

Un mundo automatizado

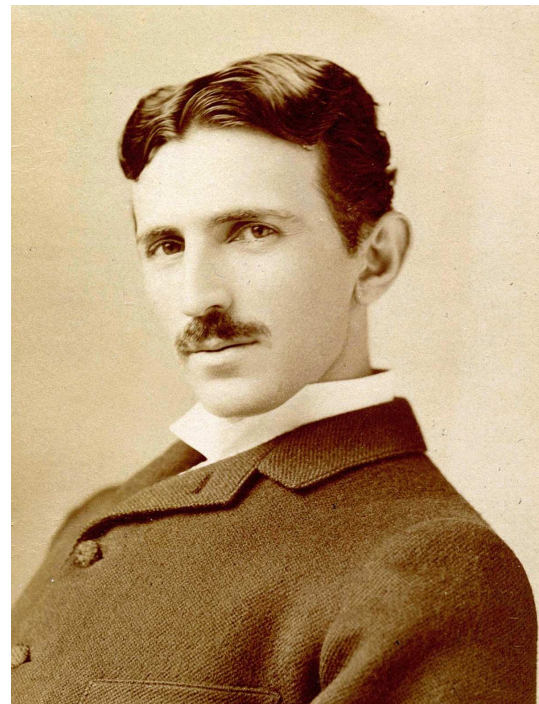
Un poco (poquito) de historia

El alumbrado público que se enciende al anochecer y se apaga al salir el sol, el semáforo del pueblo, el ascensor que se detiene exactamente en el piso que le indicamos, la planta que elabora y empaqueta las galletitas que compramos en el kiosco, un sistema de seguridad que al detectar humo toca una alarma, arroja agua y llama a los bomberos, el juego en el que tratamos de ganar un muñeco de peluche moviendo una grúa con una palanca... Todos los anteriores y muchos más, son ejemplos de dispositivos o mecanismos que utilizan algún tipo de **automatización**. Casi que no podemos imaginar alguna disciplina o actividad -tanto en el campo como en la ciudad- en la que no participen procesos automatizados. Prácticamente la totalidad de esos procesos automatizados requieren de energía eléctrica. Hacia ella nos dirigimos entonces.

Si bien el concepto de electricidad (y de su “primo mayor” el magnetismo) eran conocidos por la humanidad desde épocas muy tempranas, su generación recién se logró sobre el fin del siglo XVIII a través del italiano **Alessandro Volta**, quien creó la primera pila capaz de producir energía eléctrica, a través de una reacción química. A partir de allí varios científicos y físicos decimonónicos estudiaron y desarrollaron las aplicaciones prácticas de esa energía eléctrica.

Con la invención primero del telégrafo por **Samuel Morse** y **Alfred Vail** y la lámpara eléctrica por **Thomas A. Edison** se lograron las dos primeras implementaciones sustantivas de la energía eléctrica. La iluminación hogareña con la lámpara y la transmisión de datos a través de grandes distancias por medio del telégrafo empezaron a cambiar -literalmente- al mundo. Párrafo aparte

merece **Nikola Tesla** quien con sus estudios sobre la distribución de la energía a largas distancias y la implementación de la corriente alterna permitió que las ciudades enteras puedan tener suministro eléctrico en todos los hogares; y a través de la invención del motor eléctrico provocó lo que se conoce como **Segunda Revolución Industrial**.



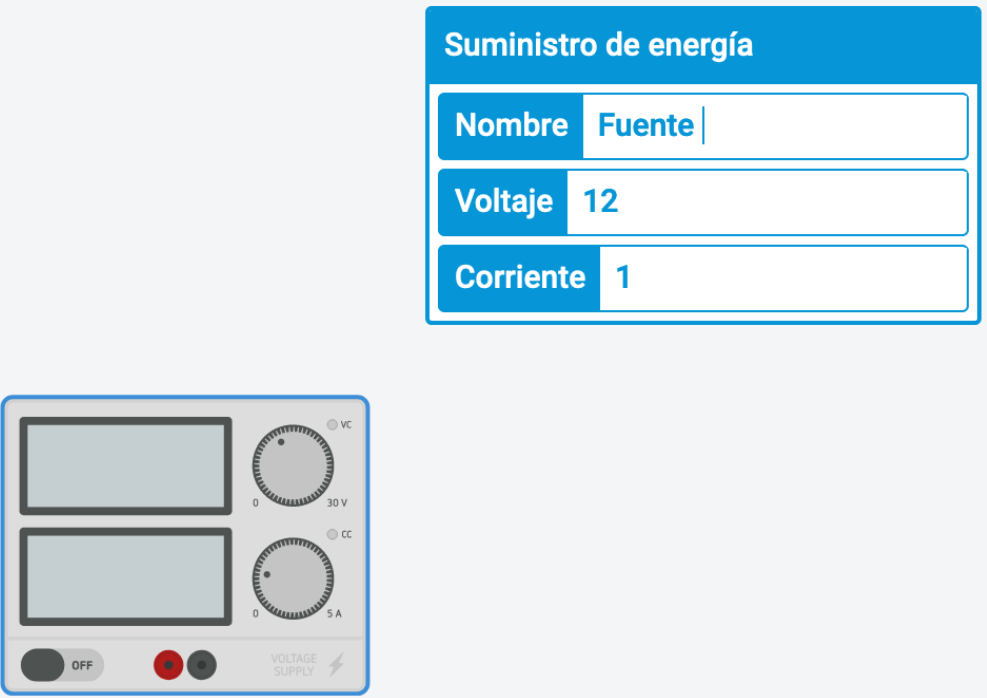
FICHA 1: Un mundo automatizado
Medimos magnitudes eléctricas.

Manos a la obra: Circuito simple

FICHA 1: Un mundo automatizado

Experimentamos con el circuito simple

Hagamos nuestro primer circuito eléctrico. Se llama **circuito simple** y consta de una **fente de energía** y un **elemento de consumo** (en este caso, una lámpara). Todo circuito eléctrico tiene que estar **cerrado**, es decir que para que la corriente eléctrica circule debe haber un camino sin cortes, cíclico. La **corriente eléctrica** sale de la fuente de energía, circula por los conductores (cables), pasa por el elemento de consumo y regresa a la fuente. Hagámoslo en el simulador [TinkerCAD](#). Primero desplegamos la pestaña de componentes y cambiamos de "Básicos" a "Todos" para que tengamos la totalidad de ellos a disposición. Busquemos "**Suministro de energía**" y lo arrastramos al lugar de trabajo. Podemos cambiarle allí sus propiedades. En este caso fijamos la **tensión** (voltaje que suministra) a 12 Volts (V). Imaginemos este componente como la fuente de nuestro celular, computadora o cualquier elemento que enchufamos a la línea de nuestra casa y que **transforma** los 220V **corriente alterna** que hay en el enchufe hogareño a (por ejemplo) 6, 9 o 12V **corriente continua** que necesita un dispositivo electrónico. Cambiamos entonces estas propiedades y le ponemos un nombre adecuado.

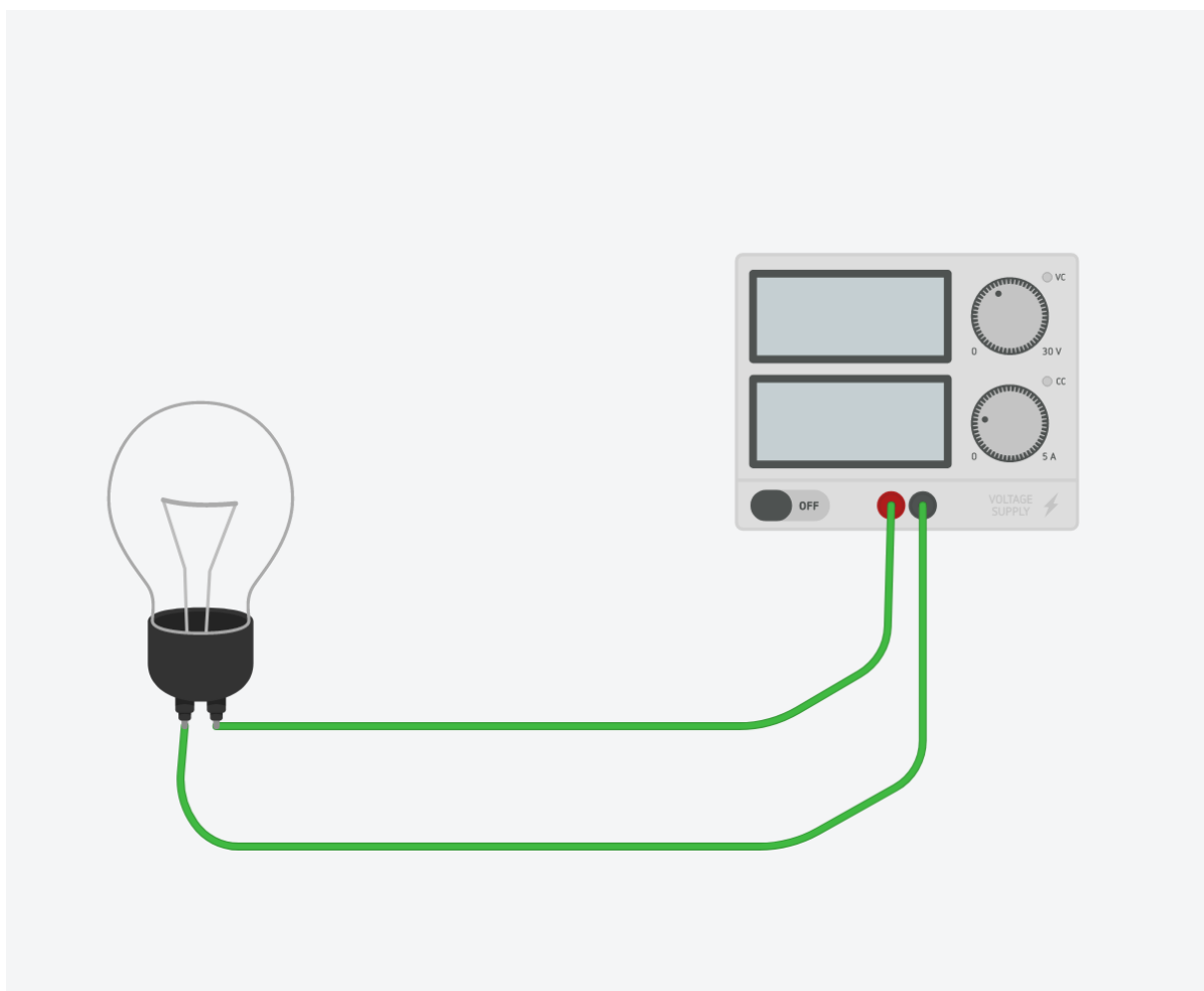


Suministro de energía	
Nombre	Fuente
Voltaje	12
Corriente	1

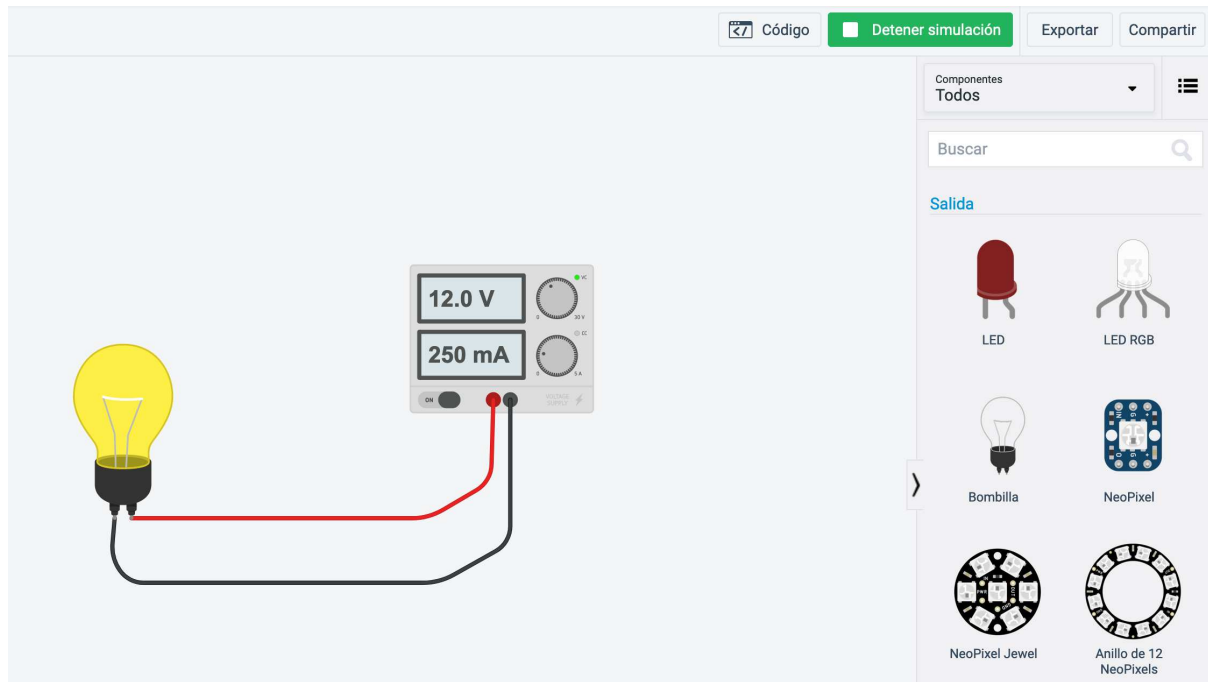
VOLTAGE SUPPLY

La fuente tiene dos conectores, uno rojo llamado **positivo (+)** y uno negro llamado **negativo (-)**. Estos colores están normalizados y son válidos en todo el mundo y para todos los dispositivos de corriente continua. En el enchufe hogareño no hay positivo ni negativo porque la corriente alterna responde a otra tipología. Por ahora utilizaremos solo **Corriente Continua (CC)** ya que para dispositivos pequeños y locales es perfecta, segura y económica. Pero para grandes distancias y grandes consumos se utiliza la **Corriente Alterna (CA)**. Por eso en nuestras casas tenemos CA y los dispositivos que enchufamos tienen que convertirla en CC. Más adelante utilizaremos corriente alterna para algunos de nuestros proyectos.

Ahora necesitamos una lámpara y cables. Los agregamos y nos aseguramos de que estén bien conectados. Los cables los hacemos con el mouse manteniendo apretado mientras dibujamos las conexiones.



¡Llegó el emocionante momento de probar el primer circuito! Le cambiamos el color a nuestros cables para mayor claridad y hacemos clic en el botón de arriba a la derecha que dice "Iniciar simulación"...



¡Nuestra lámpara se encendió! Veamos qué pasó allí:

Al darle clic al botón de inicio de simulación, fue como si enchufáramos la fuente a la pared. El visor indica 12V, que es la tensión que le habíamos configurado. La corriente DC **“sale” por el negativo** (cable negro) pasa por la lámpara, la enciende y regresa al origen por el cable rojo. Así lo hará eternamente, o hasta que se produzca alguna falla en el circuito como por ejemplo que se desconecte un cable o se queme la lámpara. También tenemos la opción de apagarlo desde la fuente porque provee un pequeño **interruptor** a la izquierda de los conectores. La fuente indica también una magnitud 250mA, quiere decir que la **corriente** que está circulando por el circuito en este momento es de 0,25 Amperes (A).

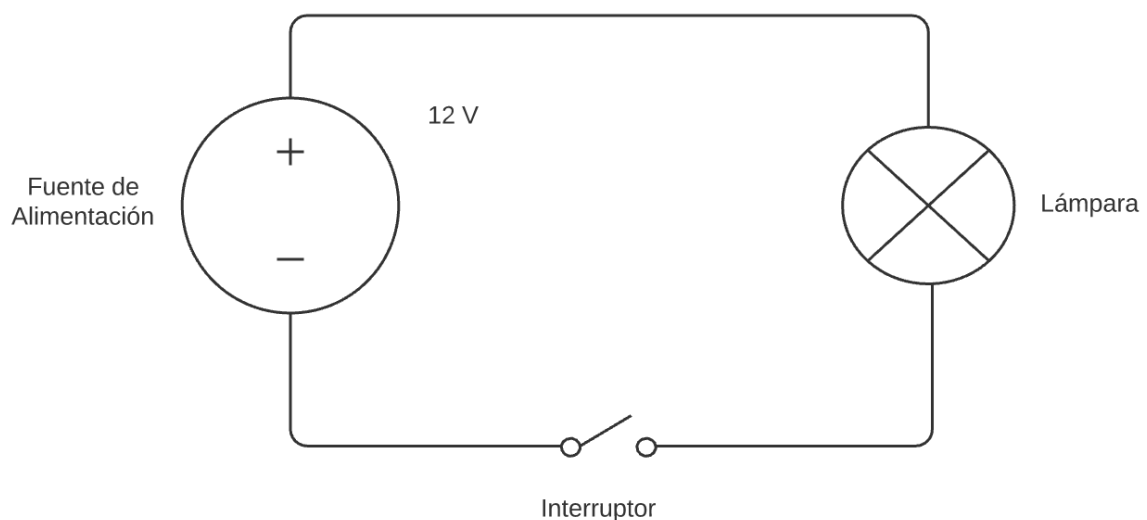
Observaciones:

1. Las lámparas y otros elementos de consumo están fabricados para trabajar de manera óptima solo en un **rango de tensión**. Debajo y sobre ese rango de tensión, funcionan mal o incluso se malogran de manera grave.
2. A igual carga (en nuestro caso es la lámpara) si bajamos la tensión también bajará proporcionalmente la corriente que circule por un circuito. Si subimos la tensión en un circuito, también sube la corriente. O sea que la tensión es directamente proporcional a la corriente.

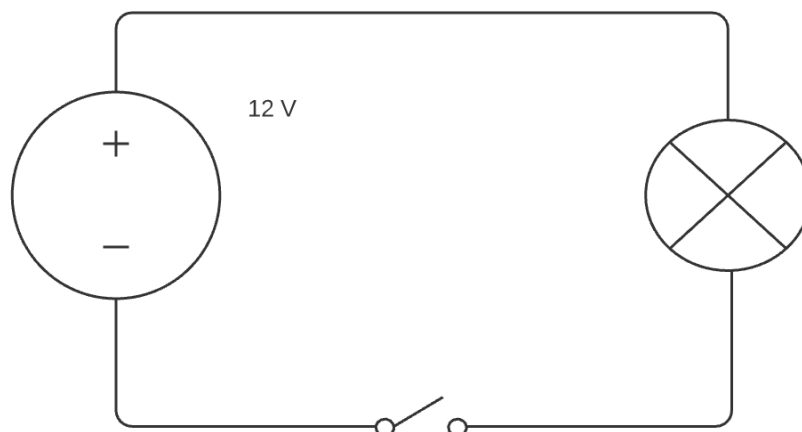
Simbología

FICHA 1: Un mundo automatizado Símbolos

En el mundo de las ciencias, siempre hay representaciones que nos permiten hacer más simples y entendibles las cuestiones reales. Sabemos que los elementos químicos son Au, Ag, K, etc. Los carteles de las rutas y las calles nos indican reglas de tránsito. Los colores del semáforo nos indican qué debemos hacer... En el mundo de la electricidad también existen símbolos y los vamos a usar mucho durante todo el curso. El **circuito simple** que acabamos de hacer e investigar se representa así:



En realidad las palabras "interruptor", "lámpara" y "fuente de alimentación" las agregamos porque es nuestro primer circuito, pero la idea de la simbología es que no haga falta poner las leyendas. O sea que nuestro circuito simple se simboliza así:



Solamente se aclaran las tensiones y conexiones que sean necesarias y críticas. En este caso se podría haber intercambiado los polos + y -, y la lámpara prendería igual, pero no siempre sucede así. En otros casos se podrían dañar los componentes si lo conectamos al revés, es por ello que siempre que trabajamos con **CC** se indica la polaridad.

Finalizando

Terminamos nuestra primera Unidad y recorrimos rápidamente el desarrollo de la investigación de la electricidad hasta llegar al circuito simple. Esta ficha es un poco larga porque aprendimos muchos conceptos que teníamos que saber antes de encarar la primera experiencia. Fijemos estos conceptos que vimos en este primer paso. ¡Nos vemos en la próxima Unidad!

Interruptores y pulsadores

Abriendo y cerrando los circuitos

En la unidad anterior para encender o apagar la lámpara usábamos la llave o interruptor de la propia fuente. Pero en nuestra vida podemos identificar multitud de **interruptores** diferentes según la función que cumplen. No es lo mismo el botón del ascensor que el pulsador del timbre, la llave de luz, el selector de velocidad del ventilador... pero todos ellos tienen la misma función: **abrir o cerrar un circuito eléctrico**. Veremos algunas maneras de conectar estos interruptores de manera de lograr diferentes resultados.

FICHA 2: Interruptores y pulsadores Interruptores

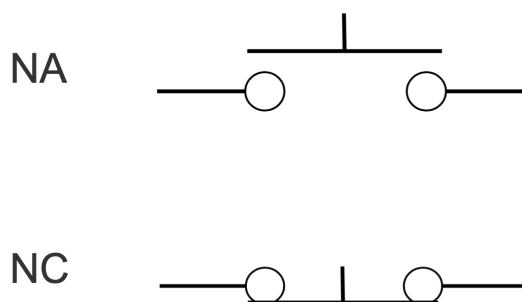
Pulsadores

En automatización generalmente se usan los **pulsadores**, que son interruptores comunes pero solamente se accionan cuando se mantienen presionados. Al soltarlos vuelven a su **estado inicial**. Los pulsadores e interruptores en general, se denominan a partir de "*qué hacen con el circuito cuando están en reposo*", es decir cuando no se los mantiene presionados. Los ejemplos más comunes son el timbre de las casas y el botón que tienen las heladeras para que al abrir la puerta se encienda la luz. Al primero se lo llama **Normalmente Abierto (NA)** y al segundo **Normalmente Cerrado (NC)**. Veamos el porqué de esta denominación:

- **Normalmente Abierto (NA):** El NA mantiene el circuito abierto hasta que se acciona, y al soltarlo vuelve a abrirlo. El timbre de una casa es el ejemplo clásico. Solo mientras se mantenga apretado, el timbre seguirá sonando.
- **Normalmente Cerrado (NC):** Es exactamente al revés que el anterior. Al estar en reposo el circuito está cerrado y al presionarlo lo abre. La luz de la heladera funciona de esa manera. Este interruptor generalmente está ubicado sobre el marco de la puerta de la heladera. Si lo presionamos con la mano, veremos que la luz de la misma, se apaga. Cuando la puerta se cierra presiona a ese interruptor y hará que la heladera no tenga la luz prendida mientras esté cerrada.

Acá tenemos los símbolos de los pulsadores. Identifiquemos los contactos y el botón según lo que acabamos de descubrir.

Pista: ¿Qué pasa cuando se va "para abajo" el pulsador?

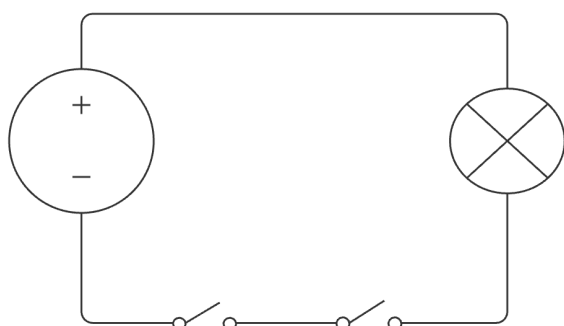


Combinamos interruptores

Hay diversas maneras de combinar los pulsadores / interruptores. Básicamente todas derivan de la combinación de tres maneras principales: AND, OR y NOT. Vamos a hacer cada una y las analizaremos.

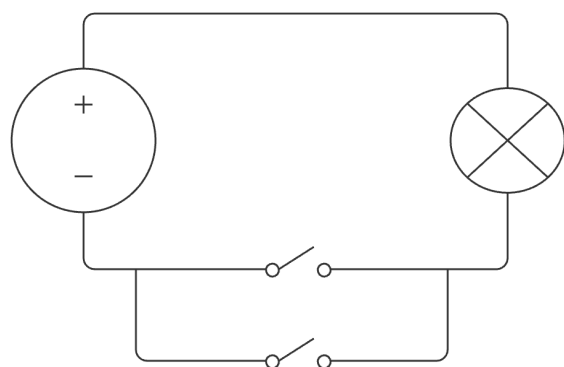
FICHA 2: Interruptores y pulsadores Interruptores combinados

Interruptores en serie (AND)



Para que se encienda la lámpara tienen que estar cerrados los dos interruptores. Se dice que están **en serie**. También, como tiene que estar accionado un interruptor “**Y además**” el otro, este circuito se denomina “**compuerta Y**” o “**compuerta AND**”

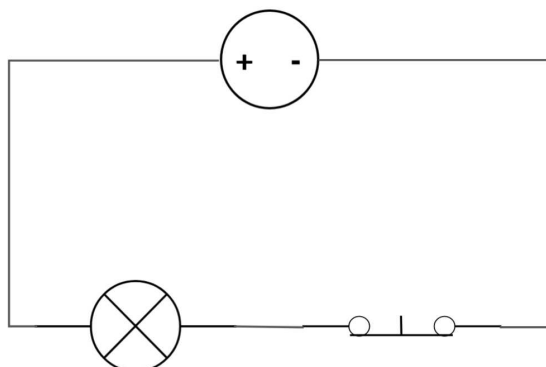
Interruptores en paralelo (OR)



Para que se encienda la lámpara es suficiente que esté cerrado uno de los dos interruptores. Se dice que están **en paralelo**. También, como tiene que estar accionado un interruptor “**O**” el otro, este circuito se denomina “**compuerta O**” o “**compuerta OR**”

Circuito Normalmente Cerrado (NOT)

Y para completar el trío fundamental falta la compuerta **NOT** (“**NO**” en inglés). Es como se conoce al circuito con un **NC** como muestra la figura:

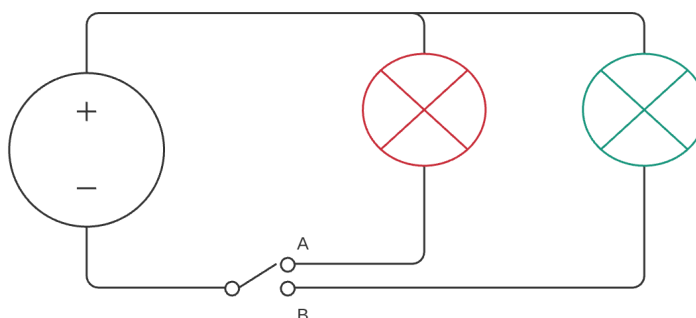


Todas las computadoras hacen sus cálculos y procesos combinando millones de estas tres maneras de conectar interruptores. Cuando circula corriente la computadora lo toma como un “1” y cuando no circula lo toma como un “0”. Esta es la base del **sistema binario** que les permite a las computadoras procesar datos, hacer operaciones matemáticas, etc. Cuando se programa una computadora lo que sucede internamente es que se combinan muchísimos interruptores **AND, OR y NOT** pero a través de un lenguaje de programación. Más adelante en este curso vamos a programar una placa **Arduino** para lograr automatismos complejos y construir con ellos experiencias super interesantes.

