

Meta_Javascript Alpha 1.0

Programación para principiantes

Jose David Cuartas Correa



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA



EDICIÓN DE PRUEBA 1.1: MAYO, 2020
© Fundación Universitaria Los Libertadores
© Jose David Cuartas Correa

LOS LIBERTADORES, FUNDACIÓN UNIVERSITARIA
Bogotá D. C., Colombia
Cra 16 No. 63 A - 68 / Tel. 2 54 47 50
www.ulibertadores.edu.co

Juan Manuel Linares Venegas
Presidente del Claustro

Patricia Martínez Barrios
Rectora

Ángela María Merchán Basabe
Vicerrectora Académica

Jose David Cuartas Correa
Diagramación y diseño portada

INTRODUCCIÓN

Con este texto busco introducir al lector en los aspectos básicos de la primera versión de Meta_Javascript, un lenguaje de meta-programación que desarollé para el principiante de la programación de aplicaciones web. Está basado en Meta_Processing y fue diseñado para ayudarte a crear código Javascript interactivo. Es una iniciativa personal que es influenciada por mi trabajo como director del Laboratorio Hipermedia¹ (Hitec Lab) en la Fundación Universitaria Los Libertadores (Bogotá, Colombia).

Lo que se busca con esta derivación de Meta_Processing es ir creando un ecosistema de aplicaciones que le permitan al usuario programar diferentes plataformas haciendo uso de un mismo lenguaje de programación. En la actualidad es casi indispensable aprender un lenguaje diferente para programar cada plataforma.

Este entorno de programación se fundamenta en las necesidades que identifiqué durante los años que oriente un curso de programación básica para diseñadores gráficos en la Fundación Universitaria Los Libertadores. También responde a parte de mis

¹Laboratorio Hipermedia <http://hiteclab.libertadores.edu.co/>

intereses investigativos desde cuando estaba realizando mi doctorado en Diseño y Creación, el cual finalicé con la tesis: “Programar el mundo en el contexto de las tecnologías libres y las culturas Hacker-Maker. Caso de estudio: Hitec Lab” (cuartas, 2017).

Meta_Javascript es una de varias iniciativas que he estado liderando en búsqueda de desmitificar las tecnologías y de contribuir en que se pueda cumplir una de las promesas incumplidas del software libre: Permitir que cualquier persona puedan modificar y adaptar el software que usa, para que pueda ajustarlo a sus necesidades particulares y gustos personales. Por ello hacer que el código de programación sea más fácil de escribir y de leer es uno de los propósitos de Meta_Processing y Meta_Javascript.

Jose David Cuartas Correa
Bogotá, Colombia
2020

Agradecimientos:

Al apoyo incondicional recibido por parte de la Fundación Universitaria Los Libertadores quienes han creído en cada proyecto que desarrollamos desde el laboratorio Hitec.

Dedicatoria:

A mi esposa Shahzadi y a mis hijas Helen y Megan, quienes llenan de amor y alegría mi existencia.

CONTENIDO

1. META_JAVASCRIPT PRIMEROS PASOS	11
1.1. ¿Cómo abrir Meta_Javascript?	11
1.2. Ventanas que se abren al iniciar Meta_Prosessing	14
1.3. Elementos básicos de la interface	15
1.4. ¿Cómo seleccionar Idiomas?.....	16
1.5. ¿Cómo ejecutar el código?	16
1.6. ¿Cómo agregar una línea de código?	17
1.7. ¿Cómo eliminar una línea de código?	17
1.8. ¿Cómo agregar instrucciones?	18
1.9. ¿Cuál es la estructura de archivos y carpetas en Meta_Javascript?.....	20
1.10. ¿Cómo abrir la carpeta data del proyecto actual?	22
1.11. ¿Cómo crear un nuevo proyecto?.....	22
1.12. ¿Cómo abrir un proyecto?.....	23
1.13. ¿Cómo guardar el proyecto actual?.....	24
1.14. ¿Cómo exportar el proyecto actual como aplicación?	24
1.15. Ícono de configuración	24
1.16. Funciones: principal, teclado y ratón	26
1.16.1. Principal	26
1.16.2. Ratón.....	27
1.16.3. Teclado	27
2. INSTRUCCIONES BÁSICAS	29
2.1. Documentar el código.....	30
2.2. Coordenadas de pantalla	31
2.3. Instrucciones para gráficos en pantalla	33
2.3.1. fondo	33
2.3.2. línea.....	34
2.3.3. rectángulo	34

2.3.4. elipse	35
2.3.5. triángulo.....	36
2.3.6. tamtexto.....	36
2.3.7. texto	37
2.3.8. imagen.....	38
2.3.9. colorlinea	38
2.3.10. relleno	39
2.3.11. sinrelleno.....	40
2.3.12. sin linea	41
2.4. Instrucciones multimedia	41
2.4.1. tocanota.....	41
2.4.2. sonido	42
3. VARIABLES Y CONDICIONES.....	43
3.1. Variables	43
3.1.1. ¿Cómo se mira el listado de variables?	44
3.1.2. ¿Cómo se crea una variable?	44
3.1.3. ¿Cómo se elimina una variable?.....	45
3.1.4. ¿Cómo se inicializa una variable?	46
3.1.5. ¿Cómo asignarle un nuevo valor a una variable?	47
3.1.6. ¿Cómo se le suma un valor a una variable?	48
3.1.7. ¿Cómo se le resta un valor a una variable?	49
3.1.8. ¿Cómo se le asigna un valor aleatorio a una variable?	50
3.2. Condiciones	51
4. EJEMPLOS DE CÓDIGO CON META_JAVASCRIPT	56
4.1. Dibujos abstractos con líneas	56
4.2. Ejemplo básico con el Teclado	57
4.3. Ejemplo básico con el Ratón	58
4.4. Animación	59
4.5. Piano	60

4.5.1. Piano simple	61
4.5.2. Piano pantalla de colores	61
4.5.3. Piano colores y notas en pantalla	62
4.6. Mini Juego	65
Fuentes de referencia	67

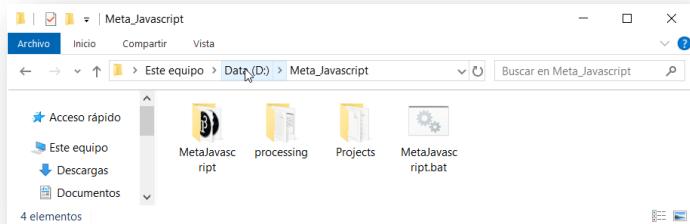
1. META_JAVASCRIPT PRIMEROS PASOS

1.1. ¿Cómo abrir Meta_Javascript?

Meta_Javascript fue desarrollado para que funcionara en los sistemas operativos: Windows, Mac y GNU/Linux. Los pasos para abrir Meta_Javascript varían un poco según el sistema operativo que se use, a continuación se describen los paso para cada sistema:

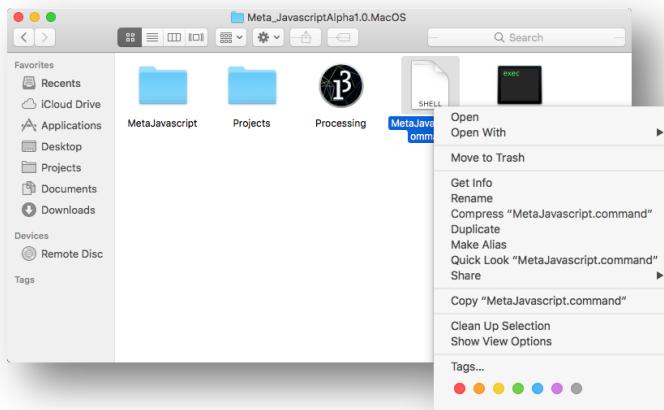
Windows

En Microsoft Windows se debe hacer doble clic en el archivo **MetaJavascript.bat**

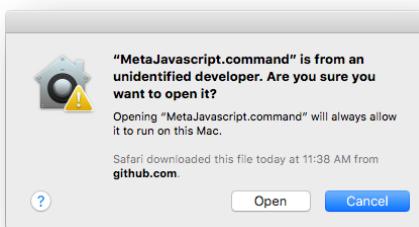


Mac OS

En Mac Os se debe hacer clic con el botón derecho del ratón sobre el archivo **MetaJavascript.command** y seleccionar la opción: **Open o Abrir**

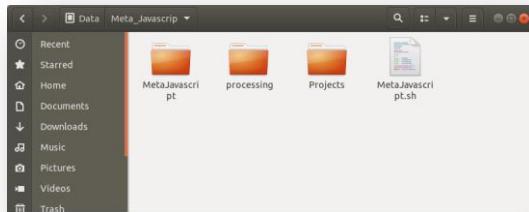


En la ventana que se abre se debe seleccionar la opción: **Open o Abrir**

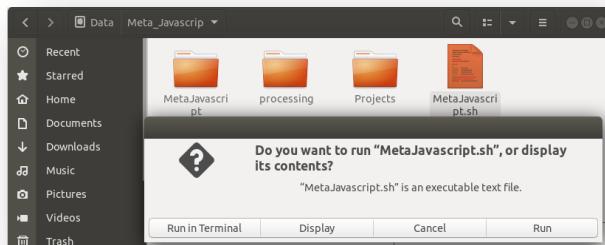


GNU/Linux

En GNU/Linux se debe hacer doble clic en el archivo **MetaJavascript.sh**



En la ventana que se abre se debe seleccionar la opción: **Run o Ejecutar** (también puede usar Ejecutar en Terminal si desea ver la ventana terminal de Meta_Javascript)

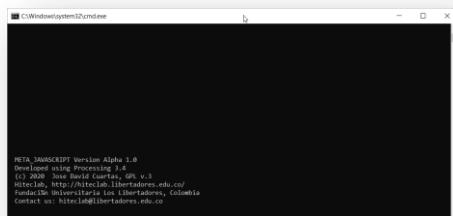


Si el script no abre al hacer doble clic sobre este, se puede ejecutar el siguiente comando en el terminal de Linux, para activar el anterior cuadro de dialogo.

