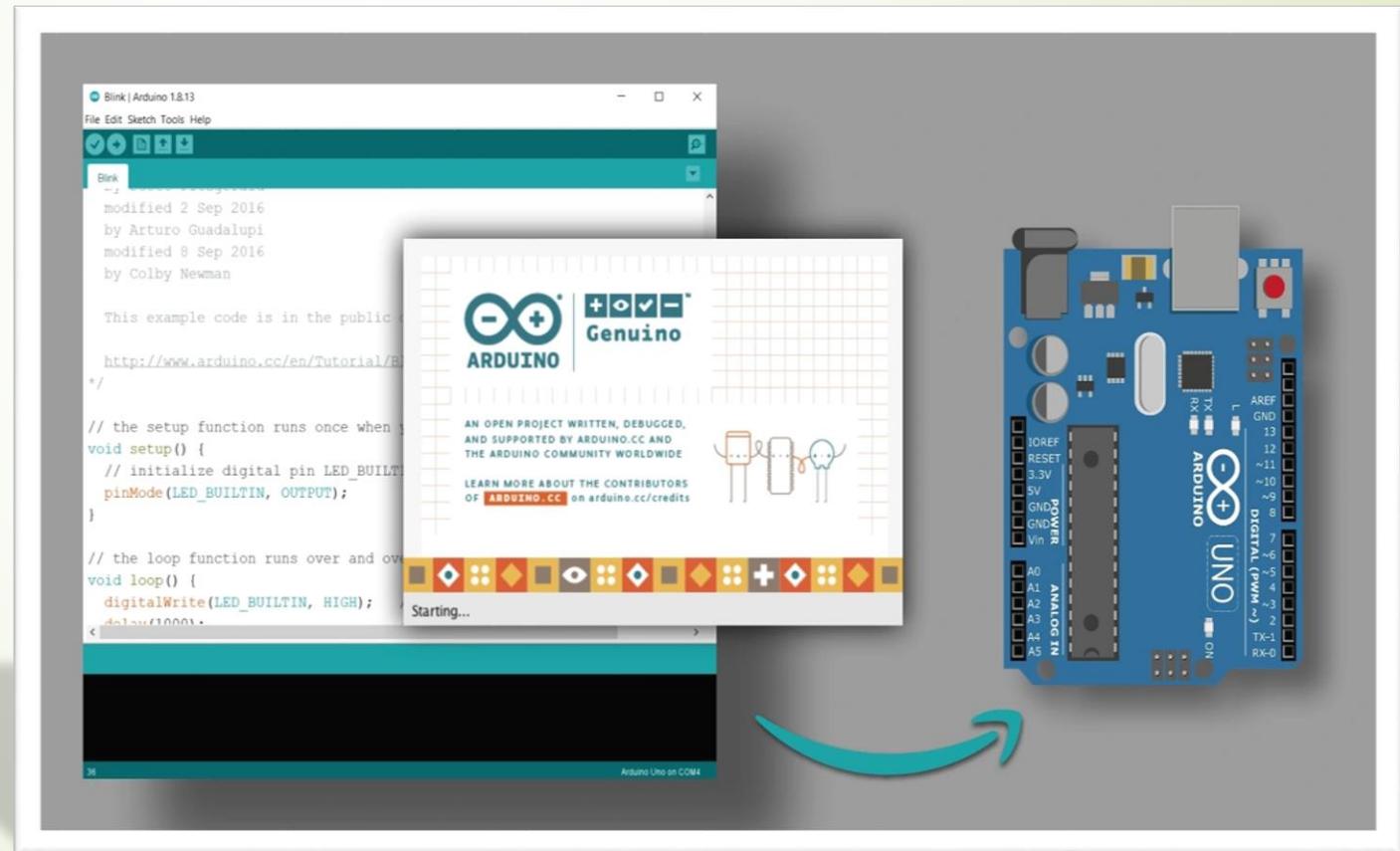


ENCUENTRO 6: IDE DE ARDUINO

SPORTIC
2022



Valor Olímpico: Excelencia

Habilidades socioemocionales: Pensamiento Creativo

Objetivos del encuentro		
Nivel I Acceso	Nivel II Uso	Nivel III Creación
<ul style="list-style-type: none">• Entender las ventajas del software libre.• Instalar el IDE de Arduino en un computador.	<ul style="list-style-type: none">• Identificar la importancia de usar un computador para crear un código para Arduino	<ul style="list-style-type: none">• Crear una solución que encienda y apague un led de forma indefinida.

