

# Tecnología en Secundaria con Arduino

---

Trabajo Final de Grado

Víctor Castro Sancho

11/01/2015

Universitat Oberta de Catalunya.

Grado en Tecnologías de la Telecomunicación. Especialidad: Telemática.

Tutor: Pere Tuset Peiró.

**Abstract.**

La asignatura de Tecnología en Educación Secundaria cuenta en la mayoría de los casos con la dificultad añadida de falta de material con qué realizar actividades prácticas. Además, en los temas relacionados con la electrónica, robótica y sistemas de control, se añade en ocasiones el desconocimiento del profesorado sobre el uso nuevas herramientas disponibles. Por ello, se ha desarrollado una solución en forma de batería de actividades prácticas utilizando Arduino como herramienta vehicular.

Para obtener un resultado que responda a las necesidades educativas reales en la materia se han estudiado el currículo educativo vigente en la Comunidad Valenciana y las tendencias en el uso de las TIC en el aula junto a distintos proyectos organizados a nivel nacional e internacional. Fruto de la investigación realizada se determinó la importancia de utilizar herramientas que permitan su libre manipulación y abaraten los costes de adquisición y reposición de material para la asignatura.

Con la información obtenida, se desarrolló una serie de baterías de actividades prácticas que responden al contenido, objetivos y criterios de evaluación de los decretos 112/2007 y 102/2008 que establecen el currículo de la ESO y el Bachillerato respectivamente en la comunidad autónoma. Como resultado, se ha obtenido un documento que recoge la colección de prácticas elaboradas incluyendo una solución propuesta para cada una de ellas.

The subject of Technology in Secondary Education has in most cases the added difficulty of providing equipment to carry out practical activities. In addition, there is some ignorance by teachers about the use of new technological tools for topics related to electronics, robotics and control systems. To address this lack, it has been developed a battery of practical activities using Arduino as main tool.

To obtain a result that meets the real educational needs on the matter, it has been taken into account both official curriculum in Valencia as trends about use of TICs in the classroom provided by national and international projects. As a result of the carried research, it was revealed the importance of using tools that enable their free manipulation and cheapen the acquisition and replacement of material for the subject.

With the gathered information, it has been developed a set of practical activities that respond to the content, objectives and assessment criteria present in both Decrees 112/2007 and 102/2008 that establish the curriculum of ESO and Bachillerato respectively in Valencia. Accordingly, it has been obtained a document which includes a collection of practices with a solution proposal for each one.

**Resumen.**

El objeto del proyecto “Tecnología en Secundaria con Arduino” es realizar una serie de baterías de actividades prácticas que utilicen la plataforma electrónica libre Arduino como herramienta vehicular. Su implementación se enfoca a los temas de electrónica, robótica y sistemas de control, relacionados con el currículo para ESO y Bachillerato de la Comunidad Valenciana. El proyecto, nace de la necesidad de incluir en las aulas herramientas que permitan su libre manipulación y abaraten los costes de adquisición.

El presente documento divide el trabajo realizado en 6 capítulos bien diferenciados: Capítulo 1. Introducción; Capítulo 2. Estado del arte; Capítulo 3. Diseño; Capítulo 4. Implementación; Capítulo 5. Resultado y Capítulo 6. Conclusiones y líneas de futuro. Además, se incluyen 4 anexos que facilitan la iniciación en el uso de las herramientas utilizadas para la realización de las prácticas: Anexo 1. Introducción a Arduino; Anexo II. Introducción a Arduino IDE; Anexo III. La protoboard y Anexo IV. Fritzing.

En primer lugar, los dos primeros capítulos introducen el propósito y alcance del proyecto y proporcionan una visión global de la situación actual en educación desde el marco legal y el social. Sobre todo, centrandó la atención en lo concerniente a la inclusión de las herramientas TIC en las aulas y en el propósito de que el alumnado alcance una mayor comprensión de las mismas siendo capaz tanto de manipularlas como de diseñarlas.

En segundo lugar, los capítulos 3 y 4 suponen las fases de diseño y planificación del trabajo. Por un lado, en el tercer capítulo se concretan los contenidos abordados por la totalidad de las actividades y se muestra el diseño estructural y la definición de cada batería y cada práctica. Por otro lado, en el cuarto capítulo se incluye el diagrama de las fases del proyecto seguido por la división en bloques del mismo y la calendarización de su desarrollo.

Finalmente, en los últimos capítulos se reúne todo el material resultante tras la elaboración del proyecto y se expone la valoración final junto a las posibles mejoras o ampliaciones a seguir en futuras actualizaciones. Adicionalmente, a través de los anexos se proporciona documentación sobre la instalación, primeros pasos y características de las principales herramientas hardware y software utilizadas para completar cada práctica.

### **Palabras clave.**

Arduino, educación, electrónica, robótica, sistemas de control.

### **Agradecimientos.**

A mi familia por su paciencia, ánimos y apoyo. Especialmente a mi madre, a mi padre (que se sentiría muy orgulloso de mi) y a mi pareja. Ellos, junto a mis mascotas Darko, Fosca y Pitidos, han hecho más llevadero todo el tiempo dedicado a mis estudios.

Gracias también a mi tutor, Pere Tuset Peiró, quien me ha guiado y asesorado durante todo el proyecto final de grado.

**Índice.**

Glosario.....	11
<b>1. Capítulo 1: Introducción.....</b>	<b>12</b>
1.1. Presentación.....	12
1.2. Objetivos.....	14
1.2.1. Objetivos generales.....	14
1.2.2. Objetivos específicos.....	15
1.3. Beneficios.....	15
1.4. Entregables.....	16
1.5. Motivación personal.....	16
<b>2. Capítulo 2: Estado del arte.....</b>	<b>18</b>
2.1. Introducción.....	18
2.1.1. Marco legal.....	18
2.1.2. Marco social y tecnológico.....	41
2.2. Antecedentes.....	42
2.2.1. Herramientas.....	42
2.2.2. Proyectos.....	47
2.3. Justificación.....	50
2.4. Expectativas.....	51
<b>3. Capítulo 3. Diseño.....</b>	<b>53</b>
3.1 Contenidos.....	53
3.1.1. Definición del contenido didáctico.....	53
3.1.2. Definición de baterías y prácticas.....	60
3.2. Diseño funcional.....	61
3.2.1. Esquema.....	62
3.2.2. Descripción de baterías y prácticas.....	62
3.3. Diseño técnico.....	73
3.3.1. Software.....	74
3.3.2. Hardware.....	75
<b>Capítulo 4. Implementación.....</b>	<b>79</b>
4.1. Diagrama del proyecto.....	79
4.2. Bloques del proyecto.....	79
4.3. Calendarización.....	81
<b>Capítulo 5. Resultado.....</b>	<b>82</b>

5.1. Prácticas.....	82
<b>Capítulo 6. Conclusiones y líneas de futuro.....</b>	<b>175</b>
6.1. Conclusiones.....	175
6.2. Líneas de futuro.....	175
<b>Anexo I. Introducción a Arduino.....</b>	<b>177</b>
<b>Anexo II. Introducción al entorno de programación Arduino (Arduino IDE).....</b>	<b>179</b>
<b>Anexo III. La protoboard.....</b>	<b>184</b>
<b>Anexo IV. Fritzing.....</b>	<b>185</b>
<b>Bibliografía.....</b>	<b>188</b>

## Índice de figuras.

Figura 1, Circuit Lab.....	43
Figura 2, Crocodile Technology.....	44
Figura 3, Fischertechnik.....	44
Figura 4, Lego Mindstorms for Education.....	45
Figura 5. Moway.....	45
Figura 6. Arduino UNO R3.....	46
Figura 7, Picaxe.....	46
Figura 8. Raspberry Pi.....	47
Figura 9, Wiring.....	47
Figura 10, Arduino.IDE.....	74
Figura 11. Fritzing.....	74
Figura 12, Arduino.....	75
Figura 13, Arduino Motor Shield.....	75
Figura 14, Protoboard.....	76
Figura 15, Calendarización de Capítulos.....	81
Figura 16, Circuito en paralelo.....	82
Figura 17, Circuito en serie.....	83
Figura 18, Esquema del circuito de la Práctica 1.3.1.1.....	84
Figura 19, Resistencia de $330\Omega$ .....	86
Figura 20, Solución propuesta para Práctica 1.3.1.1.....	86
Figura 21, Esquema y prototipo de la Práctica 1.3.1.2.....	87
Figura 22, Solución propuesta para Práctica 1.3.1.2.....	89
Figura 23, Esquema y prototipo de la Práctica 1.3.1.3.....	91
Figura 24, Solución propuesta para Práctica 1.3.1.3.....	92
Figura 25, Esquema y prototipo de la Práctica 1.3.1.4.....	93
Figura 26, Esquema y prototipo de la Práctica 1.3.1.5.....	96
Figura 27, Solución propuesta para Práctica 1.3.1.5.....	98
Figura 28, Esquema y prototipo de la Práctica 1.3.1.6.....	99
Figura 29, Solución propuesta para Práctica 1.3.1.6.....	101
Figura 30, Esquema y prototipo de la Práctica 1.3.1.7.....	102
Figura 31, Solución propuesta para Práctica 1.3.1.7.....	105
Figura 32, Esquema y prototipo solución propuesta para la Práctica 1.3.2.1.....	106
Figura 33. Esquema y prototipo de la Práctica 1.3.2.2.....	109

Figura 34, Solución propuesta para Práctica 1.3.2.2.....	111
Figura 35, Esquema y prototipo de la Práctica 1.3.2.3.....	113
Figura 36, Solución propuesta para la Práctica1.3.2.3.....	115
Figura 37, Esquema y prototipo de la Práctica 1.3.2.4.....	118
Figura 38, Solución propuesta para la Práctica 1.3.2.4.....	119
Figura 39, Solución propuesta para la Práctica 1.3.2.5.....	121
Figura 40, Esquema y prototipo de la Práctica 1.4.1.1.....	124
Figura 41, Solución propuesta Práctica 1.4.1.1.....	126
Figura 42, Esquema y prototipo de la Práctica 1.4.1.2.....	127
Figura 43, Solución propuesta Práctica 1.4.1.2.....	130
Figura 44, Esquema y prototipo de la Práctica 1.4.1.3.....	131
Figura 45, Solución propuesta Práctica 1.4.1.3.....	134
Figura 46, Esquema y prototipo de la Práctica 1.4.1.4.....	135
Figura 47, Solución propuesta Práctica 1.4.1.4.....	137
Figura 48, Solución propuesta Práctica 1.4.1.5.....	140
Figura 49, Esquema y prototipo de la Práctica1.4.2.1.....	143
Figura 50, Solución propuesta Práctica 1.4.2.1.....	148
Figura 51, Solución propuesta Práctica 1.4.2.2.....	150
Figura 52, Solución propuesta Práctica 1.4.2.3.....	152
Figura 53, Esquema y prototipo Práctica 2.2.1.1.....	155
Figura 54, Solución propuesta práctica 2.2.1.1.....	157
Figura 55, Esquema del D5611 A/B.....	158
Figura 56, Esquema y prototipo de la Práctica 2.2.1.2.....	158
Figura 57, Solución propuesta Práctica 2.2.1.2.....	163
Figura 58, Esquema y prototipo de la Práctica 2.2.1.3.....	164
Figura 59, Solución propuesta de la Práctica 2.2.1.3.....	167
Figura 60 Esquema y descripción del CI 74283.....	171
Figura 61, Esquema y prototipo de la Práctica 2.2.1.4.....	172
Figura 62, Solución propuesta Práctica 2.2.1.4.....	174
Figura 63, Esquema Arduino UNO R3.....	178
Figura 64, Descarga Arduino software.....	179
Figura 65, Instalación Arduino software.....	181
Figura 66, Comprobación encendido de Arduino.....	181
Figura 67, Icono Arduino software.....	181
Figura 68, Ejemplo blink.....	182



<b>Figura 69, Selección de tarjeta.</b> .....	182
<b>Figura 70, Selección de puerto de comunicaciones.</b> .....	183
<b>Figura 71, Cargar programa.</b> .....	183
<b>Figura 72, Protoboard y sus conexiones.</b> .....	184
<b>Figura 73, Ejemplo de uso de la protoboard.</b> .....	184
<b>Figura 74, Descarga Fritzing.</b> .....	186
<b>Figura 75, Ejecución Fritzing.</b> .....	186
<b>Figura 76, Entorno Fritzing.</b> .....	187

## Índice de tablas.

Tabla 1. Información sobre la materia de Tecnologías. ....	34
Tabla 2. Información sobre la materia Tecnología Industrial.....	39
Tabla 3. Competencias básicas de la ESO.....	41
Tabla 4. Información específica Tecnologías. ....	57
Tabla 5. Información específica sobre Tecnología Industrial.....	60
Tabla 6, Formato de la batería de actividades. ....	61
Tabla 7, Formato de las actividades.....	61
Tabla 8. Disposición de las baterías de actividades del proyecto. ....	62
Tabla 9, Batería 1.1.1. Introducción al montaje de circuitos sobre protoboard....	62
Tabla 10, Práctica 1.1.1.1. Introducción a la protoboard I.....	63
Tabla 11, Práctica 1.1.1.2. Introducción a la protoboard II.....	63
Tabla 12, Batería 1.3.1. Introducción a la electricidad y la electrónica. ....	63
Tabla 13, Práctica 1.3.1.1. LED intermitente.....	64
Tabla 14, Práctica 1.3.1.2. Regulación lumínica. ....	64
Tabla 15, Práctica 1.3.1.3. Relé, diodo y pulsador. ....	64
Tabla 16, Práctica 1.3.1.4. El condensador temporizador. ....	65
Tabla 17, Práctica 1.3.1.5. Circuito serie. ....	65
Tabla 18, Práctica 1.3.1.6. Circuito paralelo.....	65
Tabla 19, Práctica 1.3.1.7. Circuito mixto. ....	66
Tabla 20, Batería 1.3.2. Introducción al control y la robótica.....	66
Tabla 21, Práctica 1.3.2.1.Semáforo simple. ....	66
Tabla 22, Práctica 1.3.2.2.Motores con Arduino Motor Shield. ....	67
Tabla 23, Práctica 1.3.2.3.Control LED por bluetooth. ....	67
Tabla 24, Práctica 1.3.2.4.Control de luminosidad. ....	68
Tabla 25, Práctica 1.3.2.5.Vehículo teledirigido. ....	68
Tabla 26, Batería 1.4.1. Electricidad y electrónica.....	68
Tabla 27, Práctica 1.4.1.1.Alarma LDR.....	69
Tabla 28, Práctica 1.4.1.2.Termómetro LED. ....	69
Tabla 29, Práctica 1.4.1.3.Servomotor con sensor de flexión.....	69
Tabla 30, Práctica 1.4.1.4.Lógica con pulsadores. ....	70
Tabla 31, Práctica 1.4.1.5.Semáforo avanzado. ....	70
Tabla 32, Batería 1.4.2. Control y robótica. ....	70
Tabla 33, Práctica 1.4.2.1.Ayuda en estacionamiento. ....	71
Tabla 34, Práctica 1.4.2.2. Puerta automática. ....	71
Tabla 35, Práctica 1.4.2.3. Proyecto libre: Vehículo autónomo. ....	72

<b>Tabla 36, Batería 2.2.1. Sistemas automáticos.</b> .....	<b>72</b>
<b>Tabla 37, Práctica 2.2.1.1. Piano.</b> .....	<b>72</b>
<b>Tabla 38, Práctica 2.2.1.2. Contador de 0 a 9.</b> .....	<b>73</b>
<b>Tabla 39, Práctica 2.2.1.3. Puertas lógicas.</b> .....	<b>73</b>
<b>Tabla 40, Práctica 2.2.1.4. Circuitos combinacionales: El sumador.</b> .....	<b>73</b>
<b>Tabla 41, Elementos electrónicos.</b> .....	<b>78</b>
<b>Tabla 42. Diagrama del proyecto.</b> .....	<b>79</b>
<b>Tabla 43. Contenido del capítulo 1.</b> .....	<b>79</b>
<b>Tabla 44. Fases del capítulo 2.</b> .....	<b>80</b>
<b>Tabla 45. Contenido del capítulo 3.</b> .....	<b>80</b>
<b>Tabla 46. Contenido del capítulo 4.</b> .....	<b>80</b>
<b>Tabla 47. Contenido del capítulo 5.</b> .....	<b>80</b>
<b>Tabla 48. Contenido del capítulo 6.</b> .....	<b>80</b>
<b>Tabla 49. Contenido adicional para la memoria.</b> .....	<b>81</b>
<b>Tabla 50. Contenido de la presentación.</b> .....	<b>81</b>
<b>Tabla 51, Solución propuesta Práctica 1.4.1.4.</b> .....	<b>138</b>
<b>Tabla 52, Solución propuesta Práctica 1.4.1.5.</b> .....	<b>139</b>
<b>Tabla 53. Características Arduino UNO R3.</b> .....	<b>177</b>

## Glosario.

GNU (*General Public Licence*). Licencia que garantiza a los usuarios finales libertad para usar, estudiar, copiar y modificar el *software*.

Script. Archivo de órdenes o procesamiento por lotes.

TIC. Tecnologías de la información y la comunicación.

*Open Source*: Software o Hardware de libre uso, manipulación y distribución.

Hardware: Elementos físicos que constituyen un sistema informático.

Software: Programas y rutinas que permiten a la computadora realizar determinadas tareas.

## 1. Capítulo 1: Introducción.

El Capítulo 1 del presente documento supone la introducción del proyecto “Tecnología en Secundaria con Arduino” que constituye el trabajo final de carrera de los estudios de Grado en Tecnologías de Telecomunicación especialidad Telemática en la Universitat Oberta de Catalunya (UOC). Para ello, se divide en seis apartados que comprenden: Presentación, objetivos, beneficios, entregables, motivación personal y recursos necesarios.

### 1.1. Presentación.

El Trabajo Final de Grado (TFG), titulado “Tecnología en Secundaria con Arduino”, toma los contenidos y objetivos incluidos en el currículo de la Comunidad Valenciana (CV) para la materia de Tecnología en Educación Secundaria - concretamente los relacionados con electrónica, control y robótica – y plantea un modo de afrontarlos acorde a las necesidades formativas actuales, las nuevas tecnologías y las herramientas libres, utilizando la plataforma electrónica Arduino.

El Real Decreto (RD) 112/2007, por el que se establece el currículo de la Educación Secundaria Obligatoria (ESO) en la CV, define los conocimientos y líneas metodológicas que orientan la didáctica de las materias. Del mismo modo, el RD 102/2008 establece el currículo de Bachiller. Así, se fijan las competencias mínimas, objetivos principales y procedimientos que han de ser respetados por las instituciones educativas.

Ambos currículos, indican que la materia de Tecnología ha de fomentar el aprendizaje de conocimientos y el desarrollo de destrezas que permitan tanto la comprensión de los objetos técnicos como su utilización. Además, pretende que el alumnado y el profesorado utilicen las nuevas Tecnologías de la Información y la Comunicación (TIC) como herramientas en el proceso de enseñanza aprendizaje y no como fin en sí mismas.

De lo anterior, se entiende que la Tecnología se articula en torno al binomio formado por conocimiento y acción. Con lo que se plantea la necesidad de una actividad metodológica apoyada en tres principios: Adquisición de conocimientos técnicos y científicos, aplicación de los mismos al análisis y manipulación de objetos y realización

de proyectos que resuelvan un problema utilizando los conocimientos y destrezas adquiridos.

Por otro lado, existe un movimiento impulsado por personalidades influyentes de la industria tecnológica, como por ejemplo Mark Zuckerberg -creador de Facebook- o Bill Gates -fundador de Microsoft-, que pretende que la programación y la robótica se incorporen al sistema educativo con el objetivo de poder preparar al alumnado del siglo XXI para diseñar la tecnología además de utilizarla. Consecuentemente, han aparecido campañas en determinados países para promover la propuesta.

En España, Madrid ha sido la primera ciudad en actuar al respecto invirtiendo 16 millones de euros para implantar la asignatura “Programación y comunicaciones”, que se apoya en las herramientas TIC: *Scratch*, aplicación desarrollada por el Instituto Tecnológico de Massachusetts (MIT) que permite programar de modo sencillo e intuitivo y Arduino, plataforma electrónica libre que incorpora un microcontrolador programable desde su propio entorno o desde *Scratch*.

Este TFG pretende atender a la metodología propuesta por los RRDD mencionados utilizando la plataforma electrónica Arduino como herramienta vehicular en las materias de Tecnología y Tecnología Industrial. Concretamente, en los bloques: “Electrónica” y “Control y robótica” en ESO y “Elementos de máquinas y sistemas”, “Sistemas automáticos” y “Control y programación de sistemas automáticos” en Bachillerato.

El presente proyecto nace bajo la premisa de que la ciudadanía del siglo XXI ha de desarrollar una mayor competencia TIC para poder interactuar con su entorno, cada vez más tecnológico, informatizado y automatizado. Por tanto, se entiende que las nuevas generaciones no han de limitarse a utilizar las nuevas herramientas disponibles sino que han de comprenderlas y manipularlas. Así, es necesario incluir dichas competencias en la escuela desde edades tempranas.

Además, en la actualidad los bloques abarcados por el proyecto suponen una dificultad añadida para el docente de Tecnología por diversos motivos: Carencia de los conocimientos o medios para implementar sistemas de control, falta de material con que realizar las prácticas o el proyecto debido a costes, herramientas propietarias restringidas al uso especificado por el fabricante, etc. Consecuentemente, aparece el peligro de afrontar los contenidos de un modo meramente teórico.

Para afrontar el propósito del proyecto se elaboran baterías de actividades que, utilizando la plataforma Arduino, se basan en los contenidos de los bloques temáticos anteriormente apuntados y cumplen con los objetivos y metodologías planteados en los RRDD. De este modo, cada conjunto de prácticas cubre las necesidades formativas de cada bloque en su etapa educativa correspondiente según la normativa vigente y de un modo eminentemente práctico.

Para ello, a lo largo de este documento se presenta la plataforma electrónica libre Arduino junto a series de actividades prácticas en las que se incluyen: Objetivos específicos, material necesario, enunciado y solución propuesta. Con todo, se pretende ofrecer una alternativa sencilla, práctica, libre y económica con la que afrontar el contenido del currículo y cuyas limitaciones radican en la imaginación del usuario.

## **1.2. Objetivos.**

El objetivo principal del proyecto es diseñar material didáctico-práctico, basado en Arduino, para dinamizar el proceso de enseñanza aprendizaje de la materia de Tecnologías en los bloques temáticos: “Electrónica”, “Control y robótica”, “Elementos de máquinas y sistemas”, “Sistemas automáticos” y “Control y Programación de sistemas automáticos”. Además, se pretende situar al alumnado en el centro del proceso de modo que se fomente la experimentación y el autoaprendizaje.

### **1.2.1. Objetivos generales.**

- Aportar material didáctico específico basado en Arduino para los campos de electrónica en las asignaturas Tecnología de ESO y Tecnología Industrial de Bachillerato.
  - Proporcionar material didáctico para abordar los bloques de “Electrónica” y “Control y robótica” en ESO
  - Proporcionar material didáctico para abordar los bloques de “Elementos de máquinas y sistemas”, “Sistemas automáticos” y “Control y programación de sistemas automáticos” en Bachillerato.
- Fomentar el autoaprendizaje.
- Fomentar la experimentación.
- Fomentar el uso de herramientas libres en la Educación Secundaria.

- Fomentar el uso de las TIC.

### 1.2.2. Objetivos específicos.

- Diseñar una serie de baterías de actividades basadas en Arduino para abordar los contenidos y objetivos de la materia de Tecnología en ESO según el RD 112/2007.
  - Diseñar una batería de actividades para el bloque “Electrónica” en las distintas etapas educativas.
  - Diseñar una batería de actividades para el bloque “Control y robótica” en las distintas etapas educativas.
- Diseñar una serie de baterías de actividades basadas en Arduino para abordar los contenidos y objetivos de la materia de Tecnología Industrial en Bachiller según el RD 102/2008.
  - Diseñar una batería de actividades para el bloque “Elementos de máquinas y sistemas”.
  - Diseñar una batería de actividades para el bloque “Sistemas automáticos”.
  - Diseñar una batería de actividades para el bloque “Control y programación de sistemas automáticos”.

### 1.3. Beneficios.

Los beneficios aportados por el proyecto son de carácter educativo ya que como resultado se obtiene la documentación de una serie de baterías de actividades basadas en Arduino para abordar la materia de Tecnología en los bloques correspondientes a “Electrónica”, “Control y robótica”, “Elementos de máquinas y sistemas”, “Sistemas automáticos” y “Control y Programación de sistemas automáticos” de ESO y Bachiller.

En general, el proyecto facilita el proceso de enseñanza aprendizaje, tanto al profesorado como al alumnado, disponiendo material didáctico práctico, basado en el uso de herramientas libres, que fomenta la experimentación, el autoaprendizaje y el uso de las TIC. Cabe destacar que la memoria obtenida pertenece a la realización de un TFG y, por tanto, es incluida en el repositorio de la UOC donde puede ser accedida y consultada públicamente.



Por otro lado, la elaboración de las actividades supone para el autor la puesta en práctica de los conocimientos y habilidades adquiridas en las asignaturas -sobre todo las relacionadas con electrónica, programación, competencias comunicativas y gestión de proyectos- del Grado en Tecnologías de Telecomunicación cursado en la UOC. A continuación se exponen los beneficios generales obtenidos divididos entre los concernientes al educador y los concernientes al educando:

- Profesorado:
  - Fomentar la creación de proyectos que utilizan herramientas libres.
  - Compartir las prácticas y su realización con la comunidad educativa.
  - Crear situaciones de aprendizaje que impulsen la creatividad.
  - Asumir el paradigma del profesor como guía durante el aprendizaje frente al profesor como transmisor del conocimiento.
- Alumnado:
  - Aumentar conocimiento y habilidad en el uso y manipulación de las TIC.
  - Promover el auto-aprendizaje.
  - Valorar la investigación como medio para solucionar problemas.
  - Comprender el aspecto práctico del contenido teórico y relacionarlo con la realidad y sociedad actual.
  - Aprender a utilizar los actuadores y sensores de un sistema de control.
  - Comprender el funcionamiento de los sistemas robóticos.

#### **1.4. Entregables.**

Al finalizar el proyecto se obtendrá una memoria titulada “Tecnología en Secundaria con Arduino”, a incluir en el repositorio de la UOC, que contendrá, por un lado, toda la información correspondiente al trabajo y a su elaboración y, por otro lado, la documentación de la serie de baterías de actividades elaboradas por cada bloque temático (finalidad, material necesario, enunciado y solución propuesta) para cubrir los contenidos y objetivos propuestos.

#### **1.5. Motivación personal.**

Son varios los factores que influyen en el autor para la elección de este TFG. Por una parte, se encuentran la pasión por la enseñanza y el deseo por dedicar su vida profesional a la docencia. Por otra, la debilidad y fascinación hacia las nuevas tecnologías. Además, influyen el reto que supone la elaboración del proyecto y la

satisfacción personal de colaborar en el desarrollo de material didáctico basado en el uso de las TIC catalogadas como herramientas libres.

## 2. Capítulo 2: Estado del arte.

El Capítulo 2 conforma el estado actual de la temática relacionada con el proyecto y sus avances. Para comenzar, en el apartado Introducción se muestran los marcos legal y social y tecnológico en los que se basa el trabajo. Seguidamente, en el apartado Antecedentes se investigan las herramientas y proyectos existentes para afrontar la materia objeto del TFG. Después, se justifica el desarrollo del proyecto y la elección de Arduino. Para finalizar, se describen las expectativas de la publicación.

### 2.1. Introducción.

Para comprender el ámbito de actuación de este proyecto es necesario conocer tanto el marco legal como el marco social y tecnológico en el que se basa. En los siguientes subapartados se describen, en primer lugar, los aspectos legales formados por las leyes orgánicas, los reales decretos y los decretos autonómicos en materia de educación y, en segundo lugar, la actualidad social y tecnológica que engloba las necesidades ciudadanas e industriales del siglo XXI.

#### 2.1.1. Marco legal.

Las materias de Tecnología en ESO y Tecnología Industrial en Bachillerato basan en los currículos especificados por las leyes y decretos de educación sus objetivos, competencias, metodología y contenidos. Así, para embarcar cualquier acción didáctica orientada a la docencia reglada es conveniente tomar consciencia del paradigma educativo vigente. Por ello, a continuación se repasa el panorama actual de la enseñanza española en general y la valenciana, como ejemplo, en particular.

La Ley Orgánica de Educación (LOE) del 3 de mayo de 2006 regula la enseñanza educativa en España. En su artículo 6.2 determina que es competencia del Gobierno fijar los objetivos, competencias básicas, contenidos y criterios de evaluación de las enseñanzas mínimas. Además, el apartado 4 del mismo artículo, precisa que las Administraciones educativas competentes establecerán el currículo de las enseñanzas reguladas en la Ley y que éste será completado por los centros docentes.

Asimismo, la LOE fue modificada por la Ley Orgánica de Mejora de la Calidad Educativa (LOMCE) del 9 de diciembre de 2013, que redistribuye las etapas educativas, reordena las materias, modifica el sistema de evaluación, redefine la

autonomía e identidad de los centros educativos, fomenta el empleo de las nuevas tecnologías en el aula y se centra en las competencias básicas sin nombrar los objetivos.

Para dar respuesta a la competencia gubernamental de determinar el currículo educativo mínimo surgen el RD 1631/2006 del 29 de diciembre de 2006 que establece las enseñanzas mínimas correspondientes a la ESO y el RD 1467/2007 del 2 de noviembre de 2007 que fija las correspondientes a Bachillerato. De este modo queda garantizada una formación común a todo el alumnado y la validez de los títulos correspondientes en todo el territorio español.

Centrando la atención en la Comunidad Valenciana, a modo de ejemplo y con objeto de completar el currículo teniendo en cuenta las enseñanzas mínimas establecidas por el estado, los Decretos 112/2007 del 20 de julio de 2007 y 102/2008 del 11 de julio de 2008 establecen el currículo de la ESO y Bachillerato respectivamente en la comunidad. Así, estos documentos contienen los contenidos, objetivos, competencias y metodología de todas las etapas educativas y de las distintas materias.

Por un lado, el Decreto 112/2007 establece la materia Tecnologías como obligatoria en 1º y 3º de ESO con una asignación de 2 y 3 horas semanales respectivamente y la materia Tecnología como opcional según itinerario seleccionado en 4º de ESO con 3 horas semanales. Por otro lado, el Decreto 102/2008 establece las materias Tecnología Industrial I y Tecnología Industrial II como modales del Bachillerato de Ciencias y Tecnología con 4 horas semanales estipuladas.

Además, se propone que, de manera general, la metodología aplicada favorecerá la capacidad del alumnado de aprender por sí mismo, trabajar en equipo y aplicar métodos de investigación apropiados. Adicionalmente, se establece que los centros y sus departamentos didácticos desarrollarán actividades que estimulen el interés y la capacidad de uso de las TIC, el hábito de lectura y expresión oral y el uso de lenguas extranjeras.

Ambos Decretos definen la Tecnología como el conjunto de actividades y aplicaciones científico-técnicas empleadas por el ser humano para la construcción de objetos o sistemas de ellos encaminados a satisfacer sus necesidades así como resolver los problemas tanto individuales como colectivos que la sociedad actual demanda. Por

ello, la materia está encaminada a la industria y sus contenidos han de orientarse al campo de la fabricación, uso y mantenimiento de los instrumentos técnicos.

A continuación se exponen los objetivos, contenidos y criterios de evaluación para cada una de las materias citadas anteriormente. En la ESO, los contenidos se dividen en bloques de contenido que pueden ser abordados de manera conjunta y en Bachillerato se agrupan en núcleos de contenido. Cada etapa educativa posee objetivos globales y cada curso dentro de las mismas criterios de evaluación que permiten alcanzarlos.

Tecnologías
Bloques de contenidos
<p>1. Proceso de resolución de problemas tecnológicos. Constituye uno de los ejes metodológicos en torno al cual se articula la materia, de modo que el resto de los bloques proporcionan recursos e instrumentos para desarrollarlo.</p> <p>2. Hardware y software. Integra parte de los contenidos asociados a las tecnologías de la información y la comunicación.</p> <p>3. Técnicas de expresión y comunicación. Posibilita al alumnado el empleo de las técnicas básicas de dibujo y comunicación gráfica necesarias para la actividad tecnológica.</p> <p>4. Materiales de uso técnico. Recoge los contenidos básicos sobre características, propiedades y aplicaciones de los materiales técnicos más comunes, empleados en la industria, incorporando además aquellos relativos a técnicas de trabajo, hábitos de seguridad y salud y trabajo en equipo.</p> <p>5. Estructuras. Proporciona el conocimiento de las fuerzas que soporta una estructura y los esfuerzos a los que están sometidos los elementos que la forman, y determina su función dentro de la misma.</p> <p>6. Mecanismos. Su propósito es conocer los operadores básicos para la transmisión de movimientos y fuerzas.</p> <p>7. Electricidad y electrónica. Estudia los fenómenos y sistemas asociados a la fuente de energía más utilizada en las máquinas, así como aquellos cuyo funcionamiento se basa en el control del flujo de los electrones.</p> <p>8. Tecnologías de la comunicación e Internet. Desarrolla los contenidos ligados a las diversas tecnologías, cableadas e inalámbricas, que posibilitan el actual modelo de</p>

sociedad de la información.

9. Energía y su transformación. Proporciona el conocimiento de las fuentes de energía y las tecnologías asociadas para explotarlas y hacer uso de las mismas.

10. Control y robótica. Se refiere al estudio de sistemas capaces de regular su propio comportamiento, y permite aproximar varias tecnologías entre sí.

11. Neumática e hidráulica. Estudia la tecnología que emplea el aire comprimido y los líquidos como modo de transmisión de la energía necesaria para mover y hacer funcionar mecanismos. Estos contenidos están íntimamente relacionados con los contenidos de electrónica y robótica, dado que en la actualidad la industria emplea robots neumáticos o hidráulicos controlados mediante dispositivos electrónicos.

12. Tecnología y sociedad. Trata de entender los aspectos sociales del fenómeno tecnológico, tanto en lo que respecta a sus condicionantes sociales como en lo que atañe a sus consecuencias sociales y ambientales.

13. Instalaciones en viviendas. El alumnado debe adquirir conocimientos sobre los componentes que forman las distintas instalaciones de una vivienda entendiendo su uso y funcionamiento. Ha de reconocer en un plano y en el contexto real los distintos elementos, potenciando el buen uso para conseguir ahorro energético.

#### Objetivos

1. Abordar con autonomía y creatividad, individualmente y en grupo, problemas tecnológicos trabajando de forma ordenada y metódica para estudiar el problema, recopilar y seleccionar información procedente de distintas fuentes, elaborar la documentación pertinente, concebir, diseñar, planificar y construir objetos o sistemas que resuelvan el problema estudiado y evaluar su idoneidad desde distintos puntos de vista.

2. Adquirir destrezas técnicas y conocimientos suficientes para el análisis, diseño y elaboración de objetos y sistemas tecnológicos mediante la manipulación, de forma segura y precisa, de materiales y herramientas.

3. Analizar los objetos y sistemas técnicos para comprender su funcionamiento, conocer sus elementos y las funciones que realizan, aprender la mejor forma de usarlos y controlarlos, entender las condiciones fundamentales que han intervenido en su diseño y construcción y valorar las repercusiones que ha generado su existencia.

4. Expresar y comunicar ideas y soluciones técnicas, así como explorar su viabilidad y alcance utilizando los medios tecnológicos, recursos gráficos, la simbología y el

vocabulario adecuados.

5. Adoptar actitudes favorables a la resolución de problemas técnicos, desarrollando interés y curiosidad hacia la actividad tecnológica, analizando y valorando críticamente la investigación, la innovación y el desarrollo tecnológico y su influencia en la sociedad, en el medio ambiente, en la salud y en el bienestar personal y colectivo a lo largo de la historia de la humanidad.

6. Comprender las funciones de los componentes físicos de un ordenador y conocer las formas de conectarlos.

7. Manejar con soltura aplicaciones informáticas que permitan buscar, almacenar, organizar, manipular, recuperar y presentar información, empleando de forma habitual las redes de comunicación.

8. Asumir de forma crítica y activa el avance y la aparición de nuevas tecnologías, e incorporarlas a su quehacer cotidiano, analizando y valorando críticamente su influencia sobre la sociedad y el medio ambiente.

9. Actuar de forma dialogante, flexible y responsable en el trabajo en equipo, en la búsqueda de soluciones, en la toma de decisiones y en la ejecución de las tareas encomendadas con actitud de respeto, cooperación, tolerancia y solidaridad.

10. Conocer las necesidades personales y colectivas más cercanas, así como las soluciones más adecuadas que ofrece el patrimonio tecnológico del propio entorno.

11. Conocer, valorar y respetar las normas de seguridad e higiene en el trabajo y tomar conciencia de los efectos que tienen sobre la salud personal y colectiva.

### **Primer curso**

#### **Bloque 1. Proceso de resolución de problemas tecnológicos**

- Introducción al proyecto técnico y sus fases. El proceso inventivo y de diseño: identificación del problema o necesidad, exploración e investigación del entorno, búsqueda de información, diseño, planificación y organización de tareas, gestión y valoración de trabajos.
- Diseño, planificación y construcción de modelos mediante el uso de materiales, herramientas y técnicas estudiadas.
- Empleo de procesador de texto y herramientas de presentaciones para la elaboración y difusión del proyecto.

#### **Bloque 2. Hardware y software**

- Elementos que constituyen un ordenador. Unidad central y periféricos.

Funcionamiento, manejo básico y conexión de los mismos.

- Sistema operativo. Almacenamiento, organización y recuperación de la información en soportes físicos, locales y extraíbles.
  - Dispositivos que intercambian información con el ordenador: cámaras digitales, memorias externas, PDA y teléfonos móviles.
  - El ordenador como herramienta de expresión y comunicación de ideas.
- Conocimiento y aplicación de terminología y procedimientos básicos de programas como procesadores de texto y herramientas de presentaciones.

### Bloque 3. Técnicas de expresión y comunicación.

- Instrumentos de dibujo para realizar bocetos y croquis. Soportes y formatos.
- Análisis de objetos sencillos mediante la descomposición en vistas. Introducción a la representación en perspectiva.
- Utilización del ordenador como herramienta de expresión gráfica.

### Bloque 4. Materiales de uso técnico

- Materiales de uso habitual: clasificación general. Materiales naturales y transformados.
- La madera: constitución. Propiedades y características. Maderas de uso habitual. Identificación de maderas naturales y transformadas. Derivados de la madera: papel y cartón. Tableros artificiales. Aplicaciones más comunes de las maderas naturales y manufacturadas.
- Técnicas básicas e industriales para el trabajo con madera. Manejo de herramientas y uso seguro de las mismas. Elaboración de objetos sencillos empleando la madera y sus transformados como materia fundamental.
- Repercusiones medioambientales de la explotación de la madera.
- Materiales férricos: el hierro. Extracción. Fundición y acero. Obtención y propiedades características: mecánicas, eléctricas y térmicas. Aplicaciones.
- Metales no férricos: cobre y aluminio. Obtención y propiedades características: mecánicas, eléctricas y térmicas. Aplicaciones.
- Identificación de los materiales metálicos de uso común.
- Técnicas básicas e industriales para el trabajo con metales. Conformación, unión y acabado. Tratamientos. Manejo de herramientas y uso seguro de las mismas.
- Repercusiones medioambientales de la explotación de metales.
- Trabajo en el aula taller con materiales comerciales y reciclados.

### Bloque 5. Estructuras



- Tipos de estructuras resistentes: masivas, entramadas, trianguladas y colgadas. Estructuras de barras. Triangulación. Tipos de apoyo.
- Esfuerzos básicos: tipos. Elementos resistentes. Aplicaciones.
- Análisis de comportamientos estructurales mediante el diseño, planificación y construcción de distintos modelos de estructuras.

#### Bloque 6. Mecanismos

- Máquinas simples: palancas y poleas.
- Descripción y funcionamiento de mecanismos de transmisión y transformación de movimientos: poleas, engranajes, tornillo sin fin, piñón y cremallera, leva, rueda excéntrica, biela y manivela. Relación de transmisión. Aplicaciones.
- Análisis del funcionamiento en máquinas simples y simuladores físicos e informáticos.
- Aplicaciones en proyectos y maquetas, siguiendo el proceso de resolución técnica de problemas.

#### Bloque 7. Electricidad y electrónica

- Introducción a la corriente eléctrica continua, definición y magnitudes básicas: voltaje, resistencia e intensidad.
- Descripción de circuitos eléctricos simples: funcionamiento y elementos. Introducción al circuito en serie y en paralelo. Simbología.
- Efectos de la corriente eléctrica: luz y calor. Análisis de objetos técnicos que apliquen estos efectos.
- Montaje de circuitos eléctricos sencillos. Utilización de esquemas, materiales y herramientas. Simuladores físicos e informáticos.

#### Bloque 8. Tecnologías de la comunicación. Internet

- Internet: conceptos, terminología, estructura y funcionamiento.
- El ordenador como medio de comunicación: Internet y páginas web. Herramientas y aplicaciones básicas para la búsqueda, descarga, intercambio y difusión de la información. Correo electrónico, chats y otros.
- Búsqueda selectiva y crítica de información a través de Internet.

#### Bloque 9. Energía y su transformación

- Fuentes de energía: clasificación general. Energías renovables y no renovables. Ventajas e inconvenientes.
- Energías no renovables. Combustibles fósiles: petróleo, carbón y gas natural.
- Transformación de energía térmica en mecánica: la máquina de vapor, el motor

de combustión interna, la turbina y el reactor. Descripción y funcionamiento.

#### Bloque 10. Tecnología y sociedad

- La tecnología como respuesta a las necesidades humanas: fundamento del quehacer tecnológico.
- Introducción al estudio del entorno tecnológico y productivo de la Comunitat Valenciana.

#### Criterios de evaluación

1. Valorar y utilizar el proyecto técnico como instrumento de resolución ordenada de problemas.
2. Elaborar un plan de trabajo y realizar las operaciones técnicas previstas con criterios de seguridad y valorar las condiciones del entorno.
3. Identificar y conectar los componentes fundamentales del ordenador y sus periféricos, y explicar su misión en el conjunto.
4. Manejar el entorno gráfico de los sistemas operativos como interfaz de comunicación con la máquina.
5. Emplear el ordenador como herramienta de trabajo, con el objeto de comunicar, localizar y manejar información de diversas fuentes. Conocer y aplicar la terminología y procedimientos básicos de los procesadores de texto y herramientas de presentaciones.
6. Representar objetos sencillos mediante bocetos, croquis, vistas y perspectivas, con el fin de comunicar un trabajo técnico.
7. Conocer la clasificación general de los materiales de uso habitual.
8. Conocer las propiedades básicas de la madera como material técnico, sus variedades y transformados más empleados, identificarlos en las aplicaciones más usuales y emplear sus técnicas básicas de conformación, unión y acabado de forma correcta, respetando los criterios de seguridad adecuados.
9. Conocer las propiedades básicas de los metales como materiales técnicos, sus variedades y transformados más empleados, identificarlos en las aplicaciones más usuales y emplear sus técnicas básicas de conformación, unión y acabado de forma correcta, respetando los criterios de seguridad adecuados.
10. Identificar, analizar y describir, en sistemas sencillos y estructuras del entorno, elementos resistentes y los esfuerzos a los que están sometidos.

11. Señalar, en máquinas complejas, los mecanismos simples de transformación y transmisión de movimientos que las componen, y explicar su funcionamiento en el conjunto. Calcular la relación de transmisión en los casos en los que proceda.
12. Utilizar apropiadamente mecanismos y máquinas simples en proyectos y maquetas.
13. Utilizar adecuadamente las magnitudes eléctricas básicas.
14. Valorar los efectos de la energía eléctrica y su capacidad de conversión en otras manifestaciones energéticas.
15. Identificar y utilizar correctamente los elementos fundamentales de un circuito eléctrico de corriente continua y comprender su función dentro de él.
16. Emplear el ordenador como instrumento eficaz para localizar información en Internet.
17. Acceder a Internet como medio de comunicación, empleando el correo electrónico y el chat.
18. Conocer las distintas fuentes de energía, su clasificación, su transformación, sus ventajas e inconvenientes.
19. Describir los procesos de obtención y utilización de energía a partir de combustibles fósiles.
20. Distinguir las partes y describir el funcionamiento de una máquina de vapor, un motor de combustión interna, una turbina y un reactor.
21. Utilizar adecuadamente programas de simulación por ordenador.
22. Describir, comprender y valorar las oportunidades que ofrece el entorno tecnológico y productivo de la Comunidad Valenciana.

### Tercer curso

#### Bloque 1. Proceso de resolución de problemas tecnológicos

- Documentos técnicos necesarios para elaborar un proyecto.
- Diseño, planificación y construcción de prototipos mediante el uso de materiales, herramientas y técnicas estudiadas.
- Empleo de hoja de cálculo y herramientas gráficas para la elaboración, desarrollo y difusión del proyecto.
- Análisis y valoración de las condiciones del entorno de trabajo. Aplicación de las normas de seguridad al aula taller.

### Bloque 2. Hardware y software

- Instalación, desinstalación y actualización de programas. Realización de tareas básicas de mantenimiento del sistema. Gestión de recursos compartidos en redes locales.
- Herramientas básicas para el dibujo vectorial y el grafismo artístico.
- Conocimiento y aplicación de terminología y procedimientos básicos de hojas de cálculo. Fórmulas. Elaboración de gráficas
- El ordenador como herramienta para la organización de la información: gestor de bases de datos. Búsqueda de información, creación y actualización de una base de datos.

### Bloque 3. Técnicas de expresión y comunicación

- Sistemas sencillos de representación. Vistas y perspectivas. Proporcionalidad entre dibujo y realidad. Escalas. Acotación.
- Metrología e instrumentos de medida de precisión: calibre y micrómetro. Conocimiento y uso de dichos instrumentos de medida.
- Aplicaciones de dibujo asistido por ordenador.

### Bloque 4. Materiales de uso técnico

- Introducción a los plásticos. Clasificación. Obtención. Propiedades características. Identificación en objetos de uso habitual. Aplicaciones industriales y en viviendas.
- Técnicas básicas e industriales para el trabajo con plásticos. Conformación y unión. Herramientas y uso seguro de las mismas.
- Materiales de construcción: pétreos y cerámicos. Propiedades características. Identificación. Aplicaciones.
- Trabajo en el aula taller con materiales comerciales y reciclados.

### Bloque 5. Electricidad y electrónica

- Ley de Ohm. Potencia y energía eléctrica.
- Circuito en serie, paralelo y mixto.
- Corriente continua y corriente alterna. Estudio comparado.
- Electromagnetismo. Aplicaciones: electroimán. Dinamo, motor de corriente continua, relé y alternador.
- Aparatos de medida: voltímetro, amperímetro y polímetro. Realización de medidas sencillas.
- Montaje de circuitos característicos.

- Introducción a la electrónica básica: la resistencia, el condensador, el diodo y el transistor. Descripción de componentes y montajes básicos.
- Valoración de los efectos del uso de la energía eléctrica sobre el medio ambiente.

#### Bloque 6. Tecnologías de la comunicación. Internet

- El ordenador como medio de comunicación intergrupar: comunidades y aulas virtuales. Internet: foros, blogs, wikis y elaboración de páginas web.
- Actitud crítica y responsable hacia la propiedad y la distribución del software y de la información: tipos de licencias de uso y distribución.
- Introducción a la comunicación cableada e inalámbrica. El espacio radioeléctrico. Satélites y sus aplicaciones civiles.
- Introducción a la telefonía, radio y televisión. Su uso responsable.

#### Bloque 7. Energía y su transformación

- Energía eléctrica: generación, transporte y distribución.
- Centrales. Descripción y tipos de centrales hidroeléctricas, térmicas y nucleares. Tratamientos de los residuos.
- Energías renovables: sistemas técnicos para el aprovechamiento de la energía eólica, solar, mareomotriz y biomasa. Importancia del uso de energías alternativas.
- Energía y medio ambiente. Eficiencia y ahorro energético. Impacto medioambiental de la generación, transporte, distribución y uso de la energía.

#### Bloque 8. Control y robótica

- Introducción y evolución de los sistemas automáticos: mecanización, automatización y robotización.
- Elementos básicos de un sistema de control. Métodos y ejemplos de sistemas de control.
- Arquitectura de un robot: partes fundamentales. Sensores y actuadores.
- Control de un automatismo por ordenador. Tarjetas controladoras. Diagramas de flujo. Introducción a la programación.

#### Bloque 9. Tecnología y sociedad

- Tecnología y medio ambiente: impacto ambiental del desarrollo tecnológico. Contaminación. Agotamiento de los recursos energéticos y de las materias primas. Tecnologías correctoras. Desarrollo sostenible.

#### Criterios de evaluación

1. Realizar un proyecto técnico, analizando el contexto, proponiendo soluciones alternativas y desarrollando la más adecuada.
2. Elaborar los documentos técnicos necesarios para redactar un proyecto técnico, mediante el lenguaje escrito y gráfico apropiado.
3. Realizar las operaciones técnicas previstas en el proyecto técnico incorporando criterios de economía, sostenibilidad y seguridad; valorar las condiciones del entorno de trabajo.
4. Emplear el ordenador como herramienta para elaborar, desarrollar y difundir un proyecto técnico, a través de hojas de cálculo que incorporen fórmulas y gráficas.
5. Instalar, desinstalar y actualizar programas y realizar tareas básicas de mantenimiento informático. Utilizar y compartir recursos en redes locales.
6. Realizar dibujos geométricos y artísticos utilizando algún programa de dibujo gráfico sencillo.
7. Utilizar vistas, perspectivas, escalas, acotación y normalización para plasmar y transmitir ideas tecnológicas y representar objetos y sistemas técnicos.
8. Crear una base de datos sencilla; actualizar y modificar una base de datos ya creada. Localizar información utilizando un gestor de bases de datos.
9. Utilizar aplicaciones de diseño asistido por ordenador para la realización de croquis normalizados.
10. Conocer las propiedades básicas de los plásticos como materiales técnicos, su clasificación, sus aplicaciones más importantes; identificarlos en objetos de uso habitual y usar sus técnicas básicas de conformación y unión de forma correcta y con seguridad.
11. Conocer las propiedades básicas de los materiales de construcción, sus aplicaciones más importantes, su clasificación, sus técnicas de trabajo y uso, e identificarlos en construcciones ya acabadas.
12. Diseñar, simular y realizar montajes de circuitos eléctricos sencillos, en corriente continua, empleando pilas, interruptores, resistencias, bombillas, motores, electroimanes y relés, como respuesta a un fin predeterminado.
13. Describir las partes y el funcionamiento de las máquinas eléctricas básicas.
14. Describir y utilizar el electromagnetismo en aplicaciones tecnológicas sencillas.
15. Utilizar correctamente las magnitudes eléctricas básicas, sus instrumentos de

medida y su simbología.

16. Montar un circuito electrónico sencillo empleando, al menos, diodos, transistores y resistencias, a partir de un esquema predeterminado.