Llaves Selectoras

Existe un tipo especial de interruptores, que también debemos conocer ya que son muy usados: las **llaves selectoras**. No tienen dos estados (NC/NA) como las que acabamos de ver, sino que pueden seleccionar entre varias alternativas. Lo explicaremos mejor con un ejemplo:

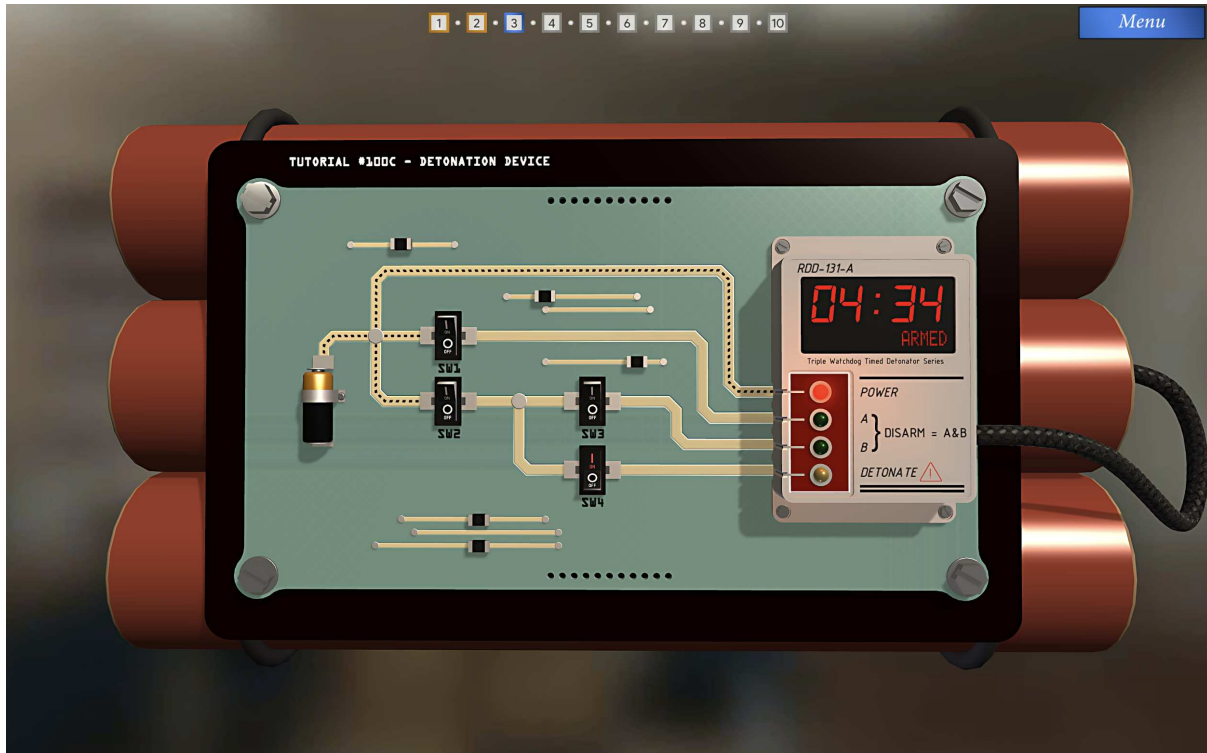
En la figura, si el interruptor está en la posición A se enciende la luz roja. Si está en la posición B se enciende la verde. Pero no es posible encender las dos lámparas al mismo tiempo.



Hay selectoras que tienen más posiciones, un ejemplo muy común es la llave que tienen las licuadoras o los ventiladores para ir cambiando de velocidad de giro de esos dispositivos.

Desactivando bombas

Para terminar con los contenidos de esta ficha, te proponemos un juego: **Bomb Squad Academy**, cuyos desafíos consisten en abrir o cerrar interruptores combinados de diversas maneras para desactivar una bomba antes de que se agote el tiempo. ¡Pero cuidado! Un error en alguno de esos pasos puede hacer que la bomba explote inmediatamente.



Trabajaremos con la versión Demo, disponible en:

<https://systemic-games.itch.io/bomb-squad-academy> y en la ficha encontraremos las instrucciones para su instalación.

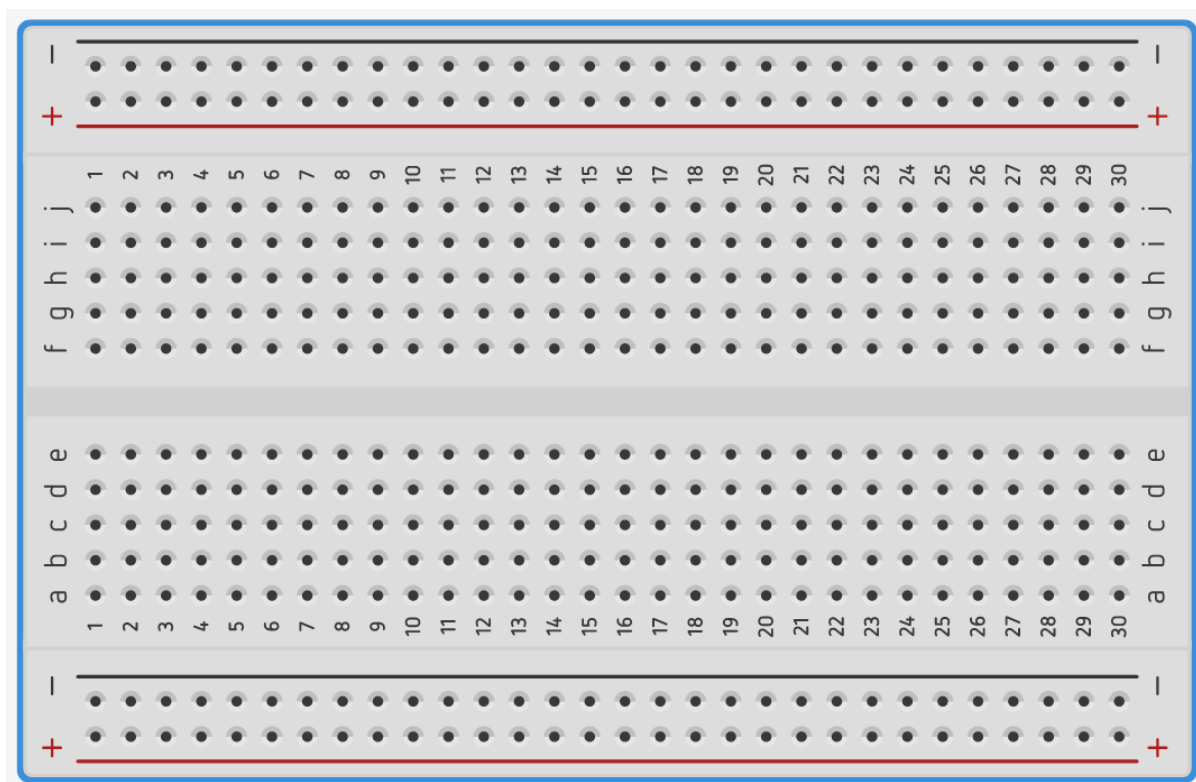
FICHA 2: Interruptores y pulsadores
Desactivando bombas

Circuitos de comando y de potencia

Conociendo el protoboard

Cuando hacemos experiencias con componentes eléctricos/electrónicos es una práctica muy incómoda, desprolija e insegura andar soldando acá y allá para hacer las conexiones pertinentes. Al rato queda una especie de araña alámbrica que no nos ayuda para nada en el aprendizaje, es compleja para buscar fallas y los componentes se van desgastando rápidamente de tanto soldar y desoldar.

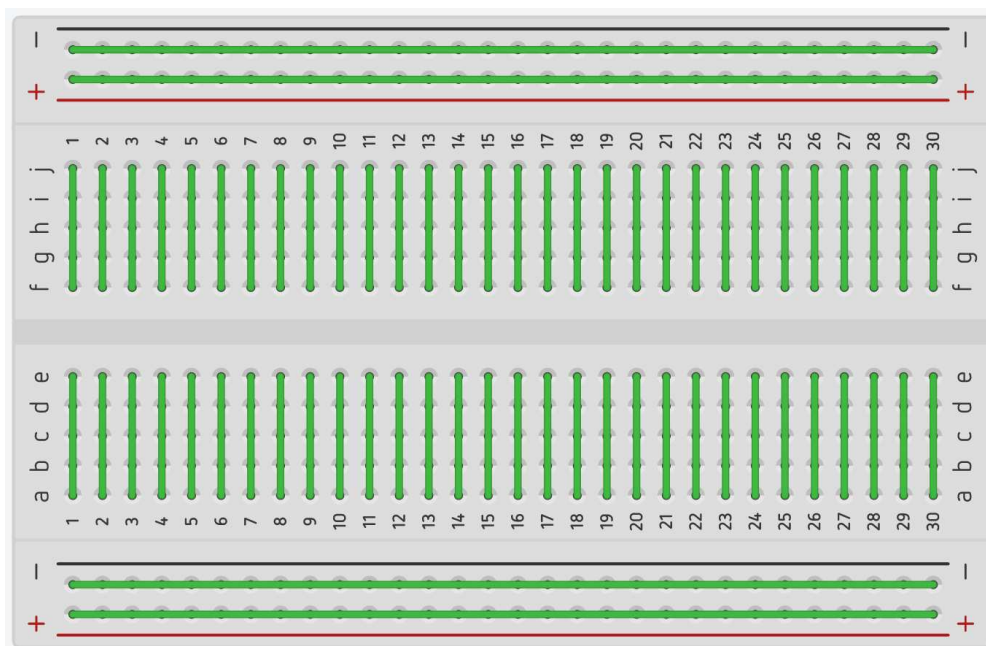
¡Es por eso que un alma benévola inventó el **protoboard**! Es una placa con muchas perforaciones para conectar a presión los cables y componentes, por lo que no hace falta soldar ni desoldar. Aparte ya vienen conectados entre sí de una manera que hace posible reproducir cualquier circuito de manera simple y clara. Es muy apropiada para hacer experimentos y pruebas. El protoboard luce de la siguiente manera:



Los números indican las "columnas" y las letras, las "filas". Las columnas están conectadas por dentro, uniendo los contactos internamente, separadas por el "pasillo" central. La imagen siguiente ayuda a comprenderlo, ya que es más simple verlo que contarlo. Las líneas verdes muestran las conexiones internas de la placa.

Las filas de los extremos tienen los símbolos + y - con los colores negro y rojo. Esas cuatro filas son especiales. Están conectadas como "filas" y no como "columnas" para facilitar la tarea de alimentar los circuitos. Simplemente se conecta la pila, batería o fuente de

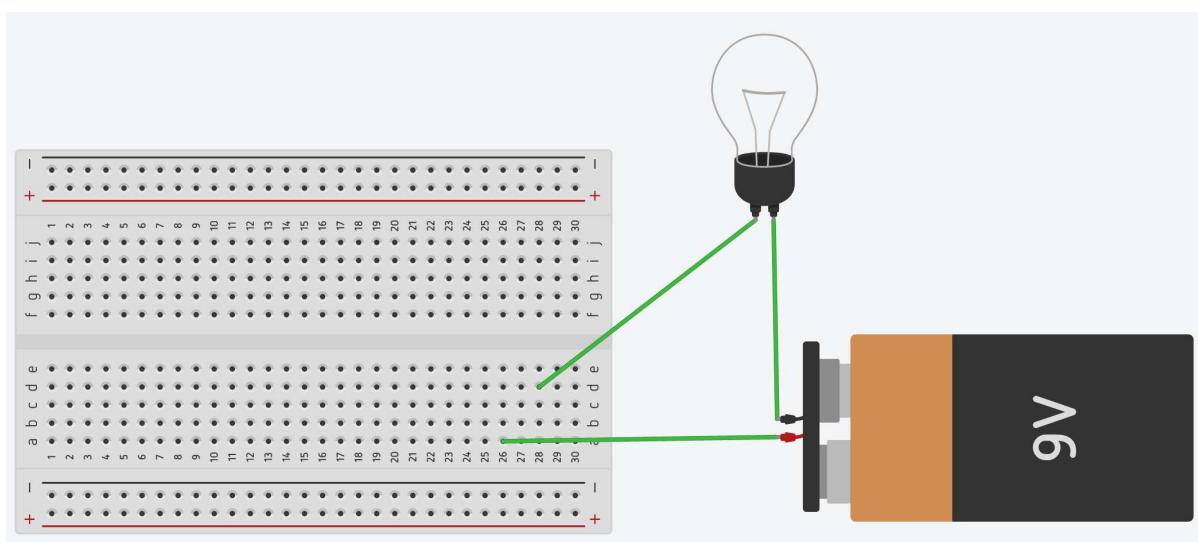
alimentación a ellas y luego, al necesitar un positivo o un negativo, se toma de cualquier lado de esa fila.



Para comenzar a tener gimnasia de uso del protoboard, hagamos nuestros primeros montajes de circuitos simples. Se sugiere su reproducción, con componentes o con el simulador Tinkercad, no solamente su estudio teórico.

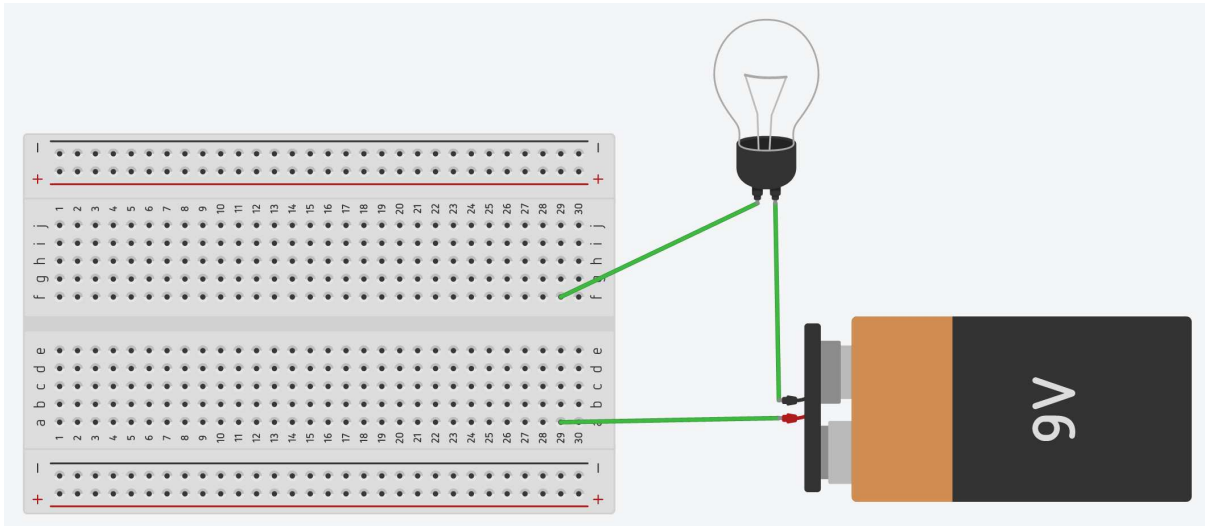
FICHA 3: Circuitos de comando y de potencia
¿Enciende o no enciende?

Ejemplo 1



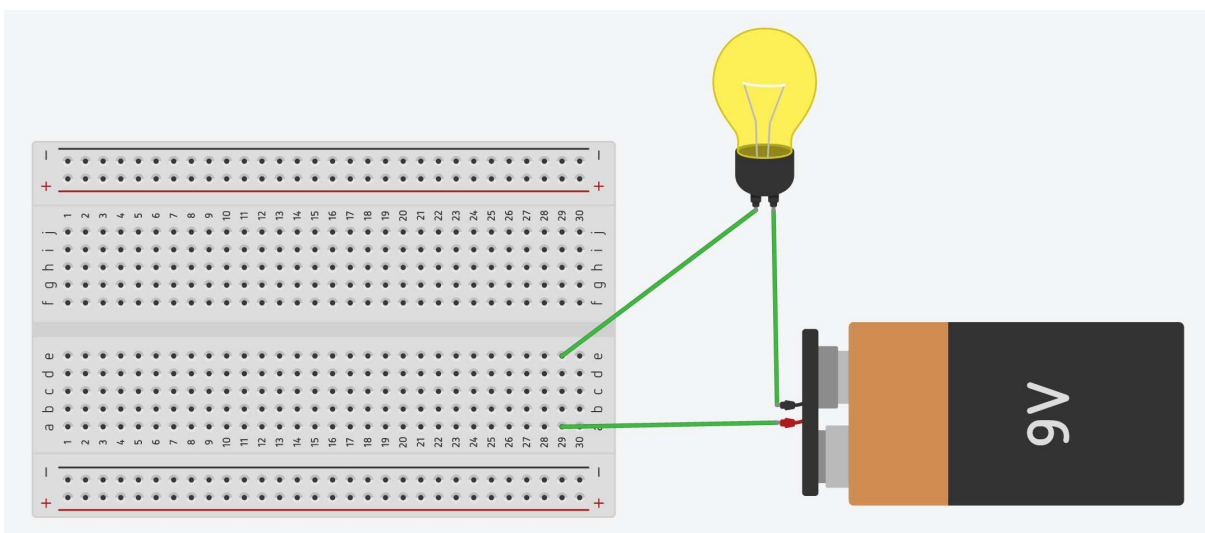
La lámpara no enciende porque está conectado en diferente línea. No cierra el circuito. Podemos solucionarlo colocando los dos cables en la misma línea.

Ejemplo 2



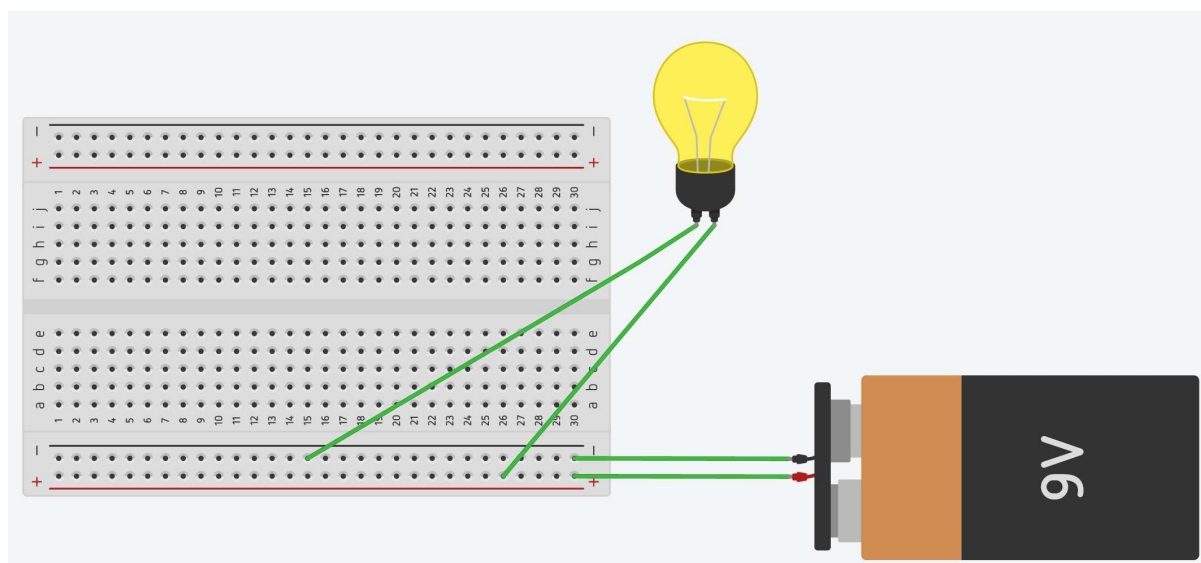
No enciende porque están conectados en la misma línea pero en diferentes sectores. Notemos que en el medio hay un pasillo separador. El circuito está abierto.

Ejemplo 3



Enciende porque está conectado en la misma línea y sector. El circuito está cerrado a través de las conexiones internas del Protoboard.

Ejemplo 4



Enciende porque usa los laterales + y - que son comunes a todo el lado del protoboard.

Un poco de (electro) magnetismo

FICHA 3: Circuitos de comando y de potencia Hacé tu propio electroimán

Desde que somos pequeñas y pequeños conocemos muchas experiencias simples que usan **imanes**. Son clásicos ejemplos el de tratar de unir dos imanes por el mismo polo y que estos insistan en repelerse; o el truco de mover una pieza metálica a través de la mesa cuando el secreto es dirigirla desde abajo con un imán. Ahora volveremos a los imanes pero no usaremos imanes naturales sino **electroimanes**.

El electroimán es básicamente un **arrollamiento de alambre** por el que se hace circular corriente eléctrica. Alrededor de esa bobina de alambre se va a generar un campo magnético cuya intensidad será proporcional a la corriente que circule y a la cantidad de vueltas que tenga ese bobinado. Generalmente se pone dentro de esa bobina de alambre un núcleo de hierro laminado que sirve para "orientar" a ese campo magnético en la dirección que lo necesitemos. Imaginemos una grúa en un depósito de chatarra: la grúa tiene en su extremo un electroimán en lugar de un gancho. Cuando se posa sobre un trozo de chatarra el operador activa el imán y puede levantarlo con facilidad. Al desactivar ese imán puede soltar la carga. Así de simple y así de útil.

El timbre de una casa, la barrera del tren, el motor eléctrico, esas puertas de los negocios en los que nos abren desde adentro por medio de un pulsador, las alarmas... infinidad de dispositivos eléctricos, electrónicos y electromecánicos funcionan gracias a los electroimanes. Vayamos a conocer juntos un tipo especial de implementación de los mismos: el **relé** o **relay**.

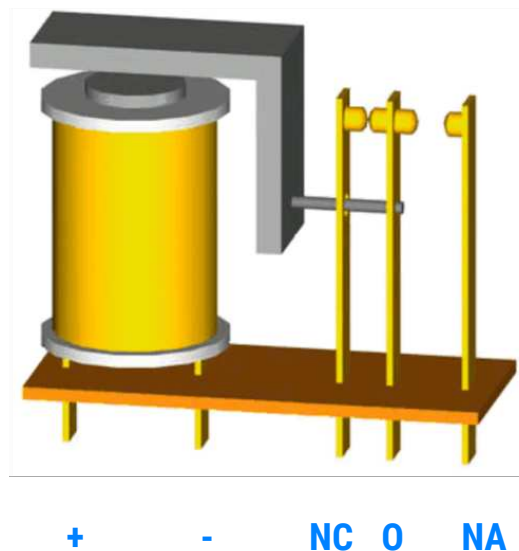
El relé o relay

FICHA 3: Circuitos de comando y de potencia Circuito de comando y de potencia

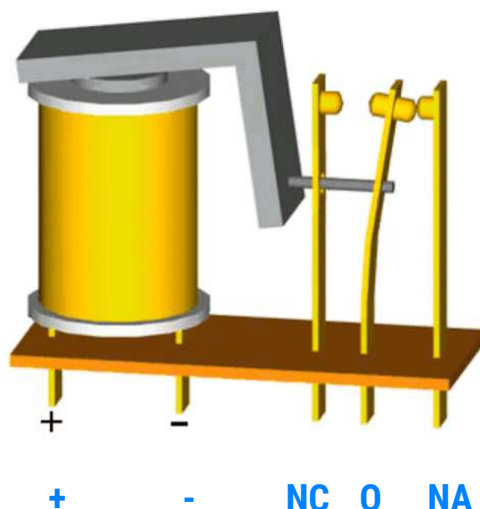
Uno de los dispositivos principales de la automatización. Tiene muchas variantes y medidas, pero el funcionamiento es prácticamente el mismo.

Básicamente un relé es un interruptor accionado por un electroimán.

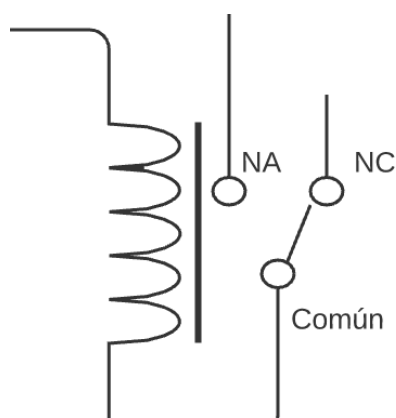
En la figura vemos un electroimán que se alimenta desde los terminales indicados con + y -. Hay una pieza metálica gris en forma de "L" y tres contactos **NA**, **O** y **NC**:



Vemos que al estar en reposo el contacto "O" (Común) está tocando con el NC. Es decir que hay conectividad entre ellos. Por eso se llama **contacto NC**, porque está "normalmente cerrado". Se toma la condición sin corriente circulando.



Al pasar corriente por los bornes + y - la cosa cambia: la bobina del electroimán atrae a la pieza en forma de "L" y el común ahora hace contacto con el **normalmente abierto** (NA) y deja de hacerlo con el NC. En símbolos se representa como muestra la figura siguiente:



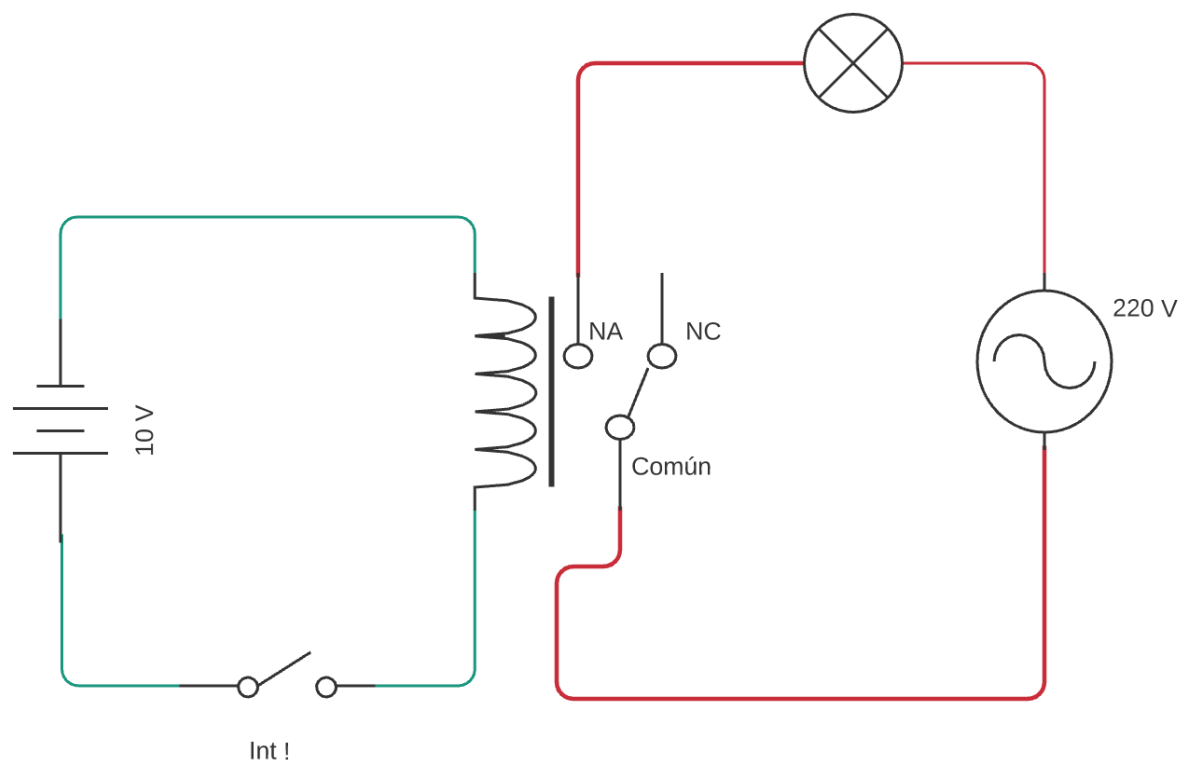
Podemos distinguir la bobina con sus terminales, el hierro de la misma (a veces se muestra sin él) y el contacto NA/NC. Cuando circula corriente por la bobina, el contacto móvil despega el NC y une el común con el NA. Cuando deja de circular, vuelve a la posición inicial. Recordemos, los símbolos siempre se dibujan en la posición "sin corriente circulando".

Animación del funcionamiento de un relé

https://es.wikipedia.org/wiki/Rel%C3%A9#/media/Archivo:Relay_principle_horizontal_new.gif

Ahora que sabemos cómo funciona un relé, veremos los usos más comunes del mismo.

