



## Arduino. Nivel I

septiembre, 2019



**Objetivos del nivel**

- Conocer el entorno de desarrollo para Arduino
- Conocer los aspectos elementales del lenguaje de programación que se utiliza en Arduino: variables, condiciones, ciclos.
- Aprender a mostrar Inputs/Outputs (entradas y salidas) tanto analógicas como digitales

**Prerrequisitos del nivel**

- Lógica de Programación Nivel II
- Electrónica Nivel II

**Acerca de este manual**

Este manual pertenece al Centro de Asesoramiento y Desarrollo Informático C.A. (CADI F1). Para obtener más información sobre este u otros cursos visite nuestro sitio Web [www.cadif1.com](http://www.cadif1.com), escribanos a la dirección de correo [cadi@cadif1.com](mailto:cadi@cadif1.com) o visítenos en nuestra sede ubicada en la Av. Pedro León Torres con calle 59, Centro Comercial Sotavento, piso 2 oficina 27, Barquisimeto estado Lara, Venezuela. Tlf. 0251-7179247, 0251-4410268.

Las marcas mencionadas en este manual son propiedad de sus respectivos dueños. Copyright 2019. Todos los derechos reservados.



## Contenido del nivel

### Capítulo 1. Conociendo Arduino

- 1.1.- Microcontroladores.
- 1.2.- Arduino.
- 1.3.- Tarjetas Disponibles.

### Capítulo 2. El Entorno de Desarrollo

- 2.1.- Entorno de Trabajo Web.
- 2.2.- Emulador.
- 2.3.- Configuración Del Entorno de Escritorio.

### Capítulo 3. Iniciando la Programación en Arduino

- 3.1.- Compilación y Transferencia.
- 3.2.- Estructura de un Programa Arduino.
- 3.3.- Comentarios.
- 3.4.- Librerías.

### Capítulo 4. Salidas. Parte 1

- 4.1.- Serial Monitor.
- 4.2.- Enviando Salidas.
- 4.3.- Retardos (delay y Delaymicroseconds).

### Capítulo 5. Salidas. Parte 2

- 5.1.- Configuración de Pines.
- 5.2.- Digital Output.
- 5.3.- Analog Output.

### Capítulo 6. Variables. Parte 1

- 6.1.- Tipos de Datos.
- 6.2.- Declaración de Variables Locales y Globales.

### 6.3.- Asignaciones.

## Capítulo 7. Variables. Parte 2

- 7.1.- Operadores Aritméticos.
- 7.2.- Operadores y Asignaciones (+,-,\*,+=, \*= ).
- 7.3.- Constantes.

## Capítulo 8. Entradas. Parte 1

## Capítulo 9. Entradas. Parte 2

- 9.1.- Analog Read.
- 9.2.- Digital Read.

## Capítulo 10. Condiciones. Parte 1

- 10.1.- Operadores Relacionales.
- 10.2.- If.
- 10.3.- If/else.

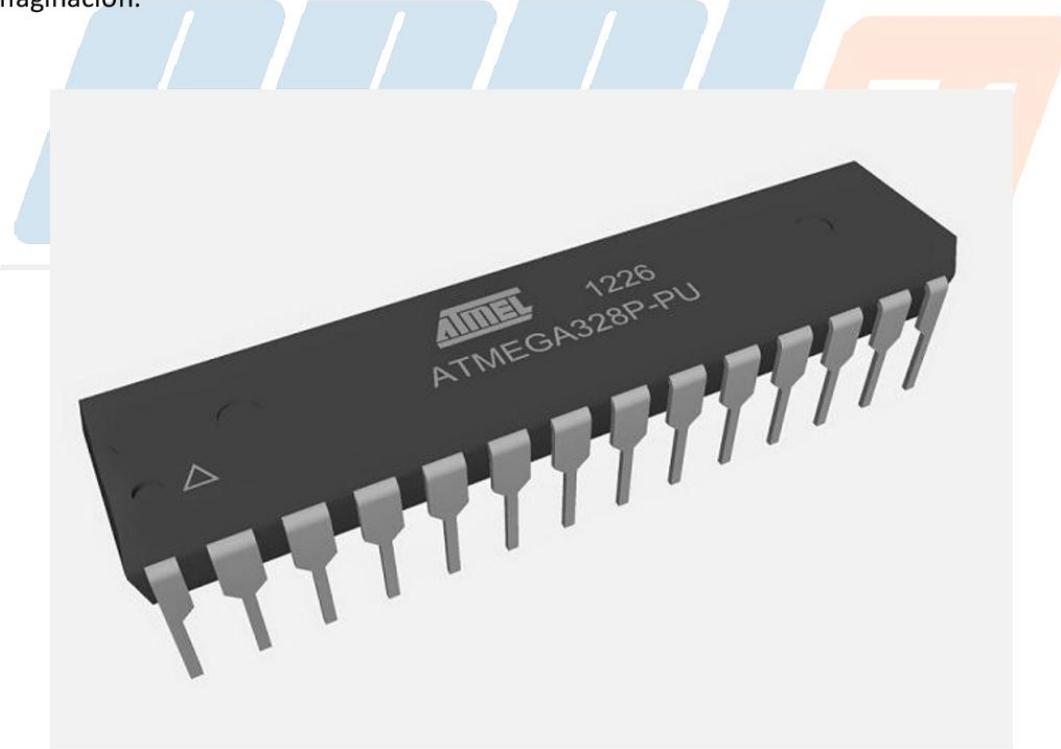
ACADEMIA DE SOFTWARE

## Capítulo 1. CONOCIENDO ARDUINO

### 1.1.- Microcontroladores

Un microcontrolador es un circuito integrado digital que puede ser usado para muy diversos propósitos debido a que es programable. Está compuesto por una unidad central de proceso (CPU), memorias (ROM y RAM) y líneas de entrada y salida (periféricos).

Un microcontrolador tiene los mismos bloques de funcionamiento básicos de una computadora lo que permite tratarlo como un pequeño dispositivo de cómputo. Puede usarse para muchas aplicaciones algunas de ellas son: manejo de sensores, controladores, juegos, calculadoras, agendas, avisos lumínicos, secuenciador de luces, cerrojos electrónicos, control de motores, relojes, alarmas, robots, entre otros. El límite es la imaginación.



Como el hardware ya viene integrado en un solo chip, para usar un microcontrolador se debe especificar su funcionamiento por software a través de programas que indiquen las instrucciones que el microcontrolador debe realizar. En una memoria se guardan los programas y un elemento llamado CPU se encarga de procesar paso por paso las instrucciones del programa.

Los lenguajes de programación típicos que se usan para este fin son ensamblador y C, pero antes de grabar un programa al microcontrolador hay que compilarlo a hexadecimal que es el formato con el que funciona el microcontrolador.

Para diseñar programas es necesario conocer los bloques funcionales básicos del microcontrolador, estos bloques son:

- CPU (Unidad central de proceso)
- Memoria ROM (Memoria de solo lectura)
- Memoria RAM (Memoria de acceso aleatorio)
- Líneas de entrada y salida (Periféricos)

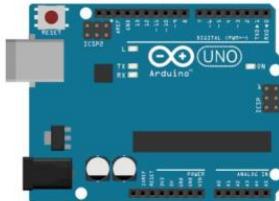
La CPU posee, de manera independiente, una memoria de acceso rápido para almacenar datos denominada registros, si estos registros son de 8 bits se dice que el microcontrolador es de 8 bits.

## 1.2.- Arduino

Es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra, los que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla.

Una placa electrónica es una PCB (“Printed Circuit Board”, “Placa de Circuito Impreso” en español). Las PCBs están fabricadas en un material no conductor, la cual consta de distintas capas de material conductor.

Una PCB es la forma más compacta y estable de construir un circuito electrónico. Así que la placa Arduino no es más que una PCB que implementa un determinado diseño de circuitería interna.



Entre las características más resaltantes de Arduino están:

- es libre y extensible: Esto quiere decir que cualquiera que desee ampliar y mejorar el diseño hardware de las placas como el entorno de desarrollo, puede hacerlo sin problemas.
- posee una gran comunidad: Gracias a su gran alcance hay una gran comunidad trabajando con esta plataforma, lo cual genera una cantidad de documentación bastante extensa, la cual abarca casi cualquier necesidad.
- su entorno de programación es multiplataforma: Se puede instalar y ejecutar en sistemas operativos Windows, Mac OS y Linux.
- lenguaje de programación de fácil compresión: Su lenguaje de programación basado en C++ es de fácil compresión que permite una entrada sencilla a los nuevos programadores y a la vez con una capacidad tan grande, que los programadores más avanzados pueden exprimir todo el potencial de su lenguaje y adaptarlo a cualquier situación.

La enorme flexibilidad y el carácter libre y abierto de Arduino hacen que se pueda utilizar este tipo de placas prácticamente para cualquier cosa, desde relojes hasta básculas conectadas, pasando por robots, persianas controladas por voz y muchas otras cosas.

La respuesta es que se puede hacer y construir casi de todo. Arduino es una plataforma para programar un microcontrolador y por lo tanto puede hacer lo que se puede imaginar, todo depende de la imaginación.

Se llegó a construir incluso un robot BB8 a tamaño real con una placa Arduino y conectividad bluetooth para poder ser controlado desde un smartphone.



Entre otras aplicaciones que se les ha dado a las placas Arduino están:

-Cámaras espías: con el uso de un módulo acelerómetro que detecte cuando se gire se podrá tomar una foto sin que la mayoría se dé cuenta, la cámara puede integrarse en un vaso o en cualquier lugar pequeño para pasar desapercibida.

- Control de tráfico: Se puede construir un semáforo inteligente con Arduino, no sólo mostrará las luces rojo, verde y amarillo para cada situación, sino que también incluye un sensor para ver un carro que se acerque al semáforo y que cambie la luz a verde en caso de que en el sentido contrario no vengan carros.



