

# Trabajo Práctico Informática II - 2025

## Programación Orientada a Objetos

- En este práctico aplicará programación orientada a objetos (POO) programando sus propias clases en C++ en la plataforma Arduino.
- Puede trabajar en un simulador de Arduino como [Wokwi](#) o [Tinkercad](#) o con el mismo Arduino que ya posea.
- Recuerde comentar su código fuente.
- En la entrega adjunte el diagrama o esquema de conexión de componentes y el código fuente.

### Ejercicio 1: Simulación de Sensores con Clases.

1.1 Crear una clase Sensor con atributos como valor, magnitud y métodos para leer (leerMedición) y simular un valor de sensor (generarMedicion ).

1.2 Usar herencia para extender la clase para diferentes tipos de sensores (temperatura, luz, ultrasonido), mínimo tres tipos de sensores. Comente cuáles otros métodos y atributos agrega.

1.3 Implementar un programa que simule la lectura de estos sensores y muestre sus valores en el monitor serial.

1.4 ¿Qué modificaría para agregar un sensor real? Agregue un “sensor no simulado” (elija un sensor y conéctelo al arduino) creando una clase que herede de la clase Sensor.

### Ejercicio 2: Control de Leds

2.1 Crear una clase Led con métodos para encender, apagar y parpadear.

2.2 Instanciar múltiples objetos Led con diferentes pines y controlar su comportamiento de forma independiente.

2.3 Modificar el programa para que los LEDs reaccionen a la lectura de un sensor simulado, utilizando los objetos previamente definidos en el ejercicio 1.

## Ejercicio 3: Clase Reporte

- 3.1 Cree una clase que almacene en formato de lista enlazada los datos generados por los sensores.
- 3.2 Implemente un mecanismo para que se pueda pedir a través de la consola serie el histórico de esos datos. En este caso, debe enviar todos los datos hacia la consola serie.
- 3.3 Agregue algún método para procesar los datos (calcular valores mínimos, máximos, promedio, etc).

## Ejercicio 4: Clase Timer

- 4.1 Cree una clase Timer (o Cronometro) en Arduino y utilicela desde el programa principal. No puede usar ninguna librería extra, sólo la función millis(). Ud. activará el temporizador ante el evento de recibir un carácter (ejemplo "A"), lo detendrá al recibir otro caracter distinto (por ejemplo "D"), y lo resteará el temporizador al recibir el carácter "R".