

Introducción

La arena de bloques de Blockimino es la principal pestaña donde el usuario tiene la capacidad de trabajar en sus proyectos relacionados a la programación de Arduino.

Esta arena es donde se puede crear y modificar el código por medio de bloques, los cuales cuentan con una organización de distintos temas y usos de librerías para una mejor practicidad y rapidez a la hora de utilizarlos

Conceptos básicos de programación por bloques

La programación por bloques es un enfoque amigable y visual que te permite construir programas utilizando bloques predefinidos que representan instrucciones y acciones. En lugar de escribir líneas de código complicadas, simplemente arrastras y sueltas estos bloques en una interfaz intuitiva.

los conceptos más básicos que se deben comprender son:

Bloques: Los bloques son los elementos fundamentales de la programación a bloques. Representan diferentes comandos, acciones o instrucciones en el programa. Los bloques están diseñados para encajar unos con otros, creando una estructura lógica.

Paletas: Las paletas son paneles o menús que contienen diferentes categorías de bloques. Cada categoría agrupa bloques relacionados por su función o propósito. Al seleccionar un bloque de la paleta, puedes arrastrarlo al área de trabajo.

Área de trabajo: El área de trabajo es donde se construye el programa. Es un espacio en blanco donde puedes arrastrar y soltar bloques para crear instrucciones y algoritmos. Los bloques se conectan entre sí para formar una secuencia lógica.

Conexiones: Los bloques tienen conectores o encajes que permiten conectarlos entre sí. Estas conexiones pueden ser de diferentes tipos, como conectores de inicio, conectores de finalización o conectores de enlace. Al unir los bloques de forma adecuada, se establecen las relaciones y la lógica del programa.

Secuencia: En la programación a bloques, la secuencia es la orden en que los bloques están conectados entre sí. Los bloques se ejecutan de arriba hacia abajo, de izquierda a derecha, siguiendo la secuencia establecida. Esto generalmente determina el flujo de ejecución del programa.

Parámetros: Los bloques pueden tener parámetros o valores asociados. Estos parámetros permiten personalizar o ajustar el comportamiento de los bloques. Por ejemplo, un bloque de movimiento puede tener un parámetro de velocidad, y puedes especificar el valor de la velocidad al arrastrar el bloque.

Ejecución: Una vez que se ha construido el programa utilizando bloques en el área de trabajo, puedes ejecutarlo para ver cómo se comporta. Al hacer clic en un botón de ejecución, el programa se procesa y realiza las acciones o tareas definidas por los bloques.

Diferencias de Blockimino con otros programas a base de bloques

Blockimino se diferencia de programas similares principalmente por contener algunos cambios dentro de los conceptos básicos de la programación a bloques, además de brindar ciertas funcionalidades y servicios con el fin de mejorar la experiencia del usuario:

- 1.- Cuenta con una breve descripción del bloque y su funcionamiento al posar el cursor sobre el mismo por un momento.
- 2.- Presenta dos versiones para programar, estos son el modo principiante y el modo avanzado. La principal diferencia entre ambos es la capacidad de entendimiento que brindan al usuario al utilizar nombres de bloques y descripciones más sencillas al utilizar el modo principiante, mientras que en el avanzado se introduce al usuario al lenguaje que utiliza Arduino en sus proyectos.
- 3.- Se tiene la capacidad de cargar y descargar proyectos desde la arena de bloques, de esta forma los proyectos creados tienen la posibilidad de ser usados en varios lugares, compartidos y utilizados sin limitaciones.
- 4.- Cuenta con un sistema que registra errores e informa de los campos de mejora para aumentar el crecimiento personal.
- 5.- Cuenta con su propia wiki, donde el usuario puede tener acceso cuando lo desee para poder comprender el funcionamiento de un bloque, además de algunos ejemplos prácticos en videos sobre los mismos.
- 6.- Existe un botón con la leyenda “verificar”, el cual se encarga de informar todos los errores posibles que se encuentran presentes dentro del proyecto y que no permiten su correcta ejecución. De esta manera es sencillo corregir errores y aprender más rápido.
- 7.- Blockimino es un programa diseñado para trabajar como una forma de programar en Arduino. Dentro de Arduino existen dos principales espacios donde se colocan las instrucciones, llamados loop y setup. Para facilitar el uso de estos campos Blockimino trabaja con un sistema de prioridad en el cual se altera el orden de los bloques automáticamente, por lo que la colocación de estos dentro de la arena de bloques no afectará su correcto funcionamiento.