Podemos separar un circuito de automatización en dos partes: **Comando y Potencia**. Esto nos brinda seguridad, claridad y prolijidad en nuestros montajes. Lo explicaremos con símbolos:



Para más claridad se han usado colores diferentes: la parte verde es simplemente una fuente de baja tensión (10 V, inofensiva) que, a través del **Int1**, alimenta a la bobina de un relé. A su vez los contactos del relé (ya en la parte roja) conectan o desconectan una lámpara a 220 V (peligrosa). Si se cierra el **Int1** el relé conectará el NA y la lámpara se encenderá. Así funciona, pero la pregunta es...

¿Si el circuito simple funciona perfectamente, por qué estamos usando un relé?

Entre otras cosas, principalmente por **seguridad**. La parte de **Potencia** (la roja) está físicamente separada de la de **Comando** (verde). Las personas que operen ese circuito van a trabajar siempre del lado verde del mismo, nunca van a operar sobre tensión peligrosa. Y cualquier problema que surgiera en el circuito de potencia será aislado por el relé que lo mantendrá desvinculado del de comando. Además es más fácil para **diseñar**, se puede **reutilizar** la parte lógica, queda físicamente separados para **encontrar fallas**... las ventajas son muchas!

En este curso vamos a trabajar prácticamente del lado verde, del lado de **comando** (la lógica que hace disparar o no a un relé). A ese relé luego puede ir conectada una lámpara como en la figura, o un motor, una máquina embotelladora, un artefacto, lo que fuere. Importa para nosotros la lógica que debemos aplicar desde el lado de comando para automatizar el circuito.

Sigamos analizando: si a la salida del relé conectamos otro relé en el NA y otro en el NC, y a la salida de esos, otros... podemos hacer una cascada lógica que activará una cantidad de

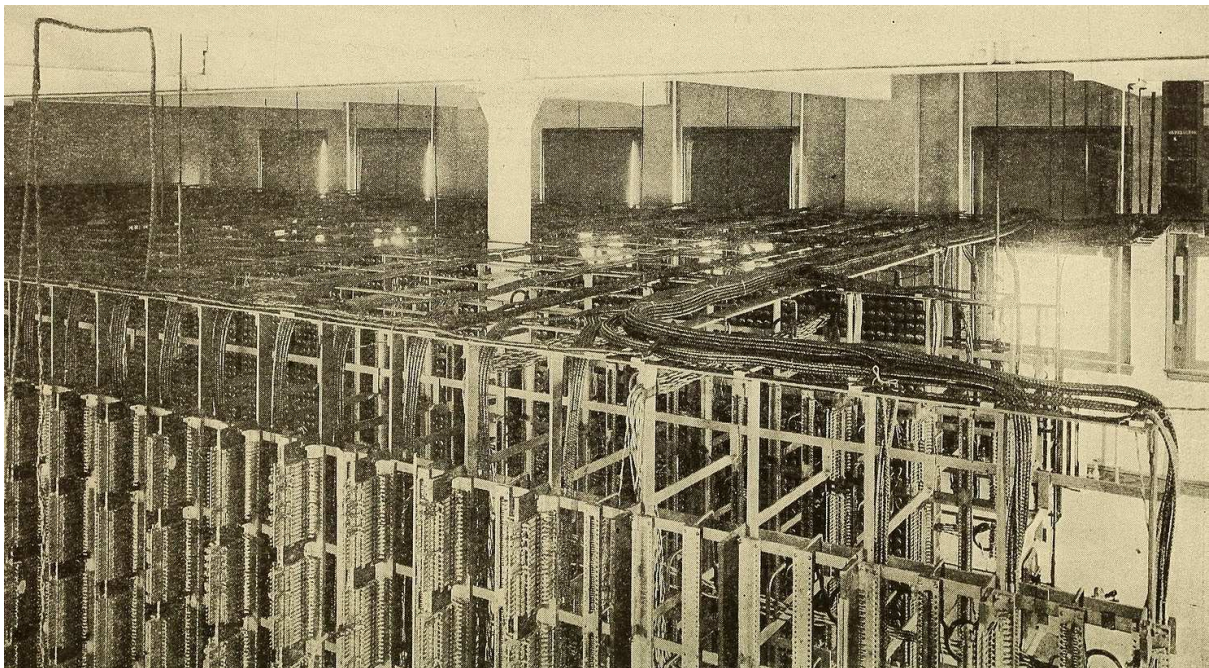
relés que puede llegar a ser increíblemente numerosa. Bien, hasta hace no mucho tiempo, todas las automatizaciones se hacían de esa manera, combinando relés. Un ejemplo clásico son las centrales telefónicas.

Todas y todos vimos películas ambientadas en las primeras décadas del Siglo XX en las que operadoras conectaban a mano el teléfono A al teléfono B para establecer una comunicación.

En la actualidad una multitud de computadoras (servidores) se encargan de comunicar al mundo entero. Todas nuestras comunicaciones son digitales, por eso podemos enviar audios, fotos, archivos, videos, todo está digitalizado; sería imposible conectar a mano cada llamada de las millones que hay en el mundo en forma simultánea.



¿Qué hubo históricamente entre las operadoras manuales y los teléfonos inteligentes? Sí, millones de relés que se encargaban de automatizar las comunicaciones. Esta foto de principios del SXX corresponde a la compañía telefónica de Philadelphia. Paneles conteniendo miles y miles de relés conectaban a los usuarios de teléfonos de esa ciudad.

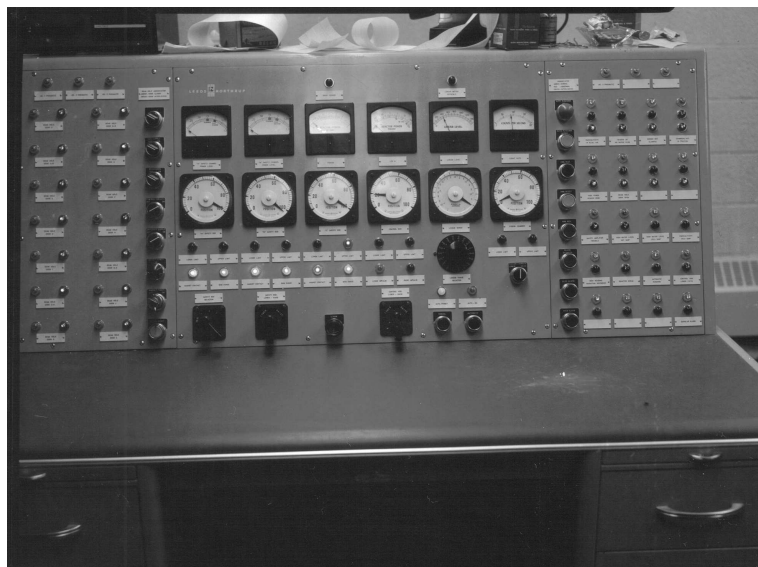


Afortunadamente los tiempos cambiaron y toda esa lógica de comandos que se ve en la foto, hoy entra en el espacio que ocupa una computadora doméstica. Lo mismo pasaba en las industrias donde para automatizar una línea de producción se necesitaban multitud de relés y hoy son reemplazados, por ejemplo, por placas Arduino que vamos a aprender a programar en este mismo curso.

Aquí se observa un típico tablero de comando industrial, con sus botones, pulsadores, llaves selectoras, luces indicadoras e instrumentos de control. También con componentes de aviso de fallas críticas, se destacan las sirenas, bien visibles sobre el tablero.



En la siguiente imagen tenemos nada más y nada menos que el tablero de control de un reactor nuclear de mediados el SXX. Estaba implementado sobre un escritorio, podemos ver hasta sus cajones!

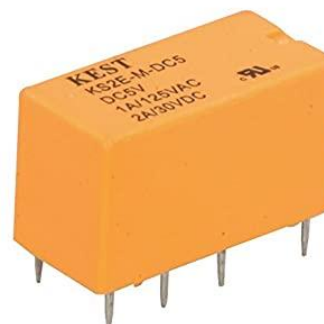


Siempre recordemos que **estos tableros son de comando, no de potencia**. Detrás de esos circuitos de comando hay circuitos de potencia, generalmente en otros espacios físicos, que son comandados por estos tableros. La profusión de luces indicadoras e instrumentos sirven a los operadores para confirmar que las tareas que ejecutaron se han cumplido. Si por ejemplo se enciende a través de un pulsador un horno de secado industrial, existe una lámpara testigo (generalmente verde) que confirma que se ha encendido el horno, y si el mismo se apaga o tiene una falla también habrá una lámpara de otro color (generalmente rojo), un testigo o algún tipo de aviso de aquello que no está funcionando bien (probablemente amarillo). Ya veremos más adelante cómo se implementa.

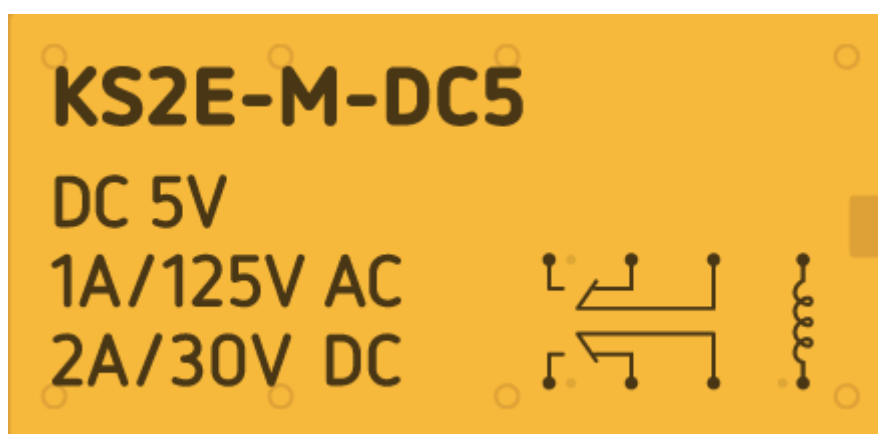
Primera experiencia con relés

FICHA 3: Circuitos de comando y de potencia Preguntas sobre relés

Para terminar esta ficha vamos a conectar nuestro primer relé a través de nuestro protoboard. Tenemos que tener en cuenta que el relé comercial tiene el aspecto de un cubito de plástico opaco con unos terminales que salen del mismo. Tenemos que identificar en cada modelo de relé qué terminales corresponden a la bobina y cuáles a los contactos NA y NC. Tanto en el caso del relé virtual de nuestro simulador como en un relé real, vamos a usar el modelo KS2E-M_DC5.

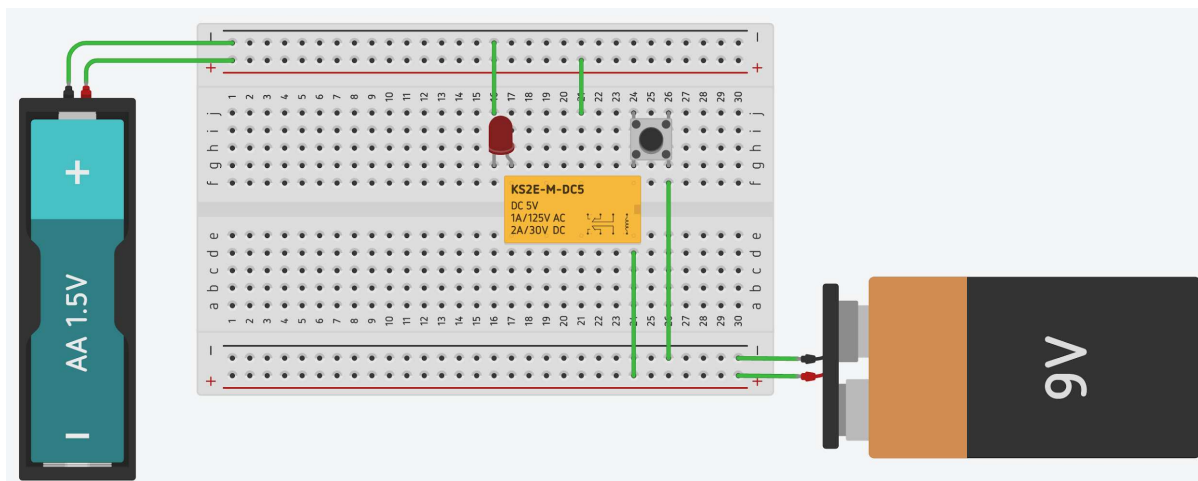


Pero... tiene 8 contactos, ¿cuál es cuál?. Para no confundirnos, analicemos cuidadosamente.

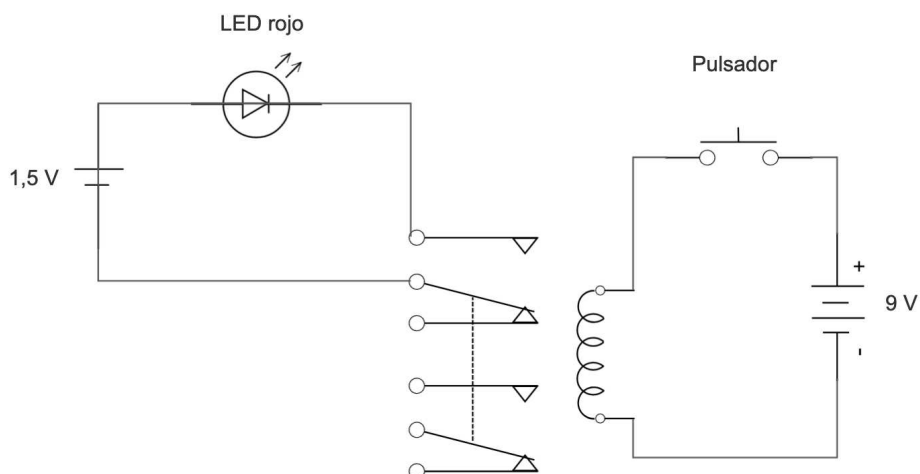


Veamos detenidamente el esquema que tiene impreso sobre la cara superior. Los dos contactos de la derecha son los de la bobina. Por ahí no tenemos problema. Ahora viene la novedad: la mayoría de los relés tienen dos juegos de NA / NC. Aprovechando el espacio proveen dos contactos en lugar de uno. Por lo tanto en el sector superior se conecta uno y en el inferior, otro. Esto se llama “**relé doble inversor**” y son totalmente independientes eléctricamente pero se mueven al unísono.

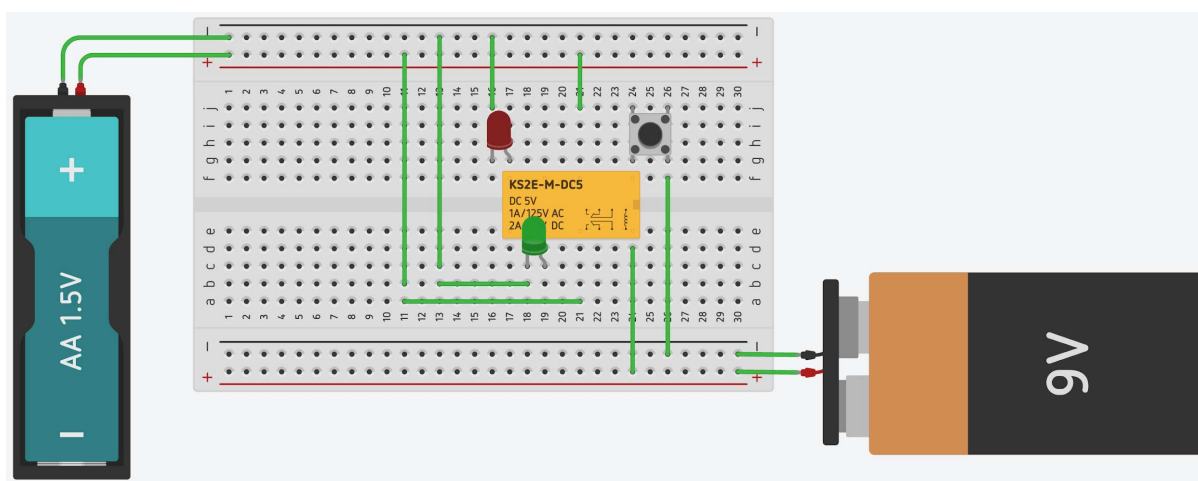
Vayamos lentamente. Hagamos dos circuitos, uno usando el relé superior y otro usando los dos relés. Además vamos a usar LEDs y pulsadores. Hagamos, usando un protoboard virtual o real, el siguiente circuito:



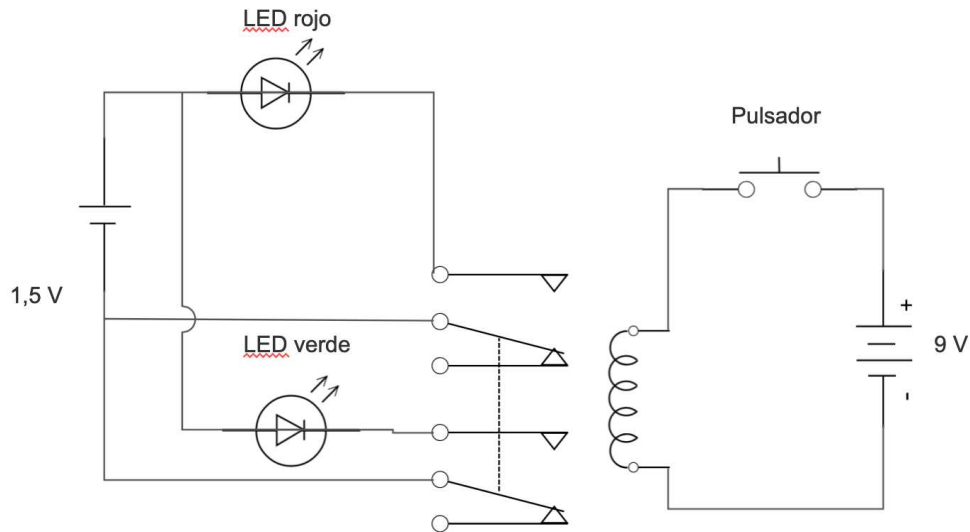
Tenemos dos circuitos separados. Uno de 9V que permite activar el relé a través del pulsador y otro que alimenta el led rojo a través de la pila de 1,5V. Cuando se presiona el pulsador, se encenderá el LED. Simbólicamente se representa de esta manera:



Usando el doble relé usamos un juego de contactos para el LED rojo y otro para el LED verde. Cuando uno está encendido, se apaga el otro. Juguemos con esta selección, a través del pulsador:



En símbolos:

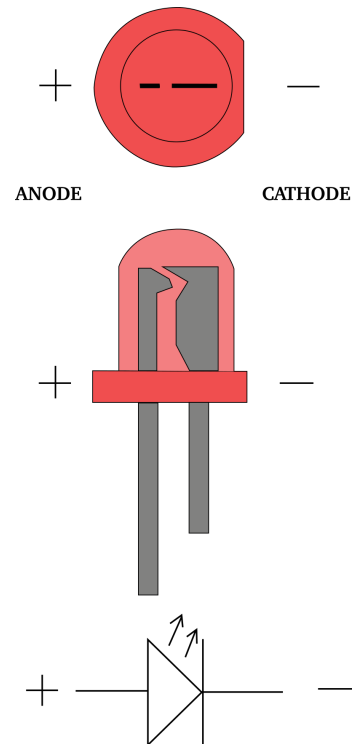


¿SABÍAS QUÉ?

LED quiere decir **Diodo Emisor de Luz** (en inglés, *Light Emitting Diode*). Es un **semiconductor** que tiene la facultad de iluminar. En sus orígenes se usó solo para testigos y pequeñas implementaciones. Es típico verlos como indicadores de encendido y apagado de televisores, equipos de audio, etc.

Más acá en la historia se desarrollaron para iluminación de automotores, domicilios y hasta alumbrado público. Su funcionamiento en frío y un consumo sustantivamente menor al del alumbrado incandescente lo hacen la tecnología más adecuada para la iluminación actual.

Una diferencia con la lámpara tradicional es que a diferencia de esta, el LED tiene polaridad. No se puede conectar de cualquier manera. Posee ánodo y cátodo. La corriente circula de cátodo a ánodo. El cátodo se identifica por su pata más corta y su periferia recta. También tenés el símbolo que lo representa en todo el mundo.



Automatización con relés

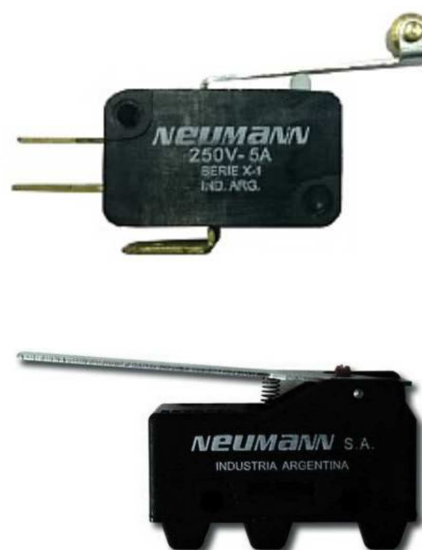
Sensores

Microswitches

¿Cómo “sabe” el ascensor que llegó al piso indicado? ¿Cómo “hace” la puerta del supermercado para abrirse justo cuando pasamos por delante de ella? ¿Cómo “detecta” la TV que apretamos un botón del control remoto? ¿Cómo “se prenden solas” las luces de la calle cuando se hace de noche?. La respuesta es la misma en todos los casos: gracias a un sensor.

Un **sensor** es un dispositivo que es capaz de transformar la variación de sus capacidades físicas (mecánica, de temperatura, lumínica, dinámica óptica, etc.) en variaciones eléctricas. Hay sensores de movimiento, de calor, de frío, de humo, de oscuridad/luz, de ruido, de impacto... la cantidad de sensores es virtualmente infinita. Hay sensores en el mercado y sensores que podemos hacer nosotras y nosotros con materiales que usamos todos los días en nuestra casa. Ya iremos más adelante con ellos. Vamos a empezar por uno de los más simples y más comunes. El sensor **microinterruptor**.

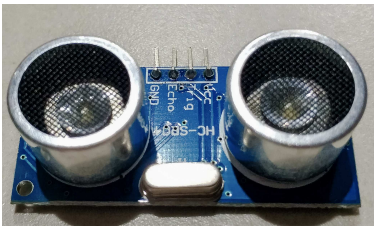


Se trata de un pulsador común que tiene un dispositivo de chapa, plástico, neumático, elástico o de otro tipo que permite que sea presionado por un objeto móvil en lugar de por una persona. Los ascensores (para detenerse en los pisos), los portones y las máquinas automáticas, la puerta de tu heladera y las de los autos, e infinidad de dispositivos usan este tipo de sensor. Suelen tener un contacto NA y uno NC y están disponibles en una multitud de tamaños y formas. Eléctricamente hablando, es el mismo tipo de sensor el que se usa para controlar el movimiento de la caja de un camión volcador y el que se coloca para verificar que la tapa de la multiprocesadora que usamos en la cocina está bien cerrada; pero claramente tienen que soportar pesos y condiciones de trabajo totalmente diferentes.



Sensores varios

FICHA 4: Automatización con relés Sensores

Si bien los microswitches son los sensores más utilizados, como se dijo hay muchos otros sensores que te describimos a continuación.

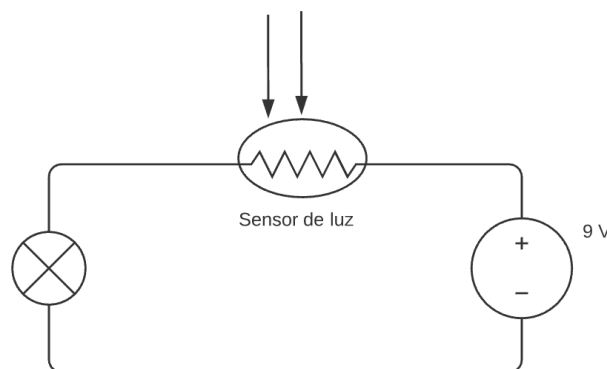
<p>Sensor ultrasónico</p>		<p>Consta de dos partes como se ve en la figura: un emisor y un receptor. El emisor envía un ultrasonido que al rebotar en un objeto y volver al receptor permite calcular la distancia a dicho objeto.</p>
<p>Sensor de humedad</p>		<p>Mide la humedad ambiente y de acuerdo a ella abre o cierra un circuito. Se usa típicamente en sistemas de riego automático.</p>
<p>Sensor de temperatura</p>		<p>Existen muchos tipos de estos sensores. Básicamente actúan abriendo o cerrando el circuito a determinadas temperaturas en las que vienen calibrados. Se usan en hornos industriales, laboratorios, etc.</p>
<p>Sensor de luz</p>		<p>Miden la luz ambiente y cierran o abren el circuito según la intensidad de la misma. Tienen un regulador para setear esa intensidad de corte.</p>
<p>Sensor de sonidos</p>		<p>Mediante un pequeño micrófono toma el sonido ambiente y actúa según su intensidad. También tiene un regulador para calibrar la intensidad de activación.</p>

<p>Sensor infrarrojo</p>		<p>Consta de un emisor de luz infrarroja (invisible para los humanos) y un receptor de la misma. Si la luz emitida choca con una superficie clara vuelve al receptor, en cambio si choca con una superficie oscura no se reflejará y el receptor indicará ese déficit.</p> <p>También se pueden colocar el receptor alineado con el receptor (por ejemplo a los lados de una puerta) y cuando pasa un objeto entre ellos corta el haz de luz infrarroja, cosa que el sensor detectará.</p>
<p>Sensor de inclinación</p>		<p>Primariamente se hacían con una ampolla de vidrio en la que había dos contactos separados y cierta cantidad de mercurio (metal líquido) dentro de la ampolla. Según se incline para un lado o para otro cierra o abre el circuito. Hoy se utilizan sensores electrónicos como acelerómetros, inclinómetros que pueden detectar prácticamente cualquier movimiento: inclinación, caídas, rotación, agitación, etc. Casi todos los celulares tienen este tipo de sensores incluidos en su hardware. Por eso cambia la distribución de la pantalla al rotar el celular.</p>

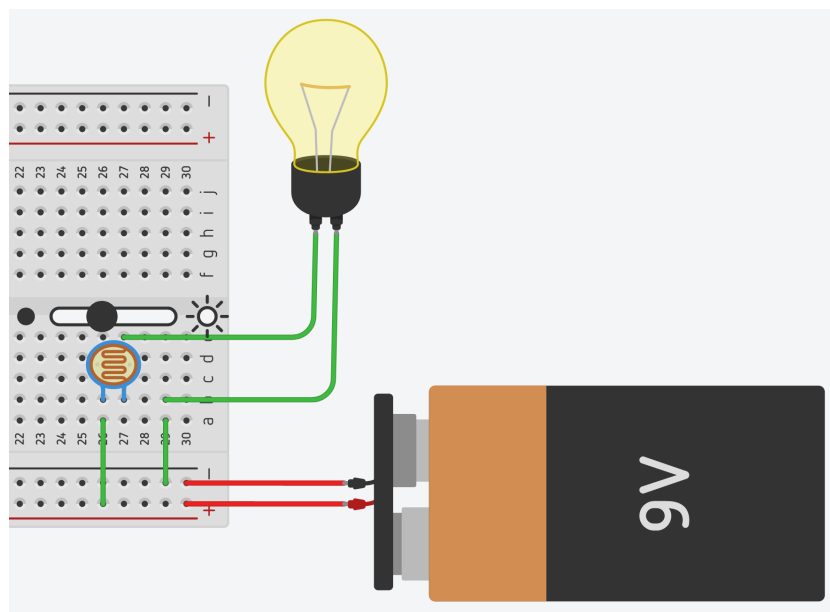
Hay sensores **analógicos** y **digitales**. Los digitales admiten dos posibilidades, **activado** y **no activado**. De ahí su denominación. El microswitch es el ejemplo clásico de los mismos: o está en una posición o está en otra. Los sensores analógicos, en cambio, admiten muchos estados entre el "apagado" y el "encendido". Por ejemplo, un sensor de luz detecta **oscuridad** y distintos grados de iluminación.

FICHA 4: Automatización con relés
Sensores para encender y apagar

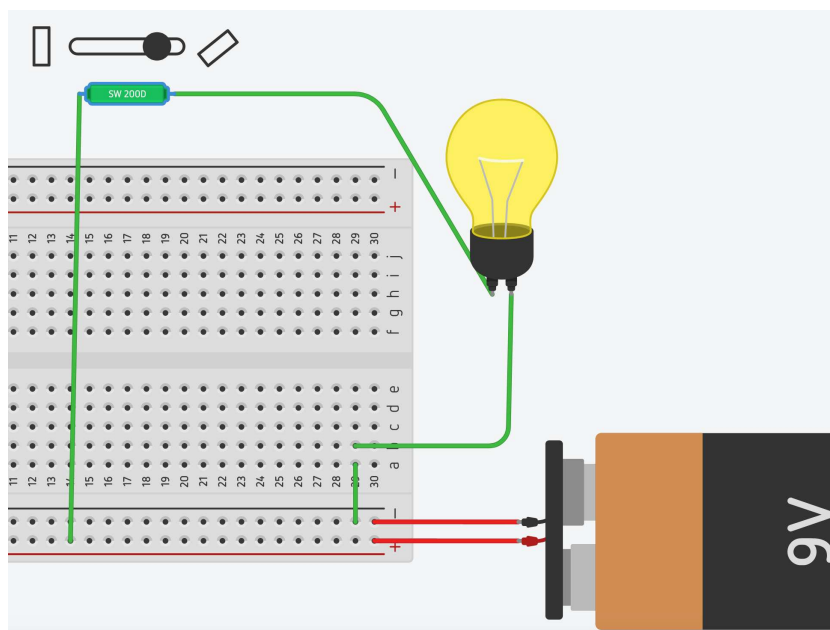
En el siguiente circuito se tendrá diferente voltaje a la salida según la intensidad de la luz que reciba el sensor:



Hagamos ahora el circuito en el protoboard y juguemos con el sensor de luz. Se puede simular, mediante un deslizador, la intensidad de luz que recibe y se ve cómo varía de la misma manera la intensidad de corriente que circula por la lámpara:



Otra prueba que te sugerimos es probar el sensor de inclinación:



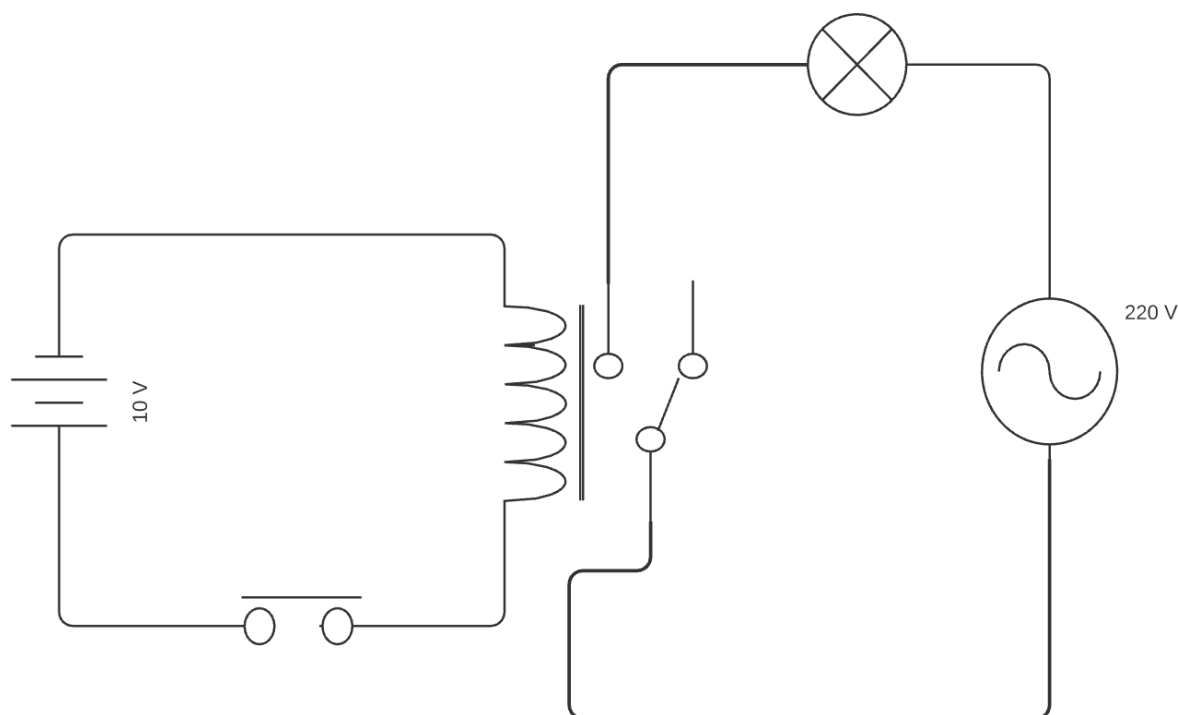
Como siempre, no nos limitemos a las imágenes y el texto. Repliquemos cada circuito en el simulador o con componentes reales. Experimentemos con ellos, juguemos, observemos y razonemos sobre qué cosas van a suceder en determinados escenarios. Luego los reproducimos y confirmamos, o no, lo que suponíamos de antemano.

FICHA 4: Automatización con relés
Mando de la Play

Relés y memoria: Enclavamiento

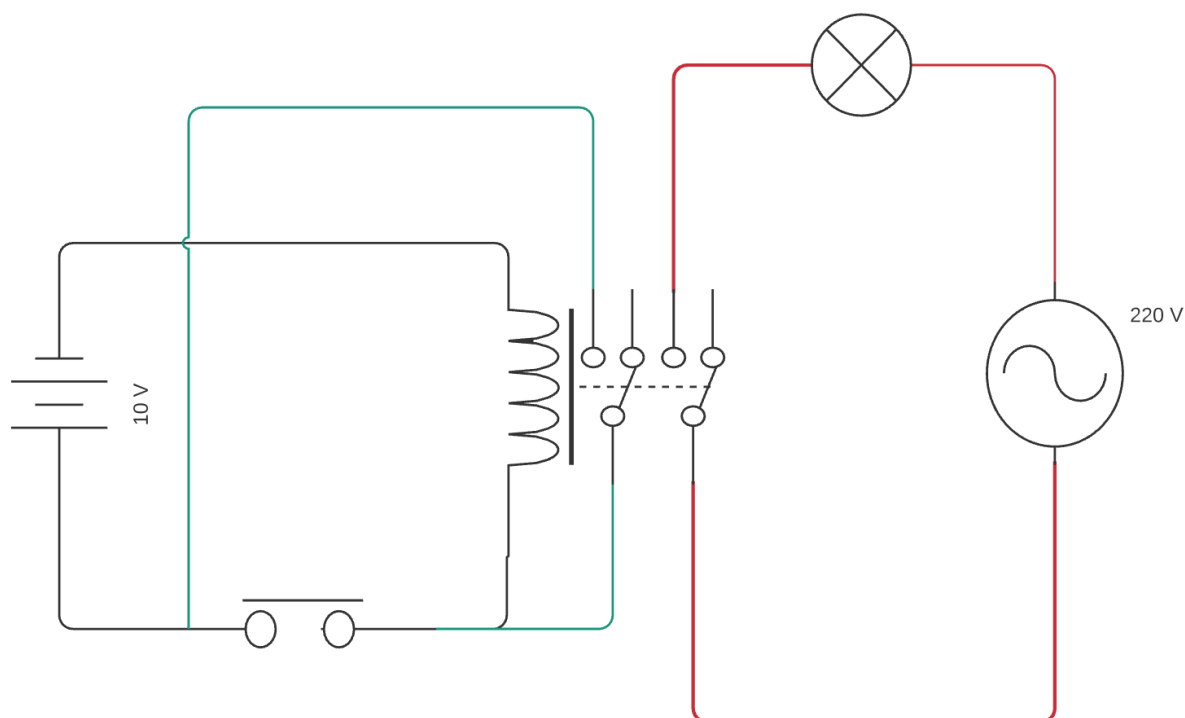
En la ficha anterior aprendimos los conceptos básicos y las conexiones NA y NC de un relé. Analicemos un caso: Cuando subimos a un ascensor, apretamos el botón pulsador correspondiente al piso al que queremos ir y listo. No tenemos que tener apretado el botón hasta que llegamos, sin embargo el ascensor parece recordar el piso, incluso generalmente tienen una luz indicadora que se enciende al apretar ese botón. ¿Cómo memoriza el relé esa selección?

Ese efecto se logra con el llamado **enclavamiento**, un artificio que permite que el relé quede activado aún cuando soltemos el botón.



En la figura, si apretamos el pulsador la lámpara se encenderá, pero si lo soltamos se abrirá el circuito y se apagará. Dijimos que el relé en realidad tenía dos juegos de contactos independientes. Usémoslo para conectar en paralelo el NA del relé con el pulsador de encendido.

Resulta un circuito extremadamente útil y, desde ya, muy usado en la práctica.

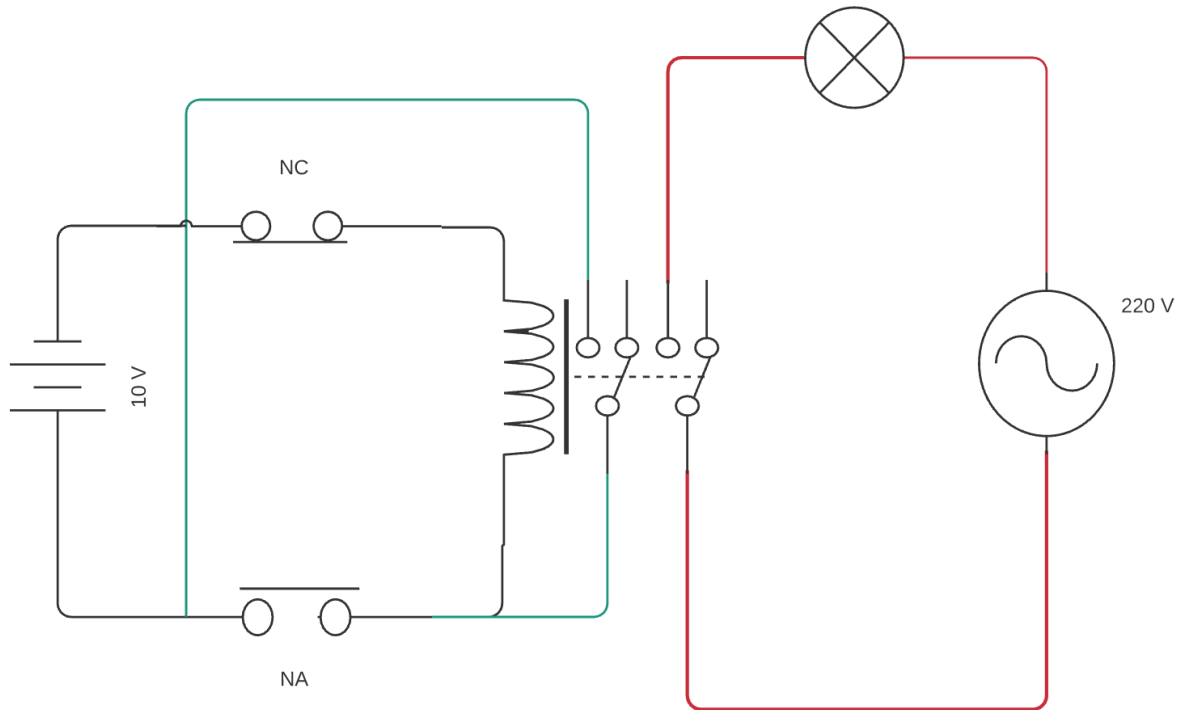


Observemos el relé doble inversor, de sus dos juegos de contactos: uno lo vamos a usar para potencia (en rojo) y el otro lo usamos en comando para conectarlo como “puente” en **paralelo** al pulsador. Al presionar el mismo, el contacto verde del relé va a cerrar el circuito y al soltar el pulsador la corriente seguirá circulando por el sector verde manteniendo el relé activado.

Es complejo darse cuenta lo que sucede, por más rápido que sueltes el botón del pulsador, jamás podrás “ganarle” a la corriente eléctrica, que prácticamente se transmite a la misma velocidad que la luz, entonces el relé quedará alimentado, esto se llama enclavamiento y permite que se usen casi siempre pulsadores en lugar de interruptores en los circuitos de comando. “Botones” en lugar de “llaves”, como sí usamos, por ejemplo, en nuestras casas.

Ahora.... ¿Cómo se apaga la luz?

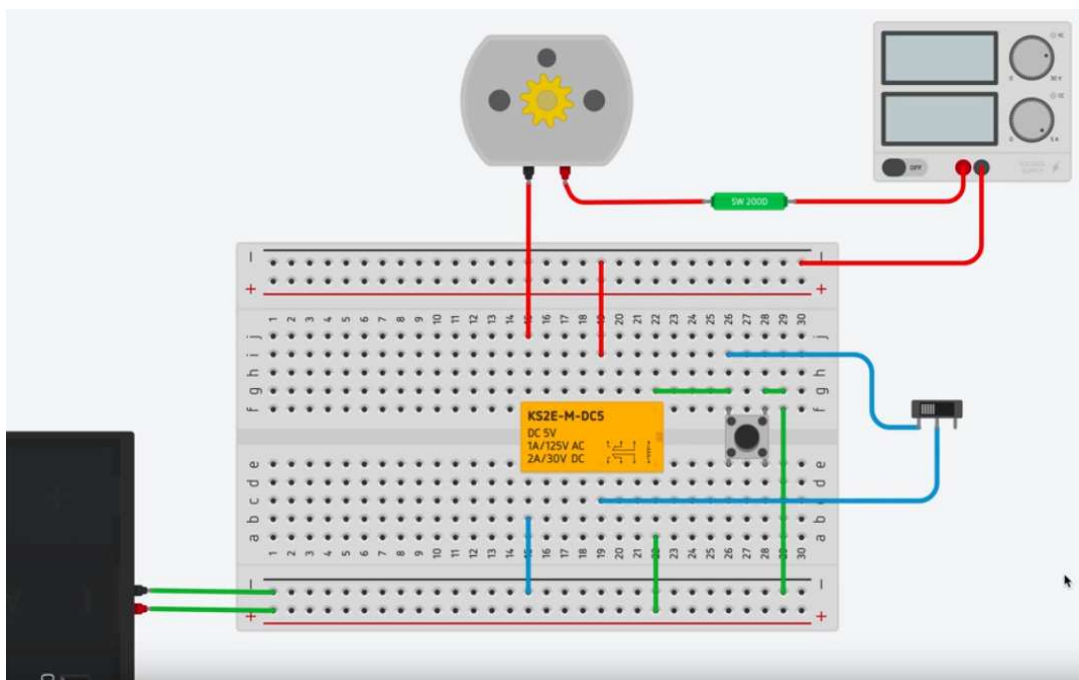
Para ello vamos a usar un pulsador particular, el **Normalmente Cerrado**. Sí, el de la puerta de la heladera!. En reposo mantiene el circuito cerrado - de ahí su nombre NC - y al oprimirlo abre el mismo. Colocando un NC en cualquier lugar entre la alimentación y la bobina del relé tendremos el efecto deseado:



El NA se usa para encender (ON) y el NC para apagar (OFF)

Para mayor claridad hemos usado cables de tres colores diferentes:

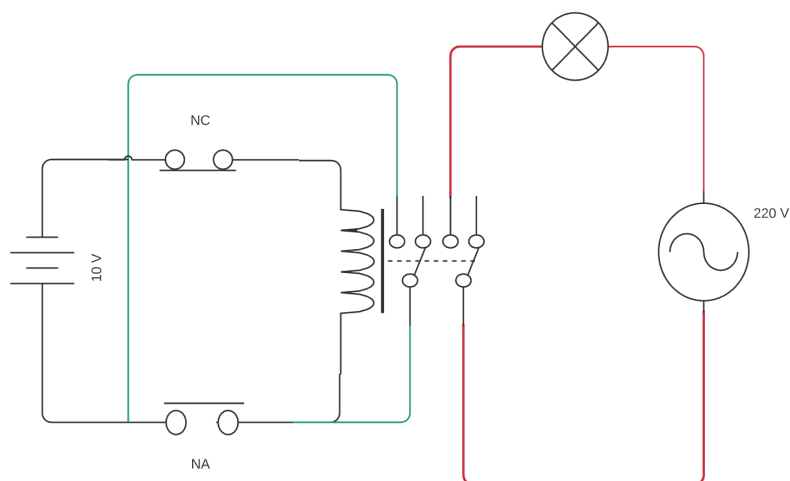
- En rojo está el circuito de potencia.
- En verde el circuito de comando
- En celeste el enclavamiento del relé



FICHA 4: Automatización con relés

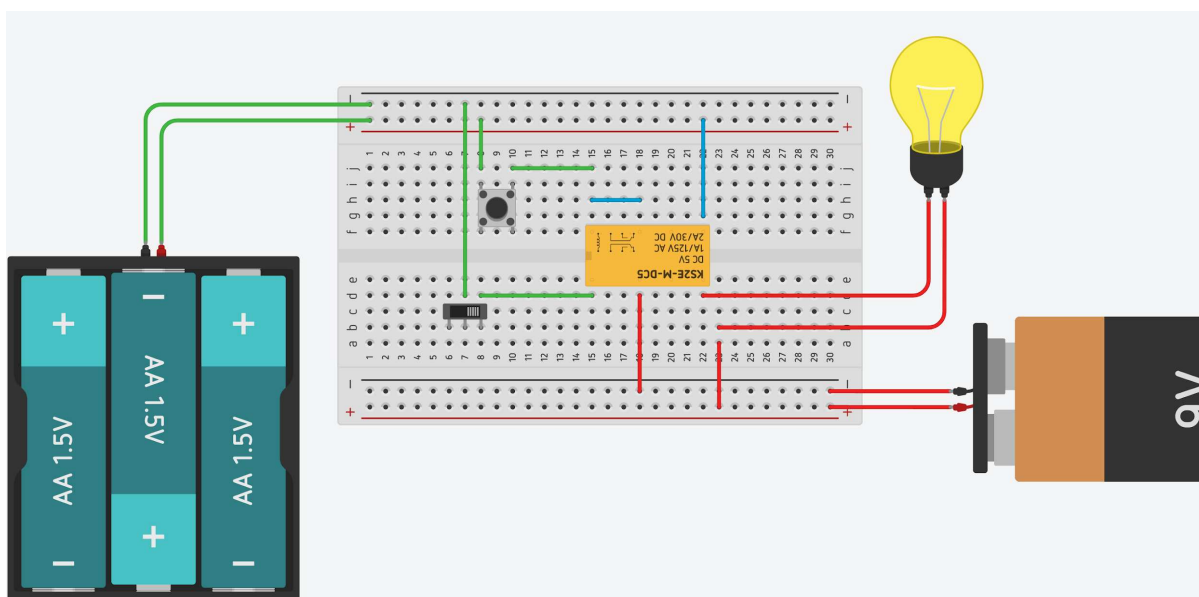
Apagando la luz

El próximo desafío consiste en agregar un pulsador en el circuito de comando para apagar la luz. Esto se consigue interrumpiendo el circuito del enclavamiento del relé: si agregamos un pulsador NC en este circuito, cuando lo presionamos, se interrumpirá la circulación de corriente por la bobina del relé, liberando el circuito de potencia.



Y para terminar: supongamos que necesitamos agregar una lámpara que indique que el circuito de potencia está desconectado... ¿cómo lo harías?

Para esto, se debe aprovechar el contacto NC del relé. A esto apunta la pista de la ficha. Una solución posible es la de la siguiente imagen:

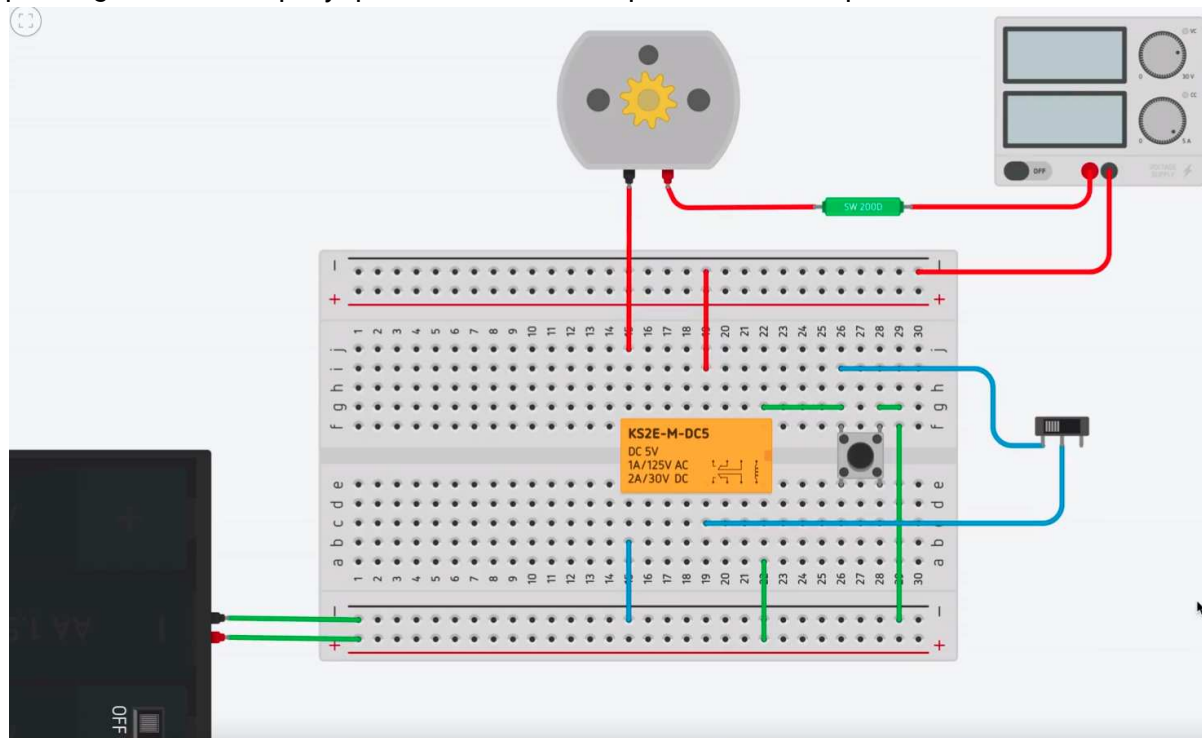


Circuito automatizado completo

FICHA 4: Automatización con relés Desafío final

Primera Parte

Y para finalizar esta unidad de sensores y relés, hagamos un circuito automático completo, que active una bomba de llenado cuando un sensor indique mediante su inclinación que hay poca agua en un tanque y que se active cuando presionamos un pulsador.



Como siempre hemos puesto diferentes colores para los sectores de este circuito:

- Verde para el circuito de comando.
- Rojo para el circuito de potencia, en este caso a 24 V alimentando a un motor.
- Celeste para el enclavamiento del relé.

El pulsador sirve para encender la bomba (representada por el motor), pero esta no arrancará hasta que el sensor de inclinación actúe por falta de agua. Cuando el agua suba de nivel, el inclinómetro se pondrá horizontal y volverá a apagar el motor. Este ciclo se repetirá hasta que cortemos el interruptor general del circuito de comando, en este caso ubicado en el portapilas situado a la izquierda. También, como vimos podemos usar un pulsador NC. En cualquier caso, por más que el inclinómetro indique falta de agua, el motor no arrancará porque el relé no va a cerrar el circuito. Por otro lado, si apretamos el pulsador pero hay suficiente agua en el tanque, el motor no arrancará.

Segunda parte

Si el motor se traba, por ejemplo a causa de una falla mecánica, va a levantar temperatura

porque seguirá recibiendo energía eléctrica pero al no poder girar convertirá esa energía recibida en energía térmica, ya que no podrá convertirla en movimiento.

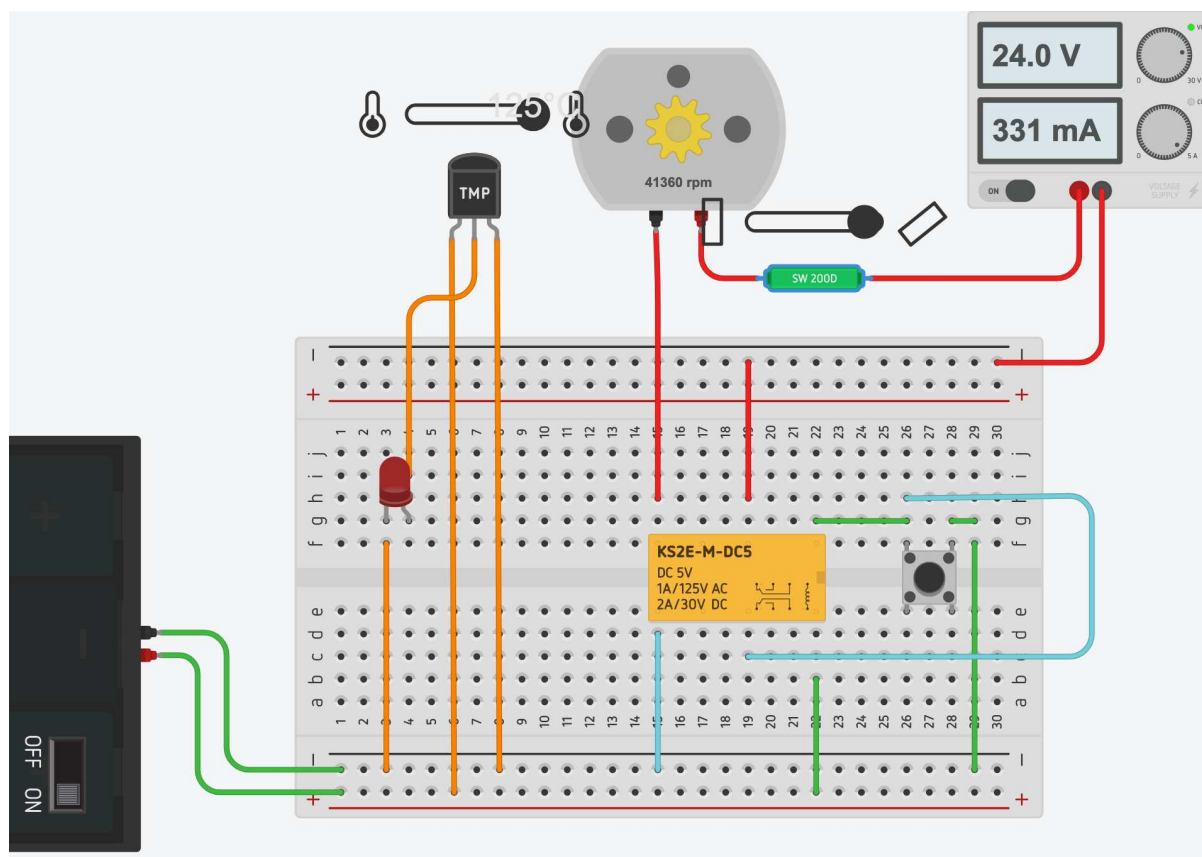
Podemos entonces hacer un indicador de motor parado, que encienda un led rojo cuando el motor levante temperatura. ¿Te animás a hacerlo?

AYUDA

El sensor de temperatura tiene tres contactos: dos de alimentación (uno +, otro -) y el central que emitirá una tensión cuando el sensor se caliente.