17. Emplear Internet como medio activo de comunicación intergrupar y publicación de información.

18. Conocer y valorar los diferentes modelos de propiedad y distribución del software y de la información en general.

19. Describir esquemáticamente los sistemas de telefonía, radio, televisión y satélites civiles, sus principios básicos de funcionamiento y conocer los aspectos prácticos más importantes a nivel de usuario.

20. Conocer los distintos medios de producción, transformación y transporte de la energía eléctrica.

21. Describir esquemáticamente el funcionamiento y tipos de centrales productoras de energía.

22. Describir esquemáticamente los sistemas técnicos para el aprovechamiento de las energías renovables.

23. Conocer y valorar el impacto medioambiental de la generación, transporte, distribución y uso de la energía, fomentando una mayor eficiencia y ahorro energético.

24. Identificar automatismos en sistemas técnicos cotidianos, diferenciando sus elementos básicos.

25. Montar, utilizando sistemas mecánicos y eléctricos, un robot sencillo con capacidad de movimiento dirigido.

26. Realizar diagramas de flujo sencillos y conocer las órdenes más utilizadas en los programas de control.

27. Reconocer el impacto que sobre el medio produce la actividad tecnológica y comprobar los beneficios y necesidad de la aplicación de tecnologías correctoras para conseguir un desarrollo sostenible.

#### Cuarto curso

##### Bloque 1. Hardware y software

- El ordenador como dispositivo de control: señales analógicas y digitales. Lógica de funcionamiento interno. Transmisión de la información por medio de señal eléctrica. Adquisición de datos. Tratamiento de la información numérica adquirida

mediante hoja de cálculo.

- Programas de control.

### Bloque 2. Técnicas de expresión y comunicación

- Diseño asistido por ordenador: dibujo en dos dimensiones. Realización de dibujos sencillos.

### Bloque 3. Electricidad y electrónica

- Descripción y análisis de sistemas electrónicos por bloques: entrada, salida y proceso. Componentes electrónicos. Circuitos integrados simples. Dispositivos de entrada: interruptores, resistencias que varían con la luz y la temperatura. Dispositivos de salida: zumbador, relé, LED y otros. Aplicaciones en montajes sencillos.
- Técnicas de montaje y conexión de circuitos electrónicos
- Electrónica digital. Aplicación del álgebra de Boole a problemas tecnológicos básicos. Puertas lógicas: simbología.
- Uso de simuladores para analizar el comportamiento de los circuitos electrónicos.

### Bloque 4. Tecnologías de la comunicación. Internet

- Descripción de los sistemas de comunicación alámbrica e inalámbrica y sus principios técnicos, para transmitir sonido, imagen y datos. Ejemplos prácticos.
- Comunicación inalámbrica: señal moduladora y portadora.
- Comunicación vía satélite y telefonía móvil. Descripción y principios técnicos.
- Sistemas de posicionamiento global. Descripción y principios técnicos.
- Grandes redes de comunicación de datos. Perspectiva de desarrollo. Control y protección de datos.
- Internet: Principios técnicos de su funcionamiento (protocolos lógicos e infraestructura física). Conexiones a Internet. Tipos.
- Utilización racional de tecnologías de la comunicación de uso cotidiano.

### Bloque 5. Control y robótica

- Experimentación con sistemas automáticos, sensores, reguladores y actuadores. Aplicaciones prácticas. Concepto de realimentación.
- Uso del ordenador como elemento de programación y control. Lenguajes de control.
- Diseño y construcción de sistemas automáticos y desarrollo de programas para controlarlos.
- Trabajo con simuladores informáticos para comprobar el funcionamiento de los



sistemas diseñados.

#### Bloque 6. Neumática e hidráulica

- Descripción y análisis de los sistemas hidráulicos y neumáticos, de sus componentes y principios físicos de funcionamiento.
- Diseño y simulación con programas informáticos de circuitos básicos, empleando simbología específica. Ejemplos de aplicación en sistemas industriales.
- Montajes sencillos.

#### Bloque 7. Tecnología y sociedad

- Desarrollo histórico de la tecnología. Hitos fundamentales: Revolución Neolítica, Revolución Industrial y aceleración tecnológica del siglo XX.
- Análisis de la evolución de los objetos técnicos e importancia de la normalización en los productos industriales.
- Aprovechamiento de materias primas y recursos naturales. Adquisición de hábitos que potencien el desarrollo sostenible.

#### Bloque 8. Instalaciones en viviendas

- Análisis de los elementos que configuran las instalaciones de una vivienda y su funcionamiento: electricidad, agua sanitaria, evacuación de aguas, sistemas de calefacción, gas, aire acondicionado, domótica y otras instalaciones.
- Acometidas, componentes, normativa, simbología, análisis, diseño y montaje en equipo de modelos sencillos de estas instalaciones. Utilización de simuladores informáticos.
- Estudio de facturas domésticas.
- Concepto de arquitectura bioclimática para el aprovechamiento energético.

#### Criterios de evaluación

1. Manejar la hoja de cálculo para el tratamiento de la información numérica y analizar pautas de comportamiento.
2. Describir básicamente una red de ordenadores de área local y su conexión a Internet y realizar su configuración básica.
3. Utilizar el ordenador como herramienta de adquisición e interpretación de datos, y como realimentación de otros procesos con los datos obtenidos.
4. Emplear herramientas de diseño asistido por ordenador para elaborar vistas en dos dimensiones de objetos sencillos.
5. Describir el funcionamiento, aplicación y componentes elementales de un sistema

electrónico real.

6. Diseñar, simular y montar circuitos electrónicos sencillos, utilizando la simbología adecuada.

7. Realizar operaciones lógicas empleando el álgebra de Boole, relacionar planteamientos lógicos con procesos técnicos y resolver mediante puertas lógicas problemas tecnológicos sencillos.

8. Analizar y describir los elementos y sistemas de comunicación alámbrica e inalámbrica, para la transmisión de imagen, sonido y datos, y los principios técnicos básicos que rigen su funcionamiento.

9. Describir las grandes redes de comunicación de datos, sus perspectivas y los principios del control y la protección de datos.

10. Conocer los principios básicos del funcionamiento de Internet. Configurar un ordenador para su acceso a Internet.

11. Hacer un uso adecuado y racional de las tecnologías de la comunicación.

12. Analizar sistemas automáticos, describir sus componentes y montar automatismos sencillos.

13. Diseñar, construir y programar un sistema automático, que sea capaz de mantener su funcionamiento de forma autónoma, en función de la información que reciba del entorno mediante sensores.

14. Utilizar simuladores informáticos para verificar y comprobar el funcionamiento de los sistemas automáticos, robots y programas de control diseñados.

15. Conocer las principales aplicaciones de las tecnologías hidráulica y neumática e identificar y describir las características y funcionamiento de este tipo de sistemas.

16. Utilizar con soltura la simbología y nomenclatura necesaria para representar circuitos y para diseñar y construir un sistema capaz de resolver un problema cotidiano, utilizando energía hidráulica o neumática.

17. Conocer los hitos fundamentales del desarrollo tecnológico y analizar la evolución de algunos objetos técnicos.

18. Valorar el desarrollo sostenible y potenciar hábitos que lo propicien, relacionándolo con la actividad tecnológica.

19. Describir los elementos que componen las distintas instalaciones de una vivienda así como las normas que regulan su diseño y utilización.

20. Realizar diseños sencillos de instalaciones básicas de una vivienda mediante la simbología adecuada; montar circuitos básicos y utilizar simuladores informáticos.

21. Valorar de forma crítica las condiciones que contribuyen al ahorro energético, habitabilidad y estética en una vivienda.

Tabla 1. Información sobre la materia de Tecnologías. Fuente: DECRETO 112/2007

<b>Tecnología Industrial</b>	
<b>Objetivos</b>	
1.	Aplicar los conocimientos adquiridos a la comprensión y análisis del funcionamiento de máquinas y sistemas técnicos.
2.	Comprender el papel de la energía en los procesos tecnológicos, sus distintas transformaciones y aplicaciones y adoptar actitudes de ahorro valorando la eficiencia energética.
3.	Comprender y explicar cómo se organizan y desarrollan procesos tecnológicos concretos, identificando y describiendo las técnicas y los factores económicos y sociales que concurren en cada caso.
4.	Analizar de forma sistemática aparatos y productos de la actividad técnica para explicar su funcionamiento, utilización y forma de control evaluando su calidad.
5.	Valorar críticamente, aplicando los conocimientos adquiridos, las repercusiones de la actividad tecnológica en la vida cotidiana y en la calidad de vida, manifestando y argumentando sus ideas y opiniones.
6.	Expresar con precisión ideas y opiniones sobre procesos o productos tecnológicos concretos, utilizando vocabulario, símbolos y formas de expresión apropiadas.
7.	Participar en la planificación y desarrollo de proyectos técnicos en equipo, en los que intervengan elementos básicos, aportando ideas y opiniones, responsabilizándose de tareas y cumpliendo sus compromisos.
8.	Actuar con autonomía y confianza al inspeccionar, manipular e intervenir en máquinas, sistemas y procesos técnicos para comprender su funcionamiento.
<b>Tecnología Industrial I</b>	
<b>1. Fuentes energéticas</b>	
-	Características de las principales fuentes primarias de energía. Su obtención, transformación y transporte.
-	Montaje y experimentación de instalaciones sencillas de transformación de energía.
-	Consumo energético. Aplicación y cálculo de energía en un proceso real y

técnicas de ahorro en el consumo.

## 2. Materiales

- Estado natural, características, obtención y transformación de los materiales más utilizados en el ámbito industrial. Aplicaciones y propiedades más relevantes de estos materiales. Impacto ambiental producido por la obtención, transformación y desecho de los materiales.
- Nuevos materiales.
- Estructura interna, modificación y propiedades.

## 3. Elementos de máquinas y sistemas

- Transmisión y transformación de movimientos. Soporte y unión de elementos mecánicos.
- Montaje y experimentación de mecanismos característicos. Elementos de un circuito genérico: generador, conductor, dispositivos de regulación y control, receptores de consumo y utilización. Representación esquematizada de circuitos. Simbología. Interpretación de planos y esquemas.
- Corriente continua; magnitudes fundamentales. Circuitos en corriente continua; cálculo de sus magnitudes fundamentales. Montaje y experimentación de algunos circuitos eléctricos y neumáticos característicos.

## 4. Procedimientos de fabricación

- Clasificación de las técnicas de fabricación: corte, arranque de material, conformación en frío y en caliente, unión y tejido de materiales. Máquinas y herramientas apropiadas para cada procedimiento. Criterios de uso y mantenimiento de herramientas. Medidas de salud y seguridad en el trabajo. Normas de salud y seguridad en el centro de trabajo.
- Impacto ambiental de los procedimientos de fabricación.

## 5. El proceso y los productos de la tecnología

- Proceso de diseño y mejora de productos. Distribución de productos. El mercado y sus leyes básicas. Control de calidad. Aplicación técnica del control de calidad. Normalización de productos.
- Planificación y desarrollo de un proyecto de diseño y comercialización de un producto.

## Criterios de evaluación

1. Calcular a partir de información adecuada el coste energético del funcionamiento

ordinario del centro docente o de su vivienda y sugerir posibles alternativas de ahorro. El alumnado ha de ser capaz de estimar la carga económica que supone el consumo cotidiano de energía, utilizando información comercial, facturas de servicios energéticos y cálculos efectuados sobre las características técnicas, utilización y consumo de las instalaciones. Esta capacidad ha de derivar en la identificación de posibles vías de reducción de costes.

2. Analizar las distintas situaciones en las que la sociedad actual hace uso de la energía.

Para realizar cualquier tipo de trabajo en la vida, se hace uso de la energía; el alumnado deberá conocer las principales fuentes de energía y las posibles formas de aplicación de las misma, así como el proceso desde su obtención hasta su consumo.

3. Describir los procesos de fabricación y características de los materiales más importantes que la industria actual demanda, estimando las repercusiones económicas y ambientales de su producción, uso y desecho.

El alumnado deberá conocer las propiedades físico-químicas así como los procedimientos de obtención y aplicaciones de los materiales más utilizados en la industria actual, tales como los metales, plásticos, aislantes y semiconductores. Asimismo deberá elaborar juicios de valor sobre los factores no estrictamente técnicos, como las repercusiones medioambientales de su producción uso y desecho.

4. Emplear aparatos de medida y efectuar medidas correctamente. A la hora de diseñar cualquier proyecto nos encontramos con magnitudes dependientes de las energías objeto del proyecto; en nuestro caso, necesitamos pues conocer las magnitudes fundamentales de la energía eléctrica para poder calcular las secciones de los conductores y también los elementos de mando y protección, por lo que deberá conocerse el empleo amperímetros, voltímetros y vatímetros.

5. Identificar los elementos funcionales que componen un producto técnico de uso conocido, señalando el papel que desempeña cada uno de ellos en el funcionamiento del conjunto. El alumnado ha de ser capaz de desarmar un artefacto, reconocer cuáles son las piezas y subconjuntos más importantes y cuáles son accesorios desde el punto de vista funcional y describir el papel de cada componente en el funcionamiento del conjunto.

6. Evaluar las repercusiones que sobre la calidad de vida tiene la producción y utilización de un producto o servicio técnico cotidiano y sugerir posibles alternativas de mejora, tanto técnicas como de otro orden.

La capacidad de valorar el equilibrio existente entre las ventajas e inconvenientes de la actividad técnica ha de extenderse sobre los factores no estrictamente técnicos y debe traducirse en una mayor capacidad de concebir otras soluciones, tanto técnicas como de otro orden, usando materiales, principios de funcionamiento y técnicas de producción distintas o modificando el modo de uso, la ubicación o los hábitos de consumo.

7. Emplear un vocabulario adecuado para describir los útiles y técnicas empleadas en un proceso de producción o la composición de un artefacto o instalación técnica común.

Este criterio busca estimar en qué grado se han incorporado al vocabulario del alumnado términos específicos y modos de expresión, técnicamente apropiados, para describir verbalmente los procesos industriales o para describir correctamente los elementos o máquinas.

8. Montar un circuito eléctrico o neumático, a partir del plano o esquema de una instalación característica. Se pretende verificar que el alumnado es capaz de interpretar el plano de una instalación, reconocer el significado de sus símbolos, seleccionar los componentes correspondientes y conectarlos sobre un armazón o un simulador, de acuerdo a las indicaciones del plano para componer un circuito que tiene una utilidad determinada.

9. Aportar y argumentar ideas y opiniones propias del equipo de trabajo, valorando y adoptando en su caso ideas ajenas. Se trata de valorar la capacidad de contribuir con esfuerzos personales a las tareas del grupo y tomar la iniciativa para exponer y defender con talante flexible el propio punto de vista.

## Tecnología Industrial II

### 1. Materiales

- Estructura interna y propiedades de los materiales. Técnicas de modificación de las propiedades. Oxidación y corrosión. Tratamientos superficiales.
- Distintos procedimientos de ensayo y medida de materiales. Procedimientos de reciclaje de materiales, importancia económica. Normas de precaución y seguridad en el manejo de materiales.

### 2. Principios de máquinas

- Motores térmicos. Motores rotativos y alternativos. Aplicaciones. Motores eléctricos; tipos y aplicaciones. Circuito frigorífico. Bomba de calor. Elementos y

aplicaciones. Energía útil. Potencia de una máquina. Par motor en el eje. Pérdidas de energía en las máquinas. Rendimiento.

### 3. Sistemas automáticos

- Elementos que componen un sistema de control: transductores y captadores de posición, proximidad, movimiento, velocidad, presión y temperatura. Actuadores.
- Estructura de un sistema automático. Entrada, proceso, salida. Sistemas de lazo abierto. Sistemas realimentados de control. Comparadores. Montaje y experimentación en simuladores de sencillos circuitos de control.

### 4. Circuitos neumáticos y oleohidráulicos

- Técnicas de producción de fluidos. Circuitos neumáticos. Bombas y compresores de aire. Circuitos hidráulicos. Fluidos para circuitos hidráulicos. Conducción de fluidos. Caudal. Presión interior. Pérdidas. Técnicas de depuración y filtrado.
- Elementos de accionamiento. Elementos de regulación y control. Simbología y funcionamiento.
- Circuitos característicos de aplicación. Interpretación de esquemas. Montaje e instalación de circuitos.

### 5. Control y programación de sistemas automáticos

- Control analógico de sistemas. Ejemplos prácticos. Circuitos lógicos combinacionales. Puertas y funciones lógicas. Procedimientos de simplificación de circuitos lógicos. Aplicación al control del funcionamiento de un dispositivo.
- Circuitos lógicos secuenciales. Reloj. Memoria. Registros. Diagrama de fases.
- Control programado. Programación rígida y flexible. El microprocesador. El microcontrolador. El autómata programable. Aplicación al control programado de un mecanismo.

### Criterios de evaluación

1. Conocer la estructura interna de los materiales de uso más habitual y su relación con las propiedades y características de los mismos. Analizar el uso de los nuevos materiales como alternativa a los empleados tradicionalmente.

Con este criterio se trata de comprobar si el alumnado sabe aplicar los conceptos relativos a estructura interna y las técnicas de ensayo y medida de propiedades.

2. Seleccionar materiales para una aplicación práctica determinada, considerando, junto a sus propiedades intrínsecas, factores técnicos, económicos y medioambientales.

Con este criterio se trata de comprobar si el alumnado sabe seleccionar un material

idóneo para una aplicación real, conjugando con criterios de equilibrio los diversos factores que caracterizan dicha situación.

3. Diseñar un procedimiento de prueba y medida de las características de una máquina o instalación, en condiciones nominales y de uso normal.

Con este criterio se puede establecer si el alumnado es capaz de identificar los parámetros principales del funcionamiento de un artefacto o instalación, en régimen normal, y comprobar el comportamiento de dispositivos similares sometidos a pruebas metódicas para formarse una opinión propia sobre la calidad de un producto.

4. Analizar la composición de una máquina (Térmica o eléctrica) o sistema automático de uso común, identificando los elementos de mando, control, potencia y describiendo su funcionamiento. El alumnado ha de poder identificar, en un aparato medianamente complejo, los elementos que desarrollan las funciones principales y, entre ellos los responsables del control y, en su caso, explicar los principios de su programación y funcionamiento.

5. Aplicar los recursos gráficos y verbales apropiados a la descripción de la composición y funcionamiento de una máquina, circuito o sistema tecnológico concreto.

Con este criterio se quiere valorar en qué medida el alumnado utiliza, no sólo un vocabulario adecuado, sino también los conocimientos adquiridos sobre simbología y representación normalizada de circuitos, representación esquemática de ideas, relaciones entre elementos y secuencias de efectos en un sistema.

6. Montar y comprobar un sistema de control automático para su aplicación a una máquina a partir de un plano o esquema. El alumnado deberá aplicar los distintos aparatos de maniobra y protección de tipo electromecánico, electrónico, neumático e hidráulico, estudiados, a los sistemas automáticos.

En cada caso se realizará una memoria descriptiva, esquema de funcionamiento, montaje práctico y comprobación del sistema.

7. Montar y comprobar un sistema de control automático: electromecánico, electrónico, neumático e hidráulico para su aplicación a una máquina a partir de un plano o esquema.

El alumnado deberá distinguir y reconocer las ventajas e inconvenientes de cada una de ellas.

*Tabla 2. Información sobre la materia Tecnología Industrial. Fuente: DECRETO 102/2008.*



Cabe destacar que la LOMCE reconfigura la estructura de las etapas educativas descritas en los párrafos anteriores. No obstante no se tiene en cuenta para el desarrollo de este proyecto puesto que, según su calendario de implantación, las modificaciones en ESO y Bachillerato no comenzarán hasta el curso escolar 2015/2016. De todos modos, no se prevén cambios considerables en los contenidos de las materias sino en su organización en las distintas etapas educativas.

<b>Competencias básicas</b>
<b>Competencia en comunicación lingüística.</b>
Utilización del lenguaje como instrumento de comunicación oral y escrita, de representación, interpretación y comprensión de la realidad, de construcción y comunicación del conocimiento y de organización y autoregulación del pensamiento, las emociones y la conducta.
<b>Competencia matemática.</b>
Habilidad para utilizar y relacionar los números, sus operaciones básicas, los símbolos y las formas de expresión razonamiento matemático, tanto para producir e interpretar distintos tipos de información, como para ampliar el conocimiento sobre aspectos cuantitativos y espaciales de la realidad, y para resolver problemas relacionados con la vida cotidiana y con el mundo laboral.
<b>Competencia en el conocimiento y la interacción con el mundo físico.</b>
Habilidad para interactuar con el mundo físico, tanto en sus aspectos naturales como en los generados por la acción humana, de tal modo que se posibilita la comprensión de sucesos, la predicción de consecuencias y la actividad dirigida a la mejora y preservación de las condiciones de vida propia, de las demás personas y del resto de los seres vivos. En definitiva, incorpora habilidades para desenvolverse adecuadamente, con autonomía e iniciativa personal en ámbitos de la vida y del conocimiento muy diversos y para interpretar el mundo.
<b>Tratamiento de la información y competencia digital.</b>
Habilidades para buscar, obtener, procesar y comunicar información, y para transformarla en conocimiento.
<b>Competencia social y ciudadana.</b>
Comprender la realidad social en que se vive, cooperar, convivir y ejercer la ciudadanía democrática en una sociedad plural, así como comprometerse a contribuir

a su mejora.

#### Competencia cultural y artística.

Supone conocer, comprender, apreciar y valorar críticamente diferentes manifestaciones culturales y artísticas, utilizarlas como fuente de enriquecimiento y disfrute y considerarlas como parte del patrimonio de los pueblos.

#### Competencia para aprender a aprender.

Habilidades para iniciarse en el aprendizaje y ser capaz de continuar aprendiendo de manera cada vez más eficaz y autónoma de acuerdo a los propios objetivos y necesidades.

#### Autonomía e iniciativa personal.

Adquisición de la conciencia y aplicación de un conjunto de valores y actitudes personales interrelacionadas, como la responsabilidad, la perseverancia, el conocimiento de sí mismo y la autoestima, la creatividad, la autocrítica, el control emocional, la capacidad de elegir, de calcular riesgos y de afrontar los problemas, así como la capacidad de demorar la necesidad de satisfacción inmediata, de aprender de los errores y de asumir riesgos. Además supone la capacidad de elegir con criterio propio, de imaginar proyectos, y de llevar adelante las acciones necesarias para desarrollar las opciones y planes personales responsabilizándose de ellos, tanto en el ámbito personal, como social y laboral y poder transformar las ideas en acciones

*Tabla 3. Competencias básicas de la ESO. Fuente: RD 1631/2006.*

### 2.1.2. Marco social y tecnológico.

La ciudadanía del siglo XXI se enfrenta a grandes cambios constantes en su estilo de vida debido al incesante avance de la tecnología. Por ello, ha de desarrollar competencias TIC que le permitan interactuar con su entorno, cada vez más informatizado, automático y cambiante, logrando así la denominada alfabetización digital que capacita el desempeño de las actividades personales y laborales de un modo más efectivo.

Además, la creciente industria tecnológica demanda y requiere personal altamente cualificado, polivalente, creativo y autosuficiente capaz de diseñar y desarrollar proyectos complejos. Así, se hace necesario que las nuevas generaciones no se limiten a utilizar las herramientas disponibles sino a comprenderlas, diseñarlas y

manipularlas. Por esta razón, es imperativa la inclusión en la escuela de actividades orientadas a fomentar estos conocimientos y habilidades.

Consecuentemente, existen proyectos, impulsados tanto por personalidades de la industria tecnológica como por instituciones docentes, orientados a introducir la programación y la robótica en el sistema educativo y el ámbito del entretenimiento. Su objetivo es que el alumnado, futura ciudadanía, logre la comprensión del mundo artificial y la capacidad para desenvolverse, de un modo efectivo y eficaz, dentro del mismo.

## 2.2. Antecedentes.

En este apartado se estudian las herramientas y proyectos orientados a la materia de estudio del TFG: Electrónica; Control y robótica; Elementos de máquinas y sistemas; Sistemas automáticos y Control y programación de sistemas automáticos en Secundaria. En primer lugar, se describen las principales herramientas disponibles. En segundo lugar, se resumen algunos de los proyectos que fomentan el estudio aprendizaje de la programación y la robótica.

### 2.2.1. Herramientas.

Hasta hace casi dos décadas, los materiales disponibles para afrontar los contenidos pretendidos por el TFG en las escuelas han estado limitados al material impreso o digital y al uso de componentes electrónicos básicos sin posibilidad de programación. De este modo, la acción docente queda reducida a clases magistrales y sesiones prácticas en las que se sueldan componentes electrónicos sobre circuitos impresos, lo que no permite situar al alumnado en el centro del proceso de enseñanza aprendizaje.

Actualmente, se dispone de múltiples soluciones al alcance de cualquier centro educativo que, además de facilitar la acción docente permiten la experimentación y el desarrollo de proyectos de un modo sencillo y económico. Por un lado se dispone de multitud de sensores y actuadores. Por otro, se han creado diversas plataformas electrónicas que integran microcontroladores programables con los que hacer interactuar los sensores y actuadores. Éstas, pueden ser propietarias o libres.

Las primeras pertenecen a marcas comerciales y están sujetas a *copyright*. Tienen la ventaja de formar equipos robustos ya que, en la mayoría de los casos sus

componentes tan solo pueden interactuar entre ellos mismos y no con los fabricados por otras marcas, lo que también aumenta su simplicidad y facilidad de empleo. Por contra, se trata de soluciones más costosas y limitadas en cuanto a manipulación y uso.

Por su parte, las plataformas libres normalmente resultan más económicas, sus componentes pueden comunicarse y acoplarse con los de otros fabricantes siempre que sean compatibles y, dada su condición, proporcionan libertad de uso, estudio, modificación, distribución y redistribución de versiones modificadas según la licencia bajo la que se comercialicen. Como desventaja, suelen ser más complejos en su utilización dada la versatilidad que poseen.

A continuación, se analizan las principales herramientas y plataformas electrónicas que pueden encontrarse en el mercado actualmente y que son utilizadas como recursos didácticos para abordar los contenidos objeto de estudio del TFM: Electrónica; Control y robótica; Elementos de máquinas y sistemas; Sistemas automáticos y Control y programación de sistemas automáticos en Secundaria. Se presentan, por orden alfabético, divididas entre herramientas propietarias y libres.

### Herramientas propietarias

#### Circuit Lab

Una aplicación basada en navegador que permite diseñar y simular circuitos electrónicos simples y complejos bajo una interfaz amigable y fácil de usar. Facilita una mesa de trabajo virtual donde almacenar los proyectos realizados y un foro de encuentro entre la empresa y los usuarios para resolver dudas. Posee licencias especiales para instituciones educativas (1200\$ cada 6 meses) y profesores y alumnos (12\$ cada 6 meses).

- Tecnología de cableado inteligente.
- Presentación de esquemas de alta calidad.
- Creación rápida de símbolos.
- Motor propietario de simulación.
- Potente motor de gráficas.
- Recursos y expresiones algebraicas programables.



Figura 1, Circuit Lab. Fuente:  
<https://www.circuitlab.com/>

#### Crocodile Technology

Es un potente simulador de sistemas y circuitos de control orientado a la enseñanza. Con él, los estudiantes pueden diseñar y probar sus diseños de circuitos eléctricos, electrónicos, mecánicos y de control permitiendo la programación de microcontroladores (PIC). También incorpora visualización de componentes en 3D. Se adquiere bajo licencias para centros educativos que oscilan, según el número de alumnos, entre los 734€ y los 1372€.

- Diseña el circuito.
- Realiza el programa para el PIC.
- Identifica y visualiza los componentes en 3D.
- Guía el PCB.
- Simula en 3D.
- Exporta el programa realizado al PIC real.

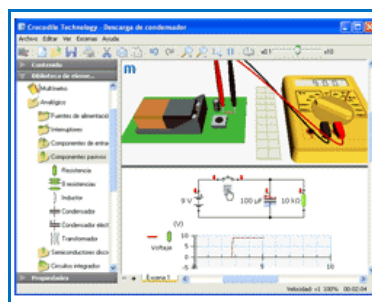


Figura 2, Crocodile Technology. Fuente: <http://www.crocodile-clips.com>

### Fischertechnik

Kits compuestos por un microprocesador central, sensores, actuadores y piezas montables y desmontables, creado por la empresa alemana Fischertechnik. Permite la creación de estructuras cuyas funcionalidades son programables desde un sistema operativo basado en Linux. Su precio es de 168 Euros y está orientado al aprendizaje autónomo mediante el juego.

- Procesador: Dual Core ARM Cortex A8.
- Memoria RAM: 128Mb DDR3.
- Memoria Flash: 64 Mb ampliables.
- Pantalla: Color, táctil de 2,4 pulgadas.
- Entradas: 8 digital /analógicas.
- Salidas para motor: 4 (9V, 250mA).
- Comunicación: Bluetooth, WiFi y USB.
- Cámara y altavoz.
- Sistema operativo: Basado en Linux.
- Alimentación: 9V DC.



Figura 3, Fischertechnik. Fuente: <http://www.fischertechnik.de>

### Legó Mindstorms

Se trata de un sistema creado por la empresa Legó, compuesto por un procesador central (bloque RCX) cuyo precio es de 155,99 libras (195,21 euros) y un conjunto de sensores y actuadores. Puede ser utilizado con las piezas de Legó clásicas formando

así todo tipo de estructuras aunque está altamente orientado a la construcción de robots o vehículos móviles.

- Microcontrolador: ARM7 32 bits.
- Memoria flash: 256Kb.
- Memoria RAM: 64Kb.
- Bluetooth: 2,0.
- USB: 12Mb/s.
- 4 Puertas de entrada de 6 contactos.
- 3 Puertas de salida de 6 contactos.
- Pantalla: 64x100 puntos.
- Altavoz: 8KHz.
- Batería: 6 pilas AA.



Figura 4, Lego Mindstorms for Education.

Fuente:

<http://shop.legoeducation.com/gb/>.

### Moway

Es una solución creada por la empresa Minirobots que pretende acercar la robótica a los centros educativos. Permite la programación de su núcleo mediante un software sencillo e intuitivo con el que controlar el robot y sus dispositivos de entrada y salida. Su precio es de 169 Euros y se enfoca principalmente a la movilidad.

- Microcontrolador: PIC18F86J50.
- Grupo de motores: I2C.
- Sensor infrarrojo anticolidión.
- Sensor de intensidad de luz direccional.
- Sensores optorreflektivos.
- Sensor de temperatura.
- Acelerómetro de 3 ejes.
- Micrófono.
- Altavoz.
- Bus de expansión SPI/I2C.
- Módulo de radiofrecuencia.
- Batería: Li-PO recargable por USB.
- Preparado para robótica colaborativa.



Figura 5. Moway. Fuente: <http://moway-robot.com>

### Herramientas libres

#### Arduino UNO R3

Arduino es una tarjeta electrónica que integra un microcontrolador Atmel Mega y un

conjunto de conexiones de entrada salida (pines) que permiten, previa programación, interactuar con el medio físico mediante sensores y actuadores electrónicos. Su precio es aproximadamente de 23 euros.

- Versión: Arduino Uno R3
- Microcontrolador: ATmega 328.
- Voltaje entrada: 7/12 V.
- Voltaje Salida: 5V.
- Frecuencia de reloj: 16MHz.
- Pines Entrada/Salida: 14.
- Entradas analógicas: 6.
- PWM (Pin de modulación ancho de pulso): 1.
- Memoria Flash: 32Kb.



Figura 6. Arduino UNO R3. Fuente: [www.http://arduino.cc/](http://arduino.cc/).

### Picaxe

Picaxe es un sistema de microcontroladores PIC fácilmente programable mediante el entorno XXXX que utiliza un lenguaje BASIC muy sencillo. También existe la posibilidad de programarlos mediante Scratch y entornos de diagramas de flujo. Inicialmente, se desarrolló para ser utilizado en Educación pero ha llegado a consolidarse como una herramienta para cualquier aficionado a la electrónica. Posee un precio aproximado de 22€.

- Nombre: Picaxe 18.
- Microcontrolador: Picaxe 18M2.
- Voltaje entrada: 3, 4,5 o 5V.
- Entradas dig.:6. (o 4 digitales y 2 analógicas).
- Salidas digitales: 8.



Figura 7, Picaxe. Fuente: <http://www.picaxe.com>

### Raspberry Pi

Una microcomputadora del tamaño de una tarjeta de crédito, con posibilidad de conectarse a una pantalla o televisión y utilizar elementos comunes como teclado y ratón. Muy orientado a la enseñanza, puede ser programado mediante el popular Scratch, entorno gráfico de programación muy utilizado en las aulas. Creada por la Raspberry Pi Foundation en Reino Unido.



Modelo B:

- CPU: ARM 1176JZF-6.
- Memoria: SDRAM 512Mb.
- Puertos: 2 USB 2.0.
- Entrada de vídeo: MIPI CSI.
- Salidas de Vídeo: RCA, HDMI, DSI.
- Salida de audio: Conector 3,5, HDMI.
- Almacenamiento: SD, MMC y SDIO.
- Red: Ethernet (RJ-45).
- Alimentación: 5V.
- Periféricos: 8xGPIO, SPI, I<sup>2</sup>C, UART.
- Sistemas Operativos: Debian, Fedora, Arch Linux, Slackware Linux, RISC OS.



Figura 8. Raspberry Pi. Fuente:  
<http://www.raspberrypi.org>

## Wiring

Wiring es una plataforma electrónica libre con su propio lenguaje de programación construido sobre Processing. Permite a los diseñadores, artistas y entusiastas de la electrónica explorar en el arte electrónico crear y programar prototipos electrónicos. Se puede utilizar para desarrollar objetos interactivos autónomos o ser conectado al software de un ordenador mediante USB. Es desarrollada por un pequeño grupo de voluntarios, su entorno de desarrollo es gratuito y la placa tiene un coste de 19,12€.

- Nombre: S Wiring Board.
- Microcontrolador: Atmel ATmega644P.
- Voltaje de entrada: 7-12V DC.
- Frecuencia de reloj: 16MHz.
- SRAM: 4096 bytes.
- EEPROM: 2048 bytes.
- Entradas analógicas: 8.
- PWM (Pin de modulación por ancho de pulso): 6.
- Memoria Flash: 65536 bytes.
- Amplificador de 5V.
- Puertos serie: 2.
- Puertos I2C: 1.

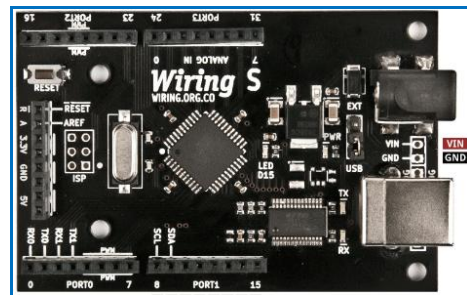


Figura 9, Wiring. Fuente:  
<http://wiring.org.co>

### 2.2.2. Proyectos.



Las necesidades de formación TIC de la ciudadanía, descritas en el apartado “Marco social y tecnológico”, junto a la especificación curricular para la integración de nuevas herramientas tecnológicas que motiven al alumnado y fomenten el uso y comprensión de las nuevas tecnologías, resumida en “Marco legal”, ha propiciado la creación de proyectos dirigidos a la enseñanza con el objetivo de introducir en el aula materias como la programación y la robótica. A continuación se muestran algunos ejemplos.

#### First Lego League

First Lego League (FLL) es un torneo que desafía a jóvenes de entre 10 y 16 años a programar un robot autónomo utilizando los componentes de Lego Mindstorms, creando así una solución innovadora a un problema previamente planteado que se encuentra relacionado con el mundo real. Su objetivo es conseguir que los asistentes se entusiasmen con la ciencia y la tecnología y adquieran valiosas habilidades para su futuro profesional y la vida.

La competición es de índole internacional y pueden participar equipos de hasta un máximo de 10 personas entre las que ha de incluirse al menos un entrenador adulto. Los torneos se dividen entre microFLL, FLL y FLL World Class. Los primeros son más simples y locales pudiendo organizarse a nivel de centro o asociación y permiten clasificar a sus vencedores para los torneos clasificatorios del segundo. En 2014, la Gran Final FLL España se ha celebrado en Barcelona.

#### Peñacorada Tecnocampus 3D

Iniciativa pionera en España enfocada a que jóvenes de entre 12 y 18 años descubran lo último en tecnología. El proyecto se enmarca en la provincia de León y en la celebración de su primer campus tecnológico, que une tecnología, educación y naturaleza, los protagonistas son la impresión 3D y la robótica *open source*. La organización cree en una educación abierta y colaborativa, por lo que sus trabajos se realizan y programan bajo licencias de código libre y GPL.

La primera edición, denominada Printer Camp, lleva a cabo sus talleres en el colegio Internacional Peñacorada y dispone de dos modalidades a escoger: Campus urbano (397€ - 497€) y Campamento (795€). En ambos se ofrece: Seguro, profesorado cualificado, material técnico, almuerzo, transporte, robot imprimible y material didáctico. La diferencia radica en que la segunda opción incluye pernocta y

actividades adicionales.

#### Asignatura de programación y comunicaciones

La Comunidad Autónoma de Madrid ha creado en 2014 una nueva materia escolar denominada Programación y comunicaciones a implantar progresivamente en todos los cursos de ESO y todos los centros de la región para el curso 2016/2017. Los objetivos fundamentales de la iniciativa son que el alumnado: Se familiarice con el entorno de programación digital, conozca los principios básicos de un lenguaje de programación, los aplique a la resolución de problemas y se introduzca en la robótica.

Para ello, se dota a los centros de: Impresoras 3D, kits de robótica, simuladores y conexión a Internet de 100Mb. Además, se prevé la organización de competiciones de robótica y aplicaciones. Asimismo, los bloques de contenidos son: Introducción a la programación con Scratch; Eventos, condicionales, bucles condicionales y variables; Proyecto Scratch; Introducción a la plataforma Arduino; Comunicación de programas con la “Shield WiFi Arduino” e Internet.

#### Code.org

Bajo el eslogan “Cada alumno de cada escuela debería tener la oportunidad de aprender ciencias de la computación” nace en 2013 la organización sin ánimo de lucro code.org. Como puede verse en su web, creen que las ciencias de la computación y la programación deberían ser parte principal del currículum educativo y para ello lanzan una serie de aplicaciones en línea y gratuitas desarrolladas acercar la materia a la docencia.

Poseen cursos elementales de programación por bloques en los que también pueden obtenerse las líneas de código generadas. Además, proporcionan al docente la herramienta “Core Studio” que proporciona un entorno de programación por bloques y de control de progreso. Actualmente, se encuentra activo el evento “Hora de código”, disponible en 30 idiomas para más de 180 países que pretende reunir del 8 al 14 de diciembre al mayor número de usuarios posibles.

#### Actualización del currículum nacional del Reino Unido

Tras el anuncio de la Real Academia de Ingeniería del Reino Unido informando de que la demanda de ingenieros en el país está creciendo mientras que el número de aspirantes disminuye, el gobierno ha actualizado el currículum nacional de educación

con el objetivo de compensar dicho desequilibrio. El Primer Ministro David Cameron se basa en la premisa de países como China e India, que poseen los mejores resultados educativos del mundo, ya tienen consolidada la materia.

De este modo, en septiembre de 2014 comienza el nuevo curso académico con nuevas asignaturas que incluyen contenidos de robótica, impresión 3D y otras tecnologías de vanguardia en los que serán entrenados los estudiantes de entre 5 y 14 años. Para comenzar, los niños con edades comprendidas entre los 5 y los 7 años (Primaria) aprenderán a programar y utilizar sistemas eléctricos simples en sus proyectos. Después, en Secundaria aprenderán modelado e impresión 3D y robótica.

### **2.3. Justificación.**

Un proyecto ha de dar solución a un problema existente. Es decir, su implementación debe responder a una necesidad. Además, si su planteamiento se basa en el uso de una herramienta a seleccionar de entre un conjunto de posibles, ésta ha de demostrar su idoneidad sobre el resto. Por ello, en este apartado se justifican tanto la necesidad del proyecto como la elección de la plataforma electrónica Arduino como instrumento vehicular del mismo.

En relación al trabajo, de lo expuesto en la introducción del presente capítulo se deduce la necesidad de crear procesos de enseñanza aprendizaje fundamentados en la búsqueda de soluciones a problemas que acontecen en el mundo real. Inclusive, el alumnado debe adoptar el rol de manipulador y creador tecnológico. Por lo tanto, es necesario adoptar una metodología dinámica y práctica que integre la manipulación y creación de objetos tecnológicos.

Bajo este marco, se consolidan y convierten en imprescindibles las materias de Tecnologías en ESO y Tecnología Industrial en Bachillerato. Pues, como establecen los correspondientes currículos oficiales, ambas tratan de fomentar el aprendizaje de conocimientos y el desarrollo de destrezas que permitan tanto la comprensión de los objetos técnicos como su utilización. Además, se centran en el uso de las nuevas tecnologías de la información y la comunicación como herramientas del proceso.

No obstante, debido a aspectos como la complejidad del material tecnológico requerido, la falta de medios o el desconocimiento de alguna de las tecnologías involucradas, existen dificultades para impartir, según la metodología requerida,

algunos contenidos de dichas asignaturas. Entre ellos, se encuentran los bloques: Electrónica; Control y robótica; Elementos de máquinas y sistemas; Sistemas automáticos y Control y programación de sistemas automáticos.

Teniendo en cuenta lo anterior, se cree conveniente el desarrollo de un proyecto para abordar el contenido expuesto mediante el uso de herramientas tecnológicas que permitan la creación y manipulación de sistemas electrónicos y de control. Así, con la producción de una serie de baterías de actividades, relacionadas con cada uno de los bloques temáticos, se pretende abordar problemáticas presentes en la vida real y situar al alumnado en el centro del proceso de enseñanza aprendizaje.

Respecto a la plataforma electrónica a seleccionar, tal y como se ha expuesto en el apartado Herramientas, existe una gran variedad de soluciones aptas para ser utilizadas como instrumentos base de las actividades. Por ello, ha de escogerse la que mejor responda al conjunto de requerimientos lectivos, funcionales y económicos. A continuación, se resumen las características por las que Arduino es considerada como la opción óptima para realizar el proyecto.

En primer lugar, se trata de una plataforma electrónica de prototipado *open source*, es decir, permite su utilización, modificación y redistribución. Consecuentemente, puede aplicarse a proyectos educativos sin restricciones de tipo comercial y, además, ser estudiada y modificada con el objetivo de adaptarla a necesidades formativas específicas. Asimismo, está compuesta de hardware y software muy económico y de sencilla manipulación.

En segundo lugar cuenta, por una parte, con una amplia comunidad de usuarios que comparten experiencias, proyectos e información y, por otra, con desarrolladores que proporcionan hardware y software compatible con el sistema que aumentan su utilidad como, por ejemplo: Scratch for Arduino, Ardublok, Virtual BreadBoard y los diversos *shields* (placas conectables) disponibles. Por ello, se considera que sus posibilidades creativas se encuentran únicamente delimitadas por la imaginación del usuario.

#### **2.4. Expectativas.**

El presente proyecto pretende dar respuesta a la metodología propuesta por las leyes y decretos en materia de educación, aplicada a las distintas materias de Tecnología en ESO y Bachillerato para los bloques: Electrónica; Control y robótica; Elementos de

máquinas y sistemas; Sistemas automáticos y Control y programación de sistemas automáticos. Así, se persigue atender, utilizando herramientas libres, a los principios de esta asignatura: Conocimientos teóricos, práctica y aplicación.

Tras la finalización del trabajo se espera obtener una memoria que comprenda las actividades agrupadas en baterías relacionadas con los distintos bloques temáticos incluidos. Asimismo, el documento forma parte del Trabajo Final de Grado de los estudios de Grado en Tecnologías de Telecomunicación: Especialidad Telemática. Por lo tanto, ha de incluirse en el repositorio de la universidad (UOC) para su libre consulta. Además, se permite su divulgación con el objetivo de ampliar su utilidad.

### 3. Capítulo 3. Diseño.

Este capítulo trata el diseño funcional y técnico de las actividades y las baterías en que éstas se agrupan en función de los distintos bloques temáticos involucrados en el proyecto. En primer lugar, en el apartado “Contenido” se define el contenido didáctico y la estructura de las baterías y prácticas. En segundo lugar, se presenta el diseño funcional de las baterías y sus actividades. Finalmente, en el apartado Diseño técnico se muestran las herramientas implicadas en el proyecto.

#### 3.1 Contenidos.

A continuación se define el material didáctico que conforma el presente proyecto y su división en baterías de actividades relacionadas con los distintos bloques temáticos objeto de estudio. Para ello, se tienen en cuenta los contenidos, objetivos y criterios de evaluación relacionados con las materias de tecnología que marcan las vigentes leyes orgánicas y decretos en materia de educación, según lo indicado en el Capítulo 2: Marco legal, del presente documento.

##### 3.1.1. Definición del contenido didáctico.

Educación Secundaria Obligatoria
Tecnologías
Bloques de contenidos
Hardware y software. Integra parte de los contenidos asociados a las tecnologías de la información y la comunicación.
Electricidad y electrónica. Estudia los fenómenos y sistemas asociados a la fuente de energía más utilizada en las máquinas, así como aquellos cuyo funcionamiento se basa en el control del flujo de los electrones.
Control y robótica. Se refiere al estudio de sistemas capaces de regular su propio comportamiento, y permite aproximar varias tecnologías entre sí.
Objetivos
1. Abordar con autonomía y creatividad, individualmente y en grupo, problemas tecnológicos trabajando de forma ordenada y metódica para estudiar el problema, recopilar y seleccionar información procedente de distintas fuentes, elaborar la documentación pertinente, concebir, diseñar, planificar y construir objetos o sistemas

que resuelvan el problema estudiado y evaluar su idoneidad desde distintos puntos de vista.

2. Adquirir destrezas técnicas y conocimientos suficientes para el análisis, diseño y elaboración de objetos y sistemas tecnológicos mediante la manipulación, de forma segura y precisa, de materiales y herramientas.

3. Analizar los objetos y sistemas técnicos para comprender su funcionamiento, conocer sus elementos y las funciones que realizan, aprender la mejor forma de usarlos y controlarlos, entender las condiciones fundamentales que han intervenido en su diseño y construcción y valorar las repercusiones que ha generado su existencia.

4. Expresar y comunicar ideas y soluciones técnicas, así como explorar su viabilidad y alcance utilizando los medios tecnológicos, recursos gráficos, la simbología y el vocabulario adecuados.

5. Adoptar actitudes favorables a la resolución de problemas técnicos, desarrollando interés y curiosidad hacia la actividad tecnológica, analizando y valorando críticamente la investigación, la innovación y el desarrollo tecnológico y su influencia en la sociedad, en el medio ambiente, en la salud y en el bienestar personal y colectivo a lo largo de la historia de la humanidad.

8. Asumir de forma crítica y activa el avance y la aparición de nuevas tecnologías, e incorporarlas a su quehacer cotidiano, analizando y valorando críticamente su influencia sobre la sociedad y el medio ambiente.

9. Actuar de forma dialogante, flexible y responsable en el trabajo en equipo, en la búsqueda de soluciones, en la toma de decisiones y en la ejecución de las tareas encomendadas con actitud de respeto, cooperación, tolerancia y solidaridad.

### Primer curso

#### Bloque 7. Electricidad y electrónica

- Introducción a la corriente eléctrica continua, definición y magnitudes básicas: voltaje, resistencia e intensidad.
- Descripción de circuitos eléctricos simples: funcionamiento y elementos. Introducción al circuito en serie y en paralelo. Simbología.
- Efectos de la corriente eléctrica: luz y calor. Análisis de objetos técnicos que apliquen estos efectos.
- Montaje de circuitos eléctricos sencillos. Utilización de esquemas, materiales y herramientas. Simuladores físicos e informáticos.

### Criterios de evaluación

13. Utilizar adecuadamente las magnitudes eléctricas básicas.
14. Valorar los efectos de la energía eléctrica y su capacidad de conversión en otras manifestaciones energéticas.
15. Identificar y utilizar correctamente los elementos fundamentales de un circuito eléctrico de corriente continua y comprender su función dentro de él.

### Tercer curso

#### Bloque 5. Electricidad y electrónica

- Ley de Ohm. Potencia y energía eléctrica.
- Circuito en serie, paralelo y mixto.
- Corriente continua y corriente alterna. Estudio comparado.
- Electromagnetismo. Aplicaciones: electroimán. Dinamo, motor de corriente continua, relé y alternador.
- Aparatos de medida: voltímetro, amperímetro y polímetro. Realización de medidas sencillas.
- Montaje de circuitos característicos.
- Introducción a la electrónica básica: la resistencia, el condensador, el diodo y el transistor. Descripción de componentes y montajes básicos.

#### Bloque 8. Control y robótica

- Introducción y evolución de los sistemas automáticos: mecanización, automatización y robotización.
- Elementos básicos de un sistema de control. Métodos y ejemplos de sistemas de control.
- Arquitectura de un robot: partes fundamentales. Sensores y actuadores.
- Control de un automatismo por ordenador. Tarjetas controladoras. Diagramas de flujo. Introducción a la programación.

### Criterios de evaluación

1. Realizar un proyecto técnico, analizando el contexto, proponiendo soluciones alternativas y desarrollando la más adecuada.
2. Elaborar los documentos técnicos necesarios para redactar un proyecto técnico, mediante el lenguaje escrito y gráfico apropiado.
3. Realizar las operaciones técnicas previstas en el proyecto técnico incorporando criterios de economía, sostenibilidad y seguridad; valorar las condiciones del entorno



de trabajo.

4. Emplear el ordenador como herramienta para elaborar, desarrollar y difundir un proyecto técnico, a través de hojas de cálculo que incorporen fórmulas y gráficas.

9. Utilizar aplicaciones de diseño asistido por ordenador para la realización de croquis normalizados.

12. Diseñar, simular y realizar montajes de circuitos eléctricos sencillos, en corriente continua, empleando pilas, interruptores, resistencias, bombillas, motores, electroimanes y relés, como respuesta a un fin predeterminado.

13. Describir las partes y el funcionamiento de las máquinas eléctricas básicas.

14. Describir y utilizar el electromagnetismo en aplicaciones tecnológicas sencillas.

15. Utilizar correctamente las magnitudes eléctricas básicas, sus instrumentos de medida y su simbología.

16. Montar un circuito electrónico sencillo empleando, al menos, diodos, transistores y resistencias, a partir de un esquema predeterminado.

24. Identificar automatismos en sistemas técnicos cotidianos, diferenciando sus elementos básicos.

25. Montar, utilizando sistemas mecánicos y eléctricos, un robot sencillo con capacidad de movimiento dirigido.

26. Realizar diagramas de flujo sencillos y conocer las órdenes más utilizadas en los programas de control.

#### Cuarto curso

##### Bloque 1. Hardware y software

- El ordenador como dispositivo de control: señales analógicas y digitales. Lógica de funcionamiento interno. Transmisión de la información por medio de señal eléctrica. Adquisición de datos. Tratamiento de la información numérica adquirida mediante hoja de cálculo.
- Programas de control.

##### Bloque 3. Electricidad y electrónica

- Descripción y análisis de sistemas electrónicos por bloques: entrada, salida y proceso. Componentes electrónicos. Circuitos integrados simples. Dispositivos de entrada: interruptores, resistencias que varían con la luz y la temperatura. Dispositivos de salida: zumbador, relé, LED y otros. Aplicaciones en montajes

sencillos.

