

# Algoritmos y Estructuras de Datos II



## Práctico Arduino Iniciando en la programación de Arduino



### OBJETIVOS

- Conocer las características básicas de la placa Arduino y su entorno de desarrollo.
- Comprender conceptos simples de circuitos electrónicos utilizando leds, resistencias, botones.

### COMPETENCIAS

- Identificar, formular y resolver problemas mediante programación.
- Trabajar en equipo, distribuyendo tareas y asignando roles.
- Aprendizaje autónomo y colaborativo.

### METODOLOGÍA

- Los alumnos deberán resolver de manera grupal los ejercicios propuestos.
- Cada grupo desarrollará su propuesta de solución en la herramienta tinkercad.
- Realizar consultas a través del canal de Slack, correspondiente a su comisión, o del aula virtual de la asignatura.

### DURACIÓN

Según planificación de la asignatura se deberán utilizar para la resolución de los ejercicios de esta parte, 1 clase de teoría y 1 clase de laboratorio.

### RECURSOS

- Simulador online <https://www.tinkercad.com/>
- Computadora.



## RESISTENCIA:

Los leds que utilizaremos funcionan con aproximadamente 3 voltios, pero en nuestro circuito tenemos por lo general 5V, por lo tanto, debemos utilizar una resistencia de protección.



LED: 3,3 V y 20mA.

Si la fuente de alimentación es de 5V entonces la resistencia debe retener 1,7V.

Cuando se elige una resistencia hay que tener en cuenta su

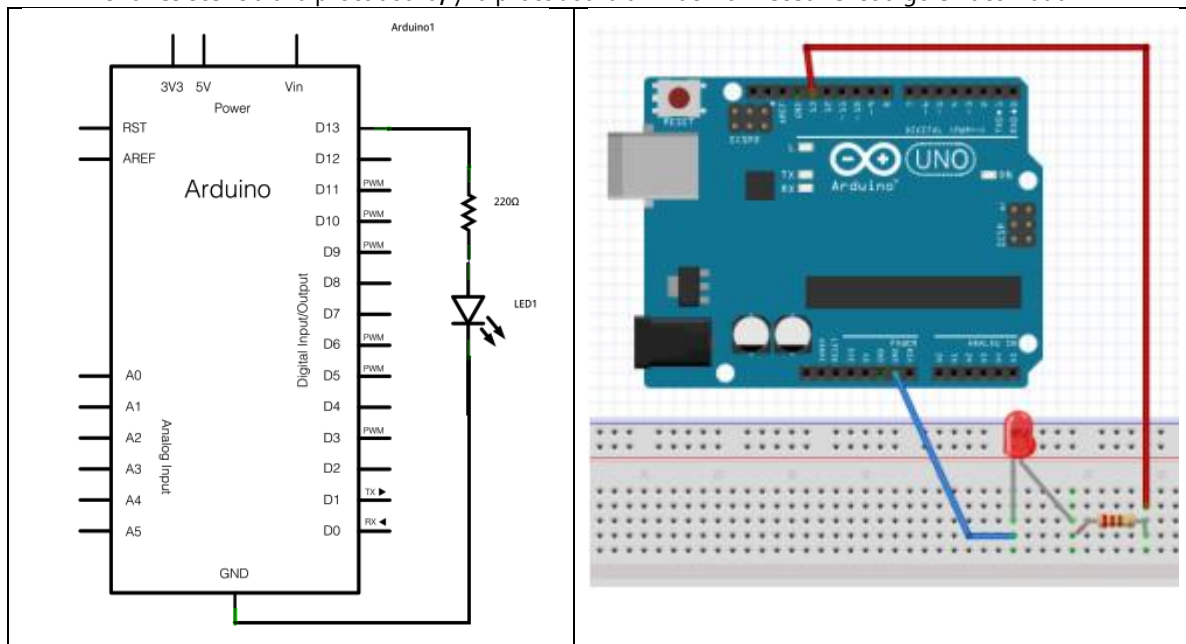
- $R=V/I$ :  $R= (5V-3,3V)/20mA= 1,7V/0,02A= 85 \text{ Ohms}$
- Potencia (cantidad de watts que debe tolerar la resistencia):  $P=V*I$  (Potencia en watts = Voltaje en Voltios \* Corriente en amperios):  $5V*0,02A=0,1W$  Entonces mi resistencia puede ser de 1/8 de watts, es decir 0,125W