Orden de prioridad y colores

En Blockimino manejamos el código con un cierto orden, por defecto los bloques ya están designados si van al loop o setup de Arduino, los bloques que floten de manera libre su orden es por prioridad, si es incluye tendrá la inclusión del módulo junto con sus segmentos de declaración de variables globales y defines necesarios, después en prioridad están los defines y variables globales, después sigue el setup, loop y finalmente las funciones.

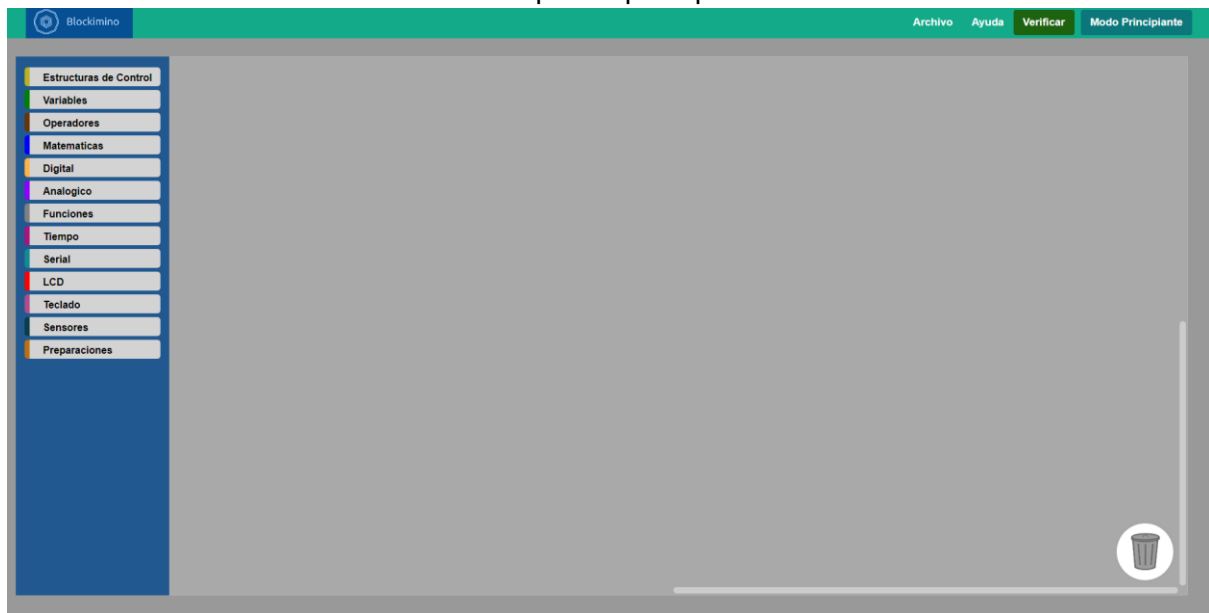
En caso de tener varios bloques de la misma prioridad, se tomará en el orden que están anidados, en caso de no estar anidados entonces el primero que se coloca es el que sale en primera posición.

Los colores son distintos por cada categoría y los colores de las preparaciones son naranja para los setup y naranja oscuro para bloques que son preparaciones, pero no están en el setup, como lo son los includes y defines. Por último, cada módulo tiene sus respectivos setups y includes según los requiere.

Todos los bloques excluyendo las variables, setup, includes y defines son por defecto de la parte de loop del código de Arduino. Si se tiene un bloque de función, solamente el código dentro de la función se tomará como parte de este. En las funciones no se permite tener segmentos flotando ya que solamente el código anidado a la función será parte de dicha función por lo que el orden del código generado seguirá el orden puesto por los bloques.