- Técnicas de montaje y conexión de circuitos electrónicos.
- Electrónica digital. Aplicación del álgebra de Boole a problemas tecnológicos básicos. Puertas lógicas: simbología.
- Uso de simuladores para analizar el comportamiento de los circuitos electrónicos.

#### Bloque 5. Control y robótica

- Experimentación con sistemas automáticos, sensores, reguladores y actuadores. Aplicaciones prácticas. Concepto de realimentación.
- Uso del ordenador como elemento de programación y control. Lenguajes de control.
- Diseño y construcción de sistemas automáticos y desarrollo de programas para controlarlos.
- Trabajo con simuladores informáticos para comprobar el funcionamiento de los sistemas diseñados.

#### Criterios de evaluación

3. Utilizar el ordenador como herramienta de adquisición e interpretación de datos, y como realimentación de otros procesos con los datos obtenidos.
5. Describir el funcionamiento, aplicación y componentes elementales de un sistema electrónico real.
6. Diseñar, simular y montar circuitos electrónicos sencillos, utilizando la simbología adecuada.
7. Realizar operaciones lógicas empleando el álgebra de Boole, relacionar planteamientos lógicos con procesos técnicos y resolver mediante puertas lógicas problemas tecnológicos sencillos.
12. Analizar sistemas automáticos, describir sus componentes y montar automatismos sencillos.
13. Diseñar, construir y programar un sistema automático, que sea capaz de mantener su funcionamiento de forma autónoma, en función de la información que reciba del entorno mediante sensores.
14. Utilizar simuladores informáticos para verificar y comprobar el funcionamiento de los sistemas automáticos, robots y programas de control diseñados.

*Tabla 4. Información específica para el proyecto sobre la materia de Tecnologías. Fuente: DECRETO*

*112/2007*

Bachillerato
Tecnología Industrial
Objetivos
<p>1. Aplicar los conocimientos adquiridos a la comprensión y análisis del funcionamiento de máquinas y sistemas técnicos.</p> <p>4. Analizar de forma sistemática aparatos y productos de la actividad técnica para explicar su funcionamiento, utilización y forma de control evaluando su calidad.</p> <p>6. Expresar con precisión ideas y opiniones sobre procesos o productos tecnológicos concretos, utilizando vocabulario, símbolos y formas de expresión apropiadas.</p> <p>8. Actuar con autonomía y confianza al inspeccionar, manipular e intervenir en máquinas, sistemas y procesos técnicos para comprender su funcionamiento.</p>
Tecnología Industrial I
3. Elementos de máquinas y sistemas
<ul style="list-style-type: none"> <li>- Corriente continua; magnitudes fundamentales. Circuitos en corriente continua; cálculo de sus magnitudes fundamentales. Montaje y experimentación de algunos circuitos eléctricos y neumáticos característicos.</li> </ul>
Criterios de evaluación
<p>4. Emplear aparatos de medida y efectuar medidas correctamente. A la hora de diseñar cualquier proyecto nos encontramos con magnitudes dependientes de las energías objeto del proyecto; en nuestro caso, necesitamos pues conocer las magnitudes fundamentales de la energía eléctrica para poder calcular las secciones de los conductores y también los elementos de mando y protección, por lo que deberá conocerse el empleo amperímetros, voltímetros y vatímetros.</p> <p>8. Montar un circuito eléctrico o neumático, a partir del plano o esquema de una instalación característica. Se pretende verificar que el alumnado es capaz de interpretar el plano de una instalación, reconocer el significado de sus símbolos, seleccionar los componentes correspondientes y conectarlos sobre un armazón o un simulador, de acuerdo a las indicaciones del plano para componer un circuito que tiene una utilidad determinada.</p>
Tecnología Industrial II
3. Sistemas automáticos
<ul style="list-style-type: none"> <li>- Elementos que componen un sistema de control: transductores y captadores de posición, proximidad, movimiento, velocidad, presión y temperatura. Actuadores.</li> <li>- Estructura de un sistema automático. Entrada, proceso, salida. Sistemas de lazo</li> </ul>

abierto. Sistemas realimentados de control. Comparadores. Montaje y experimentación en simuladores de sencillos circuitos de control.

#### 5. Control y programación de sistemas automáticos

- Control analógico de sistemas. Ejemplos prácticos. Circuitos lógicos combinacionales. Puertas y funciones lógicas. Procedimientos de simplificación de circuitos lógicos. Aplicación al control del funcionamiento de un dispositivo.
- Circuitos lógicos secuenciales. Reloj. Memoria. Registros. Diagrama de fases.
- Control programado. Programación rígida y flexible. El microprocesador. El microcontrolador. El autómatas programable. Aplicación al control programado de un mecanismo.

#### Criterios de evaluación

4. Analizar la composición de una máquina (Térmica o eléctrica) o sistema automático de uso común, identificando los elementos de mando, control, potencia y describiendo su funcionamiento. El alumnado ha de poder identificar, en un aparato medianamente complejo, los elementos que desarrollan las funciones principales y, entre ellos los responsables del control y, en su caso, explicar los principios de su programación y funcionamiento.

5. Aplicar los recursos gráficos y verbales apropiados a la descripción de la composición y funcionamiento de una máquina, circuito o sistema tecnológico concreto.

Con este criterio se quiere valorar en qué medida el alumnado utiliza, no sólo un vocabulario adecuado, sino también los conocimientos adquiridos sobre simbología y representación normalizada de circuitos, representación esquemática de ideas, relaciones entre elementos y secuencias de efectos en un sistema.

6. Montar y comprobar un sistema de control automático para su aplicación a una máquina a partir de un plano o esquema. El alumnado deberá aplicar los distintos aparatos de maniobra y protección de tipo electromecánico, electrónico, neumático e hidráulico, estudiados, a los sistemas automáticos.

En cada caso se realizará una memoria descriptiva, esquema de funcionamiento, montaje práctico y comprobación del sistema.

7. Montar y comprobar un sistema de control automático: electromecánico, electrónico, neumático e hidráulico para su aplicación a una máquina a partir de un plano o esquema.

El alumnado deberá distinguir y reconocer las ventajas e inconvenientes de cada una

de ellas.

En cada caso se realizará una memoria descriptiva, esquema de funcionamiento, montaje práctico y comprobación del sistema.

*Tabla 5. Información específica para el proyecto sobre la materia Tecnología Industrial. Fuente: DECRETO 102/2008.*

A continuación se explica cómo quedan implicadas las competencias básicas (Competencia en comunicación lingüística; Competencia matemática; Competencia en el conocimiento y la interacción con el mundo físico; Tratamiento de la información y competencia digital; Competencia social y ciudadana; Competencia para aprender a aprender y autonomía e iniciativa personal) en el desarrollo de la propuesta educativa del proyecto.

- Competencia en comunicación lingüística. Se practica mediante la lectura y escritura de documentos sobre los que trabajar para la realización de las actividades.
- Competencia matemática. Se ejercita mediante el uso de la lógica en programación, el cálculo de datos, el manejo de escalas y la elaboración de presupuestos.
- Competencia en el conocimiento y la interacción con el mundo físico. Se logra resolviendo problemas en espacios determinados en los que intervienen objetos.
- Tratamiento de la información y competencia digital. Se impulsa al utilizar herramientas TIC que comprenden tanto el uso de *hardware* como de *software* y la búsqueda de información en la red.
- Competencia social y ciudadana. Se fomenta al trabajar en grupos de dos alumnos propiciando así la cooperación.
- Competencia para aprender a aprender. Se promueve ante la necesidad de indagar en la búsqueda de soluciones para los problemas planteados en las prácticas y, sobre todo, en el proyecto.
- Autonomía e iniciativa personal. Se adquiere responsabilidad, perseverancia, conocimiento de sí mismo y creatividad así como capacidad para asumir riesgos y afrontar problemas con el desarrollo de proyectos y el trabajo en equipo.

### **3.1.2. Definición de baterías y prácticas.**

Las baterías de actividades conforman series de prácticas, secuenciales o individuales, orientadas a abordar los contenidos y objetivos de los bloques temáticos relacionados con el proyecto. Asimismo, tras la realización de cada actividad contenida en una batería se alcanza una parte de los aspectos pretendidos globalmente. Por lo tanto, cada batería está relacionada con diversos propósitos del currículo y cada actividad alcanza los mismos parcialmente.

La estructura descriptiva de las baterías es: título (“Batería” + etapa educativa -1 para ESO y 2 para Bachillerato- + curso + batería + nombre); Bloques del currículo cubiertos; Objetivos de etapa alcanzados; Criterios de evaluación y actividades contenidas. Por otra parte, la de las prácticas es: “Actividad” + batería a que pertenece + actividad; Bloques relacionados; Objetivos específicos alcanzados; Criterios de evaluación, Requisitos previos y material necesario.

Batería X.Y.Z	
Bloques relacionados:	1, 2, ..., n.
Objetivos de etapa:	1, 2, ..., n.
Criterios de evaluación:	1, 2, ..., n.
Prácticas:	X.Y.Z.1, X.Y.Z.2, ..., X.Y.Z.n.

Tabla 6, Formato de la batería de actividades.

Práctica X.Y.Z.i	
Bloques relacionados:	1, 2, ..., n.
Objetivos de etapa:	1, 2, ..., n.
Criterios de evaluación:	1, 2, ..., n.
Requisitos previos:	
Material necesario:	

Tabla 7, Formato de las actividades.

### 3.2. Diseño funcional.

El diseño funcional del conjunto de actividades del proyecto representa la correspondencia entre las prácticas y las baterías, su orden y sus características. De este modo, en el primer apartado (Esquema) se presenta el diagrama de actividades y, en el punto 2 (Descripción), la descripción de las mismas mediante la implementación de la estructura descriptiva detallada en el apartado “Definición de baterías y actividades”.

### 3.2.1. Esquema.

Batería 1.1.1. Introducción al montaje de circuitos sobre protoboard.	Práctica 1.1.1.1. Introducción a la protoboard I. Práctica 1.1.1.2. Introducción a la protoboard II.
Batería 1.3.1. Introducción a la electricidad y electrónica.	Práctica 1.3.1.1. LED intermitente. Práctica 1.3.1.2. Regulación lumínica. Práctica 1.3.1.3. Relé, diodo y pulsador. Práctica 1.3.1.4. El condensador temporizador. Práctica 1.3.1.5. Circuito en serie. Práctica 1.3.1.6. Circuito en paralelo. Práctica 1.3.1.7. Circuito mixto.
Batería 1.3.2. Introducción al control y la robótica.	Práctica 1.3.2.1. Semáforo simple. Práctica 1.3.2.2. Motores con Arduino Motor Shield. Práctica 1.3.2.3. Control LED por bluetooth. Práctica 1.3.2.4. Control de luminosidad. Práctica 1.3.2.5. Vehículo teledirigido.
Batería 1.4.1. Electricidad y electrónica.	Práctica 1.4.1.1. Alarma LDR. Práctica 1.4.1.2. Termómetro LED. Práctica 1.4.1.3. Servomotor con sensor de flexión. Práctica 1.4.1.4. Lógica con pulsadores. Práctica 1.4.1.5. Semáforo avanzado.
Batería 1.4.2. Control y robótica.	Práctica 1.4.2.1. Ayuda en estacionamiento. Práctica 1.4.2.2. Puerta automática. Práctica 1.4.2.3. Proyecto libre: Vehículo autónomo.
Batería 2.2.1. Sistemas automáticos.	Práctica 2.2.1.1. Piano. Práctica 2.2.1.2. Contador de 0 a 9. Práctica 2.2.1.3. Puertas lógicas. Práctica 2.2.1.4. Circuitos combinacionales: El sumador.

Tabla 8. Disposición de las baterías de actividades del proyecto.

### 3.2.2. Descripción de baterías y prácticas.

<b>Batería 1.1.1. Introducción al montaje de circuitos sobre protoboard.</b>	
Bloques relacionados:	7.
Objetivos de etapa:	1, 2, 3 y 5.
Criterios de evaluación:	13, 14 y 15.
1.1.1.1. Introducción a la protoboard I.	
1.1.1.2. Introducción a la protoboard II.	

Tabla 9, Batería 1.1.1. Introducción al montaje de circuitos sobre protoboard.

<b>Práctica 1.1.1.1. Introducción a la protoboard I.</b>	
Bloques relacionados:	7.
Objetivos de etapa:	1, 2, 3 y 5.
Criterios de evaluación:	13, 14 y 15.

Requisitos previos:	La <i>proto</i> board.
Material necesario:	<ul style="list-style-type: none"> <li>• 2 pila de 1,2v.</li> <li>• 1 Porta pilas</li> <li>• 1 <i>Proto</i>board.</li> <li>• Cable de cobre.</li> <li>• 1 Interruptor.</li> <li>• 2 Portalámparas.</li> <li>• 2 Bombillas de 2,4V.</li> </ul>

Tabla 10, Práctica 1.1.1.1. Introducción a la *proto*board I.

Práctica 1.1.1.2. Introducción a la <i>proto</i> board II.	
Bloques relacionados:	7.
Objetivos de etapa:	1, 2, 3 y 5.
Criterios de evaluación:	13, 14 y 15.
Requisitos previos:	La <i>proto</i> board.
Material necesario:	<ul style="list-style-type: none"> <li>• 2 pila de 1,2v.</li> <li>• 1 Porta pilas</li> <li>• 1 <i>Proto</i>board.</li> <li>• Cable de cobre.</li> <li>• 1 Interruptor.</li> <li>• 2 Portalámparas.</li> <li>• 2 Bombillas de 2,4V.</li> </ul>

Tabla 11, Práctica 1.1.1.2. Introducción a la *proto*board II

Batería 1.3.1. Introducción a la electricidad y electrónica.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
1.3.1.1. LED Intermitente. 1.3.1.2. Regulación lumínica. 1.3.1.3. Relé, diodo y pulsador. 1.3.1.4. El condensador temporizador. 1.3.1.5. Circuito serie. 1.3.1.6. Circuito paralelo. 1.3.1.7. Circuito mixto.	

Tabla 12, Batería 1.3.1. Introducción a la electricidad y la electrónica.

Práctica 1.3.1.1. LED Intermitente.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>proto</i> board; La resistencia; Tipos de



	circuitos; Ley de Ohm y El LED.	
Material necesario:		
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Diodo LED.</li> <li>• 1 Resistencia de 330Ω.</li> <li>• 2 Cables de conexión macho - macho.</li> </ul>	

Tabla 13, Práctica 1.3.1.1. LED intermitente.

Práctica 1.3.1.2. Regulación lumínica.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; El potenciómetro; Tipos de circuitos, El LED y Ley de Ohm.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Diodo LED.</li> <li>• 1 Resistencia de 330Ω.</li> <li>• 1 Potenciómetro de 10kΩ.</li> <li>• 5 Cables de conexión macho - macho.</li> </ul>

Tabla 14, Práctica 1.3.1.2. Regulación lumínica.

Práctica 1.3.1.3. Relé, diodo y pulsador.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; Tipos de circuitos; Ley de Ohm, El relé, El pulsador y El diodo.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador/fuente 5V.</li> <li>• 1 Cable USB (PC).</li> <li>• 1 Pulsador.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 LED rojo.</li> <li>• 1 Resistencia de 330Ω.</li> <li>• 1 Diodo 1N4148.</li> <li>• 7 Cables de conexión macho - macho.</li> <li>• 1 Relé.</li> </ul>

Tabla 15, Práctica 1.3.1.3. Relé, diodo y pulsador.

Práctica 1.3.1.4. El condensador temporizador.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).

Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; Tipos de circuitos; Ley de Ohm, El pulsador, El condensador y El diodo.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Conmutador.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 LED.</li> <li>• 1 Resistencia de 330<math>\Omega</math>.</li> <li>• 2 Resistencia de 10k<math>\Omega</math>.</li> <li>• 6 Cables de conexión macho - macho.</li> <li>• 1 Condensador 470<math>\mu</math>F.</li> </ul>

Tabla 16, Práctica 1.3.1.4. El condensador temporizador.

Práctica 1.3.1.5. Circuito serie.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; Tipos de circuitos y Ley de Ohm.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Resistencia de 2,2k<math>\Omega</math>.</li> <li>• 2 Resistencias de 1k<math>\Omega</math>.</li> <li>• 1 Resistencia 560<math>\Omega</math>.</li> <li>• 7 Cables de conexión macho - macho.</li> </ul>

Tabla 17, Práctica 1.3.1.5. Circuito serie.

Práctica 1.3.1.6. Circuito paralelo.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; Tipos de circuitos y Ley de Ohm.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Resistencia de 2,2k<math>\Omega</math>.</li> <li>• 1 Resistencias de 1k<math>\Omega</math>.</li> <li>• 1 Resistencia 560<math>\Omega</math>.</li> <li>• 4 Cables de conexión macho - macho.</li> </ul>

Tabla 18, Práctica 1.3.1.6. Circuito paralelo.

Práctica 1.3.1.7. Circuito mixto.	
Bloques relacionados:	5.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15,16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; Tipos de circuitos y Ley de Ohm.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Resistencia de 2,2k<math>\Omega</math>.</li> <li>• 1 Resistencias de 1k<math>\Omega</math>.</li> <li>• 1 Resistencia 560<math>\Omega</math>.</li> <li>• 4 Cables de conexión macho - macho.</li> </ul>

Tabla 19, Práctica 1.3.1.7. Circuito mixto.

Batería 1.3.2. Introducción al control y la robótica.	
Bloques relacionados:	5 y 8.
Objetivos de etapa:	1, 2, 3, 4, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	1, 2, 3, 4, 9, 12, 13, 14, 15, 16, 24, 25 y 26.
1.3.2.1. Semáforo simple. 1.3.2.2. Motores con Arduino Motor Shield. 1.3.2.3. Control LED por bluetooth. 1.3.2.4. Control de luminosidad. 1.3.2.5. Vehículo teledirigido.	

Tabla 20, Batería 1.3.2. Introducción al control y la robótica.

Práctica 1.3.2.1. Semáforo simple.	
Bloques relacionados:	5 y 8.
Objetivos de etapa:	1, 2, 3, 4, 5 y 9 (si se trabaja en equipo).
Criterios de evaluación:	2, 9, 16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; Tipos de circuitos; El LED y Fritzing.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 3 Diodos LED (Rojo, verde y amarillo).</li> <li>• 3 Resistencias de 330<math>\Omega</math>.</li> <li>• 4 Cables de conexión macho - macho.</li> <li>• Fritzing.</li> </ul>

Tabla 21, Práctica 1.3.2.1. Semáforo simple.

Práctica 1.3.2.2. Motores con Arduino Motor Shield.	
Bloques relacionados:	5 y 8.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 16, 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La protoboard; La resistencia; El motor, El LED, La Arduino Motor Shield y Tipos de circuitos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Arduino Motor Shield R3.</li> <li>• 3 Diodos LED (Rojo, verde y amarillo).</li> <li>• 3 Resistencias de 330Ω.</li> <li>• 4 Cables de conexión macho - macho.</li> </ul>

Tabla 22, Práctica 1.3.2.2. Motores con Arduino Motor Shield.

Práctica 1.3.2.3. Control LED por bluetooth.	
Bloques relacionados:	5 y 8.
Objetivos de etapa:	1, 2, 3, 5 y 9 (si se trabaja en equipo).
Criterios de evaluación:	1, 16 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La protoboard; La resistencia; El LED y Tipos de circuitos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Adafruit EZ-LINK.</li> <li>• Blue Control.</li> </ul>	<ul style="list-style-type: none"> <li>• 4 Diodos LED Verdes.</li> <li>• 1 Diodo LED Rojo.</li> <li>• 5 Resistencias de 330Ω.</li> <li>• 15 Cables de conexión macho - macho.</li> <li>• Fritzing.</li> <li>• Smartphone.</li> </ul>

Tabla 23, Práctica 1.3.2.3. Control LED por bluetooth.

Práctica 1.3.2.4. Control de luminosidad.	
Bloques relacionados:	5 y 8.
Objetivos de etapa:	2, 3, 5, 8 y 9 (si trabajan en grupos).
Criterios de evaluación:	12, 15 y 16
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La protoboard; La resistencia; El potenciómetro; Tipos de circuitos, El LED, EL LDR, El transistor y Ley de Ohm.
Material necesario:	

<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Transistor NPN.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Diodo LED.</li> <li>• 1 Resistencia de 10k<math>\Omega</math>.</li> <li>• 1 LDR.</li> <li>• 7 Cables de conexión macho - macho.</li> </ul>
---	---

Tabla 24, Práctica 1.3.2.4. Control de luminosidad.

Práctica 1.3.2.5. Vehículo teledirigido.	
Bloques relacionados:	5 y 8.
Objetivos de etapa:	1, 2, 3, 4, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	1, 2, 3, 4, 9, 12, 13, 14, 15, 16, 24, 25 y 26.
Requisitos previos:	Introducción a Arduino; Entorno de programación Arduino; La <i>protoboard</i> ; La resistencia; El LED; El motor; Tipos de circuitos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Adafruit EZ-LINK.</li> <li>• 1 Arduino Motor Shield.</li> <li>• Estructura necesaria.</li> <li>• 2 motores DC.</li> </ul>	<ul style="list-style-type: none"> <li>• 2 Diodos LED Amarillos.</li> <li>• 1 Diodo LED Rojo.</li> <li>• 1 Diodo LED Verde</li> <li>• 4 Resistencias de 330<math>\Omega</math>.</li> <li>• 15 Cables de conexión macho - macho.</li> <li>• Smartphone.</li> <li>• Fritzing.</li> <li>• Blue Control.</li> </ul>

Tabla 25, Práctica 1.3.2.5. Vehículo teledirigido.

Batería 1.4.1. Electricidad y electrónica.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 7, 12, 13 y 14.
1.4.1.1. Alarma LDR. 1.4.1.2. Termómetro LED. 1.4.1.3. Servomotor con sensor de flexión. 1.4.1.4. Lógica con pulsadores. 1.4.1.5. Semáforo avanzado.	

Tabla 26, Batería 1.4.1. Electricidad y electrónica.

Práctica 1.4.1.1. Alarma LDR.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 13 y 14.

Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el LDR y el zumbador piezoeléctrico.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 LDR.</li> <li>• 1 Resistencia de 10kΩ.</li> <li>• 1 zumbador (Buzzer) piezoeléctrico.</li> <li>• 6 Cables de conexión macho - macho.</li> </ul>

Tabla 27, Práctica 1.4.1.1.Alarma LDR.

Práctica 1.4.1.2. Termómetro LED.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 13 y 14.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el LED y el sensor de temperatura.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 3 LED rojos.</li> </ul>	<ul style="list-style-type: none"> <li>• 3 LED amarillos.</li> <li>• 3 LED verdes.</li> <li>• 9 Resistencias 330Ω.</li> <li>• 13 Cables de conexión macho - macho.</li> <li>• Sensor de temperatura.</li> </ul>

Tabla 28, Práctica 1.4.1.2.Termómetro LED.

Práctica 1.4.1.3. Servomotor con sensor de flexión.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 13 y 14.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el sensor de flexión y el servomotor.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Sensor de flexión.</li> <li>• 1 Resistencia de 10kΩ.</li> <li>• 1 Servomotor.</li> <li>• 11 Cables de conexión macho – macho.</li> </ul>

Tabla 29, Práctica 1.4.1.3.Servomotor con sensor de flexión.

Práctica 1.4.1.4. Lógica con pulsadores.
--

Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 7, 13 y 14.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el pulsador, el LED y Álgebra de boole.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 2 Pulsadores.</li> <li>• 1 LED rojo.</li> <li>• 1 LED verde.</li> <li>• 2 Resistencia de 330Ω.</li> <li>• 2 Resistencias de 10kΩ.</li> <li>• 8 Cables de conexión macho – macho.</li> </ul>	

Tabla 30, Práctica 1.4.1.4.Lógica con pulsadores.

Práctica 1.4.1.5. Semáforo avanzado.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 7, 12, 13 y 14.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el pulsador, el LED y Álgebra de boole.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Pulsador.</li> <li>• 1 LED Amarillo.</li> <li>• 2 LED rojos.</li> <li>• 2 LED verdes</li> <li>• 5 Resistencias de 330Ω.</li> <li>• 1 Resistencia de 10kΩ.</li> <li>• 9 Cables de conexión macho - macho.</li> </ul>	

Tabla 31, Práctica 1.4.1.5.Semáforo avanzado.

Batería 1.4.2. Control y robótica.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 7, 12, 13 y 14.
1.4.2.1. Ayuda en estacionamiento.	
1.4.2.2. Puerta automática.	
1.4.2.3. Proyecto libre: Vehículo autónomo.	

Tabla 32, Batería 1.4.2. Control y robótica.

## Práctica 1.4.2.1. Ayuda en estacionamiento.

Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 13 y 14.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el pulsador, el LED y el sensor de ultrasonidos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Módulo HC-SR04.</li> <li>• 1 Zumbador.</li> <li>• 2 LED Verdes.</li> <li>• 2 LED Amarillos.</li> <li>• 3 LED Rojos.</li> <li>• 7 Resistencias de 330Ω.</li> <li>• 15 Cables de conexión macho – macho.</li> </ul>	

Tabla 33, Práctica 1.4.2.1. Ayuda en estacionamiento.

Práctica 1.4.2.2. Puerta automática.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 13 y 14.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, La resistencia, el pulsador, el LED y el sensor de infrarrojos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Servomotor.</li> <li>• 1 Sensor IR SHARP.</li> <li>• 8 Cables de conexión macho - macho.</li> </ul>	

Tabla 34, Práctica 1.4.2.2. Puerta automática.

Práctica 1.4.2.3. Proyecto libre: Vehículo autónomo.	
Bloques relacionados:	1, 3 y 5.
Objetivos de etapa:	1, 2, 3, 4, 5, 8 y 9 (si se trabaja en equipo).
Criterios de evaluación:	3, 5, 6, 7, 12, 13 y 14.
Requisitos previos:	Prácticas 1.4.2.1 y 1.4.2.2.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 1 Arduino motor shield.</li> <li>• 1 Módulo HC-SR04.</li> <li>• 2 Motores DC.</li> <li>• 2 LED rojos.</li> <li>• 2 LED verdes..</li> <li>• 2 Resistencias de 330Ω.</li> <li>• 11 Cables de conexión macho - macho</li> <li>• 6 Cables de conexión comunes.</li> </ul>	



- Fuente alimentación 5V.

Tabla 35, Práctica 1.4.2.3. Proyecto libre: Vehículo autónomo.

<b>Batería 2.2.1. Sistemas automáticos.</b>	
Bloques relacionados:	3 y 5.
Objetivos de etapa:	1, 4, 6 y 8.
Criterios de evaluación:	4, 5, 6 y 7.
2.2.1.1. Piano.	
2.2.1.2. Contador de 0 a 9.	
2.2.1.3. Puertas lógicas.	
2.2.1.4. Circuitos combinacionales: El sumador.	

Tabla 36, Batería 2.2.1. Sistemas automáticos.

<b>Práctica 2.2.1.1. Piano.</b>	
Bloques relacionados:	3 y 5.
Objetivos de etapa:	1, 4, 6 y 8.
Criterios de evaluación:	4, 5, 6 y 7.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, la resistencia, el zumbador piezoeléctrico y el sensor de presión.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Resistencia de 10k<math>\Omega</math>.</li> <li>• 1 Zumbador piezo eléctrico.</li> <li>• 1 Sensor de presión.</li> <li>• 7 Cables de conexión macho - macho</li> </ul>

Tabla 37, Práctica 2.2.1.1. Piano.

<b>Práctica 2.2.1.2. Contador de 0 a 9.</b>	
Bloques relacionados:	3 y 5.
Objetivos de etapa:	1, 4, 6 y 8.
Criterios de evaluación:	4, 5, 6 y 7.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, la resistencia y el display de 7 segmentos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Resistencia de 330<math>\Omega</math>.</li> <li>• 1 Display 7 segmentos D5611 A/B.</li> <li>• 8 Cables de conexión macho - macho</li> </ul>

- 1 Cable USB.

Tabla 38, Práctica 2.2.1.2. Contador de 0 a 9.

Práctica 2.2.1.3. Puertas lógicas.	
Bloques relacionados:	3 y 5.
Objetivos de etapa:	1, 4, 6 y 8.
Criterios de evaluación:	4, 5, 6 y 7.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, la resistencia, el pulsador y el display de 7 segmentos.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> <li>• 2 Pulsadores.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 Resistencia de 330Ω.</li> <li>• 2 Resistencias de 10kΩ.</li> <li>• 1 Display 7 segmentos D5611 A/B.</li> <li>• 14 Cables de conexión macho – macho.</li> </ul>

Tabla 39, Práctica 2.2.1.3. Puertas lógicas.

Práctica 2.2.1.4. Circuitos combinacionales: El sumador.	
Bloques relacionados:	3 y 5.
Objetivos de etapa:	1, 4, 6 y 8.
Criterios de evaluación:	4, 5, 6 y 7.
Requisitos previos:	Introducción a Arduino, entorno de programación Arduino, la protoboard, la resistencia, el LED y el CI sumador.
Material necesario:	
<ul style="list-style-type: none"> <li>• 1 Placa Arduino UNO.</li> <li>• 1 <i>Protoboard</i>.</li> <li>• 1 Ordenador.</li> <li>• 1 Cable USB.</li> </ul>	<ul style="list-style-type: none"> <li>• 1 CI 74283 (DM74LS283).</li> <li>• 1 LED rojo.</li> <li>• 13 Resistencias de 330Ω.</li> <li>• 26 Cables de conexión macho – macho.</li> </ul>

Tabla 40, Práctica 2.2.1.4. Circuitos combinacionales: El sumador.

Cabe destacar que no se ha diseñado una batería específica para el primer curso de Bachillerato debido a que sus contenidos y criterios relacionados con el presente proyecto son abordados en las baterías dispuestas para los cursos tercero y cuarto de ESO. Por este motivo, si se desea impartir docencia en este nivel educativo utilizando las prácticas aquí elaboradas, es posible seleccionarlas de las baterías 1.3.1, 1.3.2, 1.4.1 y 1.4.2.

### 3.3. Diseño técnico.

En este apartado se proporciona información concreta sobre todos los elementos necesarios para la realización de las actividades del proyecto. Cada herramienta incluye: Nombre, descripción y características principales. Además, se presentan divididas en dos secciones según su naturaleza: Software o Hardware. De este modo, se facilita la tarea de selección de partes específicas del material para la programación de sesiones lectivas.

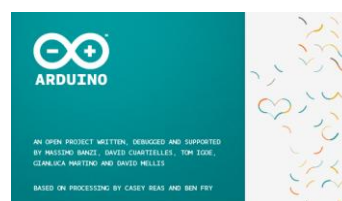
### 3.3.1. Software

**Título:** Arduino IDE.

**Autores:** Massimo Banzi, Tom Igoe y David Cuartielles.

**Versión:** 1.0.6.

**Fuente de información:** [www.arduino.cc](http://www.arduino.cc).



*Figura 10, Arduino.IDE*

*Fuente: Arduino.cc.*

**Resumen:** Arduino es la tarjeta electrónica que se utiliza como herramienta vehicular de las prácticas contenidas en el proyecto. Integra un microcontrolador ATmega328 y un conjunto de conexiones de entrada salida (pines) que permiten interactuar con sensores y actuadores electrónicos. Para programar las órdenes que el microcontrolador ha de ejecutar se utiliza el Entorno de Desarrollo Integrado (IDE) de Arduino, que actualmente se encuentra en su versión 1.0.6..

**Título:** Fritzing.

**Autores:** Friends of Fritzing Foundation y IXDS.

**Versión:** 0.9.0b.

**Fuente de información:** <http://fritzing.org>.



*Figura 11. Fritzing. Fuente: fritzing.org.*

**Resumen:** Como se indica en la propia web de la aplicación, Fritzing es una iniciativa de *hardware* abierto que pretende hacer accesible la electrónica como material creativo para cualquiera. Ofrecen una herramienta de diseño gratuita, enfocada a la placa Arduino y sus principales elementos actuadores y sensores, que permite a sus usuarios documentar de modo gráfico sus prototipos, compartirlos y enseñar electrónica en el aula.

### 3.3.2. Hardware.

**Título:** Arduino UNO.

**Autores:** Massimo Banzi, Tom Igoe y David Cuartielles.

**Versión:** Arduino UNO R3.

**Fuente de información:** [www.arduino.cc](http://www.arduino.cc).



Figura 12, Arduino. Fuente: [Arduino.cc](http://www.arduino.cc).

**Resumen:** Arduino UNO R3 es una tarjeta electrónica que integra un microcontrolador Atmel Mega y un conjunto de conexiones de entrada salida (pines) que permiten, previa programación, interactuar con el medio físico mediante sensores y actuadores electrónicos. Se trata de la herramienta principal del proyecto sobre la que se conectarán elementos hardware y se programarán interacciones. Por ello, forma parte de todas las baterías de actividades.

- Versión: Arduino Uno R3
- Voltaje entrada: 7/12 V.
- Frecuencia de reloj: 16MHz.
- Entradas analógicas: 6.
- PWM (Pin de modulación ancho de pulso).
- Microcontrolador: ATMega 328
- Voltaje Salida: 5V.
- Pins Entrada/Salida: 14.
- Memoria Flash: 32Kb.

**Título:** Arduino Motor Shield.

**Versión:** R3.

**Fuente de información:** [www.arduino.cc](http://www.arduino.cc).

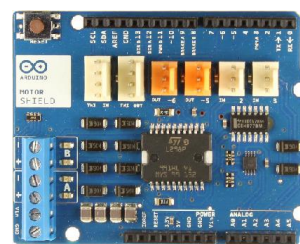


Figura 13, Arduino Motor Shield Fuente: [Arduino.cc](http://www.arduino.cc).

**Resumen:** Arduino Motor Shield es una placa que se acopla sobre Arduino UNO y sirve para interactuar con motores de corriente continua o paso a paso de un modo sencillo. Gracias a esta placa, es posible implementar proyectos en los que participen motores DC sin necesidad de manipular circuitos en H para operar sobre el sentido de

giro, característica muy deseable en etapas educativas donde no se ha estudiado electrónica con cierta complejidad.

Voltaje operativo: 5V a 12V.

Corriente máxima: 2A/Canal o 4A/Canal con suministro de energía externa.

Controlador de motor: L298P, hasta 2 motores DC o 1 paso a paso.

<b>Función</b>	<b>Pines canal A</b>	<b>Pines canal B.</b>
Dirección	D12	D13
PWM	D3	D11
Freno	D9	D8

**Título:** Protoboard.

**Resumen:** La protoboard o placa de pruebas es una herramienta que sirve para conectar elementos electrónicos sin necesidad de soldadura. Esta propiedad es indispensable en un entorno de montaje y desmontaje de circuitos como el acontecido en las aulas.

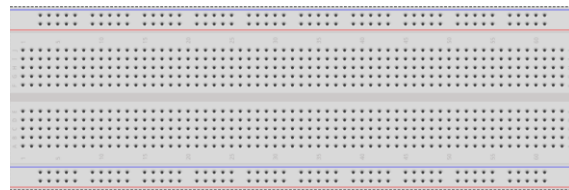
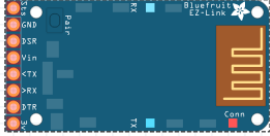


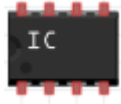

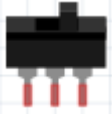


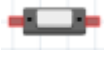



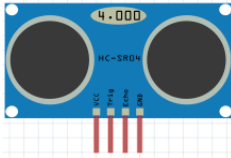






Figura 14, Protoboard. Fuente: Elaboración propia.

**Título:** Elementos electrónicos básicos.

Adafruit EZ-Link.	
Bombilla 2,4V.	
Cables.	
CI 74283 (DM74LS283).	

<p>Condensador 470<math>\mu</math>F.</p>	
<p>Conmutador.</p>	
<p>Diodo 1N4148.</p>	
<p>Display de 7 segmentos D5611 A/B.</p>	
<p>Interruptor.</p>	
<p>Fotorresistor (LDR).</p>	
<p>LED (Rojo, verde y amarillo).</p>	
<p>Motor DC.</p>	
<p>Módulo ultrasonidos (HC-SR04).</p>	
<p>Pila 1,2V.</p>	
<p>Portalámparas miniatura.</p>	
<p>Porta pilas.</p>	
<p>Potenciómetro 10k<math>\Omega</math>.</p>	

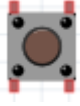




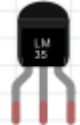
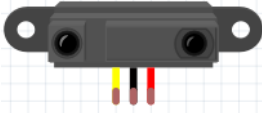
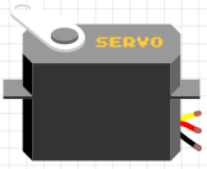

Pulsador.	
Relé.	
Resistencias 330Ω, 560Ω, 1kΩ, 2,2kΩ y 10kΩ,	
Sensor de flexión.	
Sensor de presión.	
Sensor de temperatura.	
Sensor IR Sharp.	
Servomotor.	
Zumbador piezoeléctrico.	

Tabla 41, Elementos electrónicos.

## Capítulo 4. Implementación.

En los siguientes apartados se ofrecen las distintas fases de implementación del proyecto de modo gráfico. En el primer punto (Diagrama del proyecto), se obtiene una visión global de todo el proyecto. En segundo lugar, en “Bloques del proyecto” se detallan los contenidos de cada capítulo en que se divide el trabajo. Por último, se presenta la calendarización con el orden y tiempos establecidos para la realización de los capítulos.

### 4.1. Diagrama del proyecto.

Tecnología en Secundaria con Arduino							
Capítulo 1: Introducción	Capítulo 2: Estado del arte	Capítulo 3: Diseño	Capítulo 4: Implementación	Capítulo 5: Resultado	Capítulo 6: Conclusiones	Memoria	Presentación
Presentación	Introducción	Contenidos	Diagrama del proyecto	Prácticas	Conclusiones	Abstract	Presentación
Objetivos	Antecedentes	Diseño funcional	Bloques del proyecto		Líneas de futuro	Resumen	
Beneficios	Justificación	Diseño técnico	Calendarización			Agradecimientos	
Entregables	Expectativas					Índices	
Motivación						Glosario	
						Memoria	
24/09/2014 - 08/10/2014	15/10/2014 - 04/11/2014	05/11/2014 - 19/11/2014	20/11/2014 - 29/11/2014	30/11/2014 - 30/12/2014	31/12/2014 - 04/01/2015	05/01/2015 - 11/01/2015	12/01/2015 - 18/01/2015
24/09/2014 – 18/01/2015							

Tabla 42. Diagrama del proyecto.

### 4.2. Bloques del proyecto.

Capítulo 1 - Introducción					
Presentación	Objetivos	Beneficios	Entregables	Motivación	Recursos
Presentación del proyecto	Objetivos generales	Beneficios derivados del proyecto	Documentos finales a obtener	Motivación personal	Recursos necesarios para elaborar el proyecto
	Objetivos específicos				
24/09/2014 – 26/09/2014	27/09/2014 – 29/09/2014	30/09/2014 – 01/10/2014	02/10/2014 – 03/10/2014	04/10/2014 – 05/10/2014	06/10/2014 – 08/10/2014
24/09/2014 – 08/10/2014					

Tabla 43. Contenido del capítulo 1.



Capítulo 2 – Estado del arte.			
Introducción	Antecedentes	Justificación	Expectativas
Marco legal	Herramientas para la materia	¿Por qué un proyecto así?	¿Qué se espera?
Marco social y tecnológico	Proyectos relacionados	¿Qué aporta Arduino?	
15/10/2014 – 25/10/2014	26/10/2014 – 30/10/2014	31/10/2014 – 2/11/2014	3/11/2014 - 4/11/2014
15/10/2014 – 04/11/2014			

Tabla 44. Fases del capítulo 2.

Capítulo 3 - Diseño		
Contenidos	Diseño funcional	Diseño técnico
Definición del contenido didáctico	Esquema	Software necesario
Definición de las baterías	Baterías de actividades (Relación con el currículum y actividades)	Hardware necesario
Definición de las prácticas	Actividades (Objetivos y criterios de evaluación)	
05/11/2014 – 10/11/2014	11/11/2014 – 15/11/2014	16/11/2014 – 19/11/2014
05/11/2014 – 19/11/2014		

Tabla 45. Contenido del capítulo 3.

Capítulo 4 - Implementación		
Diagrama del proyecto	Capítulos del proyecto	Calendarización
Fases del proyecto	Contenido de los capítulos	Calendarización de cada capítulo
20/11/2014 – 24/11/2014	23/11/2014 – 27/11/2014	28/11/2014 - 29/11/2014
20/11/2014 – 29/11/2014		

Tabla 46. Contenido del capítulo 4.

Capítulo 5 - Resultado
<b>Prácticas</b>
Documentos con el resultado de la implementación (Conjunto de prácticas)
30/11/2014 – 30/12/2014

Tabla 47. Contenido del capítulo 5.

Capítulo 6 - Conclusiones	
Conclusiones	Líneas de futuro
Estudio del resultado	Futuras mejoras y ampliaciones
31/12/2014 – 2/01/2015	03/01/2015 – 04/01/2015
31/12/2014 – 04/01/2015	

Tabla 48. Contenido del capítulo 6.

Memoria
<b>Documento</b>
Abstract

Resumen
Agradecimientos
Índices (Documento, figuras y tablas)
Glosario
Memoria
05/12/2015 – 11/01/2015

Tabla 49. Contenido adicional para la memoria.

Presentación
Documento
Presentación
12/01/2015 – 18/01/2014

Tabla 50. Contenido de la presentación.

### 4.3. Calendarización.

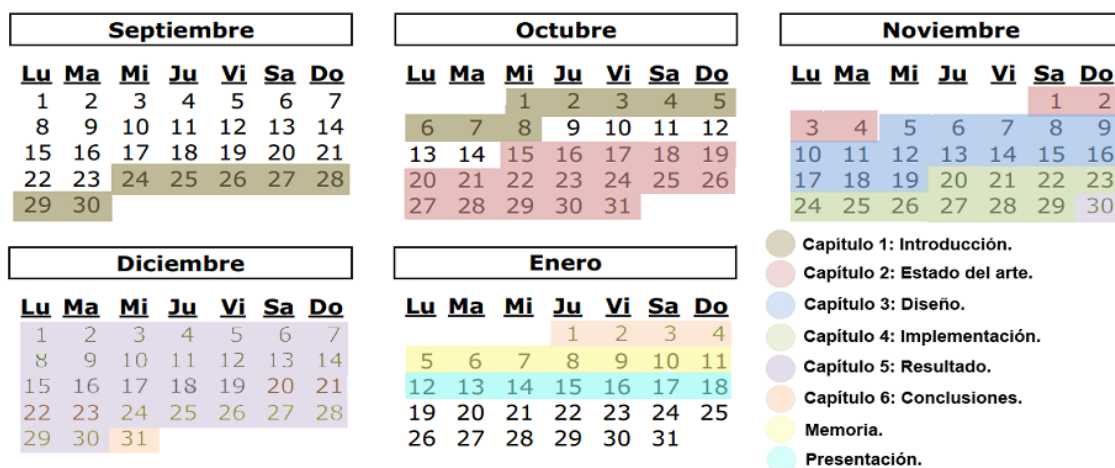


Figura 15, Calendarización de Capítulos. Fuente: Elaboración propia.

## Capítulo 5. Resultado.

El Capítulo 5 constituye el resultado final del proyecto, obtenido tras la implementación de las prácticas pertenecientes a cada batería de actividades tal y como se especifica en el punto 3.2.2 (Descripción de baterías y prácticas) del Capítulo 3. A continuación, en el apartado “Prácticas”, se exponen el título, el enunciado y la solución propuesta de todas las actividades, conformando así una colección ordenada según su numeración del título.

### 5.1. Prácticas.

#### Práctica 1.1.1.1. Introducción a la *protoboard* I.

##### Enunciado.

Implementa el circuito de la siguiente figura sobre la *protoboard* y responde a las preguntas.

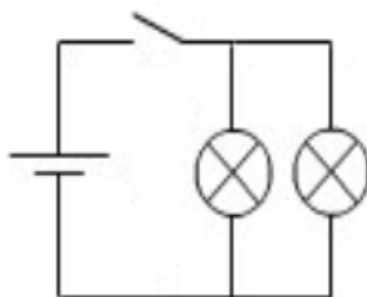


Figura 16, Circuito en paralelo. Fuente: Elaboración propia.

¿De qué tipo de circuito se trata?. ¿Qué ocurre con el resto si quitamos una bombilla cualquiera?.

##### Solución propuesta.

Circuito en paralelo. Que al activar el interruptor, la bombilla que queda seguirá iluminándose ya que los electrones continúan teniendo un circuito cerrado.

#### Práctica 1.1.1.2. Introducción a la *protoboard* II.

##### Enunciado.

Implementa el circuito de la siguiente figura sobre la *protoboard* y responde a las preguntas.

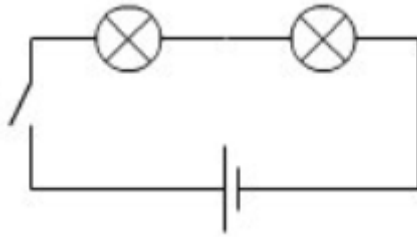


Figura 17, Circuito en serie. Fuente: Elaboración propia.

Preguntas:

¿De qué tipo de circuito se trata?. ¿Qué ocurre con la segunda bombilla si quitamos la primera?. ¿ Por qué brillan con menos intensidad que conectadas en paralelo?.

#### Solución propuesta.

Circuito en serie. No se encenderá al activar el interruptor ya que el circuito queda abierto. Porque la tensión aplicada a cada una se divide al encontrarse en serie.

#### Práctica 1.3.1.1. LED intermitente.

##### Enunciado.

En esta práctica aprenderemos a controlar el encendido y apagado cíclico de un diodo LED con Arduino a través de la programación sobre su entorno de desarrollo (IDE Arduino). Para ello, utilizaremos el pin digital de entrada/salida número 7, una resistencia adecuada para trabajar con el LED desde los 5V ofrecidos por Arduino ( $330\Omega$ ), un cable negro y uno rojo.

Como práctica introductoria, se proporciona el código necesario para el funcionamiento del circuito. Como podéis apreciar, todas las instrucciones se encuentran comentadas (el IDE entiende que todo lo escrito entre los caracteres “/” y “\*” es un comentario y no ha de procesarse), para comprender el modo de programar instrucciones es necesario que leáis atentamente dichas anotaciones y aprendáis el significado de cada línea de código.

En primer lugar debéis montar el siguiente circuito, seleccionando una resistencia adecuada para el montaje según las características del LED y la Ley de Ohm.

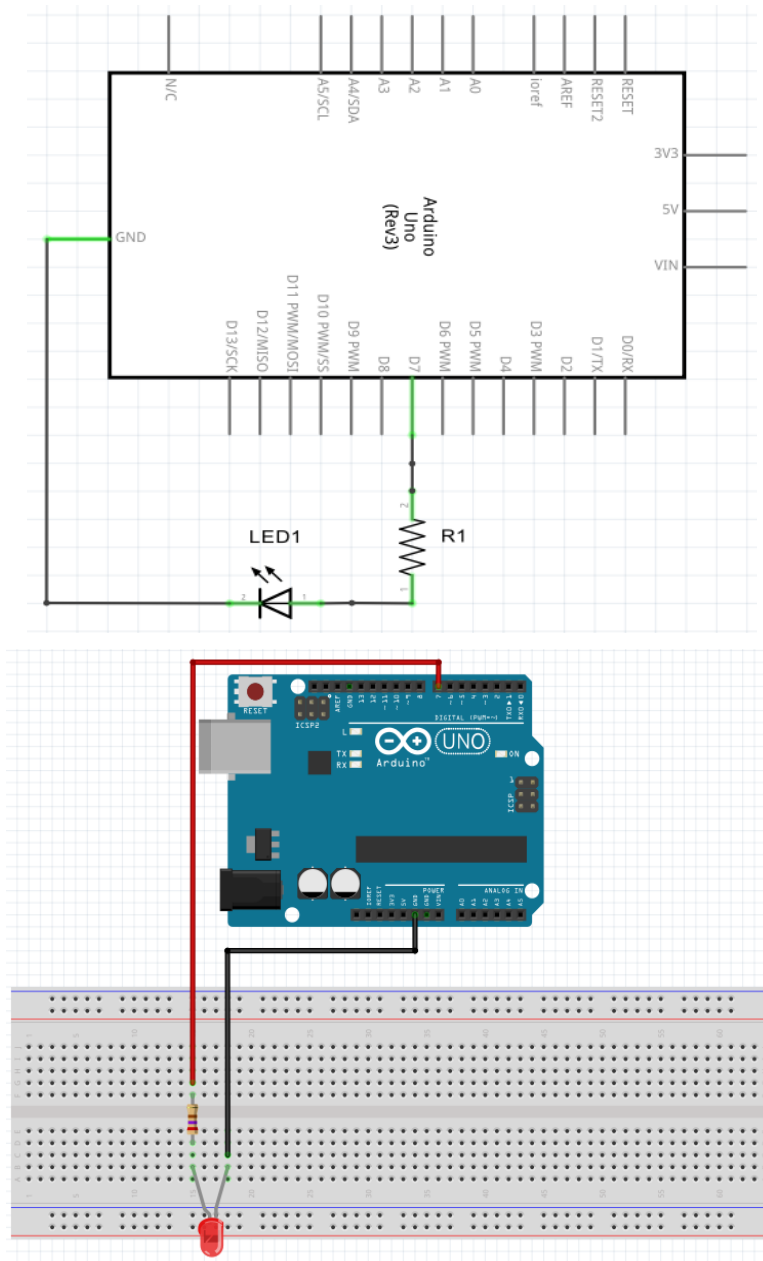


Figura 18, Esquema y prototipo de la Práctica 1.3.1.1. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, debéis crear un nuevo programa en el IDE Arduino (Archivo, nuevo) al que llamaréis “Led Intermitente” y copiad el siguiente código:

```

/* ----- */
/* LED intermitente. */
int led = 7;

```

/\* Int indica que el tipo de variable que vamos a crear es un número entero, led es el nombre de nuestra variable y 7 el número del pin de la placa Arduino a utilizar. Las líneas de código han de cerrarse con el carácter punto y coma “;” para indicar el final de la línea. Por lo tanto, estamos creando una variable que contiene el número de pin al que se conecta el LED. \*/

```
void setup() {
```

```
    pinMode(led, OUTPUT);
```

```
}
```

*/\* Una función es un segmento de código que engloba unas operaciones a realizar y devuelve un valor. Su estructura es: tipo de dato a devolver + nombre de la función y una o múltiples líneas de código encerradas entre llaves. La función no va precedida por";" pero las líneas de código en su interior sí. Setup es la función principal de cualquiera de los programas que realizaremos para Arduino y no devuelve ningún valor, de ahí void (vacío).\*/*

*/\* pinMode(led, OUTPUT); indica que el número de pin referenciado por la variable led es de salida (OUTPUT). Así, la instrucción pinMode sirve para establecer si los pins implicados en un circuito son de salida o entrada. Su estructura es: pinMode(número de pin o variable que lo contiene, OUTPUT (salida) o INPUT (entrada)). \*/*

```
void loop() {
```

```
    digitalWrite(led, HIGH);
```

```
    delay(1000);
```

```
    digitalWrite(led, LOW);
```

```
    delay(1000);
```

```
}
```

*/\* La función loop ejecuta repetidamente, una tras otra, las líneas de código contenidas. \*/*

*/\* digitalWrite sirve para especificar el valor de un pin, su primer parámetro es el número de pin (en nuestro caso la variable led) y el segundo establece la tensión de salida (LOW=0V y HIGH=5V).\*/*

*/\* delay realiza una pausa sobre la ejecución del script cuya duración viene determinada en milisegundos (en nuestro caso 1000ms = 1s). \*/*

```
/* ----- */
```

En tercer lugar, debéis cargar el programa en la placa Arduino (Archivo, cargar).

