

PROGRAMACIÓN FÍSICA

TEMA	Comenzando la programación de una placa Arduino o similar
FORMATO	Colectivo (los alumnos trabajan en pequeños grupos)
TIEMPO DE PREPARACIÓN	2 horas
DURACIÓN DE LA ACTIVIDAD	30-45 minutos
NIVEL DE DIFICULTAD	Medio

OBJETIVOS PEDAGÓGICOS

- Comenzar la programación de una placa Arduino o similar.
- Comprender el procedimiento de codificación y transferencia de la placa.
- Aprender a modificar un programa y probarlo.

MATERIALES NECESARIOS

Un proyector

Preferiblemente para cada grupo de estudiantes:

- 1 Computadora
- 1 Placa Arduino o similar
- 1 Protoboard
- LEDs

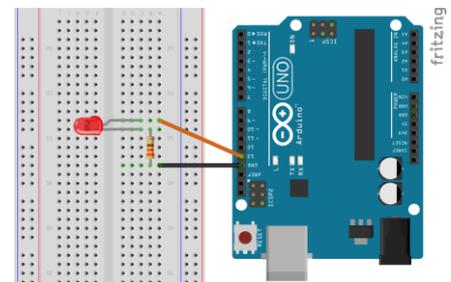
Preparación:

- Para comenzar a trabajar con placas Arduino o similares, deberá instalar el IDE de Arduino en las computadoras que se utilizarán. Recordando que el IDE de Arduino es el *software* que permite programar la placa.
- Además de instalar el IDE, también debe instalar el controlador que permitirá que su computadora se comunique con la placa Arduino. Si utiliza Mac o Linux, no es necesario instalar el controlador. Si utiliza Windows, haga lo siguiente: conecte su placa Arduino a su computadora usando el cable USB, aparecerá una ventana (Asistente para agregar nuevo *hardware*, o "*Found New Hardware Wizard*"), y Windows debe reconocer e instalar automáticamente los controladores.

PROGRAMACIÓN FÍSICA

Realización de la actividad:

- Si es posible, divida la clase en grupos pequeños. Cada grupo debe tener acceso a una computadora con un IDE de Arduino instalado, una placa Arduino o similar y un cable para conectar la placa a la computadora. Si esto no es posible, intente turnar los materiales para dar acceso a todos los estudiantes, o trabaje de forma colectiva, proyectando su pantalla.
- Comience presentando la placa Arduino y ejemplos de cosas que se pueden crear con ella.
- Luego, pídale a cada grupo que conecte su placa a la computadora y abra el IDE de Arduino.
- Muestre los pasos para transferir el ejemplo de Blink a la placa (lo que hace que el LED interno parpadee):
 - Para encontrar el código de ejemplo, vaya a Archivo> 01.Basics > Blink.
 - En Herramientas> Puerto, seleccione el puerto al que está conectado su Arduino.
 - Herramientas> Placa , verifique si su tablero está definido correctamente (ej .: Arduino / Genuino Uno).
 - Haga clic en Subir (icono con la flecha para la derecha en la esquina superior izquierda).
 - Si todo va bien, verá el mensaje "Subido" (o *Done uploading*) debajo del área de programación.
- Luego pida a los grupos que calculen cuánto tiempo (segundos) el LED está encendido y apagado. Luego, desafíelos a averiguar en qué parte del programa se establece este tiempo, sugiriendo que modifiquen el intervalo utilizando otros valores.
- También puede sugerir que creen otros patrones para los parpadeos del LED (agregando nuevas líneas de código, copiando y pegando desde el propio programa).
- Finalmente, si hay tiempo, puede pedirles que conecten un LED externo al *protoboard*, como en el ejemplo a continuación. Para hacer esto, desconecte el Arduino de la computadora y conéctelo nuevamente después de completar el montaje. Después, puede sugerirles que modifiquen el programa y el circuito para encender el LED en otros pines de Arduino (como en el pin 12 u 11, por ejemplo).
- Si hay suficientes materiales, los estudiantes pueden continuar conectando dos LED a la placa, modificando el programa inicial para que los dos LED parpadeen simultáneamente, o en patrones alternos creados por los estudiantes.



Discusión y reflexión:

- Después de completar la actividad, discuta los conceptos aprendidos con la clase, haciendo pruebas y diseñando el código, si es posible. Algunas preguntas que pueden guiar la discusión:
 - ¿Qué es un programa de computadora? ¿Cómo funciona este programa?
 - ¿En qué lugar del código se establece el intervalo de tiempo de parpadeo?
 - ¿Qué significa LOW y HIGH para Arduino? ¿Y pinMode (13, OUTPUT)?
 - ¿Qué pasaría si el fragmento de código dentro de la función *void loop* se cambiara a *void setup*?

Créditos:

Irene Karaguilla Ficheman (LSITec)

Cassia Fernandez (LSITec e USP)