Ítem	Actividad	Duración (minutos)
1	Actividad Rompe Hielo	15
2	Repaso encuentro anterior	5
3	Del computador a la placa. Subiendo un código a Arduino UNO	70
3.3	Trabajo #6 con el reto	10
5	Cierre de encuentro	10
6	Presentación de próximo encuentro	10
Total		120

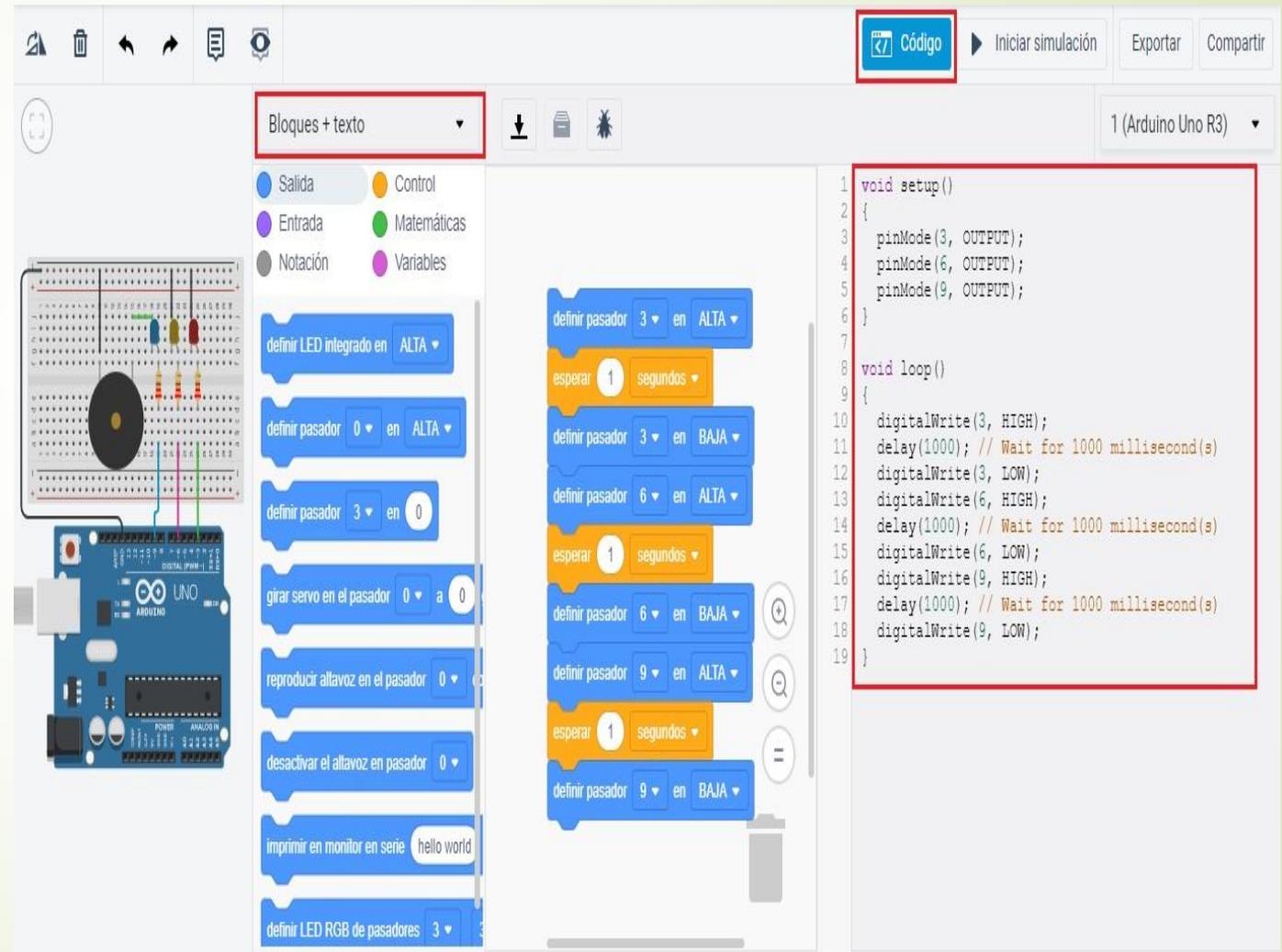
Actividad 1: Actividad rompe hielo “Chindogus”