Preguntas.

¿Por qué ha de incluirse una resistencia entre la fuente de energía y el diodo LED?.

Demuestra por qué se ha seleccionado una resistencia de 330Ω.

Solución propuesta.

Valor de  $R_1$ :  $I_{LED} = \frac{V_{in} - V_{LED}}{R} \Rightarrow 0,01A = \frac{5V - 2V}{R} \Rightarrow R = \frac{3V}{0,01A} = 300\Omega$ , por lo que seleccionaremos la resistencia de valor estándar 330 $\Omega$ .

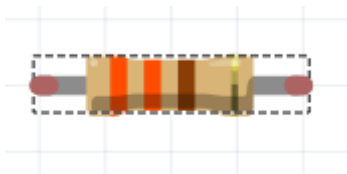


Figura 19, Resistencia de 330 $\Omega$ . Fuente: Elaboración propia a partir de Fritzing.

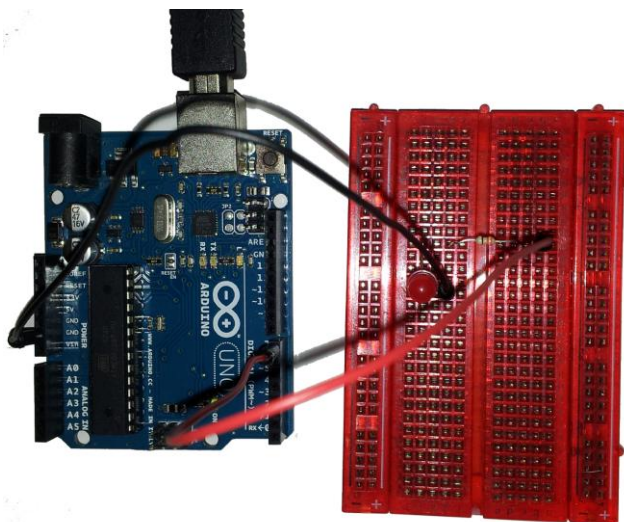


Figura 20, Solución propuesta para Práctica 1.3.1.1. Fuente: Elaboración propia.

### Práctica 1.3.1.2. Regulación lumínica.

#### Enunciado.

Esta práctica pretende mejorar la comprensión del funcionamiento del potenciómetro y el LED. Para ello, aprenderemos a utilizar los pins de entrada analógica de la placa Arduino, con lo que podremos obtener los valores de tensión que permite el potenciómetro según variamos su resistencia y la intensidad lumínica del LED según la corriente aplicada.

En primer lugar debéis montar el siguiente circuito:

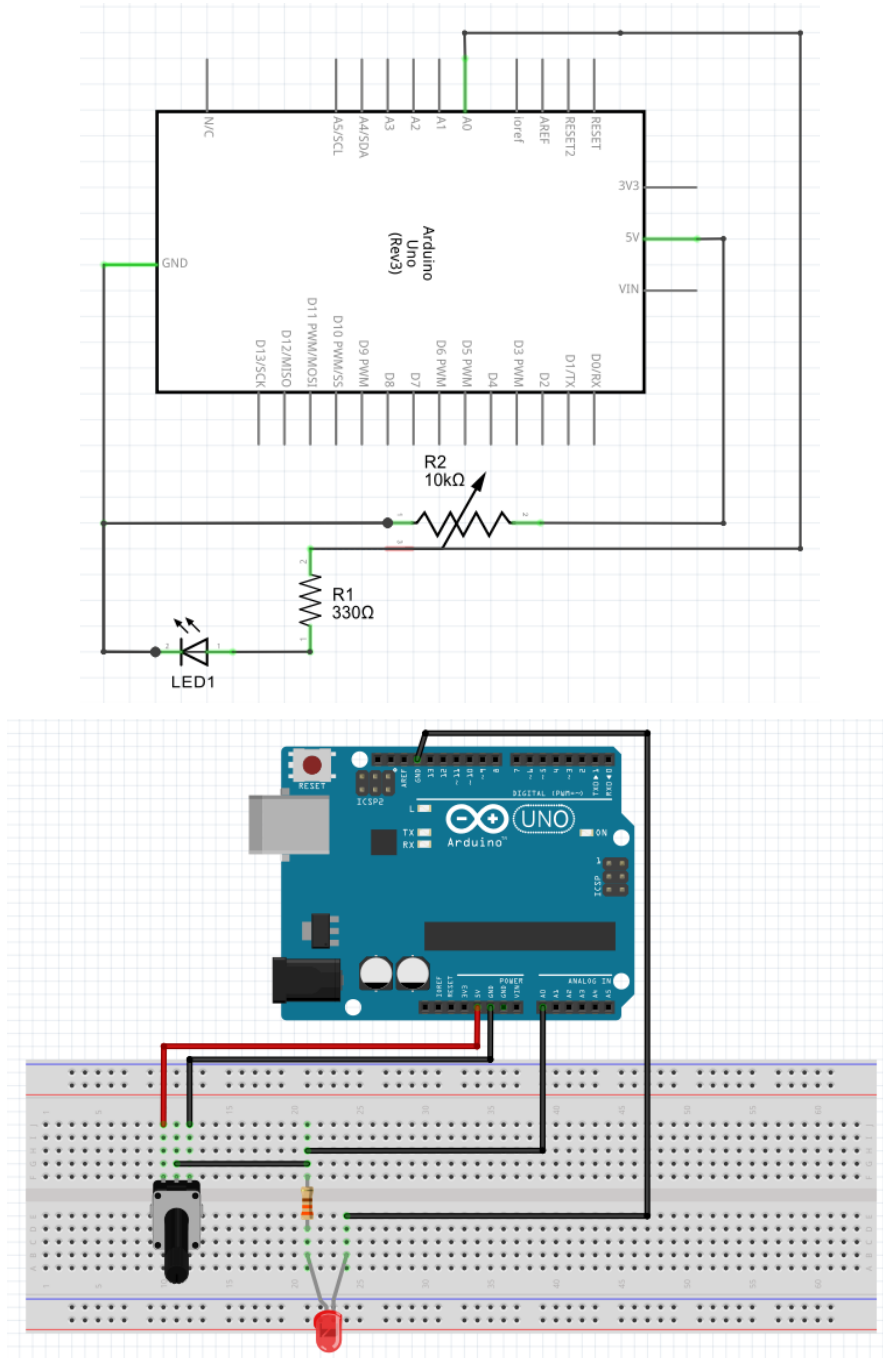


Figura 21, Esquema y prototipo de la Práctica 1.3.1.2. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, debéis crear un nuevo programa en el IDE Arduino al que llamaréis “Actividad 2 Regulación lumínica” y copiad el siguiente código:

```

/* ----- */
/* Regulación lumínica. */

int punoMedidaA = 0;
/* Creación de la variable con el pin analógico asociado al punto de medida. */

```



```
void setup() {
  Serial.begin(9600);
}
```

/\* La función Serial se utiliza para la comunicación entre la tarjeta Arduino y la computadora. En este caso, con begin se establece la comunicación con una frecuencia de comunicación en bits por segundos. Así, mediante estas líneas de código se inicializa la comunicación serial con la placa a 9600 bits por segundo, que será el valor que adoptaremos en nuestras prácticas que requieran el uso de esta función. \*/

```
void loop() {
```

```
  int valor_punoMedida = analogRead(punoMedidaA);
```

/\* analogRead es una función que se utiliza para leer el valor del pin de entrada analógico. Esta parte del programa lee el valor de voltaje del puntos A y lo almacena en la variable detipo entero valorPuntoMedida. \*/

```
  float voltaje = valor_punoMedida * (5.0 / 1023.0);
```

/\* float indica que la variable es de tipo decimal. La lectura del voltaje de entrada en los pins de entrada analógicos comprende los valores desde el 0 hasta el 1023. Para convertir estos valores al rango que nos interesa (desde 0V hasta 5V) es necesario realizar una pequeña operación matemática: Multiplicar el valor de voltaje obtenido por 5 y dividirlo por 1023. \*/

```
  float intensidad = ((voltaje-voltaje*0.4)/330)*1000;
```

/\* calculamos el valor de la intensidad que pasa por el LED aplicando la Ley de Ohm ( $I=V/R$ ) donde V es el voltaje que llega al conjunto R1 - LED (voltaje que llega menos la proporción que consume el LED) y R es la resistencia de R1 (la resistencia del LED es tan pequeña que podemos despreciarla). \*/

```
  String mostrar = "Voltaje : " + String(voltaje) + " ; Intensidad: " + String(intensidad) + "mA";
```

/\* String, indica que la variable es una cadena de texto. Si su contenido es texto, éste ha de ir entre comillas (“”) y si es el valor de una variable, ésta ha de convertirse primero en texto si no lo es mediante la función String(). Además, podemos combinar ambas posibilidades dentro de una misma cadena de texto con el operador “+”. \*/

```
  Serial.println(mostrar);
```

/\* La instrucción println de Serial sirve para imprimir líneas en la consola (Monitor serial) del IDE Arduino. En este caso, se imprime la información contenida en la variable mostrar que hemos creado anteriormente. \*/

```
}
```

/\* ----- \*/

En tercer lugar, debéis cargar el programa en la placa Arduino y abrir el monitor serial (herramientas, monitor serial).

Por último, comprobar tanto los valores obtenidos por el monitor como la intensidad lumínica del LED cuando se varía la resistencia del potenciómetro.

Preguntas.

¿Cuál es la resistencia total del circuito cuando el potenciómetro adopta su valor mínimo?, ¿Y cuando adopta su valor máximo?. ¿A partir de cuantos amperios comienza a brillar el LED?

Solución propuesta.

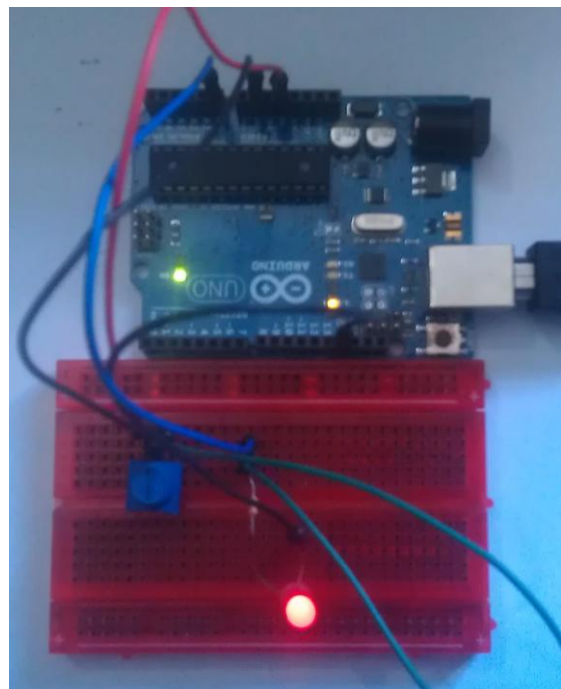
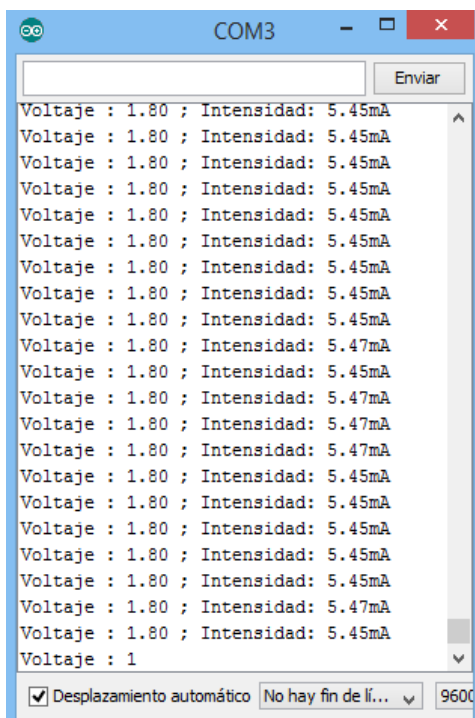


Figura 22, Solución propuesta para Práctica 1.3.1.2. Fuente: Elaboración propia.

$$R_{TOTAL1} = R_{POTENCIÓMETRO\_MIN} + R_1 + R_{LED} = 0 + 330 + 0 = 330\Omega.$$

$$R_{TOTAL1} = R_{POTENCIÓMETRO\_MAX} + R_1 + R_{LED} = 10k + 330 + 0 = 10330\Omega = 10,33k\Omega.$$

Sobre los 3 mA.

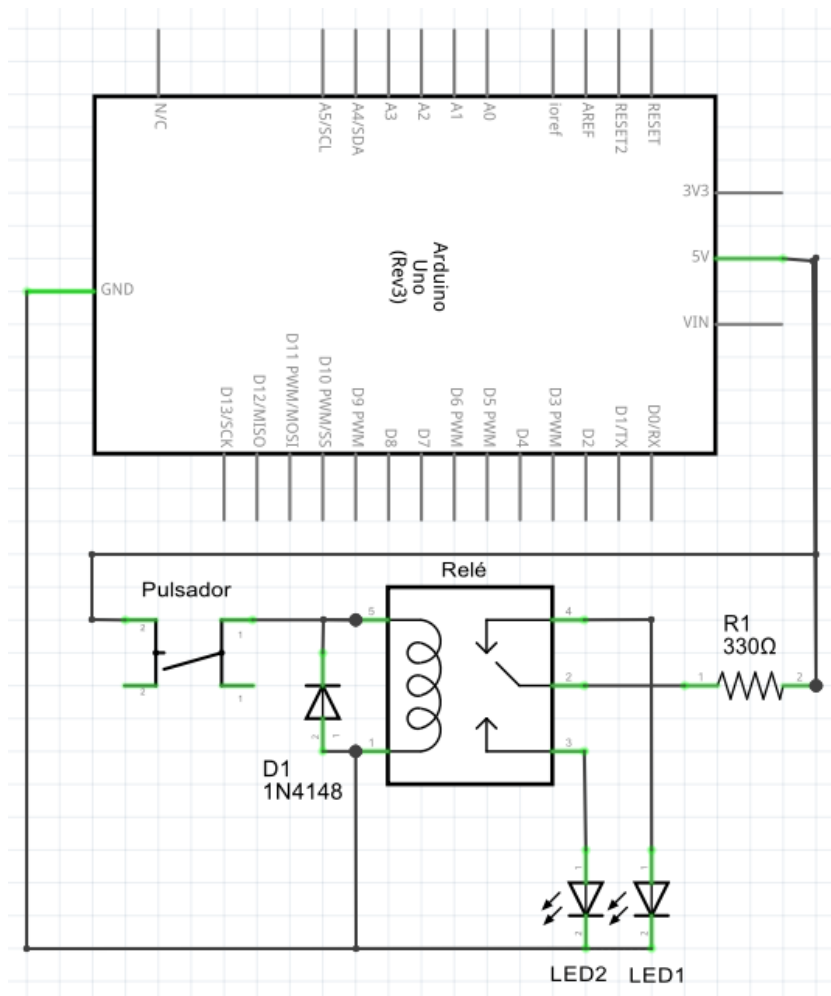
### Práctica 1.3.1.3. Relé, diodo y pulsador.

## Enunciado.

En esta práctica comprobaremos el funcionamiento del relé, el diodo y el pulsador. Para ello, montaremos un circuito que contendrá los 3 elementos y encenderá un LED rojo si no accionamos el pulsador y un LED verde si lo accionamos.

Como ya sabemos, el relé es un componente que funciona con electroimán y que se basa en el uso de una bobina. Cuando aplicamos corriente sobre la bobina se genera un campo magnético que atrae al imán que hace de selector de circuito. Al dejar de aplicar esta corriente el selector vuelve a su estado natural y, además, se genera una pequeña descarga de corriente en sentido contrario al natural del circuito que podría dañarlo. Por este motivo, utilizaremos un diodo que protegerá al relé. Por otra parte, utilizaremos un pulsador para aplicar la corriente al relé.

En primer lugar, debéis montar el siguiente circuito.



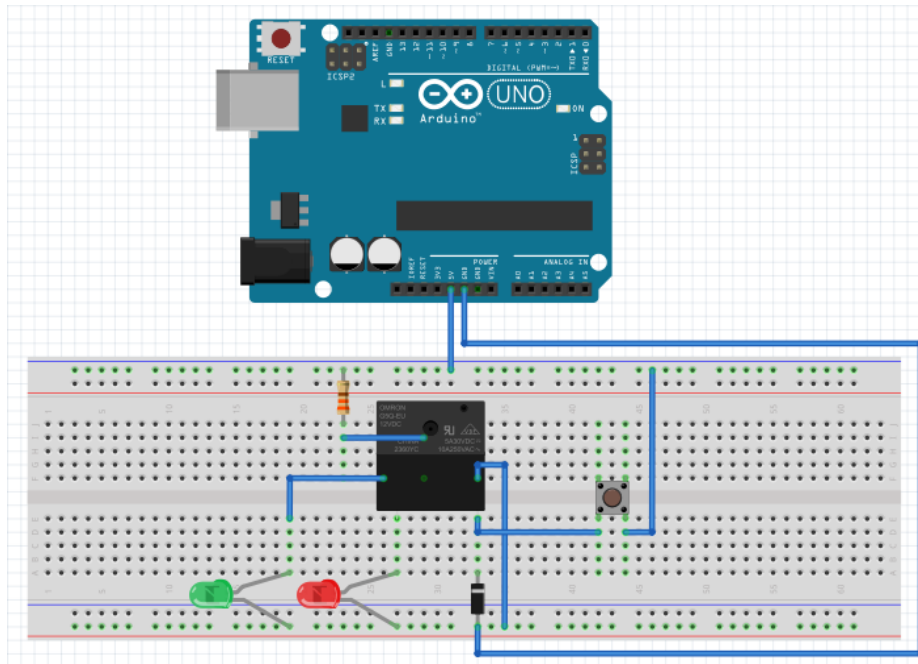


Figura 23, Esquema y prototipo de la Práctica 1.3.1.3. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, alimentad la placa Arduino conectándola al ordenador o a una fuente de entre 7 y 12V.

Por último, comprobad el funcionamiento del circuito.

Preguntas.

¿Cómo interactúa el pulsador con el relé?. ¿Por qué está el diodo 1N4148 en polarización inversa?

Solución propuesta.

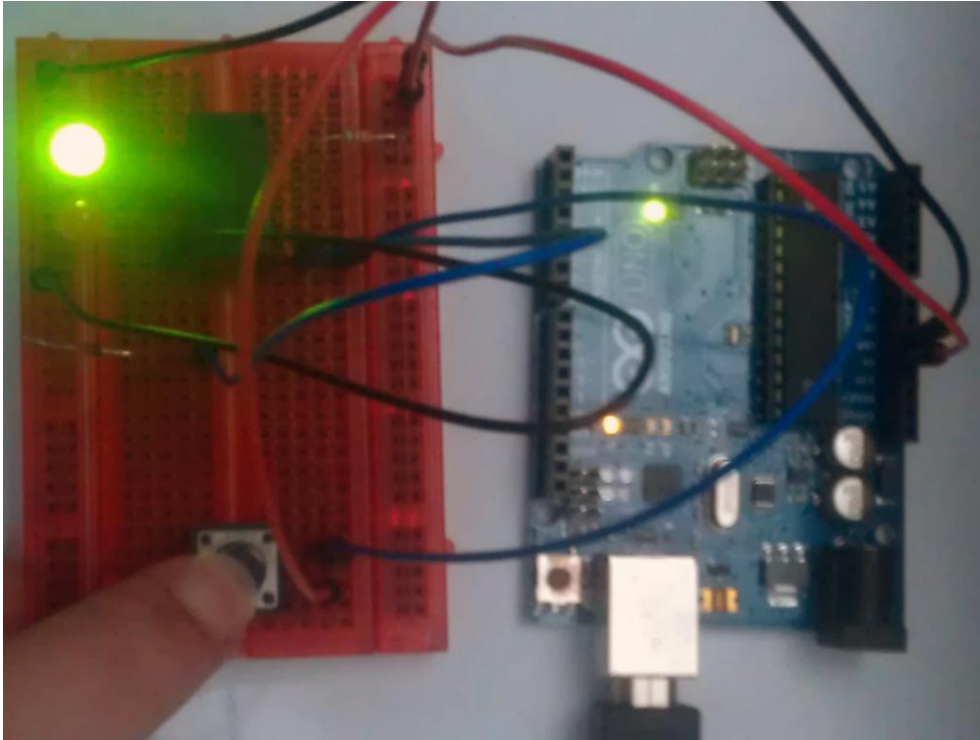


Figura 24, Solución propuesta para Práctica 1.3.1.3. Fuente: Elaboración propia.

El pulsador permite o impide el paso de corriente hacia la bobina del relé. Si en ésta existe corriente el electroimán actúa uniendo el terminal común con el terminal normalmente abierto. De lo contrario, el relé permanece en reposo y los terminales en contacto son el común y el normalmente cerrado.

El diodo ofrece tan poca resistencia que se comporta como un cortocircuito si se encuentra en polarización directa. Por lo tanto, si en el circuito el diodo estuviese en polarización directa al pulsar el pulsador la corriente pasaría completo a través suyo y no por el relé, inutilizándolo. Al encontrarse en polarización inversa, lo que se consigue es que la corriente en sentido contrario que produce la bobina al descargarse pase a través suya y no a través del relé, evitando así su deterioro.

#### Práctica 1.3.1.4. El condensador temporizador.

##### Enunciado.

En esta práctica trabajaremos con el condensador para construir un sistema que continúe proporcionando iluminación durante un tiempo tras haber cortado el suministro eléctrico. Utilizando la placa Arduino, comprobaremos como la tensión proporcionada por la carga del condensador va mermando hasta llegar a 0. Recordad que el tiempo tanto de carga como de descarga del condensador es:

$$t = 5 \cdot R \cdot C$$

En primer lugar montad el siguiente circuito:

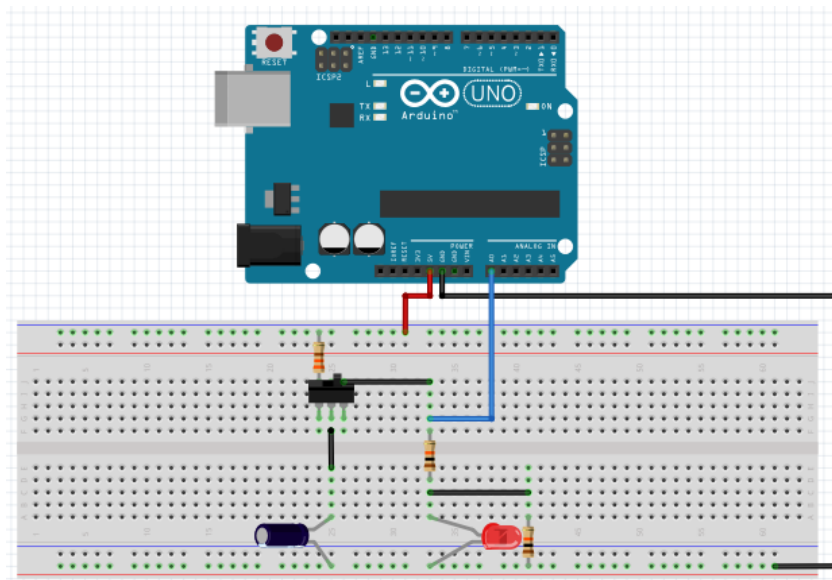
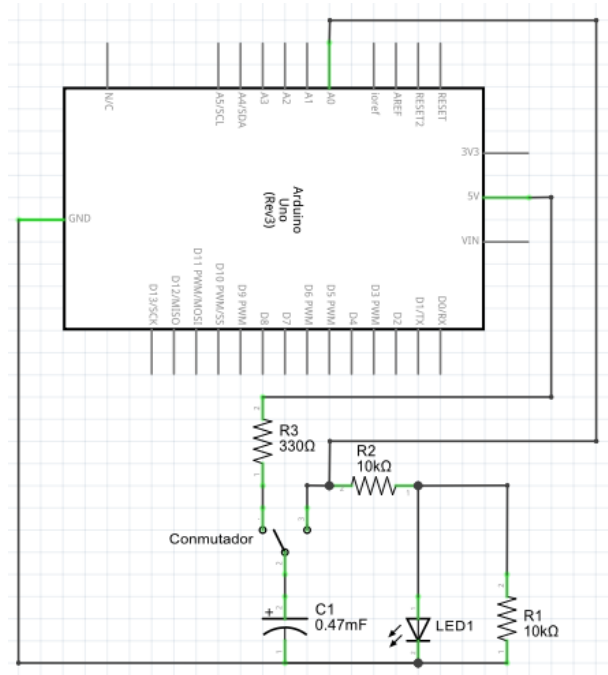


Figura 25, Esquema y prototipo de la Práctica 1.3.1.4. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

/\* ----- \*/

/\* El condensador temporizador. \*/

```

int puntoMedida = 0;

/* Crea la variable con el pin analógico asociado al punto de medida. */

void setup() {
  Serial.begin(9600);
}

void loop() {

  int valorPuntoMedida = analogRead(puntoMedida);

  /* Lee la tensión recibida en el pin puntoMedida y lo guarda en la variable
  valorPuntoMedida. */

  float voltaje = valorPuntoMedida * (5.0 / 1023.0);

  /* Convierte el valor de 0-1023 recibido a un valor de 0-5. */

  String mostrar = "Voltaje : " + String(voltaje) + "V ; Intensidad : " + String(voltaje/10) +
  "mA .";

  /* Crea una cadena de texto y la guarda en la variable mostrar. */

  Serial.println(mostrar);

  /* Muestra el contenido de la variable mostrar en el monitor serial. */
}
/* ----- */

```

Por último, conectad Arduino UNO al PC, cargad el programa, mostrad en pantalla el monitor serial y comprobad cómo se descarga el condensador sobre el circuito con LED interactuando con el conmutador.

Preguntas.

Teniendo en cuenta que a un diodo es necesario aplicarle una tensión mínima para que deje pasar la corriente (de lo contrario actúa como un circuito abierto). ¿Por qué es necesario conectar una resistencia en paralelo con el LED?

Solución propuesta.

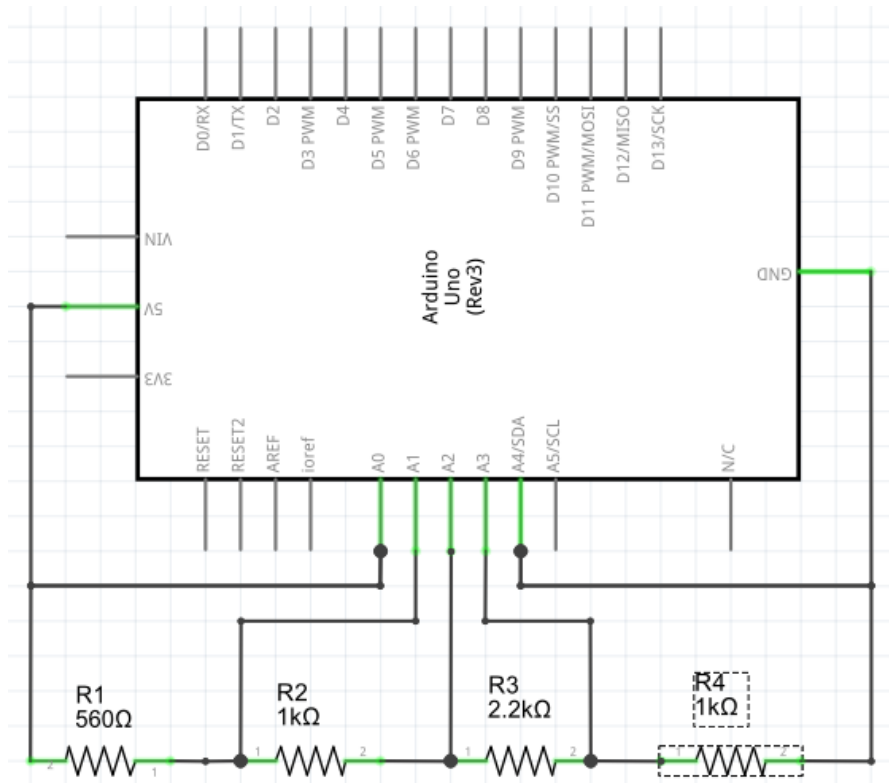
A un diodo es necesario aplicarle una tensión mínima para que deje pasar la corriente ya que de lo contrario actúa como un circuito abierto. Por lo tanto, cuando el condensador aún aplica tensión sobre el LED pero ésta no es suficiente, el LED no dejará pasar la corriente y quedará atrapada en el circuito. La resistencia en paralelo posibilita la descarga completa del condensador.

### Práctica 1.3.1.5. Circuito serie.

#### Enunciado.

En esta práctica trabajaremos la Ley de Ohm sobre un circuito de resistencias en serie.

En primer lugar montad el siguiente circuito:





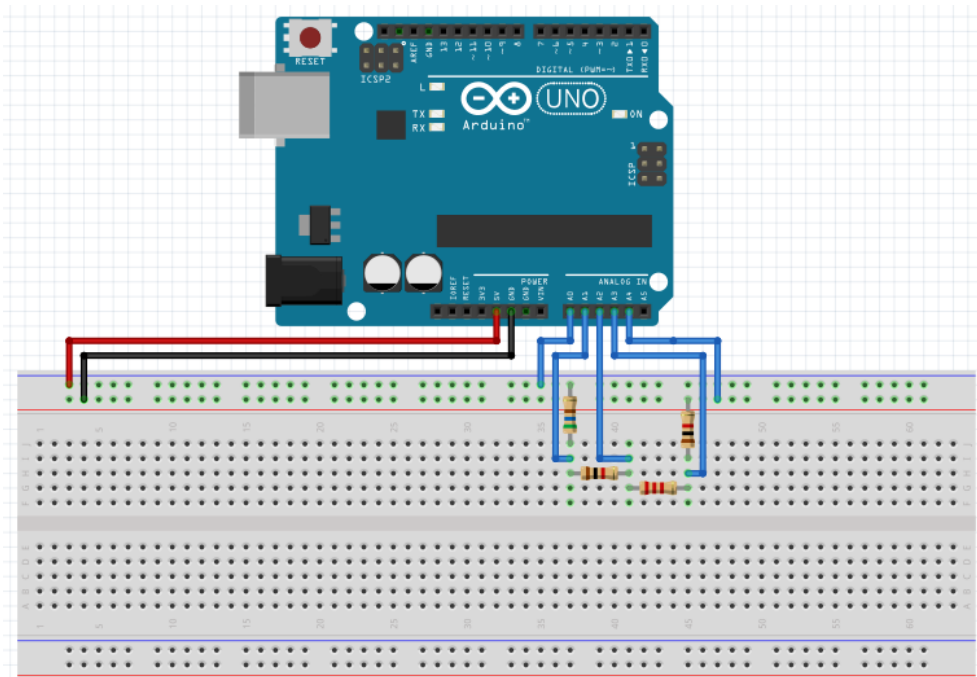


Figura 26, Esquema y prototipo de la Práctica 1.3.1.5. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

```

/* ----- */
/* Circuito serie. */

int puntoA = 0;
int puntoB = 1;
int puntoC = 2;
int puntoD = 3;
int puntoE = 4;

int r1 = 560;
int r2 = 1000;
int r3 = 2200;
int r4 = 1000;

/* Crea la variable con el pin analógico asociado al punto de medida y el valor de las resistencias. */

void setup() {
  Serial.begin(9600);

  int valorA = analogRead(puntoA);
  int valorB = analogRead(puntoB);
  int valorC = analogRead(puntoC);
  int valorD = analogRead(puntoD);
  int valorE = analogRead(puntoE);

```

```
/* Lee la tensión recibida en los pins analógicos correspondientes a los puntos de
medida (A-E). */
```

```
float vA = valorA * (5.0 / 1023.0);
float vB = valorB * (5.0 / 1023.0);
float vC = valorC * (5.0 / 1023.0);
float vD = valorD * (5.0 / 1023.0);
float vE = valorE * (5.0 / 1023.0);
```

```
/* Convierte el valor de 0-1023 recibido a un valor de 0-5. */
```

```
float vR1 = vA - vB;
float vR2 = vB - vC;
float vR3 = vC - vD;
float vR4 = vD - vE;
float vCircuito = vA - vE;
```

```
/* Calcula la tensión aplicada sobre cada resistencia y el voltaje total. */
```

```
float iR1 = vR1/r1*1000;
float iR2 = vR2/r2*1000;
float iR3 = vR3/r3*1000;
float iR4 = vR4/r4*1000;
float iCircuito = vCircuito/(r1+r2+r3+r4)*1000;
```

```
/* Calcula la intensidad de corriente, en miliamperios (mA), de cada resistencia y sobre
el circuito.. */
```

```
Serial.println("      Tension | Corriente");
Serial.println("Circuito : " + String(vCircuito) + " | " + String(iCircuito));
Serial.println("  R1 : " + String(vR1) + " | " + String(iR1));
Serial.println("  R2 : " + String(vR2) + " | " + String(iR2));
Serial.println("  R3 : " + String(vR3) + " | " + String(iR3));
Serial.println("  R4 : " + String(vR4) + " | " + String(iR4));
Serial.println();
```

```
/* Muestra toda la información deseada sobre diferentes líneas. */
```

```
}
```

```
void loop(){
```

```
}
```

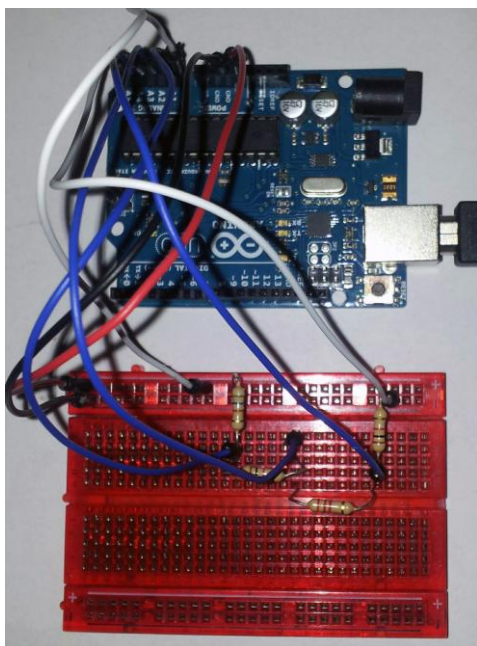
```
/* ----- */
```

En tercer lugar, carga el programa a la placa Arduino y abre el monitor serial.

Preguntas.

¿Tienen todas las resistencias el mismo voltaje?. ¿Atraviesa a todas las resistencias la misma corriente?. Si las resistencias R2 y R4 son iguales, ¿Por qué pueden diferir sus valores de tensión y voltaje?.

### Solución propuesta.



	Tension	Corriente
Circuito :	5.00	1.05
R1 :	0.59	1.05
R2 :	1.05	1.05
R3 :	2.32	1.06
R4 :	1.04	1.04

COM3

Enviar

Desplazamiento automático No hay fin de línea 9600 baud

Figura 27, Solución propuesta para Práctica 1.3.1.5. Fuente: Elaboración propia.

No, como puede comprobarse en el monitor serial, los valores de tensión para cada resistencia en un circuito serie son distintos (siempre y cuando no sean todas las resistencias iguales).

Si, en un circuito serie todas las resistencias son atravesadas por la misma corriente.

Las diferencias entre los valores obtenidos para R2 y R4 se deben, al igual que las posibles variaciones decimales entre las corrientes obtenidas por la forma de calcularlas, se deben al margen de error del valor real de una resistencia respecto a su valor nominal.

### Práctica 1.3.1.6. Circuito paralelo.

#### Enunciado.

En esta práctica trabajaremos la Ley de Ohm sobre un circuito de resistencias en paralelo.

En primer lugar montad el siguiente circuito:

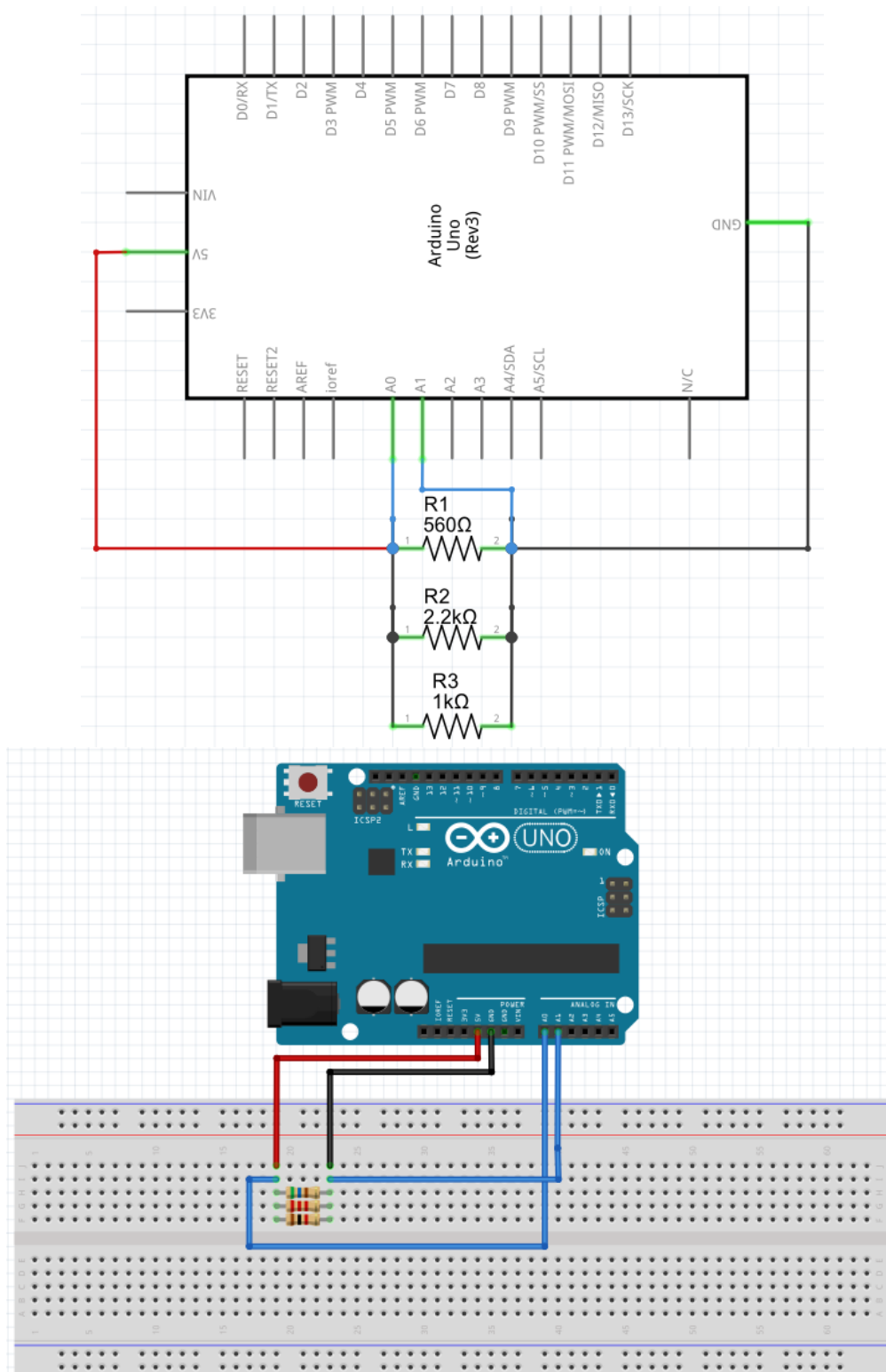


Figura 28, Esquema y prototipo de la Práctica 1.3.1.6. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

```

/* ----- */
/* Circuito paralelo. */

int puntoA = 0;
int puntoB = 1;

int r1 = 560;
int r2 = 2200;
int r3 = 1000;

/* Crea la variable con el pin analógico asociado al punto de medida y el valor de las
resistencias. */

void setup() {
  Serial.begin(9600);

  int valorA = analogRead(puntoA);
  int valorB = analogRead(puntoB);

  /* Lee la tensión recibida en los pins analógicos correspondientes a los puntos de
medida (A-B). */

  float vA = valorA * (5.0 / 1023.0);
  float vB = valorB * (5.0 / 1023.0);

  /* Convierte el valor de 0-1023 recibido a un valor de 0-5. */

  float vR1 = vA - vB;
  float vR2 = vA - vB;
  float vR3 = vA - vB;
  float vCircuito = vA - vB;

  /* Calcula la tensión aplicada sobre cada resistencia y la diferencia de potencial del
circuito. */

  float iR1 = vR1/r1*1000.0;
  float iR2 = vR2/r2*1000.0;
  float iR3 = vR3/r3*1000.0;

  float rTotal = 1.0/((1.0/r1)+(1.0/r2)+(1.0/r3));

  float iCircuito = vCircuito/rTotal*1000.0;

  /* Calcula la intensidad de corriente, en miliamperios (mA), de cada resistencia y sobre
el circuito. */

  Serial.println("      Tension | Corriente");
  Serial.println("Circuito : " + String(vCircuito) + " | " + String(iCircuito));

```

```

Serial.println(" R1 : " + String(vR1) + " | " + String(iR1));
Serial.println(" R2 : " + String(vR2) + " | " + String(iR2));
Serial.println(" R3 : " + String(vR3) + " | " + String(iR3));
Serial.println();

Serial.println("Resistencia total : " + String(rTotal));

/* Muestra toda la información deseada sobre diferentes líneas. */
}

void loop(){
}
/* ----- */

```

En tercer lugar, carga el programa a la placa Arduino y abre el monitor serial.

Preguntas.

¿Tienen todas las resistencias el mismo voltaje?. ¿Atraviesa a todas las resistencias la misma corriente?.

Solución propuesta.

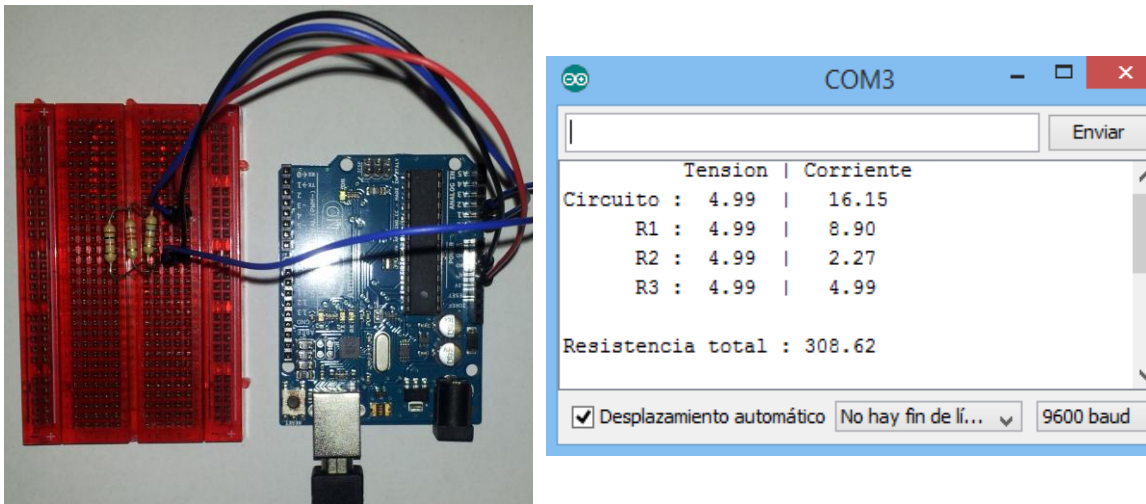


Figura 29, Solución propuesta para Práctica 1.3.1.6. Fuente: Elaboración propia.

Si, como puede observarse en el monitor serial todas las resistencias tienen la misma tensión.

No, ya que en un circuito paralelo la intensidad de cada circuito cerrado es diferente.

### Práctica 1.3.1.7. Circuito mixto.

## Enunciado.

En esta práctica comprobaremos el cumplimiento de la Ley de Ohm sobre un circuito de resistencias en serie y paralelas (mixto).

En primer lugar montad el siguiente circuito:

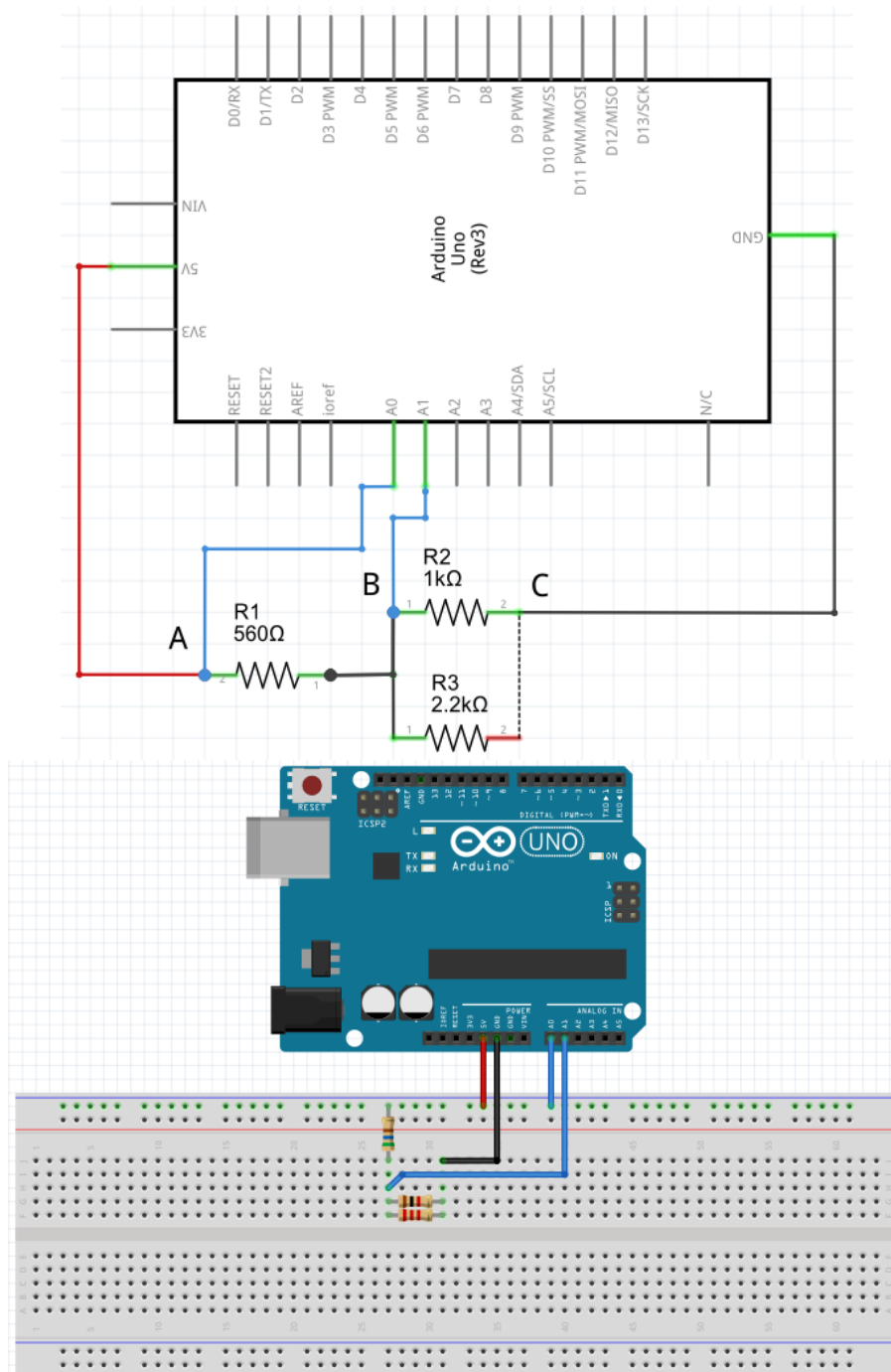


Figura 30, Esquema y prototipo de la Práctica 1.3.1.7. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

```

/* ----- */
/* Circuito mixto. */

int puntoA = 0;
int puntoB = 1;

int r1 = 560;
int r2 = 1000;
int r3 = 2200;

/* Crea la variable con el pin analógico asociado al punto de medida y el valor de las
resistencias. */

void setup() {
  Serial.begin(9600);

  int valorA = analogRead(puntoA);
  int valorB = analogRead(puntoB);

/* Lee la tensión recibida en los pins analógicos correspondientes a los puntos de
medida (A-B). */

  float vA = valorA * (5.0 / 1023.0);
  float vB = valorB * (5.0 / 1023.0);

/* Convierte el valor de 0-1023 recibido a un valor de 0-5. */

  float vR1 = vA - vB;
  float vR2 = vB;
  float vR3 = vB;
  float vCircuito = vA;

/* Calcula la tensión aplicada sobre cada resistencia y la diferencia de potencial del
circuito. */

  float iR1 = vR1/r1*1000.0;

```



```

float iR2 = vR2/r2*1000.0;
float iR3 = vR3/r3*1000.0;

float rParalelo = 1.0/((1.0/r2)+(1.0/r3));
float rTotal = r1+rParalelo;

float iCircuito = vCircuito/rTotal*1000.0;

/* Calcula la intensidad de corriente, en miliamperios (mA), de cada resistencia y sobre
el circuito. */

Serial.println("      Tension | Corriente");
Serial.println("Circuito : " + String(vCircuito) + " | " + String(iCircuito));
Serial.println("  R1 : " + String(vR1) + " | " + String(iR1));
Serial.println("  R2 : " + String(vR2) + " | " + String(iR2));
Serial.println("  R3 : " + String(vR3) + " | " + String(iR3));
Serial.println();

Serial.println("Resistencia en paralelo : " + String(rParalelo));
Serial.println("Resistencia total : " + String(rTotal));

/* Muestra toda la información deseada sobre diferentes líneas. */
}

void loop(){
}
/* ----- */

```

En tercer lugar, cargad el programa a la placa Arduino y abrid el monitor serial.

Preguntas.

¿Atraviesa las tres resistencias la misma intensidad de corriente?. ¿Se cumple que la suma de las corrientes de R2 y R3 es igual a la de R1?. ¿Qué resistencias tienen la misma caída de tensión?

Solución propuesta.