### 1.3.- Tarjetas Disponibles

Arduino cuenta con varios prototipos y modelos de placas a elegir, dependiendo de que tan grande sea el proyecto para el cual será usado. Entre los modelos más populares están:

- Arduino UNO.
- Arduino Nano.

- Arduino Leonardo.
- Arduino YUN.

Cada una de estas placas tiene características particulares que las diferencian de las demás.



#### Arduino UNO:

Es una placa basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión USB, conector jack de alimentación, terminales para conexión ICSP y un botón de reseteo. Tiene toda la electrónica necesaria para que el microcontrolador opere, simplemente hay que conectar la energía por el puerto USB ó con un transformador AC-DC.

**Características:**

- Microcontrolador: ATmega328
- Voltaje de entrada: 7-12V
- Pines: 14 pines digitales de I/O
- Entradas analógicas: 6
- Memoria flash: 32k
- Reloj: 16MHz

**ACADEMIA DE SOFTWARE****Arduino Nano:**

Es una placa de desarrollo de tamaño compacto, completa y compatible con protoboards, basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión Mini-USB, terminales para conexión ICSP y un botón de reseteo. Posee las mismas capacidades que un Arduino UNO, tanto en potencia del microcontrolador como en conectividad, solo se ve recortado en su conector USB, conector jack de alimentación y los pines cambia un formato de pines header.

**Características:**

- Microcontrolador: ATMega328
- Voltaje de operación: 5V
- Voltaje de alimentación (Recomendado): 7-12V

- I/O Digitales: 14 (6 son PWM)
- Memoria Flash: 32KB
- EEPROM: 1KB
- Frecuencia de trabajo: 16MHz



Arduino Leonardo:

Utiliza un microcontrolador ATmega32U4 que permite un diseño mucho más sencillo. Una de las ventajas de este microcontrolador es que dispone de USB nativo por hardware y por lo tanto no necesita de ninguna conversión serie-USB. También permite a la placa ser utilizada y programada como un dispositivo de entrada para emular un teclado, ratón, etc.

Características:

- Microcontrolador: ATmega32u4
- Tensión de funcionamiento: 5V
- Alimentación recomendada: 7-12V
- Pines I/O Digitales: 20
- Canales PWM: 7
- Entradas analógicas: 12

- Corriente Maxima de los pines I/O: 40 mA
- Corriente Maxima de los pines 3.3V: 50 mA
- Memoria Flash: 32 KB (4 KB usados para el bootloader)
- SRAM: 2.5 KB
- EEPROM interna: 1 KB
- Velocidad: 16 MHz



## Arduino YUN:

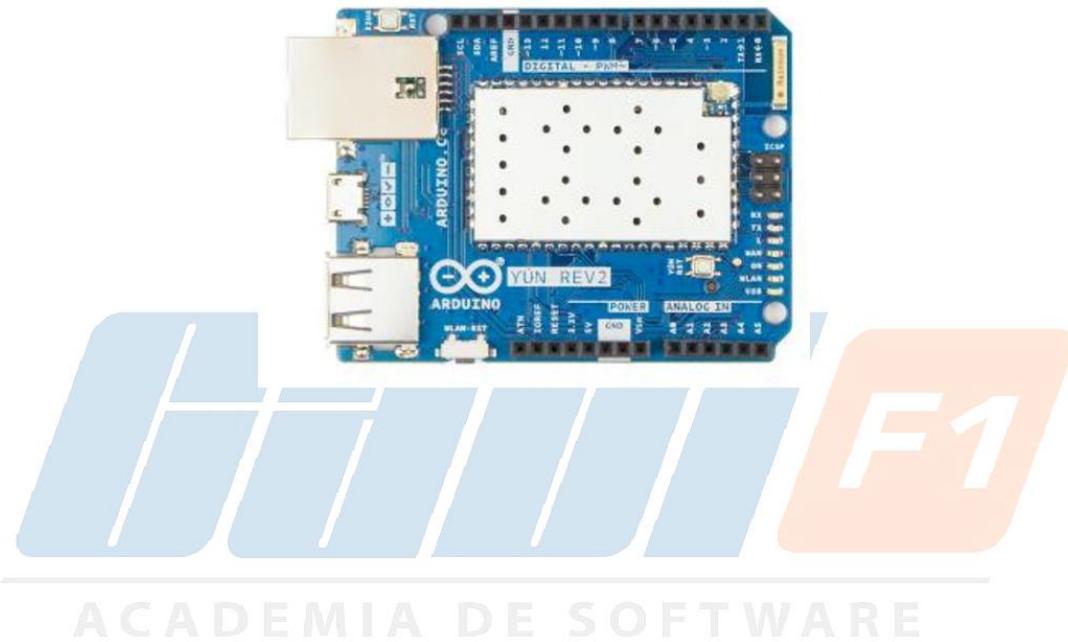
Tiene un sistema basado en Linux que permite conexiones y aplicaciones de red avanzadas.

Se puede conectar la placa a la red Wi-Fi o a una red cableada de una forma sencilla gracias al panel web de Yún. El panel web permite administrar la configuración de la placa y la carga de sketches.

### Características:

- Procesador: Atheros AR9331
- Arquitectura: MIPS @400MHz
- Alimentación: 3.3V
- Puerto Ethernet: IEEE 802.3 10/100Mbit/s
- Conexión WiFi: IEEE 802.11b/g/n
- USB Type-A: 2.0 Host/Device

- Lector de tarjetas: Micro-SD
- RAM: 64 MB DDR2
- Memoria Flash: 32 MB
- Soporte para PoE tipo 802.3af



## Capítulo 2. EL ENTORNO DE DESARROLLO

### 2.1.- Entorno de Trabajo Web

Para trabajar en Arduino existen muchos entornos de trabajo con el cual puede ser manejado, entre están:

- Arduino Software (IDE) (Desktop)
- Arduino Web
- Visual Studio Code
- Emulador TinkerCad

#### Arduino Web Editor

Permite escribir código y programar sketches a cualquier placa arduino desde el navegador, ya sea (Chrome, Firefox, Safari o Edge). Es recomendable utilizar Google Chrome.

Este IDE es parte de Arduino Create, una plataforma online que permite a los desarrolladores escribir código, acceder a tutoriales, configurar placas y compartir proyectos. Diseñado para proveer a los usuarios un flujo de trabajo continuo.

El Arduino Web está alojado de manera online, por lo tanto, siempre estará actualizado. Además, este IDE permite guardar código a la nube. Automáticamente reconocerá cualquier Arduino conectado a la computadora y lo configurará acorde.

Ingresar a Arduino Web Editor y crear una cuenta.

Completado el formulario dar click en el botón “CREATE ACCOUNT”. (Activar la cuenta a través del email de activación).

SIGN UP

Username: jomiva5016@gmail.com

Email: jomiva5016@gmail.com

Password:

Confirm password:

I confirm to have read the [privacy policy](#) \* and to accept the [Terms of service](#) \*

I confirm my consent to receive your newsletter

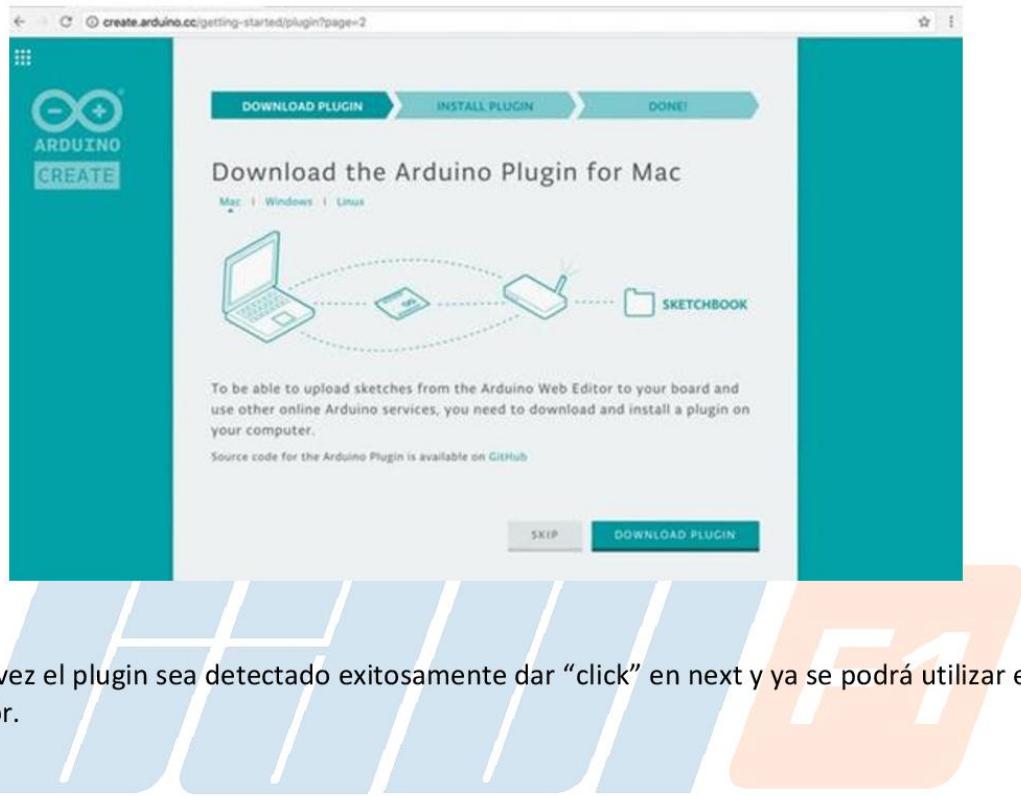
I confirm my consent to the processing of my personal data for marketing purposes, consisting in commercial offers sent via email

