Para armar el circuito físico debemos seguir los siguientes pasos:

1. Introducir programación al Arduino.
2. Armar el circuito de acuerdo a los diagramas.

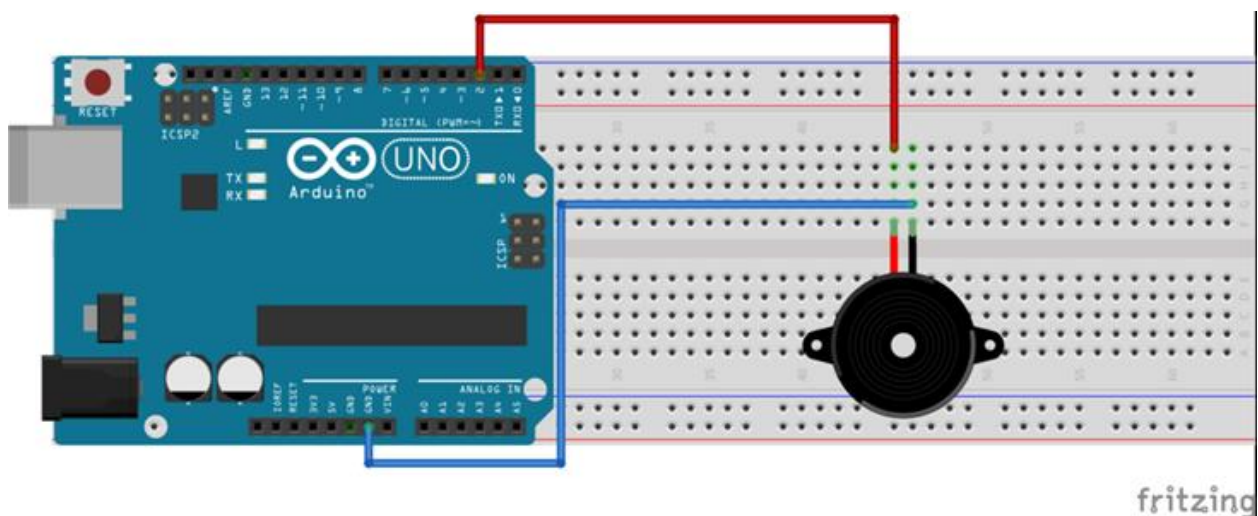


Fig. # 1: Diagrama de conexión de un zumbador activo básico

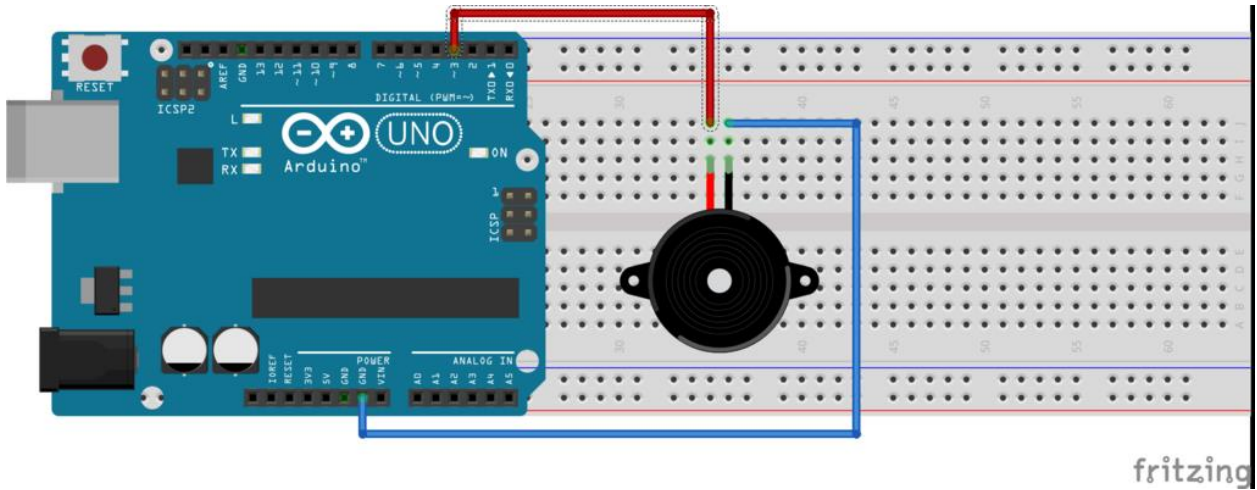


Fig. # 2: Diagrama de conexión de un zumbador pasivo básico

2. Semáforo

Este proyecto es utilizado como una pequeña muestra de lo que puede ser capaz de hacer un Arduino con programación básica.

Los materiales requeridos son los siguientes:

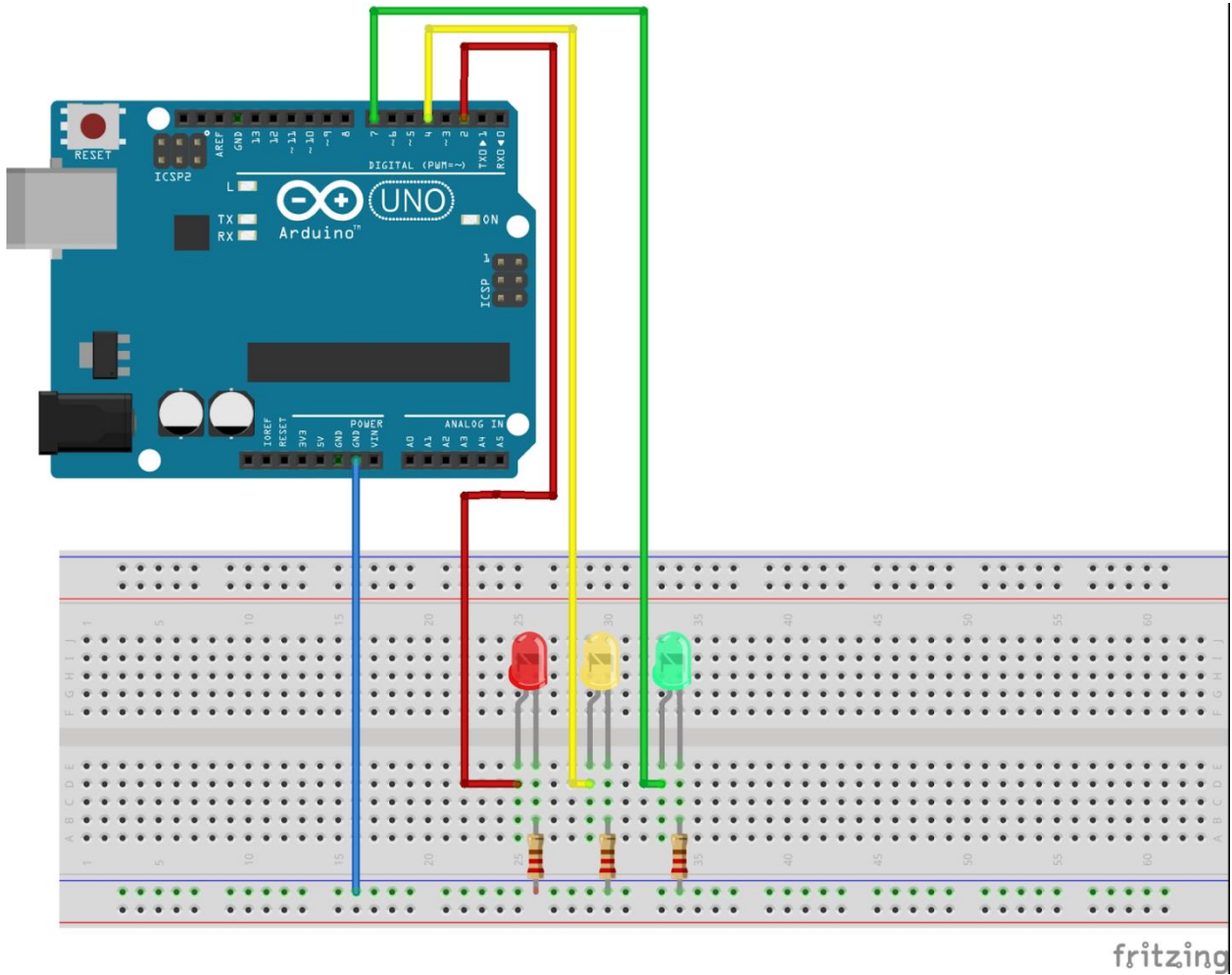
- Arduino
- Protoboard
- Leds x 3
- Resistencia 220 Ohm x 3
- Cables

Para armar el circuito físico debemos seguir los siguientes pasos:

1. Conectar led rojo en fila #10 "E" y #11 "E", y resistencia de #11 "B" a tierra del protoboard.
2. Conectar led amarillo en fila #13 "E" y #14 "E" y resistencia de #14 "B" a tierra del protoboard.
3. Conectar led verde en fila #16 "E" y #17 "E" y resistencia de #17 "B" a tierra del protoboard.
4. Conectar desde Arduino al protoboard de la siguiente manera:

Arduino	Protoboard
7 (Digital)	10 B
4 (Digital)	13 B
2 (Digital)	16 B
GND (Power)	Tierra (-)

5. Introducir el código al Arduino.



fritzing

Fig. # 3: Diagrama de conexión del semáforo.

3. Display de 7 Segmentos

Durante la práctica se utiliza un cátodo común el cual es un conjunto simple de leds el cual a través de programación y alimentación eléctrica de forma adecuada pueden encender varios segmentos para formar luminiscencia que formen números del cero al nueve. El programa mostrará en el display la cuenta del cero al nueve y se reiniciará al llegar al nueve con el contador automáticamente.

El arreglo interno de los leds es el siguiente para un cátodo común:

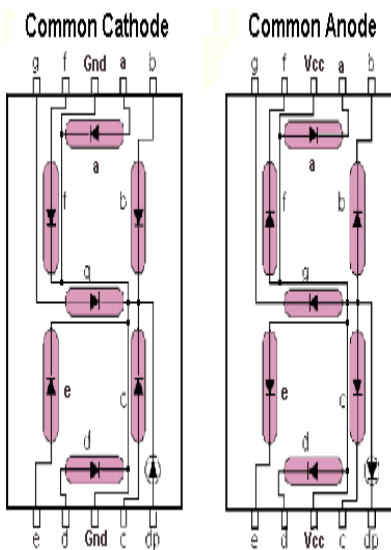


Fig. # 4: Diagrama del funcionamiento interno de un cátodo común.

Los materiales requeridos son los siguientes:

- Protoboard
- Arduino
- Display 7 segmentos (Catodo Comun)
- Resistencias de 1 (Kilo Ohm) x 7
- Cables requeridos

Para armar el circuito físico debemos seguir los siguientes pasos:

1. Conectar las resistencias al protoboard empezando de la fila 10 columna "E", a la fila 10 columna "F", posteriormente de la fila 11 y así sucesivamente hasta conectar las siete resistencias.
2. Conectar los cables de las salidas digitales del Arduino al protoboard, empezando por el #2 del Arduino y la fila #10 del protoboard hasta terminar de cubrir todas las filas que contienen las resistencias.
3. Conectar el cátodo de la fila #20 al #25 del protoboard.

- Conectar las salidas de la resistencia hacia el cátodo con el siguiente arreglo.

Inicio	Fin
10 A (Rojo)	23 J
11 B (Amarillo)	24 i
12 C (Azul)	23 A
13 A (Verde)	21 A
14 B (Naranja)	20 B
15 C (Morado)	21 J
16 A (Negro)	20 i

- Conectar de la sección de poder del Arduino la parte de tierra a la tierra del protoboard (-).
- Conectar del 22 B a la tierra del protoboard (-) .
- Introducir el código al Arduino.