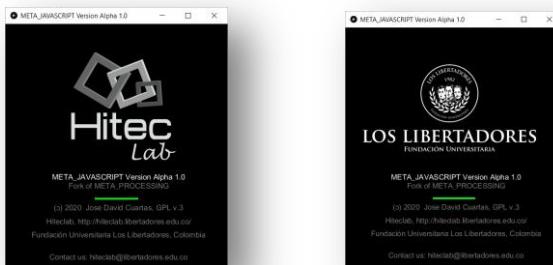
```
gsettings set org.gnome.nautilus.preferences executable-text-activation ask
```

1.2. Ventanas que se abren al iniciar Meta_Prosessing

Una vez se hace ejecuta el archivo Meta_Prosessing sin importar el sistema operativo en el que se esté trabajando primero se abre la ventana terminal donde se pueden ver mensajes que provienen de la ventana principal de Meta_Javascript.

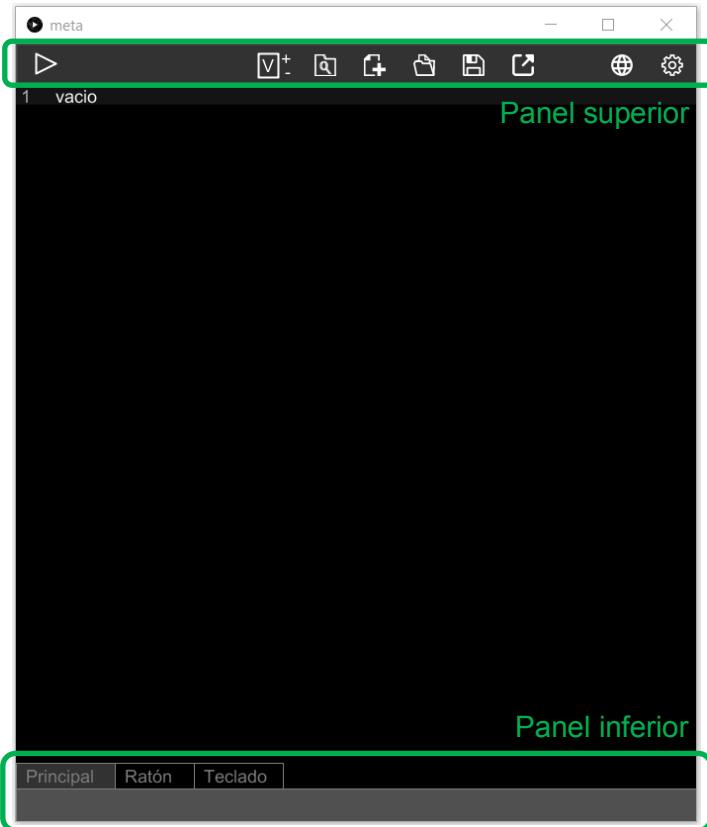


Luego se abre una segunda ventana animada de bienvenida a Meta_Javascript. Al hacer clic sobre esta ventana o al cerrarla se abre la ventana principal de Meta_Javascript.



1.3. Elementos básicos de la interface

En el panel superior están los botones: Ejecutar, Variables, Data, Nuevo, Abrir, Guardar, Exportar, Idiomas y Configuración.



En el panel inferior se encuentran las pestañas: Principal, Ratón y Teclado. Y se encuentra la barra

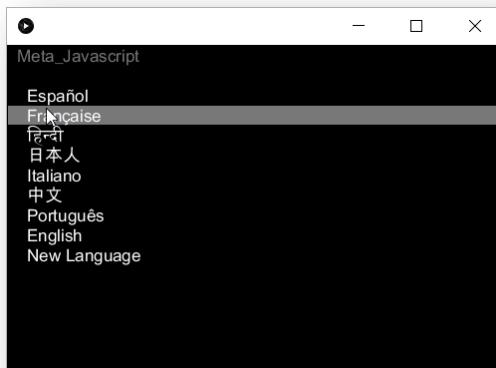
de descripción: en donde se muestra los nombres de los botones y el prototipo de las instrucciones.

1.4. ¿Cómo seleccionar Idiomas?

Para cambiar el idioma de **Meta_Javascript** se debe hacer clic en el ícono **idiomas** en la barra superior.



Luego en la ventana que se abre se debe hacer clic en el idioma deseado.



1.5. ¿Cómo ejecutar el código?

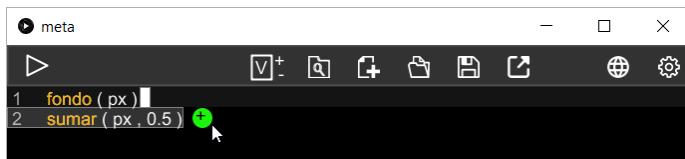
Para ejecutar el código se debe hacer clic en el ícono **ejecutar** en la barra superior.



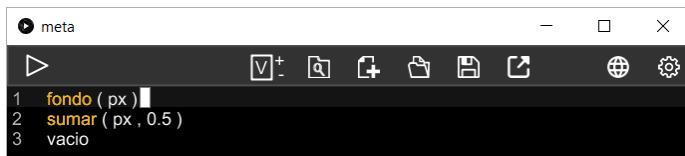
Se espera uno segundos y debe aparecer una nueva ventana en la que se ejecutará el código creado.

1.6. ¿Cómo agregar una línea de código?

Para agregar una línea de código se debe mover el cursor del ratón hasta que aparezca un círculo verde con el carácter más (+) en su interior.

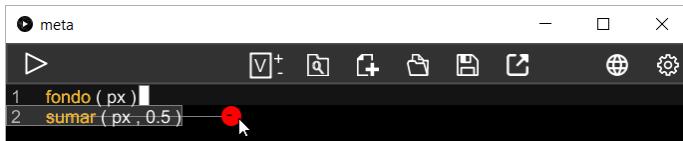


Aparecerá una nueva línea de código una vez se haga clic en el círculo verde.

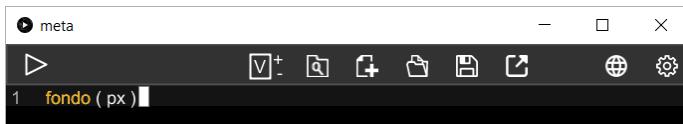


1.7. ¿Cómo eliminar una línea de código?

Para eliminar una línea de código se debe mover el cursor del ratón hasta que aparezca un círculo rojo con el carácter menos (-) en su interior, y se vea una línea gris que tacha toda la instrucción.

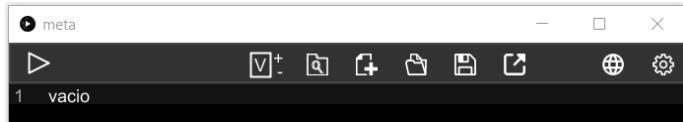


La línea desaparece una vez se hace clic.



1.8. ¿Cómo agregar instrucciones?

Para agregar una instrucción se debe hacer clic en la palabra que dice **vacío**.



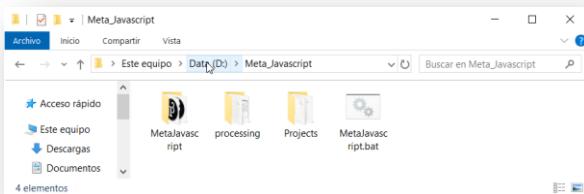
Una vez se hace esto, se abrirá una nueva ventana en donde aparecerán todas las instrucciones disponibles en Meta_Javascript organizadas por categorías así:



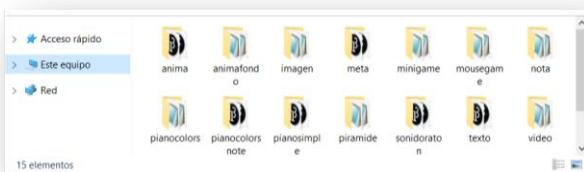
Cuando se ubica el cursor del ratón sobre alguna de estas instrucciones se resalta la instrucción, es este ejemplo se puede ver cómo se resalta la instrucción **triangulo**. Al hacer clic se abrirá una nueva ventana en la que se podrán ingresar las propiedades de cada instrucción. La descripción de cada una de estas instrucciones se hará en el capítulo 2.