I confirm my consent to automated processing of my personal data, by means of profiling, in order to receive commercial offers customized on the basis of my browsing and purchasing behavior

No soy un robot reCAPTCHA

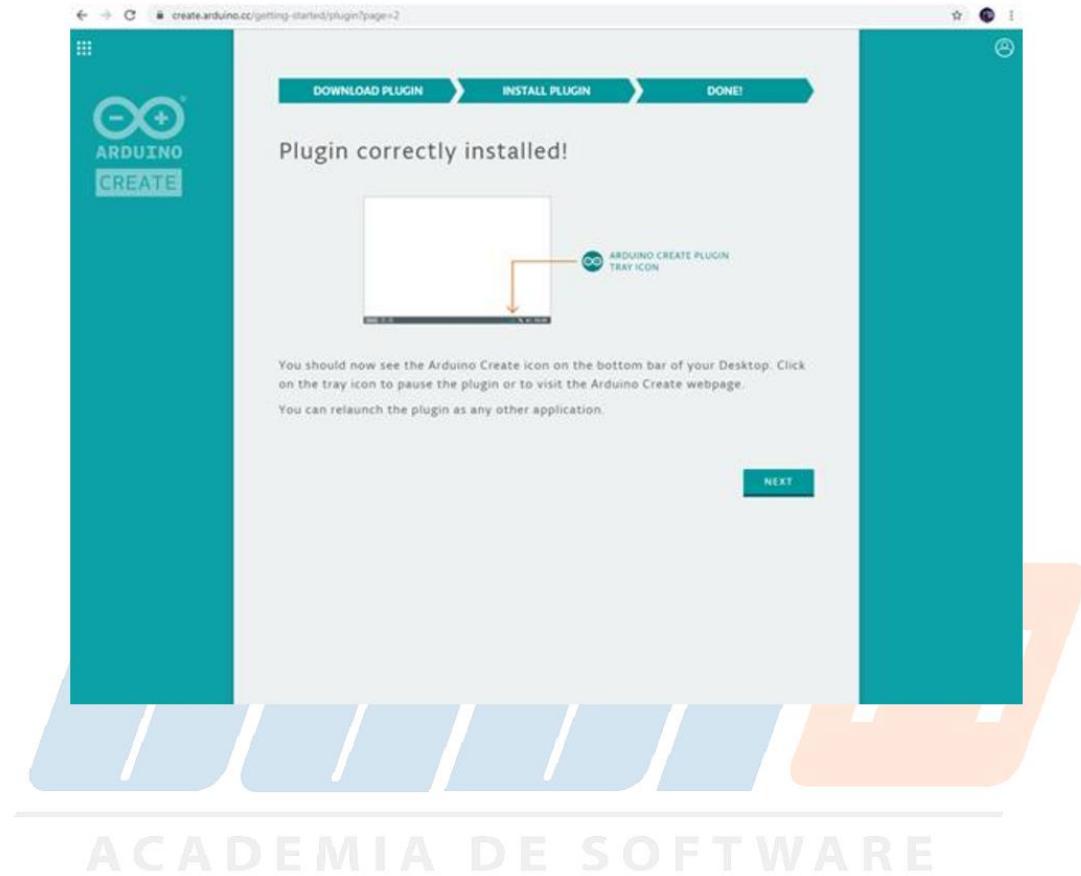
CREATE ACCOUNT

Una vez que la cuenta sea creada exitosamente, ir al link [create.arduino.cc/editor](http://create.arduino.cc/editor). Iniciar sesión con la nuestra previamente creada y descargar el plugin que se indica para poder instalar los drivers y poder programar las placas de Arduino.

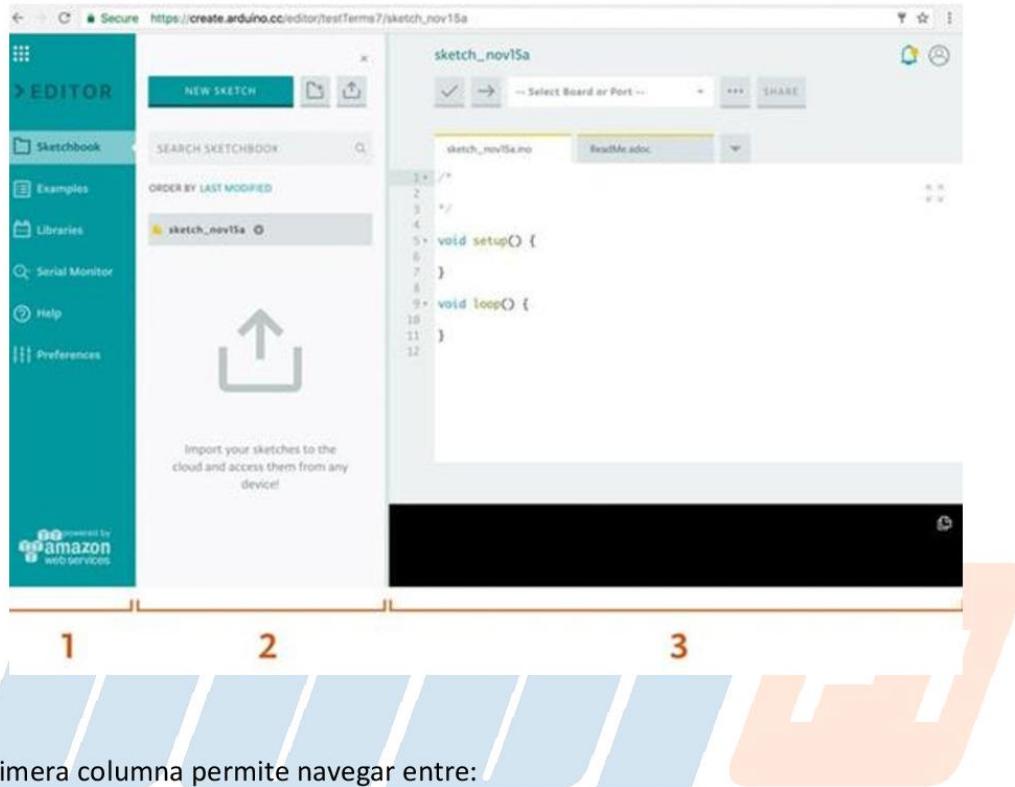


Una vez el plugin sea detectado exitosamente dar “click” en next y ya se podrá utilizar el editor.

ACADEMIA DE SOFTWARE



ACADEMIA DE SOFTWARE

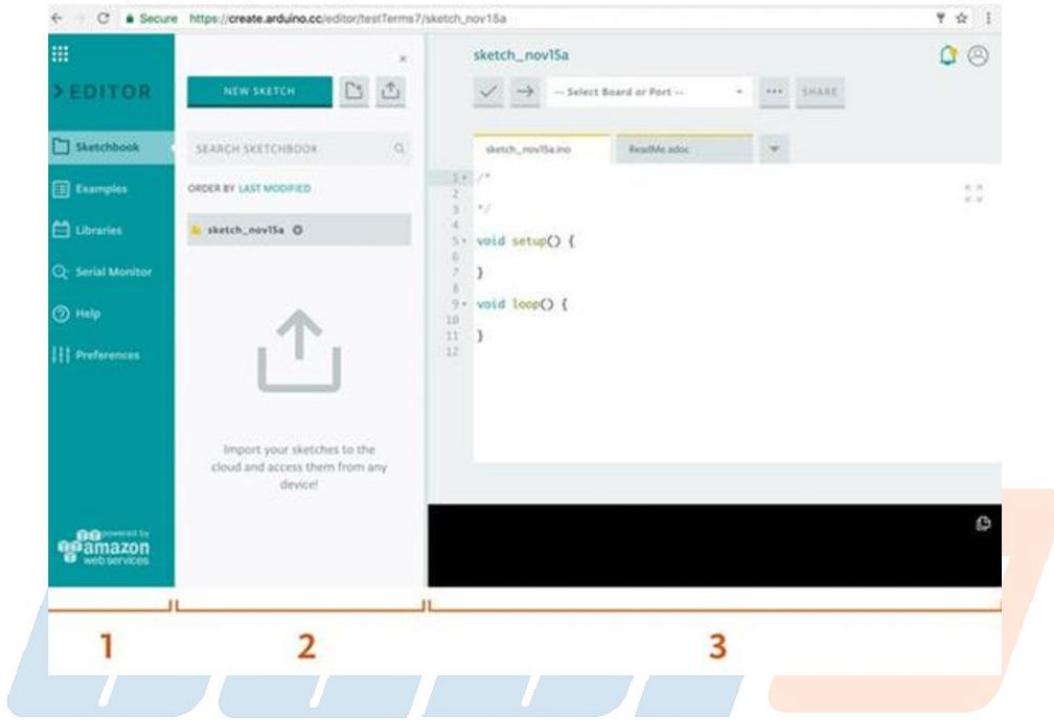


La primera columna permite navegar entre:

- Tu Sketchbook: una colección de todos los sketches.
- Ejemplos: Sketches que demuestran todos los comandos básicos de Arduino.
- Librerías: Son paquetes que pueden ser incluidos al sketch para proveer funcionalidades extras.
- Monitor Serial: Permite recibir y enviar información a la placa a través del cable USB.
- Help: Son links de ayuda y un glosario sobre los términos y conceptos de Arduino.
- Preferencias: Es una opción que permite personalizar la vista y el comportamiento del editor, tal como el tamaño de la fuente y el color del tema.

Al seleccionar una de estas opciones se mostrarán en el side panel los ítems (segunda columna).

La tercera columna es el code area donde se podrá escribir, verificar y programar código en las placas, guardar sketches en la nube y compartirlos.



Como programar un "Blink" desde el editor web:

Conectar el Arduino a la computadora. Seleccionar el tipo de Arduino y el puerto serial desde el select despegable dentro del code area.