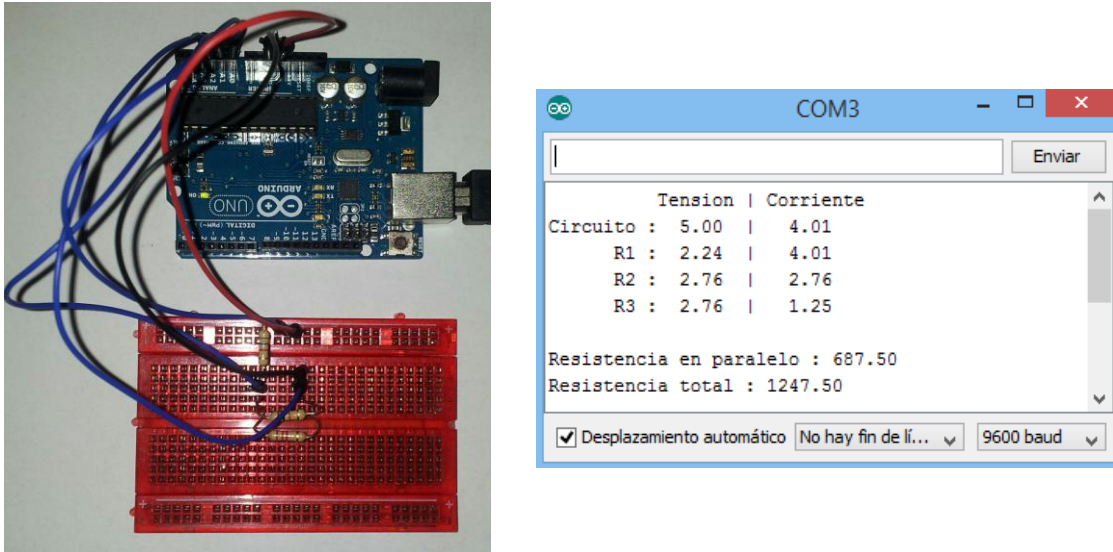


Figura 31, Solución propuesta para Práctica 1.3.1.7. Fuente: Elaboración propia.

No, puesto que dos de las resistencias se encuentran en paralelo y otra en serie con ellas.

Si. La resistencia equivalente que forman ambas se encuentra en serie con R1 y por tanto es atravesada por la misma intensidad de corriente que R1. Dicha intensidad es la se reparte en el paralelo de R2 y R3, por lo que su suma es igual a la corriente de R1.

Las resistencias R2 y R3 tienen la misma caída de tensión por encontrarse en paralelo. Ojo, si ambas tuviesen un valor igual al doble que R1, las tres tendrían la misma caída de tensión.

### Práctica 1.3.2.1. Semáforo simple.

#### Enunciado.

Se pretende implementar el prototipo de un semáforo simple para controlar el paso de los vehículos por una carretera. El modo de funcionamiento es el siguiente: Primero, el circuito comienza con un LED rojo encendido. Tras 7 segundos se apaga y se enciende un LED verde que, a su vez, se apaga tras 7 segundos dando paso al encendido de un LED amarillo que durará 2 segundos antes de apagarse y volver a repetir la secuencia.

Elaborar el prototipo y esquema del circuito utilizando la herramienta Fritzing e implementarlo usando y programando la placa Arduino UNO.

#### Solución propuesta.

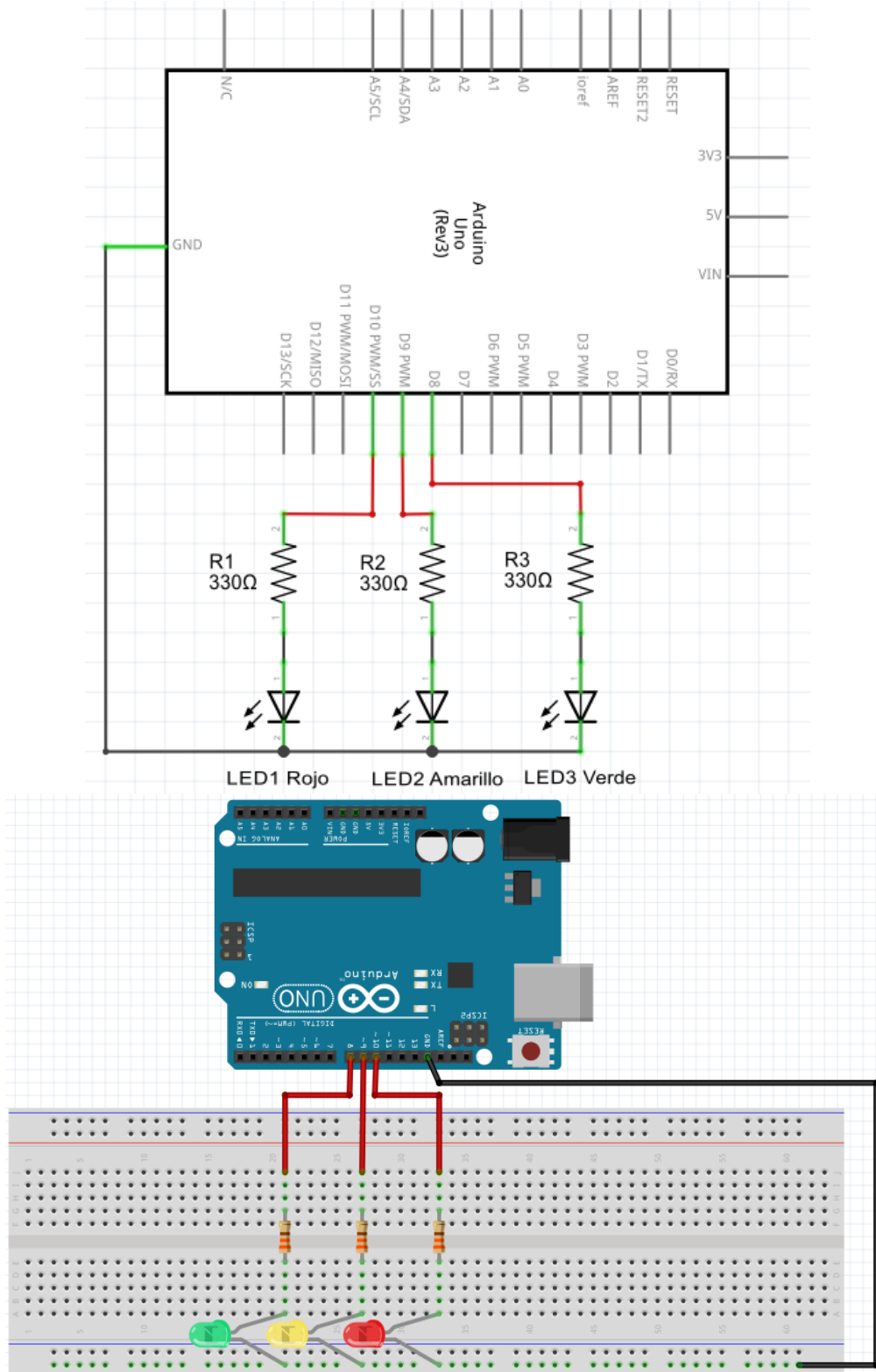


Figura 32, Esquema y prototipo solución propuesta para la Práctica 1.3.2.1. Fuente: Elaboración propia a partir de Fritzing.

/\* ----- \*/

/\* Semáforo simple. \*/

```
int ledVerde = 8;
```

```

int ledAmarillo = 9;
int ledRojo = 10;

void setup() {
  pinMode(ledVerde, OUTPUT);
  pinMode(ledAmarillo, OUTPUT);
  pinMode(ledRojo, OUTPUT);

  digitalWrite(ledAmarillo, LOW);
  digitalWrite(ledRojo, LOW);
  digitalWrite(ledVerde, LOW);
}

void loop() {

  digitalWrite(ledAmarillo, LOW);
  digitalWrite(ledRojo, HIGH);
  delay(7000);

  digitalWrite(ledRojo, LOW);
  digitalWrite(ledVerde, HIGH);
  delay(7000);

  digitalWrite(ledVerde, LOW);
  digitalWrite(ledAmarillo, HIGH);
  delay(2000);
}

/* ----- */

```

### Práctica 1.3.2.2. Motores con Arduino Motor Shield.

#### Enunciado.

En esta actividad aprenderemos a controlar un motor de corriente continua utilizando la placa Arduino Motor Shield R3 acoplada sobre la Arduino UNO R3. Esta placa shield simplifica enormemente el modo en que podemos operar sobre motores DC o servomotores.

Como podéis observar en la figura 16, la placa integra en su parte inferior izquierda una serie de conexiones en regleta. GND y Vin sirven para conectar fuentes de alimentación externas (necesarias para el funcionamiento de la placa), A+ y A- son para conectar los terminales de un motor (motor A) y B+ y B- para conectar los terminales de otro motor (motor B).

Por otra parte, los pins utilizados para controlar los motores son:

Pin digital 12: HIGH gira el motor A hacia delante; LOW gira el motor A hacia atrás.

Pin digital 9: HIGH freno motor A activado; LOW freno motor A desactivado.

Pin analógico 3: Voltaje aplicado al motor A (0-255).

Pin digital 13: HIGH gira el motor B hacia delante; LOW gira el motor B hacia atrás.

Pin digital 8: HIGH freno motor B activado; LOW freno motor B desactivado.

Pin analógico 11: Voltaje aplicado al motor B (0-255).

De este modo tenemos que un motor se funcionará en un sentido u otro a una velocidad determinada mientras no se encuentre activo su freno, caso en el que se parará. Para comprobarlo visualmente, se asociarán una serie de LED a los distintos modos de funcionamiento del motor: Verde (avance), amarillo (parado) y rojo (retroceso).

Además, para controlar desde el PC el funcionamiento del motor, necesitamos que el programa de instrucciones ejecute ordenes en función de lo que escribimos en el teclado. Veremos una nueva forma de utilizar la función Serial, que permite la lectura de lo escrito en el monitor serial.

En primer lugar montad el siguiente circuito:

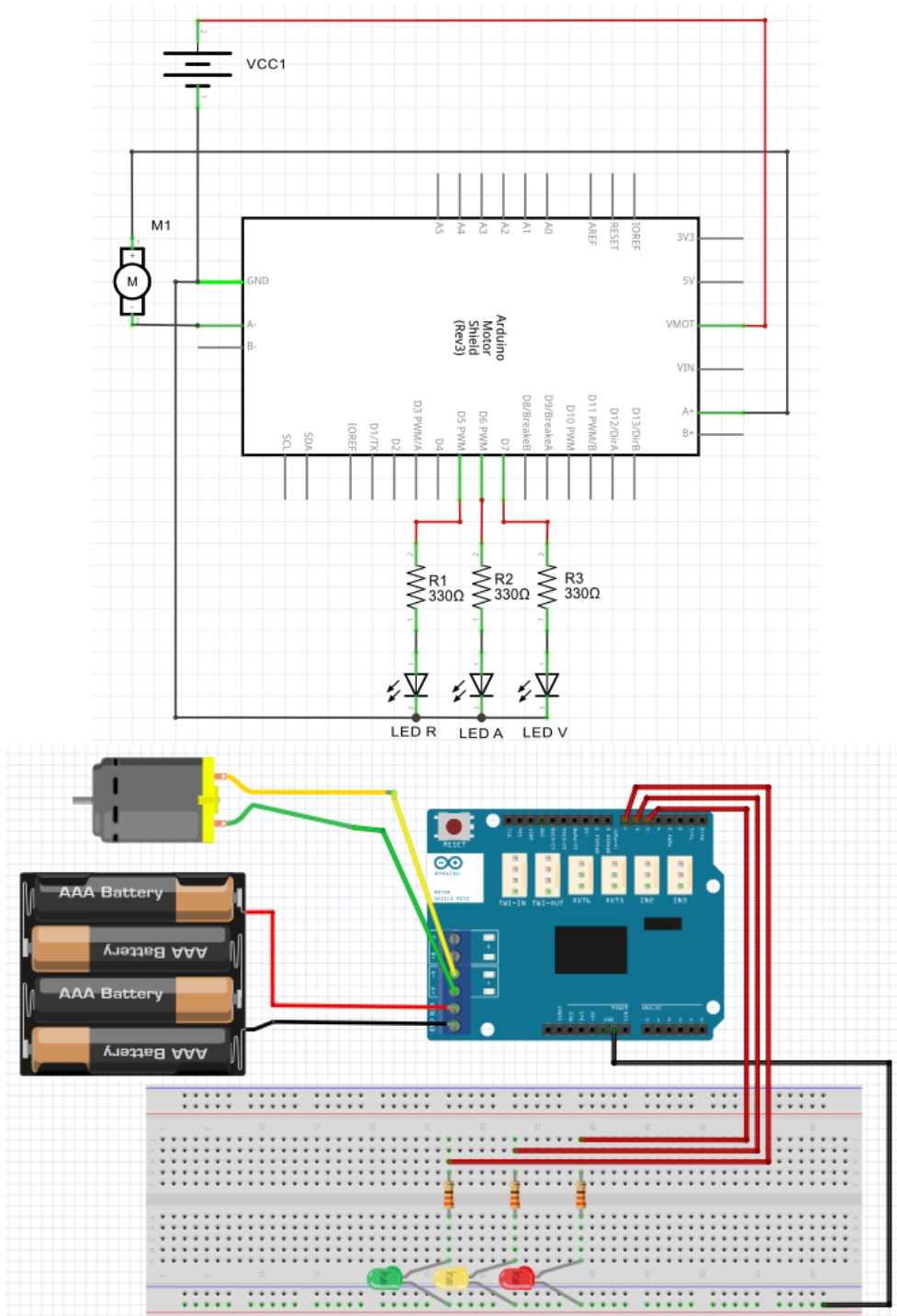


Figura 33. Esquema y prototipo de la Práctica 1.3.2.2. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

```
/* ----- */
/* Motores con Arduino Motor Shield.. */
```

```
byte dato;
int rojo = 5;
int amarillo = 6;
int verde = 7;
```

```

/* ASCII : Avance (U)= 85; Retroceso (D)= 68; Parar (C)= 67. */

void setup() {

  Serial.begin(9600);
  /* Inicialización de la comunicación. */

  pinMode (12, OUTPUT); /* Pin del motor A. */
  pinMode (9, OUTPUT); /* Pin del freno A. */
  pinMode (verde, OUTPUT);
  pinMode (amarillo, OUTPUT);
  pinMode (rojo, OUTPUT);
  /* Configuración de los pins utilizados. */
}

void loop() {

  if (Serial.available())
  dato= Serial.read();
  /* En caso de existir datos disponibles en el monitor serial, lectura y almacenamiento
  de los mismo en la variable dato. */

  switch(dato){ /* Comparación del dato recibido con los valores U (Avanzar), D
  (Retroceder) y C (Parar) expresados en código ASCII. */

  case 85: { /* (U) Avanzar. */
    /* Motor A avance. */
    digitalWrite (verde, HIGH);
    digitalWrite (amarillo, LOW);
    digitalWrite (rojo, LOW);

    digitalWrite (12, HIGH); /* Motor A marcha alante. */
    digitalWrite (9, LOW); /* Freno A desbloqueado. */
    analogWrite (3, 255); /* Velocidad A máxima. */
    break;
  }
  case 68: { /* (D) Retroceder. */
    /* Motor A retroceder. */
    digitalWrite (verde, LOW);
    digitalWrite (amarillo, LOW);
    digitalWrite (rojo, HIGH);

    digitalWrite (12, LOW); /* Motor A marcha atras. */
    digitalWrite (9, LOW); /* Freno A desbloqueado. */
    analogWrite (3, 255); /* Velocidad A máxima. */
    break;
  }
}

```

```

case 67: { /* (C) Parar. */
  /* Motor A parar. */
  digitalWrite (verde, LOW);
  digitalWrite (amarillo, HIGH);
  digitalWrite (rojo, LOW);

  digitalWrite (12, HIGH); /* Motor A marcha alante. */
  digitalWrite (9, HIGH); /* Freno A desbloqueado. */
  analogWrite (3, 255); /* Velocidad A máxima. */
  break;
}
}
Serial.println(dato);
}
/* ----- */

```

En tercer lugar, cargad el programa a la placa Arduino, abrid el monitor serial e interactuar con la placa:

U: Motor modo avance.

D: Motor modo retroceso.

C: Motor modo parar.

¡No olvidéis pulsar “intro” o “enviar” para introducir las órdenes y que estas van en mayúsculas!

#### Solución propuesta.

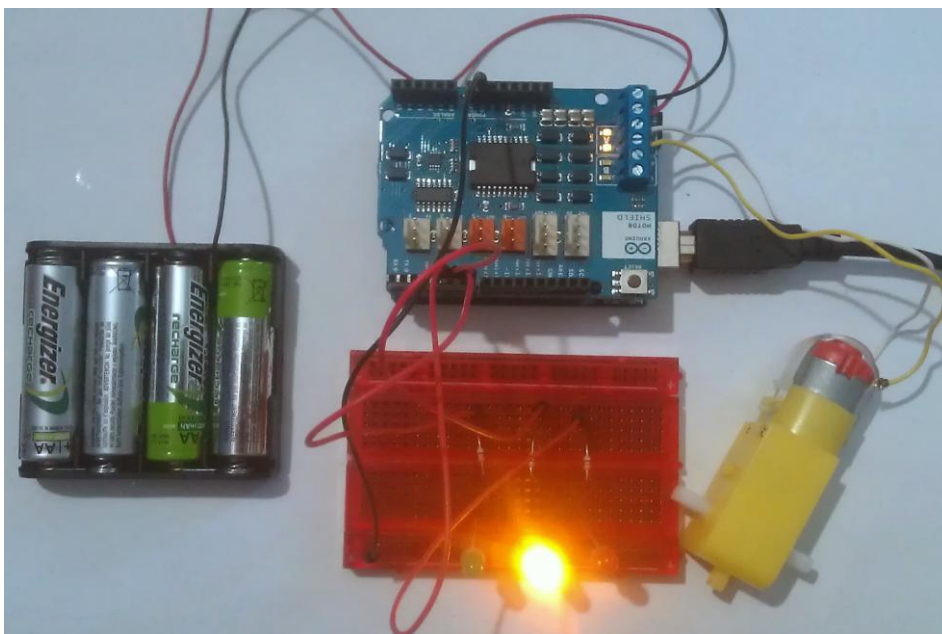


Figura 34, Solución propuesta para Práctica 1.3.2.2. Fuente: Elaboración propia.

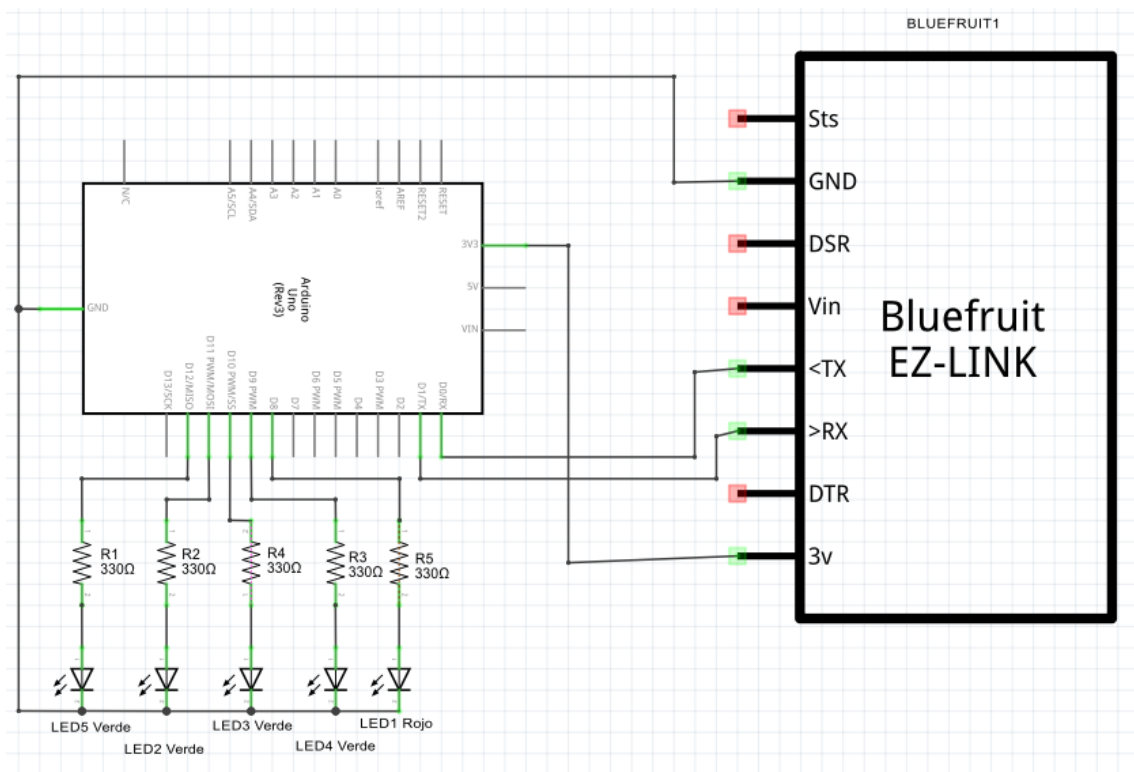


**Práctica 1.3.2.3. Control LED por bluetooth.**

**Enunciado.**

Con este ejercicio aprenderemos a interactuar con la placa Arduino desde nuestro teléfono móvil estableciendo la comunicación por Bluetooth. La tarea a realizar consiste en la implementación de un circuito con cinco LED (cuatro verdes situados en forma de cruz y uno rojo central) que se encenderán y apagarán según las órdenes recibidas desde un *smartphone* con Bluetooth.

En primer lugar realizad el montaje del siguiente circuito:



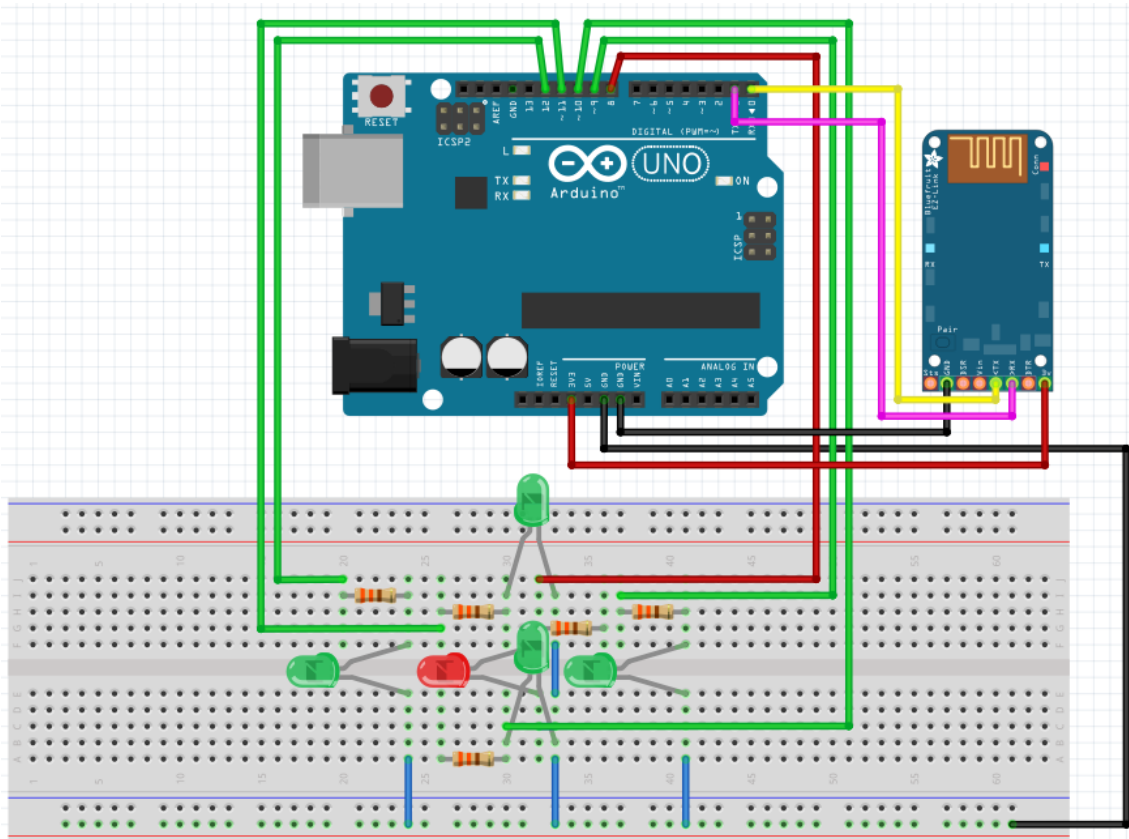


Figura 35, Esquema y prototipo de la Práctica 1.3.2.3. Fuente: Elaboración propia a partir de Fritzing.

Como veis, se introduce un nuevo componente llamado Bluefruit EZ-LINK. Se trata de un módulo esclavo de bluetooth que utilizaremos para la comunicación inalámbrica del circuito con el Smartphone. Además, es necesario descargar la aplicación gratuita Blue Control desde Google Play Store en el dispositivo Android seleccionado para la comunicación. Es importante observar que el pin receptor de datos de la tarjeta bluetooth (Rx) se conecta al transmisor de datos de Arduino UNO (Tx) y el transmisor al receptor.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

```

/* ----- */
/* Control LED por Bluetooth. */

int ledRojo=8;
int ledVerdeDerecha=9;
int ledVerdeAbajo=10;
int ledVerdeArriba=11;
int ledVerdelzquierda=12;
    
```

```

byte dato;

void setup() {
  Serial.begin(9600);
  pinMode(ledRojo,OUTPUT);
  pinMode(ledVerdeDerecha,OUTPUT);
  pinMode(ledVerdeAbajo,OUTPUT);
  pinMode(ledVerdeArriba,OUTPUT);
  pinMode(ledVerdelzquierda,OUTPUT);
}

void loop() {
  if (Serial.available())
    dato= Serial.read();
  switch(dato){
    case 85:/* "U" Arriba. */{
      digitalWrite(ledVerdeArriba, HIGH);
      digitalWrite(ledVerdeAbajo, LOW);
      digitalWrite(ledVerdeDerecha, LOW);
      digitalWrite(ledVerdelzquierda, LOW);
      digitalWrite(ledRojo, LOW);
      break;
    }
    case 82:/* "R": Derecha. */
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND.*/
    break;
  }
  case 67:/* "C": Centro.*/
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND.*/
    break;
  }
  case 76:/* "L": Izquierda.*/
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND.*/
    break;
  }
  case 68:/* "D": Abajo.*/

```

```

/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND. */
    break;
  }
}
}

/* ----- */

```

En tercer lugar, rellena el código que falta.

En cuarto lugar, carga el programa a la placa, empareja el Smartphone con el Adafruit EZ-LINK e inicia la aplicación Blue Control.

#### Solución propuesta.

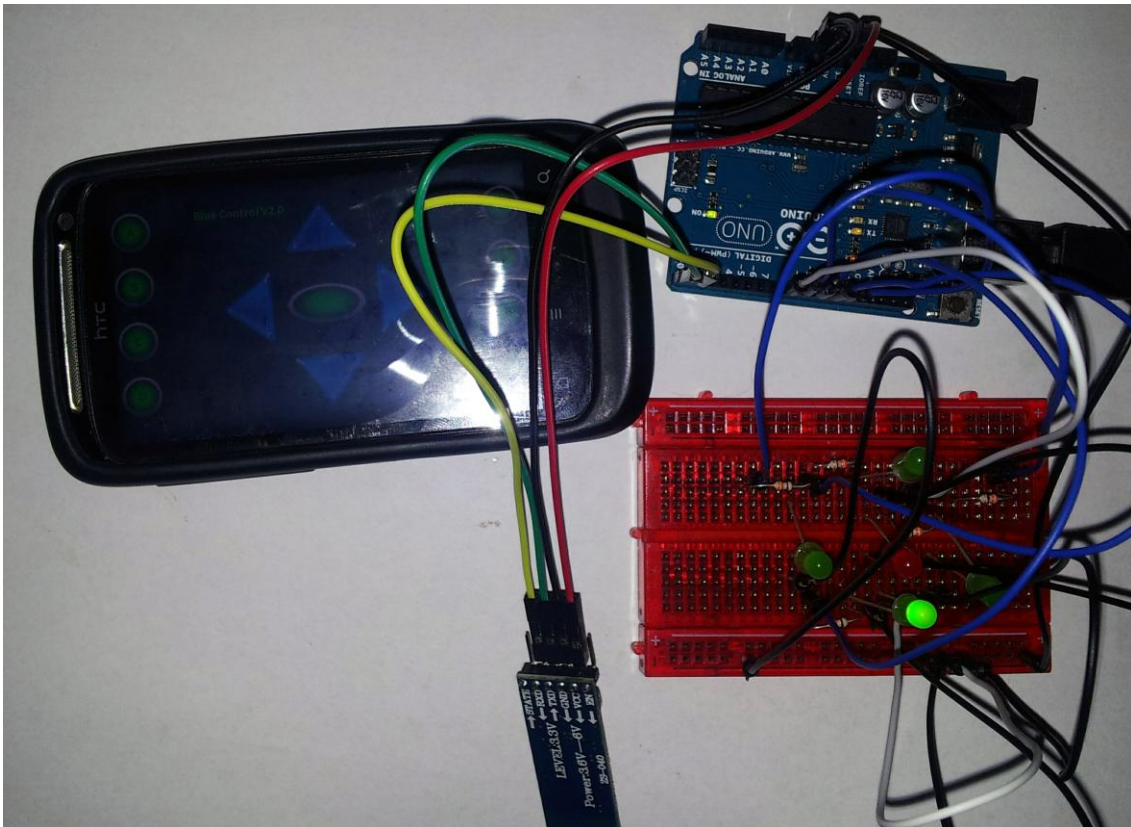


Figura 36, Solución propuesta para la Práctica 1.3.2.3. Fuente: Elaboración propia.

```

/* ----- */
/* Práctica X. Control LED por Bluetooth. */

```

```

int ledRojo=8;
int ledVerdeDerecha=9;
int ledVerdeAbajo=10;
int ledVerdeArriba=11;

```

```

int ledVerdelzquierda=12;

byte dato;

void setup() {
  Serial.begin(9600);
  pinMode(ledRojo,OUTPUT);
  pinMode(ledVerdeDerecha,OUTPUT);
  pinMode(ledVerdeAbajo,OUTPUT);
  pinMode(ledVerdeArriba,OUTPUT);
  pinMode(ledVerdelzquierda,OUTPUT);
}

void loop() {
  if (Serial.available())
    dato= Serial.read();
  switch(dato){
    case 85:/* "U" Arriba. */{
      digitalWrite(ledVerdeArriba, HIGH);
      digitalWrite(ledVerdeAbajo, LOW);
      digitalWrite(ledVerdeDerecha, LOW);
      digitalWrite(ledVerdelzquierda, LOW);
      digitalWrite(ledRojo, LOW);
      break;
    }
    case 82:/* "R": Derecha. */
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND.*/
      digitalWrite(ledVerdeArriba, LOW);
      digitalWrite(ledVerdeAbajo, LOW);
      digitalWrite(ledVerdeDerecha, HIGH);
      digitalWrite(ledVerdelzquierda, LOW);
      digitalWrite(ledRojo, LOW);
      break;
    }
    case 67:/* "C": Centro.*/
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND.*/
      digitalWrite(ledVerdeArriba, LOW);
      digitalWrite(ledVerdeAbajo, LOW);
      digitalWrite(ledVerdeDerecha, LOW);
      digitalWrite(ledVerdelzquierda, LOW);
      digitalWrite(ledRojo, HIGH);
      break;
    }
    case 76:/* "L": Izquierda.*/
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND.*/
      digitalWrite(ledVerdeArriba, LOW);
      digitalWrite(ledVerdeAbajo, LOW);
      digitalWrite(ledVerdeDerecha, LOW);
      digitalWrite(ledVerdelzquierda, HIGH);
      digitalWrite(ledRojo, LOW);
      break;
    }
    case 68:/* "D": Abajo.*/
/* RELLENA EL CÓDIGO PARA ENCENDER EL LED CORRESPOND. */
      digitalWrite(ledVerdeArriba, LOW);

```

```

digitalWrite(ledVerdeAbajo, HIGH);
digitalWrite(ledVerdeDerecha, LOW);
digitalWrite(ledVerdeIzquierda, LOW);
digitalWrite(ledRojo, LOW);
break;
}
}
}
/* ----- */

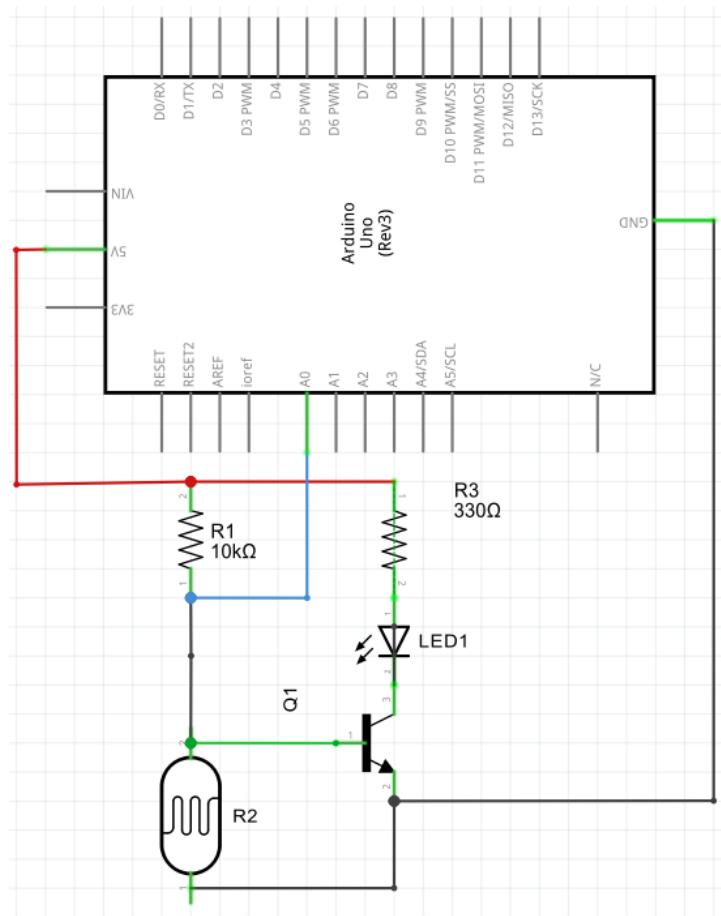
```

**Práctica 1.3.2.4. Control de luminosidad.**

**Enunciado.**

En esta práctica utilizaremos el transistor para regular la luminosidad de un LED en función de la corriente que permita llegar a la base del transistor una foto resistencia (LDR).

En primer lugar montad el siguiente circuito:



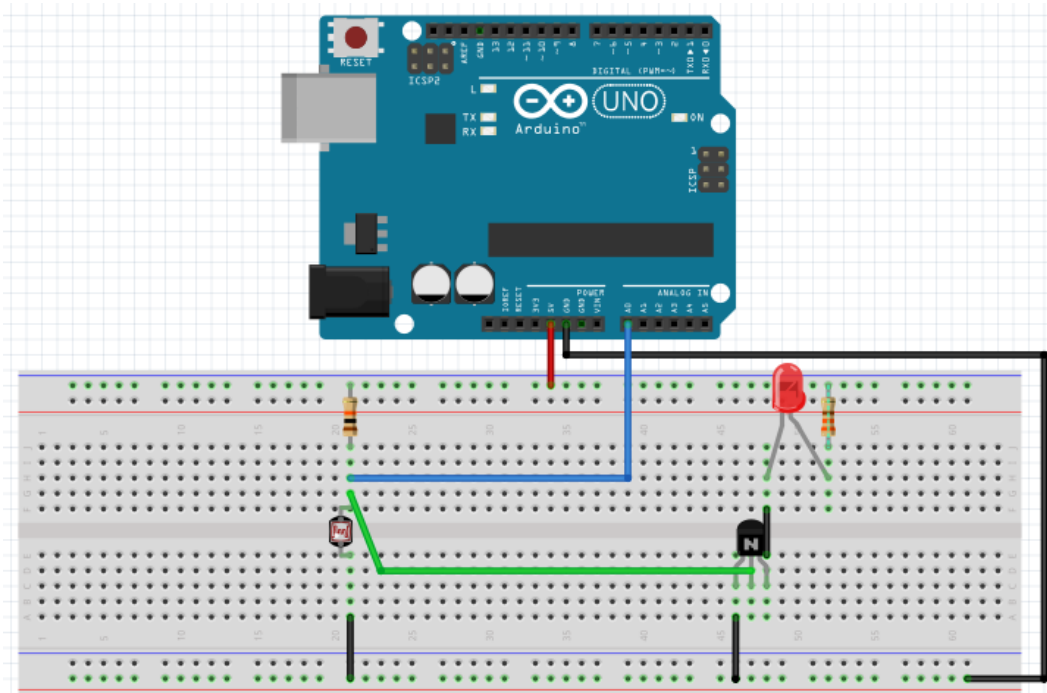


Figura 37, Esquema y prototipo de la Práctica 1.3.2.4. Fuente: Elaboración propia a partir de Fritzing.

En segundo lugar, cread un nuevo sketch y copiad el siguiente código:

```

/* ----- */
/* Control de luminosidad. */

int puntoMedida = 0;

/* Crea la variable con el pin analógico asociado al punto de medida. */

void setup() {
  Serial.begin(9600);
}

void loop() {

  int valorPuntoMedida = analogRead(puntoMedida);

  /* Lee la tensión recibida en el pin puntoMedida y lo guarda en la variable
  valorPuntoMedida. */

  float voltaje = valorPuntoMedida * (5.0 / 1023.0);

  /* Convierte el valor de 0-1023 recibido a un valor de 0-5. */

  String mostrar = "Voltaje : " + String(voltaje) + "V";

  /* Crea una cadena de texto y la guarda en la variable mostrar. */

```

```
Serial.println(mostrar);
```

```
/* Muestra el contenido de la variable mostrar en el monitor serial. */
```

```
}
```

```
/* ----- */
```

Por último, conectad Arduino UNO al PC, cargad el programa, mostrad en pantalla el monitor serial y comprobad cómo varía el voltaje aplicado a la base del transistor.

Solución propuesta.

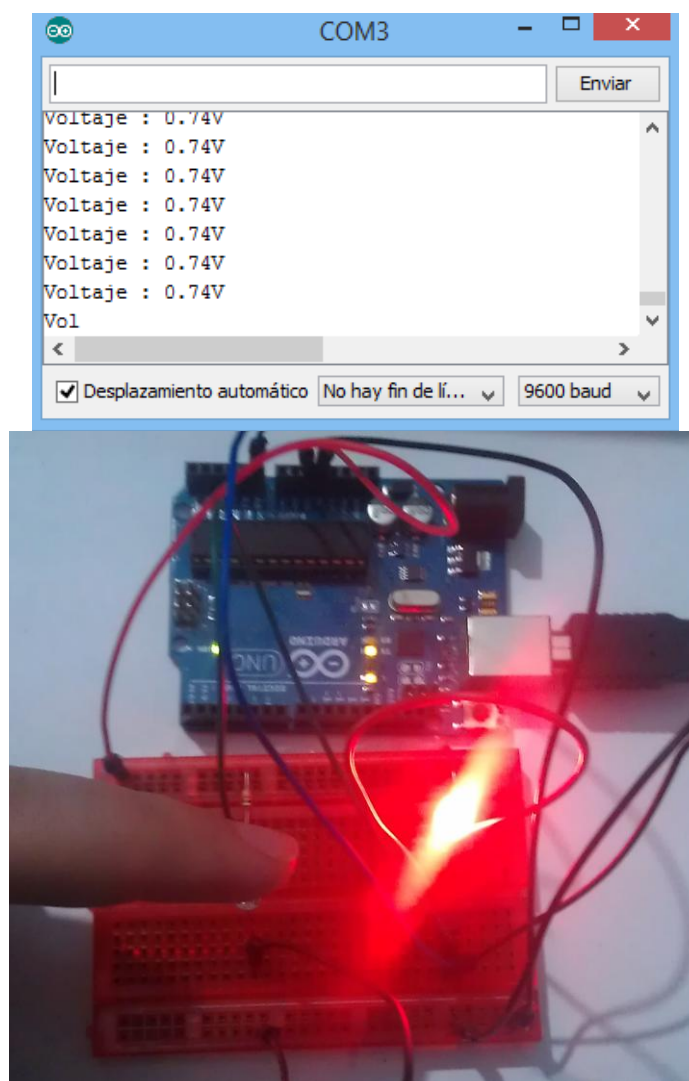


Figura 38, Solución propuesta para la Práctica 1.3.2.4. Fuente: Elaboración propia.

### Práctica 1.3.2.5. Vehículo teledirigido.

Enunciado.



Esta práctica conforma la realización de un proyecto en el que construiréis un vehículo móvil controlado desde un *Smartphone*. Para ello, es necesario el conjunto de conocimientos y habilidades adquirido durante el desarrollo de las actividades anteriores. La implementación del proyecto ha de ser dividida en tareas más sencillas, según la metodología de proyectos. Los pasos a seguir son:

1. Puesta en común de soluciones propuestas y decisión final (grupo).
2. Croquis de la solución adoptada.
3. Prototipado y esquema del circuito electrónico (Fritzing).
4. Listado de material.
5. Presupuesto.
6. Plano a escala 1:2 de la estructura a construir.
7. Construcción de la estructura.
8. Realización del circuito.
9. Ensamblaje de componentes.
10. Programación del *script*.
11. Puesta en marcha.

Para introducir la tarjeta EZ-LINK en el prototipado y esquema del circuito, es necesario descargar la librería de Adafruit desde:

<http://www.adafruit.com/blog/2013/12/13/fritzing-friday-bluefruit-ez-link/>

o

<https://github.com/adafruit/Fritzing-Library>

Solución propuesta.

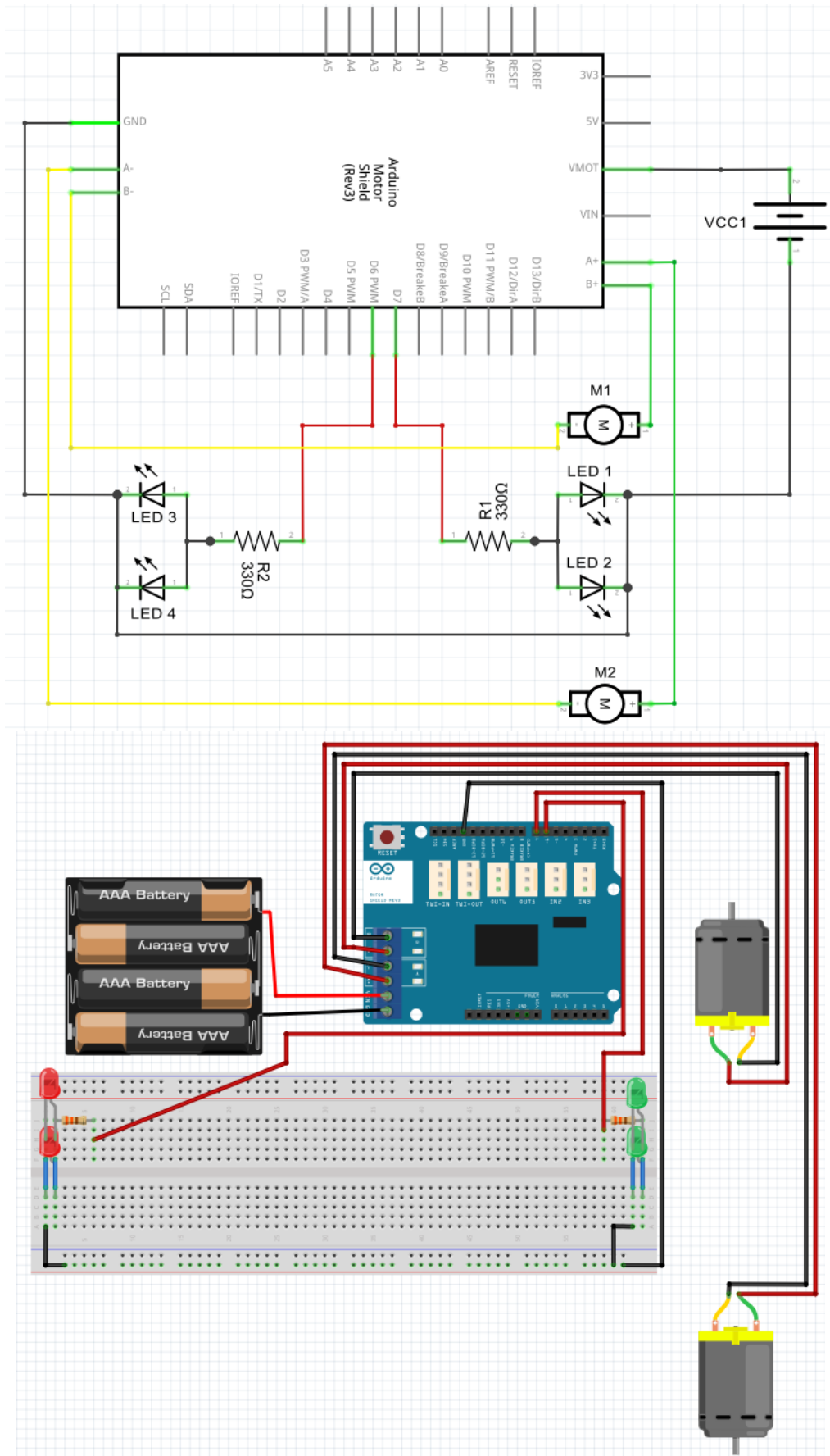


Figura 39, Solución propuesta para la Práctica 1.3.2.5. Fuente: Elaboración propia.

/\* -----\*-----\*-----\*

```

/* Vehículo teledirigido. */

byte dato;
int rojo = 6;
int verde = 7;

/* ASCII : Avance (U)= 85; Retroceso (D)= 68; Parar (C)= 67. */

void setup() {

  Serial.begin(9600);
  /* Inicialización de la comunicación. */

  pinMode (12, OUTPUT); /* Pin del motor A. */
  pinMode (9, OUTPUT); /* Pin del freno A. */

  pinMode (13, OUTPUT); /* Pin del motor B. */
  pinMode (8, OUTPUT); /* Pin del freno B. */

  pinMode (verde, OUTPUT);
  pinMode (rojo, OUTPUT);
  /* Configuración de los pins utilizados. */
}

void loop() {

  if (Serial.available())
    dato= Serial.read();
  /* En caso de existir datos disponibles en el monitor serial, lectura y almacenamiento
  de los mismo en la variable dato. */

  switch(dato){ /* Comparación del dato recibido con los valores U (Avanzar), D
  (Retroceder) y C (Parar) expresados en código ASCII. */

    case 85: { /* (U) Avanzar. */

      digitalWrite (verde, HIGH);
      digitalWrite (rojo, LOW);

      digitalWrite (12, HIGH); /* Motor A marcha alante. */
      digitalWrite (9, LOW); /* Freno A desbloqueado. */
      analogWrite (3, 255); /* Velocidad A máxima. */

      digitalWrite (13, HIGH); /* Motor B marcha alante. */
      digitalWrite (8, LOW); /* Freno B desbloqueado. */
      analogWrite (11, 255); /* Velocidad B máxima. */

      break;
    }
    case 68: { /* (D) Retroceder. */

      digitalWrite (verde, LOW);
      digitalWrite (rojo, HIGH);

      digitalWrite (12, LOW); /* Motor A marcha atras. */

```



## Práctica 1.4.1.1. Alarma LDR.

## Enunciado.

Necesitamos ser avisados cuándo una determinada estancia se queda sin iluminación y para ello disponemos de un zumbador como actuador y una LDR como sensor. A continuación debéis dar solución al problema implementando una alarma con LDR.

En primer lugar, construid el circuito en base al esquema y prototipo siguientes.