Solución

Se ha agregado el sensor de temperatura conectado con conductores de color naranja y se lo ha colocado cerca del motor para tomar su temperatura. Mediante el deslizador se simula un calentamiento para comprobar que el LED rojo se enciende en el tablero, avisando que algo no está funcionando bien. Más adelante veremos cómo usar Arduino, no solo para indicar fallas sino para cortar la corriente de manera automática y así proteger materiales, instalaciones y fundamentalmente la vida humana.



Un poco de electrónica básica

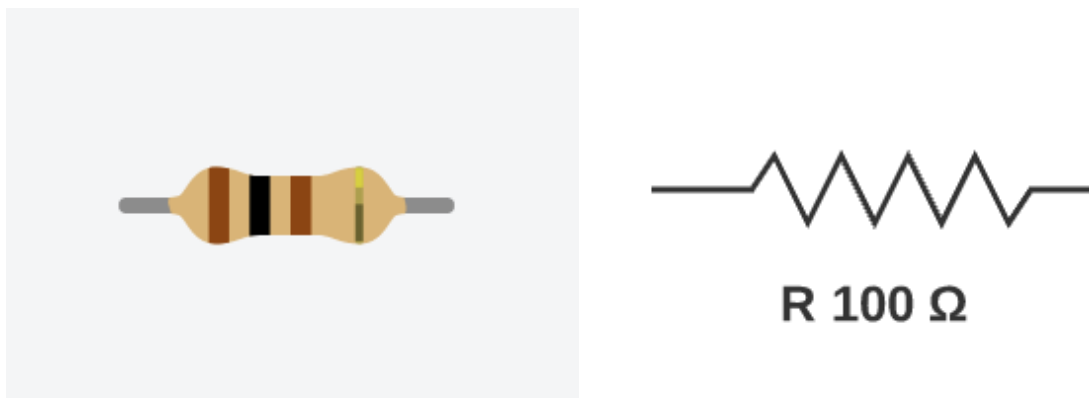
Las resistencias

FICHA 5: Un poco de electrónica básica

Calculamos el valor de las siguientes resistencias

Son componentes muy comunes en los circuitos electrónicos, y tienen la finalidad de **reducir la corriente eléctrica** que circula por un conductor oponiéndose a la misma. Para esta acción usan las propiedades físicas de su material de construcción, mayormente carbón. El valor de la resistencia se mide en **Ohms** (se representa con la letra omega Ω) y las vamos a usar mucho, ya que como dijimos los transistores, leds, chips y demás semiconductores consumen muy poca energía, si no usamos resistencias podemos quemarlos fácilmente.

Vemos el aspecto y el símbolo de la resistencia estándar de 100 Ω :











El cuerpo de la resistencia tiene unas tiritas de color, esas tiras permiten saber el **valor** de la misma, sin necesidad de medirlas con un tester. Es una manera muy simple de codificar:

- La primera tira es la primera cifra del valor.
- La segunda tira es la segunda cifra del valor.
- La tercera tira es el multiplicador, de otra manera: **la cantidad de ceros que hay que agregar.**

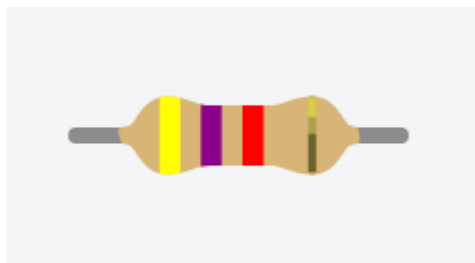
En el ejemplo de la figura es marrón - negro - marrón. Podemos inferir que el negro representa a un 1 y el marrón equivale a 0.

$$\text{Marrón} + \text{Negro} + \text{Marrón} = 1 + 0 + 1 \text{ cero} = 100 \Omega$$

Acá la tabla completa con los colores de las resistencias. Es muy fácil memorizarla para identificarlas en nuestros circuitos.

COLOR		VALOR
NEGRO		0
MARRÓN		1
ROJO		2
NARANJA		3
AMARILLO		4
VERDE		5
AZUL		6
VIOLETA		7
GRIS		8
BLANCO		9

Por ejemplo, calculemos una resistencia Amarillo - Violeta - Rojo



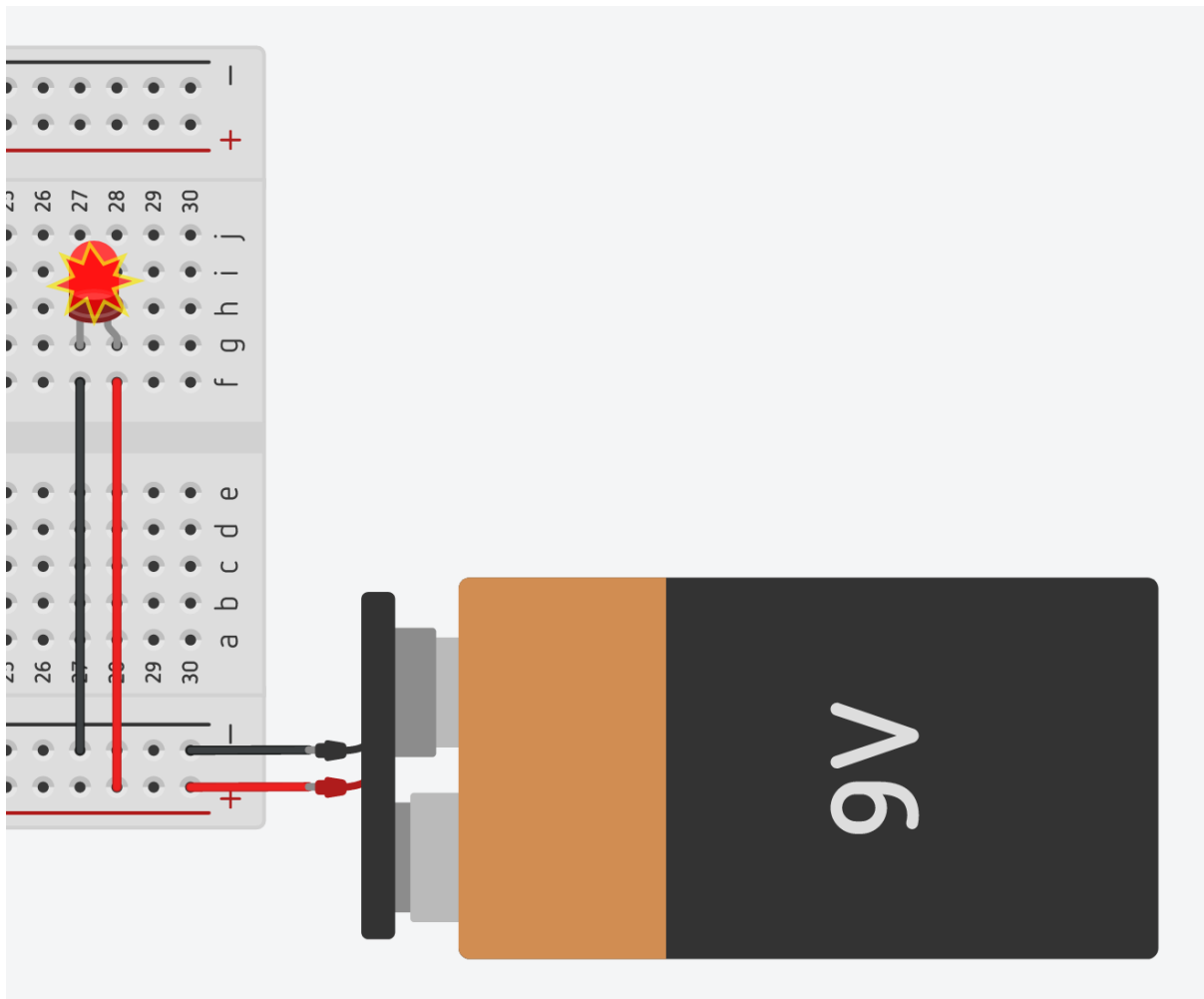
- Primer número: 4
- Segundo número: 7
- Cantidad de ceros: 2

Valor = 4700 Ω

Uso de las resistencias

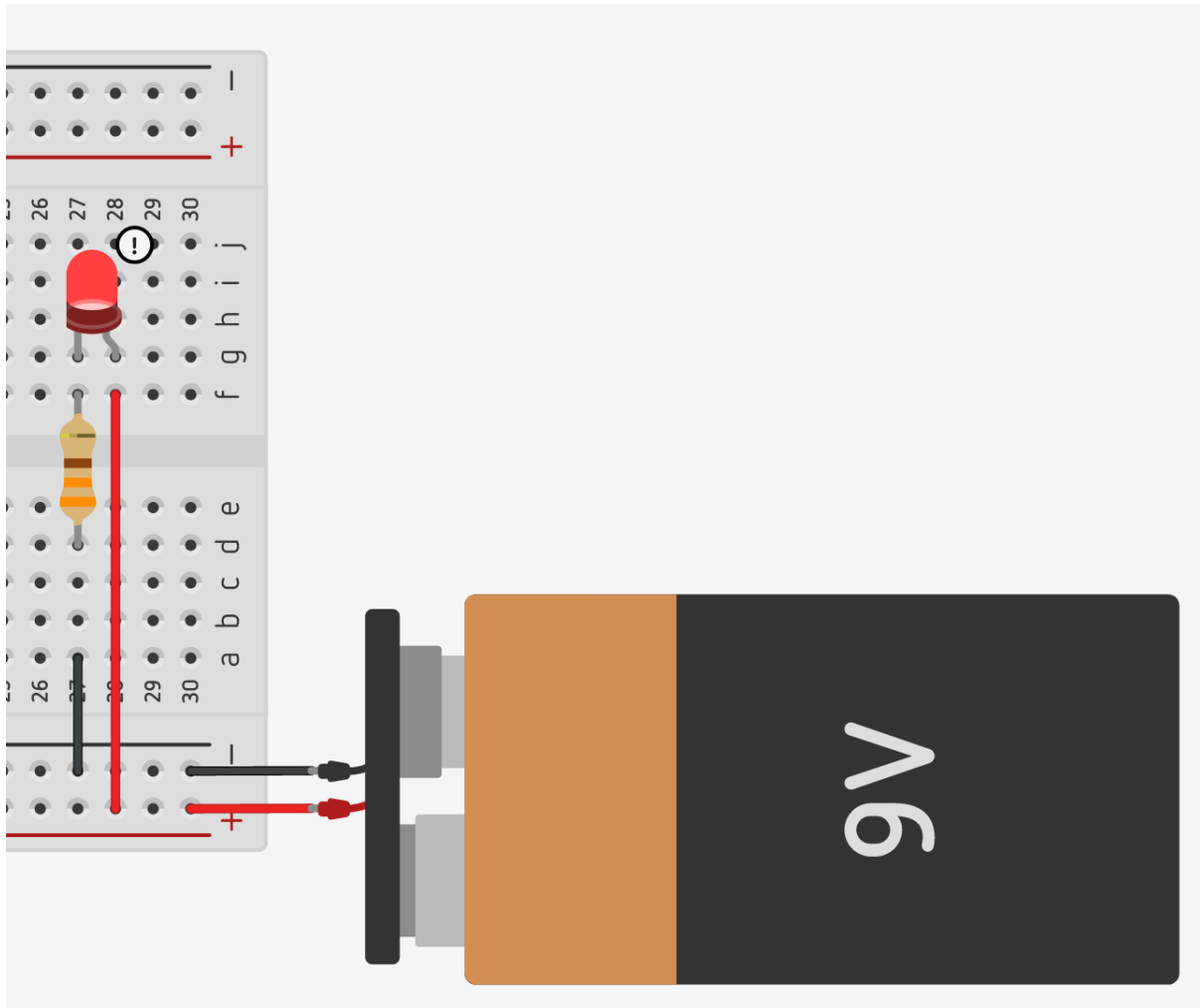
FICHA 5: Un poco de electrónica básica
Uso de las resistencias

Veamos unos ejemplos prácticos. Probemos primero conectar un LED directamente a una batería de 9V en nuestro protoboard y demos inicio a la simulación:

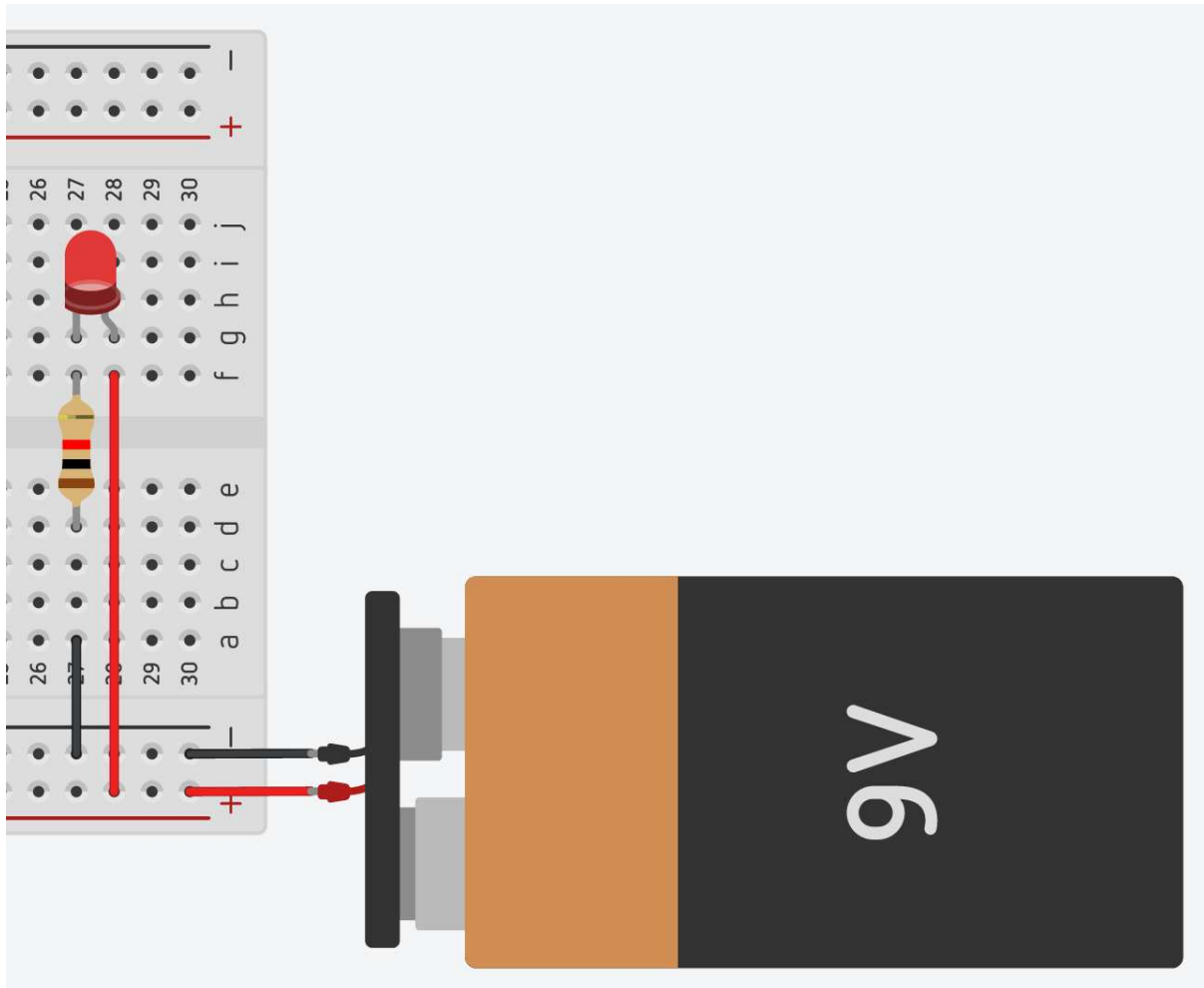



Vemos que el LED se quema por **sobretensión**. Efectivamente la tensión de trabajo de un LED es de alrededor de 2V y al aplicarle casi 5 veces este valor, no resiste y se quema. Afortunadamente estamos trabajando con un simulador, así que ¡no hay problema!

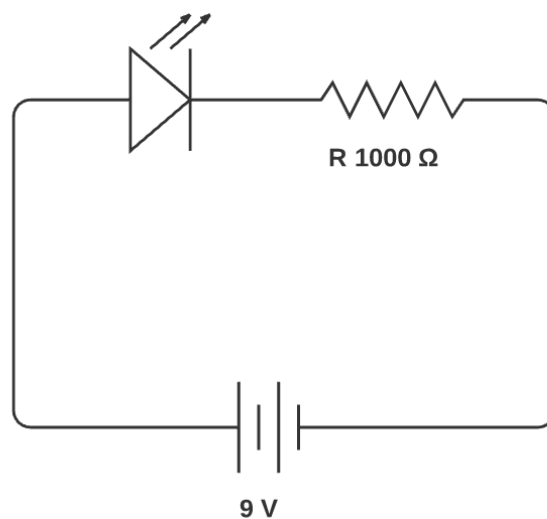
Tenemos que colocar una resistencia para que baje la tensión a como máximo 2V que es lo que necesitamos. Probamos con una resistencia de $330\ \Omega$ y el LED no se quema pero nos advierte con el símbolo (!) que está al límite de su salud física.



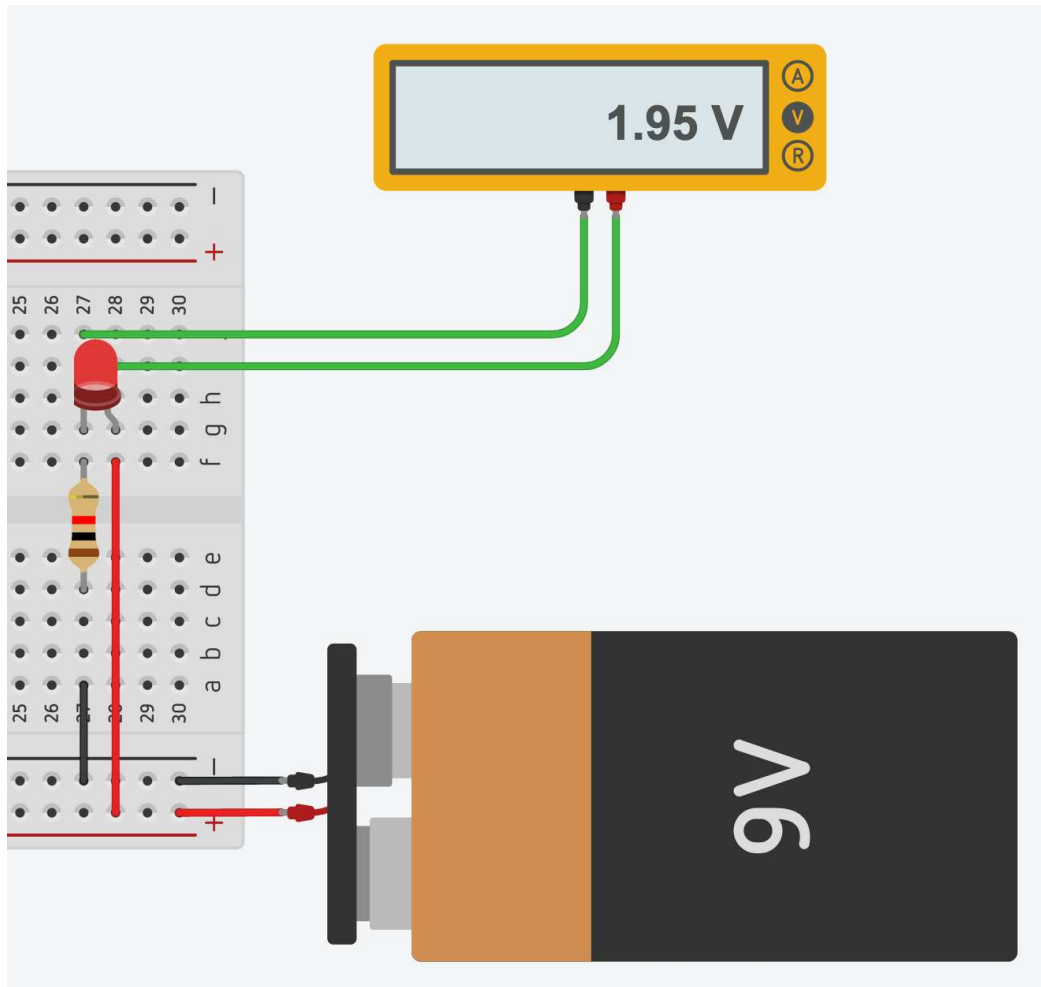
Subamos el valor de la resistencia para que baje la corriente que circula por el LED. Usamos ahora una de $1000\ \Omega$ ($1\ \text{K}\Omega$). Podemos comprobar que el LED se enciende y no está trabajando al límite de su capacidad. Entonces tomamos nota que cada vez que alimentamos un LED con una batería de 9V tenemos que poner en serie una resistencia de $1\ \text{K}\Omega$.



Debajo tenemos el mismo circuito de manera simbólica. La resistencia la vamos a colocar entre el  y el LED.



Para comprobar la tensión que llega al LED podemos usar un **tester** colocándolo en la posición V DC (Tensión en Corriente Continua). Por supuesto tenemos nuestro tester virtual en el simulador online.



Comprobamos que efectivamente la tensión es muy cercana a 2V que era el valor buscado.

LEY DE OHM

La resistencia, la tensión y la corriente en un circuito están directamente relacionadas por la llamada **Ley de Ohm**. Esta ley es una de las fundamentales para todo el cálculo eléctrico / electrónico y dice que:

"La intensidad de corriente (I) que circula por un circuito es directamente proporcional a la tensión aplicada (E) e inversamente proporcional a la resistencia total (R)"

La fórmula para calcularla es: $Corriente = \frac{Tension}{Resistencia}$

Y las unidades de medida son:

- **Corriente** [Ampere]
- **Tensión** [Volt]
- **Resistencia** [Ohm]

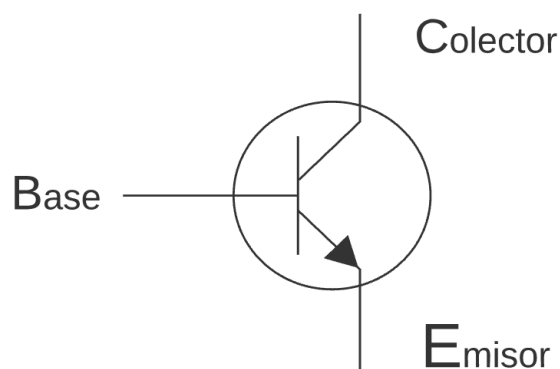
El transistor

A mediados del siglo pasado un invento cambió para siempre el desarrollo de los dispositivos electrónicos, **el transistor**. Resultó de investigaciones de científicos de Europa y EEUU sobre los materiales semiconductores, es decir que no son ni totalmente conductores ni totalmente aislantes. Estos materiales pueden ser lo uno o lo otro según varíen las condiciones físicas externas de temperatura, radiación, etc. Ejemplos de materiales semiconductores son el Antimonio, Carbono, Cadmio, Aluminio, y los más usados Germanio y Silicio, sobre todo este último. Ya hablamos del **diodo** y del **LED** en un capítulo anterior, ambos son implementaciones de materiales semiconductores. Como se dijo, estudiando estos materiales se diseñó el **transistor**, un componente que tiene tres terminales, y puede conducir o no corriente eléctrica entre dos de ellas según se inyecte corriente en el tercer terminal.

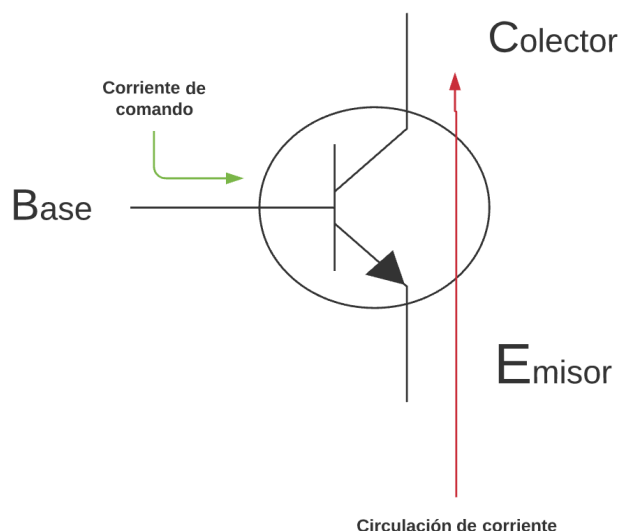


El aspecto de un transistor común varía en tamaño y forma según la potencia que disipan y la corriente que pueden manejar. Vemos en la figura de arriba algunos tipos de los más comunes.

Este es el **símbolo** del transistor y el nombre de sus terminales:



Veamos el funcionamiento del transistor. Tiene tres terminales llamados **Emisor, Base y Colector**. Según se estimule la Base podrá circular corriente o no entre el Emisor y el Colector.

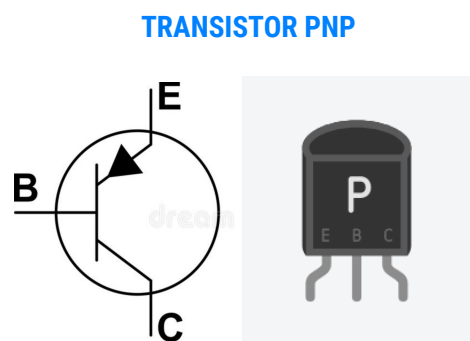
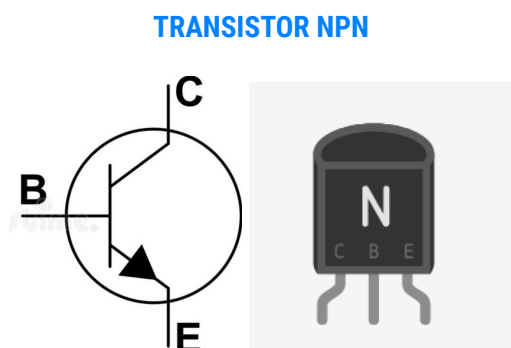


Es decir funciona como NA/NC, tal cual un relé, pero con enormes ventajas sobre el mismo:

- Consume muchísima menos corriente, literalmente microamperes, es decir fracciones millonésimas de un Ampere, la unidad de medida de la corriente eléctrica.
- Ocupa un espacio infinitamente menor que un relé.
- No tiene partes móviles, o sea que no hay desgaste mecánico.
- Como no hay resortes ni contactos móviles el cambio de estado es instantáneo, no hay inercia, se puede pasar de cerrado a abierto millones de veces por segundo sin problemas.
- Al consumir mucha menos corriente permite alimentar los circuitos con pilas comunes, por lo tanto favorece el desarrollo de dispositivos portátiles.

¿SABÍAS QUÉ...?