1.9. ¿Cuál es la estructura de archivos y carpetas en Meta_Javascript?

Dentro de la carpeta de **Meta_Javascript** se encuentra el archivo que ejecuta **Meta_Javascript** y tres sub-carpetas. La carpeta **MetaJavascript** contiene los archivos que le permiten funcionar. En la carpeta **processing** hay una distribución de este lenguaje que se usa para ejecutar **Meta_Javascript**.

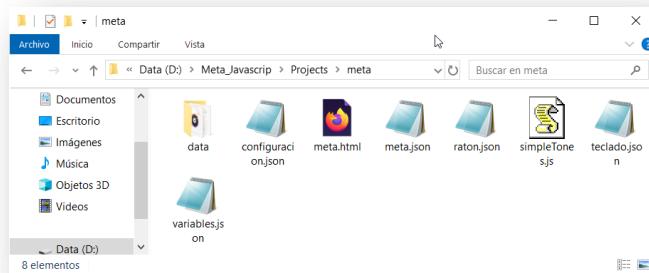


En la carpeta **Projects** contiene las carpetas de cada uno de los proyectos de programas escritos usando el entorno de programación **Meta_Javascript**.



En la carpeta de cada proyecto se encuentran algunos archivos **.json**, un archivo **.js** y un archivo **.html**. Los archivos **.json** contienen las instrucciones en lenguaje **Meta_Javascript**, el archivo **.js** contiene la

librería necesaria para reproducir notas usando el navegador de Internet y el archivo **.html** contienen el código JavaScript del proyecto, y es generado cada vez se hace clic en el ícono **ejecutar**.



Y en la carpeta **data** de cada proyecto se guardan los archivos que se usaran en la ejecución del programa como pueden ser imágenes, sonidos y videos.



1.10. ¿Cómo abrir la carpeta data del proyecto actual?

Para abrir la carpeta data del proyecto actual, se debe hacer clic en el ícono **data** en la barra superior.



Una vez se hace clic se abre en otra ventana la carpeta data del proyecto actual



1.11. ¿Cómo crear un nuevo proyecto?

Para crear un nuevo proyecto se debe hacer clic en el ícono **nuevo** en la barra superior.



En la ventana que se abre se debe escribir el nombre que se le quiere dar al nuevo proyecto y hacer clic en el botón aplicar.

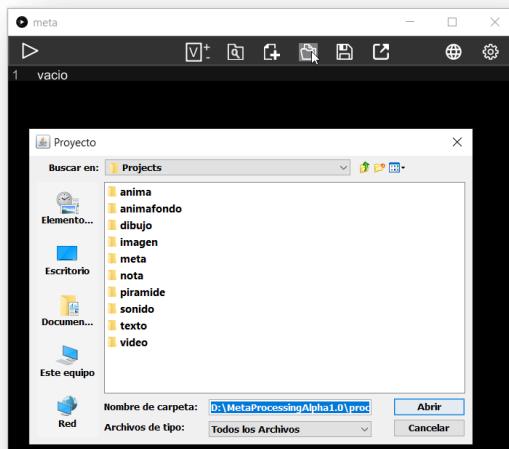


1.12. ¿Cómo abrir un proyecto?

Para abrir un proyecto se debe hacer clic en el ícono **abrir** en la barra superior.



A continuación se abre una nueva ventana en la que se puede seleccionar la carpeta del proyecto que se quiere abrir y se hace clic en botón abrir.



1.13. ¿Cómo guardar el proyecto actual?

Para guardar el proyecto actual se debe hacer clic en el ícono **guardar** en la barra superior.



1.14. ¿Cómo exportar el proyecto actual como aplicación?

Para exportar el proyecto actual como aplicación se debe hacer clic en el ícono **exportar** en la barra superior.



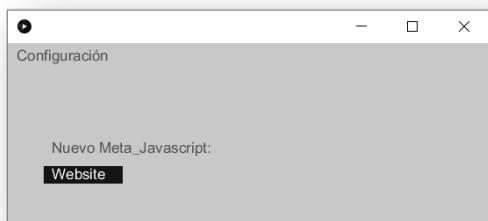
La aplicación se guarda en la subcarpeta llamada **app** dentro de la carpeta del proyecto actual.

1.15. Ícono de configuración

Para cambiar las acceder a la opción de configuración se debe hacer clic en el ícono **configuración** en la barra superior.



A continuación se abre una nueva ventana en la que se ofrece la opción para revisar cuál es última versión de Meta_Javascript.



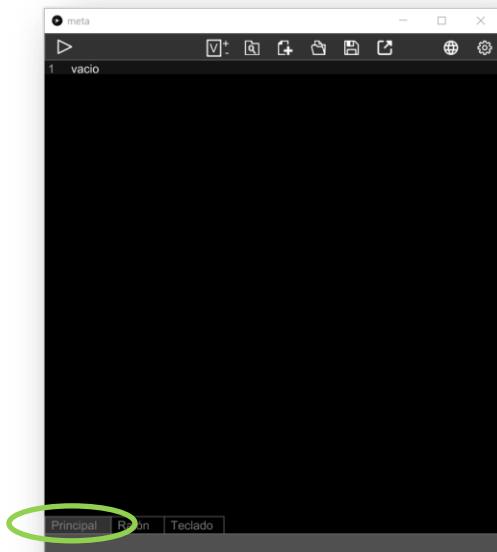
La opción **Nuevo Meta_Javascript** permite revisar última versión de Meta_Javascript publicada en internet. La información que aparece después de los dos puntos es la versión disponible para descargar. Si se desea descargar esa nueva versión se puede hacer clic en el botón **Website** que apunta a al sitio web oficial de descarga de Meta_Javascript.

1.16. Funciones: principal, teclado y ratón

Para escribir el código en Meta_Javascript, se pueden usar tres funciones: Principal, Ratón y Teclado. Cada una de estas funciones se selecciona haciendo clic en su pestaña correspondiente.

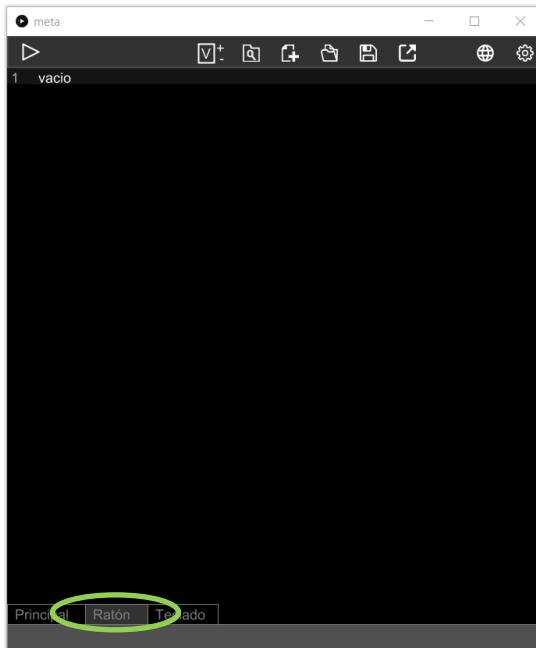
1.16.1. Principal

El código que se escribe en la pestaña **Principal** se ejecuta en un ciclo infinito, hasta que se cierre la ventana de la aplicación.



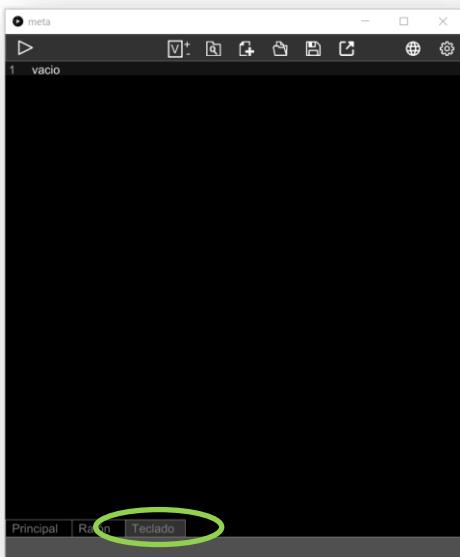
1.16.2. Ratón

El código que se escribe en la pestaña **Ratón** se ejecuta en el momento que se presiona cualquier botón del ratón.



1.16.3. Teclado

El código que se escribe en la pestaña **Teclado** se ejecuta en el momento que se presiona cualquier tecla.

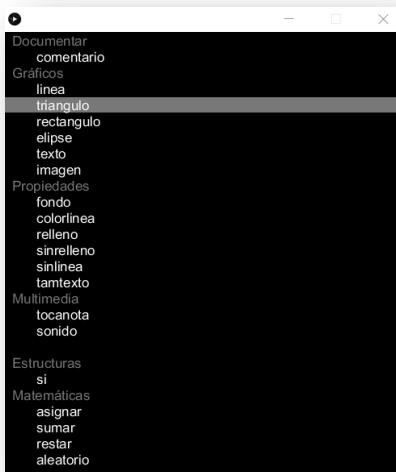


2. INSTRUCCIONES BÁSICAS

En esta sección se abordarán las instrucciones básicas para programar con Meta_Javascript. Como se explicó en el punto 1.8. para agregar una instrucción se debe hacer clic en la palabra que dice **vacío**.