Conceptos básicos y descripción de la interfaz

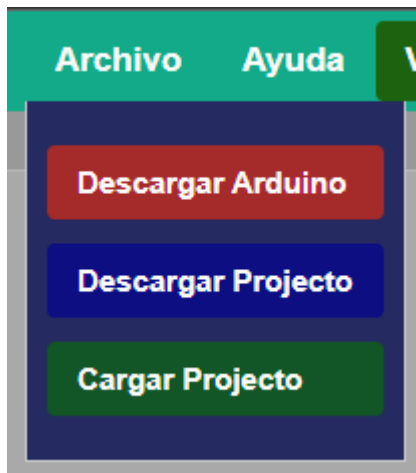
La interfaz se encuentra dividida en tres partes principales:



1.- Barra de tareas: En la parte superior y de un color verdoso se encuentra la barra de tareas, en esta se encuentran las opciones de carga y descarga de proyectos, la ayuda de la wiki, la verificación de correcciones del programa, y el cambio entre modos principiantes y avanzados.



“Archivo” cuenta con un pequeño menú desplegable al pasar el cursor sobre el mismo el cual incluye 3 botones:

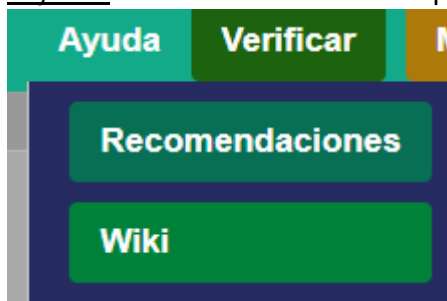


Descargar Arduino es el encargado de descargar el código creado actualmente por los bloques que existan dentro de la arena de bloques en un formato de texto, de esta forma es posible utilizar el código creado copeland y pegando en el compilador de Arduino.

Descargar proyecto es el encargado de descargar el código creado actualmente por los bloques que existan dentro de la arena de bloques en un formato único de blockimino llamado .blckmno, el cual puede ser reutilizado para generar el proyecto actual en bloques dentro de otros dispositivos en los cuales se esté ejecutando el programa

Cargar proyecto se encarga de permitir la subida de proyectos .blckmno y de no alterar su funcionamiento dentro de la página de arena de bloques de Blockimino.

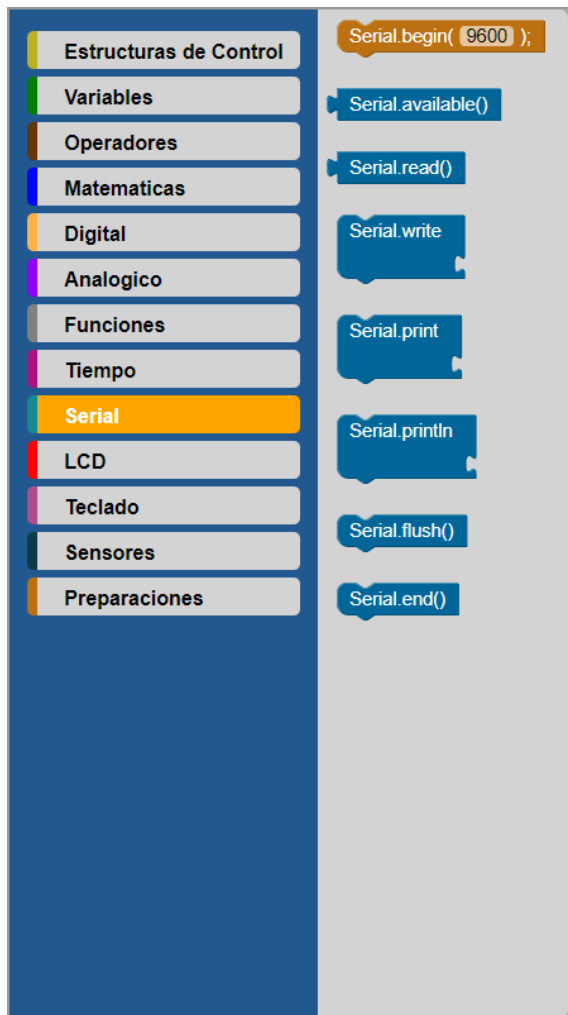
“Ayuda” También cuenta con un par de opciones:



Recomendaciones muestra un pequeño mensaje con los temas de los cuales el usuario ha tenido más problemas al programar, y los cuales le conviene repasar para mejorar progresivamente.