### Actividad 1.1: Blink en Arduino: (T: 10')

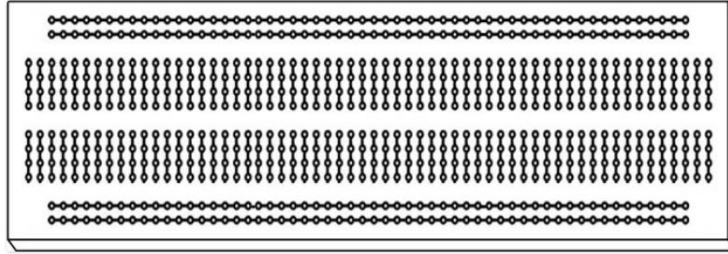
- Acceder al proyecto: <https://www.tinkercad.com/embed/d7jltY2Ccgw?editbtn=1> revisar el código y ejecutar la simulación

### Actividad 1.2: Blink con led externo y resistencia:

- Crear un nuevo circuito en tinkercad. Agregar la placa Arduino UNO, una protoboard, conectar un led y una resistencia a la protoboard, y la protoboard al Arduino. Reusar el código en actividad 1.1.

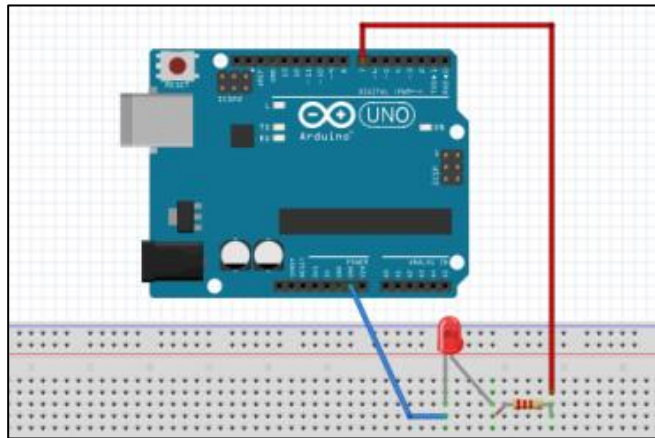


**Nota:** El valor de la resistencia en serie con el LED puede ser de un valor diferente a 220 ohm; el LED se encenderá también con valores de hasta 1K ohm.



*Conexiones internas de una protoboard*

- Verificar su funcionamiento.
- Si conectamos el led al pin digital número 7, ¿qué ocurre con nuestro programa?
- Modificar el circuito conectando el led al pin digital 7.



- Modificar el código del sketch para que el led, conectado al pin digital número 7, se encienda y apague intermitentemente, cada un segundo.

### **Actividad 1.3:** Efectos con leds

Ya sabemos cómo hacer parpadear un led, podemos hacer otros efectos con leds, encendiéndose y apagándose en una secuencia concreta para hacer efectos y comprender la programación de Arduino.

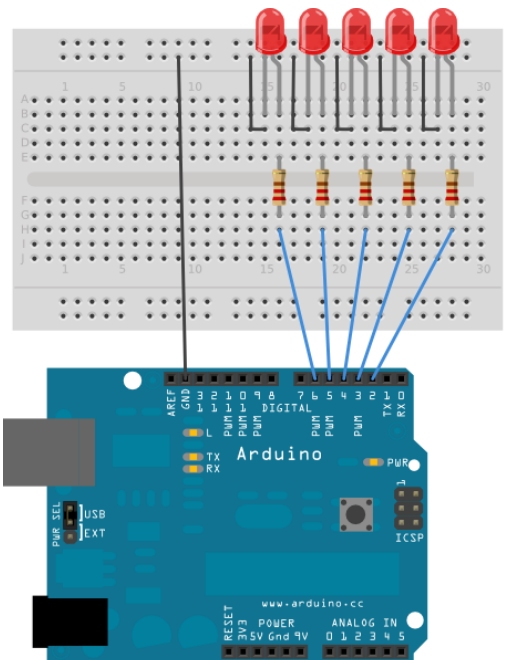
#### **Ejemplo: Coche Fantástico Simple**

Hacer el efecto de las luces del coche fantástico con 5 leds conectados a los pines 2, 3, 4, 5 y 6.

Como primera aproximación se puede ejecutar en cada loop el encendido y apagado de cada led esperando un tiempo entre cada encendido y apagado.

**Nota:** Para hacer "debug", sacar por puerto serie el estado de cada led.

- Diseñar el circuito utilizando el simulador tinkercad.