Luego, se abrirá una nueva ventana en donde se muestran todas las instrucciones disponibles en Meta_Javascript organizadas por categorías así:

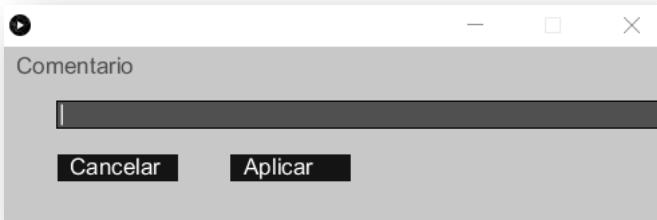


2.1. Documentar el código

Documentar el código de programación es una de las primeras cosas que debe aprender cualquier persona que quiera aprender a programar. Para este propósito todos los lenguajes de programación permiten agregar líneas de **comentario**. La principal característica de esta línea de código es que no se ejecuta, está allí solo para darle información al programador sobre cómo funciona esa parte del código. Para agregar un comentario se debe hacer clic en la opción **comentario** dentro de la categoría de **Documentar**.



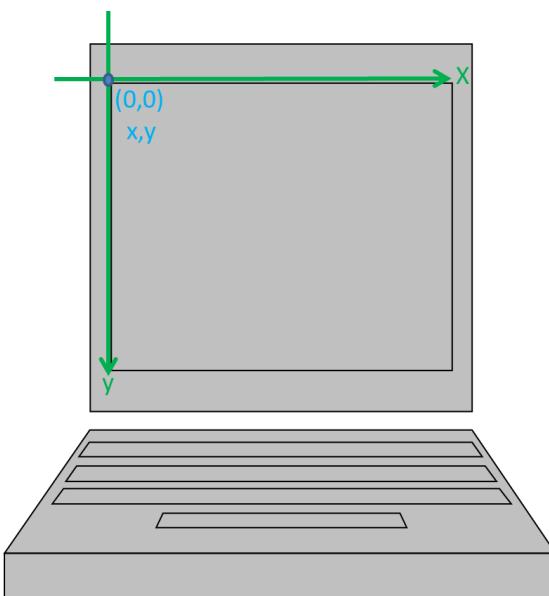
Entonces se abrirá una nueva ventana en la cual se escribe el comentario y se hace clic en **aplicar**.



2.2. Coordenadas de pantalla

Para poder hacer gráficos en pantalla es necesario primero conocer cómo funcionan las coordenadas de pantalla en Meta_Javascript. La posiciones en pantalla se miden en pixeles y cada pantalla tiene un cierto número de pixeles en el eje X y en el eje Y.

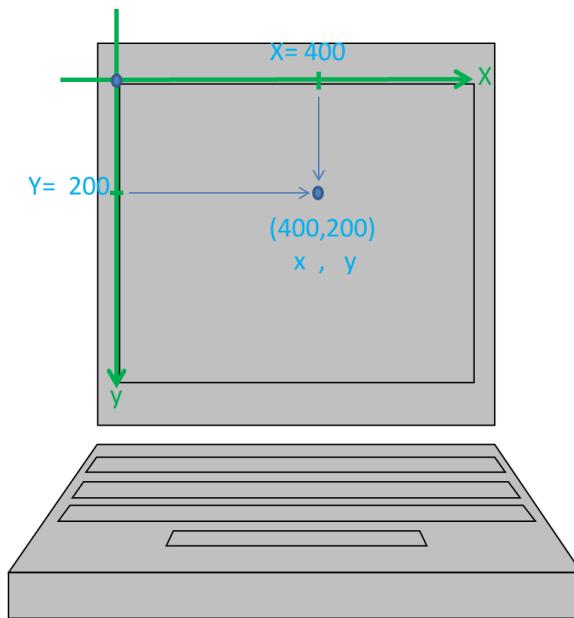
Como se puede observar en la gráfica 1, la esquina superior izquierda de la pantalla es el punto de origen del sistema de coordenadas. Este punto es la posición 0 en el eje X y la posición 0 en el eje Y. Las coordenadas siempre se organizan primero el valor en el eje X, luego una coma y después el valor en el eje Y, así pues este punto es (0,0).



Gráfica 1 Punto de origen en el sistema de coordenadas

Las posiciones en el eje X aumentan de izquierda a derecha y las posiciones en el eje Y aumentan de arriba hacia abajo. En la gráfica 2 se muestra un ejemplo para ilustrar un poco mejor este concepto.

Si se quiera ubicar el punto (400,200) en pantalla, lo que se hace es contar 400 pixels hacia la derecha desde el punto (0,0) de la pantalla y contar 200 pixels de hacia abajo desde el punto (0,0) de la pantalla. Así se ubica el punto (400,200).



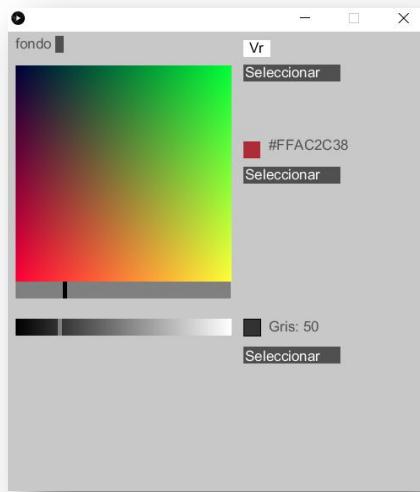
Gráfica 2 Punto en la posición 400 en X y 200 en Y de la pantalla

2.3. Instrucciones para gráficos en pantalla

Entre algunas de las instrucciones para mostrar en pantalla se encuentra: línea, triángulo, rectángulo, elipse, texto, imagen. A continuación se explicará cómo usar cada una de ellas.

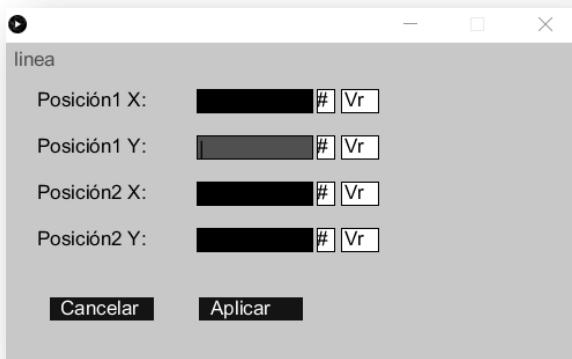
2.3.1. fondo

La instrucción **fondo** sirve para definir el color de fondo de toda la ventana de la aplicación. Esta instrucción borra todo que se esté mostrando en pantalla y deja toda la pantalla del color seleccionado.



2.3.2. Línea

La instrucción **Línea** sirve para dibujar una línea en pantalla. Para trazar una línea se requiere definir la posición x,y del punto donde inicia la línea y la posición x,y del punto donde termina la línea.



2.3.3. rectángulo

La instrucción **rectángulo** sirve para dibujar un rectángulo en pantalla. Para usar esta instrucción se debe definir en las dos primeras casillas la posición x,y de la esquina superior izquierda desde donde se dibujará el cuadrado, y en las dos casillas siguientes definir su ancho y alto.



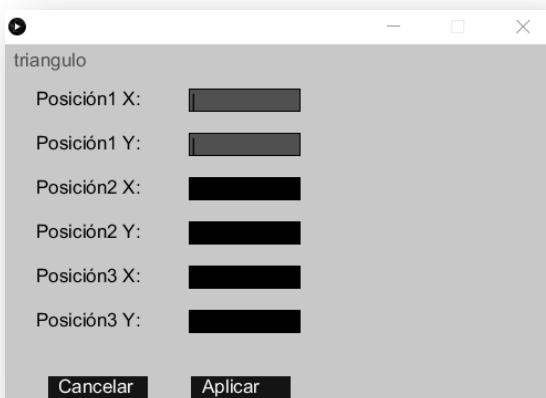
2.3.4. elipse

La instrucción **elipse** sirve para dibujar una elipse en la pantalla. Para usar esta instrucción se debe definir en las dos primeras casillas el punto central x,y desde donde se dibujará la elipse y en las dos casillas siguientes definir su ancho y alto.



2.3.5. triángulo

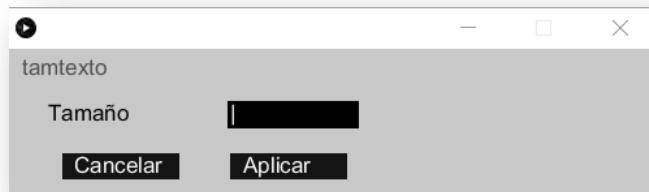
La instrucción **triángulo** sirve para dibujar un triángulo en pantalla. Para dibujar cualquier triángulo se requiere definir los tres puntos correspondientes a cada una de sus esquinas. Para usar esta instrucción se debe definir en las dos primeras casillas la posición del primer punto, en las dos siguientes casillas definir la posición del segundo punto y en las últimas dos casillas de finir la posición del tercer punto.