En ejemplos seleccionar “Basics” y posteriormente “Blink”. El sketch del código Blink será mostrado en el code area. Para programarlo al Arduino presionar el botón “Upload” (La flecha hacia la derecha a la izquierda del select despegable). Si la programación fue exitosa mostrará un mensaje “Success: Done uploading”.

```

16 modified 2 Sep 2016
17 by Arturo Guadalupi
18 */
19
20 // the setup function runs once when you press reset or power the board
21 void setup() {
22     // initialize digital pin LED_BUILTIN as an output.
23     pinMode(LED_BUILTIN, OUTPUT);
24 }
25
26 // the loop function runs over and over again forever
27 void loop() {
28     digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
29     delay(1000);                      // wait for a second
30     digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage
31     delay(1000);                      // wait for a second
32 }
33
34

```

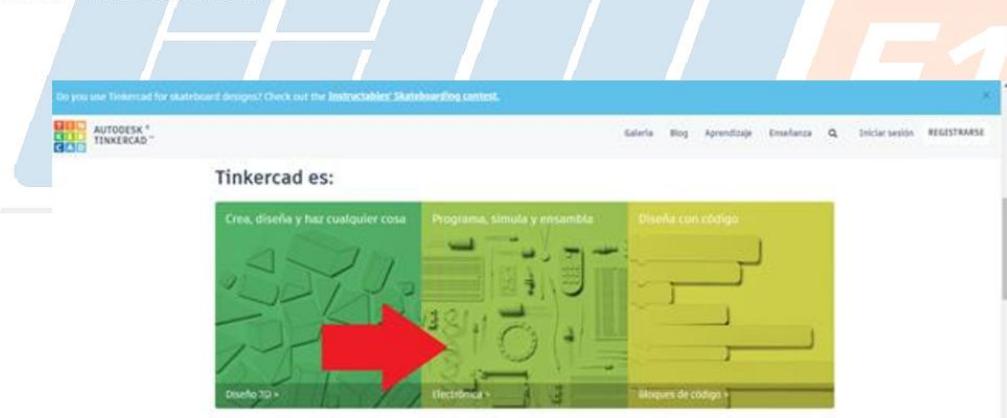
## 2.2.- Emulador

Para trabajar con Arduino sin tener físicamente una placa se puede usar un emulador de software. Existen muchas opciones en el mercado, una de ellas es TinkerCad.

Es una colección de software de herramientas online y gratuita que ayuda a la gente del mundo a pensar, crear y diseñar. Es la introducción ideal a AutoDesl, el líder en diseño 3D, ingeniería y software de entretenimiento



En la página de tinkerCad, en la sección de electrónica dar “click” para ingresar y poder utilizar tal herramienta.



Dar “click” en el botón Comenzar a utilizar tinkerCad, posteriormente indicará que se registre o se cree una cuenta, completar el registro e ingresar a “circuits” para empezar a diseñar un circuito.

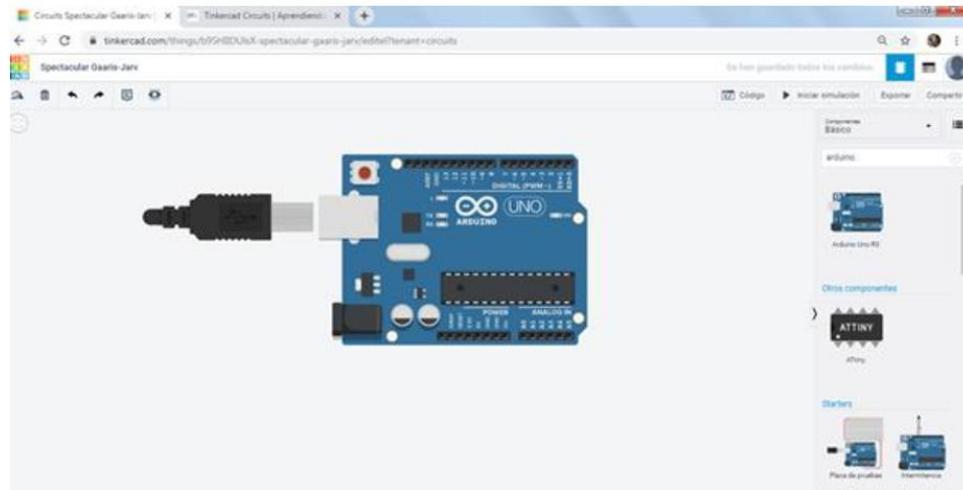


Es tan fácil como...

Una vez completado los pasos anteriores se mostrará un área de trabajo en el cual del lado derecho habrá un panel con componentes Electrónicos los cual se pueden arrastrar al área del trabajo en caso de que se quieran usar:



Buscar el Arduino para utilizarlo:



Una vez el Arduino se encuentre en el área de trabajo, en la parte superior derecha en “código” dar click en el panel, desplegar el select y cambiar el tipo de código de “Bloque” a “Texto”, se podrá visualizar el código por defecto que da tinkerCad que corresponde a un “Blink”. Iniciar simulación y visualizar como empieza a encender y apagar el led del Arduino.

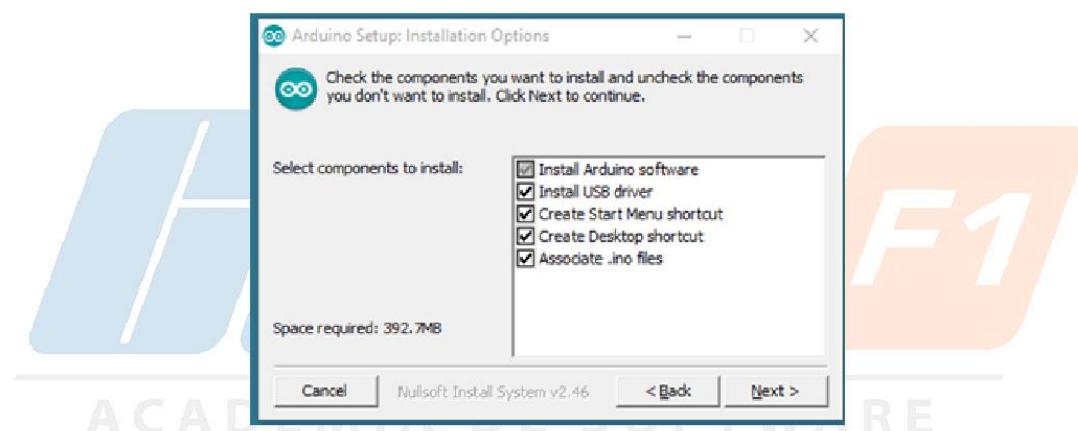
A

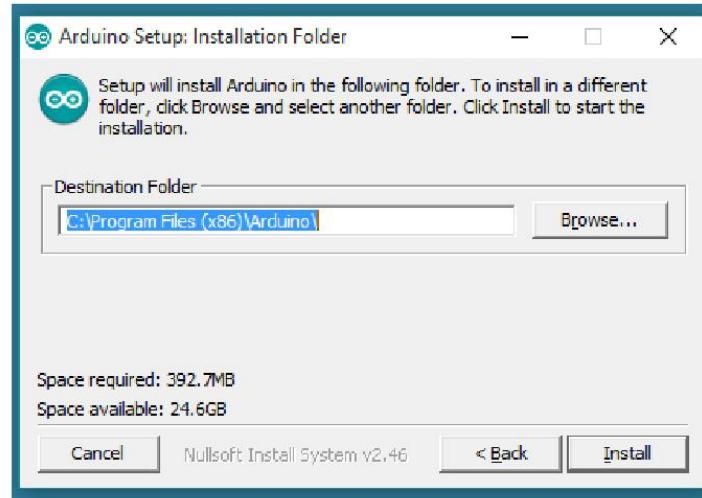
Código	Iniciar simulación	Exportar	Compartir
Texto			1 (Arduino Uno R3)
<pre> 1 void setup() 2 { 3     pinMode(13, OUTPUT); 4 } 5 6 void loop() 7 { 8     digitalWrite(13, HIGH); 9     delay(1000); // Wait for 1000 millisecond(s) 10    digitalWrite(13, LOW); 11    delay(1000); // Wait for 1000 millisecond(s) 12 }</pre>			

### 2.3.- Configuración Del Entorno de Escritorio

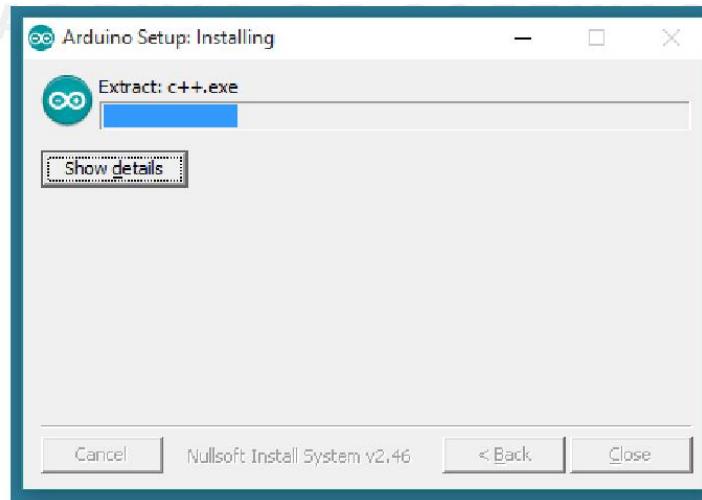
Si se desea trabajar con un IDE de escritorio, la opción más básica es descargar la herramienta provista en el sitio oficial de Arduino. Este Software es de código abierto y facilita escribir y transferir códigos a la placa. Está disponible para Windows, MAC OS X y Linux. El entorno está programado en Java, basado en procesamiento y software de código abierto. Desde la página oficial de Arduino podremos descargarlo.