- Revisa la actividad inicial que esta en Classroom
- Consulta en Internet
- Responde las preguntas en tu cuaderno
- Al iniciar la clase socializamos en el salón



Actividad 2: Del computador a la placa. Subiendo un código a Arduino UNO

- Esta actividad el objetivo principal es que se **pueda realizar su desarrollo de Tinkercad al subirlo a una placa de Arduino UNO**, para lograr esto **IDE de Arduino** debe estar instalado en los computadores, realice el proceso apoyándose en la guía de instalación.
- En primer lugar, debemos aprender como convertir los bloques de programación en lenguaje C++. Para tal fin, nos podemos basar en actividad anterior. En la parte superior



The screenshot displays the Tinkercad IDE interface. On the left, a breadboard circuit is shown with an Arduino Uno board connected to three LEDs. The central workspace contains a block-based program with the following steps:

- definir pasador 3 en ALTA
- esperar 1 segundos
- definir pasador 3 en BAJA
- definir pasador 6 en ALTA
- esperar 1 segundos
- definir pasador 6 en BAJA
- definir pasador 9 en ALTA
- esperar 1 segundos
- definir pasador 9 en BAJA

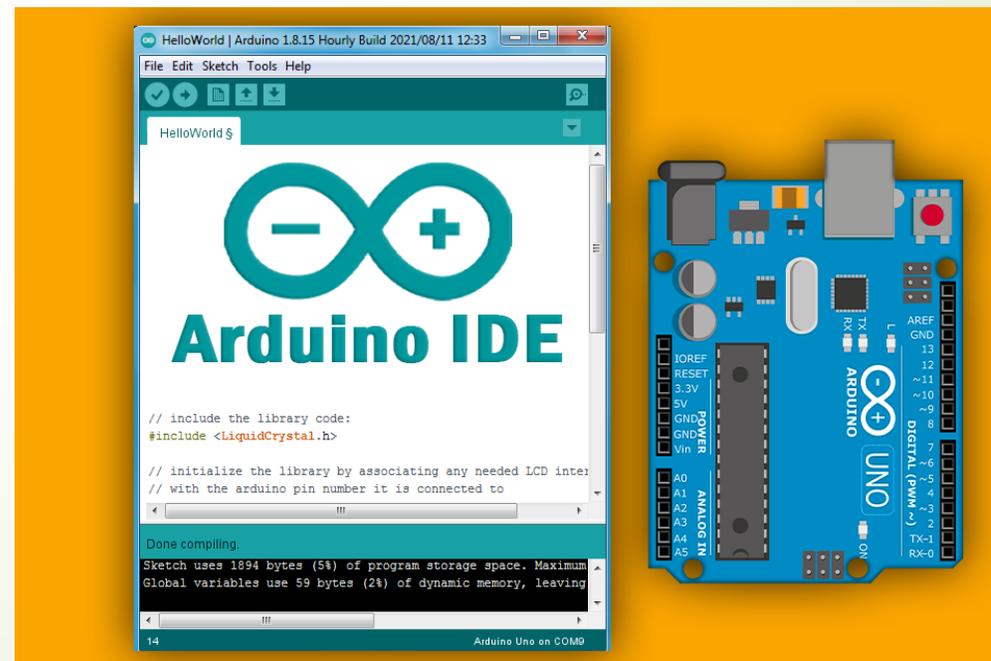
On the right, the C++ code is displayed in a text editor:

```
1 void setup()
2 {
3   pinMode(3, OUTPUT);
4   pinMode(6, OUTPUT);
5   pinMode(9, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(3, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(3, LOW);
13  digitalWrite(6, HIGH);
14  delay(1000); // Wait for 1000 millisecond(s)
15  digitalWrite(6, LOW);
16  digitalWrite(9, HIGH);
17  delay(1000); // Wait for 1000 millisecond(s)
18  digitalWrite(9, LOW);
19 }
```

Como podemos observar en la imagen anterior, al seleccionar la opción de **Bloques + texto** del módulo de **Código**, Tinkercad se encarga de convertir los bloques de programación en el código C++ que necesitamos para subirlo a la placa Arduino UNO. [Descargue de Classroom el código para el IDE de Arduino.](#)

El IDE de Arduino es un entorno de desarrollo integrado que permite la configuración de las placas de Arduino. Facilita el envío y/o recepción de información desde y hacia la placa y/o el computador.