2.3.6. tamtexto

La instrucción **tamtexto** sirve para definir el tamaño de la fuente a la hora de mostrar un texto en pantalla. Para usar esta instrucción se debe llenar la casilla de tamaño con el número correspondiente al tamaño de fuente que se desea aplicar.

Para que esta instrucción tenga efecto se debe agregar antes de la instrucción **texto**.



2.3.7. **texto**

La instrucción texto sirve para mostrar texto en pantalla. Para usar esta instrucción se debe escribir en la primera casilla el texto que se desea mostrar, y definir en las dos siguientes casillas la posición x,y de la esquina inferior izquierda desde donde se comenzará a mostrar el texto en pantalla.



2.3.8. imagen

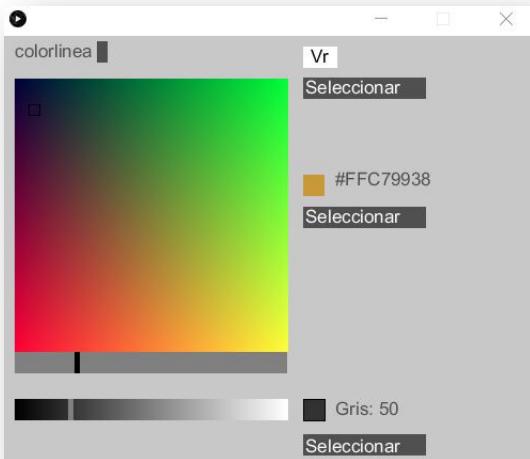
La instrucción **imagen** sirve para mostrar una imagen en pantalla. Para usar esta instrucción primero se debe guardar dentro de la carpeta **data** del proyecto la imagen que se quiere usar, luego se debe seleccionar la imagen que fue guardada en la carpeta data y por último definir en las dos siguientes casillas la posición x,y de la esquina superior izquierda desde donde se comenzará a mostrar la imagen en pantalla.



2.3.9. colorlinea

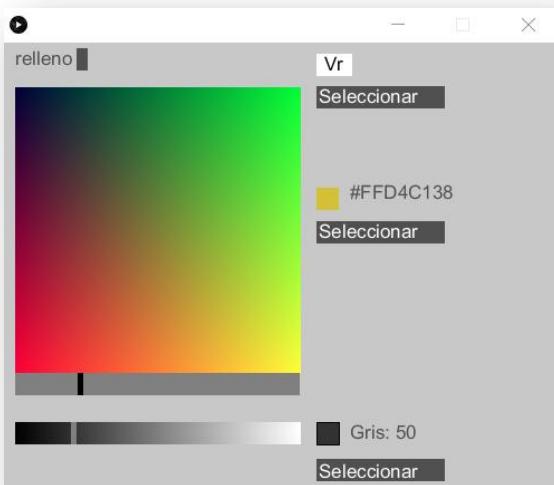
La instrucción **colorlinea** sirve para definir el color de las líneas y color del borde de las figuraras rectángulo, elipse y triángulo. Para asignar el color de línea usando esta instrucción se puede, utilizar el selector de color, o el selector de escala de grises y hacer clic en el botón **Aplicar**. También se puede

usar una variable para cambiar de forma dinámica el color. Para que esta instrucción tenga efecto se debe agregar antes de las instrucciones con las que se dibuja líneas, o figuras en pantalla.



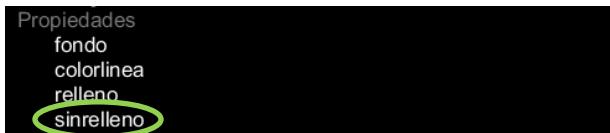
2.3.10. relleno

La instrucción **relleno** sirve para definir el color de relleno de las figuraras rectángulo, elipse y triángulo. Para asignar el color de relleno usando esta instrucción se puede, utilizar el selector de color, o el selector de escala de grises y hacer clic en el botón **Aplicar**. También se puede usar una variable para cambiar de forma dinámica el color. Para que esta instrucción tenga efecto se debe agregar antes de las instrucciones con las que se dibuja figuras en pantalla.



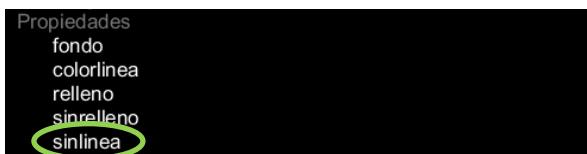
2.3.11. sinrelleno

La instrucción **sinrelleno** sirve para quitar el relleno de las figuras rectángulo, elipse y triángulo. Esta se verán transparentes y solo se verá su borde. Para usar esta instrucción solo se necesita que en la ventana de agregar instrucción, se haga clic en sobre la instrucción **sinrelleno** dentro de la categoría **Propiedades** y automáticamente se agregará esta línea al código del proyecto.



2.3.12. sinlinea

La instrucción **sinlinea** sirve para quitar el borde de las figuras rectángulo, elipse y triángulo. Si se agrega esta instrucción antes de la instrucción **línea** entonces no se mostrará en pantalla la línea. Para usar esta instrucción solo se necesita que en la ventana de agregar instrucción, se haga clic en sobre la instrucción **sinlinea** dentro de la categoría Propiedades y automáticamente se agregará esta línea al código del proyecto.

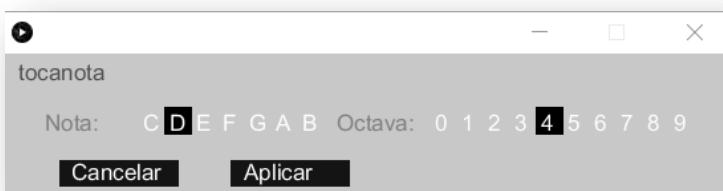


2.4. Instrucciones multimedia

Entre algunas de las instrucciones multimedia que se puede usar con Meta_Javascript se encuentra: **tocanota**, **sonido** y **video**. A continuación se explicará cómo usar cada una de ellas.

2.4.1. tocanota

La instrucción **tocanota** sirve para reproducir el sonido de una nota de la escala musical. Para usar esta instrucción se debe seleccionar la nota que se quiere tocar y luego seleccionar la octava que se quiere que suene la nota. Por último se debe hacer clic en el botón **Aplicar**.



2.4.2. sonido

La instrucción **sonido** sirve para reproducir un archivo de sonido en formato wav o mp3. Para usar esta instrucción primero se debe guardar dentro de la carpeta **data** del proyecto, el archivo de sonido que se quiere usar, luego se debe hacer clic en el botón **Seleccionar** para elegir el archivo que fue guardado previamente en la carpeta data y por último se debe hacer clic en el botón **Aplicar**.



3. VARIABLES Y CONDICIONES

En esta sección se abordará los conceptos de variables y condiciones, los cuales son fundamentales a la hora de aprender a programar.

3.1. Variables

Una variable es un espacio de memoria reservado para almacenar un valor que va cambiando durante el trascurso de la ejecución del programa. En Meta_Javascript se hay dos tipos de variables: Las variables del sistema y las variables creadas por el usuario.

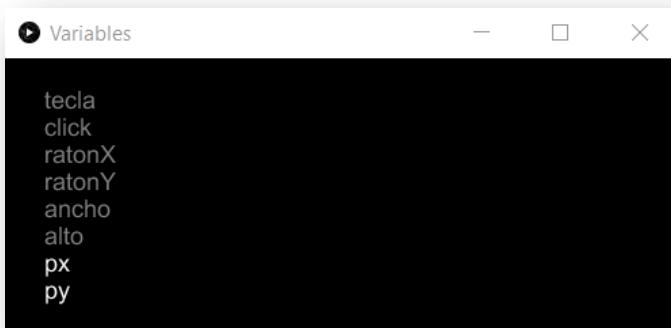
Las variables del sistema son: **tecla**, **click**, **ratonX**, **ratonY**, **ancho** y **alto**. La variable **tecla** almacena el valor de la última tecla presionada en el teclado. La variable **click** amacena el valor del último botón presionado en el ratón. La variable **ratonX** almacena la posición actual del ratón en el eje X. La variable **ratonY** almacena la posición actual del ratón en el eje Y. La variable **ancho** almacena el valor del ancho de la pantalla en la que ese está ejecutando el código. Y la variable **alto** almacena el valor del alto de la pantalla en la que ese está ejecutando el código.

3.1.1. ¿Cómo se mira el listado de variables?

Para mirar el listado de variables del proyecto se debe hacer clic en el ícono **variable**.



En la ventana que se abre se verán las variables que se están usando. Las que se muestran en color gris son las variables del sistema y las que se muestran de color blanco son las variables creadas por el usuario.



3.1.2. ¿Cómo se crea una variable?

Para crear una variable se debe hacer clic en el ícono más (+) que está a un lado del ícono **variable**.



En la ventana que se abre se debe escribir el nombre que se le quiere dar de la variable que se va a crear, en este ejemplo se le da el nombre **px**.