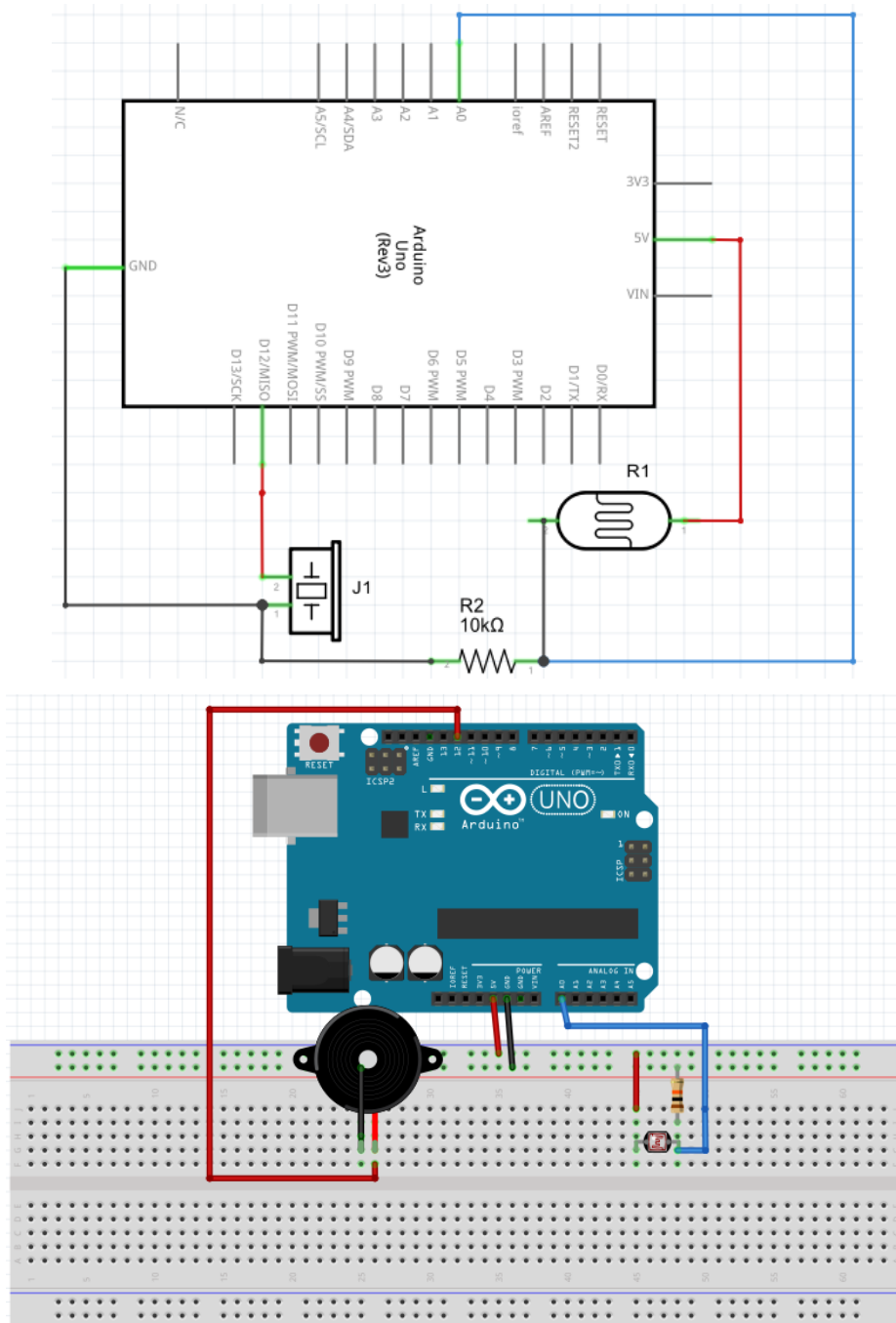


Figura 40, Esquema y prototipo de la Práctica 1.4.1.1. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código:

```

/* ----- */
/* Alarma con LDR. */

int sensor = 0;
int espera = 3000;
int zumbador = 12;
/* Variables pin del sensor, tiempo de espera y pin zumbador. */

void setup() {
  pinMode(zumbador, OUTPUT);
  /* Establece el pin del zumbador en modo salida. */

  Serial.begin(9600);
  /* Inicia la comunicación con el PC */.
}

void loop() {
  int valor = analogRead(sensor);
  /* Guarda en la variable "valor" el valor obtenido en el pin sensor. */
  if (valor<=450) {
    tone(zumbador, 262, 2000);
  }
  /* Si valor es menor o igual a 450 suena el zumbador a una frecuencia de 262Hz
durante 2 segundos. */

  Serial.println(valor);
  /* imprime en el monitor serial el valor de la variable "valor". */

  delay(espera);
  /* Espera durante el valor de la variable "espera" milisegundos. */
}
/* ----- */

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad los valores obtenidos desde el monitor serial.

Solución propuesta.

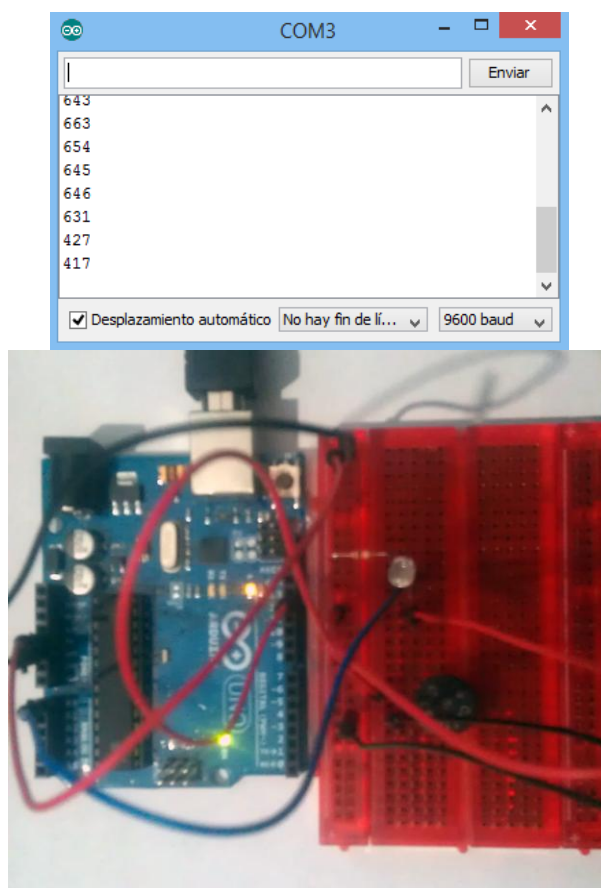


Figura 41, Solución propuesta Práctica 1.4.1.1. Fuente: Elaboración propia.

### Práctica 1.4.1.2. Termómetro LED.

Enunciado.

En esta práctica construiremos un termómetro LED. El sistema constará de un sistema de LED con tres colores diferentes (rojo, verde y azul), un sensor de temperatura y la placa Arduino UNO. La temperatura mínima corresponde al encendido del primer LED azul y la máxima del encendido de todos los LED. Así, determinado nivel de temperatura se ve reflejado en el termómetro por el encendido del LED correspondiente y todos los anteriores.

En primer lugar, construid el circuito en base al esquema y prototipo siguientes.

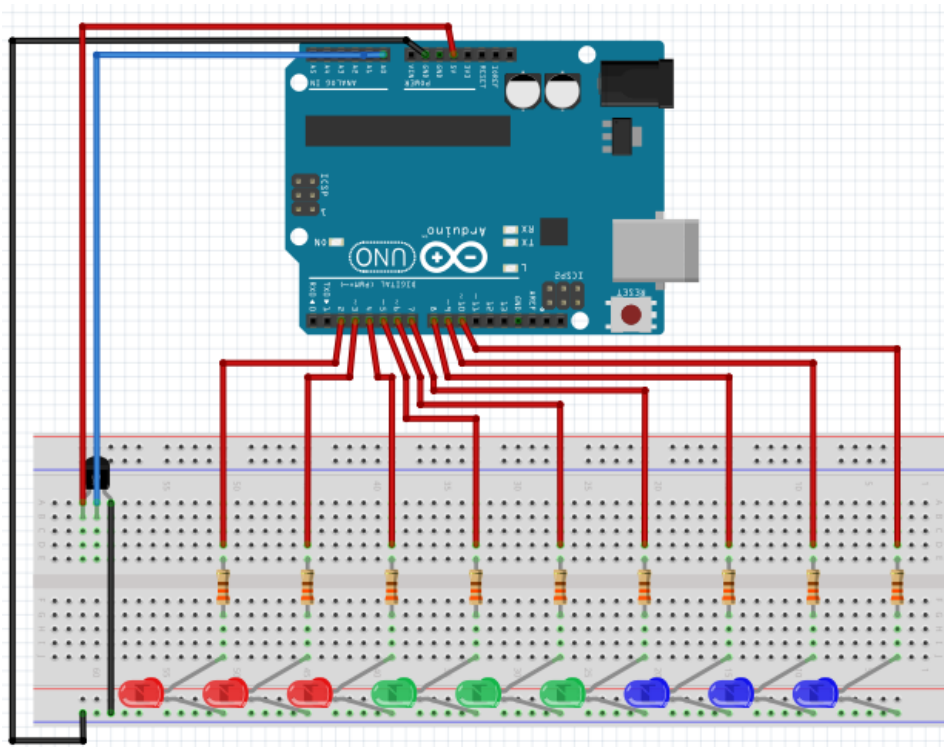
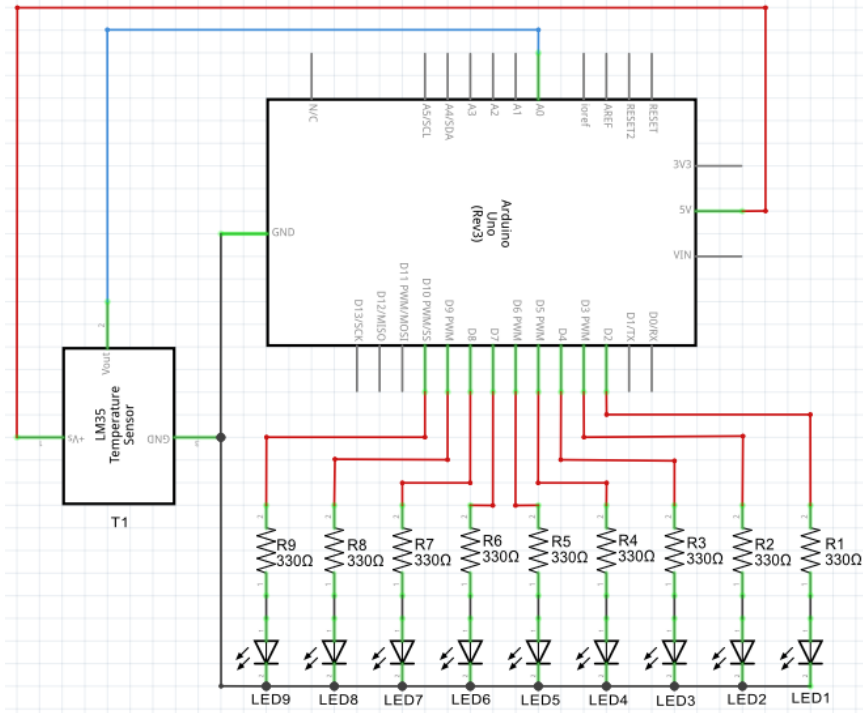


Figura 42, Esquema y prototipo de la Práctica 1.4.1.2. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código:

/\* ----- \*/

/\* Termómetro LED. \*/



```

int sensorTemperatura = 0; /* Pin analógico del sensor de temperatura. */

int led9 = 2, led8 = 3, led7 = 4, led6 = 5, led5 = 6, led4 = 7, led3 = 8, led2 = 9, led1 =
10;
/* Las variables de un mismo tipo se pueden escribir en la misma línea. */

int ledCantidad = 9;
int leds[] = {led1, led2, led3, led4, led5, led6, led7, led8, led9};
/* La variable leds es un array que contiene a las variables enteras que representan los
LED del circuito. */

void setup() {
  for (int i=0; i<ledCantidad; i++){
    pinMode(leds[i], OUTPUT);
  }
  /* La función for repite las órdenes encerradas entre llaves mientras se cumple la
condición encerrada entre paréntesis.
En este caso se lee: Empezando por i = 0 y mientras i sea menor a ledCantidad (9)
suma 1 a i cada vez que se ejecuten
las siguientes órdenes. El pin perteneciente a leds[i] es de salida.*/

  Serial.begin(9600);
  /* Inicia la comunicación con el PC. */
}

void loop() {

  float voltaje = analogRead(sensorTemperatura) * 5.0 / 1023.0;
  /* Convierte el valor obtenido por el sensor de temperatura de 0-1023 a 0-5V. */

  float gradosC = (voltaje - 0.5) * 100.0;
  /* Convierte el valor de voltaje a la temperatura correspondiente (la fórmula
corresponde a los valores especificados por
el fabricante del sensor de temperatura. */

  Serial.println("voltaje: " + String(voltaje) + "V; Temperatura: " + String(gradosC) + "C");
  /* Muestra la información en el monitor serial. */
}

```

```

int encender = gradosC/5;
/* Guarda en la variable "encender" la cantidad de LED que se encenderán. En este
caso, 1 por cada 5 grados centígrados, por
lo que el termómetro LED sólo puede indicar desde los 0°C hasta los 45°C. */

for (int i=0; i<encender; i++){
  digitalWrite (leds [i], HIGH);
}
/* Enciende los LED desde el primero hasta el indicado por la variable "encender". */

for (int i=encender; i<ledCantidad; i++){
  digitalWrite (leds [i], LOW);
}
/* Apaga los LED desde el indicado por la variable "encender" hasta el último
(ledCantidad). */
delay(5000);
/* Espera 5 segundos. */
}
/* ----- */

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad los valores obtenidos desde el monitor serial y el termómetro LED.

Preguntas.

¿Por qué en las condiciones de las funciones “for” la variable “i” tiene valores desde 0 hasta menor que la variable “encender”?

Solución propuesta.

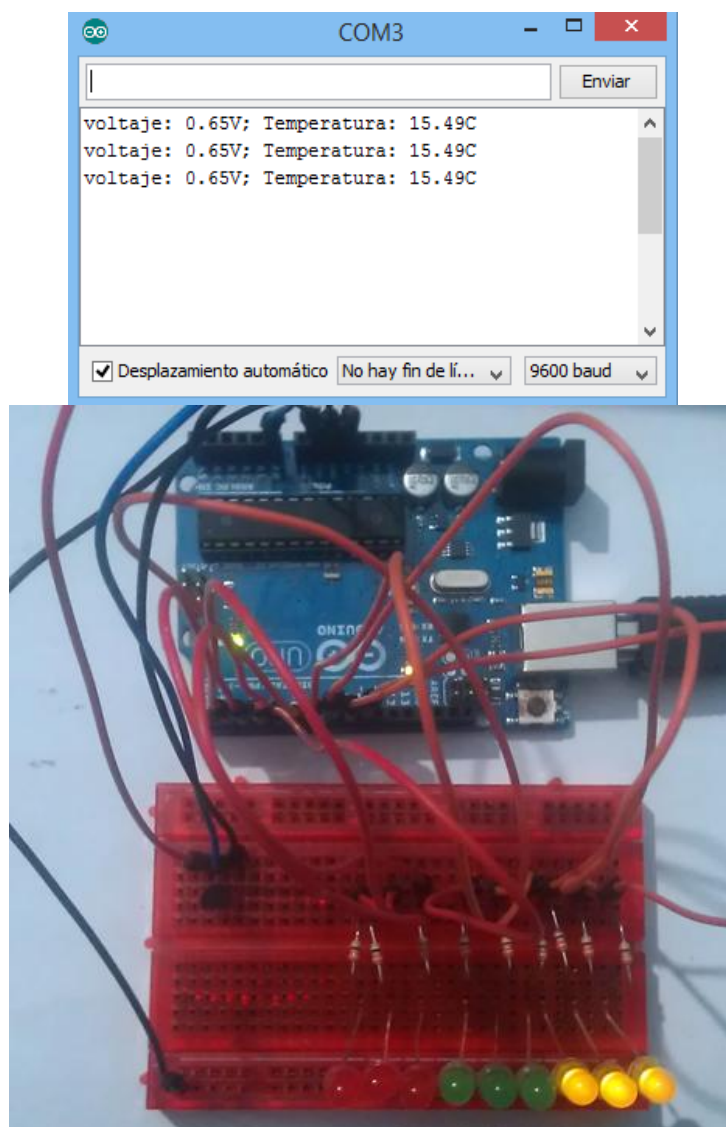


Figura 43, Solución propuesta Práctica 1.4.1.2. Fuente: Elaboración propia.

Por que las variables de tipo array, como la que recorre la orden contenida en esta función “for”, comienzan en la posición 0 y no la 1. Por ello, la variable led1 ocupa la posición 0 en el array leds y la variable led9 ocupa la posición 8.

### Práctica 1.4.1.3. Servomotor con sensor de flexión.

#### Enunciado.

En esta práctica utilizaremos el sensor de flexibilidad para interactuar con un servomotor. De este modo, según sobremos el sensor se moverá el servomotor, algo que, en la industria robótica, permite manejar maquinaria con guantes que integren sensores de flexión.

En primer lugar, construid el circuito en base al esquema y prototipo siguientes.

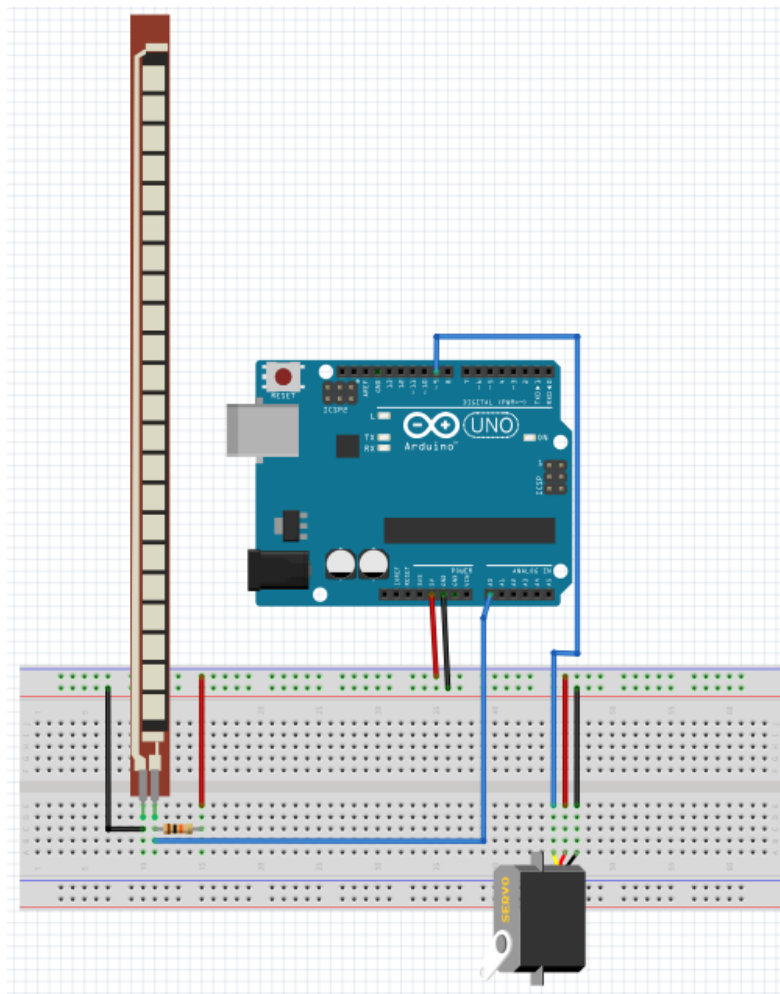
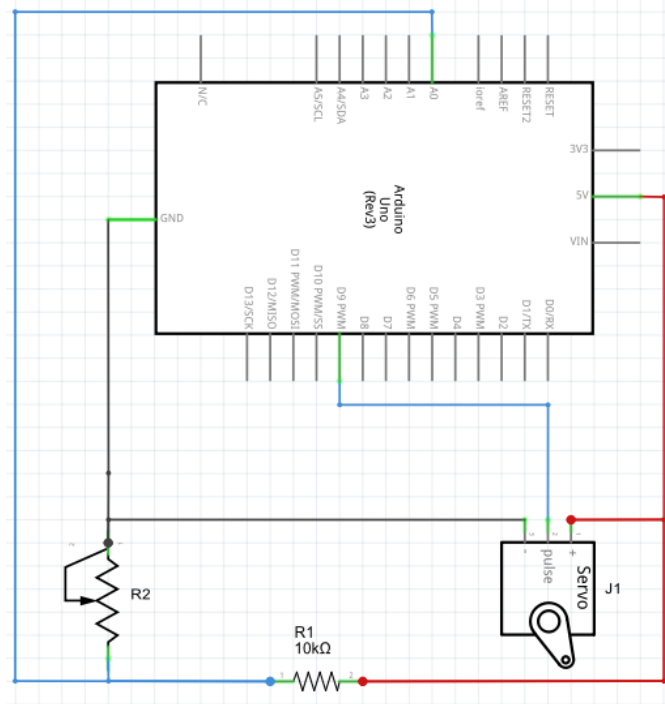


Figura 44, Esquema y prototipo de la Práctica 1.4.1.3. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código:

```

/* ----- */
/* Servomotor con sensor de flexión. */

#include <Servo.h>
/* Una librería es un conjunto de instrucciones para un fin determinado. El IDE Arduino
permite incluir en sus programas librerías que
aumenten las posibilidades de programación y simplifiquen las tareas. En este caso, la
librería a utilizar se llama Servo y posee
instrucciones para controlar servomotores. */

Servo servo1;
/* Crea un objeto de tipo Servo llamado servo1, que forma parte de la clase Servo.h. */

const int pinFlex = 0;
/* Define el pin del sensor de flexibilidad. */

void setup(){

  Serial.begin(9600);
  /* Inicia la comunicación serie. */

  servo1.attach(9);
  /* Define el pin de control del servo. */
}

void loop(){
  int posicionFlex;
  int posicionServo;
  /* Crea ls variables para las posiciones del sensor de flexibilidad y del servomotor. */

  posicionFlex = analogRead(pinFlex);
  /* Guarda en la variable "posicionFlex" el valor del sensor de flexibilidad (0 - 1023). */

  posicionServo = map(posicionFlex, 600, 900, 0, 180);
  posicionServo = constrain(posicionServo, 0, 180);

```

/\* El rango de actuación del sensor de flexibilidad utilizado es de 600 a 900 (dentro del rango de lectura de Arduino (0 - 1023).

Con la función map, plasmamos este rango en el rango de valores del servomotor (en este caso el servo puede adoptar posiciones de

entre 0 y 180 grados), que limitamos mediante la función constrain. PArá más información: <http://arduino.cc/en/Reference/HomePage>.

Por lo tanto, se guarda en la variable "posicionServo" la posición del servomotor en función de su rango de actuación y el valor

obtenido del sensor de flexibilidad. \*/

```
servo1.write(posicionServo);
```

```
/* Indica la posición en que se ha de situar el servomotor, según la variable "posicionServo". */
```

```
Serial.println("Sensor: " + String(posicionFlex) + " ; Servo: " + String(posicionServo) + ".");
```

```
/* Muestra en el monitor serie la información especificada. */
```

```
delay(20);
```

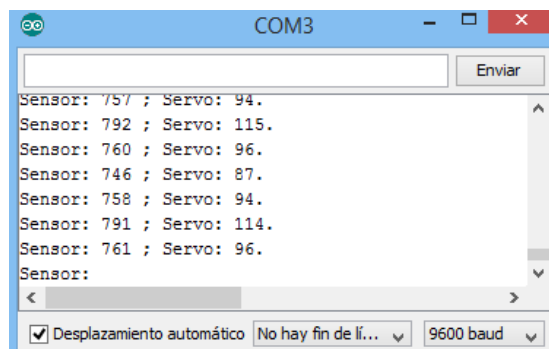
```
/* Espera 20 ms hasta actualizar el servomotor respecto al sensor de flexibilidad. */
```

```
}
```

```
/* ----- */
```

En tercer lugar, cargad el programa en la placa Arduino e interactuad con el sensor de flexibilidad observando los valores obtenidos desde el monitor serial y el movimiento del servomotor.

### Solución propuesta.



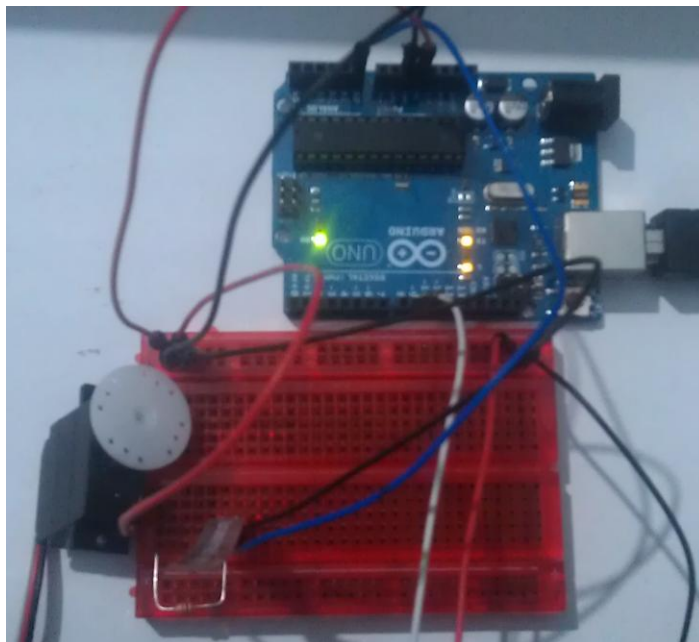


Figura 45, Solución propuesta Práctica 1.4.1.3. Fuente: Elaboración propia.

#### Práctica 1.4.1.4. Lógica con pulsadores.

##### Enunciado.

Poseemos un circuito con dos pulsadores y dos LED conectados a la placada Arduino mediante sus respectivas resistencias. Tras montar dicho circuito según el esquema de la figura 7, debéis completar el código adjunto para que:

- Si no se acciona ningún pulsador, no se enciende ningún LED.
- Si se acciona cualquier pulsador, se enciende sólo el LED rojo.
- Si se accionan ambos pulsadores, se enciende sólo el LED rojo.

Realiza la tabla lógica antes de comenzar con el código del programa.

¡Tened en cuenta que cuando se presiona un pulsador el valor recibido en el pin de entrada es 0 (LOW) y cuando no se presiona es 1 (HIGH).

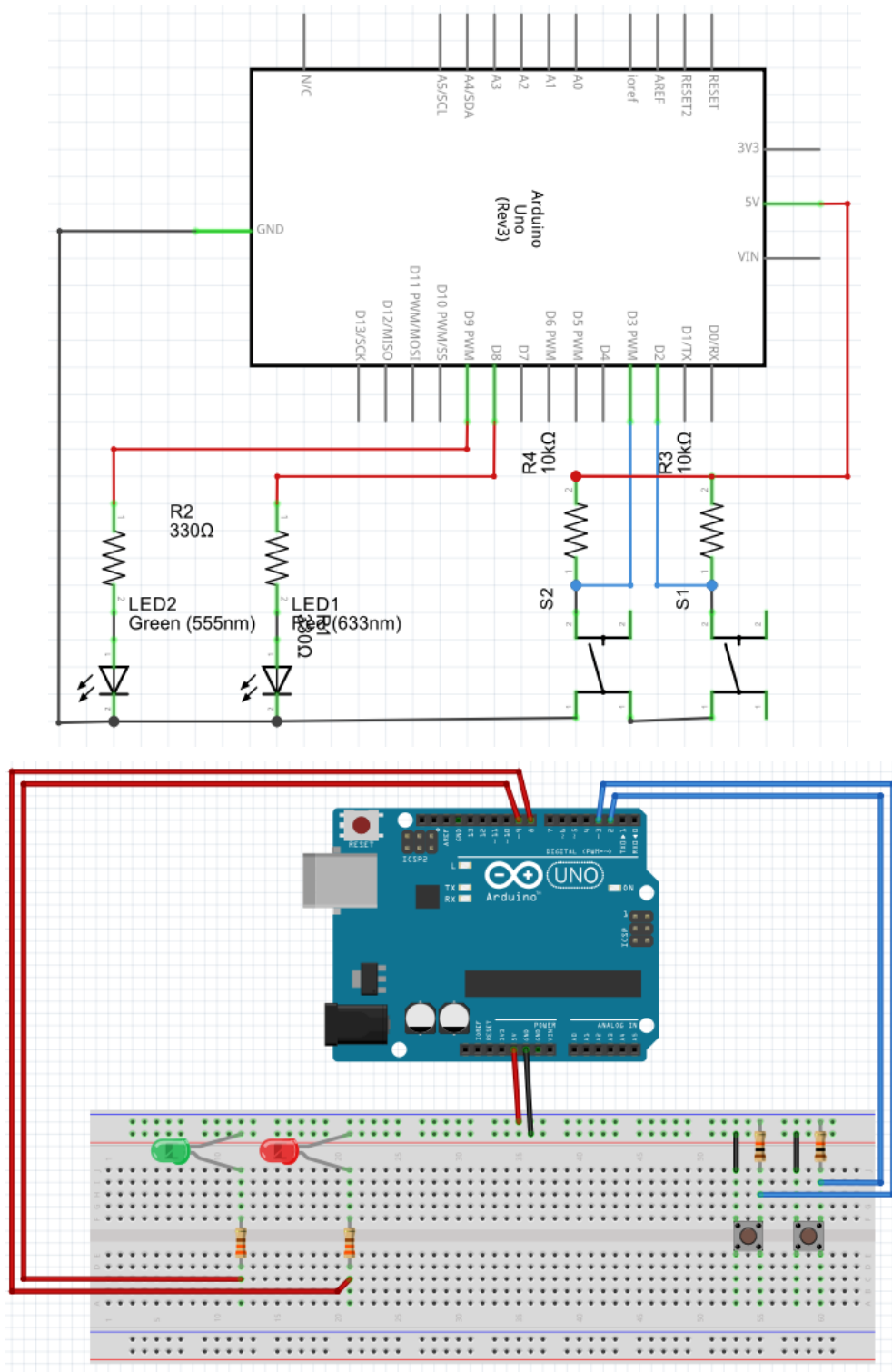


Figura 46, Esquema y prototipo de la Práctica 1.4.1.4. Fuente: Elaboración propia con Fritzing.

/\* ----- \*/  
 /\* Lógica con pulsadores. \*/

```
int pinLedRojo = 8;
int pinLedVerde = 9;
```



```

int pinPulsador1 = 2;
int pinPulsador2 = 3;
/* Asignación de los pins utilizados. */

void setup(){

  pinMode(pinLedRojo, OUTPUT);
  pinMode(pinLedVerde, OUTPUT);
  pinMode(pinPulsador1, INPUT);
  pinMode(pinPulsador2, INPUT);
  /* Establece los pins asociados a los LED como salida y los asociados a los
  pulsadores como entrada. */
}
void loop(){

  boolean pulsador1 = digitalRead(pinPulsador1);
  boolean pulsador2 = digitalRead(pinPulsador2);
  /* Almacena en las variables booleans (0,1) "pulsador1" y "pulsador2" las lecturas
  de los pines asociados a los pulsadores.*/

  if (/* ESCRIBE AQUI TU CODIGO*/){

    digitalWrite(pinLedRojo, LOW);
    digitalWrite(pinLedVerde, HIGH);
    /* Si el pulsador 1 se encuentra accionado y el pulsador 2 se encuentra accionado,
    LED rojo apagado y LED verde encendido. */

  }else if (/* ESCRIBE AQUI TU CODIGO*/){

    digitalWrite(pinLedRojo, HIGH);
    digitalWrite(pinLedVerde, LOW);
    /* Si no, si el pulsador1 se encuentra accionado o el pulsador 2 se encuentra
    accionado, LED rojo encendido y LED verde apagado. */
  }else{

    digitalWrite(pinLedRojo, LOW);
    digitalWrite(pinLedVerde, LOW);
    /* Si no (ningún pulsador accionado), ambos LED apagados. */
  }
}
/* ----- */

```

Solución propuesta.

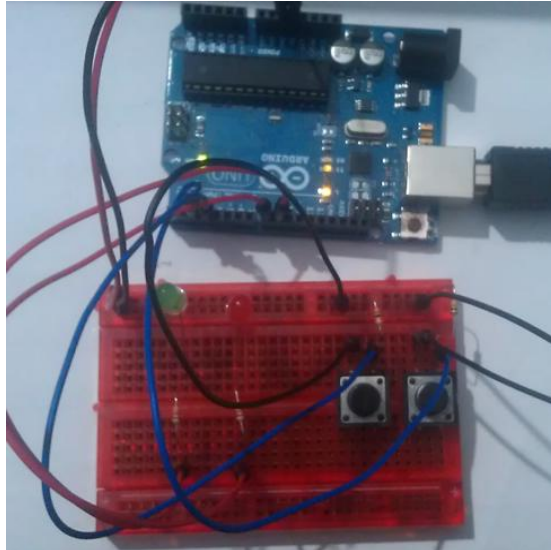


Figura 47, Solución propuesta Práctica 1.4.1.4. Fuente: Elaboración propia.

```

/* ----- */
/* Lógica con pulsadores. */

int pinLedRojo = 8;
int pinLedVerde = 9;

int pinPulsador1 = 2;
int pinPulsador2 = 3;
/* Asignación de los pins utilizados. */

void setup(){

  pinMode(pinLedRojo, OUTPUT);
  pinMode(pinLedVerde, OUTPUT);
  pinMode(pinPulsador1, INPUT);
  pinMode(pinPulsador2, INPUT);
  /* Establece los pins asociados a los LED como salida y los asociados a los
  pulsadores como entrada. */
}
void loop(){

  boolean pulsador1 = digitalRead(pinPulsador1);
  boolean pulsador2 = digitalRead(pinPulsador2);
  /* Almacena en las variables booleanas (0,1) "pulsador1" y "pulsador2" las lecturas
  de los pines asociados a los pulsadores.*/

  if (pulsador1 == LOW && pulsador2 == LOW){

    digitalWrite(pinLedRojo, LOW);
    digitalWrite(pinLedVerde, HIGH);
    /* Si el pulsador 1 se encuentra accionado y el pulsador 2 se encuentra accionado,
    LED rojo apagado y LED verde encendido. */

  }else if (pulsador1 == LOW|| pulsador2 == LOW){

    digitalWrite(pinLedRojo, HIGH);
    digitalWrite(pinLedVerde, LOW);
    /* Si no, si el pulsador1 se encuentra accionado o el pulsador 2 se encuentra
  
```

```

accionado, LED rojo encendido y LED verde apagado. */
}else{

    digitalWrite(pinLedRojo, LOW);
    digitalWrite(pinLedVerde, LOW);
    /* Si no (ningún pulsador accionado), ambos LED apagados. */
}
}
}
/* ----- */

```

Pulsador 1	Pulsador 2	LED verde	LED rojo
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Tabla 51, Solución propuesta Práctica 1.4.1.4.

**Práctica 1.4.1.5. Semáforo avanzado.**

**Enunciado.**

Se pretende instalar en la ciudad un nuevo paso de peatones regulado por un semáforo avanzado. Os han encargado el prototipo del proyecto, por lo que debéis elaborar el prototipado y esquema del circuito con Fritzing, la programación con el IDE Arduino y el prototipo con Arduino UNO. El funcionamiento del semáforo avanzado es:

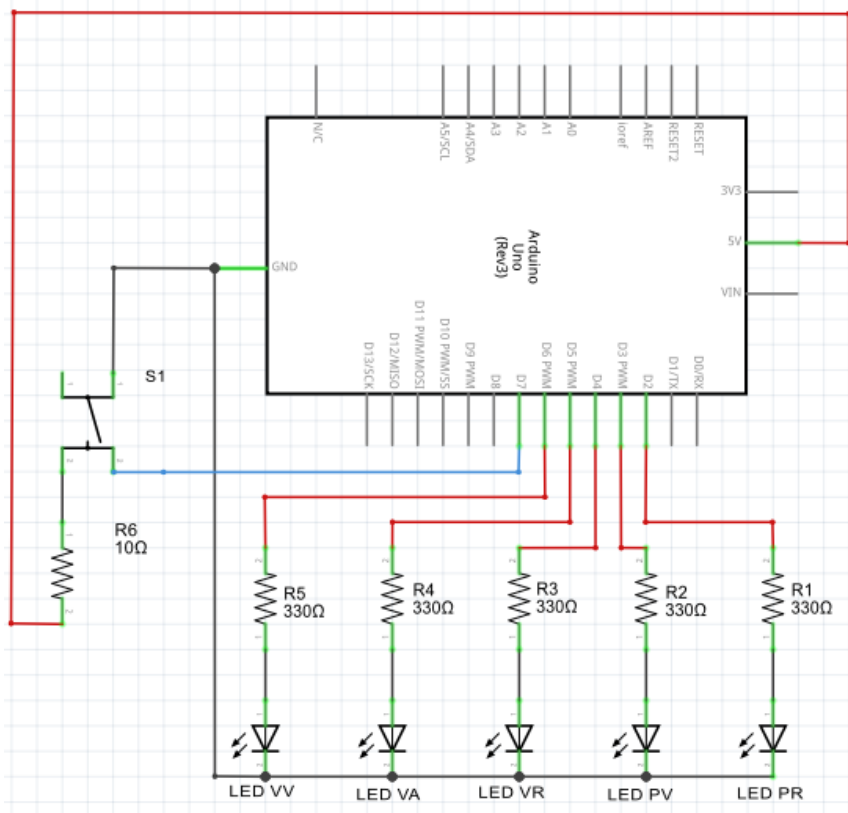
- Existen dos partes indicativas luminosas: La de los vehículos y la de los peatones.
- Vehículos:
  - Rojo: Dura 10 segundos, siempre sigue al Amarillo y supone Verde para los peatones.
  - Amarillo: Dura 2 segundos, siempre sigue al Verde.
  - Verde: Dura 15 segundos, siempre sigue al Rojo y supone Rojo para los peatones.
- Peatones:
  - Rojo: Dura 15 segundos, supone Verde o Amarillo para los vehículos.
  - Verde: Dura 10 segundos, supone Rojo para los vehículos.
- Los peatones disponen de un pulsador y al accionarlo:
  - Si el semáforo vehículos se encuentra en Verde, pasa a Amarillo.

Elabora la tabla de la verdad del circuito, diseña el prototipo y esquema del circuito, implementa el programa, monta el circuito y carga el programa en la placa Arduino.

Solución propuesta.

Botón	Vehículos			Peatones	
	Rojo	Amarillo	Verde	Rojo	Verde
0	0	0	1	1	0
0	0	1	0	1	0
0	1	0	0	0	1
1	0	1	0	1	0

Tabla 52, Solución propuesta Práctica 1.4.1.5.



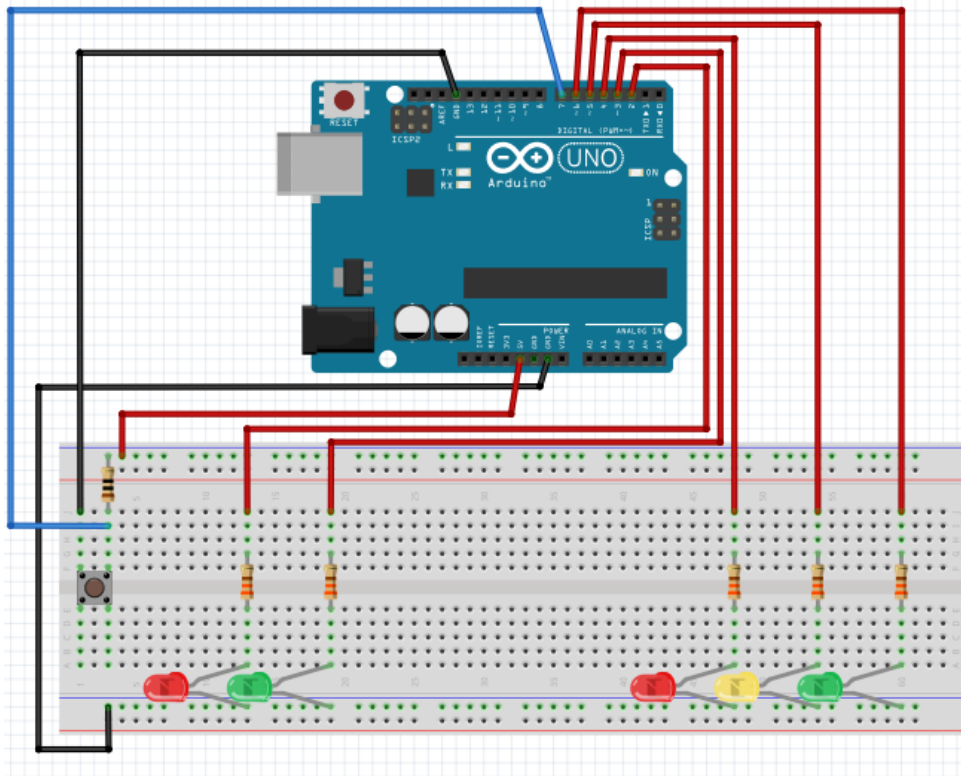


Figura 48, Solución propuesta Práctica 1.4.1.5. Fuente: Elaboración propia.

/\* ----- \*/  
 /\* Semáforo Avanzado. \*/

```
int pinPR = 2;
int pinPV = 3;
int pinVR = 4;
int pinVA = 5;
int pinVV = 6;
```

```
int pinBoton = 7;
/* Asignación de los pins utilizados. */
```

```
int espera[] = {15, 2, 10};
int i = 0;
int contar = espera[i];
/* Guarda en un array el tiempo de espera de cada estado del semáforo (Verde,
Amarillo, Rojo), crea la variable entera que hace de índice en el recorrido del
array (la inicializa a 0: Verde) y crea la variable que contendrá el tiempo transcurrido
en cada estado. */
```

```
void setup(){
```

```
pinMode(pinPR, OUTPUT);
pinMode(pinPV, OUTPUT);
pinMode(pinVR, OUTPUT);
pinMode(pinVA, OUTPUT);
pinMode(pinVV, OUTPUT);
pinMode(pinBoton, INPUT);
/* Establece los pins asociados a los LED como salida y al pulsador como entrada. */
```

```
}
```

```

void loop(){

  boolean solicitud = digitalRead(pinBoton);
  /* Almacena en la variable boobleana (0,1) solicitud la lectura del pin asociado al
  pulsador (El pulsador ha de mantenerse pulsado unos 2 segundos).*/

  if (contar == 0){
    i++;
    if (i>2){
      i = 0;
    }
    contar = espera[i];
  }
  /* Si la variable "contar" ha lleado a 0, significa que se ha agotado el tiempo de un
  estado del semáforo, por lo tando pasa al siguiente. Además, si i > 2 (3)
  significa que se ha llegado al último registro del array espera (semáforo en rojo) y ya
  no hay más registros, por lo tanto, i vuelve a valer 0 (verde). Tras esta
  última comprobación la variable contar adquiere el nuevo valor temporal del estado
  de semáforo correspondiente. */

  switch (i){
    case 0:{
      digitalWrite(pinPR, HIGH);
      digitalWrite(pinPV, LOW);
      digitalWrite(pinVR, LOW);
      digitalWrite(pinVA, LOW);
      digitalWrite(pinVV, HIGH);
      break;
    }
    case 1:{
      digitalWrite(pinPR, HIGH);
      digitalWrite(pinPV, LOW);
      digitalWrite(pinVR, LOW);
      digitalWrite(pinVA, HIGH);
      digitalWrite(pinVV, LOW);
      break;
    }
    case 2:{
      digitalWrite(pinPR, LOW);
      digitalWrite(pinPV, HIGH);
      digitalWrite(pinVR, HIGH);
      digitalWrite(pinVA, LOW);
      digitalWrite(pinVV, LOW);
      break;
    }
  }
  /* Ejecuta las órdenes correspondientes a los distintos estados del semáforo según
  la variable i. */

  if (solicitud == 0 && i == 0){
    contar = 0;
  }else{
    contar = contar-1;
  }
  /* Comprueba si se ha accionado el pulsador y si el estado del semáfor es Verde

```

```
(vehículos). Si es cierto, aplica ala variable contar el valor 0 para que en la
siguiente ejecución pase al siguiente estado (Amarillo). */
delay (1000);
}
```

```
/* ----- */
```

### Práctica 1.4.2.1. Ayuda en estacionamiento.

#### Enunciado.

En esta práctica introducimos el sensor de ultrasonidos, que nos sirve para medir distancias utilizando ultrasonidos. El sensor, envía una secuencia de ultrasonidos y espera a que sus ondas reboten en algún objeto y vuelvan a su receptor. Sabiendo la velocidad de la onda y el tiempo transcurrido hasta la recepción podemos calcular la distancia según la ecuación  $\text{velocidad} = \text{distancia} / \text{tiempo}$ . Utilizando esta tecnología junto a un zumbador y una serie de LED simularemos el funcionamiento de la prestación “Ayuda en el estacionamiento” que incorporan muchos modelos actuales de automóvil.

En primer lugar, construid el circuito en base al esquema y prototipo siguientes.

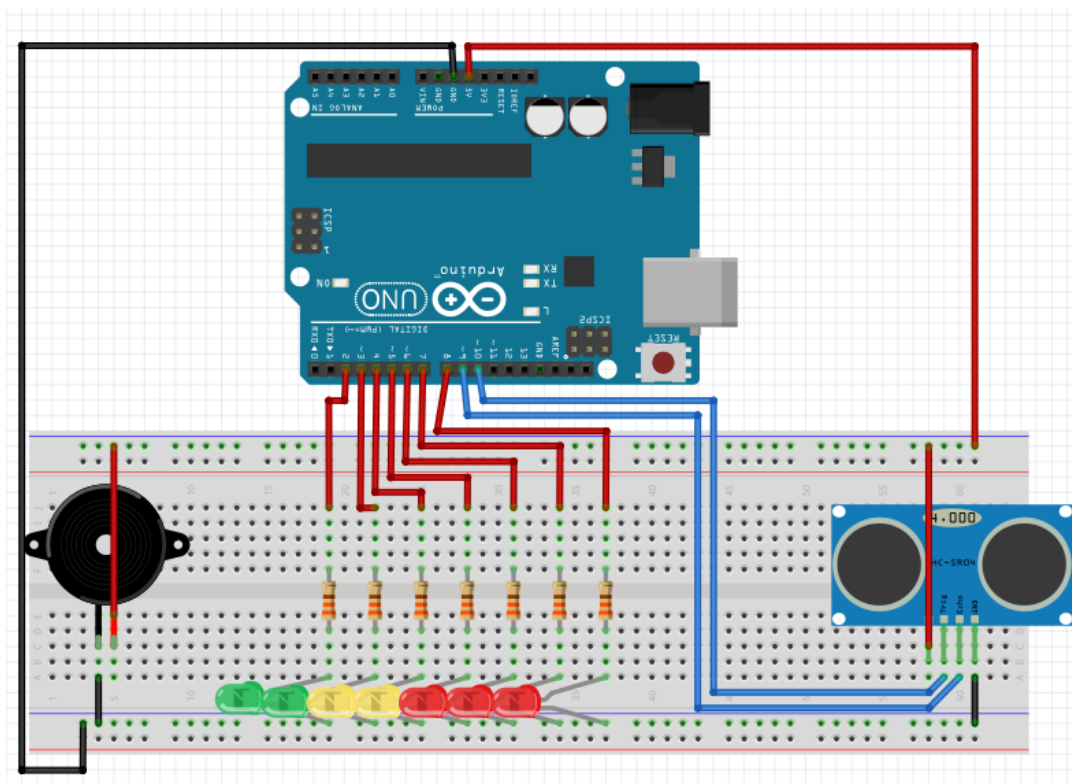
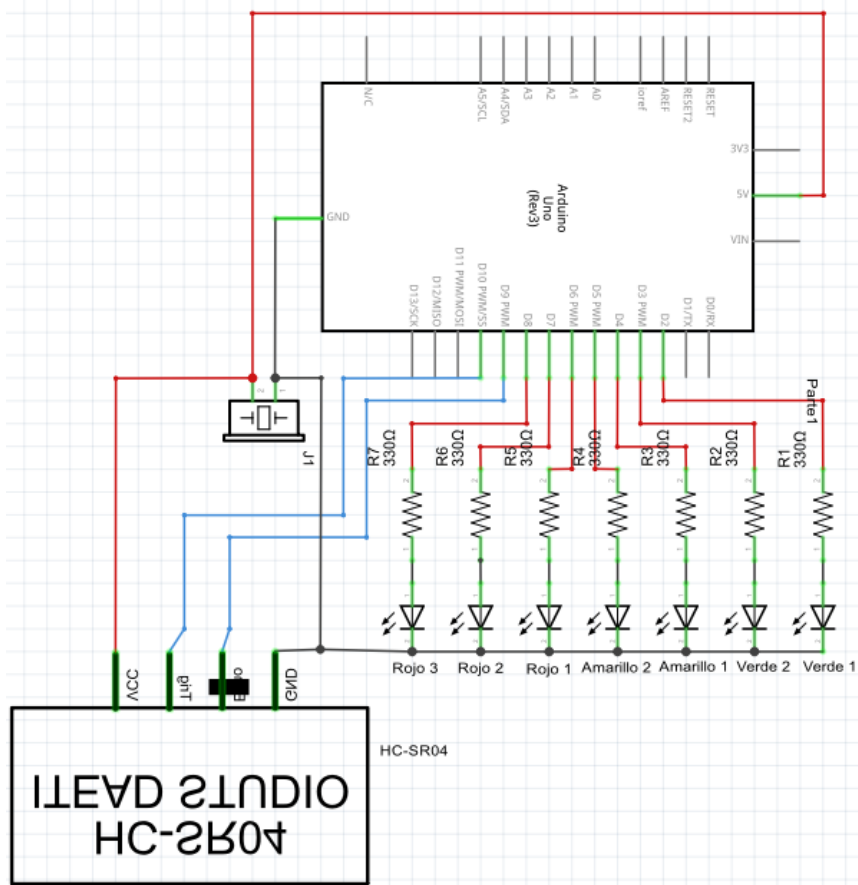