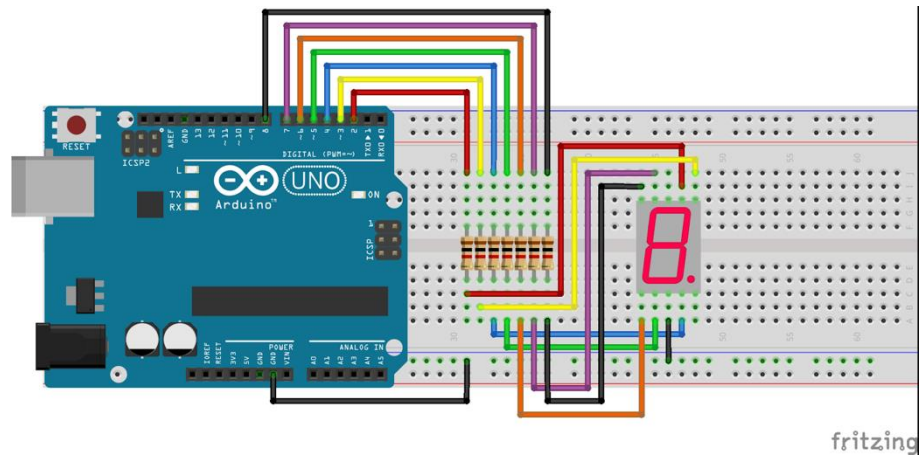


Fig. # 5: Diagrama de conexión del display de 7 segmentos.

4. Sensor de temperatura LM35 y led RGB

Durante la práctica utilizaremos un sensor de temperatura y un led el cual es la unión de tres leds de los colores básicos encapsulados compartiendo el cátodo. La salida del sensor es lineal con la temperatura, incrementando el valor a razón de 10 mv por cada grado centígrado, el rango de medición es de -55 a 150 grados centígrados con una precisión es de 0.5°.

El programa medirá la temperatura y encenderá el led de acuerdo a la lectura correspondiente.

1. Si $T \leq T_{min}$, encenderá de color azul.
2. Si $T_{max} > T > T_{min}$, encenderá verde.
3. Si $T > T_{max}$, encenderá rojo.

Se crean variables, podemos modificar los parámetros de temperaturas y se definen los pines como salidas.

Los materiales requeridos son los siguientes:

- Protoboard
- Arduino
- Led RGB
- Sensor de temperatura LM35
- Resistencias de 1 (Kilo Ohm) x 3
- Cables requeridos

Para armar el circuito físico debemos seguir los siguientes pasos:

1. Conectar el sensor de temperatura en la fila #10 a la #12 "E".
2. Conectar las resistencias al protoboard empezando de la fila 20 columna "E", a la fila 20 columna "F", posteriormente de la fila 21 y así sucesivamente hasta conectar las tres resistencias.
3. Identificar la "patita" que corresponde a tierra y conectar desde la columna #20 "J" del protoboard de forma consecutiva hasta conectar tierra en la columna #23 o en alguna columna que no contenga una resistencia.
4. Conectar desde Arduino al protoboard de la siguiente manera:

Arduino	Protoboard
2 (Digital)	22 A
3 (Digital)	21 B

4 (Digital)	20 A
5V (Power)	10 A
GND (Power)	12 A
GND #2 (Power)	23 i
A0 (Analog)	11 B

5. Introducir el código al Arduino.

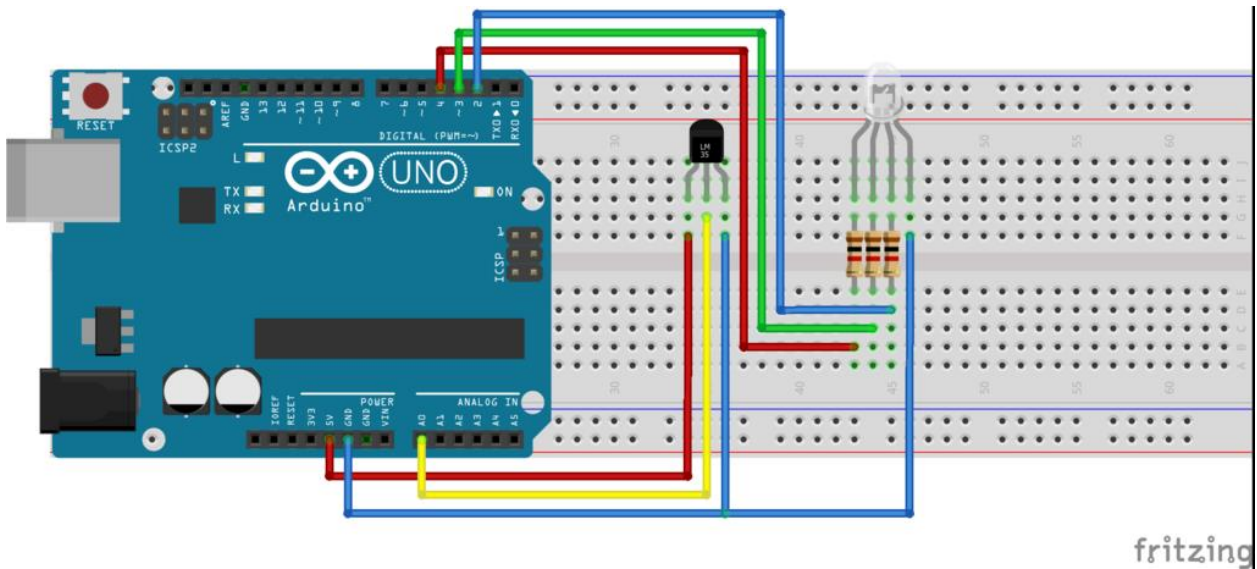


Fig. # 6: Diagrama de conexión del sensor de temperatura y el led RGB.