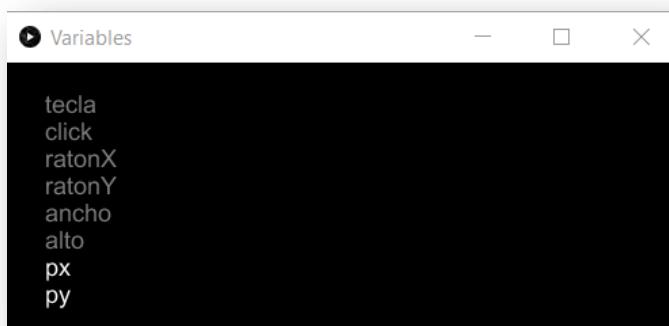


3.1.3. ¿Cómo se elimina una variable?

Para eliminar una variable se debe hacer clic en el ícono menos (-) que está a un lado del ícono **variable**.



En la ventana que se abre se debe hacer clic sobre el nombre de la variable que se quiere eliminar. Solo se pueden eliminar las variables que han sido creadas por el usurario, es decir, solo le pueden eliminar las variables que su texto es color blanco. Las variables en color gris son las variables del sistema y no se pueden eliminar.

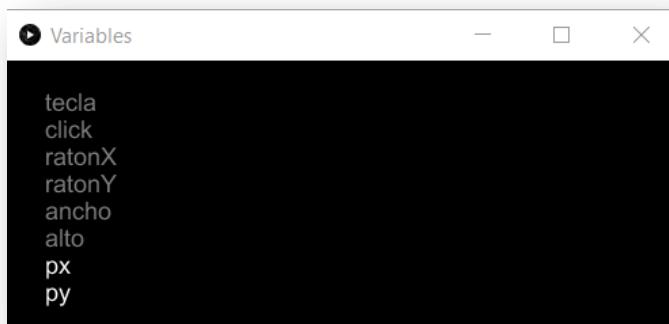


3.1.4. ¿Cómo se inicializa una variable?

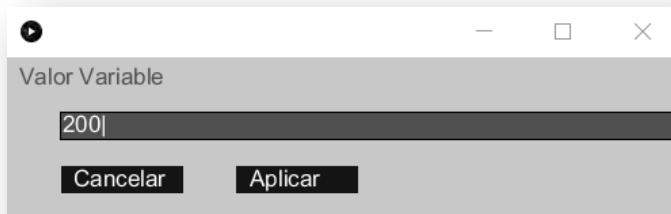
Para inicializar una variable creada por el usuario se debe hacer clic en el ícono **variable**.



En la ventana que se abre se hace clic sobre la variable que se quiere inicializar.



Hecho esto aparecerá otra ventana en la que se debe escribir el valor con el que se quiere inicializar la variable. En este ejemplo se inicializa la variable con el valor 200.

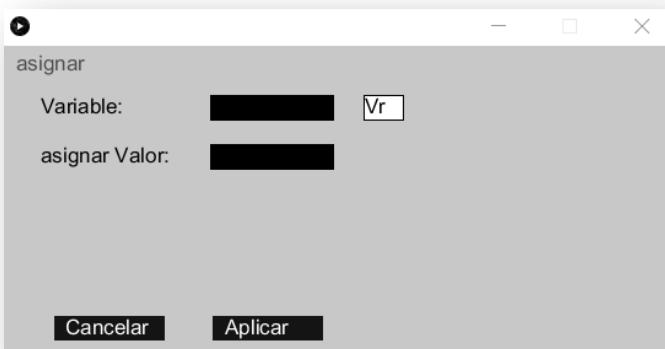


3.1.5. ¿Cómo asignarle un nuevo valor a una variable?

Dentro del código se le puede asignar un nuevo valor a una variable, para ello se debe agregar la instrucción **asignar** que se encuentra dentro de la categoría **Matemáticas** en la ventana de agregar instrucción.



Luego en la ventana que se abre, en la primera casilla se selecciona la variable a la que se le quiere asignar un nuevo valor y en la segunda casilla se escribe el valor que se le va a asignar.

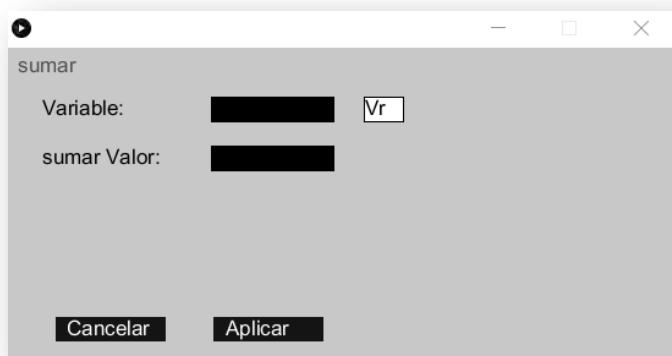


3.1.6. ¿Cómo se le suma un valor a una variable?

Dentro del código se le puede sumar un valor a una variable, para ello se debe agregar la instrucción **sumar** que se encuentra dentro de la categoría **Matemáticas** en la ventana de agregar instrucción.



Luego en la ventana que se abre, en la primera casilla se selecciona la variable a la que se le quiere sumar el valor y en la segunda casilla se escribe el valor que se le va a sumar.

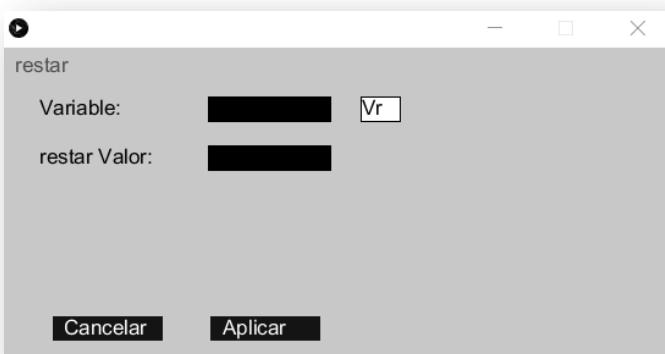


3.1.7. ¿Cómo se le resta un valor a una variable?

Dentro del código se le puede restar un valor a una variable, para ello se debe agregar la instrucción **restar** que se encuentra dentro de la categoría **Matemáticas** en la ventana de agregar instrucción.

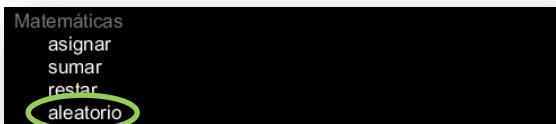


Luego en la ventana que se abre, en la primera casilla se selecciona la variable a la que se le quiere restar el valor y en la segunda casilla se escribe el valor que se le va a restar.



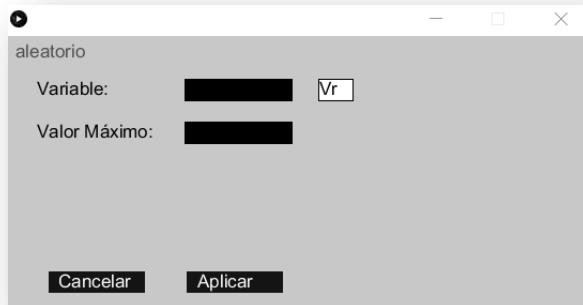
3.1.8. ¿Cómo se le asigna un valor aleatorio a una variable?

Dentro del código se le puede asignar un valor aleatorio a una variable, para ello se debe agregar la instrucción **aleatorio** que se encuentra dentro de la categoría **Matemáticas** en la ventana de agregar instrucción.



Luego en la ventana que se abre, en la primera casilla se selecciona la variable a la que se le quiere asignar el valor aleatorio y en la segunda casilla se escribe el valor máximo que se podría generar

aleatoriamente. De manera que se generaría un valor aleatorio entre 0 y el valor máximo que se le asigne a la instrucción.



3.2. Condiciones

Las condiciones son un tipo de estructuras algorítmicas que le permiten al programa tomar decisiones según se cumpla una condición.

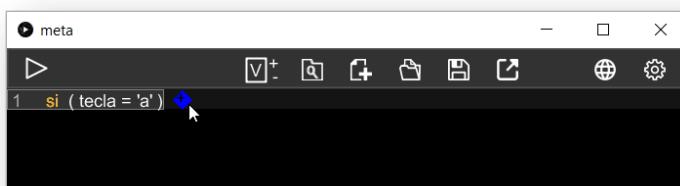
Para agregar una condición se debe agregar la instrucción **si** que se encuentra dentro de la categoría Estructuras en la ventana de agregar instrucción.



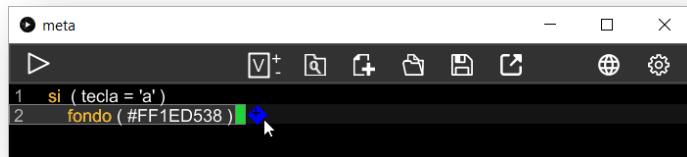
Luego en la ventana que se abre en la primera casilla se puede escribir un valor o seleccionar una variable. Luego se debe seleccionar un operador, el cual puede ser igual (=), menor que (<), mayor que (>) o diferente (!=). Y en la segunda casilla se puede escribir un valor, seleccionar una variable, elegir una tecla o elegir uno de los botones del ratón.