Figura 49, Esquema y prototipo de la Práctica 1.4.2.1. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código:



```
/* ----- */
/* Ayuda en estacionamiento. */

int pinVerde1 = 2;
int pinVerde2 = 3;
int pinAmarillo1 = 4;
int pinAmarillo2 = 5;
int pinRojo1 = 6;
int pinRojo2 = 7;
int pinRojo3 = 8;

int pinEcho = 9;
int pinTrig = 10;
int pinZumbador = 11;

int tiempo;
int distancia;

void setup () {
  Serial.begin (9600);

  pinMode (pinTrig, OUTPUT);
  pinMode (pinEcho, INPUT);
  pinMode (pinZumbador, OUTPUT);

  pinMode (pinVerde1, OUTPUT);
  pinMode (pinVerde2, OUTPUT);
  pinMode (pinAmarillo1, OUTPUT);
  pinMode (pinAmarillo2, OUTPUT);
  pinMode (pinRojo1, OUTPUT);
  pinMode (pinRojo2, OUTPUT);
  pinMode (pinRojo3, OUTPUT);
}
/* Configuramos los pins de entrada y salida. A tener en cuenta que el pin "Echo" del
sensor de ultrasonidos es de entrada (proporciona
```

los valores necesarios para calcular la distancia) y el "Trig" es de salida (accionamos el sensor mediante el envío de un pulso). \*/

```
void loop () {
  digitalWrite(pinTrig, LOW);
  delayMicroseconds(5);
  /* Apagamos el envío de pulsos al sensor y esperamos 5 microsegundos (Para
  estabilizar el funcionamiento del sensor). */
  digitalWrite (pinTrig, HIGH);
  delayMicroseconds (10);
  /* Enviamos un pulso al sensor para iniciar el funcionamiento y esperamos 10
  microsegundos. */
  tiempo = pulseIn (pinEcho, HIGH);
  /* Guardamos en la variable tiempo el instante en que el pinEcho (recepción del
  sensor) cambia de de LOW a HIGH. */
  distancia = int(tiempo*0.017);
  /* En base a los valores obtenidos calculamos la distancia (en valor entero y
  centímetros) según la ecuación proporcionada
  por el fabricante. */

  Serial.print (distancia);
  Serial.println (" cm");
  if(distancia == 0)
    distancia = 50;
  /* Si no existen objetos que hagan rebotar la señal, de la ecuación anterior obtemos
  que la distancia resulta igual a 0. Por tanto,
  indicamos que si se obtiene este valor, aplique una distancia fuera del rango que
  mediremos en los siguientes condicionales. */
  if (distancia <= 6) {tone (pinZumbador, 262, 200);
    digitalWrite(pinVerde1, HIGH);
    digitalWrite(pinVerde2, HIGH);
    digitalWrite(pinAmarillo1, HIGH);
    digitalWrite(pinAmarillo2, HIGH);
    digitalWrite(pinRojo1, HIGH);
    digitalWrite(pinRojo2, HIGH);
    digitalWrite(pinRojo3, HIGH);
    delay (10);
```

```
}else if (distancia > 6 && distancia <= 12) {  
  tone (pinZumbador, 1200, 100);  
  digitalWrite(pinVerde1, HIGH);  
  digitalWrite(pinVerde2, HIGH);  
  digitalWrite(pinAmarillo1, HIGH);  
  digitalWrite(pinAmarillo2, HIGH);  
  digitalWrite(pinRojo1, HIGH);  
  digitalWrite(pinRojo2, HIGH);  
  digitalWrite(pinRojo3, LOW);  
  delay (100);  
}  
}else if (distancia > 12 && distancia <= 18) {  
  tone (pinZumbador, 1000, 100);  
  digitalWrite(pinVerde1, HIGH);  
  digitalWrite(pinVerde2, HIGH);  
  digitalWrite(pinAmarillo1, HIGH);  
  digitalWrite(pinAmarillo2, HIGH);  
  digitalWrite(pinRojo1, HIGH);  
  digitalWrite(pinRojo2, LOW);  
  digitalWrite(pinRojo3, LOW);  
  delay (500);  
}  
}else if (distancia > 18 && distancia <= 24) {  
  tone (pinZumbador, 800, 100);  
  digitalWrite(pinVerde1, HIGH);  
  digitalWrite(pinVerde2, HIGH);  
  digitalWrite(pinAmarillo1, HIGH);  
  digitalWrite(pinAmarillo2, HIGH);  
  digitalWrite(pinRojo1, LOW);  
  digitalWrite(pinRojo2, LOW);  
  digitalWrite(pinRojo3, LOW);  
  delay (1000);  
}  
}else if (distancia > 24 && distancia <= 30) {  
  tone (pinZumbador, 600, 100);  
  digitalWrite(pinVerde1, HIGH);  
  digitalWrite(pinVerde2, HIGH);  
  digitalWrite(pinAmarillo1, HIGH);  
  digitalWrite(pinAmarillo2, LOW);  
  digitalWrite(pinRojo1, LOW);
```

```

digitalWrite(pinRojo2, LOW);
digitalWrite(pinRojo3, LOW);
delay (1250);
}else if (distancia > 30 && distancia <= 36) {
  tone (pinZumbador, 400, 100);
  digitalWrite(pinVerde1, HIGH);
  digitalWrite(pinVerde2, HIGH);
  digitalWrite(pinAmarillo1, LOW);
  digitalWrite(pinAmarillo2, LOW);
  digitalWrite(pinRojo1, LOW);
  digitalWrite(pinRojo2, LOW);
  digitalWrite(pinRojo3, LOW);
  delay (1300);
}else if (distancia > 36 && distancia <= 42) {
  tone (pinZumbador, 262, 100);
  digitalWrite(pinVerde1, HIGH);
  digitalWrite(pinVerde2, LOW);
  digitalWrite(pinAmarillo1, LOW);
  digitalWrite(pinAmarillo2, LOW);
  digitalWrite(pinRojo1, LOW);
  digitalWrite(pinRojo2, LOW);
  digitalWrite(pinRojo3, LOW);
  delay (1500);
}else if (distancia > 42){
  digitalWrite(pinVerde1, LOW);
  digitalWrite(pinVerde2, LOW);
  digitalWrite(pinAmarillo1, LOW);
  digitalWrite(pinAmarillo2, LOW);
  digitalWrite(pinRojo1, LOW);
  digitalWrite(pinRojo2, LOW);
  digitalWrite(pinRojo3, LOW);
}
noTone (pinZumbador);
}

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad los valores obtenidos desde el monitor serial según acercáis objetos al sensor de ultrasonido.

## Solución propuesta.

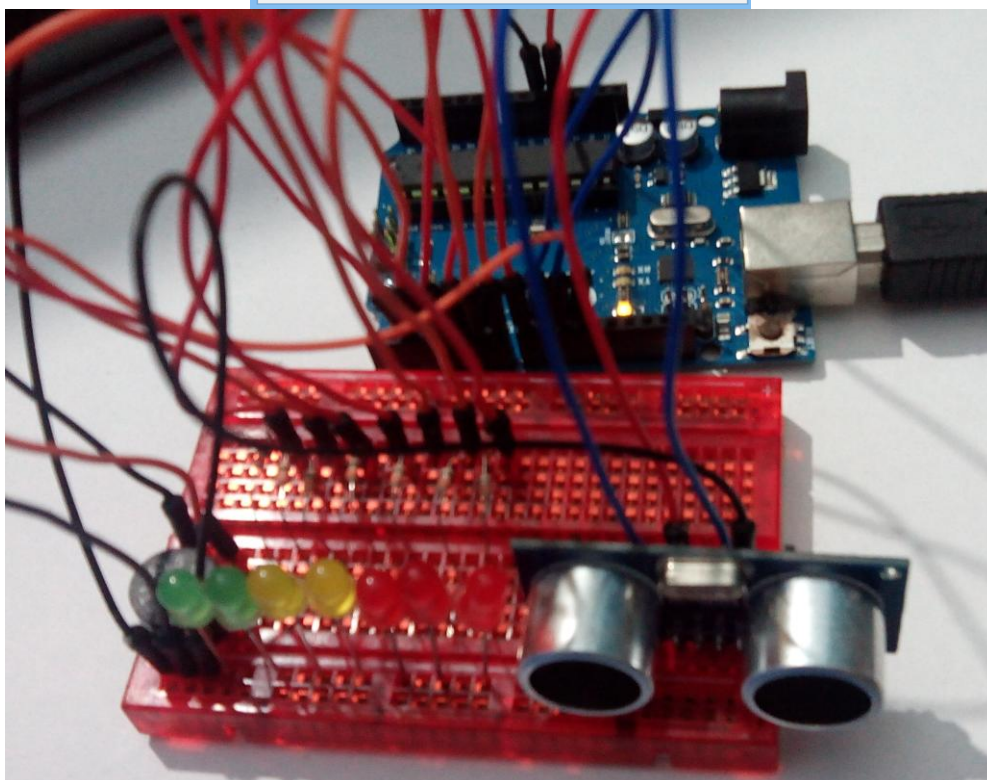
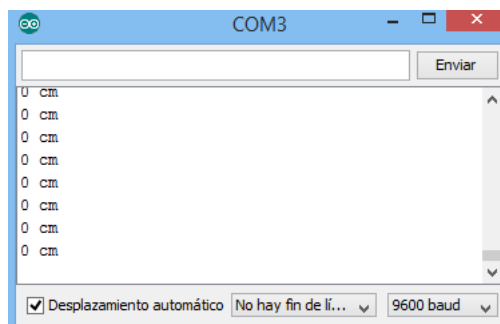


Figura 50, Solución propuesta Práctica 1.4.2.1. Fuente: Elaboración propia.

### Práctica 1.4.2.2. Puerta automática.

#### Enunciado.

Se quiere construir una puerta automática que se abra al detectar la presencia de objetos y se cierre al dejar de detectarlos, como la de multitud de establecimientos. Para realizar el prototipo del sistema automático disponemos de un servo motor y un sensor de infrarrojos (IR SHARP). Debéis diseñar, programar e implementar la puerta en base a los conocimientos adquiridos en las prácticas anteriores y según el siguiente fragmento de código.

/\* ----- \*/

```

/* Puerta automática. */

#include <Servo.h>
Servo servo;
/* Incluye la librería de servomotores y crea una variable de tipo servo (objeto de la
clase servo). */

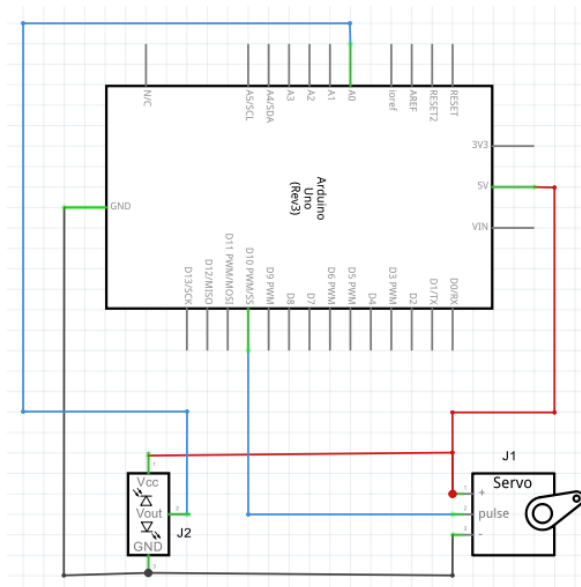
int pinSensor = 0;

void setup () {
  servo.attach(10);
}
/* Indica el pin al que se encuentra conectado el servo motor (10). */
void loop (){
  /* ESCRIBE TU CÓDIGO. */
}
/* ----- */

```

Recordad que para observar los distintos valores entregados por el sensor y calibrar adecuadamente la apertura y cierre de la puerta podéis utilizar las opciones de la función “Serial”.

Solución propuesta.



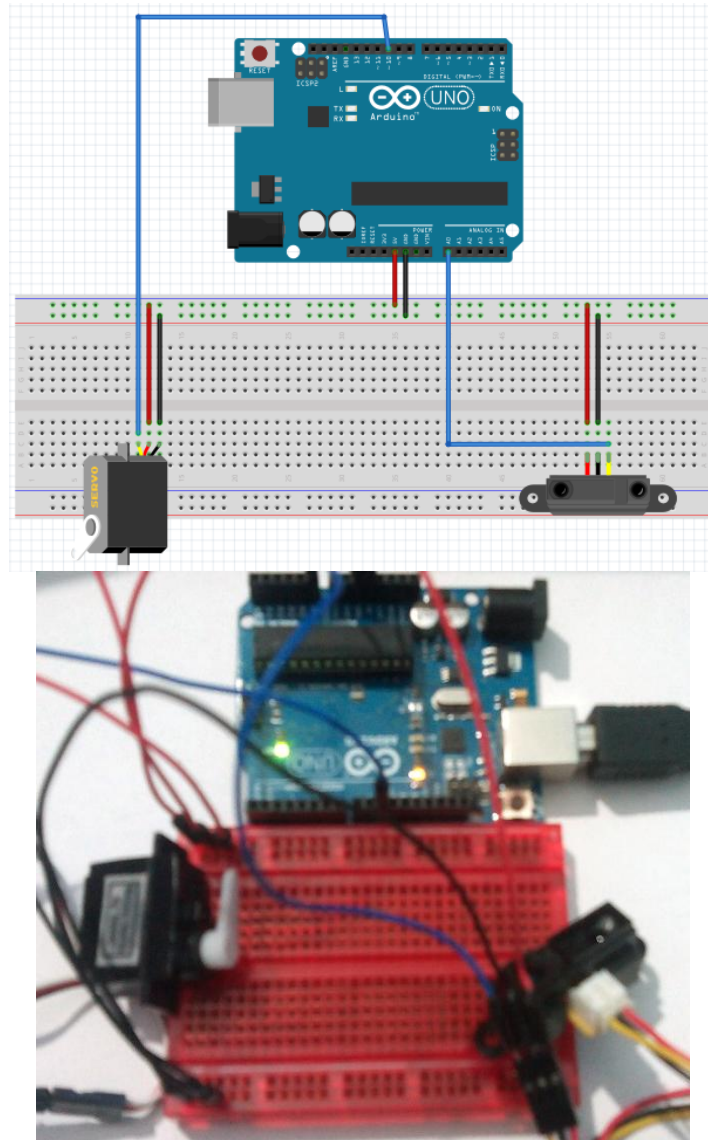


Figura 51, Solución propuesta Práctica 1.4.2.2. Fuente: Elaboración propia.

```

/* ----- */
/* Puerta automática. */

#include <Servo.h>
Servo servo;
/* Incluye la librería de servomotores y crea una variable de tipo servo (objeto de la
clase servo). */

int pinSensor = 0;

void setup () {
  servo.attach(10);
}
/* Indica el pin al que se encuentra conectado el servo motor (10). */
void loop (){
  float valor = analogRead(pinSensor);
  int posicionServo;
  /* Guarda en la variable valor la lectura del sensor y crea una variable donde indicar la
posición del servomotor. */
  if(valor >= 350){

```

```

    posicionServo = 90;
  }else{
    posicionServo = 0;
  }
  /* Si el sensor envia un valor mayor a 350 (tiene un objeto justo delante) la variable
  que indica la posicion del sensor
  guarda el valor 90 grados (puerta abierta). De lo contrario, guarda el valor 0 grados
  (puerta cerrada). */
  servo.write(posicionServo);
  delay(100);
  /* Posiciona el servomotor según la variable posicionServo y espera 0,1 segundos. */
}
/* ----- */

```

### Práctica 1.4.2.3. Proyecto libre: Vehículo autónomo.

#### Enunciado.

Esta práctica conforma la realización de un proyecto en el que construiréis un sistema automático utilizando el conjunto de conocimientos y habilidades adquirido durante el desarrollo de las actividades anteriores. El sistema ha de ser capaz de funcionar y regularse por sí mismo, como por ejemplo un vehículo móvil que detecte obstáculos y los evite o un sistema domótico. La implementación del proyecto ha de ser dividida en tareas más sencillas, según la metodología de proyectos. Los pasos a seguir son:

1. Puesta en común de soluciones propuestas y decisión final (grupo).
2. Croquis de la solución adoptada.
3. Prototipado y esquema del circuito electrónico (Fritzing).
4. Listado de material.
5. Presupuesto.
6. Plano a escala 1:2 de la estructura a construir.
7. Construcción de la estructura.
8. Realización del circuito.
9. Ensamblaje de componentes.
10. Programación del *script*.
11. Puesta en marcha.

#### Solución propuesta.



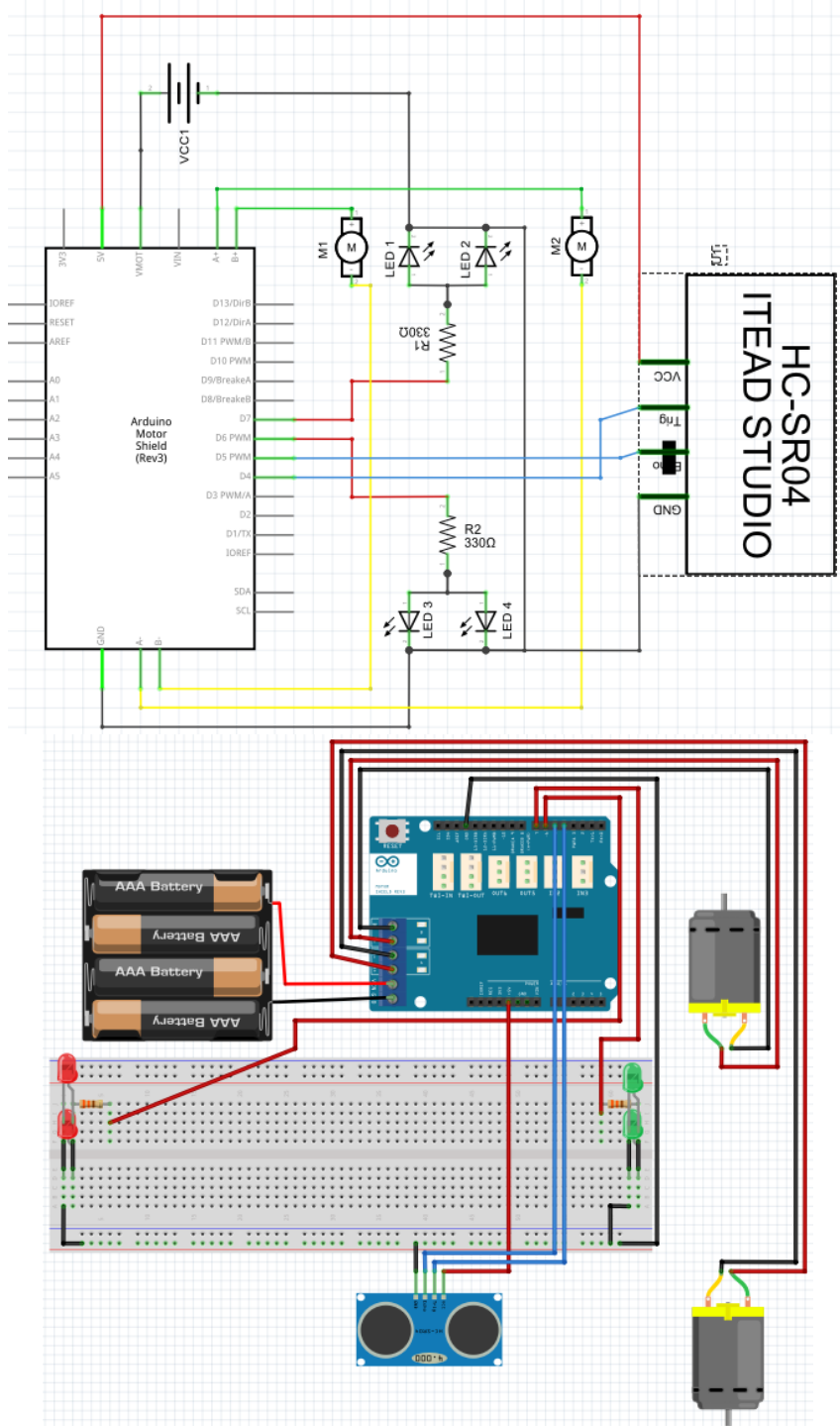


Figura 52, Solución propuesta Práctica 1.4.2.3. Fuente: Elaboración propia.

/\* ----- \*/  
 /\* Proyecto libre: vehículo autónomo. \*/

int rojo = 6;  
 int verde = 7;

int pinEcho = 9;  
 int pinTrig = 10;

int tiempo, distancia, accion;

```

void setup() {

  Serial.begin(9600);
  /* Inicialización de la comunicación. */

  pinMode (12, OUTPUT); /* Pin del motor A. */
  pinMode (9, OUTPUT); /* Pin del freno A. */

  pinMode (13, OUTPUT); /* Pin del motor B. */
  pinMode (8, OUTPUT); /* Pin del freno B. */

  pinMode (verde, OUTPUT);
  pinMode (rojo, OUTPUT);

  pinMode (pinTrig, OUTPUT);
  pinMode (pinEcho, INPUT);

  /* Configuración de los pins utilizados. */
}

int obtenerDistancia(){
  digitalWrite(pinTrig, LOW);
  delayMicroseconds(5);
  /* Apagamos el envío de pulsos al sensor y esperamos 5 microsegundos (Para
  estabilizar el funcionamiento del sensor). */
  digitalWrite (pinTrig, HIGH);
  delayMicroseconds (10);
  /* Enviamos un pulso al sensor para iniciar el funcionamiento y esperamos 10
  microsegundos. */
  tiempo = pulseIn (pinEcho, HIGH);
  /* Guardamos en la variable tiempo el instante en que el pinEcho (recepción del
  sensor) cambia de de LOW a HIGH. */
  return int(tiempo*0.017);
}

void loop() {

  distancia = obtenerDistancia();

  if(distancia<=30)
    accion = 2; /* Girar a la izquierda. */
  else
    accion = 1; /* Avanzar. */

  switch(accion){

    case 1: { /* Avanzar. */

      digitalWrite (verde, HIGH);
      digitalWrite (rojo, LOW);

      digitalWrite (12, HIGH); /* Motor A marcha alante. */
      digitalWrite (9, LOW); /* Freno A desbloqueado. */
      analogWrite (3, 255); /* Velocidad A máxima. */

      digitalWrite (13, HIGH); /* Motor B marcha alante. */

```

```

digitalWrite (8, LOW); /* Freno B desbloqueado. */
analogWrite (11, 255); /* Velocidad B máxima. */

break;
}
case 2: { /* Giro izquierda. */
digitalWrite (verde, LOW);
digitalWrite (rojo, LOW);

digitalWrite (12, HIGH); /* Motor A marcha alante. */
digitalWrite (9, LOW); /* Freno A desbloqueado. */
analogWrite (3, 255); /* Velocidad A máxima. */

digitalWrite (13, LOW); /* Motor A marcha atrás. */
digitalWrite (8, LOW); /* Freno A desbloqueado. */
analogWrite (11, 255); /* Velocidad A máxima. */

break;
}
}
}
/* ----- */

```

### Práctica 2.2.1.1. Piano.

#### Enunciado.

En esta práctica construiremos un simulador de piano. Para crear este sistema de control en lazo abierto utilizaremos un potenciómetro de presión como elemento captador, la placa Arduino como elemento de proceso y el zumbador piezo eléctrico como elemento de salida.

En primer lugar, construid el circuito en base al esquema y prototipo de la figura siguiente teniendo en cuenta que el objeto del potenciómetro de presión no se encuentra disponible, a priori, en fritzing y se ha utilizado en su lugar el objeto sensor de presión, que incluye 2 puntos de conexión en lugar de 3. Por ello, hay que considerar que los conectores de los extremos se conectan uno a 5V y el otro a tierra mientras que el de en medio se conecta a la resistencia de 10k y a la entrada analógica 0 de Arduino.

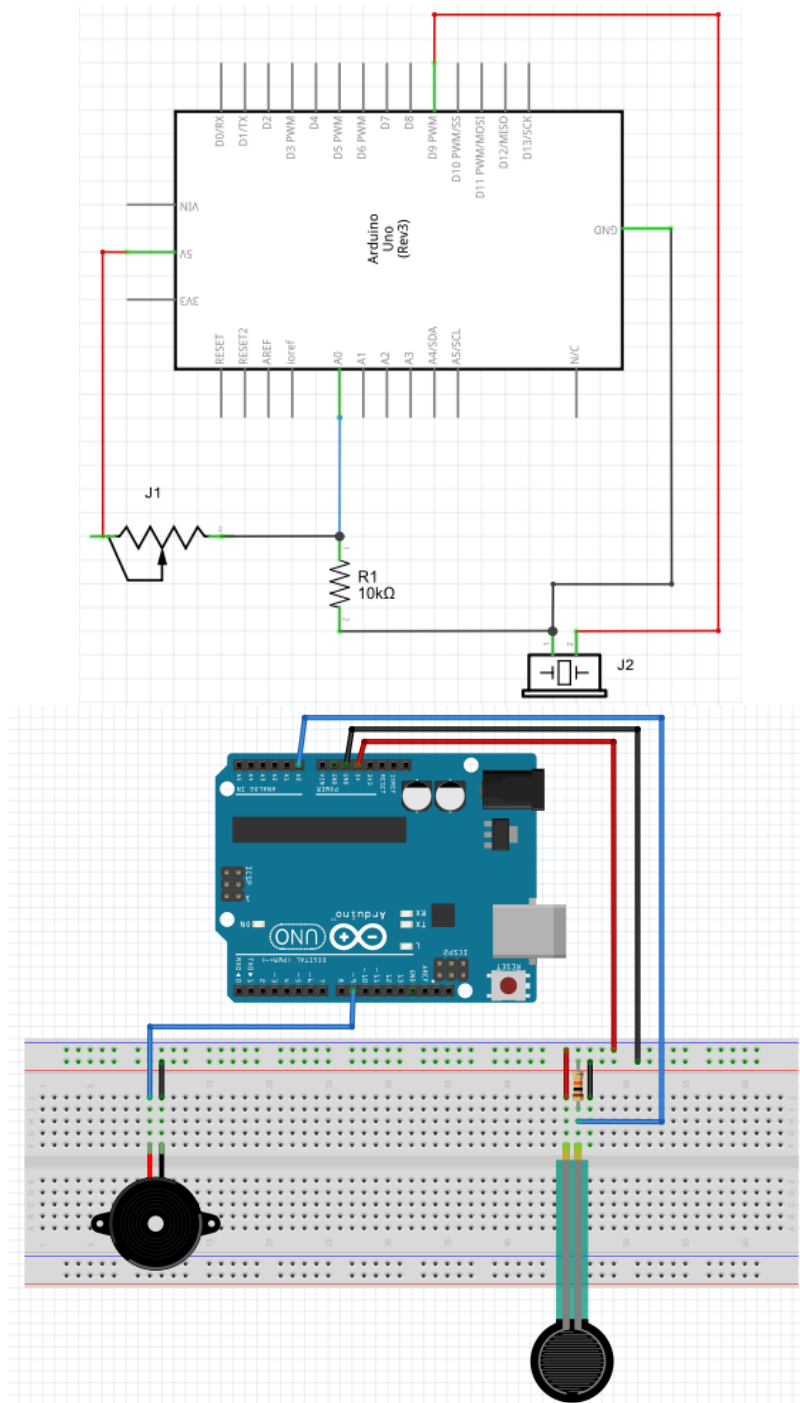


Figura 53, Esquema y prototipo de la Práctica 2.2.1.1. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa y copiad el siguiente código:

```

/* ----- */
/* Piano. */

int zumbador = 9;
int sensor = 0;
/* Asignamos los pins del zumbador y el sensor. */
    
```

```

float DO = 261.1, RE = 293.7, MI = 329.6, FA = 349.2, SOL = 392, LA = 440, SI =
493.9, DOBIS = 523.3;
float nota;
/* Creamos las variables que contienen las frecuencias de cada nota musical y una
variable "nota" que indicará la nota a sonar por el zumbador. */

void setup(){
pinMode (zumbador, OUTPUT);
}

void loop(){
  int valorSensor;
  valorSensor = analogRead(0);
  /* Lee el valor entregado por el sensor de presión y lo almacena en la variable
"valorSensor". */
  float valor = map(valorSensor, 0, 1023, 0, 800);
  /* Pasa el valor obtenido en rango 0-1023 al rango 0-800 (100 valores para cada nota
musical). */
  if(valor != 0){
    if(valor <=100)
      nota = DO;
    else if(valor >100 && valor <=200)
      nota = RE;
    else if(valor >200 && valor <=300)
      nota = MI;
    else if(valor >300 && valor <=400)
      nota = FA;
    else if(valor >400 && valor <=500)
      nota = SOL;
    else if(valor >500 && valor <=600)
      nota = LA;
    else if(valor >600 && valor <=700)
      nota = SI;
    else if (valor >700)
      nota = DOBIS;
    tone(zumbador, nota, 100);
    /* Dependiendo del valor de la variable "valor" la variable "nota" adquiere el valor de
las distintas variables que contienen las frecuencias de
cada nota musical. La función "tone" envía un pulso al pin "zumbador" con una
frecuencia "nota" durante 100 milisegundos. */
  }else
    noTone(zumbador);
  /* Si el valor de "valor" es distinto a "0" la variable "nota" adquiere el valor
correspondiente y se ejecuta la función "tone". De lo contrario,
mediante la función "noTone" se detiene en envío de señal al zumbador. */
}

/* ----- */

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad el funcionamiento, si todo ha salido bien, deslizando un dedo desde un extremo del sensor hasta el otro el zumbador debe emitir la escala de Do mayor.

#### Solución propuesta.

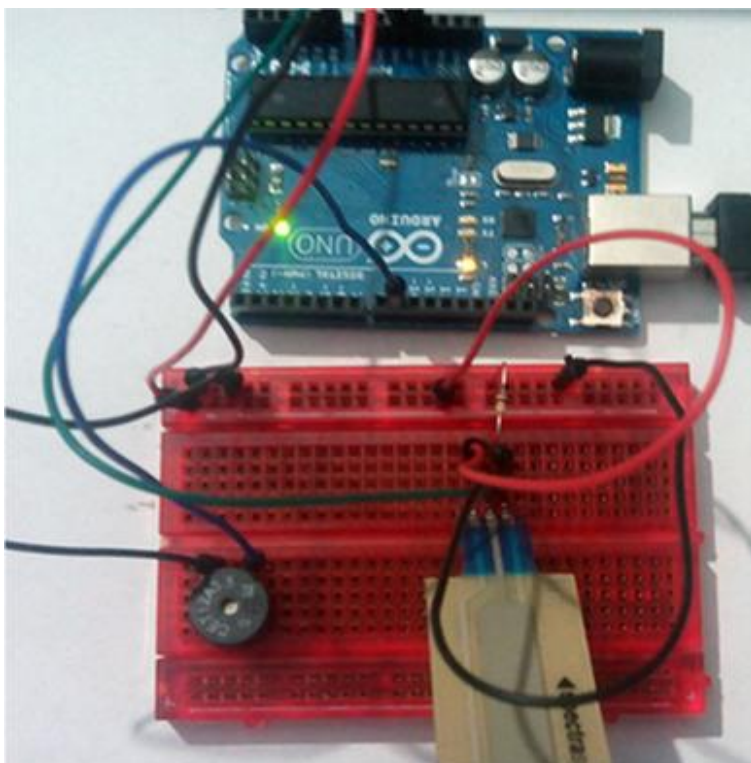


Figura 54, Solución propuesta práctica 2.2.1.1. Fuente: Elaboración propia.

#### Práctica 2.2.1.2. Contador de 0 a 9.

##### Enunciado.

En esta práctica comprobaremos el funcionamiento del display de 7 segmentos. Para ello, implementaremos un contador de 0 a 9 con la ayuda de Arduino y representaremos los dígitos correspondientes sobre el display de ánodo común.

En primer lugar, construid el circuito en base al esquema y prototipo de la figura siguiente teniendo en cuenta que el modelo de display empleado en el circuito difiere del representado en el esquema mediante fritzing. Por ello, debéis guiaros por las indicaciones correspondientes al display de 7 segmentos D5611 A/B.



Figura 55, Esquema del D5611 A/B. Fuente: Elaboración propia.

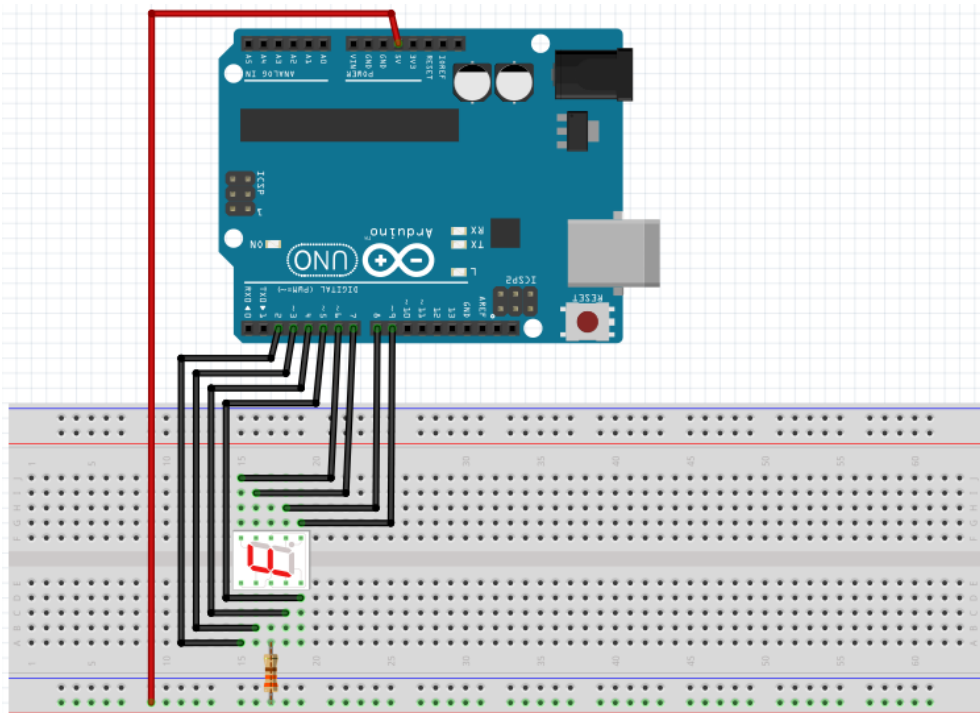
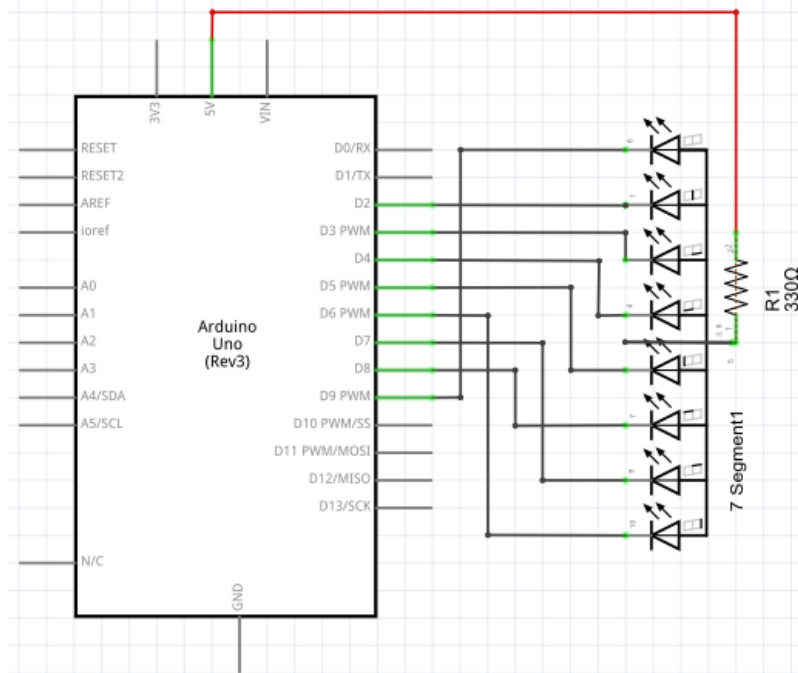


Figura 56, Esquema y prototipo de la Práctica 2.2.1.2. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código:

```

/* ----- */
/* Contador de 0 a 9. */

int e = 2;
int d = 3;
int c = 4;
int b = 5;
int a = 6;
int f = 7;
int g = 8;
/* Especificamos los pins a los que corresponde cada parte del display. */

void setup(){
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  /* Indicamos que los pins digitales son de salida. */
}

void loop(){

  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,LOW);
  digitalWrite(g,HIGH);
  /* Configuración correspondiente al "0". */
  delay(1000);
  /* Espera 1 segundo. */

  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
  /* Configuración correspondiente al "1". */
  delay(1000);
  /* Espera 1 segundo. */

  /* ESCRIBE AQUI TU CÓDIGO. */
  /* Configuración correspondiente al "2". */
  delay(1000);
  /* Espera 1 segundo. */

```



```

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "3". */
delay(1000);
/* Espera 1 segundo. */

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "4". */
delay(1000);
/* Espera 1 segundo. */

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "5". */
delay(1000);
/* Espera 1 segundo. */

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "6". */
delay(1000);
/* Espera 1 segundo. */

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "7". */
delay(1000);
/* Espera 1 segundo. */

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "8". */
delay(1000);
/* Espera 1 segundo. */

/* ESCRIBE AQUI TU CÓDIGO. */
/* Configuración correspondiente al "9". */
delay(1000);
/* Espera 1 segundo. */
}
/* ----- */

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad el funcionamiento, si todo ha salido bien, el display debe mostrar una cuenta ascendente desde el 0 hasta el 9 con un segundo entre un número y el siguiente.

#### Solución propuesta.

```

/* ----- */
/* Contador de 0 a 9. */

int e = 2;
int d = 3;
int c = 4;
int b = 5;
int a = 6;
int f = 7;
int g = 8;

```

```

/* Especificamos los pins a los que corresponde cada parte del display. */

void setup(){
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  /* Indicamos que los pins digitales son de salida. */
}

void loop(){

  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,LOW);
  digitalWrite(g,HIGH);
  /* Configuración correspondiente al "0". */
  delay(1000);
  /* Espera 1 segundo. */

  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
  /* Configuración correspondiente al "1". */
  delay(1000);
  /* Espera 1 segundo. */

  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,HIGH);
  digitalWrite(g,LOW);
  /* Configuración correspondiente al "2". */
  delay(1000);
  /* Espera 1 segundo. */

  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,LOW);

```

```

/* Configuración correspondiente al "3". */
delay(1000);
/* Espera 1 segundo. */

```

```

digitalWrite(a,HIGH);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
/* Configuración correspondiente al "4". */
delay(1000);
/* Espera 1 segundo. */

```

```

digitalWrite(a,LOW);
digitalWrite(b,HIGH);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,HIGH);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
/* Configuración correspondiente al "5". */
delay(1000);
/* Espera 1 segundo. */

```

```

digitalWrite(a,LOW);
digitalWrite(b,HIGH);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
/* Configuración correspondiente al "6". */
delay(1000);
/* Espera 1 segundo. */

```

```

digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,HIGH);
digitalWrite(e,HIGH);
digitalWrite(f,HIGH);
digitalWrite(g,HIGH);
/* Configuración correspondiente al "7". */
delay(1000);
/* Espera 1 segundo. */

```

```

digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);

```

```

/* Configuración correspondiente al "8". */
delay(1000);
/* Espera 1 segundo. */

digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,HIGH);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
/* Configuración correspondiente al "9". */
delay(1000);
/* Espera 1 segundo. */
}
/* ----- */

```

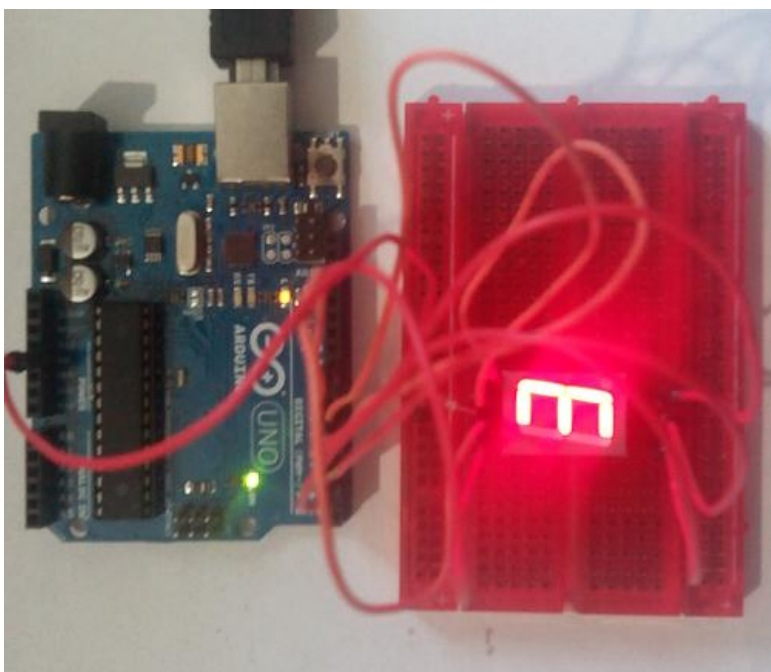


Figura 57, Solución propuesta Práctica 2.2.1.2. Fuente: Elaboración propia.

### Práctica 2.2.1.3. Puertas lógicas.

#### Enunciado.

En esta práctica implementaremos el funcionamiento de las puertas lógicas AND, OR, NAND, NOR y XOR utilizando dos pulsadores como elementos de entrada binaria, la programación de funciones en Arduino como proceso para implementar las puertas lógicas y un display de 7 segmentos como elemento de salida para mostrar el valor obtenido (0 o 1).

En primer lugar, construid el circuito en base al esquema y prototipo siguientes.

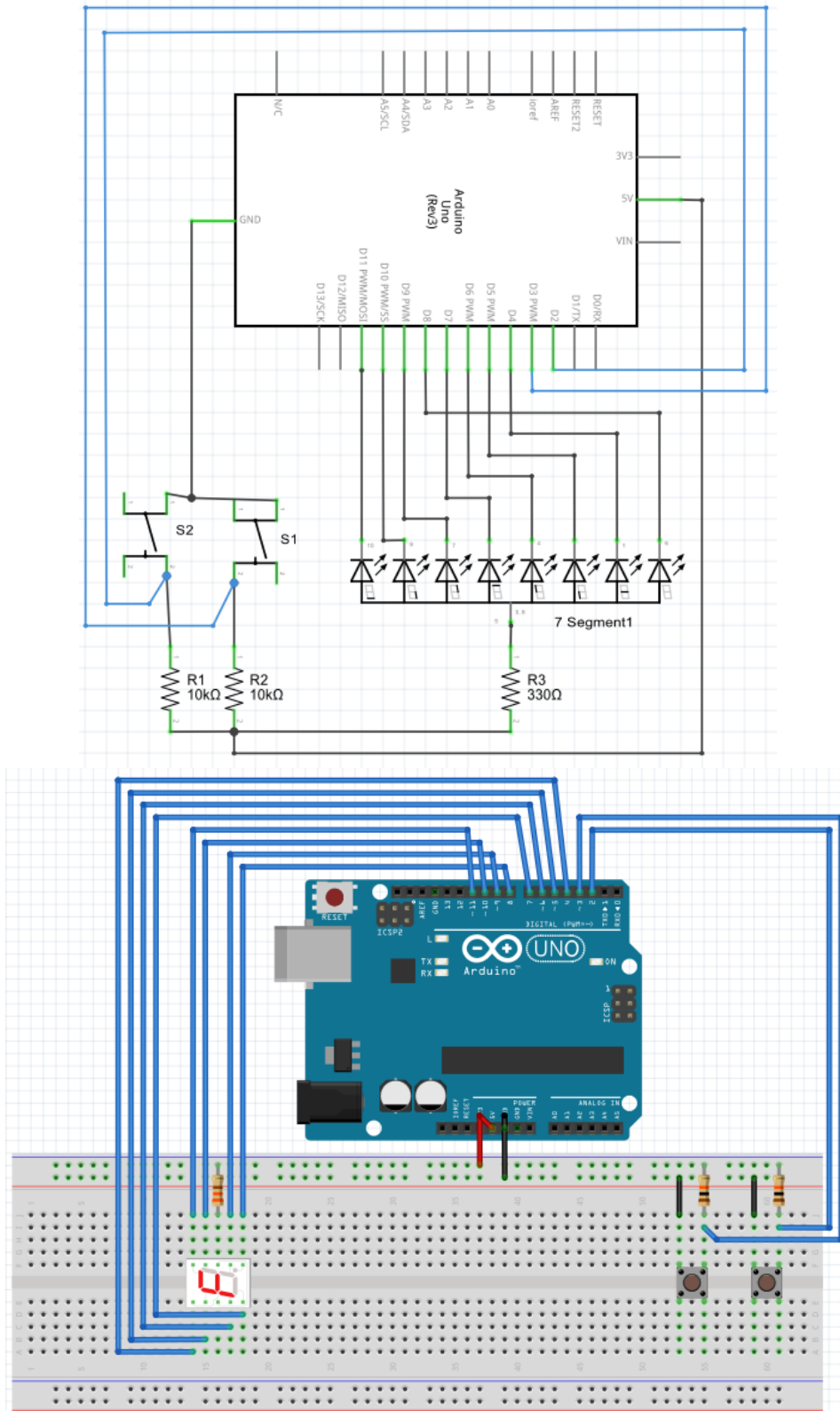


Figura 58, Esquema y prototipo de la Práctica 2.2.1.3. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código y completad las funciones correspondientes a las puertas lógicas que faltan por implementar. Si es necesaria más información visitad: <http://arduino.cc/en/Reference/HomePage>.

```

/* ----- */
/* Puertas lógicas. */

int pinsDisplay[]={4, 5, 6, 7, 8, 9, 10};
/* Pins del display en el orden: e, d, c, b, a, f y g. */

boolean cero[]={0, 0, 0, 0, 0, 0, 1};
boolean uno[]={1, 1, 0, 0, 1, 1, 1};
/* Representación de los números 0 y 1 en el display. */

int pinPulsador1 = 2, pinPulsador2 = 3;
/* Pins asociados a los pulsadores 1 y 2. */

int puerta = 49;
boolean mostrar = 1;
/* Inicia la variable "puerta" con el número 1 en código ASCII y crea la variable auxiliar
"mostrar" que será utilizada para mostrar
únicamente una vez por el monitor serial cada interacción con el mismo. */

void setup(){

  for(int i=0;i<=6;i++){
    pinMode(pinsDisplay[i], OUTPUT);
  }

  pinMode(pinPulsador1, INPUT);
  pinMode(pinPulsador2, INPUT);
  /* Establece los pins asociados al display como salidas y los asociados a los
pulsadores como entradas. */

  Serial.begin(9600);
  /* Inicia la comunicación con la placa. */
}

void loop(){

  boolean pulsador1 = digitalRead(pinPulsador1);
  boolean pulsador2 = digitalRead(pinPulsador2);
  /* Almacena en las variables booleanas (0,1) "pulsador1" y "pulsador2" las lecturas
de los pines asociados a los pulsadores. */

  if(Serial.available()){
    puerta = Serial.read();
    mostrar = 1;
  }
  /* Si existe información disponible desde el monitor serial guardarla en la variable
"puerta" y fijar la variable "mostrar" en 1. */
  switch (puerta){

```

```

case 49: { /* Puerta AND. */
  if(pulsador1 == 0 && pulsador2 == 0){
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], uno[i]);
    }
  }else{
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], cero[i]);
    }
  }
  break;
}
case 50: { /* Puerta OR. */
  /* ESCRIBE TU CÓDIGO. */
  break;
}
case 51: { /* Puerta NAND. */
  /* ESCRIBE TU CÓDIGO. */
  break;
}
case 52: { /* Puerta NOR. */
  /* ESCRIBE TU CÓDIGO. */
  break;
}
case 53: { /* Puerta XOR. */
  /* ESCRIBE TU CÓDIGO. */
  break;
}
default: { /* Cualquier otro valor */
  if(pulsador1 == 0 && pulsador2 == 0){
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], uno[i]);
    }
  }else{
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], cero[i]);
    }
  }
  break;
}
/* Según el valor de la variable "puerta" se ejecuta el código para la correspondiente
puerta lógica. */
}
if (mostrar){
  Serial.println("1 - AND ; 2 - OR ; 3 - NAND ; 4 - NOR ; 5 - XOR ");
  Serial.println(String(puerta-48));
  mostrar = 0;
}
}

```

```

/* Si la variable "mostrar" no es igual a 0 escribe por el serial y fija la variable
"mostrar" a 0 para no volver a mostrar nada
hasta que ésta vuelva a ser distinta de 0. Es decir, hasta que se ingrese nueva
información por el monitor serial. */
}
/* ----- */

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad el funcionamiento para cada opción.

### Solución propuesta.

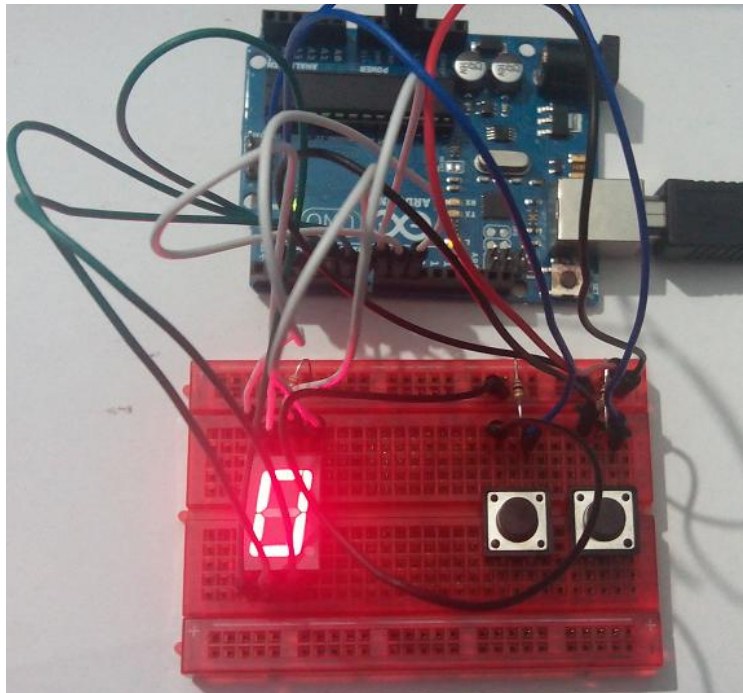


Figura 59, Solución propuesta de la Práctica 2.2.1.3. Fuente: Elaboración propia.

```

/* ----- */
/* Puertas lógicas. */

int pinsDisplay[]={4, 5, 6, 7, 8, 9, 10};
/* Pins del display en el orden: e, d, c, b, a, f y g. */

boolean cero[]={0, 0, 0, 0, 0, 0, 1};
boolean uno[]={1, 1, 0, 0, 1, 1, 1};
/* Representación de los números 0 y 1 en el display. */

int pinPulsador1 = 2, pinPulsador2 = 3;
/* Pins asociados a los pulsadores 1 y 2. */

int puerta = 49;
boolean mostrar = 1;
/* Inicia la variable "puerta" con el número 1 en código ASCII y crea la variable auxiliar
"mostrar" que será utilizada para mostrar
únicamente una vez por el monitor serial cada interacción con el mismo. */

```



```

void setup(){

  for(int i=0;i<=6;i++){
    pinMode(pinsDisplay[i], OUTPUT);
  }

  pinMode(pinPulsador1, INPUT);
  pinMode(pinPulsador2, INPUT);
  /* Establece los pines asociados al display como salidas y los asociados a los
  pulsadores como entradas. */

  Serial.begin(9600);
  /* Inicia la comunicación con la placa. */
}

void loop(){

  boolean pulsador1 = digitalRead(pinPulsador1);
  boolean pulsador2 = digitalRead(pinPulsador2);
  /* Almacena en las variables booleanas (0,1) "pulsador1" y "pulsador2" las lecturas
  de los pines asociados a los pulsadores. */

  if(Serial.available()){
    puerta = Serial.read();
    mostrar = 1;
  }
  /* Si existe información disponible desde el monitor serial guardarla en la variable
  "puerta" y fijar la variable "mostrar" en 1. */
  switch (puerta){
    case 49: { /* Puerta AND. */
      if(pulsador1 == 0 && pulsador2 == 0){
        for(int i=0;i<=6;i++){
          digitalWrite(pinsDisplay[i], uno[i]);
        }
      }else{
        for(int i=0;i<=6;i++){
          digitalWrite(pinsDisplay[i], cero[i]);
        }
      }
      break;
    }
    case 50: { /* Puerta OR. */
      if(pulsador1 == 0 || pulsador2 == 0){
        for(int i=0;i<=6;i++){
          digitalWrite(pinsDisplay[i], uno[i]);
        }
      }else{
        for(int i=0;i<=6;i++){
          digitalWrite(pinsDisplay[i], cero[i]);
        }
      }
      break;
    }
    case 51: { /* Puerta NAND. */

```

```

if(pulsador1 == 0 && pulsador2 == 0){
  for(int i=0;i<=6;i++){
    digitalWrite(pinsDisplay[i], cero[i]);
  }
}else{
  for(int i=0;i<=6;i++){
    digitalWrite(pinsDisplay[i], uno[i]);
  }
}
break;
}
case 52: { /* Puerta NOR. */
  if(pulsador1 == 0 || pulsador2 == 0){
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], cero[i]);
    }
  }else{
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], uno[i]);
    }
  }
}
break;
}
case 53: { /* Puerta XOR. */
  if(pulsador1 == 0 ^ pulsador2 == 0){
    /* Esta es la forma más eficaz, pero la siguiente también es válida:
    if((pulsador1 == 0 && pulsador2 == 1) || (pulsador1 == 1 && pulsador2 == 0)){ */
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], uno[i]);
    }
  }else{
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], cero[i]);
    }
  }
}
break;
}
default: { /* Cualquier otro valor */
  if(pulsador1 == 0 && pulsador2 == 0){
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], uno[i]);
    }
  }else{
    for(int i=0;i<=6;i++){
      digitalWrite(pinsDisplay[i], cero[i]);
    }
  }
}
break;
}
/* Según el valor de la variable "puerta" se ejecuta el código para la
correspondiente puerta lógica. */
}
if (mostrar){
  Serial.println("1 - AND ; 2 - OR ; 3 - NAND ; 4 - NOR ; 5 - XOR ");
  Serial.println(String(puerta-48));
}

```

```

    mostrar = 0;
  }
  /* Si la variable "mostrar" no es igual a 0 escribe por el serial y fija la variable
  "mostrar" a 0 para no volver a mostrar nada
  hasta que ésta vuelva a ser distinta de 0. Es decir, hasta que se ingrese nueva
  información por el monitor serial. */
}
/* ----- */

```

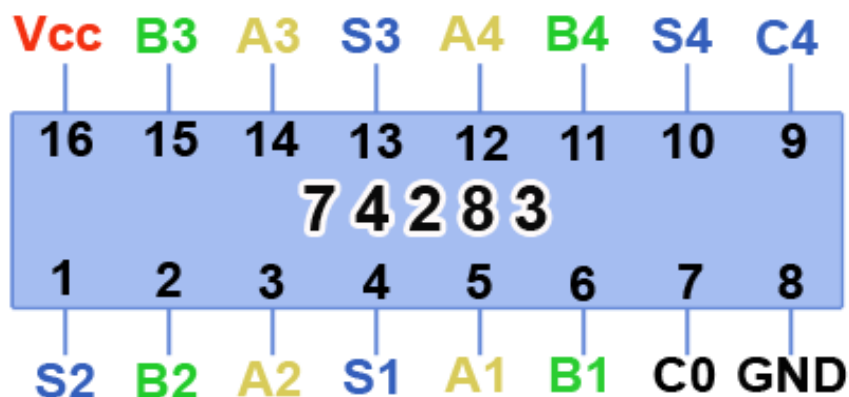
#### Práctica 2.2.1.4. Circuitos combinacionales: El sumador.