Wiki es una página en la que se encuentra la información de todos los bloques, como es que estos funcionan, en conjunto con videos que pueden ser de gran ayuda para el usuario.

2.- Paleta de categorías: Esta paleta de color azul marino se encuentra en el extremo izquierdo de la página, contiene 13 categorías de bloques separados en menús desplegables, los cuales al ser cliqueados revelan los bloques relacionados a esa categoría dispuestos para ser utilizados dentro de la arena de bloques.



3.- Arena de bloques: Este espacio vacío de color gris que ocupa la mayor parte de la página es donde se contienen los bloques que representan el código de tu programa. Aquí puedes soltar los bloques que requieras para que estos sean verificados y posteriormente descargados.



Movilización, generación y eliminación de bloques

La forma de generar los bloques es muy sencilla, se deben seguir los siguientes pasos:

- 1.- Hacer clic sobre la categoría deseada en la paleta de categorías.
- 2.- Ubicar el bloque que se desea, algunas categorías contienen varios bloques por lo que puede llegar a ser necesario subir o bajar los bloques con ayuda de la rueda del mouse para encontrar el deseado.
- 3.- arrastrar y soltar dentro de la arena de bloques.

Los bloques solamente pueden ser colocados dentro de la arena de bloques. Dentro de la arena se encuentra un pequeño bote de basura, cuando se desee eliminar uno o varios bloques el usuario debe arrastrar el bloque deseado y eliminarlo soltándolo dentro del bote o devuelta a la sección del menú de categorías.

Si por error se elimina o se mueve un bloque no deseado siempre se puede usar la combinación de teclas ctrl+z lo que instantáneamente retrocederá el cambio de posición o la eliminación realizada en los bloques.

Los bloques se pueden arrastrar de forma separada o unida a varios bloques, para lograr esto depende de que parte del bloque sea posicionado el mouse para arrastrarlo. Generalmente se moverá solo un bloque, sin embargo si este bloque contiene bloques dentro del mismo, estos bloques se moverán en él, a menos de que estos bloques sean elegidos de forma separada.

Tipos de Arduino y de señales

En Blockimino, el término "tipos de Arduino" se refiere a las placas de desarrollo de hardware que están diseñadas para programar y controlar dispositivos electrónicos. Los dos modelos de Arduino más populares son Arduino Uno y Arduino Mega. A continuación, te explicaré brevemente cada uno de ellos y también te hablaré sobre los pines de conexión utilizados en estas placas.

Arduino Uno:

Arduino Uno es una placa de desarrollo de tamaño compacto y fácil de usar. Está basada en el microcontrolador ATmega328P de Atmel. Es ampliamente utilizada debido a su simplicidad y versatilidad. El Uno tiene 14 pines digitales de entrada/salida (6 de ellos pueden usarse como salidas PWM), 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB para la programación y comunicación, y una fuente de alimentación mediante un conector de alimentación o la conexión USB. Es una excelente opción para proyectos más pequeños y aplicaciones básicas.

Arduino Mega:

Arduino Mega es una placa de desarrollo más grande y avanzada en comparación con el Uno. Está basada en el microcontrolador ATmega2560 de Atmel. La Mega tiene 54 pines digitales de entrada/salida (15 de ellos pueden usarse como salidas PWM), 16 entradas

analógicas, un oscilador de cristal de 16 MHz, múltiples puertos de comunicación serial, una conexión USB y opciones adicionales de memoria y almacenamiento. La Mega es ideal para proyectos más grandes y complejos que requieren un mayor número de pines de entrada/salida y capacidad de procesamiento.

En cuanto a los pines de conexión, Arduino utiliza un "pinout" o diseño de pin específico para facilitar la conexión de dispositivos externos. Los pines digitales pueden ser configurados como entradas o salidas, lo que permite interactuar con componentes electrónicos como sensores, LED, motores, pantallas, etc. Los pines analógicos permiten leer señales analógicas, como sensores de luz o temperatura, y convertirlas en valores digitales para su procesamiento.