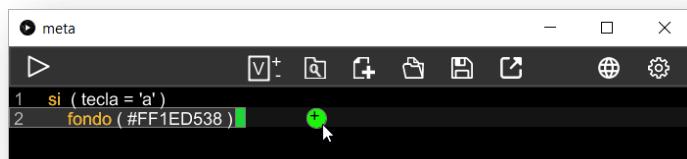
Una vez se hace clic en el botón **Aplicar** se podrá ver la condición en la ventana de código de Meta_Javascript. Para agregar una instrucción dentro de la condición, se debe mover el cursor del ratón hasta que aparezca un rombo azul con el carácter más (+) en su interior.



Una vez se hace clic aparecerá la nueva línea de código vacía y se hace clic para asignarle la instrucción deseada. Si se quiere crea una instrucción más dentro de la condición entonces, se debe mover el cursor del ratón hasta que aparezca un rombo azul con el carácter más (+) en su interior.

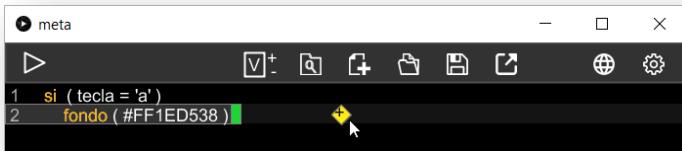


Si una vez se termina de agregar las instrucciones dentro de la condición, lo que se quiere es agregar otra línea de código pero por fuera de la condición, entonces se debe mover el cursor del ratón hasta que aparezca un circulo verde con el carácter más (+) en su interior.

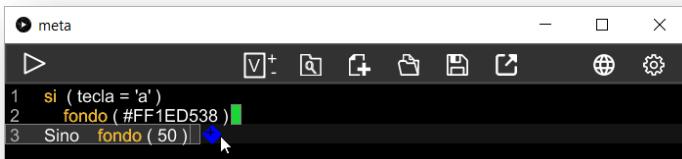


Si por el contrario lo que se quiere es agregar una línea para el caso cuando no se cumpla dicha condición entonces se podrán agregar líneas dentro del sino. Para esto se debe mover el cursor del ratón

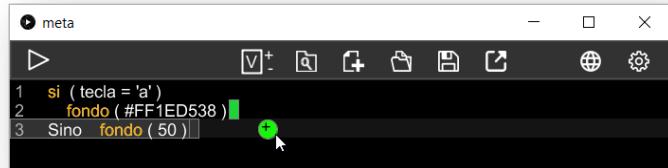
hasta que aparezca un rombo amarillo con el carácter más (+) en su interior.



Una vez se seleccione la instrucción que se quiere usar se vería que adelante de la instrucción aparece la palabra **Sino**. Para cuando se quiera agregar más instrucciones dentro del **Sino** se debe mover el cursor del ratón hasta que aparezca un rombo azul con el carácter más (+) en su interior.



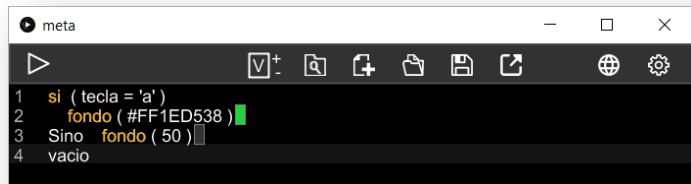
Si por el contrario se quiere es agregar otra línea de código pero por fuera del **Sino**, entonces se debe mover el cursor del ratón hasta que aparezca un circulo verde con el carácter más (+) en su interior.



A screenshot of the Meta_Javascript IDE interface. The title bar says "meta". The toolbar includes icons for play, search, file operations, and settings. The code editor shows the following script:

```
1 si ( tecla = 'a' )
2 fondo ( #FF1ED538 )
3 Sino fondo ( 50 )
```

Al hace esto entonces aparecerá una nueva línea vacía por fuera de las instrucciones de la condición.



A screenshot of the Meta_Javascript IDE interface. The title bar says "meta". The toolbar includes icons for play, search, file operations, and settings. The code editor shows the following script, with the third line now being empty:

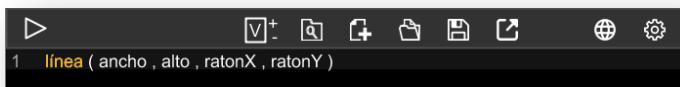
```
1 si ( tecla = 'a' )
2 fondo ( #FF1ED538 )
3 Sino fondo ( 50 )
4 vacio
```

4. EJEMPLOS DE CÓDIGO CON META_JAVASCRIPT

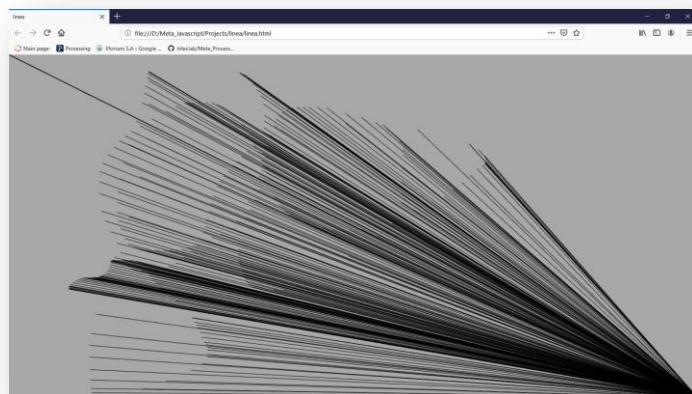
A continuación se presentan una serie de ejemplos de cómo usar Meta_Javascript. El primero de ellos nos muestra cómo hacer dibujos abstractos con líneas. El segundo no permite experimentar con la función teclado. El tercero nos permite dibujar círculos en pantalla al presionar el ratón. El cuarto es un ejemplo de cómo se puede crear una animación. El quinto nos muestra tres formas de programar un piano. Y el último nos muestra cómo crear un sencillo mini juego.

4.1. Dibujos abstractos con líneas

La siguiente línea de código se debe escribir en la pestaña **Principal** y al ejecutar el código se debe mover el cursor del ratón para generar los dibujos abstractos con líneas.



Al ejecutarse se verá en el navegador algo así:



4.2. Ejemplo básico con el Teclado

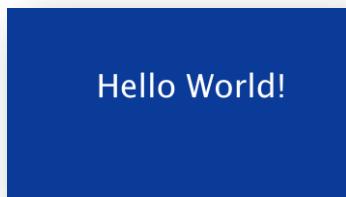
El siguiente código se debe escribir en la pestaña **Teclado** de manera que se ejecute en el momento que se presione cualquier tecla.

A screenshot of the Processing IDE showing the keyboard tab. The code is as follows:

```
> [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
1 fondo (#FF0B3B99)  
2 tamtexto (180)  
3 texto (Hello World!, 500, 500)
```

The code defines a background color, sets the text size to 180 pixels, and prints the text "Hello World!" at coordinates (500, 500).

La idea es que al presionar cualquier tecla, se pone la pantalla azul y aparece en la pantalla un texto en color blanco.



4.3. Ejemplo básico con el Ratón

El siguiente código se debe escribir en la pestaña **Ratón** de manera que se ejecute en el momento que se presione cualquiera de los botones del ratón.

A screenshot of the Meta_Javascript IDE interface. At the top, there's a toolbar with various icons. Below it is a code editor window containing the following three lines of code:

```
1 sinlínea
2 relleno ( #FFFDD738 )
3 elipse ( ratonX , ratonY , 80 , 80 )
```

La idea es que al presionar cualquiera de los botones del ratón se dibujen círculos amarillos en pantalla.



4.4. Animación

A continuación se muestra un ejemplo de cómo se puede hacer una animación en Meta_Javascript. Primero se debe crear dos variables una llamada **px** y otra llamada **py**. Se debe inicializar la variable **px** con el valor 10.



Y variable **py** se debe inicializar con el valor 950.



Luego se debe agregar el siguiente código en la pestaña **Principal**:

A screenshot of the Meta_Javascript code editor. The title bar says "anima". The main workspace shows the following code:

```
1 fondo ( px )
2 sumar ( px , 0.5 )
3 restar ( py , 0.5 )
4 relleno ( #FFF2E138 )
5 ellipse ( 200 , py , 200 , 200 )
6 relleno ( #FF0B259A )
7 rectangulo ( 0 , 800 , ancho , 400 )
```

The code uses color-coded syntax highlighting where numbers are blue and strings are yellow.

Al hacer clic en el ícono **ejecutar** se vería que la animación empieza recreando una escena nocturna en el mar, y lentamente irá amaneciendo hasta tener el firmamento completamente iluminado.



4.5. Piano

A continuación se muestra tres ejemplos de cómo hacer un piano en Meta_Javascript. El primero es un piano simple, el segundo es un piano que además cambia el color de la pantalla y el tercero es un piano que cambia el color de la pantalla y muestra un texto con la nota que suena.

4.5.1. Piano simple

Para programar este piano se debe agregar en la pestaña **Teclado** el siguiente código:

```
1 si ( tecla = 'a' )
2 tocanota ( C4 )
3 si ( tecla = 's' )
4 tocanota ( D4 )
5 si ( tecla = 'd' )
6 tocanota ( E4 )
7 si ( tecla = 'f' )
8 tocanota ( F4 )
9 si ( tecla = 'g' )
10 tocanota ( G4 )
11 si ( tecla = 'h' )
12 tocanota ( A4 )
13 si ( tecla = 'i' )
14 tocanota ( B4 )
```

Con este código se puede tocar la 7 notas musicales usando las teclas a, s, d, f, g, h, j. Para que funcione bien el teclado no puede estar en modo mayúsculas.