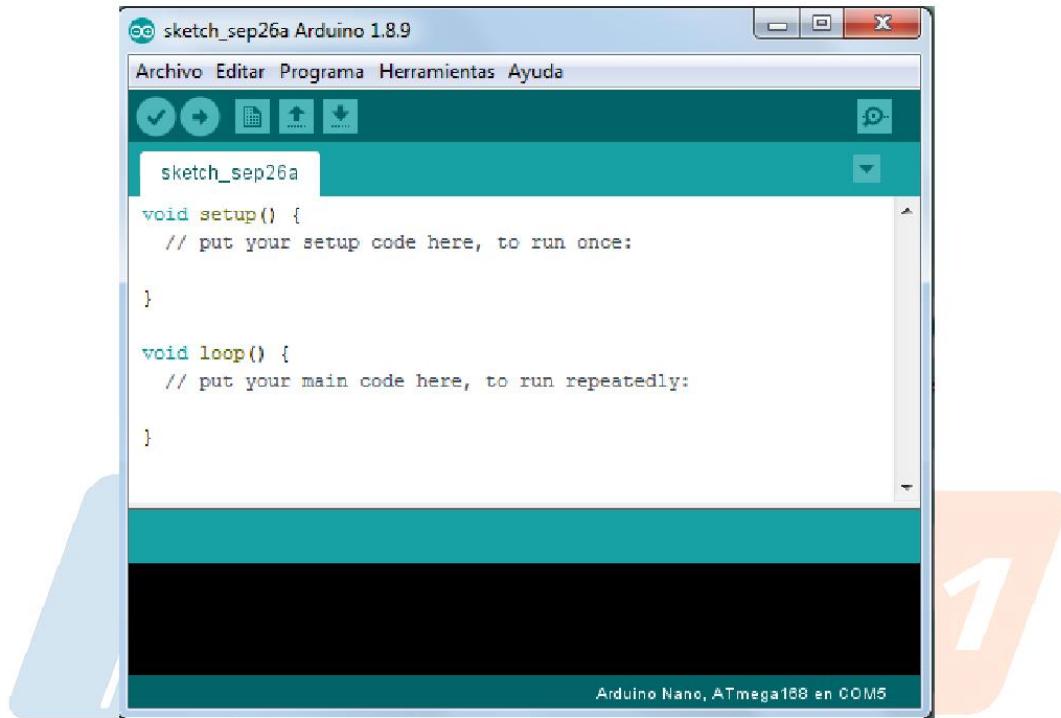
Cuando la descarga finalice, se procede con la instalación, permitir la instalación de los drivers en caso de que se obtenga una ventana de advertencia del sistema operativo.



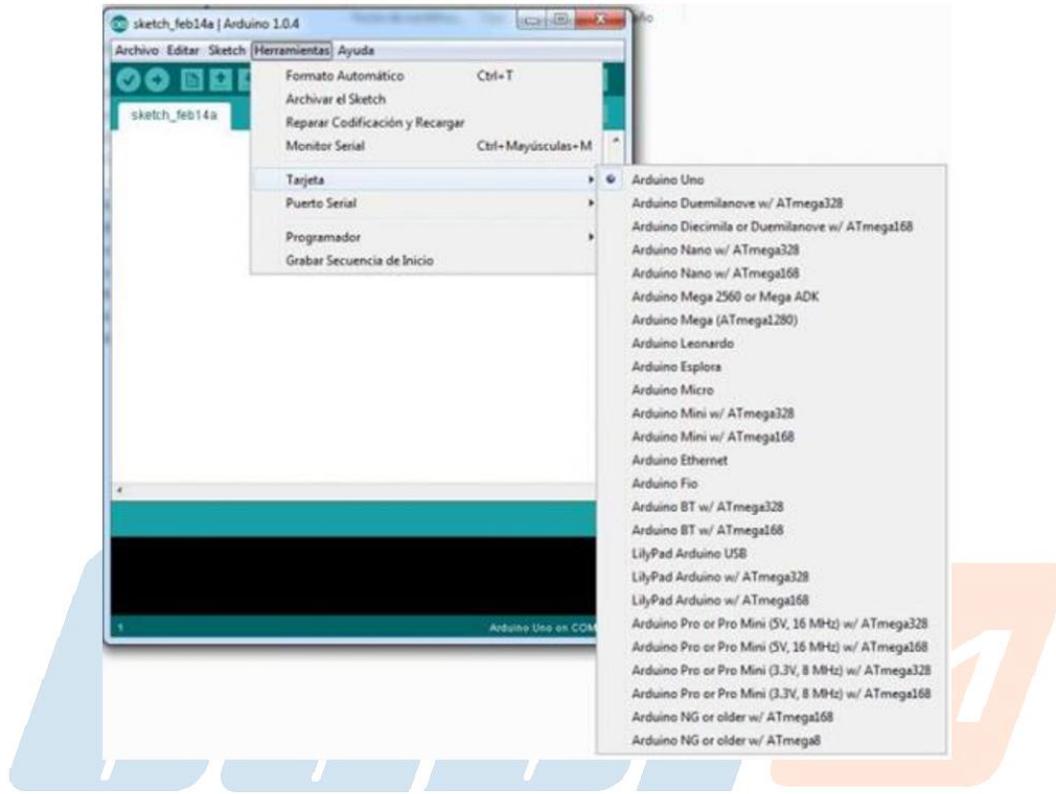


Escoger el directorio donde se desea instalar la aplicación (Se recomienda dejar por defecto). El proceso extraerá e instalará todos los archivos requeridos para ejecutar apropiadamente el IDE. Al completarse la instalación se podrá acceder a la aplicación a través del acceso directo creado en el escritorio o a través del ejecutable dentro de la carpeta donde fue instalado Arduino IDE.



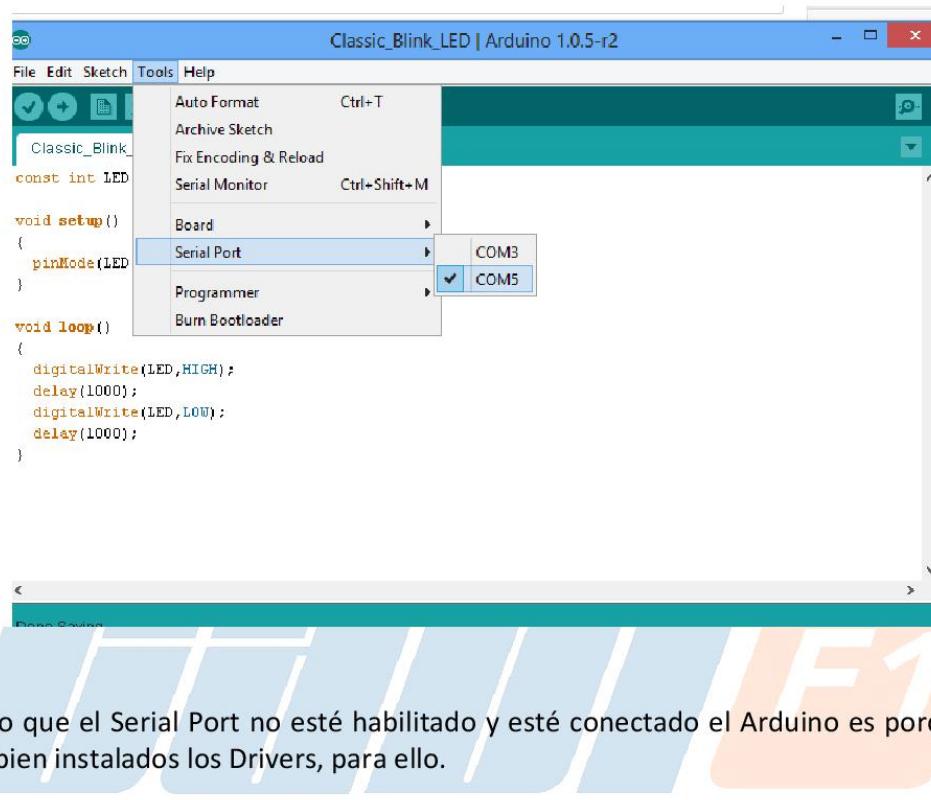


Antes de programar, se deberá de seleccionar en el Sketch con que placa Arduino se está trabajando y a que puerto USB de del ordenador está conectado el Arduino, para ello ir a la pestaña "Herramientas" y dentro de esta pestaña se podrá ver el apartado "Tarjeta" y "Puerto Serial", seleccionar el modelo del Arduino y el puerto USB al que está conectado. Normalmente el puerto USB es detectado automáticamente, pero si hay conectados varios Arduinos al PC, Habrá que seleccionar el puerto.