El **IDE de Arduino** se identifica con un **símbolo de infinito** con el signo **más (+)** y **menos (-)** en su interior.



Mejor coincidencia

 **Arduino**
Aplicación

Aplicaciones

 **arduino-1.8.9-windows.exe** >

Buscar en Internet

 **arduino** - Ver resultados web >

Carpetas (6+)

Documentos (6+)

Fotos (1+)

 arduino



Arduino
Aplicación

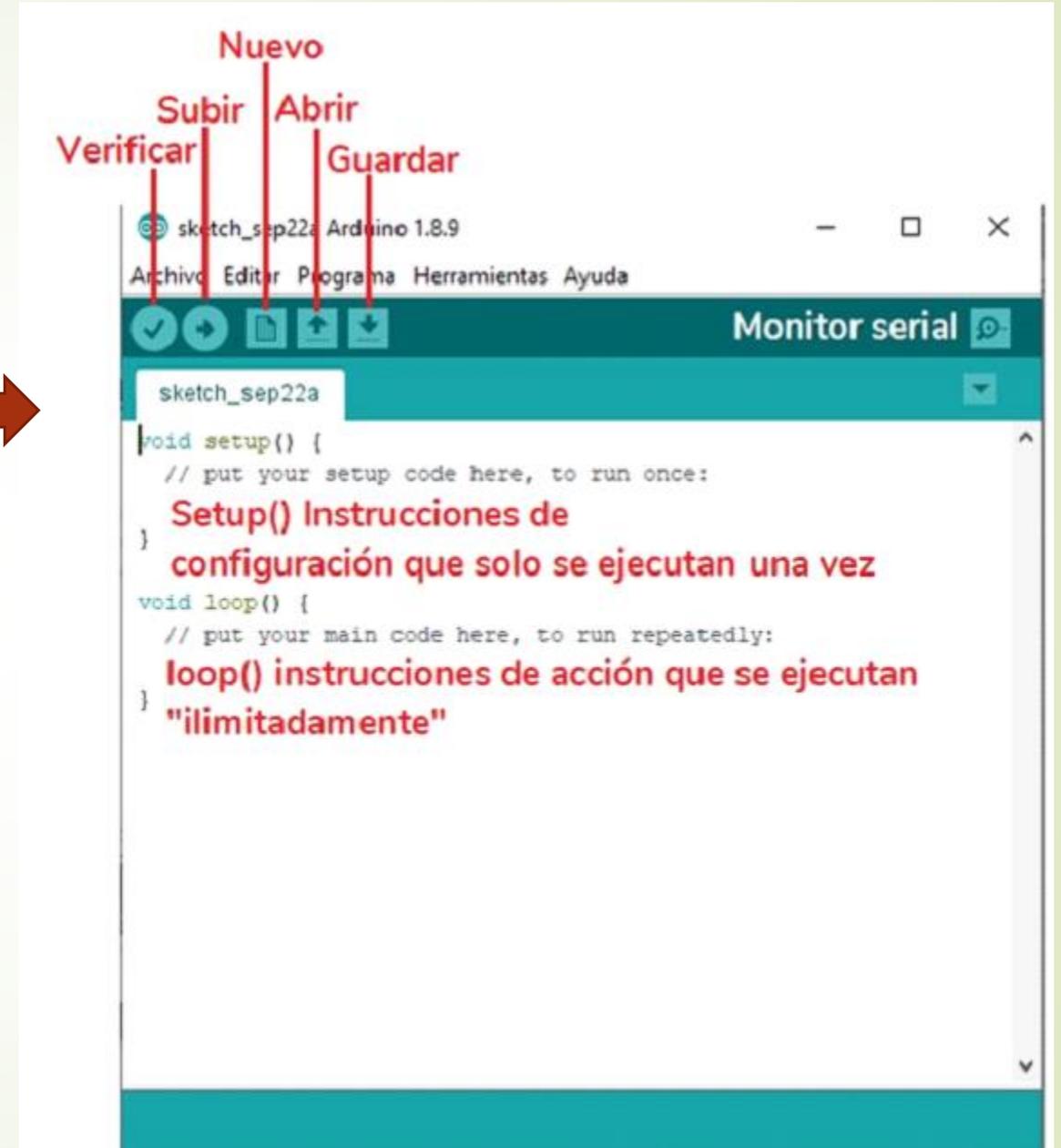
 **Abrir**

Recent

-  galga_hx711 - en galga_hx711
-  CELDA_DE_CARGA
-  galga_hx711 - en galga_hx711
-  CARGA_2
-  galga_hx711 - en galga_hx711
-  Balanza_digital_final
-  sketch_may16a

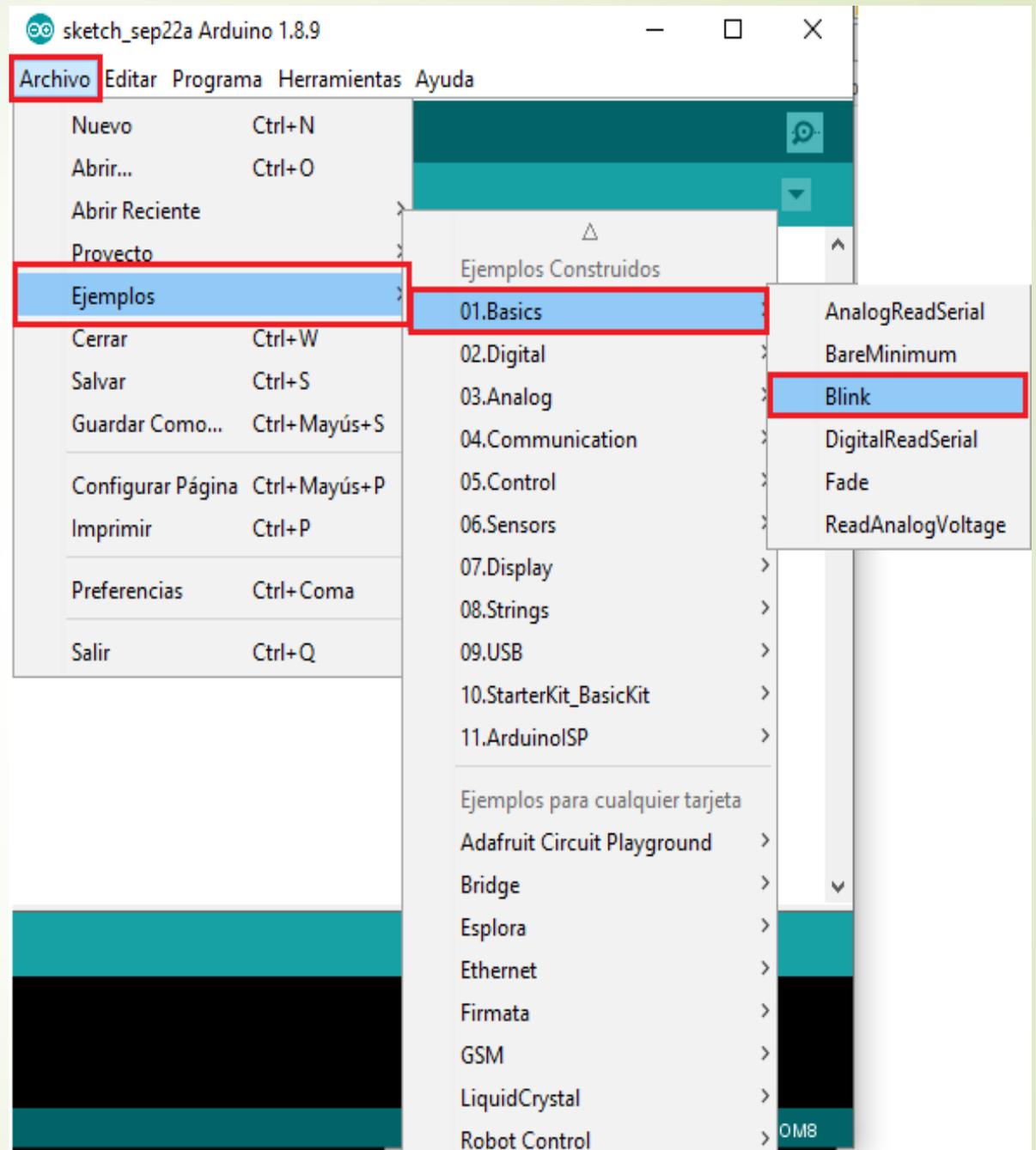
Al dar clic en el ícono, se abrirá un proyecto nuevo de Arduino, y el/la joven verá una pantalla como esta:

Una de las ventajas más conocidas de Arduino, es que al ser una plataforma open source (libre), la misma comunidad es la encargada que desarrollar librerías y ejemplos gratuitos para quien los pueda necesitar.



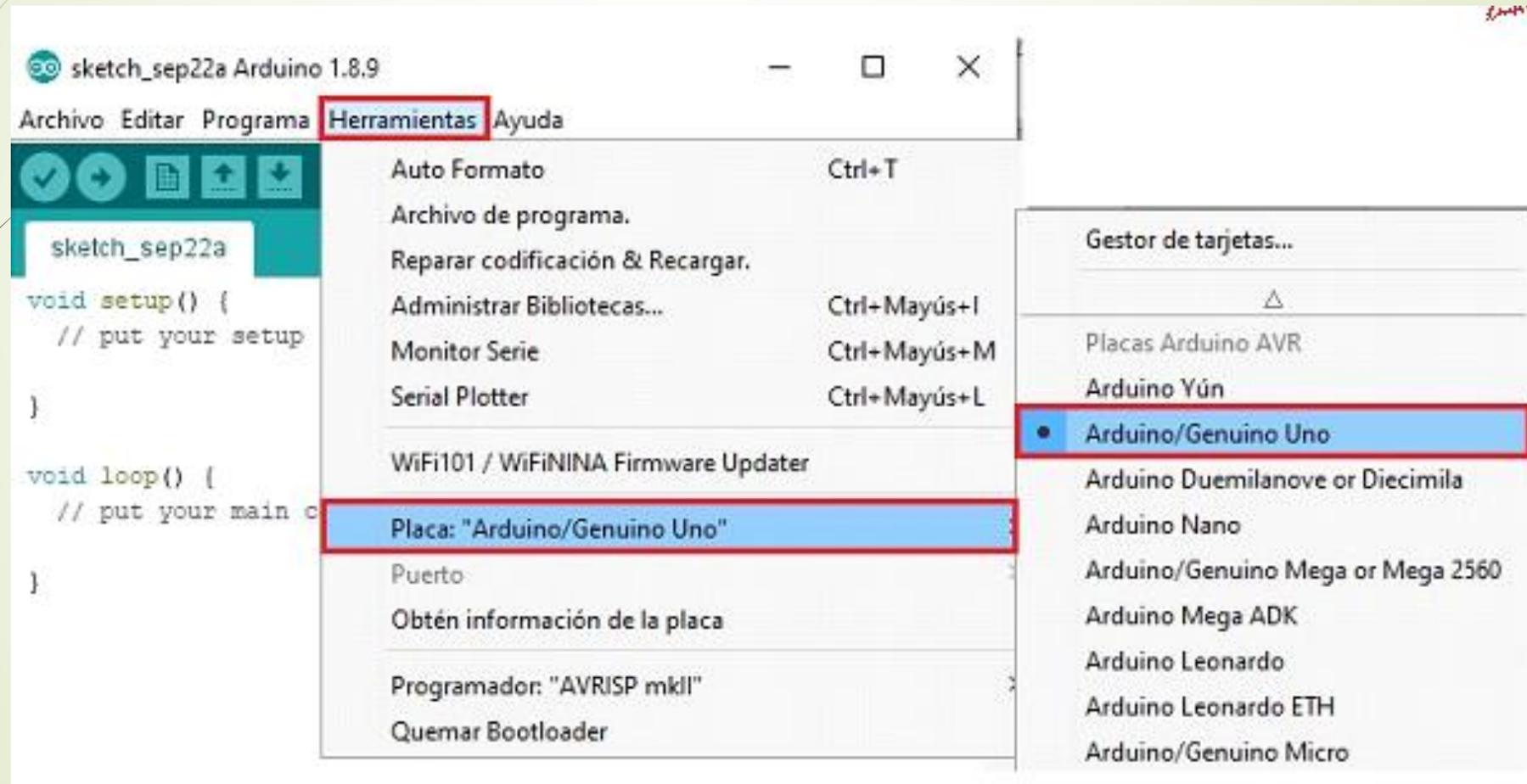
Por ejemplo, **al dar clic en el menú Archivo**, encontramos la opción de Ejemplos que le brinda al joven la oportunidad de construir su programa con código existente.