El diseño del pinout puede variar según el modelo de Arduino, pero en general, cada pin se identifica con un número y una función específica (entrada, salida, PWM, analógico, digital, etc.).

Es importante tener en cuenta que el nivel lógico alto, representado por el valor 1, corresponde al voltaje de alimentación utilizado en el sistema Arduino. En la mayoría de los casos, este voltaje es de 5V, por lo que cualquier valor de voltaje por encima del umbral específico (generalmente alrededor de 2.0V) se considerará como 1 digital.

Los analógicos se refieren a valores medibles de un rango, en este caso de voltaje, mientras que los digitales se utilizan más en usos donde se ocupa un si o un no, ya que, en digital, por lo general se maneja que si la señal es débil se toma como apagado, si es lo suficientemente fuerte entonces la señal digital es marcada como positiva.

Variables locales y globales

En programación, las variables globales y locales son dos conceptos importantes relacionados con el alcance y la visibilidad de las variables en un programa. A continuación, te explicaré qué son las variables globales y locales y cómo funcionan en el contexto de la programación.

Variables Globales:

Las variables globales son declaradas fuera de cualquier función o bloque de código en un programa. Estas variables son accesibles desde cualquier parte del programa, es decir, tienen un alcance global. Se denominan "globales" porque están disponibles en todo el ámbito del programa. Las variables globales se inicializan una vez y pueden ser utilizadas y modificadas por cualquier función o bloque de código dentro del programa.

Una característica importante de las variables globales es que conservan su valor incluso después de que la función en la que se utilizan haya terminado su ejecución. Esto significa que su valor persiste a lo largo del programa, a menos que se modifique explícitamente. Sin embargo, el uso excesivo de variables globales puede dificultar la comprensión y el mantenimiento del código, ya que cualquier parte del programa puede acceder y modificar su valor.

Variables Locales:

Las variables locales son declaradas dentro de una función o bloque de código específico y solo son accesibles dentro de ese ámbito local. Tienen un alcance limitado y su existencia está restringida a la función o bloque en el que se definen. Esto significa que no se pueden acceder a las variables locales desde fuera de la función o bloque en el que se encuentran.

Una vez que la función o bloque de código en el que se declaran las variables locales ha finalizado su ejecución, las variables locales se eliminan de la memoria y su valor se pierde. Cada vez que la función o bloque se ejecuta nuevamente, se crean nuevas instancias de las variables locales y se les asigna un nuevo valor. Esto permite reutilizar los nombres de las variables en diferentes funciones o bloques sin conflictos.

El uso de variables locales tiene la ventaja de mantener el código modular y más fácil de entender, ya que las variables solo son relevantes dentro del ámbito específico en el que se utilizan. Además, las variables locales ayudan a evitar colisiones o conflictos de nombres con otras partes del programa.

En Blockimino podrás utilizar variables globales si se manejan afuera de todo código y aparecen en color verde, bool siendo más oscuro. Dentro de las funciones existe un apartado para integrar las variables locales las cuales no serán accesibles afuera de dicha función ya que son locales a esa función y solo esa, no son compartidas entre funciones.

Los nombres tanto de las variables como de las funciones en Blockimino está limitado a que siempre sea distinto para que siempre se genere un código correcto y lo mas claro posible. En Arduino es posible poner nombres repetidos en casos particulares, los cuales podrás aprender con la experiencia, un caso, por ejemplo, es la declaración de variables locales que tengan el mismo nombre que alguna global, es valido pero por simplicidad, fuera de los bloques "for" que usan "i" de manera local, no es permitido en Blockimino.

Inclusión de librerías en Arduino

En Arduino, puedes incluir bibliotecas utilizando la directiva `#include`. Esta directiva permite importar código de bibliotecas externas para usar en tu programa Arduino. Para incluir una biblioteca, debes seguir estos pasos:

Encuentra la biblioteca que deseas utilizar. Puedes buscar bibliotecas en el administrador de bibliotecas de Arduino o descargarlas manualmente desde fuentes confiables en línea. Asegúrate de obtener la versión correcta compatible con tu placa Arduino y la versión del IDE de Arduino que estás utilizando.