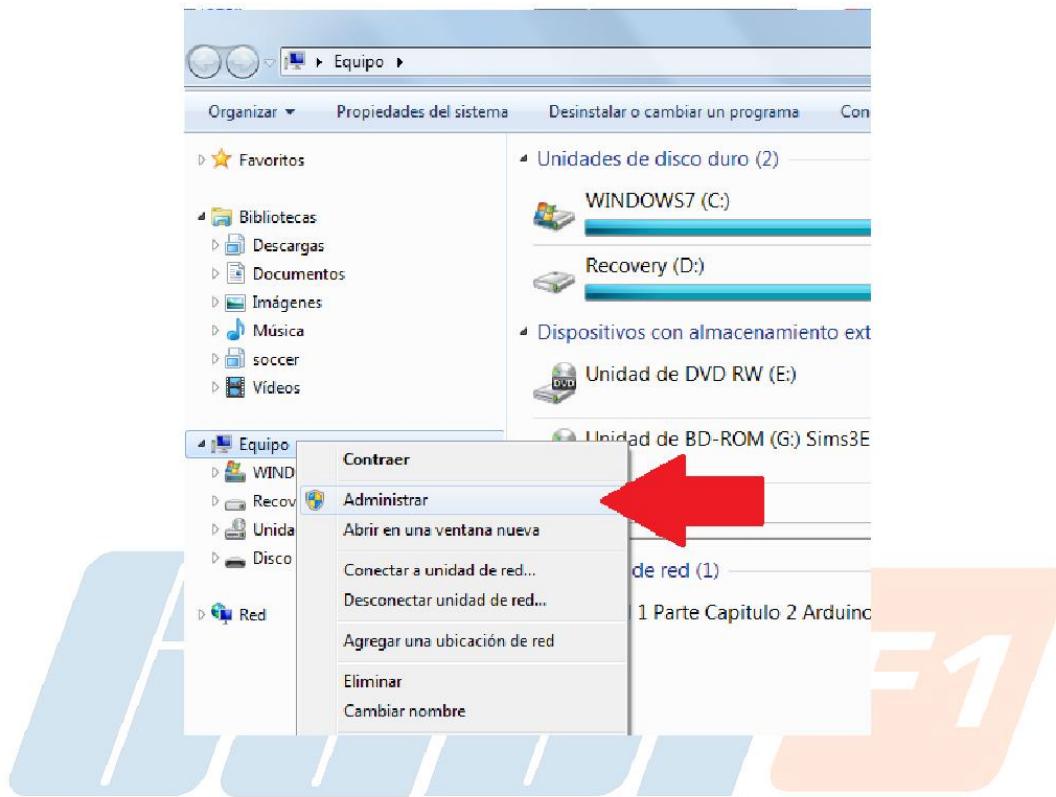
Verificar que el Arduino está siendo reconocido por la computadora:

En “puerto serial” se habilitará y mostrará el puerto que está siendo usado, en el cual está conectado el Arduino. Seleccionar el puerto con el que esté conectado el Arduino al que se quiere programar.



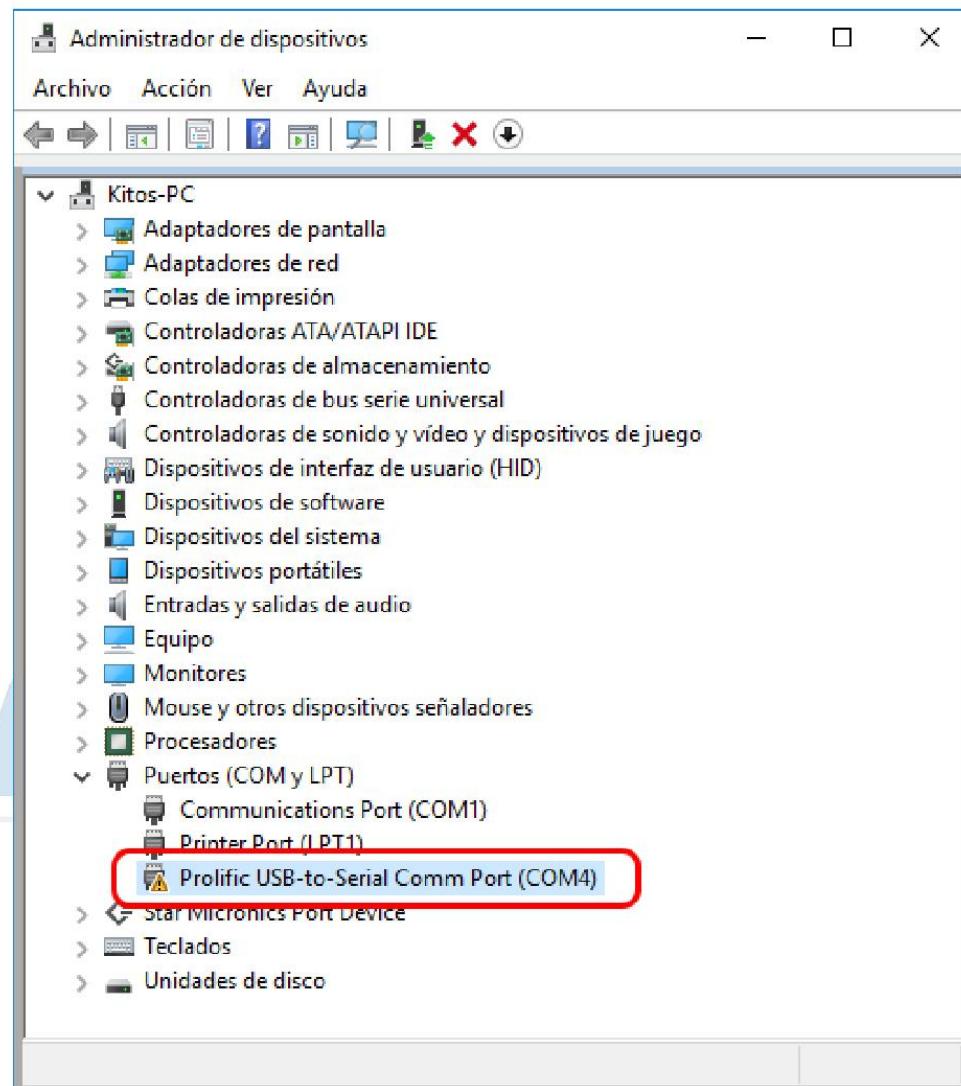
En caso que el Serial Port no esté habilitado y esté conectado el Arduino es porque no están bien instalados los Drivers, para ello.

Abrir la Administración de equipos.

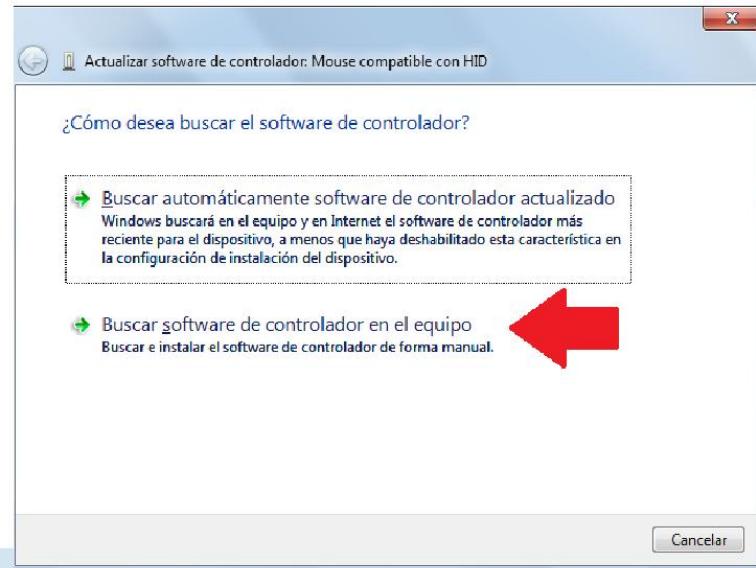


Dentro del administrador de equipos, Dar “click” en administrador dispositivos.

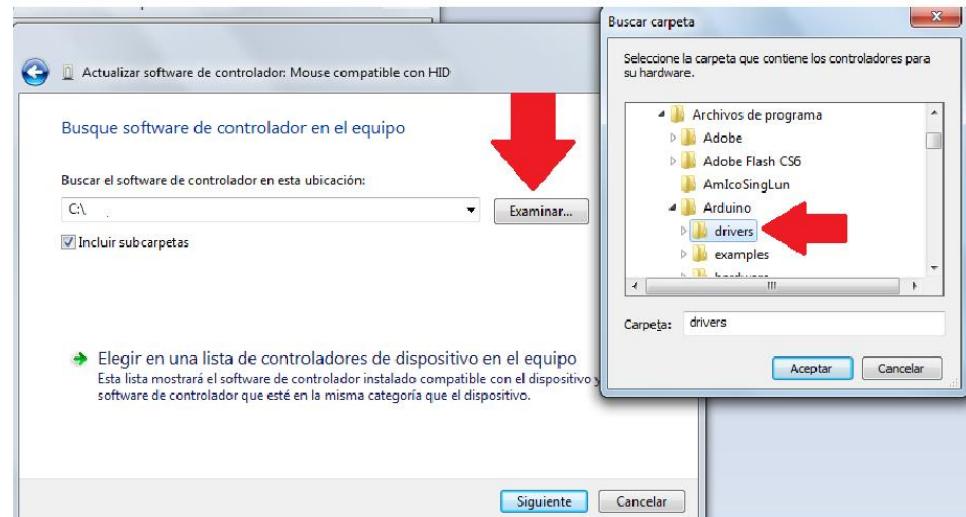
Dar Click derecho sobre el dispositivo no reconocido y “Propiedades”. Una vez allí en la pestaña “Controlador”, ir a “actualizar controlador”.



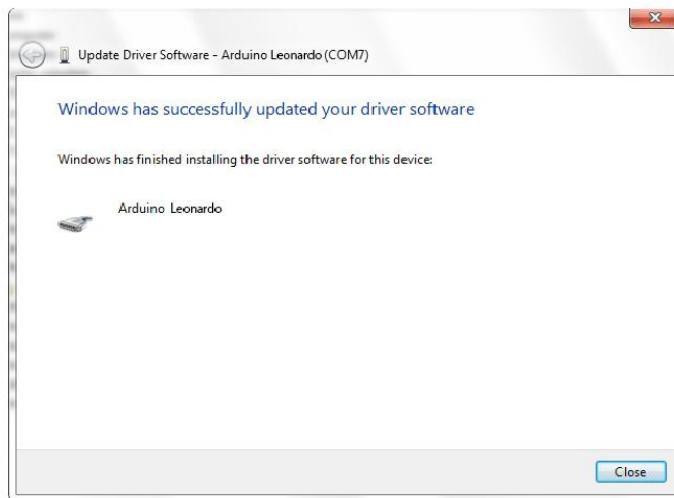
Escoger la segunda opción, “Buscar el software de controlador en el equipo”.