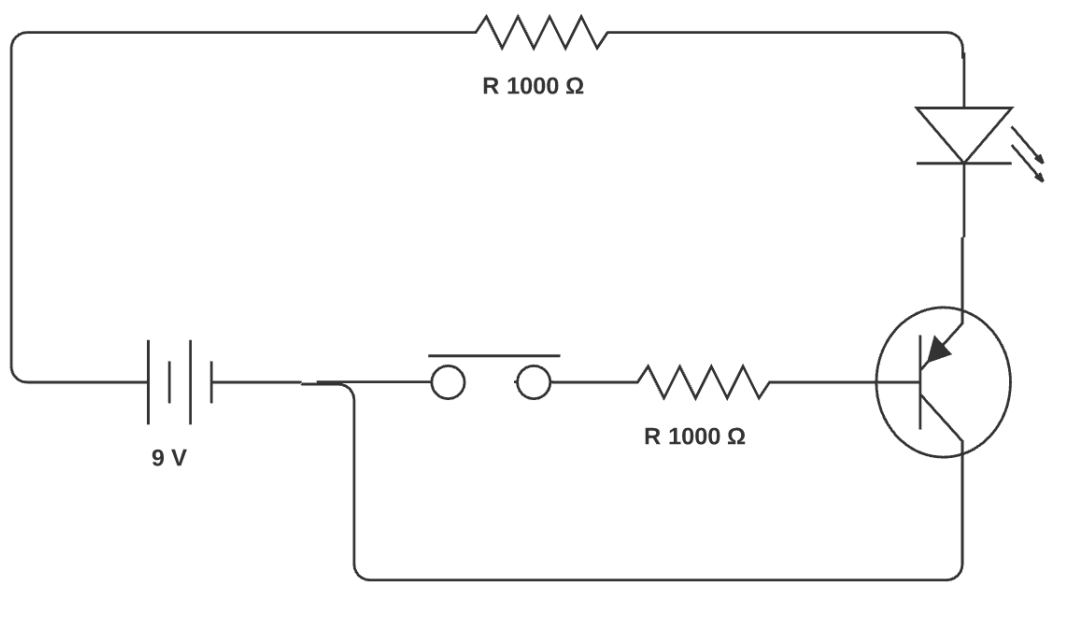
Hay dos tipos de transistores: el **NPN** y el **PNP**. Básicamente funcionan igual, pero en el NPN la corriente circula entre el emisor y el colector mientras que en el PNP circula desde el colector al emisor. Es decir en sentido contrario. Imaginate que entre colector y emisor hay un diodo y la base es la "llave" que maneja la circulación o no por ese diodo.



El transistor como interruptor

FICHA 5: Un poco de electrónica básica El transistor como interruptor

Para terminar la unidad vamos a hacer un circuito de comando pero usando un transistor en lugar de un relé.



El circuito funciona de la siguiente manera. Al apretar el botón pulsador le damos tensión a la base del transistor, quien al recibir ese estímulo deja pasar la corriente entre el colector y el emisor, en la dirección de la flecha roja, por lo que el LED se enciende. Al soltar el pulsador, deja de pasar corriente por la base del transistor y por lo tanto abre el circuito entre su base y emisor, apagando el LED.

Tomamos nota que usamos, y usaremos, varias resistencias. Esto es porque de la misma manera que un LED, los transistores (y de hecho todos los componentes) tienen tensiones de trabajo que hay que respetar. Cuando alguien planifica un circuito usa constantemente las **fichas técnicas** de los componentes para saber en qué valores de tensión, corriente, temperatura, etc el dispositivo trabaja sin correr peligro de romperse.

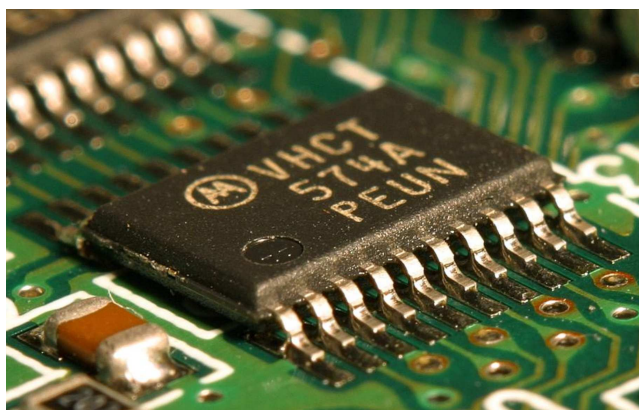
FICHA 5: Un poco de electrónica básica Compuertas OR y AND con transistores

Ahora que ya vimos que podemos usar el transistor como un interruptor... ¿Podremos hacer compuertas OR con transistores, como hacíamos con los interruptores mecánicos? ¿Y compuertas AND?

La respuesta a ambas preguntas es "sí". Por lo tanto, si combinamos muchos transistores podemos hacer circuitos lógicos complejos, complejísimos. Simplemente hay que combinarlos como antes combinábamos los relés. En la ficha, tenemos dos circuitos con transistores que se comportan como compuertas lógicas. Falta definir, en base a cómo se comportan en función de sus entradas, cuál es una compuerta AND y cuál una OR.

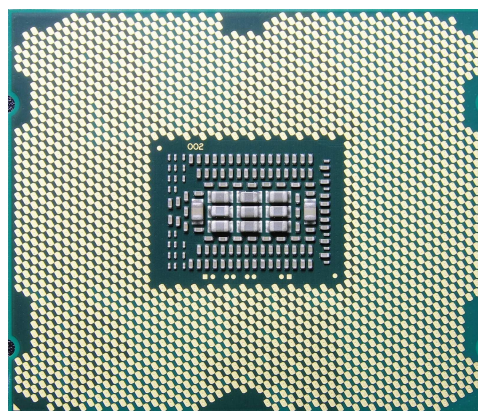
El chip

Cuando introdujimos los relés, vimos una foto de un enorme salón de la Cía. Telefónica de Philadelphia con decenas y decenas de tableros que a su vez albergaban miles de relés. Hoy un procesador de computadora hogareña tiene alrededor de setecientos millones de transistores. Sí, lo que antes ocupaba un edificio entero hoy cabe en un dispositivo más chiquito que un alfajor. La miniaturización de los semiconductores lo ha hecho posible. Y en gran parte se lo debemos al **chip, microchip o circuito integrado**, tres maneras de denominar a lo mismo. Básicamente es un proceso de construcción de dispositivos que combina láminas microscópicas de material semiconductor. Este material es barato (silicio, las playas están repletas de él!) pero el proceso es muy caro, hay muy pocas líneas de montaje de microchips en el mundo y cada microchip tiene una línea dedicada. Esa placa de material semiconductor es muy pequeña, apenas unos milímetros de lado, y lleva un encapsulado de material plástico o cerámico. Seguramente los reconocemos en los dispositivos electrónicos, el aspecto es este:



Los procesadores de las computadoras y celulares son casos especiales de estos chips, con un tamaño mayor -no mucho, solo unos milímetros más- debido a que son los que más transistores llevan para poder hacer millones de operaciones lógicas por segundo. En la foto vemos un procesador moderno: el procesador Intel i7.

Pasemos a la próxima ficha donde nos meteremos en el interminable y genial mundo de Arduino.



Introducción a Arduino

¿Qué es Arduino?

Es una **plataforma electrónica** de **software y hardware libres** cuyo componente principal es la **placa Arduino**. Esta placa permite leer información de sensores (tanto analógicos como digitales), programar su comportamiento mediante un lenguaje de programación propio y controlar diferentes dispositivos electrónicos (como luces o motores). El que regula ese trabajo de entrada-salida es su componente principal, el **microcontrolador**.



Se puede asemejar la placa a una pequeña computadora. Es posible hacer la combinación lógica que se desee, sin tener que conectar relés o transistores, Arduino ya cuenta con una cantidad enorme de compuertas en el chip del microcontrolador y estas “conexiones” se hacen y deshacen mediante software. Genial... ¿verdad?.

¿Y para qué vimos todo lo anterior si ahora ya no es necesario?

¡Es una gran pregunta! La respuesta es: “Por dos motivos fundamentales”.

A saber:

- Es uno de los objetivos de este curso aprender los conceptos teóricos -al menos de manera básica- de todo lo que se haga. Que no sea una “caja mágica” donde se conectan componentes según un diagrama, se hace un circuito pero sin tener el conocimiento de lo que allí está sucediendo. Es importante que sepamos que dentro de la placa Arduino hay miles de transistores, que estos transistores se agrupan como compuertas AND, OR y NOT y que combinando estas compuertas mediante un programa se hace que la placa se comporte como se necesite para resolver una situación problemática. No es magia, es razonamiento, pensamiento lógico e implementación tecnológica.
- No obstante la miniaturización. Se siguen utilizando relés y transistores por afuera de la placa Arduino, también resistencias y diodos. Entonces es menester que tengamos idea de cómo se complementa la lógica de la placa.

La placa Arduino

FICHA 6: Introducción a Arduino
Conociendo una placa Arduino

En realidad no hay “una” placa Arduino sino que hay muchos modelos de la misma. Todas tienen el mismo principio pero poseen más o menos entradas, salidas, espacio de memoria, microcontrolador, son más grandes, más pequeñas (por ejemplo para poder automatizar prendas de vestir!), etc. Y como es hardware libre hay muchos fabricantes que toman las placas Arduino y hacen sus propias versiones. Según la **filosofía del software y el hardware**

libre tenemos el derecho de **saber y estudiar cómo algo está construido**, de **hacer nuestra propias versiones** y de **distribuir esos cambios** para que todo el mundo se beneficie y sea una idea colaborativa y abierta. Es lo contrario al modelo de patentes y registros donde una idea o un diseño es propiedad de una compañía y ni siquiera podemos saber cómo ha sido concebida, mucho menos poder adaptarla o modificarla.

¿SABÍAS QUÉ...?

El proyecto Arduino nació en Italia y existe mucha información y documentación en su sitio oficial: <https://www.arduino.cc>

Nosotros usaremos una placa llamada **Arduino Uno** que es una de las más comunes dentro de la familia Arduino. En la siguiente figura vemos las partes principales de la misma:

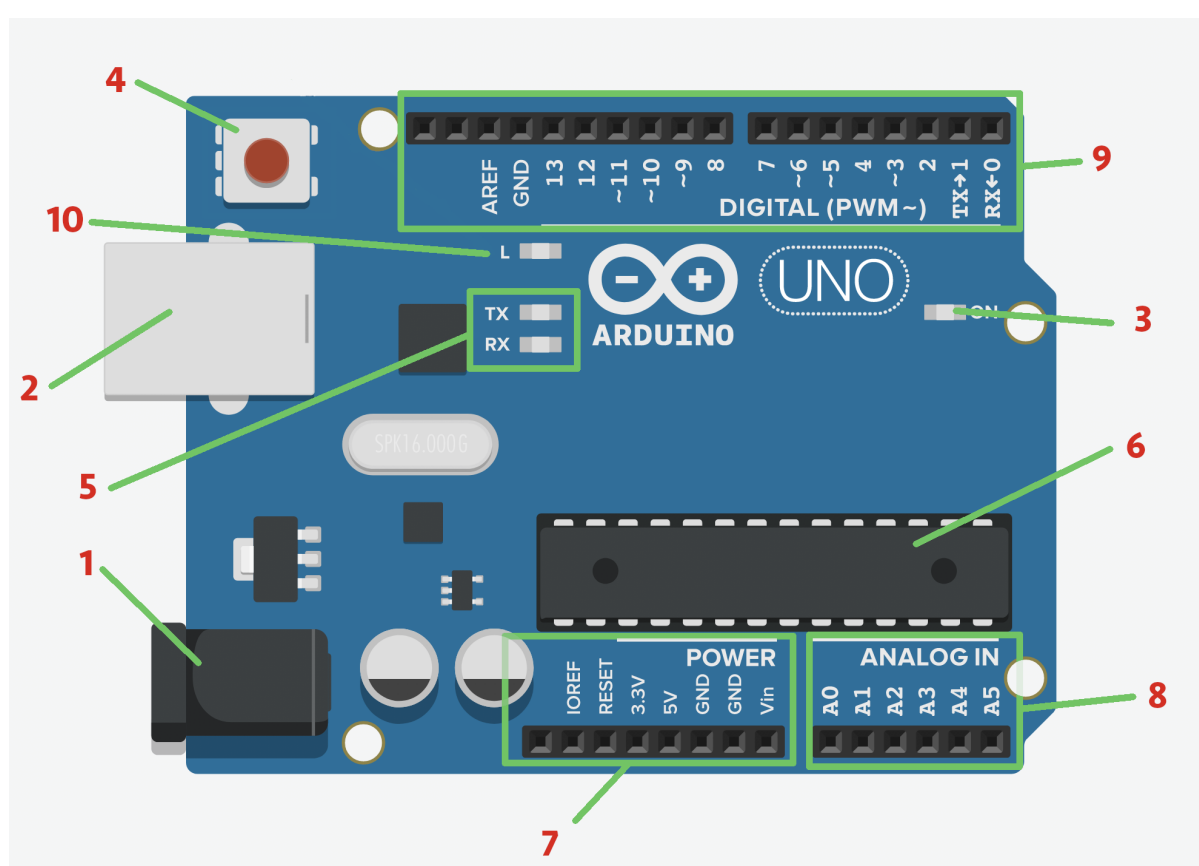


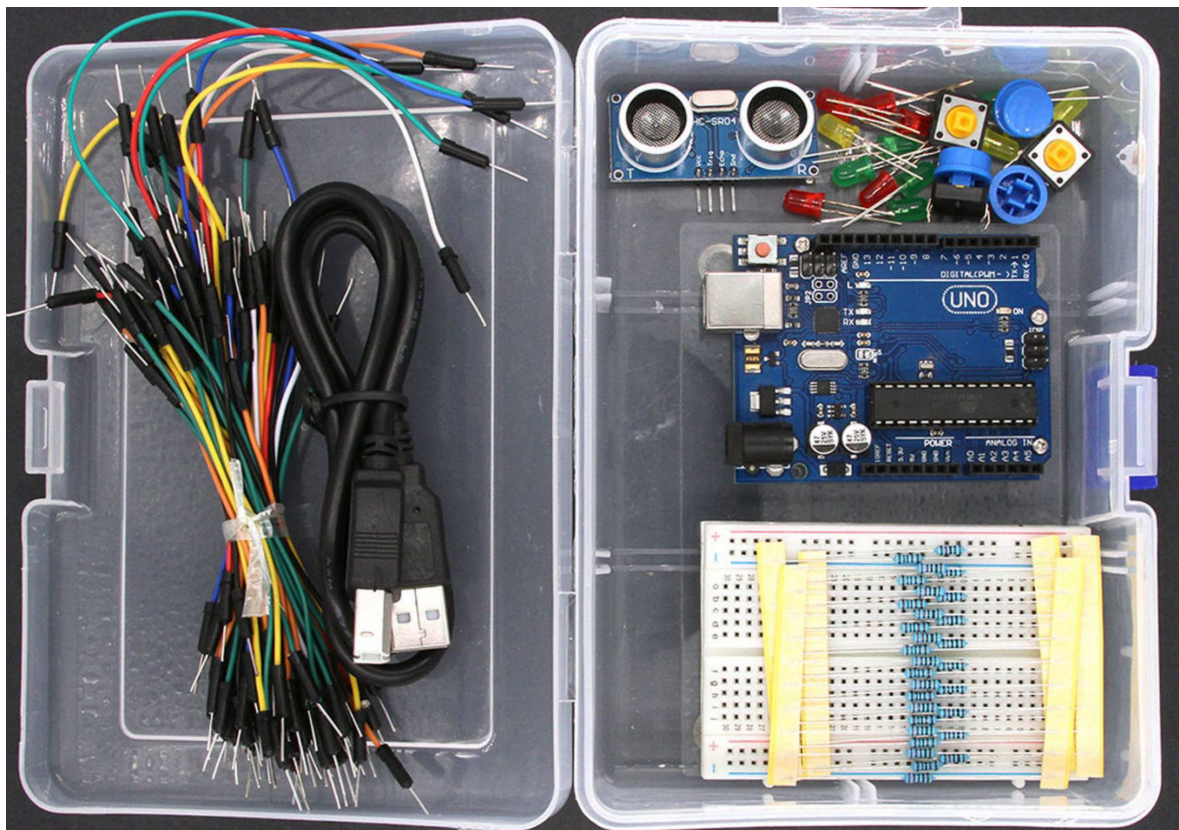
FIGURA: Placa Arduino Uno

REFERENCIAS

1. **Conector** para la alimentación. La placa se alimenta con una tensión entre 7 y 12 Volts.
2. **Conector USB** para cargar el programa (llamado "**sketch**") desde la computadora. También puede usarse para alimentar la placa.
3. LED testigo de **encendido** de la unidad.
4. Botón pulsador de **Reset**. Como su nombre lo indica, reinicia la placa y se usa en caso de cuelgues, fallas, etc.

5. Estos dos LEDs indican **comunicación** entre la placa y la computadora, cuando transmitís datos (por ejemplo, cargar el sketch) los verás parpadear acompañando la actividad.
6. **Microcontrolador** ATmega 328
7. Pines de **alimentación**. Hay pines de +5V, +3, 3V, - (GND), para sacar de allí alimentación hacia otros componentes fuera de la placa.
8. 6 **entradas** con denominación desde A0 a A5 para conectar sensores analógicos.
9. **Pines programables** para usar como entradas o salidas digitales. Los que tienen el símbolo ~ pueden usarse también como salidas analógicas.
10. **LED integrado**, conectado internamente al Pin 13. Se usa mucho para verificar procesos.

KIT ARDUINO

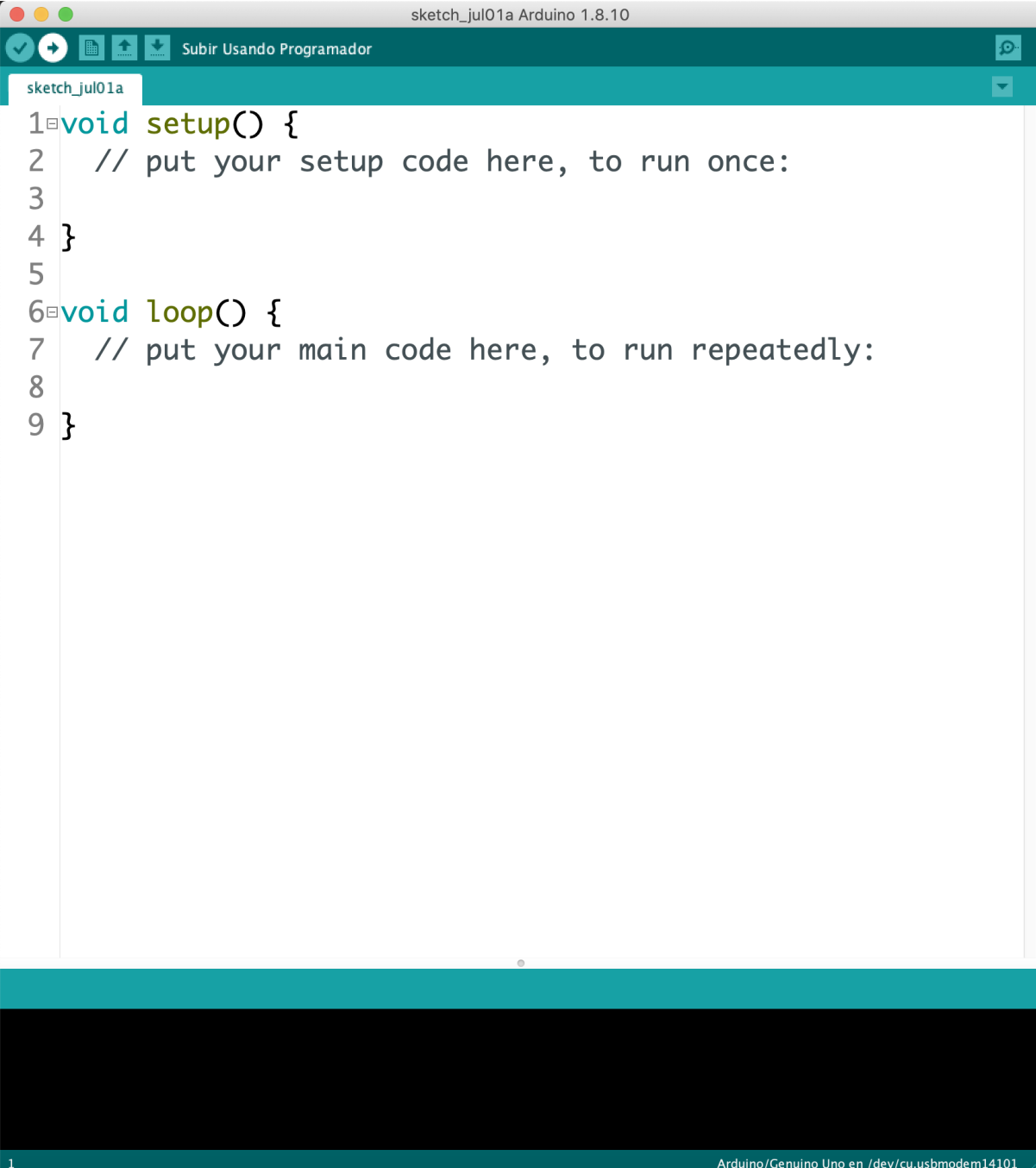


Las placas Arduino suelen venderse en forma de Kits que incluyen además un protoboard, microswitches, pulsadores, relés, resistencias, transistores, sensores, portapilas, cables de conexión, etc. Con estos Kits se pueden hacer todos los proyectos directamente en la placa aunque se sugiere hacerlos primero en el simulador y luego pasar al protoboard. No son particularmente caros y permiten emprender proyectos más complejos. La misma placa Arduino Uno se usa en innumerables funciones industriales, artísticas, tecnológicas, de educación, etc.

Programación y carga del sketch en la placa Arduino real

Como se dijo, Arduino tiene su propio lenguaje de programación. Este lenguaje es textual y tiene un **IDE (Integrated Development Environment, Entorno de Desarrollo Integrado)** que se puede descargar libre y gratuitamente de su [sitio oficial](#).