5. Receptor Infrarrojo

Durante la práctica utilizaremos un receptor infrarrojo y con la ayuda de una librería crearemos una función que controla cuántas veces se debe apagar y prender el led de acuerdo a la señal que es enviada a través de nuestro control.

Lista de material:

- Arduino
- Protoboard
- Receptor infrarrojo
- Control
- Resistencia 1 (Kilo Ohm) x1
- Led
- Cables

Para armar el circuito físico debemos seguir los siguientes pasos:

1. Conectar el sensor infrarrojo en la fila #10 "A" hasta la #12 "A".
2. Conectar la resistencia de la fila #20 "C" a la 24 "C".
3. Conectar el led de la fila #24 "D" a 25 "D".
4. Conectar un cable de la fila #11 "E" a #25 "E"
5. Conectar desde Arduino al protoboard de la siguiente manera:

Arduino	Protoboard
11 (Digital)	10 B
2 (Digital)	20 D
5V (Power)	12 B
GND (Power)	11 C

6. Introducir el código al Arduino.

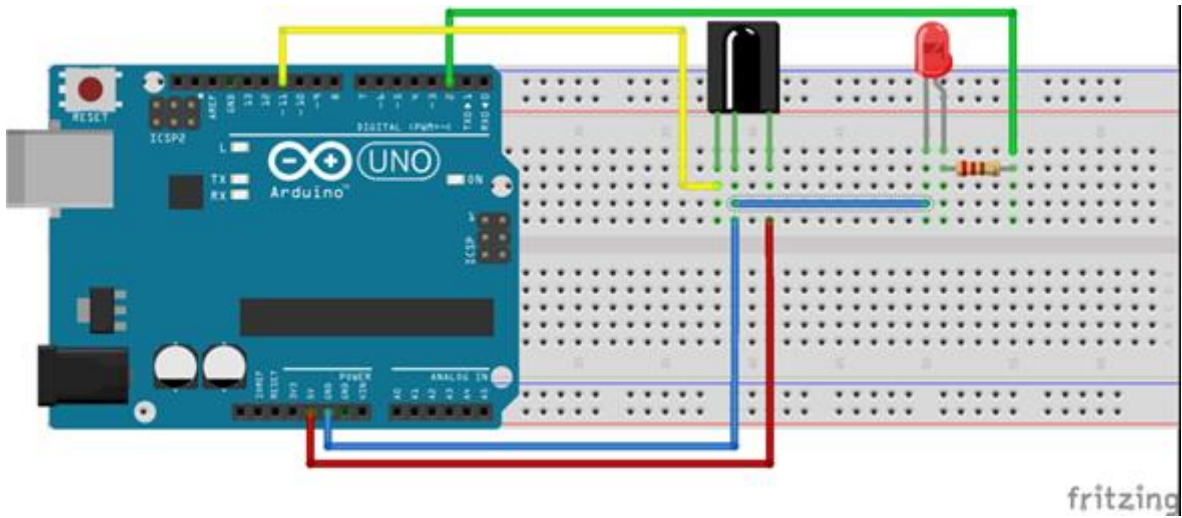


Fig. #7: Diagrama de conexión del sensor de Infrarrojo.

(G) Glosario de términos:

Arduino:

Un **Arduino** es un placa desarrolladora de bajo consumo y alta capacidad de operación con softwares multiplataforma accesible el cual tiene un entorno de programación simple, duplicables y de fuente abierta. Se le pueden añadir módulos de forma simple y existen amplias librerías listas para utilizar.

La placa se compone de diferentes elementos básicos:

Conexiones de entrada: A través de pines del Arduino puede recibir datos del exterior mediante sensores.

Microcontrolador: Es el cerebro del Arduino el cual interpreta información de los datos recibidos del entorno para tomar una acción programable.

Conexiones de Salida: Dependiendo del arreglo pueden conectarse diversos actuadores y sistemas lógicos para provocar una respuesta deseada.

Puertos de comunicación: Sirven para conectarse de forma directa, así como ingresarle la programación deseada.

La diversidad, las librerías, y los proyectos creados para los Arduinos son increíblemente bastos y por ende nos apoyamos en este.

Protoboard:

Un **Protoboard** es una placa que posee orificios conectados eléctricamente entre sí siguiendo un patrón vertical en las orillas y un patrón horizontal en el centro el cual se emplea usualmente para hacer pruebas de funcionamiento para circuitos electrónicos.

Sensor:

Un **sensor** es un dispositivo que capta magnitudes físicas tales como temperatura, variación de luz, sonido y demás, lo convierte a un equivalente eléctrico el cual puede ser procesado y transferido para recabar información.

(H) Archivos de programación

Buzzer Activo

https://github.com/TalosElectronics1/Kit_136_Elementos/blob/master/P4_Buzzer_Activo/P4_Buzzer_Activo.ino

```
/*
 * Talos Electronics
 * 3 de octubre de 2016
 */
//Declaracion de variables constantes
const int Buzzer=2;
const int Tiempo_Alto=500;
const int Tiempo_Bajo=1000;

void setup()
{
  // Inicializar el pin 2 como salida
  pinMode(Buzzer,OUTPUT);
}

void loop()
{
  digitalWrite(Buzzer,HIGH);//Encender el buzzer
  delay(Tiempo_Alto);//Esperar el tiempo programado
  digitalWrite(Buzzer,LOW);//Apagar el buzzer
  delay(Tiempo_Bajo);//Esperar el tiempo programado
}
```

Buzzer Pasivo

https://github.com/TalosElectronics1/Kit_136_Elementos/tree/master/P5_Buzzer_Pasivo/P5_Buzzer_pasivo

```
const int pinBuzzer = 3;

void setup()
{
}

void loop()
{
  //generar tono de 440Hz durante 1000 ms
  tone(pinBuzzer, 440);
  delay(1000);

  //detener tono durante 500ms
  noTone(pinBuzzer);
  delay(500);

  //generar tono de 523Hz durante 500ms, y detenerlo durante 500ms.
  tone(pinBuzzer, 523, 300);
  delay(500);
}

//Activa un tono de frecuencia determinada en un pin dado
tone(pin, frecuencia);
//Detiene el tono en el pin
noTone(pin);
```