Una vez dentro, seleccionar "Examinar" y buscar la ruta de la carpeta "drivers" dentro de la carpeta donde se descargó la carpeta "Arduino". Aceptar y Siguiente. Después de haber instalado correctamente el driver, se habilitará la opción "serial port" en el IDE.

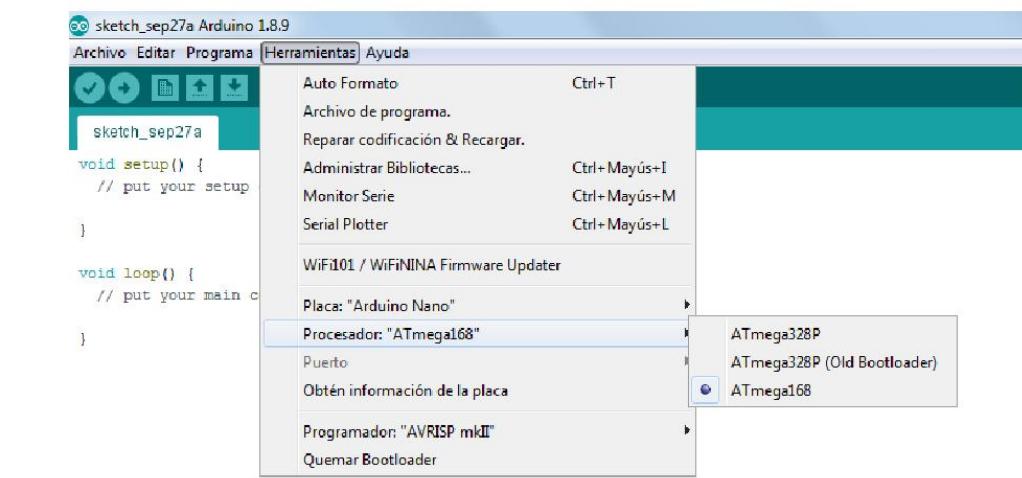


Después de haber instalado correctamente el driver, se habilitará la opción “serial port” en el IDE.



### Seleccionar Procesador de la placa Arduino

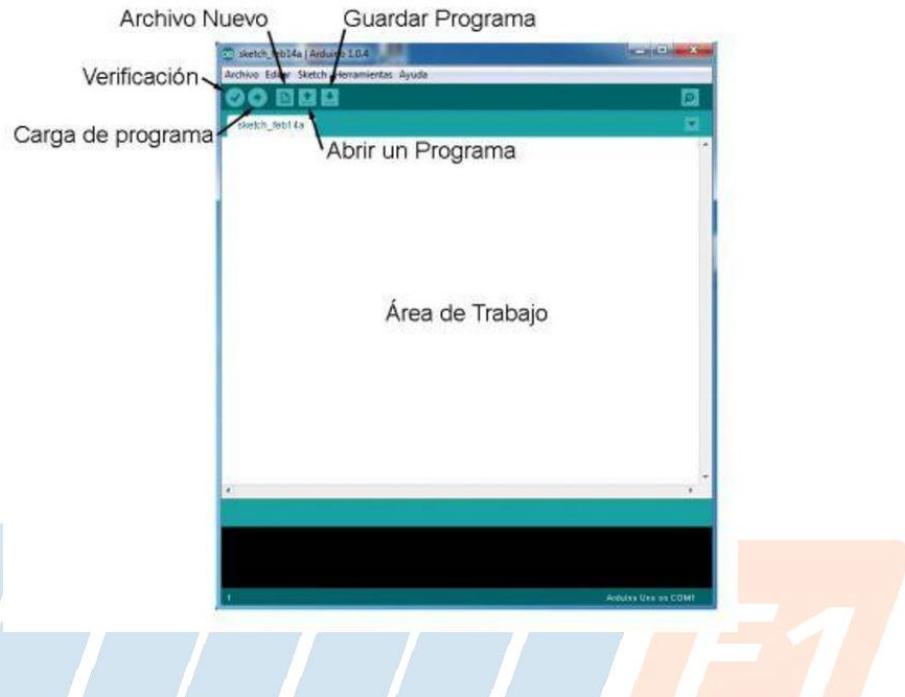
Dependiendo del microprocesador se debe seleccionar el que posee el Arduino. Este se puede visualizar en la placa del Arduino. Para algunos casos del ATmega328P se debe seleccionar el de Old Bootloader.



## Capítulo 3. INICIANDO LA PROGRAMACIÓN EN ARDUINO

### 3.1.- Compilación y Transferencia

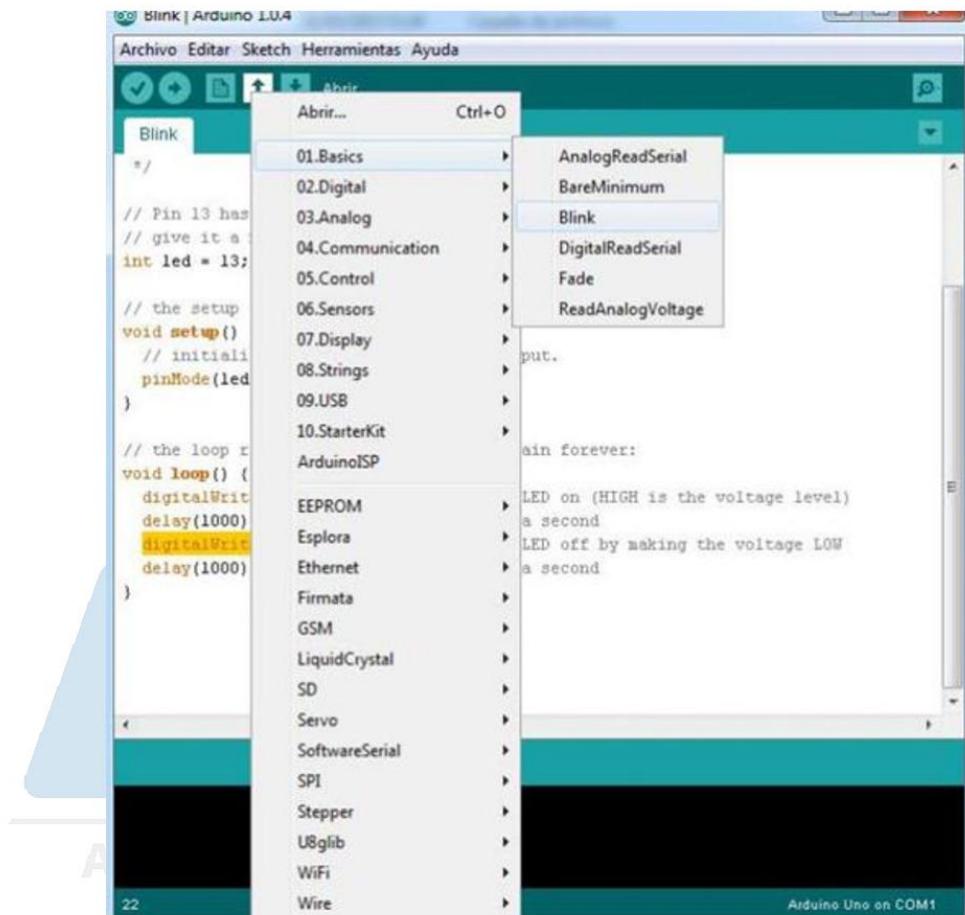
- Área de trabajo: Esta es el área donde se deberá escribir el programa que se quiere ejecutar en el Arduino.
- Verificación: Verifica que la sintaxis del código es correcta y que se ha cometido ningún error pulsando sobre este ícono, si hay algún error, no se cargará el programa en el Arduino.
- Carga de programa: Cuando esté listo el programa y no exista ningún error, se cargará el programa al Arduino pulsando sobre este botón.
- Archivo nuevo: Abrirá una nueva área de trabajo.
- Abrir un programa: Al pulsar sobre este botón, Se desplegará la opción de abrir un archivo desde una ubicación específica o cargar en nuestra área de trabajo una serie de programas o librerías ya creadas y que tiene Arduino por defecto.
- Guardar programa: Guardará en una ubicación especificada por el usuario el contenido del área de trabajo



Si ya se han instalado los drives del Arduino, abierto el Sketch y configurado tanto la placa que se trabajará como el puerto del ordenador al que está conectado, ya está todo listo para poder cargar el primer programa.

Como ejemplo y a modo de verificar que la placa funciona correctamente, se carga un programa que trae por defecto el software de Arduino y que se llama "blink" (en español "parpadeo"), este es un sencillo programa que lo único que va a hacer es poner a parpadear un led que está asociado a la patilla 13 del Arduino.

Para cargar este programa en el área de trabajo, pulsar sobre el icono "abrir" y se mostrará una lista con varios programas que trae Arduino por defecto. Cargar el programa "Blink" que se encuentra dentro de "Basics".



Una vez cargado se podrá ver la estructura básica del programa.

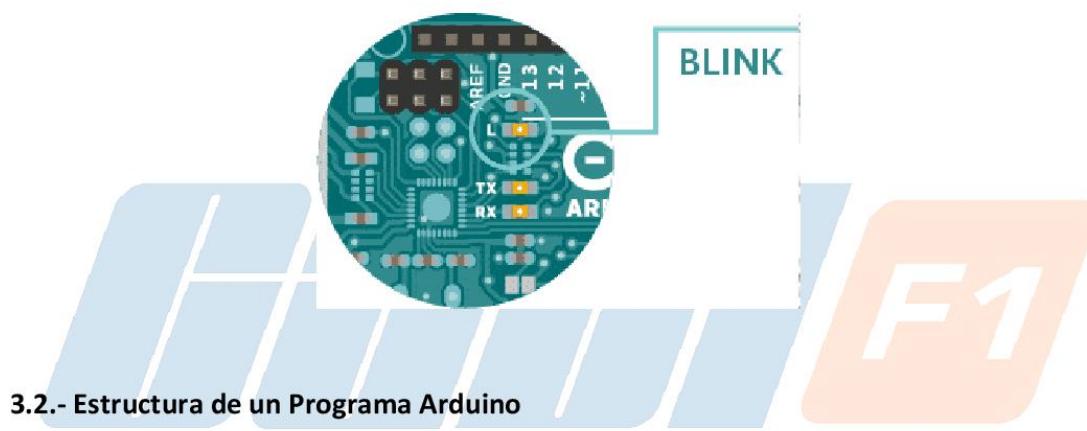
Se puede observar como enciende y apaga el led, para comenzar a entender el código, para encender el led, usa la instrucción "digitalWrite(led, HIGH);", espera 1000 milisegundos (1 segundo) "delay(1000);" y lo apaga "digitalWrite(led, LOW);" esperando otros 1000 milisegundos apagado.