El IDE tiene una pequeña botonera superior con las funciones principales y un gran lugar para escribir el código:



```
sketch_jul01a Arduino 1.8.10
Subir Usando Programador
sketch_jul01a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

1 Arduino/Genuino Uno en /dev/cu.usbmodem14101

FIGURA: Entorno de desarrollo para Arduino

Download the Arduino IDE



ARDUINO 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows 7 and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
 Get 

Mac OS X 10.10 or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

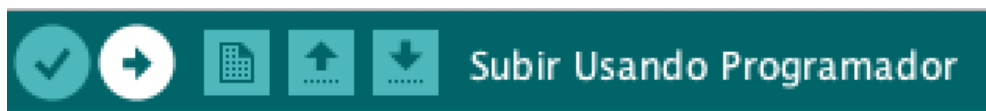
Release Notes
 Source Code
 Checksums (sha512)

Hay versiones para Linux, Mac y Windows.

¿Cómo se lleva a cabo un proyecto en Arduino?

La dinámica para hacer un proyecto en Arduino es la siguiente:

- Se identifica un **problema** que tiene una solución computacional.
- Se piensa y desarrolla una **estrategia** para solucionar esa situación problemática
- Se **programa en la computadora** con el IDE oficial o con alguna alternativa. (Recordemos que es software libre, hay muchas variantes para programar a Arduino).
- Se enciende la placa Arduino alimentándola a través del conector (1). Deberá encenderse el LED (3). Se conecta el cable USB (2) para pasar los datos a la placa. Como el cable USB también provee alimentación es posible usar solamente este último. La placa ahora está **vinculada** al IDE
- Se carga el programa (sketch) usando el botón del menú y a través del **cable USB** (2). Los LEDs (5) titilarán y indicando que todo ha sido satisfactorio.

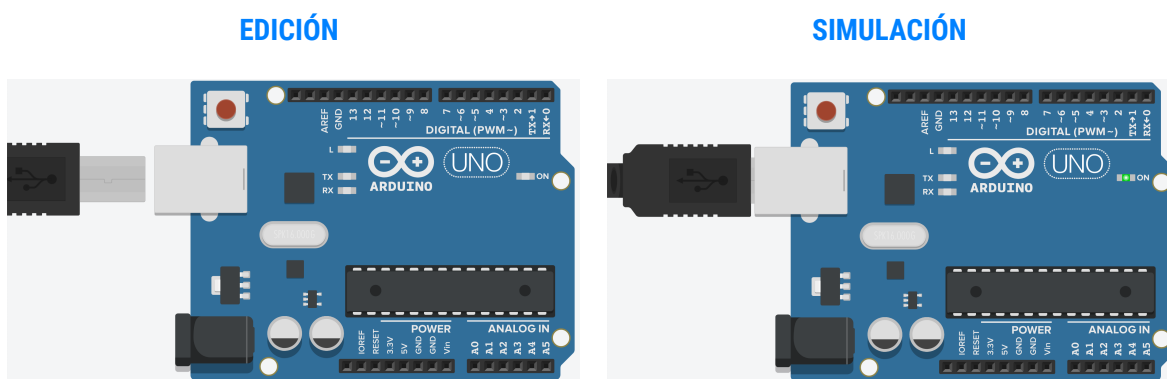


- El programa arrancará inmediatamente
- Si la placa está alimentada por pilas a través del conector (1) es posible desconectar el cable USB y la placa funcionará autónomamente con el sketch cargado en su **memoria**. Si, en cambio, está alimentada a través del USB (2) se debe mantener conectado.
- Para hacer un cambio en el programa, hay que repetir el proceso de carga y así actualizar el sketch residente en la placa.

Puede parecer engorroso al principio pero es un proceso simple y que se incorpora muy intuitivamente.

Programación y carga de Arduino en simulador virtual

Nuestro entorno de [simulación virtual](#) también puede simular Arduino, por lo que si no tenemos a mano una placa, es totalmente posible hacer los desafíos de manera virtual. Simplemente se busca en la sección **"Microcontroladores"** del menú y se arrastra una placa Arduino Uno, que aparecerá con el cable USB desconectado. Al comenzar la simulación el sistema "conectará" el USB, se pondrá verde el LED de encendido(3), cargará el sketch y comenzará el programa. Si querés hacer cambios, tanto en el circuito como en la programación, es importante que estés en modo "Edición".

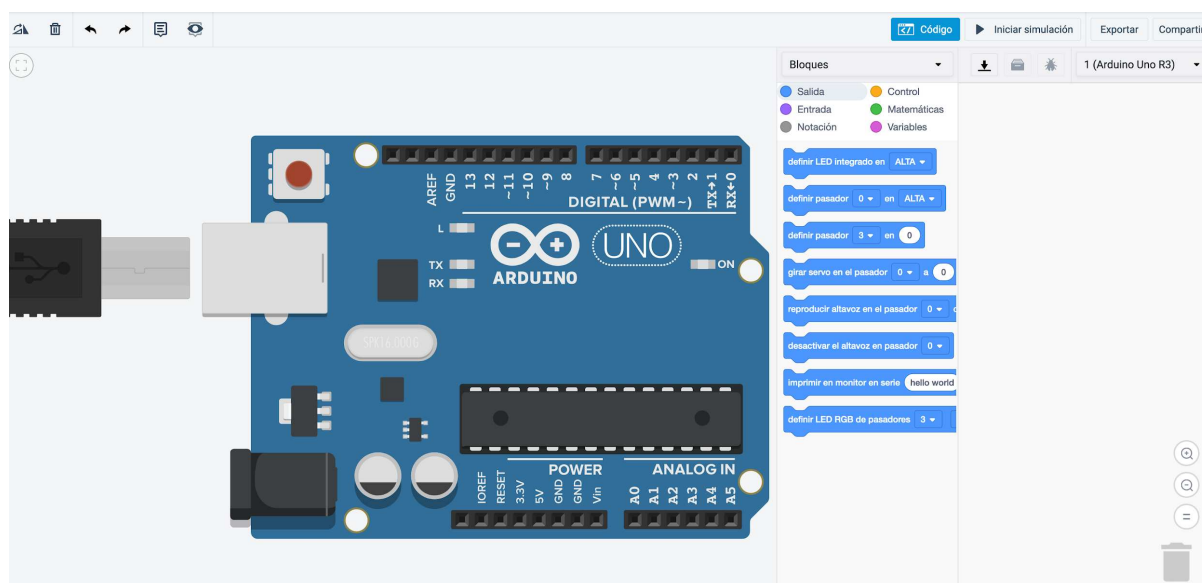


La placa "desenchufada" en el modo *Edición* y "enchufada" durante la *Simulación*.

Para programar la placa, se usará un entorno propio basado en bloques al que se accede presionando el ícono **"Código"** que se encuentra sobre el menú de selección de componentes.

 Código

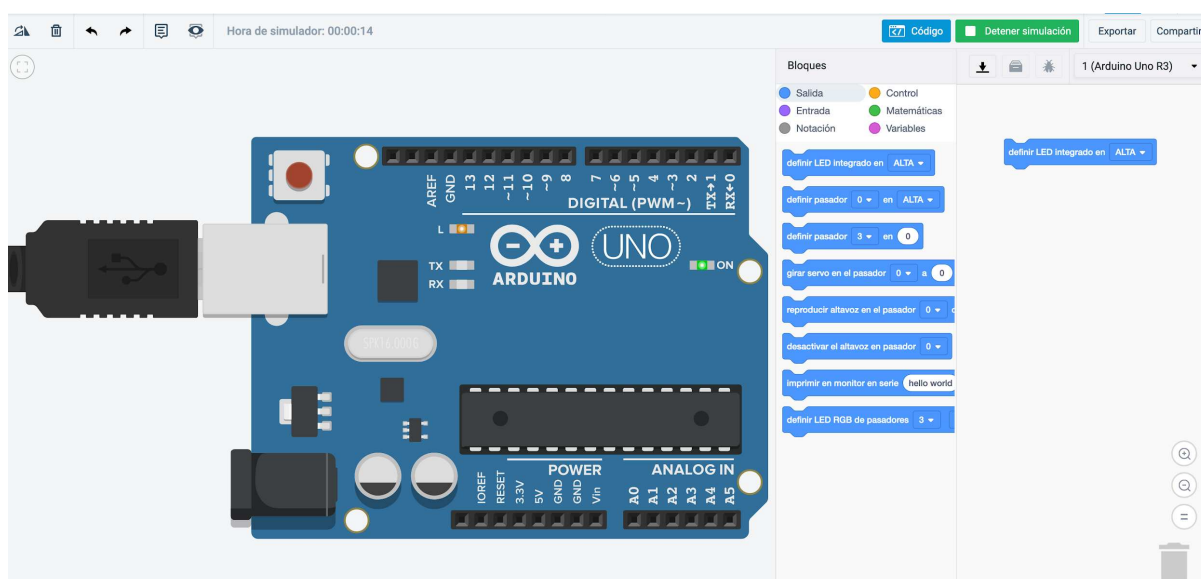
Al presionar este botón, se abrirá el entorno de programación Arduino de Tinkercad.



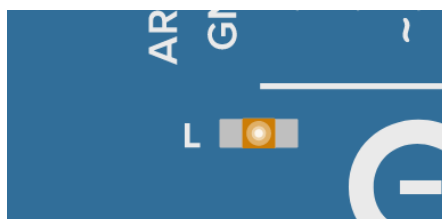
Observemos algunas cosas interesantes. Arrastramos primero el bloque “**definir LED integrado en ALTA**”, de la categoría “**Salida**”, al área de trabajo.



“**LED integrado**” es el LED (que señalamos con el número 10 cuando reconocimos las partes de una placa Arduino). Ahora presionamos el botón para **Iniciar simulación**. Observemos qué pasa con ese LED.



¡Encendió! ¡Felicitaciones! Hemos completado la primera experiencia Arduino.



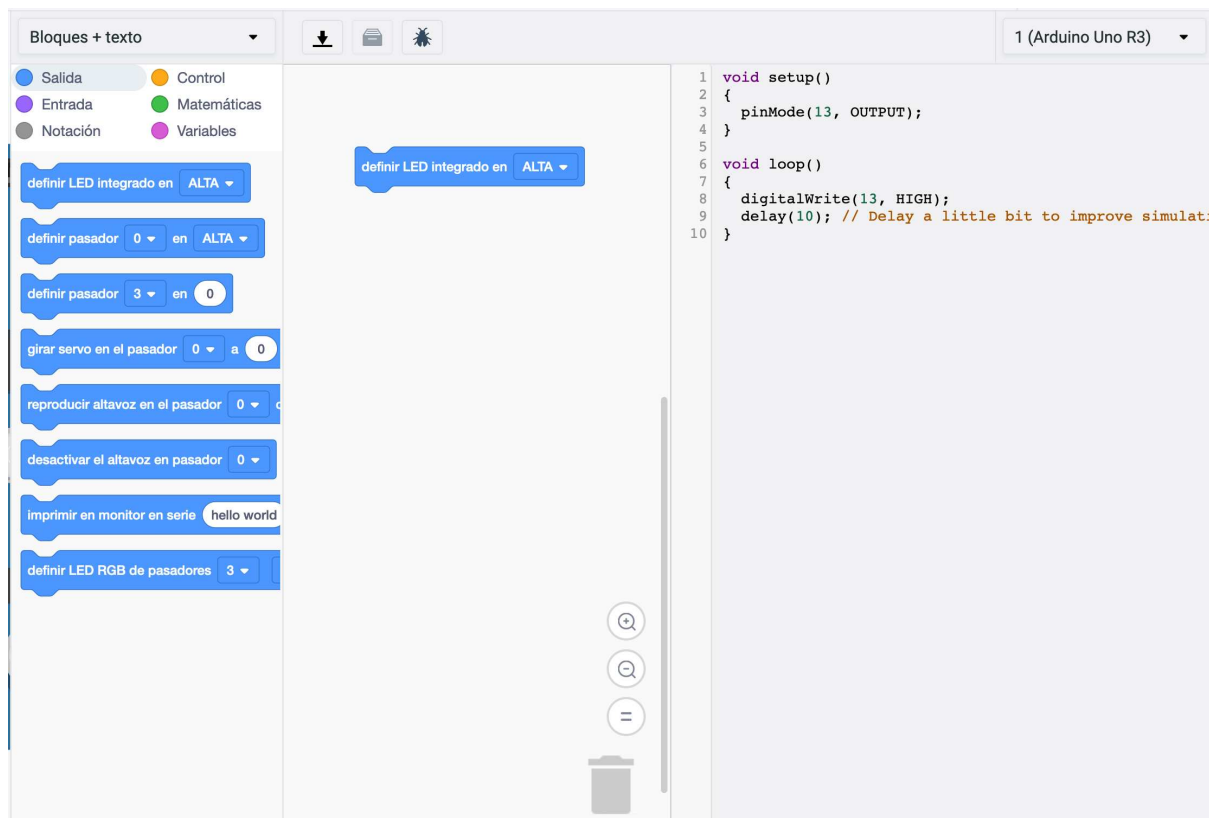
Detenemos la simulación con el botón verde y hacemos clic en el desplegable con la etiqueta “**Bloques**”

Bloques

Seleccionamos “**Bloques + texto**”

Bloques + texto

Y allí tenemos un nuevo universo...

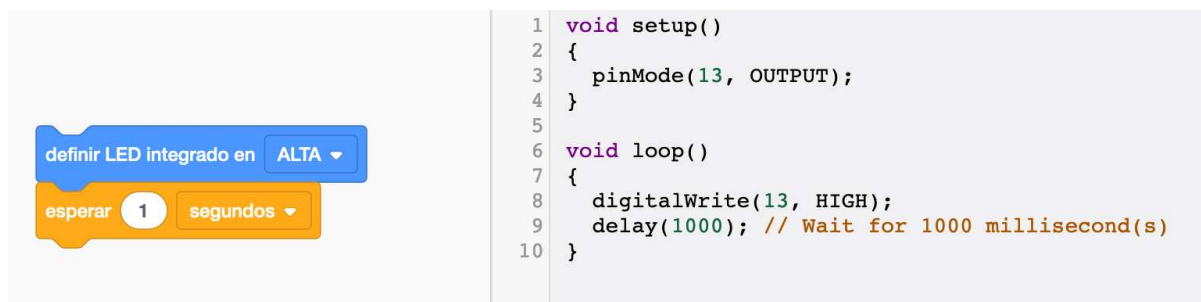


The screenshot shows the Arduino IDE interface. On the left, there is a 'Bloques + texto' menu with categories: Salida (blue), Entrada (purple), Notación (grey), Control (orange), Matemáticas (green), and Variables (pink). Below the menu, several blue blocks are visible: 'definir LED integrado en ALTA', 'definir pasador 0 en ALTA', 'definir pasador 3 en 0', 'girar servo en el pasador 0 a 0', 'reproducir altavoz en el pasador 0', 'desactivar el altavoz en pasador 0', 'imprimir en monitor en serie hello world', and 'definir LED RGB de pasadores 3'. The main workspace contains one 'definir LED integrado en ALTA' block. On the right, the text editor shows the following code:

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(10); // Delay a little bit to improve simulating
10 }
```

Se habilitará un sector más donde podrás ver directamente el código textual propio de Arduino.

Vamos al menú **“Control”** de los bloques y agregamos el bloque **“Esperar 1 segundo”** a continuación del que teníamos. ¿Qué sucede con el código del programa? El código cambia en tiempo real! Esto es sumamente útil y nos permitirá trabajar con comodidad y rapidez en los proyectos.



The screenshot shows the Arduino IDE interface with the 'esperar 1 segundos' block added to the workspace. The text editor shows the updated code:

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10 }
```



Es posible guardar el proyecto en cualquier momento como archivo **.ino**, compatible con el IDE oficial.

El trabajar con código textual es algo muy práctico pero conviene usarlo cuando ya tengamos incorporadas tareas y conceptos fundamentales. Mientras tanto, es conveniente trabajar solo con los bloques. Lo más importante por ahora es comprender la técnica y la lógica, no la sintaxis de un lenguaje de programación.

Arduino es **cíclico**: una vez que se corra el programa, se ejecutará en loop eternamente... o por lo menos hasta que se detenga la simulación.

Primer desafío

FICHA 6: Introducción a Arduino Primer desafío

El primer desafío propone construir un programa que haga titilar el LED integrado. Una solución posible es la siguiente:



Llegamos hasta aquí comprendiendo las bases de la automatización. Recorrimos el sendero desde el circuito simple hasta los rudimentos de la placa Arduino. Llegó el momento de colocar **sensores** y **actuadores** (palabra nueva) a nuestro Arduino para construir proyectos más realistas e interesantes.

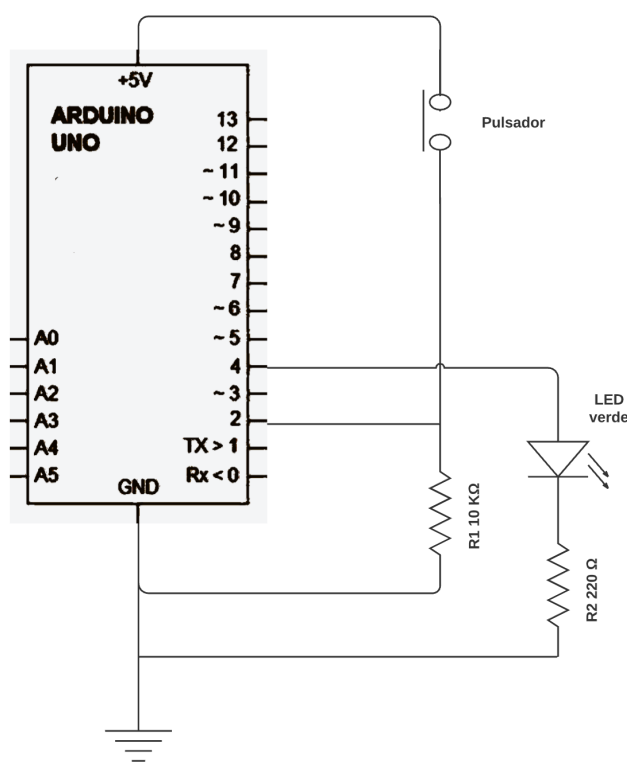
Activación de un LED a través de Arduino

Demos un paso más: usemos una **entrada** y una **salida** de todas las que dispone la placa Arduino. Como entrada usaremos un pulsador y a la salida se conectará un LED.

El pulsador de entrada tiene -lógicamente- dos **estados** posibles: presionado y sin presionar. Estos estados son 1 (o "Alta") y 0 (o "Baja"), respectivamente. Cuando no hay tensión se toma como 0 y cuando se aplica 5 V a una entrada pasa a estado 1.

Denominaciones de estados de entrada digital			
Circuito cerrado	1	ALTA	5 V
Circuito abierto	0	BAJA	0 V

Por medio del pulsador enviaremos 0 o 1 a una entrada, y cuando tengamos un 1 se programará la placa Arduino para que encienda un LED. Se puede hacer de muchas maneras, aquí tenemos una opción:

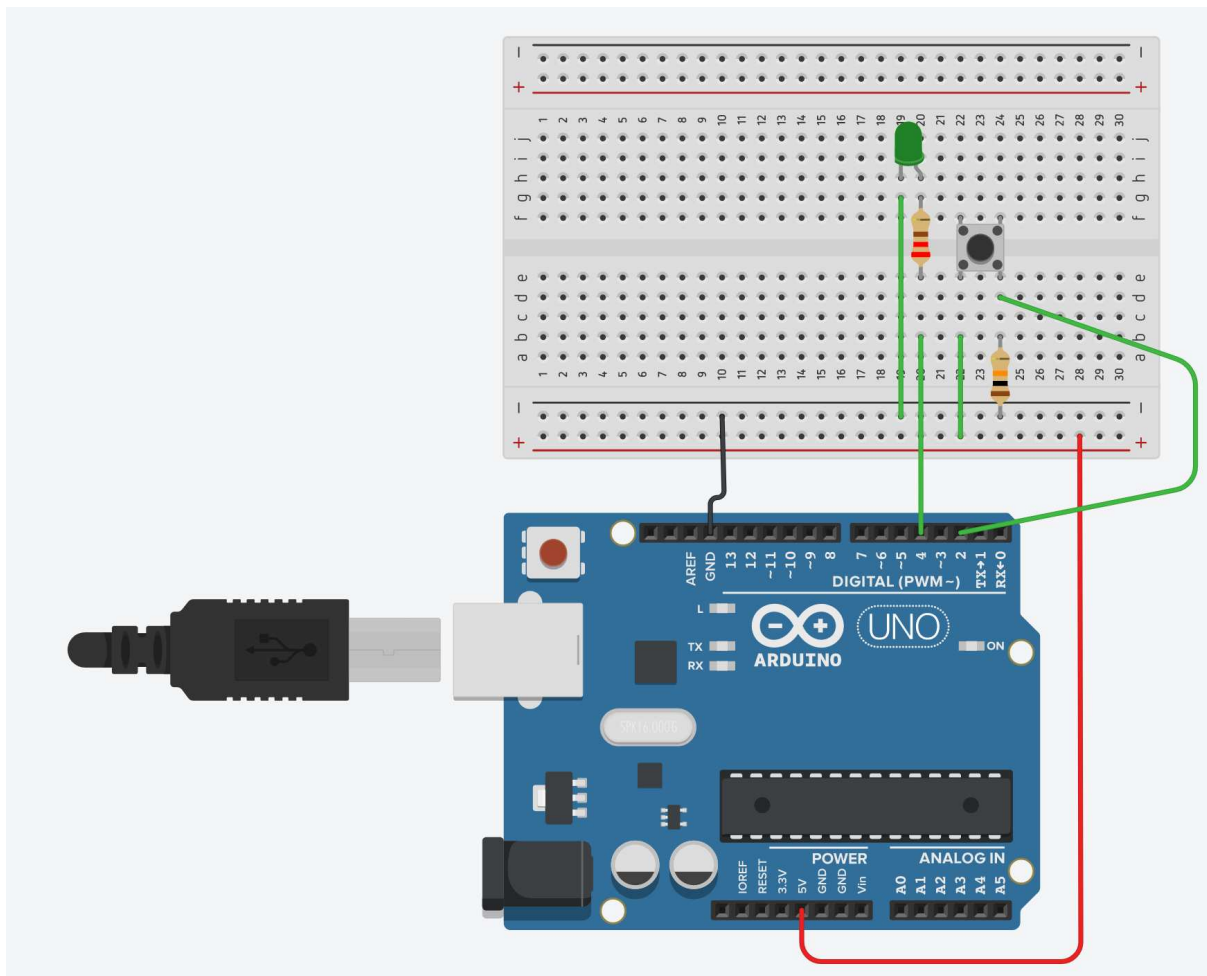


Analicemos:

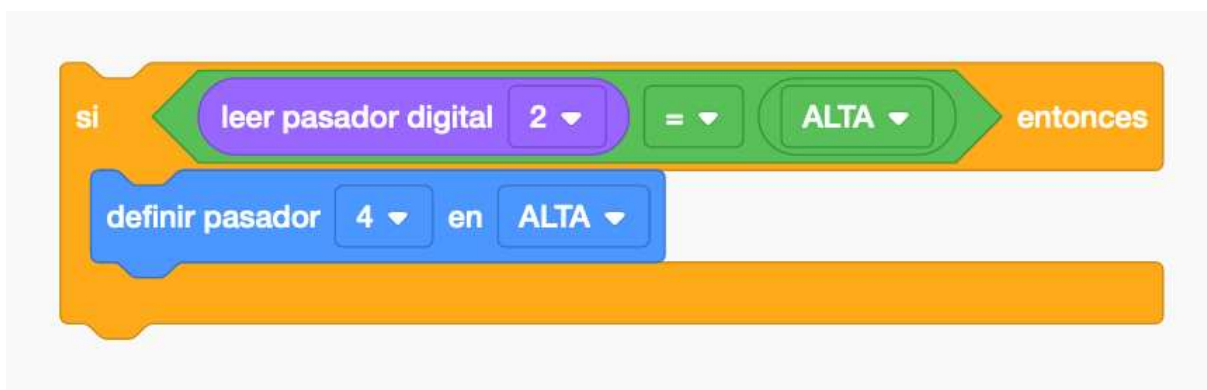
- Un pulsador NA que al presionarlo pone en estado 1 a la entrada 2 de la placa. Este pulsador toma +5V de la misma placa Arduino.
- Tiene en serie una resistencia de 10 KΩ (llamada R1) para cerrar el circuito de comando a través del terminal GND (de *Ground*, "Tierra" en inglés) de la placa.
- La salida se ha hecho en el terminal 4 de la placa.

- El LED se conectó a través de una resistencia de 220 Ω (llamada R2). Recordemos que el LED trabaja con 2V y la salida de la placa tiene 5V.

Una posible armado y programación del circuito es el siguiente:



¿Cómo podemos preguntar, desde nuestro programa, “si el botón fue presionado”? ¿A qué estado corresponde “estar presionado”? ¿Existe algún bloque que nos permita hacernos este tipo de preguntas en nuestros programas? Y aquí tenemos por primera vez el uso del bloque “Si ... entonces”.



Esto se llama **alternativa condicional** y es fundamental para el uso de cualquier tipo de sensores. Este bloque amarillo verifica si se cumple una *condición*, en caso de que así fuese ejecuta la/s instrucciones que contenga en su interior.

En este caso se pregunta si la lectura del Pin 2 está en ALTA (estado 1). Es decir, es la manera de preguntar si el pulsador está presionado. Si está presionado estará en ALTA porque le llevará los +5V al terminal 2. Si no está presionado tendrá 0V, por lo tanto estará en BAJA (estado 0).

Cuando se presione el pulsador la condición será verdadera y se ejecutará la acción que pone el Pin 4 en ALTA, dándole tensión al LED verde y encendiendolo.

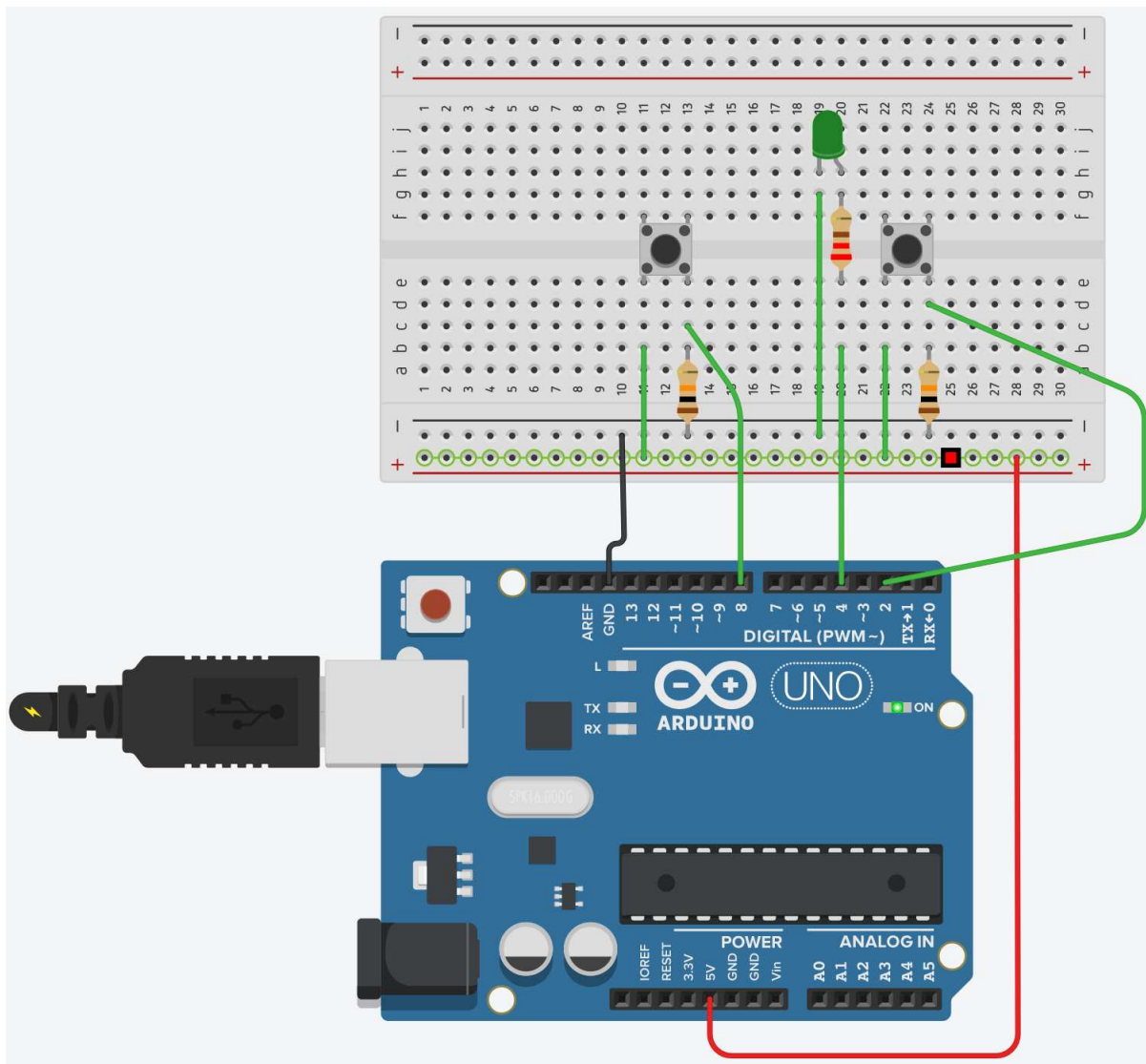
El LED quedará encendido por siempre, ya que el Pin 4 está en ALTA y nunca le dijimos que lo vuelva a BAJA.

Segundo desafío

El próximo desafío consiste en agregar un nuevo pulsador, conectarlo a la placa Arduino y modificar el programa para que, al presionarlo, se apague el LED.

FICHA 6: Introducción a Arduino
Segundo desafío

Solución del circuito



Solución de la programación



Como solución, agregamos otra alternativa condicional que en este caso está continuamente leyendo el estado del Pin 8. Si se presiona el pulsador este Pin estará en ALTA y se cumplirá la condición, por lo tanto pasará a BAJA el Pin 4, apagando el LED.

Recordá que las instrucciones en Arduino están en una repetición ("loop") eterno. Imaginate que luego de la última instrucción vuelve a la primera y así sucesivamente. En el ejemplo anterior, si pulsamos los botones, las veces que quieras y el LED se encenderá y apagará en múltiples oportunidades.

Esto se ve más claro en el lenguaje Arduino textual

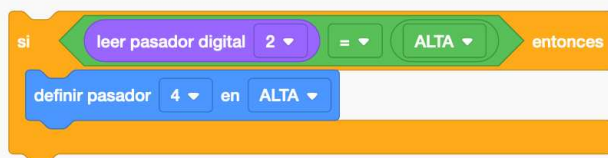
```
1 void setup()  
2 {  
3  
4 }  
5  
6 void loop()  
7 {  
8     delay(10);  
9 }
```

Todo lo que vaya entre los corchetes de las líneas 2 y 4 se ejecutará **una sola vez, al principio de todo**. Se llama **setup** y allí pondremos la configuración inicial que necesitemos hacer, por ejemplo cuando asignamos un Pin digital como entrada o salida, se coloca ahí.

Todo lo que va entre los corchetes de las líneas 7 y 9 se ejecuta **en un loop infinito**.

Aquí vemos la implementación de la primera parte del desafío anterior, y podemos identificar qué

pasa en cada momento.