Extra Especial

https://github.com/TalosElectronics1/Kit_136_Elementos/tree/master/P5_Buzzer_Pasivo/P5_Buzzer_Pasivo_SuperMario

```
/*
Arduino Mario Bros Tunes
With Piezo Buzzer and PWM

Connect the positive side of the Buzzer to pin 3,
then the negative side to a 1k ohm resistor. Connect
the other side of the 1 k ohm resistor to
ground(GND) pin on the Arduino.

by: Dipto Pratyaksa
last updated: 31/3/13
*/

/*****
 * Public Constants
 *****/
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
```

```

#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978

#define melodyPin 3
//Mario main theme melody
int melody[] = {
    NOTE_E7, NOTE_E7, 0, NOTE_E7,
    0, NOTE_C7, NOTE_E7, 0,
    NOTE_G7, 0, 0, 0,
    NOTE_G6, 0, 0, 0,

    NOTE_C7, 0, 0, NOTE_G6,
    0, 0, NOTE_E6, 0,
    0, NOTE_A6, 0, NOTE_B6,
    0, NOTE_AS6, NOTE_A6, 0,

    NOTE_G6, NOTE_E7, NOTE_G7,
    NOTE_A7, 0, NOTE_F7, NOTE_G7,
    0, NOTE_E7, 0, NOTE_C7,
    NOTE_D7, NOTE_B6, 0, 0,

    NOTE_C7, 0, 0, NOTE_G6,
    0, 0, NOTE_E6, 0,
    0, NOTE_A6, 0, NOTE_B6,
    0, NOTE_AS6, NOTE_A6, 0,

    NOTE_G6, NOTE_E7, NOTE_G7,
    NOTE_A7, 0, NOTE_F7, NOTE_G7,
    0, NOTE_E7, 0, NOTE_C7,
    NOTE_D7, NOTE_B6, 0, 0
};

```

```

//Mario main them tempo
int tempo[] = {
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,

    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,

    9, 9, 9,
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,

    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,

    9, 9, 9,
    12, 12, 12, 12,
    12, 12, 12, 12,
    12, 12, 12, 12,
};
//Underworld melody
int underworld_melody[] = {
    NOTE_C4, NOTE_C5, NOTE_A3, NOTE_A4,
    NOTE_AS3, NOTE_AS4, 0,
    0,
    NOTE_C4, NOTE_C5, NOTE_A3, NOTE_A4,
    NOTE_AS3, NOTE_AS4, 0,
    0,
    NOTE_F3, NOTE_F4, NOTE_D3, NOTE_D4,
    NOTE_DS3, NOTE_DS4, 0,
    0,
    NOTE_F3, NOTE_F4, NOTE_D3, NOTE_D4,
    NOTE_DS3, NOTE_DS4, 0,
    0, NOTE_DS4, NOTE_CS4, NOTE_D4,
    NOTE_CS4, NOTE_DS4,
    NOTE_DS4, NOTE_GS3,
    NOTE_G3, NOTE_CS4,
    NOTE_C4, NOTE_FS4, NOTE_F4, NOTE_E3, NOTE_AS4, NOTE_A4,
    NOTE_GS4, NOTE_DS4, NOTE_B3,
    NOTE_AS3, NOTE_A3, NOTE_GS3,
    0, 0, 0
};
//Underworld tempo
int underworld_tempo[] = {
    12, 12, 12, 12,
    12, 12, 6,
    3,
    12, 12, 12, 12,
    12, 12, 6,
    3,
    12, 12, 12, 12,
    12, 12, 6,
    3,
    12, 12, 12, 12,
    12, 12, 6,
    6, 18, 18, 18,
    6, 6,
    6, 6,
    6, 6,
    18, 18, 18, 18, 18, 18,
    10, 10, 10,
    10, 10, 10,
    3, 3, 3
};

```

```

void setup(void)
{
  pinMode(3, OUTPUT);//buzzer
  pinMode(13, OUTPUT);//led indicator when singing a note
}
void loop()
{
  //sing the tunes
  sing(1);
  sing(1);
  sing(2);
}
int song = 0;

void sing(int s) {
  // iterate over the notes of the melody:
  song = s;
  if (song == 2) {
    Serial.println(" 'Underworld Theme'");
    int size = sizeof(underworld_melody) / sizeof(int);
    for (int thisNote = 0; thisNote < size; thisNote++) {

      // to calculate the note duration, take one second
      // divided by the note type.
      //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
      int noteDuration = 1000 / underworld_tempo[thisNote];

      buzz(melodyPin, underworld_melody[thisNote], noteDuration);

      // to distinguish the notes, set a minimum time between them.
      // the note's duration + 30% seems to work well:
      int pauseBetweenNotes = noteDuration * 1.30;
      delay(pauseBetweenNotes);

      // stop the tone playing:
      buzz(melodyPin, 0, noteDuration);

    }
  } else {
    Serial.println(" 'Mario Theme'");
    int size = sizeof(melody) / sizeof(int);
    for (int thisNote = 0; thisNote < size; thisNote++) {

      // to calculate the note duration, take one second
      // divided by the note type.
      //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
      int noteDuration = 1000 / tempo[thisNote];

      buzz(melodyPin, melody[thisNote], noteDuration);

      // to distinguish the notes, set a minimum time between them.
      // the note's duration + 30% seems to work well:
      int pauseBetweenNotes = noteDuration * 1.30;
      delay(pauseBetweenNotes);

      // stop the tone playing:
      buzz(melodyPin, 0, noteDuration);

    }
  }
}

void buzz(int targetPin, long frequency, long length) {
  digitalWrite(13, HIGH);
  long delayValue = 1000000 / frequency / 2; // calculate the delay value between
  transitions
  //// 1 second's worth of microseconds, divided by the frequency, then split in