Esta es una característica destacada de Arduino, porque permite a los nuevos desarrolladores, construir sobre lo construido y reducir los tiempos de implementación de muchas soluciones



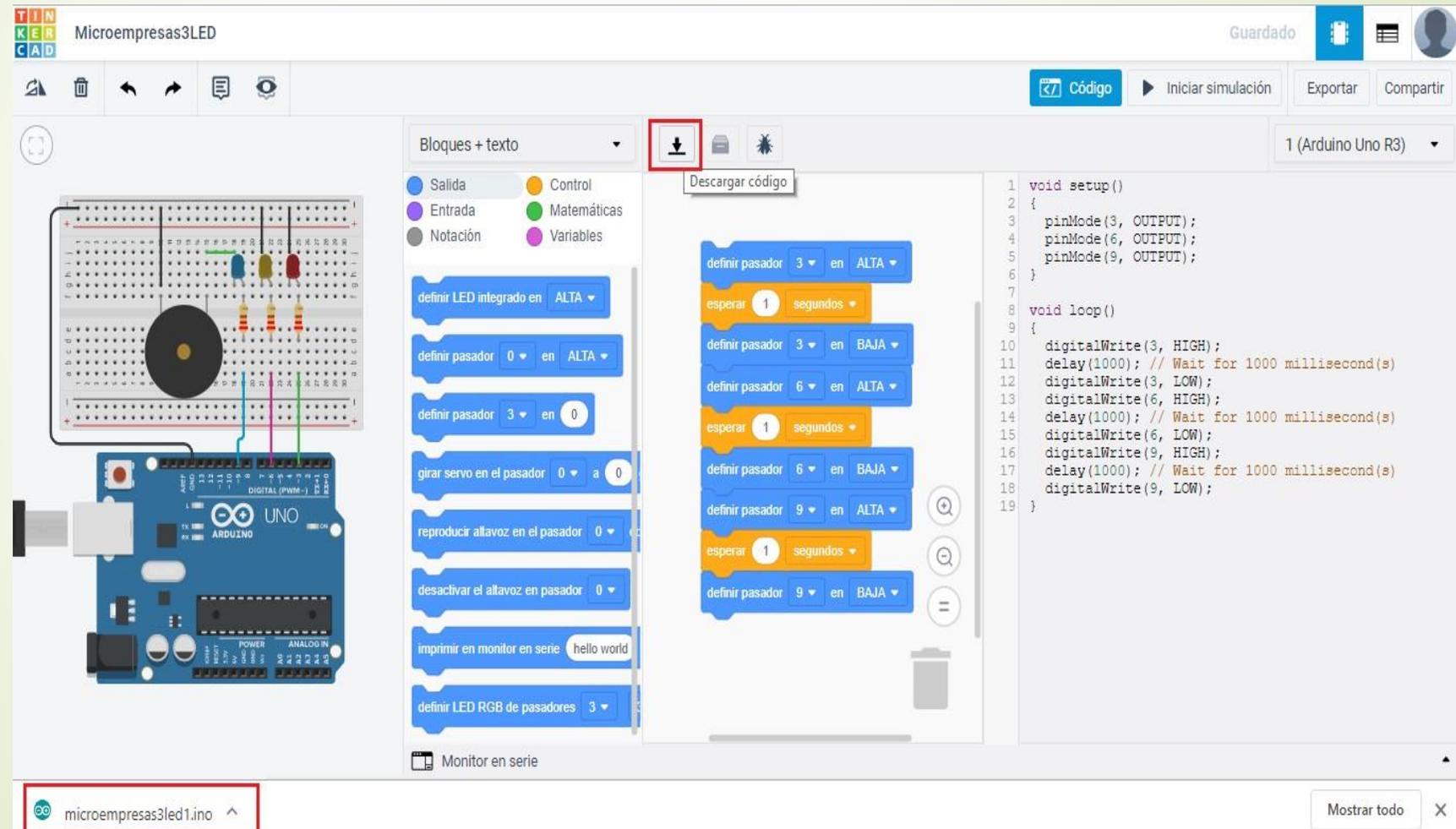
ES IMPORTANTE, que se escoja correctamente la placa con la cual está trabajando.