4.5.2. Piano pantalla de colores

Para programar este piano se debe agregar en la pestaña **Teclado** el siguiente código:

The screenshot shows a Scratch script editor window. At the top, there's a toolbar with icons for play, stop, clear stage, and other controls. Below the toolbar is a script area containing the following code:

```
1 si ( tecla = 'a' )
2   tocanota ( C4 )
3   fondo ( #FF2AD538 ) [green]
4 si ( tecla = 's' )
5   tocanota ( D4 )
6   fondo ( #FFC72538 ) [red]
7 si ( tecla = 'd' )
8   tocanota ( E4 )
9   fondo ( #FF3348AE ) [blue]
10 si ( tecla = 'f' )
11   tocanota ( F4 )
12   fondo ( #FFFCDD138 ) [yellow]
13 si ( tecla = 'g' )
14   tocanota ( G4 )
15   fondo ( #FF48C0E ) [orange]
16 si ( tecla = 'h' )
17   tocanota ( A4 )
18   fondo ( #FFEF1EC8 ) [purple]
19 si ( tecla = 'j' )
20   tocanota ( B4 )
21   fondo ( #FF57938D ) [teal]
```

At the bottom of the editor, there are three tabs: "Principal", "Ratón", and "Teclado".

Con este código se puede tocar la 7 notas musicales usando las teclas a, s, d, f, g, h, j. Para que funcione bien el teclado no puede estar en modo mayúsculas.

4.5.3. Piano colores y notas en pantalla

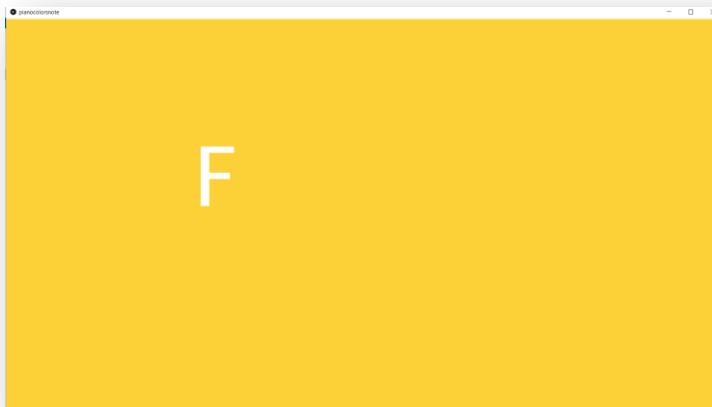
Para programar este piano se debe agregar en la pestaña **Teclado** el siguiente código:

The screenshot shows a code editor window with a dark theme. The menu bar at the top includes icons for file operations like New, Open, Save, and Print, along with a gear icon for settings. Below the menu is a toolbar with various icons. The main area contains a script with numbered lines from 1 to 29. The script uses pseudocode-like commands such as 'tamtexto', 'si', 'tocanota', 'fondo', and 'texto'. The colors of the text blocks correspond to the piano keys: green for C, red for D, blue for E, orange for F, purple for G, pink for A, and teal for B. At the bottom of the editor, there is a navigation bar with tabs labeled 'Principal', 'Ratón', and 'Teclado'.

```
1 tamtexto ( 220 )
2 si ( tecla = 'a' )
3 tocanota ( C4 )
4 fondo ( #FF2AD538 ) █
5 texto ( C , 500 , 500 )
6 si ( tecla = 's' )
7 tocanota ( D4 ) █
8 fondo ( #FFC72538 ) █
9 texto ( D , 500 , 500 )
10 si ( tecla = 'd' )
11 tocanota ( E4 ) █
12 fondo ( #FF3348AE ) █
13 texto ( E , 500 , 500 )
14 si ( tecla = 'f' )
15 tocanota ( F4 ) █
16 fondo ( #FFFCB138 ) █
17 texto ( F , 500 , 500 )
18 si ( tecla = 'g' )
19 tocanota ( G4 ) █
20 fondo ( #FFF48C0E ) █
21 texto ( G , 500 , 500 )
22 si ( tecla = 'h' )
23 tocanota ( A4 ) █
24 fondo ( #FFEF1EC8 ) █
25 texto ( A , 500 , 500 )
26 si ( tecla = 'j' )
27 tocanota ( B4 ) █
28 fondo ( #FF68B0A0 ) █
29 texto ( B , 500 , 500 )
```

Con este código se puede tocar la 7 notas musicales usando las teclas a, s, d, f, g, h, j. Para que funcione bien el teclado no puede estar en modo mayúsculas.

El piano al ejecutarse se vería así:



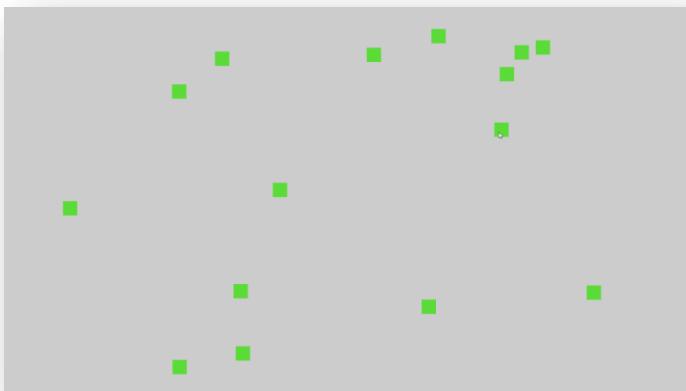
4.6. Mini Juego

A continuación se muestra un ejemplo de cómo se puede hacer un mini juego en Meta_Javascript. En la pestaña **Principal** se debe agregar el siguiente código:

Y en la pestaña **Ratón** se debe agregar el siguiente código:

El juego consiste en una serie de cuadrados verdes que van apareciendo aleatoriamente en la pantalla. El propósito del juego es tratar hacer desaparecer todos los cuadros verdes haciendo clic sobre ellos.

El juego al ejecutarse se vería algo así:



Fuentes de referencia

Cuartas, J.D. (2014). Digitópolis I: Diseño de Aplicaciones Interactivas para Creativos y Comunicadores. Bogotá: Fundación Universitaria Los Libertadores.

Cuartas, J.D. (2017). Programar el mundo en el contexto de las tecnologías libres y las culturas Hacker-Maker. Caso de estudio: Hitec Lab. (Tesis doctoral). Doctorado en Diseño y Creación. Universidad de Caldas, Manizales.

Hitec Lab (2020). Hitec Lab Homepage. Recuperado de <http://hiteclab.libertadores.edu.co/>

Hitec Lab (2020). Meta_Javascript. Recuperado de https://github.com/hiteclab/Meta_Javascript

Hitec Lab (2020). Meta_Javascript. Recuperado de https://github.com/hiteclab/Meta_Javascript

Libertadores, L. (2019). Institución Universitaria Los
Libertadores. Recuperado de
<http://www.ulibertadores.edu.co/>

Processing. (2019). Processing. Recuperado de
<http://www.processing.org/>

Processing. Fue creado por el autor de este libro con el propósito de brindarle al principiante en la programación de aplicaciones web una herramienta para crear código usando su idioma nativo. Está basado en Meta_Processing y fue diseñado para escribir y leer el mismo código en diferentes idiomas, como Español, Francés, Hindi, Japonés, Italiano, Chino, Portugués e Inglés. Es un entorno de programación diseñado para ofrecer una experiencia amigable, que le evite al usuario cometer errores comunes de sintaxis. Meta_Javascript es un software libre publicado bajo licencia GPL v3.

Lo que se busca con esta derivación de Meta_Processing es ir creando un

plataformas haciendo uso de un mismo lenguaje de programación. En la actualidad es casi indispensable aprender un lenguaje diferente para programar cada plataforma.

El profesor Jose David Cuartas es el director y fundador del laboratorio Hipermédia

doctor en Diseño y Creación de la Universidad de Caldas. Es promotor del uso desarrollo de software libre para los entornos del arte, el diseño y el entretenimiento. Su trabajo investigativo se enfoca en explorar estrategias para promover las culturas Hacker y Maker en el contexto académico universitario, buscando que las personas inexpertas descubran las oportunidades creativas que existen al aprender a programar. Desde el laboratorio Hipermédia es uno de los mentores de la iniciativa llamada: "Mujeres en tecnología", con la que se busca acercar al público femenino a las culturas Hacker y Maker y contribuir en desmitificar las tecnologías. El profesor Cuartas es el autor de la serie de tres libros llamada Digitópolis, la cual está disponible para descarga gratuita en google books. El primer libro de la serie (Digitópolis I), le permite al lector aprender a programar usando el lenguaje Processing. El segundo libro (Digitópolis II), es una guía para aprender a desarrollar videojuegos 2D con Gdevelop. Y el tercer libro (Digitópolis III), es para aquellos interesados en aprender a crear recorridos virtuales en Blender 3D.



10 of 10