```

```

half since
  /// there are two phases to each cycle
  long numCycles = frequency * length / 1000; // calculate the number of cycles
for proper timing
  /// multiply frequency, which is really cycles per second, by the number of
seconds to
  /// get the total number of cycles to produce
  for (long i = 0; i < numCycles; i++) { // for the calculated length of time...
    digitalWrite(targetPin, HIGH); // write the buzzer pin high to push out the
diaphragm
    delayMicroseconds(delayValue); // wait for the calculated delay value
    digitalWrite(targetPin, LOW); // write the buzzer pin low to pull back the
diaphragm
    delayMicroseconds(delayValue); // wait again on the calculated delay value
  }
  digitalWrite(13, LOW);
}
}

```

7 Segmentos

https://github.com/TalosElectronics1/Kit_136_Elementos/blob/master/P3_Display%2007%20segmentos/P3_Display_7_segmentos/P3_Display_7_segmentos.ino

```

/*
 * Talos Electronics
 * 3 de octubre de 2016
 */

//Declaracion variables constantes
const int a = 2; // segmentos del display
const int b = 3;
const int c = 4;
const int d = 5;
const int e = 6;
const int f = 7;
const int g = 8;

//Declaracion de variables que seran modificadas
int Numero=0;
void setup()
{
  //Inicializar los segmentos como salidas
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
}

void loop()
{
  switch(Numero)
  {
    //Mostrar un 0 en el display
    case 0:
      digitalWrite(a,HIGH);
      digitalWrite(b,HIGH);
      digitalWrite(c,HIGH);
      digitalWrite(d,HIGH);
      digitalWrite(e,HIGH);
      digitalWrite(f,HIGH);
      digitalWrite(g,LOW);
      break;

    //Mostrar un 1 en el display

```

```

case 1:
digitalwrite(a, LOW); // 1
digitalwrite(b, HIGH);
digitalwrite(c, HIGH);
digitalwrite(d, LOW);
digitalwrite(e, LOW);
digitalwrite(f, LOW);
digitalwrite(g, LOW);
break;

//Mostrar un 2 en el display
case 2:
digitalwrite(a, HIGH);
digitalwrite(b, HIGH);
digitalwrite(c, LOW);
digitalwrite(d, HIGH);
digitalwrite(e, HIGH);
digitalwrite(f, LOW);
digitalwrite(g, HIGH);
break;

//Mostrar un 3 en el display
case 3:
digitalwrite(a, HIGH);
digitalwrite(b, HIGH);
digitalwrite(c, HIGH);
digitalwrite(d, HIGH);
digitalwrite(e, LOW);
digitalwrite(f, LOW);
digitalwrite(g, HIGH);
break;

//Mostrar un 4 en el display
case 4:
digitalwrite(a, LOW);
digitalwrite(b, HIGH);
digitalwrite(c, HIGH);
digitalwrite(d, LOW);
digitalwrite(e, LOW);
digitalwrite(f, HIGH);
digitalwrite(g, HIGH);
break;

//Mostrar un 5 en el display
case 5:
digitalwrite(a, HIGH);
digitalwrite(b, LOW);
digitalwrite(c, HIGH);
digitalwrite(d, HIGH);
digitalwrite(e, LOW);
digitalwrite(f, HIGH);
digitalwrite(g, HIGH);
break;

//Mostrar un 6 en el display
case 6:
digitalwrite(a, HIGH);
digitalwrite(b, LOW);
digitalwrite(c, HIGH);
digitalwrite(d, HIGH);
digitalwrite(e, HIGH);
digitalwrite(f, HIGH);
digitalwrite(g, HIGH);
break;

//Mostrar un 7 en el display
case 7:
digitalwrite(a, HIGH);
digitalwrite(b, HIGH);
digitalwrite(c, HIGH);
digitalwrite(d, LOW);

```

```

digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
break;

//Mostrar un 8 en el display
case 8:
digitalWrite(a, HIGH);
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
break;

//Mostrar un 9 en el display
case 9:
digitalWrite(a, HIGH);
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
break;
}
delay(1000); //Esperar 1 segundo
Numero++; //incrementar la cuenta

//Resetear el contador cuando llegue a 10
if(Numero==10)
{
Numero=0;
}
}

```

Sensor de Temperatura

https://github.com/TalosElectronics1/Kit_136_Elementos/blob/master/P7_LM35_RGB/P7_LM35_RGB/P7_LM35_RGB.ino

```

/*
 * Talos Electronics
 */
const int LM35 = A0;
const int R=4;
const int G=3;
const int B=2;
const int Temperatura_Minima=20;
const int Temperatura_Maxima=25;

float Temperatura;
void setup ()
{
Serial.begin(9600); //Inicializar puerto serial
pinMode(R,OUTPUT); //Inicializar el pin 4 como salida
pinMode(G,OUTPUT); //Inicializar el pin 3 como salida
pinMode(B,OUTPUT); //Inicializar el pin 2 como salida
//El led inicie apagado
digitalWrite(R,LOW);
digitalWrite(G,LOW);
digitalWrite(B,LOW);
}
void loop()
{
Temperatura = analogRead(LM35); //Leer la entrada analogica
}

```

```

//Calcular la temperatura
Temperatura = 5.0*Temperatura*100.0/1024.0;
//Mostrar la temperatura en el monitor Serial
Serial.print(Temperatura);
Serial.println(" °C");

//Codicion para que encienda el led de color Verde
if(Temperatura>Temperatura_Minima&&Temperatura<Temperatura_Maxima)
{
  digitalWrite(R,LOW);
  digitalWrite(G,HIGH);
  digitalWrite(B,LOW);
}

//Condicion para que encienda el led de color Azul
if(Temperatura<=Temperatura_Minima)
{
  digitalWrite(R,LOW);
  digitalWrite(G,LOW);
  digitalWrite(B,HIGH);
}

//Condicion para que encienda el led de color Rojo
if(Temperatura>=Temperatura_Maxima)
{
  digitalWrite(R,HIGH);
  digitalWrite(G,LOW);
  digitalWrite(B,LOW);
}
delay(500);
}