Para esta actividad, usaremos la placa de **Arduino UNO**, por lo que se debe seleccionar esta opción en el menú de **Herramientas, Placa** y asegurarse de que esta está seleccionada, así:



Al conectar la placa de Arduino UNO en el puerto USB, Herramientas/Puerto debe indicar un COM donde la placa ha sido detectada, de no ser así trate instalando los drivers del CH340 ([Descargar al final de este enlace](#)).

Solo queda disponer de un código para subirlo a la placa Arduino UNO. Recordemos que de Tinkercad se puede copiar el código traducido de los bloques, sin embargo, también es posible descargar el archivo **.ino**



The screenshot shows the Tinkercad workspace with an Arduino Uno board connected to a breadboard. The breadboard contains a servo motor and three LEDs. The code editor on the right displays the following C++ code:

```
1 void setup()
2 {
3   pinMode(3, OUTPUT);
4   pinMode(6, OUTPUT);
5   pinMode(9, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(3, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(3, LOW);
13  digitalWrite(6, HIGH);
14  delay(1000); // Wait for 1000 millisecond(s)
15  digitalWrite(6, LOW);
16  digitalWrite(9, HIGH);
17  delay(1000); // Wait for 1000 millisecond(s)
18  digitalWrite(9, LOW);
19 }
```

A red box highlights the 'Descargar código' button in the top toolbar. Another red box highlights the filename 'microempresas3led1.ino' at the bottom of the interface.

Vamos a traer el código en texto para analizar cada línea. El/la joven también podrá copiar este código y pegarlo en un proyecto nuevo del IDE de Arduino.

El/la joven podrá escoger entre los pines del 0-13 para encender y apagar los tres LED. También podrá cambiar el tiempo de encendido/apagado de cada uno.

```
void setup()
{
  pinMode(3, OUTPUT); //Inicia setup()
  pinMode(6, OUTPUT); //Se define el pin 3 como salida
  pinMode(9, OUTPUT); //Se define el pin 6 como salida
} //Se define el pin 9 como salida
//Termina setup()

void loop()
{
  digitalWrite(3, HIGH); //Inicia loop()
  delay(1000); //Envía 5V al pin 3, se enciende LED rojo
  digitalWrite(3, LOW); //Espera 1000 milisegundos (un segundo)
  digitalWrite(6, HIGH); //Envía 0V al pin 3, se apaga el LED rojo
  delay(1000); //Envía 5V al pin 6, se enciende LED amarillo
  digitalWrite(6, LOW); //Espera 1000 milisegundos (un segundo)
  digitalWrite(9, HIGH); //Envía 0V al pin 6, se apaga el LED amarillo
  delay(1000); //Envía 5V al pin 9, se enciende LED azul y el zumbador
  digitalWrite(9, LOW); //Espera 1000 milisegundos (un segundo)
} //Envía 0V al pin 9, se apaga el LED azul y el zumbador
//El código se devuelve al loop() y repite la sentencia
```

RESPONDA LAS SIGUIENTES PREGUNTAS

- ¿Puede una solución creada en Arduino salvar una vida?
- ¿Qué funciones extra le gustaría darle a Arduino?
- ¿Les parece que Arduino UNO es muy grande o pequeño para soluciones robóticas?



Trabajo #6 con el reto:

- ▶ Durante diez (10) minutos los(as) jóvenes, en sus propios grupos de robótica, revisarán cuanto espacio físico ocupará la solución electrónica, robótica o mecánica propuesta. El/la docente acompañará a los(as) jóvenes para identificar posibles tamaños de piezas que no tengan físicamente disponibles.