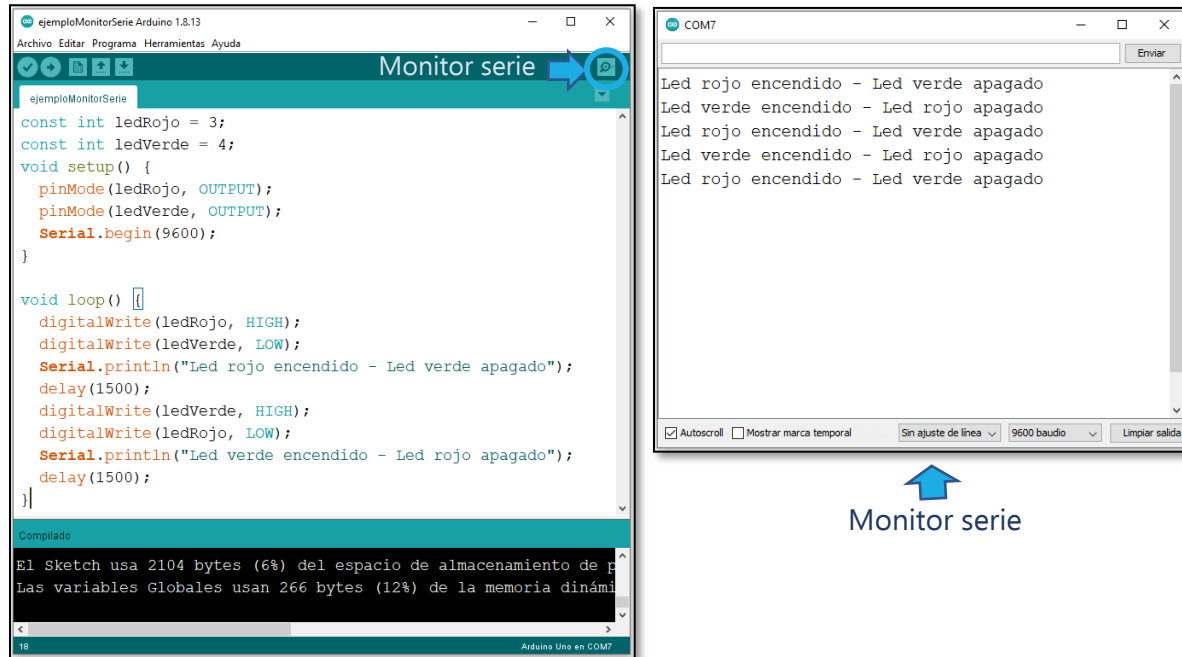


## MONITOR SERIE

El monitor serie es una de las herramientas de desarrollo más interesantes que nos aporta el entorno Arduino, mediante este monitor se pueden ver los valores de las variables, así como mensajes de texto que permiten conocer como está funcionando el programa, o simplemente interactuar con él.

El monitor serie consiste en una consola de entrada y salida, esto quiere decir que se puede mostrar datos enviados por la placa Arduino y también se puede enviar datos a la placa.

Se accede al monitor serial a través del menú Herramientas/Monitor serie o haciendo click en el icono que parece una lupa ubicado a la derecha en el IDE.



Monitor serie


### Actividad 1.4: Monitor Serie (opcional)

Construir un circuito que permita el encendido de dos leds, uno rojo y otro verde, en forma intermitente.

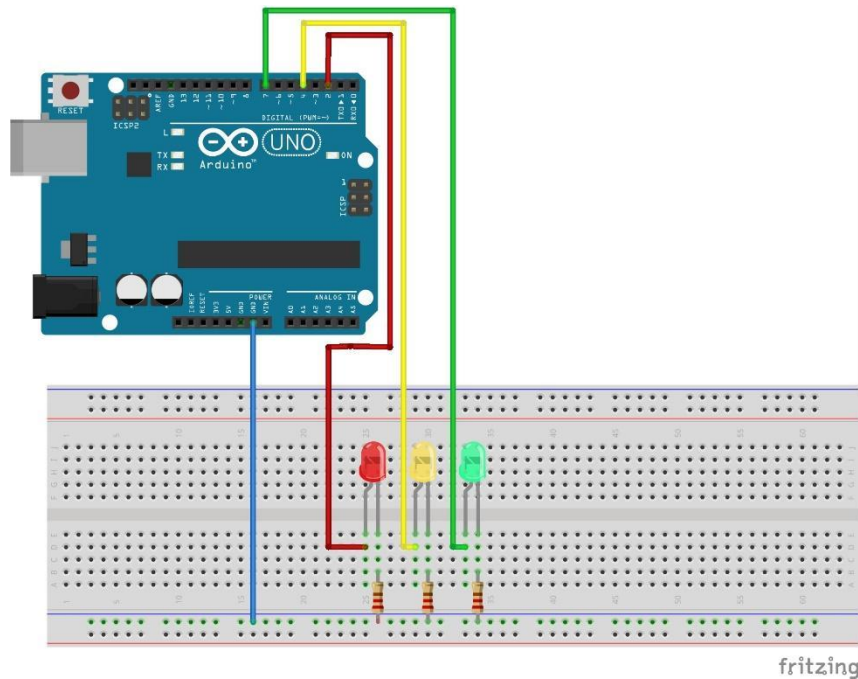
Codificar un sketch de Arduino que permita observar a través del monitor serie de Arduino, en que secuencia se encienden los leds, tal como lo muestra la figura anterior.