```

Receptor Infrarrojo

https://github.com/TalosElectronics1/Kit_136_Elementos/blob/master/P10_Receptor_IR/P10_Receptor_IR/P10_Receptor_IR.ino

```

#include <IRremote.h> //Libreria para IR
//Declar los pines como variables constantes
const int RECV_PIN = 11;
const int parpadeo = 0;
const int Led = 2;

void Blink(int numero_veces);//Funcion para prender y apagar el led

//Funciones para la libreria IR
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup()
{
  Serial.begin(9600);//Inicializar el puerto serial a 9600
  pinMode(Led, OUTPUT);//Inicializar el pin 2 como salida
  irrecv.enableIRIn(); // Inicializar receptor IR
}

void loop()
{
  if (irrecv.decode(&results))
  {
    Serial.println(results.value); //Descomentar esta linea para observar el
    codigo de cualquier control IR
    irrecv.resume(); // Receive the next value

    if (results.value == 16724175) Blink(1);//Si se presiona el boton 1 que
    parpade 1 vez
    if (results.value == 16718055) Blink(2);
    if (results.value == 16743045) Blink(3);
    if (results.value == 16716015) Blink(4);
  }
}

```

```

    if (results.value == 16726215) Blink(5);
    if (results.value == 16734885) Blink(6);
    if (results.value == 16728765) Blink(7);
    if (results.value == 16730805) Blink(8);
    if (results.value == 16732845) Blink(9);
  }
}

void Blink(int numero_veces)//Cuerpo de la funcion
{
  for (int i = 0; i < numero_veces; i++)
  {
    digitalWrite(Led, HIGH);
    delay(250);
    digitalWrite(Led, LOW);
    delay(250);
  }
}

```

Semaforo

```

/**/ Definiciones **/
int rojo=2;      //definimos el valor del pin para el led rojo
int amarillo=4;  //definimos el valor del pin para el led amarillo
int verde=7;     //definimos el valor del pin para el led verde

/**/ Programa **/

void setup() {
  pinMode(verde,OUTPUT); //declaramos el pin verde como salida
  pinMode(amarillo,OUTPUT); //declaramos el pin amarillo como salida
  pinMode(rojo,OUTPUT); //declaramos el pin rojo como salida
}

void loop() {
  digitalWrite(verde,HIGH); //encendemos el led rojo
  delay(2000); //esperamos 2 segundos
  digitalWrite(verde,LOW); //apagamos el led rojo
  delay(500); //esperamos medio segundo

  digitalWrite(amarillo,HIGH); //encendemos el led amarillo
  delay(2000); //esperamos 2 segundos
  digitalWrite(amarillo,LOW); //apagamos el led amarillo
  delay(500); //esperamos medio segundo

  digitalWrite(rojo,HIGH); //encendemos el led verde
  delay(2000); //esperamos 2 segundos
  digitalWrite(rojo,LOW); //apagamos el led verde
  delay(500); //esperamos medio segundo
}

```

(I) Otras referencias útiles

Tutoriales

<https://www.youtube.com/watch?v=anNPq1kbUok> (Protoboard)

<https://learn.adafruit.com/series/learn-Arduino>

<https://hackaday.io/project/7562-collection-of-Arduino-tutorials>

<https://www.element14.com/community/community/Arduino/Arduino-tutorials/blog/2014/09/04/jeremy-blum-Arduino-tutorials>

<https://Arduinobasics.blogspot.com/p/Arduino-basics-projects-page.html>

Bibliografías

<https://pimylifeup.com/category/projects/beginner/>

<https://www.techworld.com/picture-gallery/apps-wearables/-fun-raspberry-pi-projects-for-beginners-3636867/>

https://www.reddit.com/r/raspberry_pi/

<https://www.reddit.com/r/Arduino/>

<https://www.taloselectronics.com/blogs/tutoriales/kit-136-elementos>

<https://www.makeuseof.com/tag/10-great-Arduino-projects-for-beginners/>

<https://www.taloselectronics.com/blogs/tutoriales/ensamble-kit-drone-de-carreras-para-ensamblar-esc-simonk-y-motores-mars-power>

Tinkercad Tutorial

<https://www.youtube.com/watch?v=wi7hY2esvRo> (Completo)

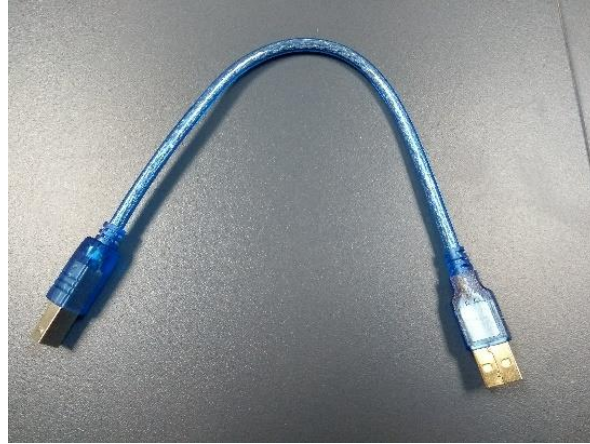
<https://www.youtube.com/watch?v=hZmSG-IALAM>

<https://www.youtube.com/watch?v=YPvQdx2Sue8> (Semaforo)

(J) Catálogo de piezas necesarias:

(Extracto del documento preparado por Lorena Ramírez para el laboratorio)