```
1 void setup()
2 {
3   pinMode(2, INPUT);
4   pinMode(4, OUTPUT);
5 }
6
7 void loop()
8 {
9   if (digitalRead(2) == HIGH) {
10    digitalWrite(4, HIGH);
11  }
12  delay(10); // Delay a little bi
13 }
```

Tercer desafío

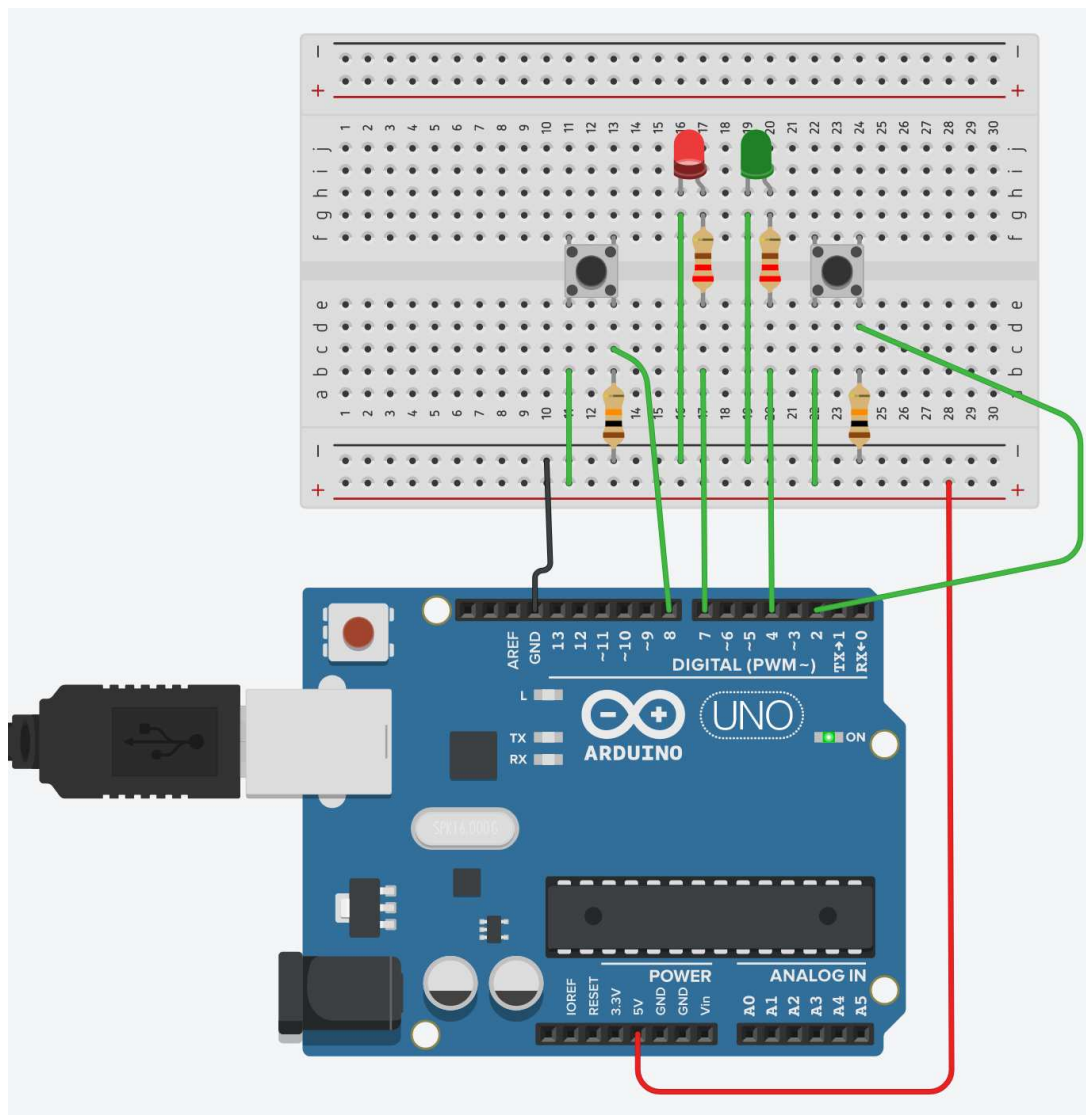
FICHA 6: Introducción a Arduino Tercer desafío

Agregar un nuevo led y cambiar el comportamiento de los pulsadores.

Agregá un segundo LED (rojo) de manera que:

- Cuando aprietes un pulsador se prenda el LED verde y se apague el rojo.
- Cuando aprietes el otro pulsador se prenda el LED rojo y se apague el verde.

Solución para el circuito



Solución de la programación

```
si leer pasador digital 2 = ALTA entonces
  definir pasador 4 en ALTA
  definir pasador 7 en BAJA
si leer pasador digital 8 = ALTA entonces
  definir pasador 4 en BAJA
  definir pasador 7 en ALTA
```

No hace falta aprender el lenguaje textual Arduino, todos los ejemplos los haremos usando bloques. Pero sí es bueno ir espiando lo que sucede tras los bloques, al menos en esta etapa inicial, para comprender cómo funciona nuestra placa y así poder programarla usando todas sus posibilidades.

FICHA 6: Introducción a Arduino
Comprendiendo código

Proyecto Final

Conociendo los sensores analógicos

En el capítulo anterior usamos pulsadores y también sensores (analógicos y digitales) para poner un "1" en las entradas. Tanto los pulsadores como los microswitches y otros sensores puntuales nos ofrecen dos estados inequívocos. Un pulsador está presionado o no, no hay estados intermedios. En cambio un sensor de temperatura, por ejemplo, no tiene un estado taxativo sino que maneja un rango de valores. Lo que hace Arduino es dividir en 1024 partes su rango de trabajo que sabemos que es entre 0 y 5V. Entonces el valor de 0V es "0" y el valor de 5V corresponde al "1023" de Arduino. Supongamos que el sensor le devuelve 2,5V (el punto medio) para Arduino eso será "512".

Este manejo de entradas analógicas nos permite, al igual que cuando construimos los circuitos con relés, trabajar en Arduino con sensores analógicos (como el sensor de temperatura que utilizamos para encender la luz de alerta cuando se trababa el motor). Entonces, también podemos usar Arduino para que un horno corte a una determinada temperatura, que se dispare una alarma al haber cierta cantidad de humo o que se activen los regadores al registrar una humedad mínima en una plantación. Es con estos sensores que se le saca el mayor jugo a la automatización de Arduino.

Aplicaremos este concepto -y todo lo trabajado- en un proyecto completo para nuestra habitación. Una lámpara para la mesa de luz o escritorio con las siguientes funcionalidades:

- Un botón para encender la luz inmediatamente.
- Un botón para apagarla.
- Un sensor de temperatura que a determinado nivel de calor enciende un ventilador personal.
- Un pulsador que enciende la luz para leer pero que a los 5' se apaga sola por si el lector se duerme en ese lapso.

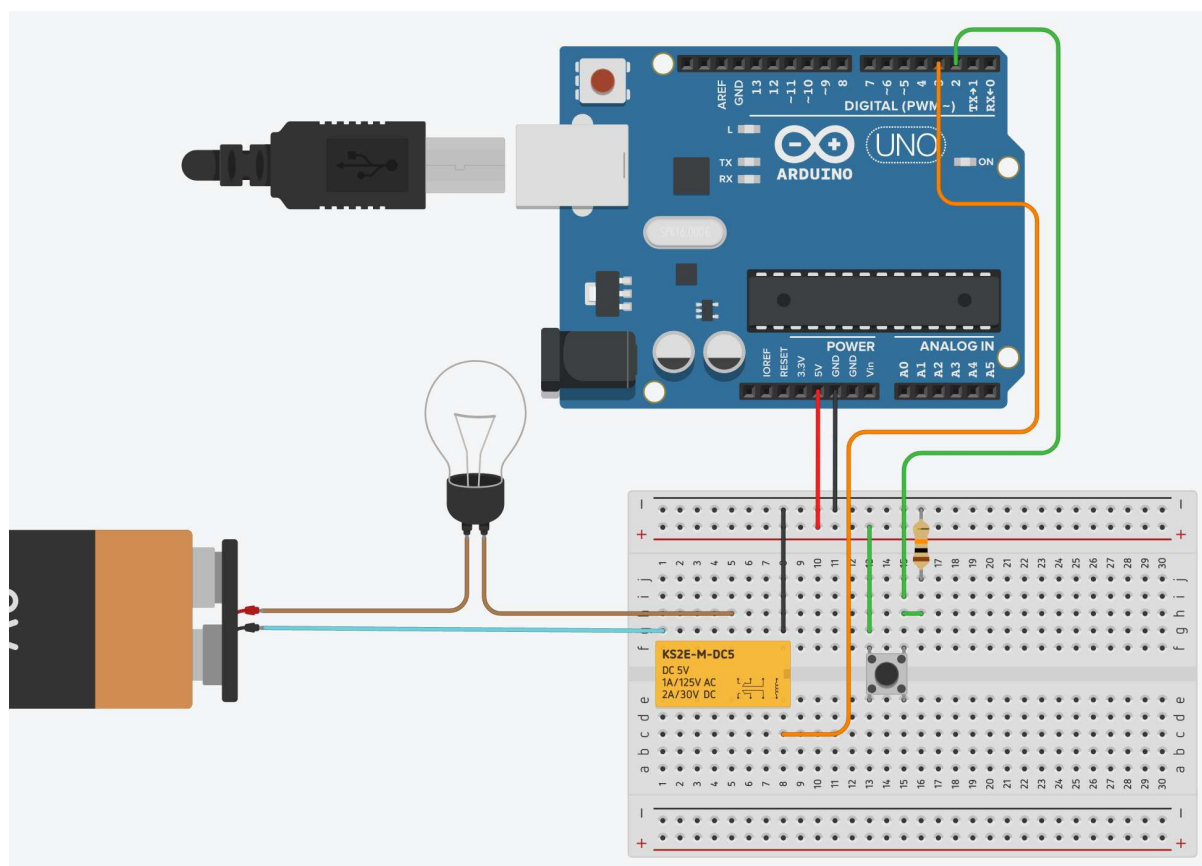
Armemos nuestro proyecto final

Como regla general, tomemos una tarea a la vez y resolvámosla. Luego pasemos a otra y así sucesivamente. Ser ordenados en este punto nos permitirá hacer un código prolijo, que permita escalarlo (mejorarlo o ampliarlo) más adelante, y también nos facilitará la de por sí engorrosa acción de encontrar y solucionar errores.

Un botón para encender la luz inmediatamente

Es necesario para esta tarea un pulsador, el relé y una resistencia de 10 KΩ para el circuito de comando; y para el circuito de potencia necesitamos una batería de 9V y una lámpara de esa tensión.

Una manera de conectar esta etapa es la siguiente:



Y un posible programa:



Analicemos:

- Con colores **marrón y celeste** se representan el **+9V** y el **-9V** (circuito de potencia)
- Con colores **rojo y negro** se representan el **+5V** y el **-5V** que viene de la placa Arduino y sirve para manejar el relé.
- El cable **naranja** que viene del Pin 3 es el que manejará la activación del mismo según la programación mostrada en los bloques.
- Con color **verde** se muestra el circuito de comando del botón pulsador de encendido. Como se ve en la programación, el pulsador está conectado al Pin 2 y la placa lo lee constantemente. Cuando lo detecta en ALTA (pulsador apretado) define un 1 en el Pin 3, que como se dijo activará la bobina del relé y encenderá la lámpara.

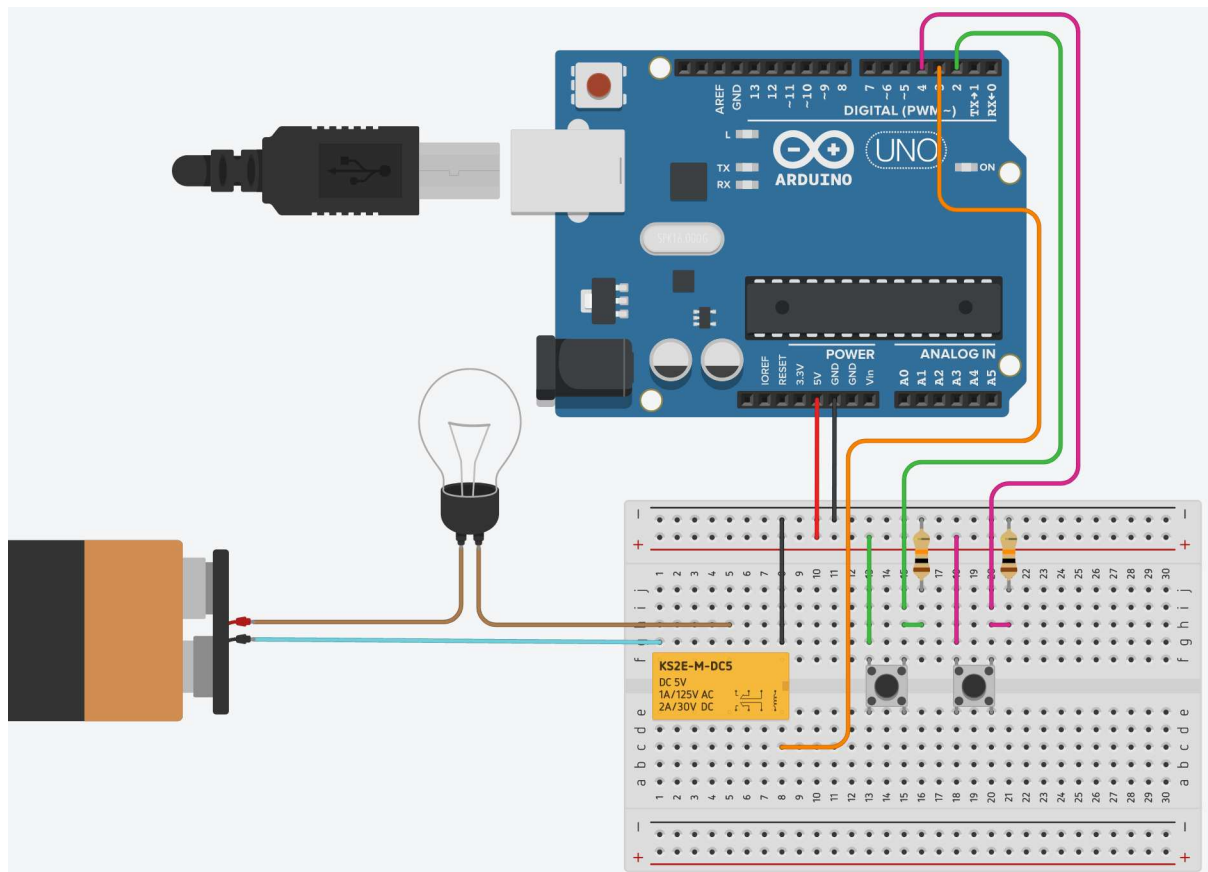
Luego de completado este paso, si se aprieta el pulsador la lámpara quedará encendida. Pero no es posible aún apagarla. Esto se hará en el próximo paso.

Puede que no se active el relé. En ese caso comprobemos la polaridad: el terminal 1 deber ir siempre al - y el terminal 16 a +5V.

Un botón para apagar la lámpara.

Como se dijo necesitamos apagar la lámpara. Una forma sería poniendo un NC en serie con la bobina del relé, pero como solo tenemos NA lo haremos usando un Pin (el 4) de la placa Arduino, que al detectar el pulsado del botón apagado ponga en 0 el Pin 3 y de esta forma quitarle la alimentación al relé para que apague la lámpara.

La parte nueva se ha agregado con color fucsia/rosa:

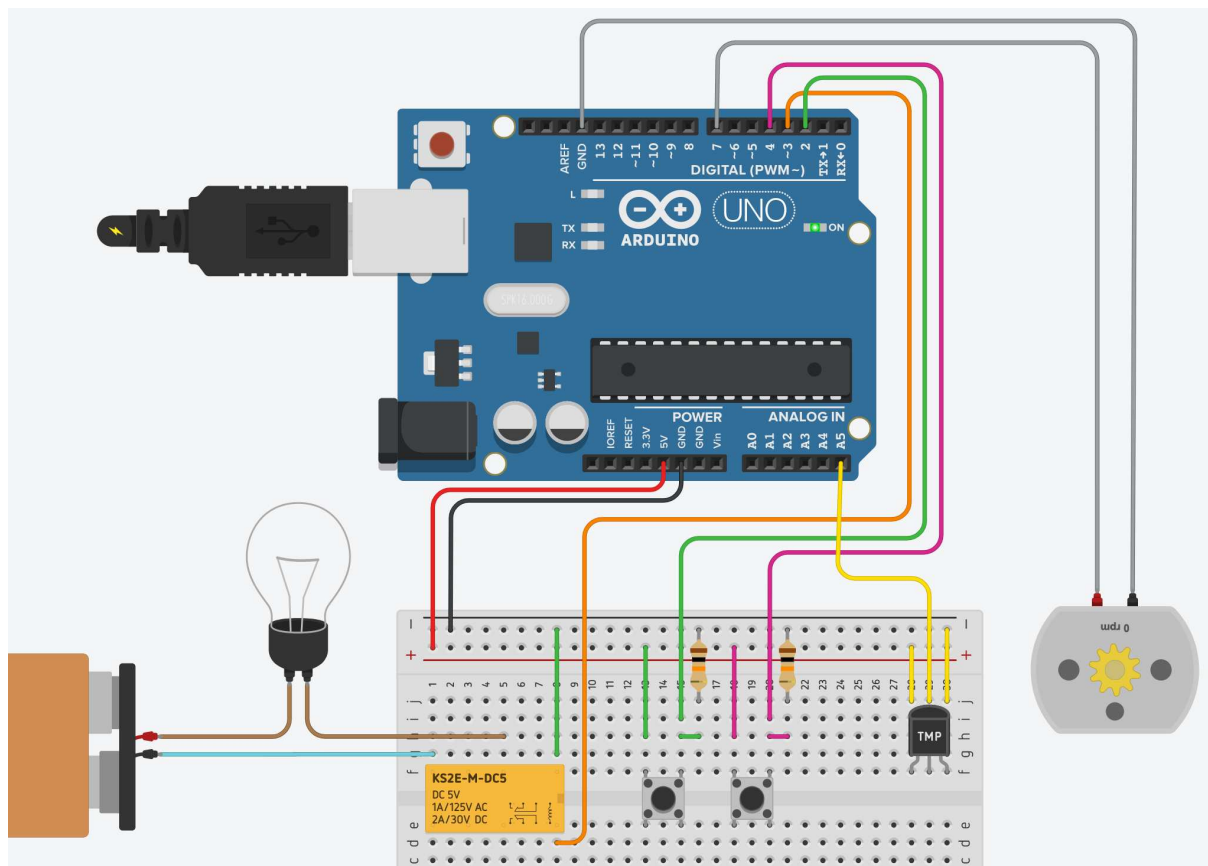


Y el programa correspondiente queda:



Un sensor de temperatura que encienda un ventilador personal

Sigamos agregándole funcionalidades a este proyecto. Esta parte trae algunas novedades: El sensor es analógico, o sea que podemos tomar valores intermedios como se ha dicho y por otro lado, un motor deberá ser alimentado al llegar a cierta temperatura.



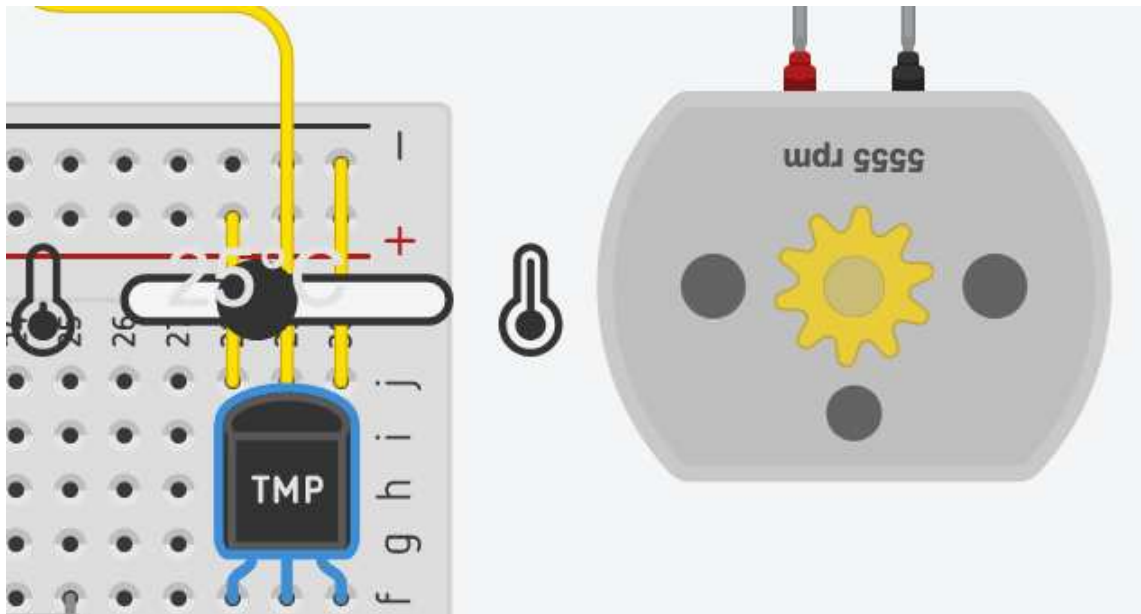
Vemos en la figura que en color gris se ha conectado un motor (ventilador) y con amarillo el sensor de temperatura. El valor de corte del sensor lo podemos variar según la temperatura en la que queremos que el ventilador actúe y corte. Para tener una referencia, si queremos que el ventilador se encienda a partir de los 24°C, debemos colocar 150 en el bloque de comparación¹. El programa es el siguiente:



Cuando el programa se va haciendo más complejo es conveniente agregar **comentarios** (en bloques color gris) para ir identificando las diferentes partes y que se puedan hacer cambios, encontrar errores más rápidamente, etc.

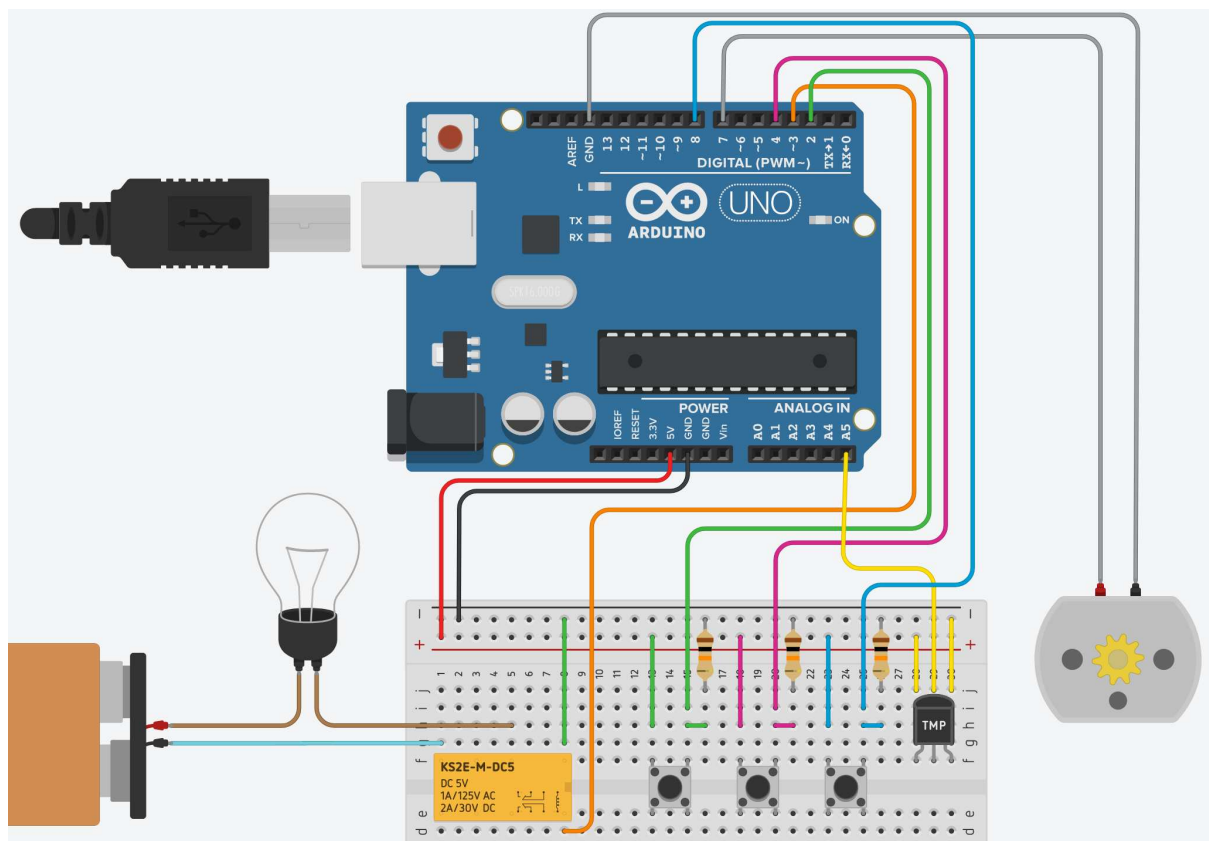
Juguemos con el sensor llevándolo a diferentes temperaturas y tomemos nota de cómo se comporta el motor según el corte que le programemos.

¹ Si se quiere profundizar al respecto, se puede buscar información sobre el funcionamiento del sensor TMP36. Una forma fácil de calcular aproximadamente el valor que debe ir en el bloque es $2 * T + 102$, donde T es la temperatura. Por ejemplo, para saber el valor correspondiente a 24°, calculamos: $2 * 24 + 102 = 150$.



Último paso, timer para dejar la luz encendida 5 minutos

Bien, debemos incluir un nuevo botón pulsador, muy parecido a los ejemplos que ya vimos, y además incorporaremos un timer digital que activa el Pin 3 por 5' y luego lo desactiva automáticamente. El circuito es el siguiente (se han puesto en azul las conexiones nuevas):



Y el programa correspondiente:

The image shows a sequence of code blocks in a Scratch-like environment, organized into four sections, each starting with a comment block:

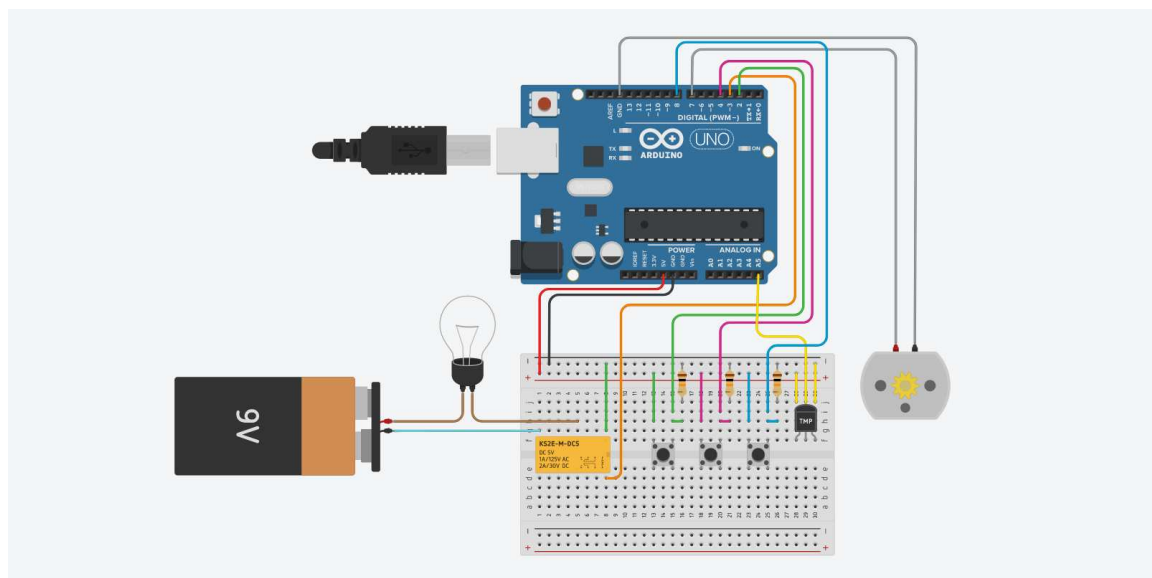
- comentario Pulsador encendido**
 - si leer pasador digital 2 = ALTA entonces
 - definir pasador 3 en ALTA
- comentario Pulsador apagado**
 - si leer pasador digital 4 = ALTA entonces
 - definir pasador 3 en BAJA
- comentario Sensor ventilador**
 - si leer pasador analógico A5 > 20 entonces
 - definir pasador 7 en ALTA
 - si no
 - definir pasador 7 en BAJA
- comentario Temporizador**
 - si leer pasador digital 8 = ALTA entonces
 - definir pasador 3 en ALTA
 - esperar 300 segundos
 - definir pasador 3 en BAJA

Donde podemos observar que el bloque que provoca el timer por software es:



Hemos puesto 300 porque es el equivalente en segundos a los cinco minutos que nos pedía la consigna.

La función **“esperar”** en Arduino hace eso: espera un tiempo determinado. En este caso, por 5’ la placa no va a realizar ninguna otra función. Si por ejemplo se aprieta el botón pulsador de apagar... no lo hará, debemos esperar a que finalicen los 300” que definimos. Esto es así porque, como está en espera, no detecta que el Pin 4 cambia de estado: está “esperando”, no “leyendo”. Esto provoca un incordio cuando empezamos a trabajar con **delay()** ya que estamos acostumbrados a que nuestros programas hagan una cosa mientras se va haciendo otra “en segundo plano”. Hay maneras de resolver este aparente déficit, pero están fuera del alcance de este curso. Una solución fácil pero “no digital” es cortar la alimentación del circuito. Si te interesa investigar soluciones digitales, [acá](#) hay un buen punto de partida. Pero tengamos en cuenta que será menester usar el IDE de Arduino para programar ya que los bloques que tiene el simulador no son suficientes para realizarlo.



No hemos puesto los diagramas de los circuitos eléctricos en este proyecto. Trabajamos directamente sobre el protoboard, ya que al dividir el proyecto en partes, cada una de las partes era igual o muy similar a algo que ya habíamos hecho. Esta también es la ventaja que te otorga la división en subtareas. No obstante... ¿Podrías hacer el diagrama eléctrico completo del proyecto?

Este curso se ha orientado a que comprendas la lógica de la automatización y dar los primeros rudimentos en Arduino. Gracias por habernos acompañado hasta aquí. Es la tarea ahora andar por caminos más ríspidos y desconocidos, investigando, probando, equivocándose y volviendo a andar. Así hemos aprendido todos. Seguramente nos encontraremos en esos senderos y podremos compartir dudas, logros y emociones.