##### Enunciado.

En esta práctica comprobaremos el funcionamiento de un circuito combinacional sumador. Para ello utilizaremos un circuito integrado (CI) del tipo 74283, concretamente el DM74LS283. En la figura X podéis observar su representación y descripción.

Utilizando Arduino, se enviarán dos números de 4 bits cada uno: A(A1, A2, A3, A4) y B(B1, B2, B3, B4). El CI los sumará y enviará el resultado S(S1, S2, S3, S4) a Arduino además de permitir la iluminación de un LED rojo si se produce acarreo al final de la suma.

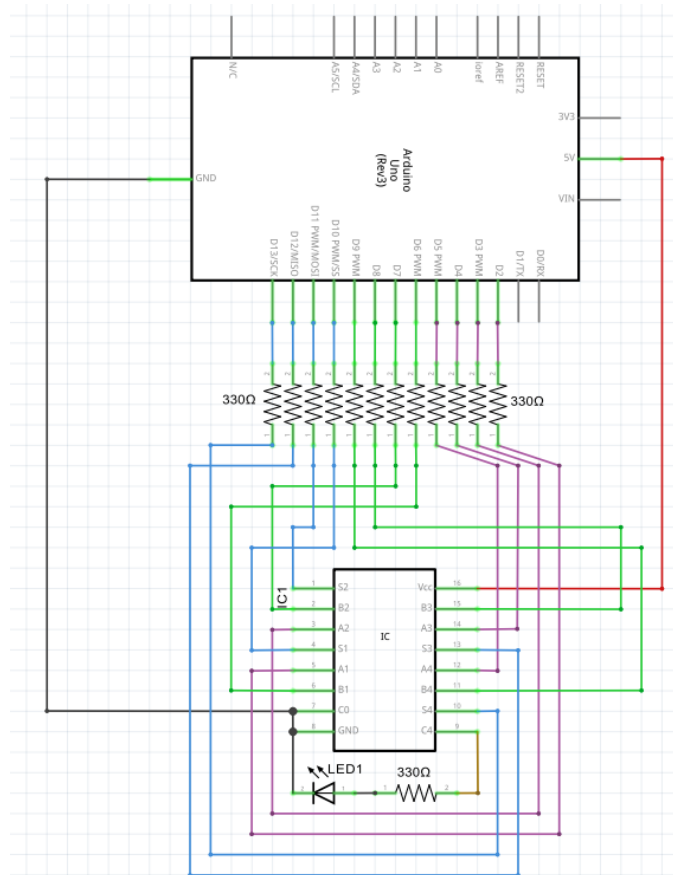
Por ejemplo, si queremos sumar el 3 y el 2 enviaremos al CI los números 0011 y 0010 y debemos obtener 0101 con el LED rojo apagado.



Pin	Símbolo	Descripción
1	S2	Bit 2 del resultado.
2	B2	Bit 2 del número B.
3	A2	Bit 2 del número A.
4	S1	Bit 1 del resultado.
5	A1	Bit 1 del número A.
6	B1	Bit 1 del número B.
7	C0	Acarreo inicial.
8	GND	Tierra o masa.
9	C4	Acarreo final.
10	S4	Bit 4 del resultado.
11	B4	Bit 4 del número B.
12	A4	Bit 4 del número A.
13	S3	Bit 3 del resultado.
14	A3	Bit 3 del número A.
15	B3	Bit 3 del número B.
16	Vcc	Positivo (5V).

Figura 60 Esquema y descripción del CI 74283. Fuente: Elaboración propia.

En primer lugar, construid el circuito en base al esquema y prototipo siguientes.



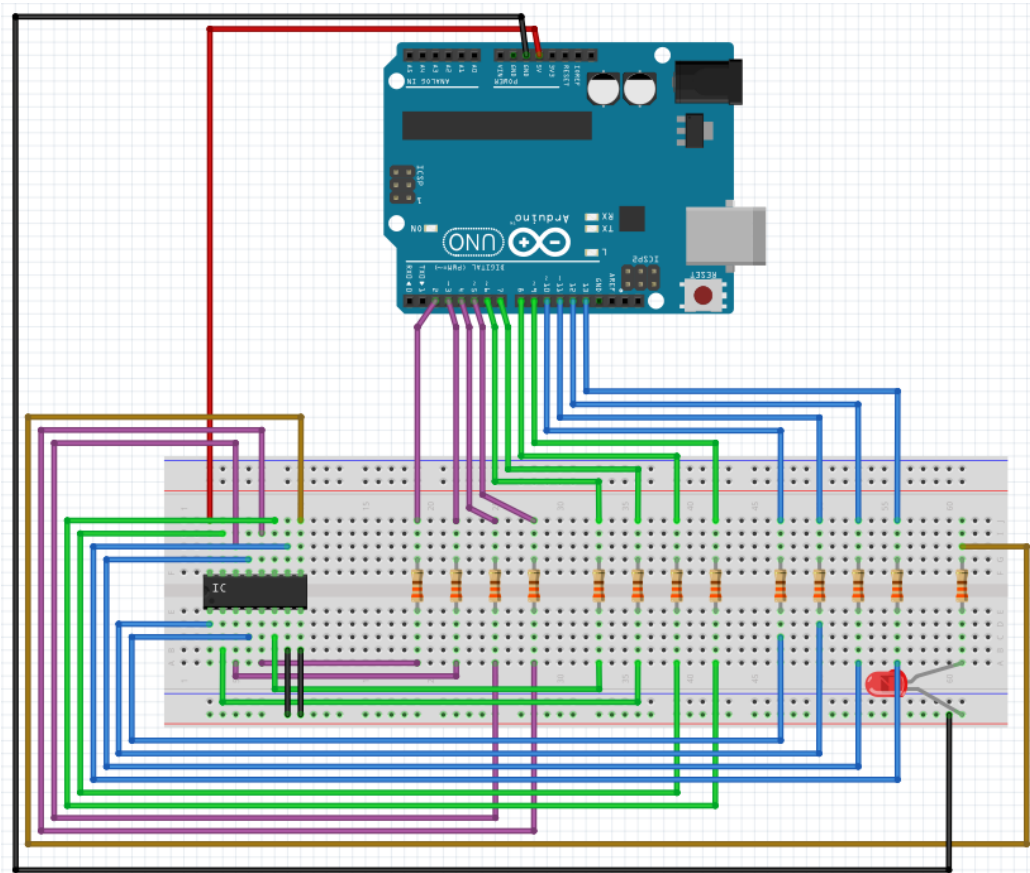


Figura 61, Esquema y prototipo de la Práctica 2.2.1.4. Fuente: Elaboración propia con Fritzing.

En segundo lugar, cread un nuevo programa copiad el siguiente código:

```

/* ----- */
/* Circuitos combinatoriales: El sumador. */

int pinsA[]={2, 3, 4, 5};
int pinsB[]={6, 7, 8, 9};
int pinsRes[]={10, 11, 12, 13};
/* Asignación de los pins para cada bit de los números a sumar y el resultado, en
variables de tipo array. */

void setup(){

  Serial.begin(9600);
  /* Inicia la comunicación. */
  for(int i=0;i<=3;i++){
    pinMode(pinsA[i], OUTPUT);
    pinMode(pinsB[i], OUTPUT);
    pinMode(pinsRes[i], INPUT);
  }
  /* Establece los pins de los números a sumar como salidas (entradas para el circuito
integrado) y del resultado como entradas
(salida del circuito integrado). */

```

```

/* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
boolean numeroA[] = {1,0,1,1};
boolean numeroB[] = {1,0,1,0};
boolean numeroRes[4];
/* "numeroA" y "numeroB" son los array que contienen los bits de los numeros a
sumar ("A" y "B").
Es aquí donde se deben manipular dichas array para obtener distintos resultados. */

for(int i=0;i<=1;i++){
  boolean aux=numeroA[3-i];
  numeroA[3-i]=numeroA[i];
  numeroA[i]=aux;

  aux=numeroB[3-i];
  numeroB[3-i]=numeroB[i];
  numeroB[i]=aux;
}
/* Hay que tener en cuenta que los números binarios introducidos en "numeroA" y
"numeroB" se encuentran en orden de array,
es decir, si numeroA[]={0,0,0,1} entonces el primer valor del array (numeroA[0]) es 0
y el último (nuemroA[3]) es 1. Así,
los números se encuentran en orden inverso a cómo los utilizamos de modo natural
para realizar las operaciones "a mano".
Para simplificar la tarea y eliminar la necesidad de escribir numeroA y numeroB al
revés cada vez que queramos cambiar sus
valores, esta función realiza el cambio de forma automática.*/

for(int i=0;i<=3;i++){
  digitalWrite(pinsA[i], numeroA[i]);
  digitalWrite(pinsB[i], numeroB[i]);
  delay(500);
  numeroRes[i]=digitalRead(pinsRes[i]);
}
/* Función para escribir en los pins correspondientes a cada número a sumar (A y B)
los respectivos bits y leer el resultado.
Por ejemplo, si el número A vale 0001, el número B vale 0010 y nos encontramos en
el instante i=0:
En pinsA[0] (pin 2) se escribe numeroA[0] (un 1); En pinsB[0] (pin 6) se escribe
numeroB[0] (0); En numeroRes[0] se escribe
el valor obtenido por el pin pinsRes[0] (pin 10), que ha de ser un "1" (0+1=1). */

Serial.println(String(numeroA[3])+" "+String(numeroA[2])+" "+String(numeroA[1])+"
"+String(numeroA[0]));
Serial.println(String(numeroB[3])+" "+String(numeroB[2])+" "+String(numeroB[1])+"
"+String(numeroB[0]));
Serial.println("-----");

```

```

Serial.println(String(numeroRes[3])+" "+String(numeroRes[2])+"
"+String(numeroRes[1])+" "+String(numeroRes[0]));
/* Muestra en el monitor serial los bits de los números A, B y resultado. */
}

void loop(){
}
/* ----- */

```

En tercer lugar, cargad el programa en la placa Arduino y comprobad el funcionamiento cambiando los valores de las variables tipo array “numeroA” y “numeroB”. Tened en cuenta que cuando se produzca acarreo final, en lugar de obtener un resultado de 5 bits, seguiremos obteniendo uno de 4 bits pero se encenderá el LED rojo de la protoboard.

### Solución propuesta.

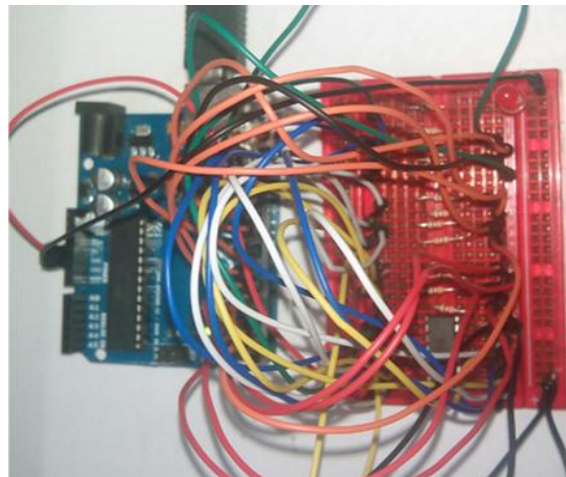
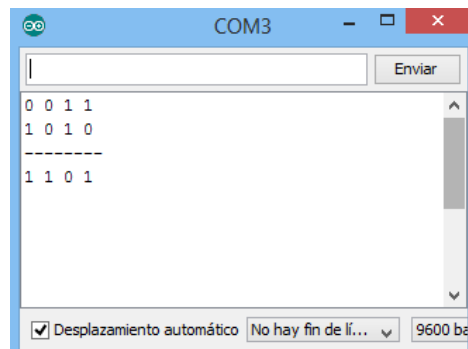


Figura 62, Solución propuesta Práctica 2.2.1.4. Fuente: Elaboración propia.

## Capítulo 6. Conclusiones y líneas de futuro.

### 6.1. Conclusiones.

Tras la finalización del proyecto “Tecnología en secundaria con Arduino” se han obtenido una serie de Baterías de actividades prácticas, que abordan los contenidos referentes a electrónica, robótica y sistemas de control especificados para Secundaria según los Decretos en materia de educación de la Comunidad Valenciana. Como herramienta para el desarrollo de los ejercicios, se ha utilizado la plataforma electrónica Arduino.

Por una parte, se ha dispuesto material didáctico innovador que facilita el objetivo actual de situar al alumnado en el centro del proceso de enseñanza aprendizaje y, por otra parte, se ha procurado contribuir a la divulgación de una plataforma electrónica que ofrece multitud de oportunidades en el aula. Además, se ha provisto una propuesta de solución para cada una de las prácticas, con el objetivo de facilitar la comprensión de la plataforma electrónica a aquellos docentes no iniciados.

Con todo, el trabajo pretende la inclusión en las aulas de Arduino, una plataforma basada en hardware libre y con un precio muy asequible, para permitir experimentar tanto al profesorado como al alumnado en el diseño de nuevos proyectos que fomenten la creatividad y el aprendizaje. Además, se promociona el uso de hardware y software libre en educación, desvinculando así a las marcas comerciales de la enseñanza obligatoria y post obligatoria preuniversitaria.

### 6.2. Líneas de futuro.

Un proyecto orientado a satisfacer necesidades educativas ha de someterse a constantes mejoras fruto de la investigación docente con el fin de aumentar su efectividad. Además, si su temática se encuentra relacionada con la tecnología ha de mantenerse actualizado para dar respuesta a los problemas utilizando métodos y herramientas vigentes. De lo anterior, se entiende la conveniencia de ampliar, en el corto o medio plazo, el alcance de este trabajo.

En principio, se considera que las líneas de mejora a implementar en el futuro próximo son tres: Ampliación del número de prácticas y ejemplos de proyectos libres junto a un aumento en la dificultad de los programas a realizar para controlar la placa Arduino;



Inclusión de un mayor número de sensores (Acelerómetros, giroscopios, setector de movimiento, etc.) y de módulos y *shields* de comunicación (NFC, XBee, Wifi, Ethernet, etc.) y adaptación de las prácticas al entorno de programación Scratch (S4A).

## Anexo I. Introducción a Arduino.

Arduino es una placa con microcontrolador *open source* que posee un conector USB para conectar con un ordenador y una serie de ranuras para interactuar con elementos electrónicos externos (sensores y actuadores). Puede ser alimentada mediante el USB o a través de una batería de hasta 9V y permite la ejecución de órdenes desde un ordenador o de modo independiente previa programación. Para ello, utiliza el Entorno de Desarrollo Integrado Arduino (Arduino IDE) disponible de manera gratuita.

Actualmente existen diversos modelos de placas Arduino orientados a situaciones específicas, en el caso de estas actividades el modelo seleccionado es Arduino UNO R3, una plataforma que cubre las necesidades derivadas de la resolución de los problemas planteados en las siguientes actividades. Dado su carácter *open source*, existe una comunidad (comunidad Arduino) de aficionados y profesionales que han dedicado sus esfuerzos a ampliar las posibilidades de esta herramienta mediante la creación de tutoriales, material didáctico y las denominadas shields.

Las *shields* son placas que se acoplan sobre Arduino a través de sus ranuras para ofrecer nuevas características específicas de manera muy sencilla. Así, existen *shields* para conectar Arduino a una red Ethernet, operar con varios motores eléctricos de modo independiente, acceder a distintos tipos de comunicaciones inalámbricas, disponer de pantallas TFT para interactuar con la placa y mucho más. A continuación se detallan las partes y características de la placa.

• Microcontrolador	ATmega328	• Memoria flash	32KB
• Voltaje de entrada (recomendado)	7-12V	• SRAM	2KB
• Voltaje de entrada (Límite)	6-20V	• EEPROM	1KB
• Voltaje operativo	5V	• Frecuencia de reloj	16MHz
• Pines digitales E/S	14 (6 PWM)	• Longitud	68,6mm
• Corriente en pines E/S	40mA	• Ancho	53,4mm
• Corriente en pines E/S (3,3V)	50mA	• Peso	25g
• Pines analógicos de entrada	6		

Tabla 53. Características Arduino UNO R3.

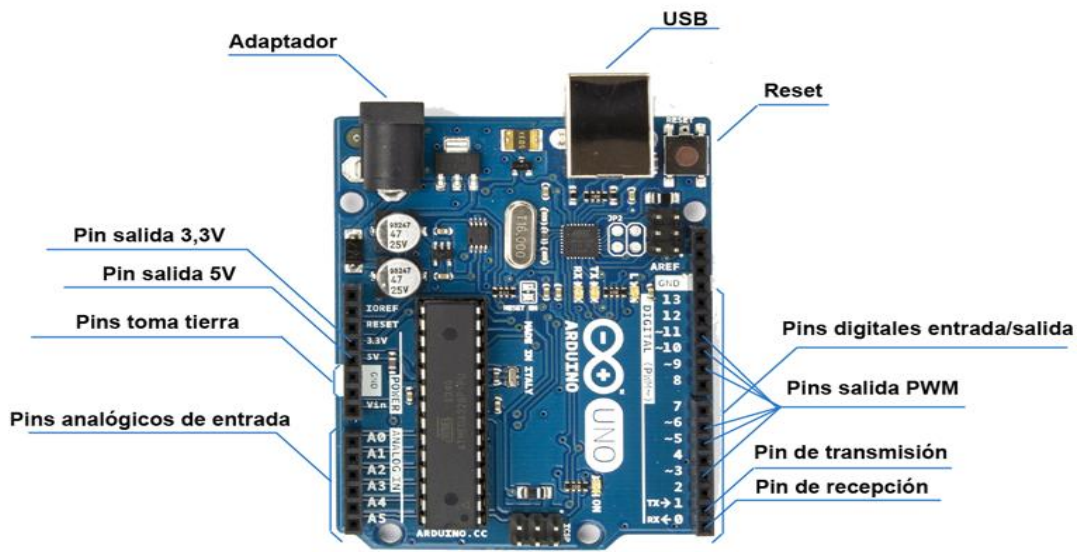


Figura 63, Esquema Arduino UNO R3. Fuente: Elaboración propia.

## Anexo II. Introducción al entorno de programación Arduino (Arduino IDE).

El entorno de programación de Arduino (IDE Arduino) es un software gratuito proporcionado directamente desde la página web de Arduino. A través de un entorno amigable, permite la programación de órdenes de ejecución para la placa así como la comunicación de datos con la misma. A continuación, se explica cómo descargar, instalar y probar el IDE Arduino para el sistema operativo Windows. La información ha sido extraída de <http://arduino.cc/>, donde puede encontrarse más información.

En primer lugar, accedemos a la página web: <http://arduino.cc/> y seleccionamos el apartado de descargas (download). Buscamos el instalador para Windows del IDE (Windows installer) y pinchamos para iniciar la descarga. En caso de no poseer los permisos necesarios para instalar programas en el ordenador, descargar la versión ZIP (Windows ZIP file) y descomprimir directamente, saltando el siguiente paso.

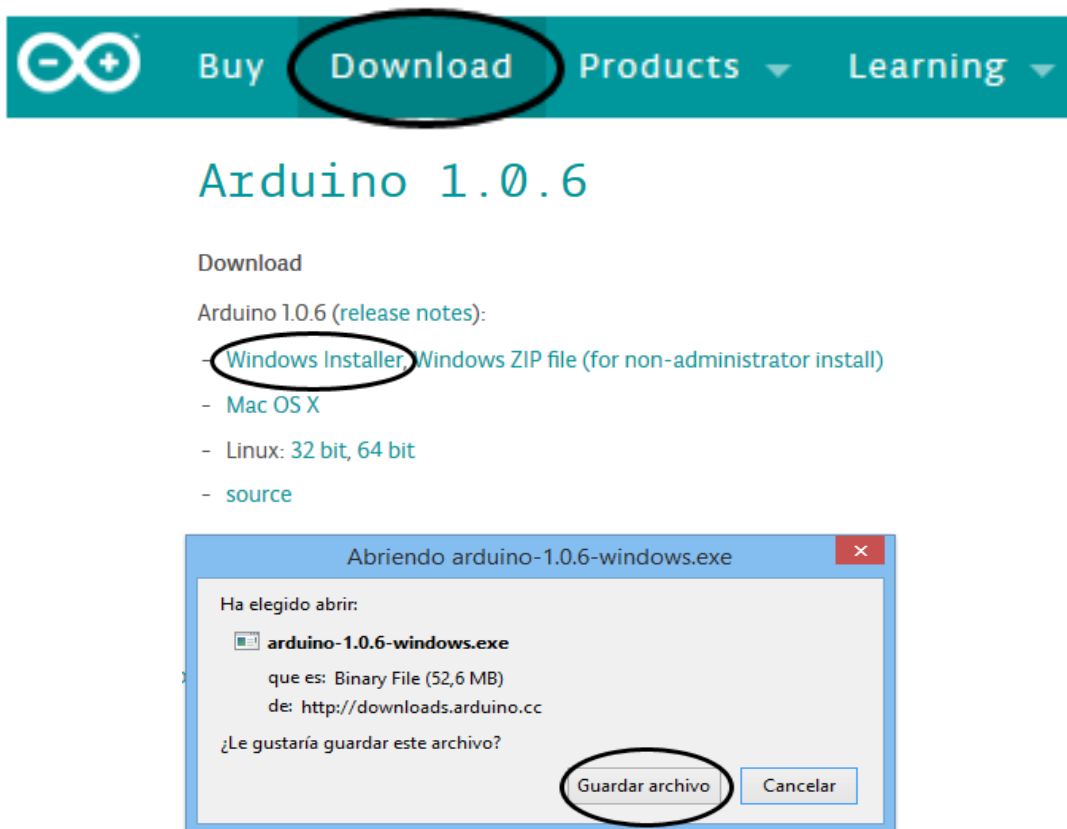
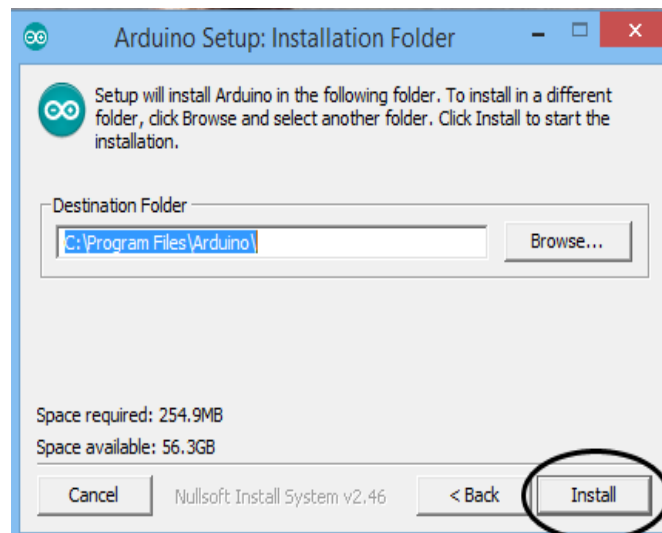
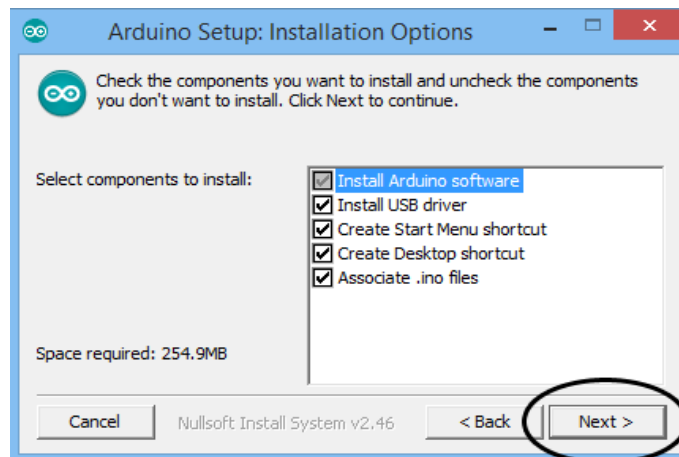
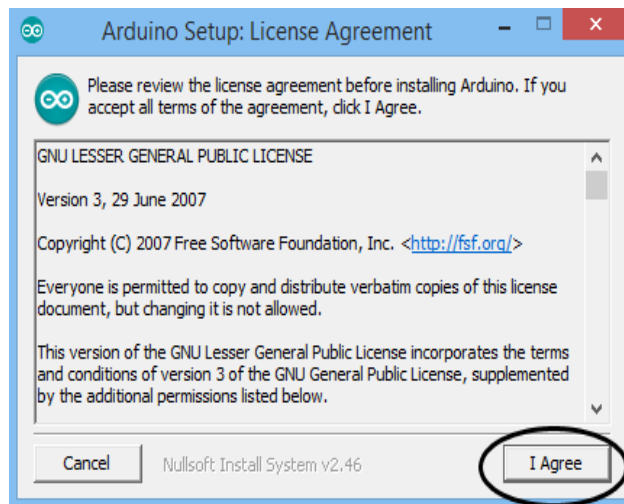


Figura 64, Descarga Arduino software. Fuente: Elaboración propia.

En segundo lugar, ejecutamos el instalador y seguimos las instrucciones hasta finalizar la instalación.



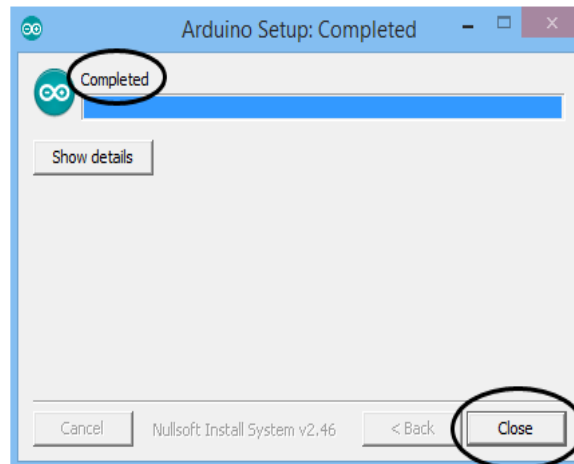


Figura 65, Instalación Arduino software. Fuente: Elaboración propia.

En tercer lugar, conectamos la placa Arduino al PC mediante USB y comprobamos que la luz verde etiquetada como PWR se enciende. También observaremos que se inicia de forma automática la instalación de drivers en Arduino UNO, esperamos a que finalice el proceso.

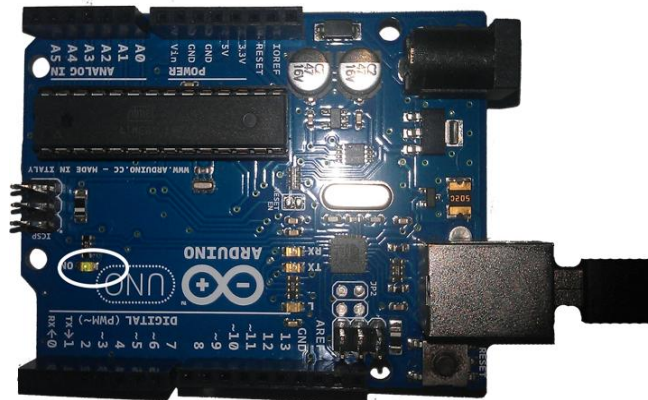


Figura 66, Comprobación encendido de Arduino. Fuente: Elaboración propia.

En cuarto lugar, ejecutamos la aplicación Arduino que ya instalamos en el paso 2.



Figura 67, Icono Arduino software. Fuente: Elaboración propia.

En quinto lugar, abrimos el ejemplo “Parpadeo de LED” (Blink) siguiendo la secuencia: Archivo, Ejemplos, 01.Basics, Blink.

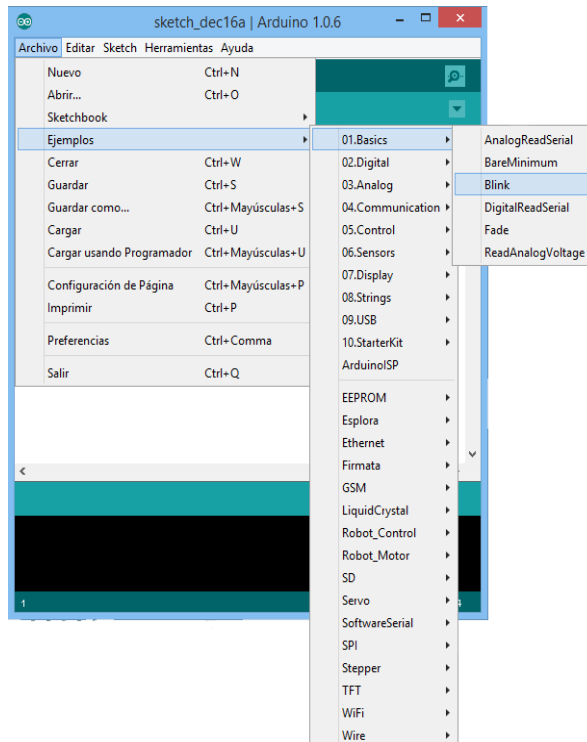


Figura 68, Ejemplo blink. Fuente: Elaboración propia.

En sexto lugar, seleccionamos qué tipo de placa utilizaremos en la actividad. Para ello, accedemos a herramientas, tarjeta y comprobamos que se encuentra activa la Arduino UNO. De lo contrario, la escogemos.

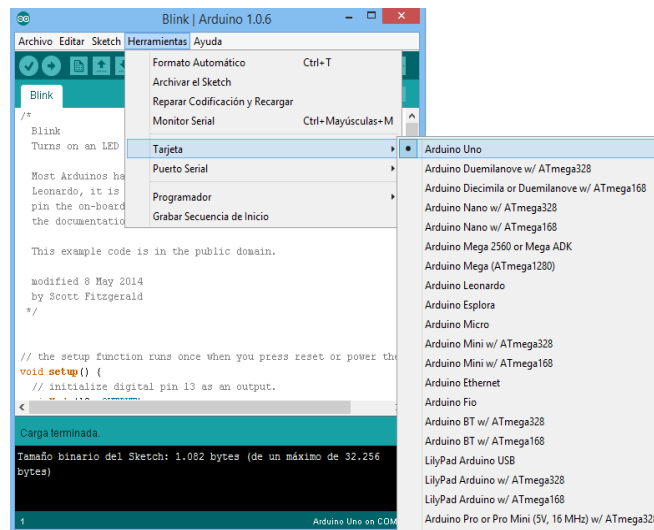


Figura 69, Selección de tarjeta. Fuente: Elaboración propia.

En séptimo lugar, mediante herramientas, puerto serial, indicamos el puerto de comunicaciones que utiliza Arduino UNO. Por defecto, suele ser el puerto COM3. Si no sabemos de qué puerto se trata, podemos desconectar la placa y volver a acceder al

menú. La opción que ha desaparecido es la que debemos seleccionar cuando volvamos a conectar la placa.

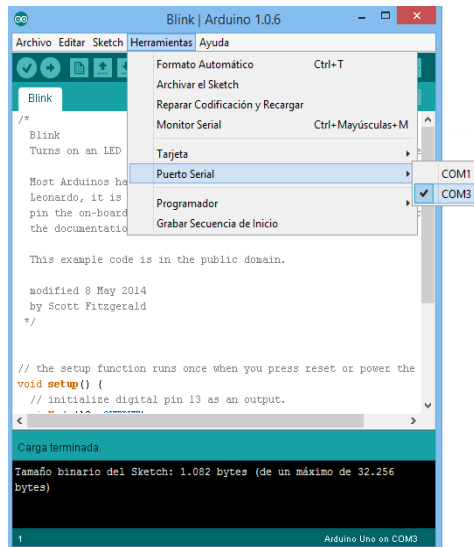


Figura 70, Selección de puerto de comunicaciones. Fuente: Elaboración propia.

Por último, cargamos el programa (ejemplo abierto) en la placa Arduino UNO. Basta con pulsar el botón “cargar” de la barra de herramientas y esperamos a que el entorno transfiera los datos.

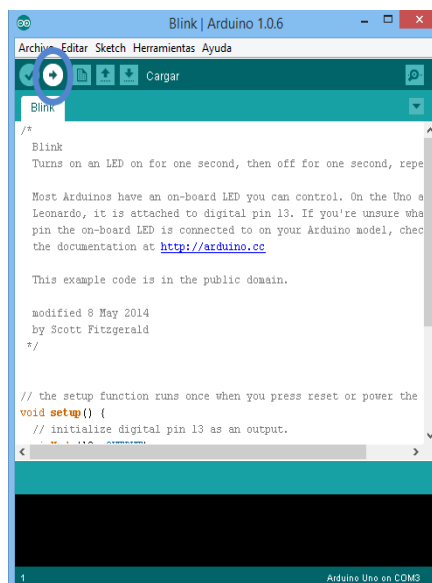


Figura 71, Cargar programa. Fuente: Elaboración propia.

Si todo ha funcionado correctamente, el LED integrado en Arduino UNO asociado al pin digital de entrada/salida número 13 parpadeará indefinidamente. Con todo, hemos descargado e instalado el IDE Arduino, abierto y cargado un programa y comprobado el correcto funcionamiento de la placa.



### Anexo III. La protoboard.

La placa de pruebas o *protoboard* es un tablero con orificios conectados internamente mediante material conductor de modo que conforman un conjunto de filas y columnas independientes a las que insertar elementos electrónicos. Su función es facilitar el montaje y desmontaje de circuitos electrónicos, eliminando la necesidad de construir circuitos impresos difícilmente reutilizables, para poder crear prototipos. En la figura 2 se muestra un ejemplo de uso.

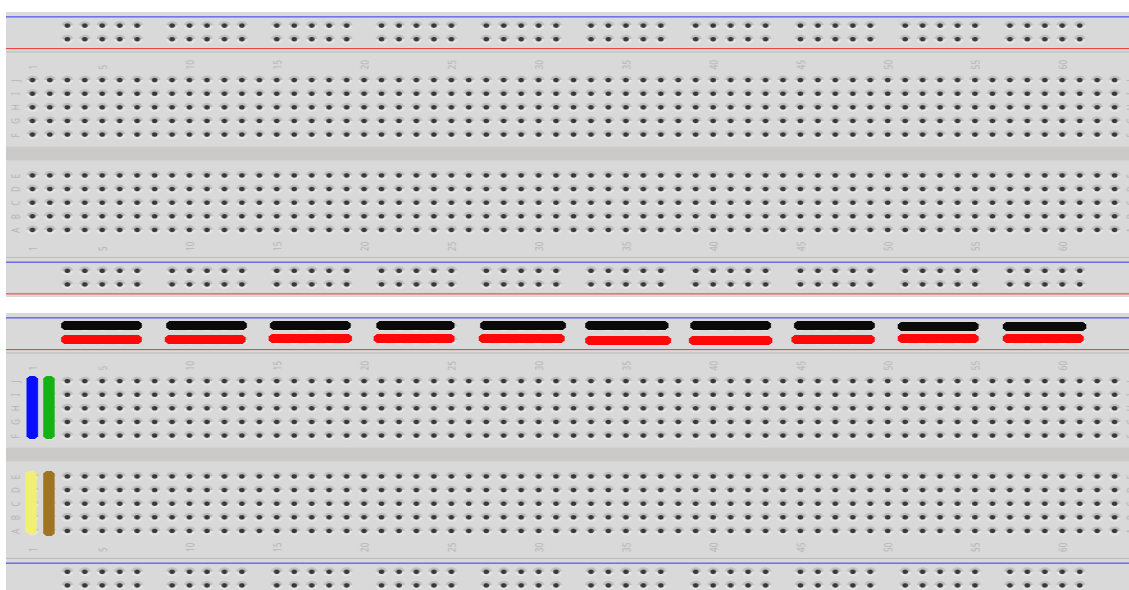


Figura 72, Protoboard y sus conexiones. Fuente: Elaboración propia partir de Fritzing.

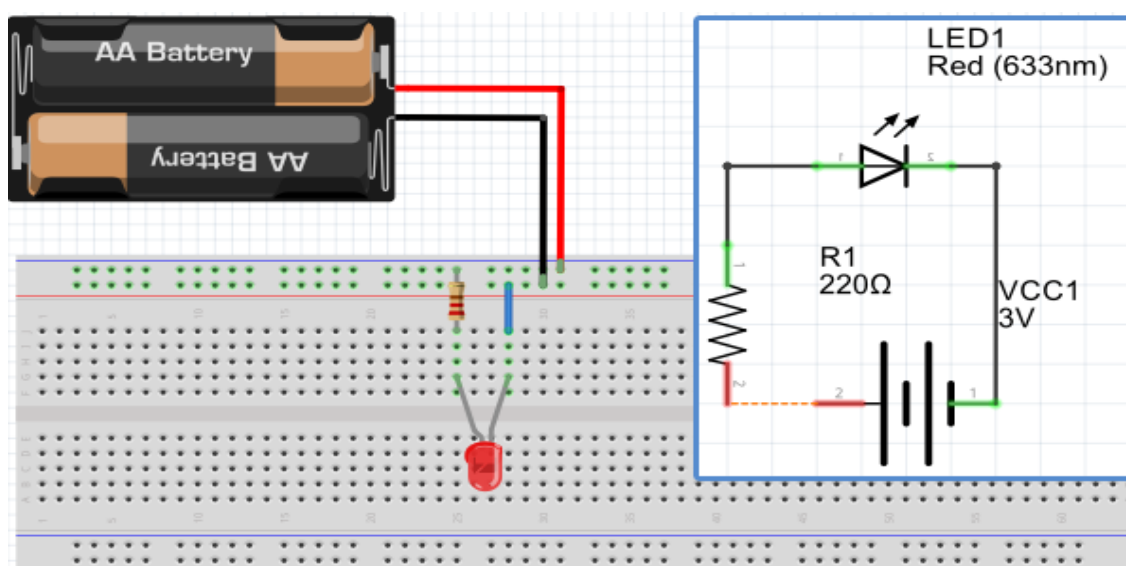


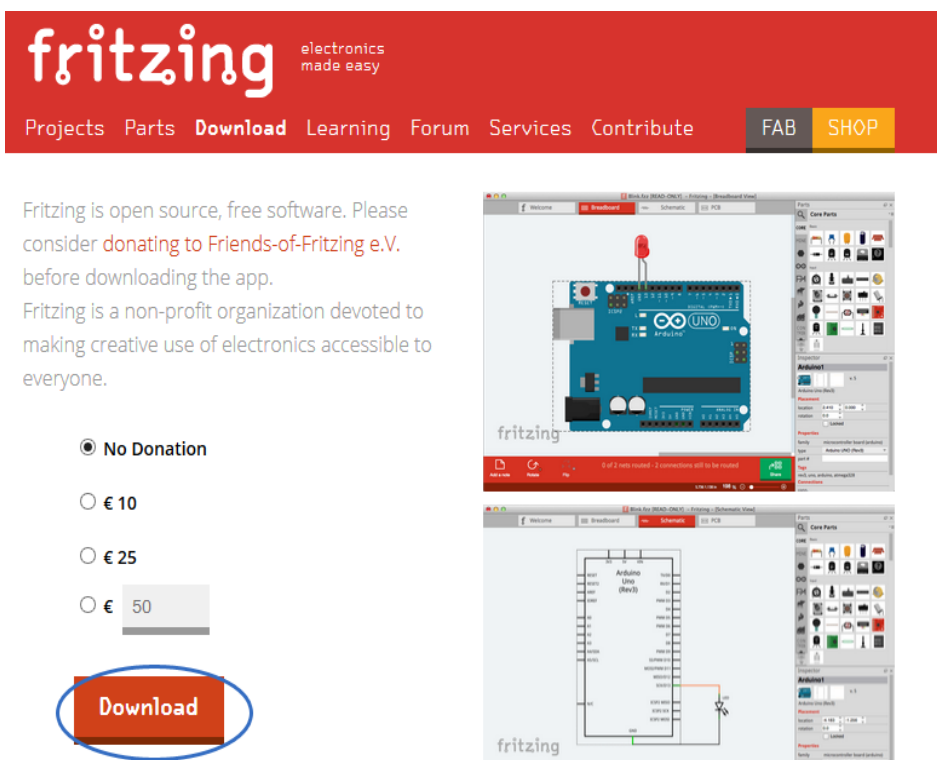
Figura 73, Ejemplo de uso de la protoboard. Fuente: Elaboración propia a partir de Fritzing.

## Anexo IV. Fritzing.

Fritzing es una iniciativa de hardware libre que persigue convertir la electrónica en un material creativo accesible para todo el mundo. Para ello, ofrece una herramienta software, una web de punto de encuentro comunitario y diversos servicios basados en el espíritu de Processing (lenguaje de programación) y Arduino. De este modo, la fundación Friends-of-Fritzing fomenta un ecosistema creativo que permite a los usuarios: Documentar prototipos, compartirlos, enseñar electrónica y diseñar PCB (tarjeta de circuito impreso) profesionales.

A continuación, se explica cómo descargar, instalar y ejecutar Fritzing para el sistema operativo Windows. La información ha sido extraída de <http://fritzing.org>, donde se puede encontrar más información.

En primer lugar, accedemos a la página web: <http://fritzing.org> y seleccionamos el apartado de descargas (*download*). En caso de escoger realizar una donación a la fundación Friends-of-Fritzing seleccionar el importe, de lo contrario, seleccionar *No donation* y pulsar sobre *download*. A continuación, seleccionamos el sistema operativo y aceptamos la descarga.



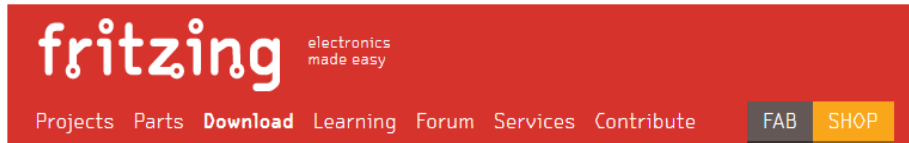
**fritzing** electronics made easy

Projects Parts **Download** Learning Forum Services Contribute FAB SHOP

Fritzing is open source, free software. Please consider [donating to Friends-of-Fritzing e.V.](#) before downloading the app. Fritzing is a non-profit organization devoted to making creative use of electronics accessible to everyone.

No Donation  
 € 10  
 € 25  
 € 50

**Download**



Fritzing is open source, free software. Be aware that the development of it depends on the [active support of the community](#). Select the download for your platform below.

Version **0.9.1b** was released on Dec. 2, 2014.

**Windows 32 bit**

Windows 64 bit

Mac OS X 10.6 and up

Linux 32 bit

Linux 64 bit

Source

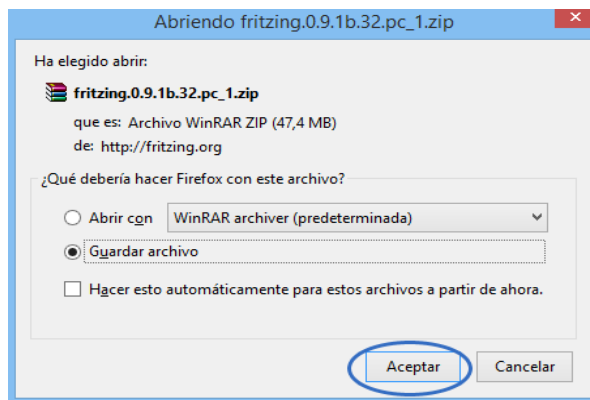
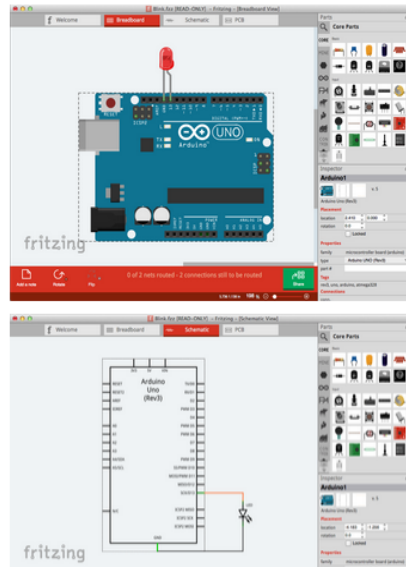


Figura 74, Descarga Fritzing. Fuente: Elaboración propia.

En segundo lugar, descomprimos el archivo descargado y ejecutamos la aplicación.

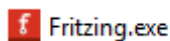


Figura 75, Ejecución Fritzing. Fuente: Elaboración propia.

Básicamente, el entorno se divide en 4 secciones: Menú de visionado, colección de elementos, opciones de elemento y zona de trabajo.

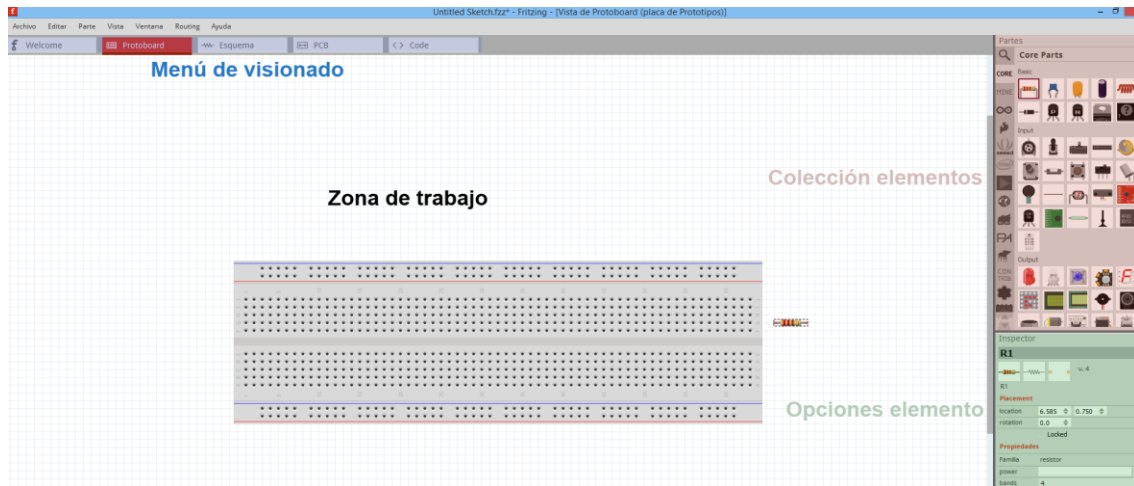


Figura 76, Entorno Fritzing. Fuente: Elaboración propia.

Desde el menú de visionado podemos seleccionar las distintas vistas del proyecto (Protoboard, esquema, PCB y code) así como la página de inicio. Por otro lado, la colección de elementos contiene las librerías (ampliables) de múltiples elementos electrónicos (desde resistencias hasta placas Arduino) que podemos introducir en nuestro proyecto tan solo arrastrando el icono correspondiente hasta la zona de trabajo. Además, podemos especificar diversas opciones del mismo desde el apartado opciones de elemento.

Para ampliar información acceder a la página <http://fritzing.org>.

**Bibliografía.**

Nicolau, F.; Cuenca, M.; Marco, M.; Pérez, A. (2010) Competencia comunicativa para profesionales de las TIC. Barcelona: FUOC.

Arduino. (2014). [en línea]. <http://www.arduino.cc> [fecha de consulta: 6 de Octubre de 2014].

BOE 106 (2006). Ley Orgánica 2/2006, de 3 de mayo de educación.

BOE 238 (2007). Real Decreto 1631/2006, de 29 de diciembre, por el cual se establecen las enseñanzas mínimas correspondientes a la educación secundaria obligatoria.

BOE 266 (2007). Real Decreto 1467/2007, de 2 de noviembre, por el que se establece la estructura del bachillerato y se fijan sus enseñanzas mínimas.

BOE 295 (2013). Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa. LOMCE.

Circuit Lab (2014) [en línea] <https://www.circuitlab.com/> [fecha de consulta: 3 de noviembre de 2014].

Code.org (2014) [en línea] <http://code.org/> [fecha de consulta: 2 de noviembre de 2014].

Crocodile Technology (1994) [en línea] <http://www.crocodile-clips.com> [fecha de consulta: 3 de noviembre de 2014].

DOCV 5562 (2007) DECRETO 112/2007, de 20 de julio, por el cual se establece el currículum de la educación secundaria obligatoria en la Comunidad Valenciana.

DOCV 5806 (2008) DECRETO 102/2008, de 11 de julio por el cual se establece el currículum del bachillerato en la Comunidad Valenciana.

El Mundo. La programación, el latín de los nuevos tiempos. (2014). [en línea]. [http://www.elmundo.es/madrid/2014/09/18/541ae28d268e3e93018b457e.html?a=l123e373053ad6ab17c5cead438deb3701d&t=1411111490&utm\\_content=buffer90ee1&utm\\_medium=social&utm\\_source=facebook.com&utm\\_campaign=buffer](http://www.elmundo.es/madrid/2014/09/18/541ae28d268e3e93018b457e.html?a=l123e373053ad6ab17c5cead438deb3701d&t=1411111490&utm_content=buffer90ee1&utm_medium=social&utm_source=facebook.com&utm_campaign=buffer) [fecha de consulta: 2 de Octubre de 2014].

El País: Madrid (2014) Qué van a aprender los alumnos de ESO con la nueva asignatura de programación. [en línea] [http://ccaa.elpais.com/ccaa/2014/09/04/madrid/1409854378\\_913511.html](http://ccaa.elpais.com/ccaa/2014/09/04/madrid/1409854378_913511.html) [fecha de consulta: 2 de noviembre de 2014]

Fischertechnik (2013) [en línea] <http://www.fischertechnik.de/home.aspx> [fecha de consulta: 1 de noviembre de 2014].

Fritzing (2014) [en línea] <http://fritzing.org/> [fecha de consulta: 1 de noviembre de 2014].

Gimp. (2001). [en línea]. <http://www.gimp.org/> [fecha de consulta: 6 de Octubre de 2014].

Lego Group (2013) [en línea] <http://shop.legoeducation.com/gb/> [fecha de consulta: 1 de noviembre de 2014].

Minirobots (2012) [en línea] <http://moway-robot.com/> [fecha de consulta: 1 de noviembre de 2014].

Open Office. (1999) [en línea]. <https://www.openoffice.org/es/> [fecha de consulta: 6 de Octubre de 2014].

Picaxe (2014) [en línea] <http://www.picaxe.com> [fecha de consulta: 3 de noviembre de 2014].

Presidencia de la Comunidad de Madrid: Madrid.org (2014) [en línea] [http://www.madrid.org/ignaciogonzalez/descargas/pdf/140915\\_ASIGNATURA\\_PROGRAMACION.pdf](http://www.madrid.org/ignaciogonzalez/descargas/pdf/140915_ASIGNATURA_PROGRAMACION.pdf) [fecha de consulta: 2 de noviembre de 2014].

Raspberry Pi Foundation (2014) [en línea] <http://www.raspberrypi.org/> 10 Junio 2014 [fecha de consulta: 1 de noviembre de 2014].

Scratch. (2003). [en línea] <http://scratch.mit.edu/about/> [fecha de consulta: 6 de Octubre de 2014].

Tecnocampus3D (2014) [en línea] <http://www.tecnocampus3d.com> [fecha de consulta: 2 de noviembre de 2014].

Wiring (2003) [en línea] <http://wiring.org.com> [fecha de consulta: 3 de noviembre de 2014].

3D Printer World (2014) [en línea] <http://www.3dprinterworld.com/article/britain-adds-3d-printing-new-curriculum> [fecha de consulta: 2 de noviembre de 2014].