- 1 cable USB



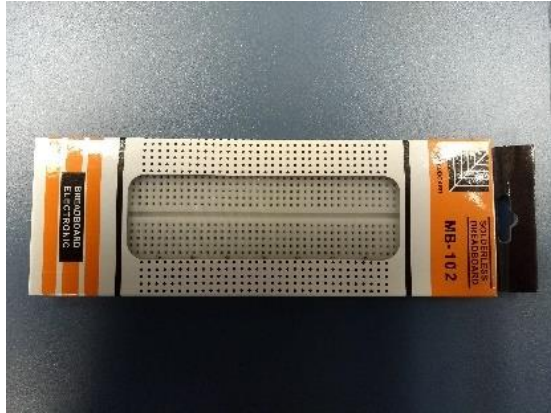
- 1 Arduino UNO



- 65 Dupont MM



- 1 Proto 830 puntos



- 10 led rojo



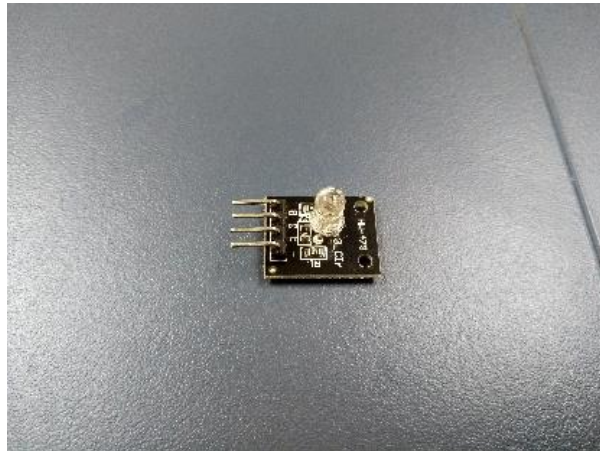
- 10 led verde



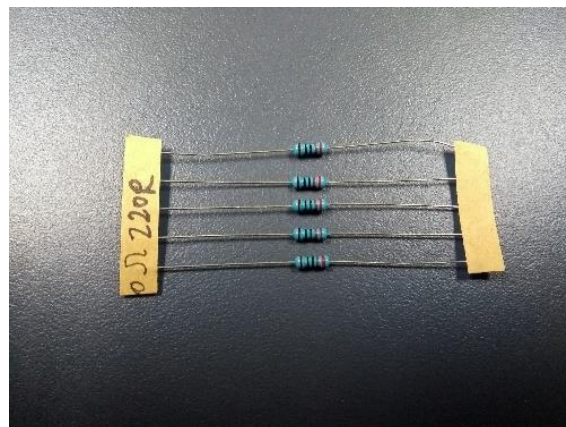
- 10 led amarillo



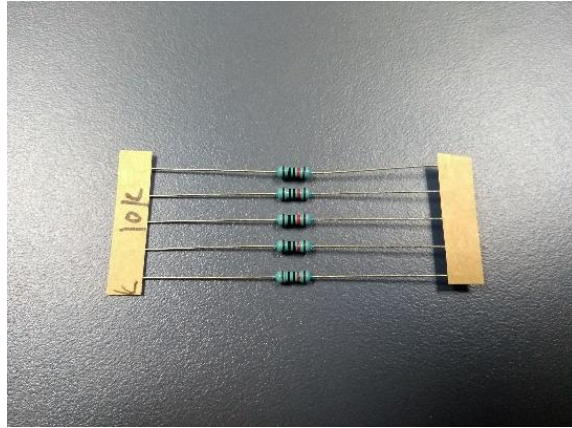
- 1 Módulo led RGB



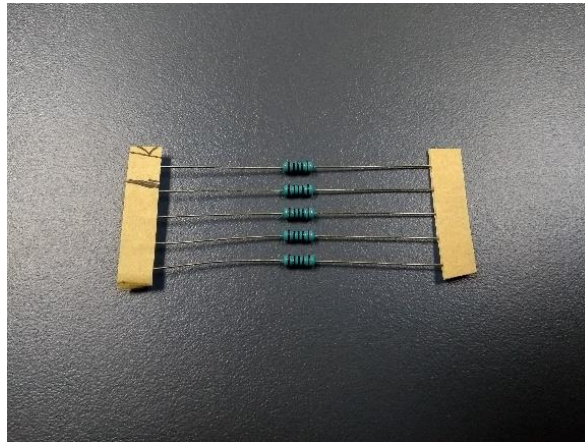
- 5 Resistencias 220R



- 5 Resistencias 10K



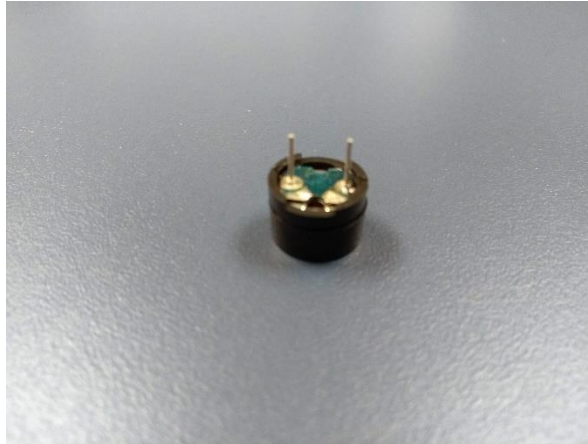
- 5 Resistencias 1K



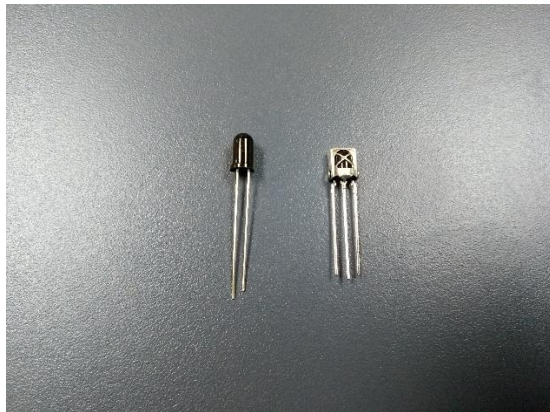
- 1 Zumbador activo



- 1 Zumbador pasivo



- 1 Sensor de flama
1 Receptor infrarrojo



- 2 Display 7 segmentos
1 Pantalla 7 segmentos 4 caracteres