Abre tu programa Arduino en el IDE de Arduino.

En la parte superior de tu programa, antes de la declaración `void setup()`, agrega la línea `#include <nombre_de_la_biblioteca.h>`, donde `nombre_de_la_biblioteca` es el nombre de la

biblioteca que deseas incluir. El archivo de encabezado de la biblioteca generalmente tiene la extensión.h.

A continuación, puedes utilizar las funciones y clases proporcionadas por la biblioteca en tu programa Arduino.

Es importante mencionar que solo debes incluir una biblioteca una sola vez en tu programa. Esto se debe a que el proceso de compilación de Arduino combina todos los archivos de encabezado (*.h) y código fuente (*.cpp) en un solo archivo antes de compilar. Si incluyes la misma biblioteca múltiples veces, puede haber conflictos y errores de compilación debido a la redefinición de símbolos y duplicación de código.

Forma de uso y compatibilidad de bloques

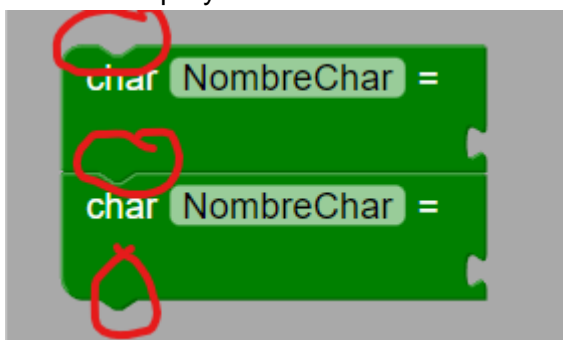
Algunos bloques deben de ser utilizados en conjunto de otros bloques para tener un funcionamiento y generar código, mientras que otros pueden trabajar con varios bloques de forma libre y opcional, pero con algunas excepciones.

Para facilitar y agilizar el aprendizaje, los bloques que requieren de otro bloque en específico para funcionar avisan automáticamente al usuario al momento en que este decide terminar su proyecto, por lo que no interrumpe las actividades de este.

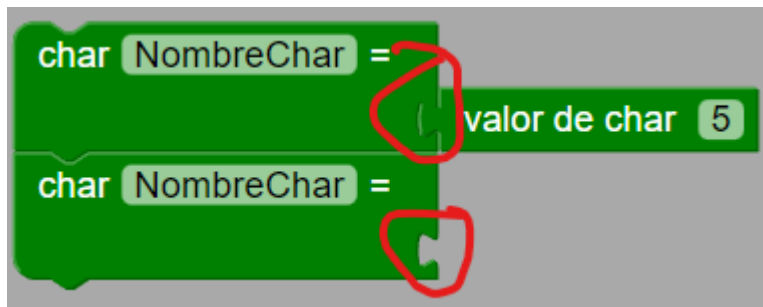
En contraste al caso anterior, para los bloques que no requieren de un bloque en específico para funcionar no mandan ninguna alerta, y si se intentan conectar con bloques incompatibles, estos simplemente negaran la conexión separándose de forma leve del bloque objetivo y algunas veces avisará del error en el momento, mientras que otras veces se podrá observar en la retroalimentación de “verificar”.

Existe 2 principales tipos de conexiones:

1.- Las conexiones de secuencia y unión son las encargadas de conectar la secuencia de los bloques, aunque estos se ordenan automáticamente por prioridad ayuda a mantener un orden en el proyecto.



2.- Las conexiones de rompecabezas o entradas de valor, son las encargadas de recibir bloques que normalmente no generan código, si no que suministran valores para los demás bloques.



Recursos adicionales

Wiki oficial de Blockimino:

-wiki.com

*Consultable desde la arena de bloques en la sección de ayuda

Introducción a la programación a bloques:

-<https://www.youtube.com/watch?v=NH-NqakuVuU>

Gracias por utilizar Blockimino, cualquier pregunta o sugerencia es posible mandarla a través del foro en la sección de ayuda.