### Actividad 1.5: Efectos con leds bis. (opcional)

Teniendo en cuenta el circuito anterior modificar el código del proyecto para lograr diferente secuencia de encendido y apagado.

Por ejemplo, funcionamiento de un semáforo de tres tiempos. 

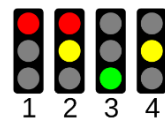
**Nota:** Puede agregar más leds o cambiar colores.



- Diseñar el circuito utilizando el software fritzing o tinkercad y posteriormente trasladar el diseño a un circuito físico, utilizando el Arduino y la protoboard.
- Cargar el sketch a la placa y verificar su funcionamiento.

### Actividad 1.6: Semáforo 4 de tiempos. (Opcional)

Teniendo en cuenta el ejercicio anterior, modificarlo para simular un semáforo de cuatro tiempos.



### Actividad 1.7: Semáforos en cruce de calles. (Opcional)

Modificar el ejercicio 1.5, agregando un nuevo semáforo, para simular el funcionamiento de semáforos en un cruce de calles.

- Diseñar el circuito utilizando fritzing o tinkercad y, posteriormente, armar el circuito físico.
- Codificar un sketch que permita simular el funcionamiento de los semáforos.
- Mostrar por monitor serie la secuencia en que se habilita el cruce de calles.

## LAS ENTRADAS ANALÓGICAS

Si el Arduino fuera puramente un dispositivo digital, no seríamos capaces de medir estas variables y limitaría el alcance de nuestros proyectos. Por suerte, el Arduino puede interactuar con el mundo analógico también.

Además de proporcionar una salida “cuasi” analógica por medio de PWM, el Arduino también puede adquirir una entrada analógica de entre 0 y 5 voltios. El Arduino estándar tiene seis entradas analógicas etiquetadas: ANALOG IN A0, A1, A2, A3, A4, A5.

¿Cuál es la diferencia entre el mundo digital y el analógico? En el mundo digital, todo tiene dos estados: un interruptor sólo puede estar encendido o apagado, un LED puede estar encendido o apagado. En el mundo analógico, las cosas tienen un rango de valores: la música tiene notas que abarcan una gama de frecuencias, un coche acelera a través de una gama de velocidades, la temperatura varía entre un máximo y un mínimo.

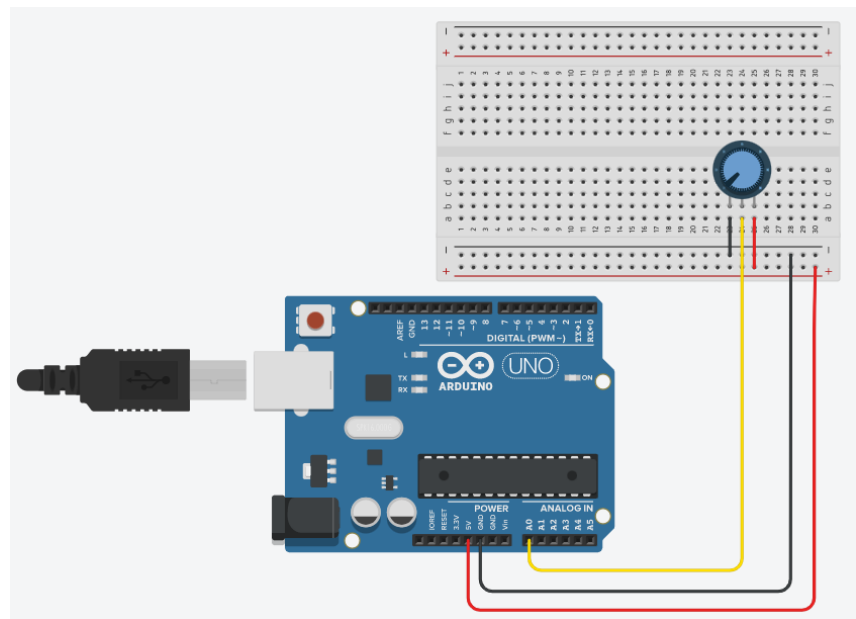
Arduino cuenta con un convertidor analógico-digital (ADC), el cual es de 10 bits, lo cual significa que puede distinguir entre 1024 valores