Con el programa en el área de trabajo, dar "cargar" para pasar el código del programa al Arduino, ver como parpadean rápidamente los leds "TX" y "RX" de la placa que indican que se está llevando a cabo la comunicación entre el ordenador y Arduino. Una vez

cargado el programa, ver como el led asociado a la patilla 13, empieza a parpadear a intervalos de 1 segundo.

Se puede cambiar el tiempo que el led permanece encendido o apagado modificando valor que hay encerrado entre paréntesis en la función "delay".

Una vez completada la programación, Se podrá ver como en la placa parpadea un LED con la L a su lado. Si este es el caso, la programación a la placa fue exitosa.



### 3.2.- Estructura de un Programa Arduino

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes principales que por defecto son agregadas a cualquier programa:

- `setup()`: constituye la preparación del programa y `loop()` es la ejecución. En la función `Setup()` se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar el `pinMode` (p. ej. si un determinado pin digital es de entrada o salida) e inicializar la comunicación serie.
- `loop()`: incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.) de ahí su nombre `loop(bucle)`.

Ejemplo de un programa basico

The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_sep26a Arduino 1.8.9". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for file operations. The main window displays the following code:

```
void setup() {
    pinMode(pin, OUTPUT); //Establece 'pin' como salida
}

void loop() {
    digitalWrite(pin, HIGH); // Activa 'pin'
    delay(1000);           // Pausa un segundo
    digitalWrite(pin, LOW); // Desactiva 'pin'
    delay(1000);
}
```

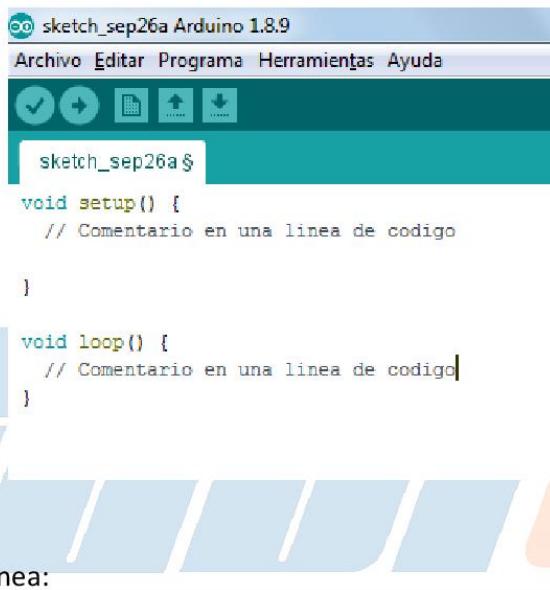
The code consists of two functions: setup() and loop(). The setup() function sets pin as an output. The loop() function alternates the pin between HIGH and LOW states every second. At the bottom of the IDE, there is a status bar with the text "Arduino Nano, ATmega168 en COM5" and a page number "2".

### 3.3.- Comentarios

Los comentarios son líneas en el programa que se utilizan para informar o documentar la forma en que funciona el programa. Son ignorados por el compilador y no se exportan al procesador, por lo que no ocupan ningún espacio. Hay 2 tipos de comentarios: de línea y de bloque.

- Comentarios en una línea del código:

Para poder hacer un comentario en una línea de código se debe utilizar // (doble slash) antes del texto que queremos que sea el comentario. Todo contenido después de las barras dentro de la misma línea será tomado como un comentario.



The screenshot shows the Arduino IDE interface with the title bar "sketch\_sep26a Arduino 1.8.9". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for save, open, upload, and download. The code editor window contains the following code:

```
void setup() {  
    // Comentario en una linea de codigo  
}  
  
void loop() {  
    // Comentario en una linea de codigo  
}
```

A blue banner at the bottom of the slide contains the text "- Comentarios Multilínea:" followed by the F1 Academy logo.

Para hacer un comentario multilínea se debe encerrar entre /\* \*/ el comentario, es necesario cerrar el comentario con \*/ porque sino todas las líneas siguientes del código será tomado como comentario.



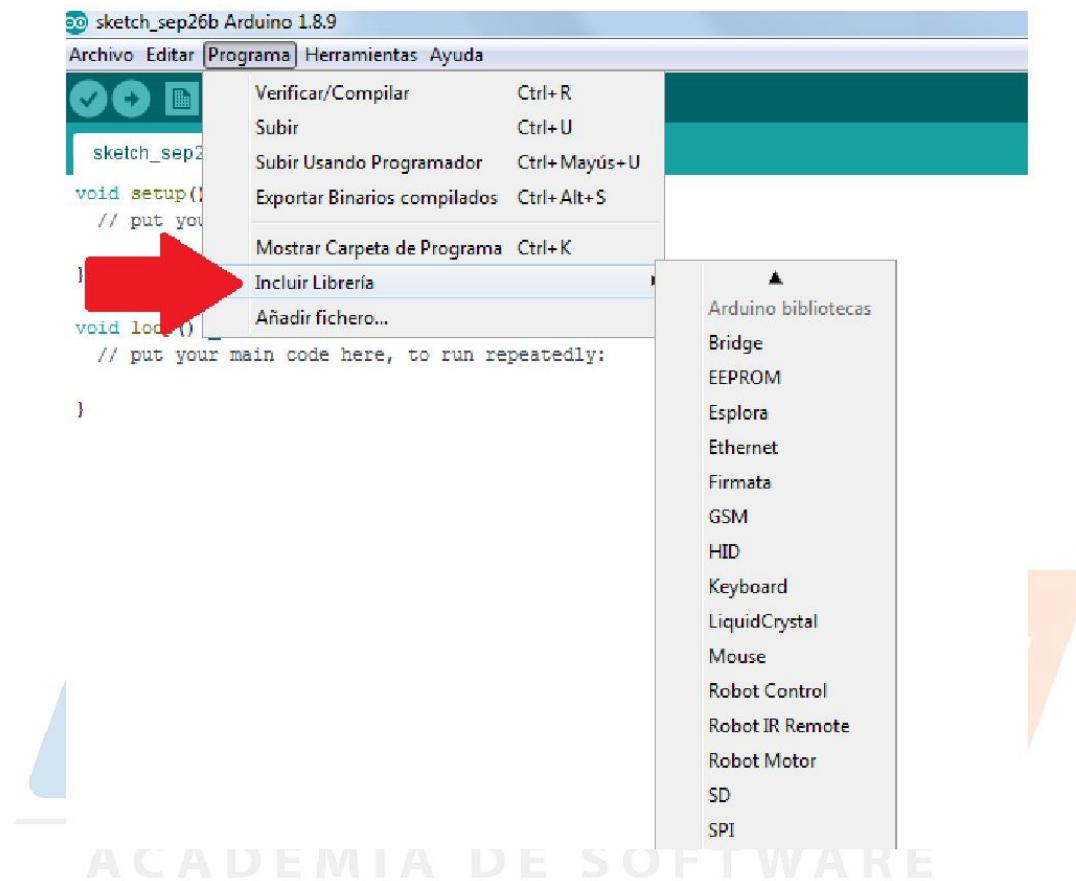
```
sketch_sep26a Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
sketch_sep26a §

void setup() {
    /* Esto
       es un comentario
       multilinea
       en Arduino
    */
}

void loop() {
    /* Esto
       es un comentario
       multilinea
       en Arduino
    */
}
```

### 3.4.- Librerías

El entorno de Arduino puede ser extendido a través del uso de librerías, como la mayoría de otros lenguajes de programación. Las librerías proveen funcionalidades extra para el uso de sketches, tal como, manejar hardware o manipular información. Para usar una librería en un sketch, seleccionar Programa > Incluir librería (Sketch > Import library).



## Librerías Estándar

- EEPROM: Incorpora las instrucciones necesarias para gestionar las memorias no volatil.
- Ethernet: Se emplea para usar la conexión de internet del Arduino Ethernet Shield, Arduino Ethernet Shield 2 y Arduino Leonardo ETH.
- Firmata: Es un protocolo genérico para la comunicación con microcontroladores desde software instalado en un ordenador.
- GSM: Permite a una placa Arduino hacer la mayoría de las operaciones que se pueden hacer con un teléfono GSM.
- LiquidCrystal: Permite a un Arduino controlar LiquidCrystal displays (LCDs).
- SD: Permite leer y guardar contenido en una SD.

- Servo: Implementa la clase del mismo nombre, destinada a facilitar la comunicación de Arduino con servomotores.
- Stepper: Permite controlar motores de paso unipolares o bipolares.
- TFT: Permite al arduino conectarse con el TFT LCD para dibujar líneas, formas e imágenes, entre otros.
- WiFi: Con el Arduino Wifi Shield, permite a la placa arduino conectarse a internet.

Librerías USB :

- Keyboard: Las funciones de esta librería permite enviar KeyStrokes a una computadora a través del puerto
- micro USB.
- Mouse: Las funciones de la librería mouse permiten controlar el movimiento de un cursor en una computadora
- conectada.