### *Actividad 2.2:* Valores del potenciómetro:

Conectar un potenciómetro a la protoboard y desde esta al Arduino. Imprimir en el monitor serie los valores leídos según el estado del potenciómetro.

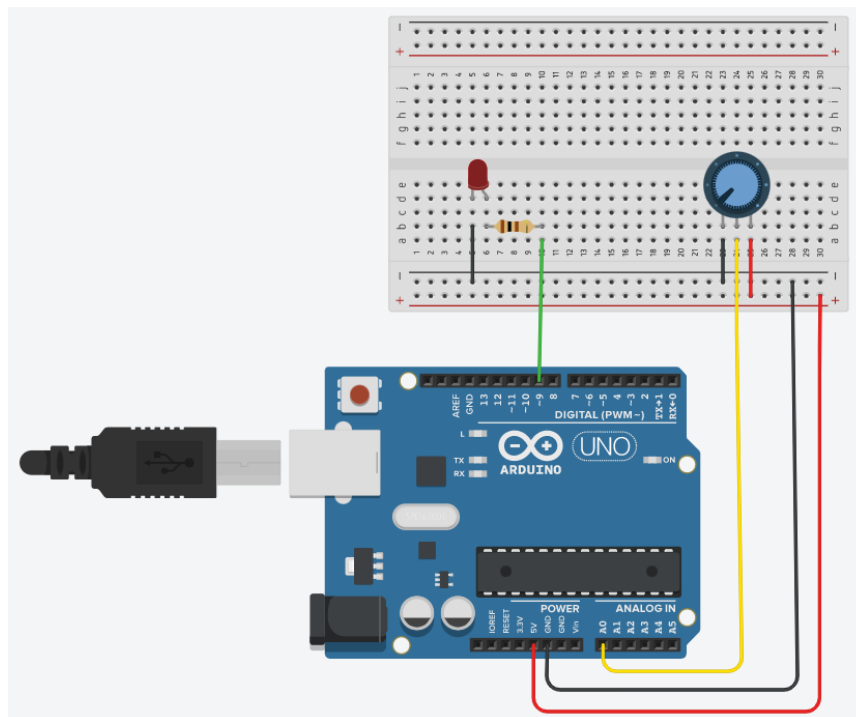
La salida deberá tener el formato  
“El valor del potenciómetro en el pin Ax es: 999”

/\*donde X es el pin analógico donde está conectado el potenciómetro y 999 es el valor leído.\*/\*



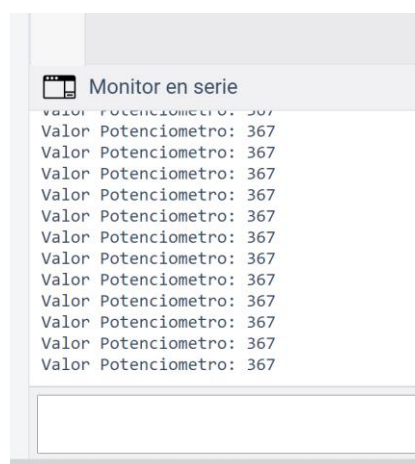
### Actividad 2.3: Control de iluminación de un led:

Armar el circuito según el siguiente diseño.



- Codificar un programa que permita, a partir del control con el potenciómetro, manejar el grado de iluminación del led.
- Los valores capturados desde el potenciómetro deben poder observarse en el Monitor Serial.

Por ejemplo



### Actividad 2.4: Control Velocidad del Efecto de Leds (opcional)

- Con el mismo montaje del ejercicio del auto fantástico, añadir un potenciómetro, al pin A0, para controlar la velocidad del efecto de los leds.
- Añadir en el código al inicio del loop() la línea "tiempoLeds = analogRead(A0);" para que la velocidad de los leds sea la "sensada" desde la entrada analógica de un potenciómetro.
- Conexión del potenciómetro:

