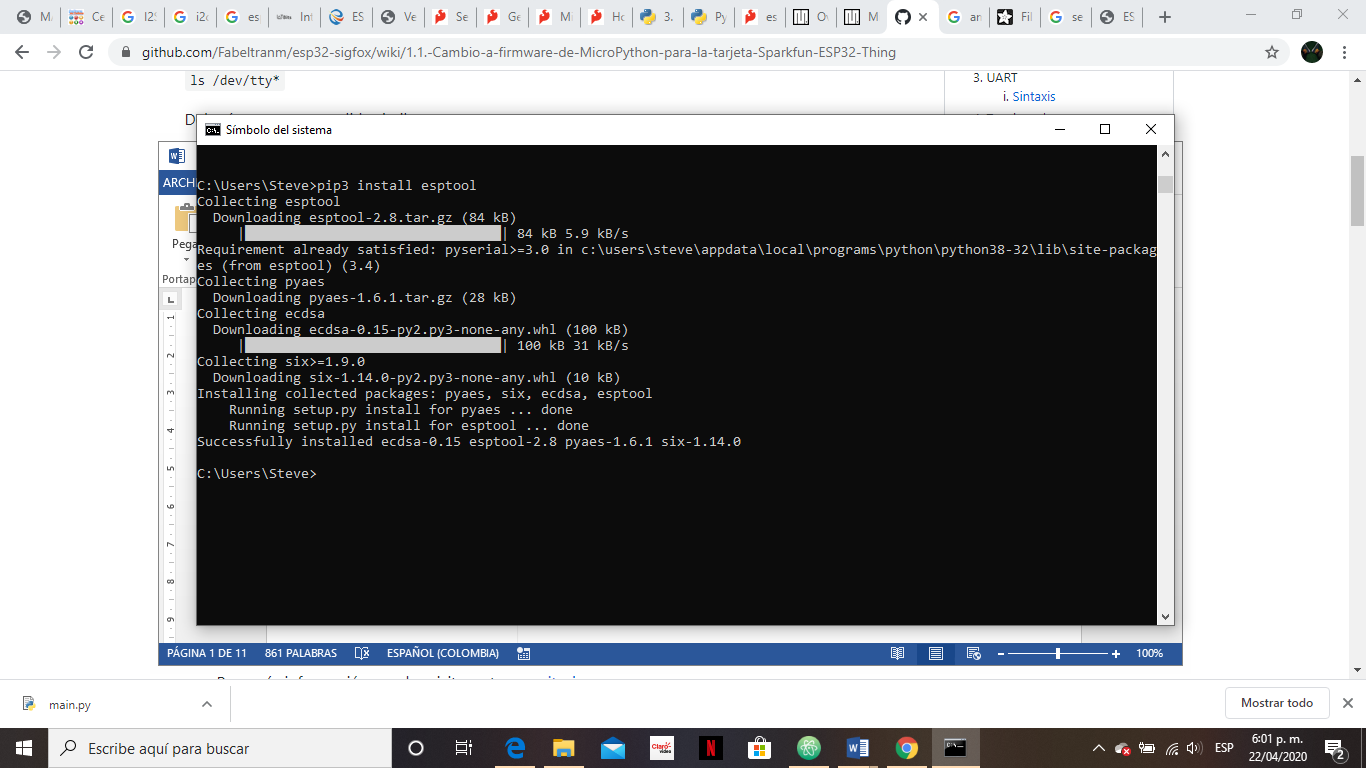
# Resultados

Software

* Instalación de MicroPython en el procesador ESP32

Antes de instalar el firmware de MicroPython, se debe eliminar el firmware que trae por defecto la tarjeta ESP32, para ello es necesario tener instalado Python v3.x.x, esptool y descargar el firmware de MicroPython, Ver Figura (tal)

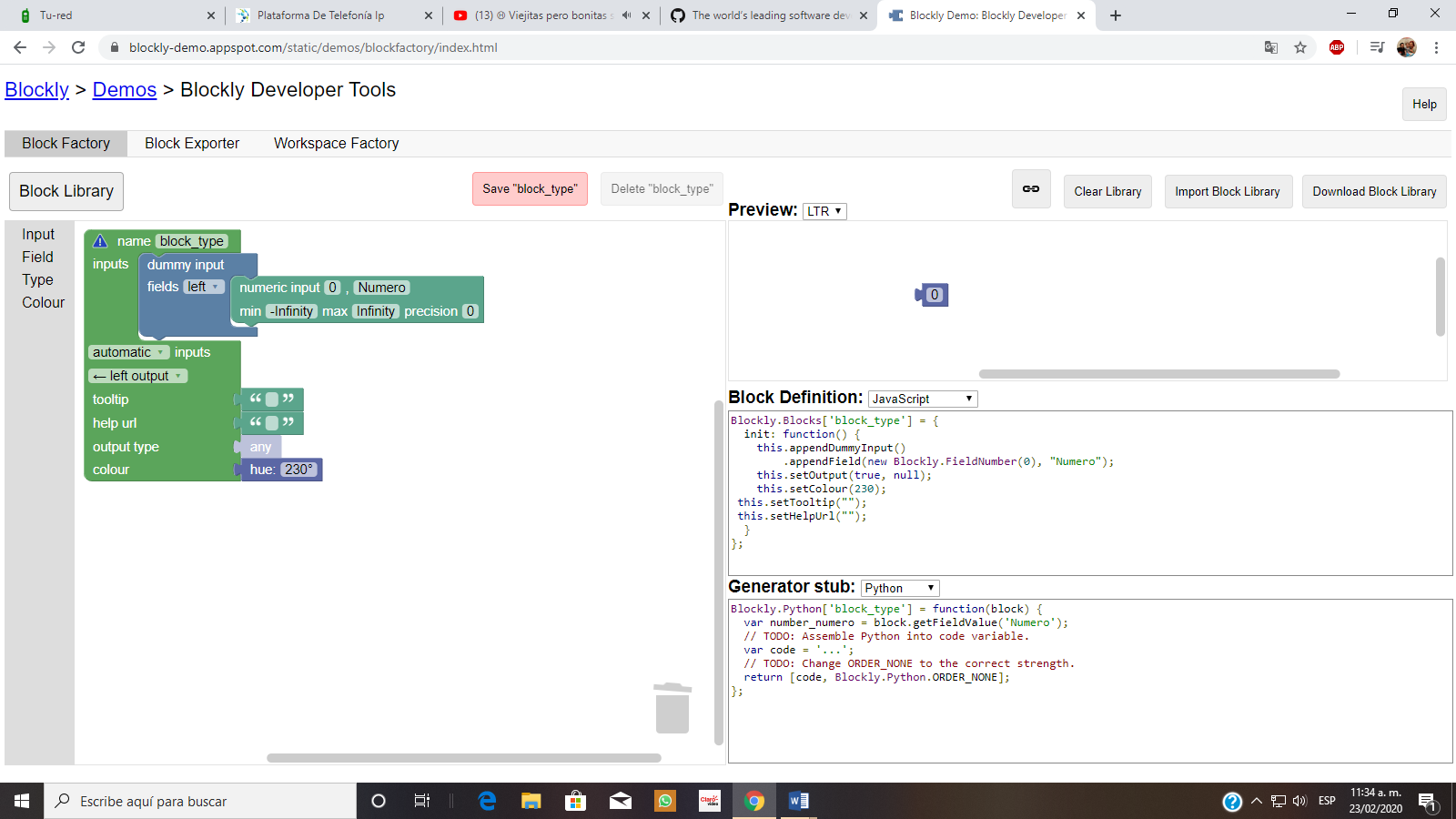
Y siguiendo las instrucciones suministradas por el fabricante en la página <http://docs.micropython.org/en/latest/esp32/tutorial/intro.html#esp32-intro>

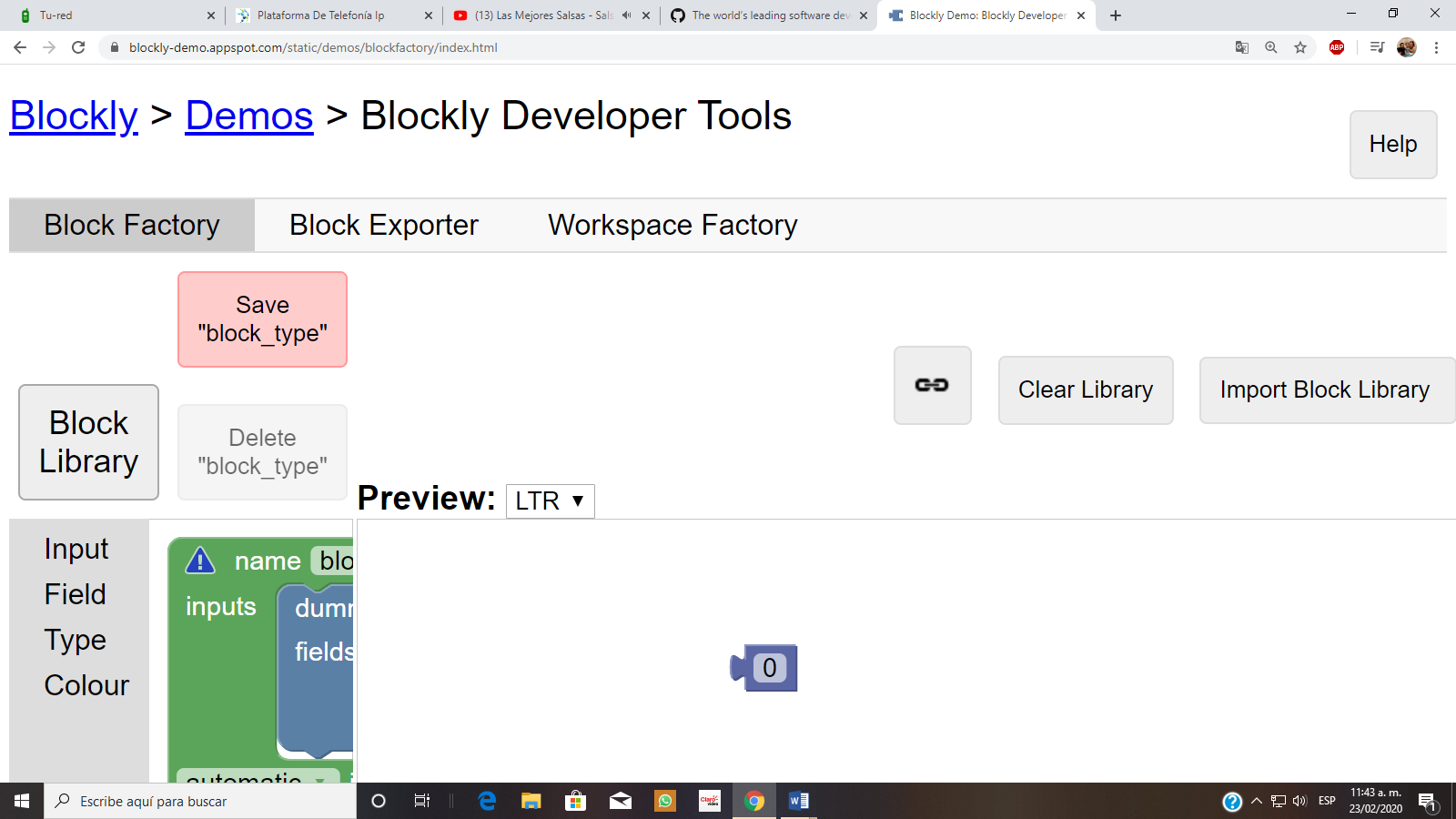
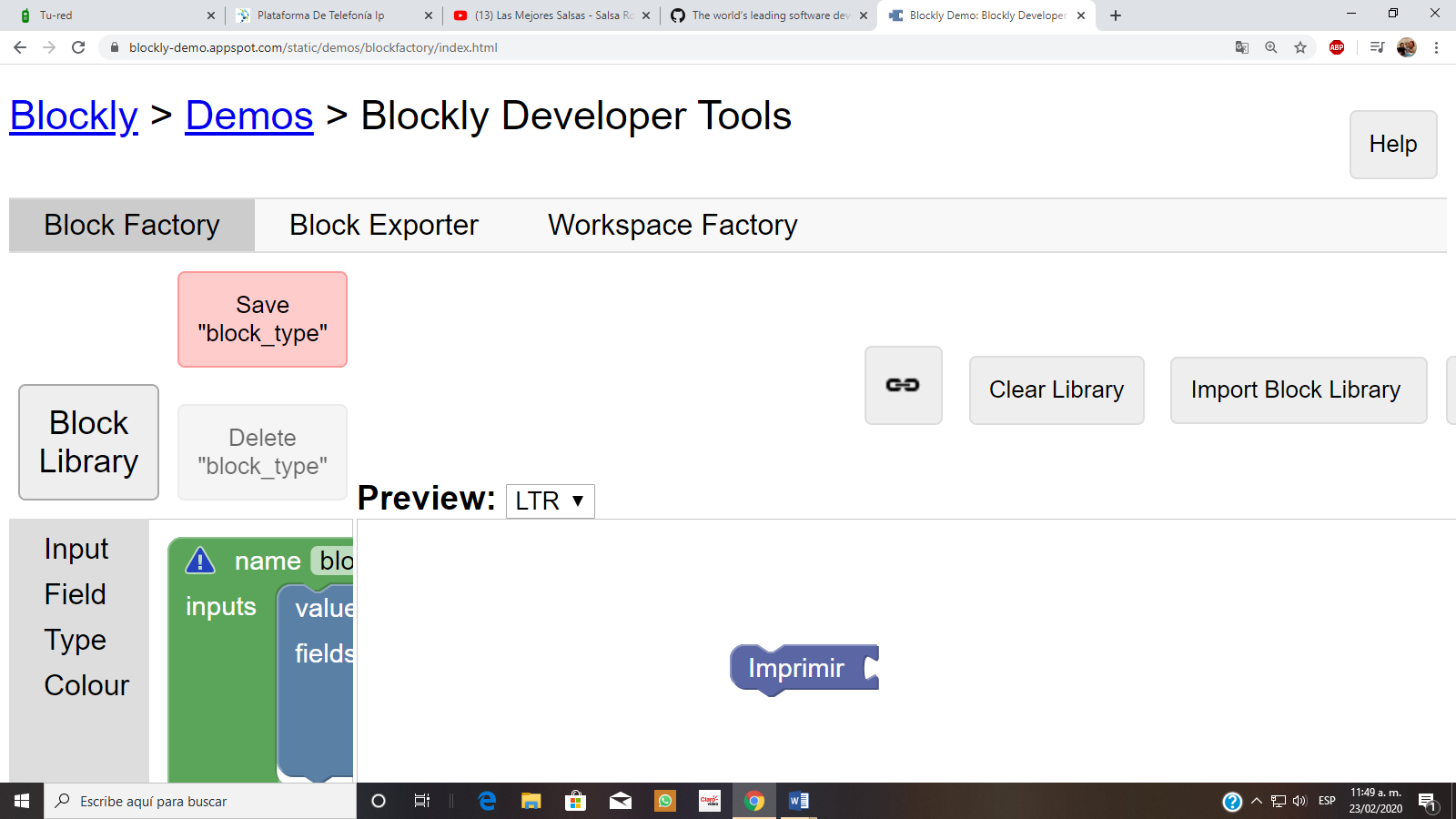
* Selección de bloques funcionales

Entre las funcionalidades de la tarjeta usa la Modulación por ancho de pulso (PWM), Conversor análogo digital (ADC), I2S, I2C, TouchPad,

* Creación de bloques bajo el entorno de Blockly

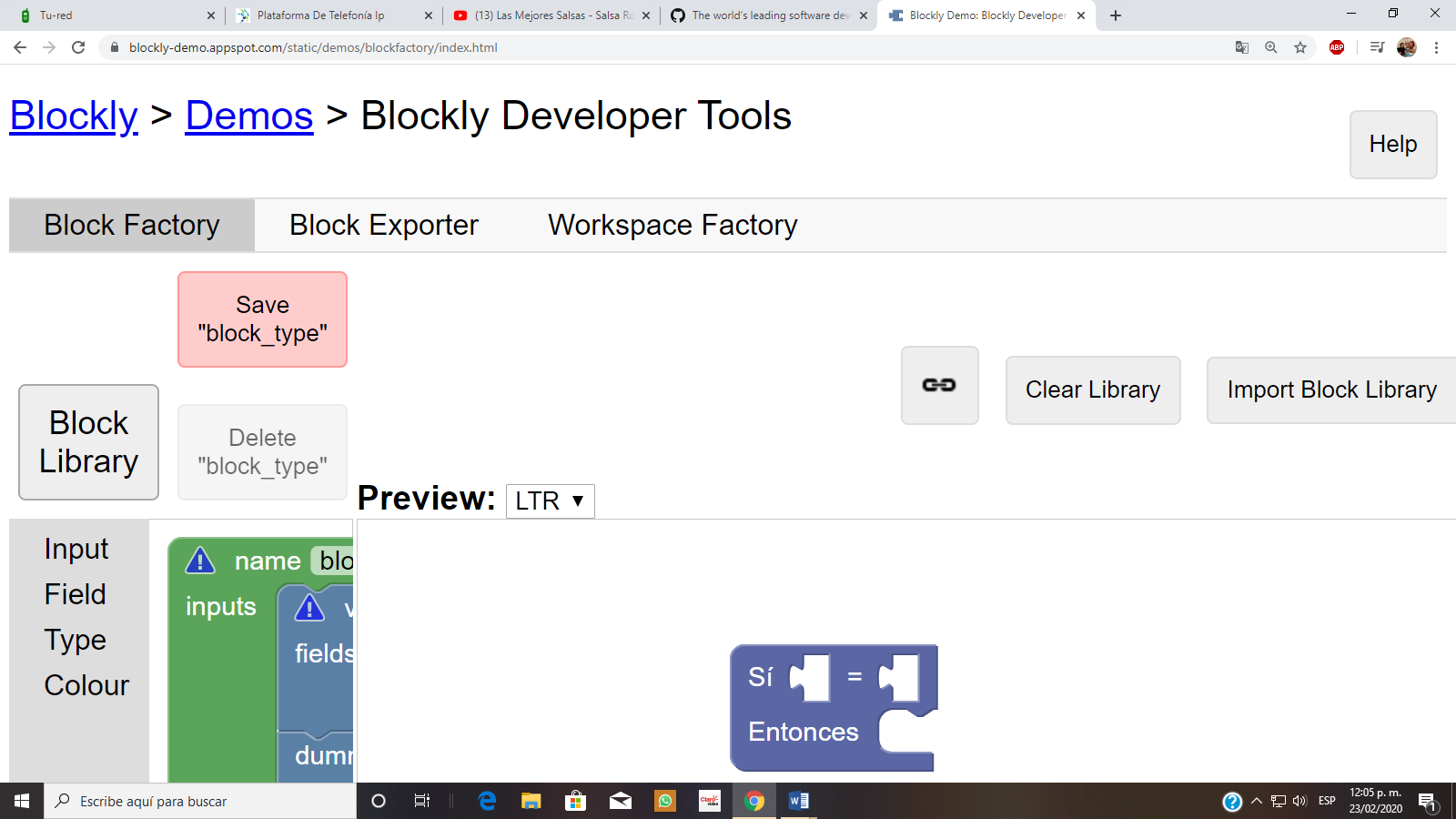
Usando Blockly Developer Tools permite crear bloques según sea su tipo.

En la figura (tal) se puede evidenciar la forma del bloque construido, la definición del bloque en JavaScript y su generador en Python. Los dos códigos generados al crear cada uno de los bloques son fundamentales al momento de hacer la integración con el programa base.

1. Paso de información: Estos bloques solo contienen información en su interior y esta puede ser transferida a otros bloques. En la figura (tal) se puede observar la forma del bloque, por ejemplo, este bloque contiene en un interior un número, el cual puede ser modificado por el usuario.
2. Recibir información: Estos reciben información del tipo 1 o del tipo 2, y en su interior pueden realizar algún procesamiento final. En la figura (tal) se puede observar la forma del bloque, por ejemplo, este bloque recibe información y la envía por el puerto serie.
3. Combinados: Estos pueden recibir información del tipo 1 y 3, procesar esta información recibida y además de ello pueden pasar información o retornar algún valor de cualquier tipo. En la figura (tal) se observa un bloque de función que analiza que el valor ingresado sea un número y retorna un valor booleano en caso de que este lo sea o no.

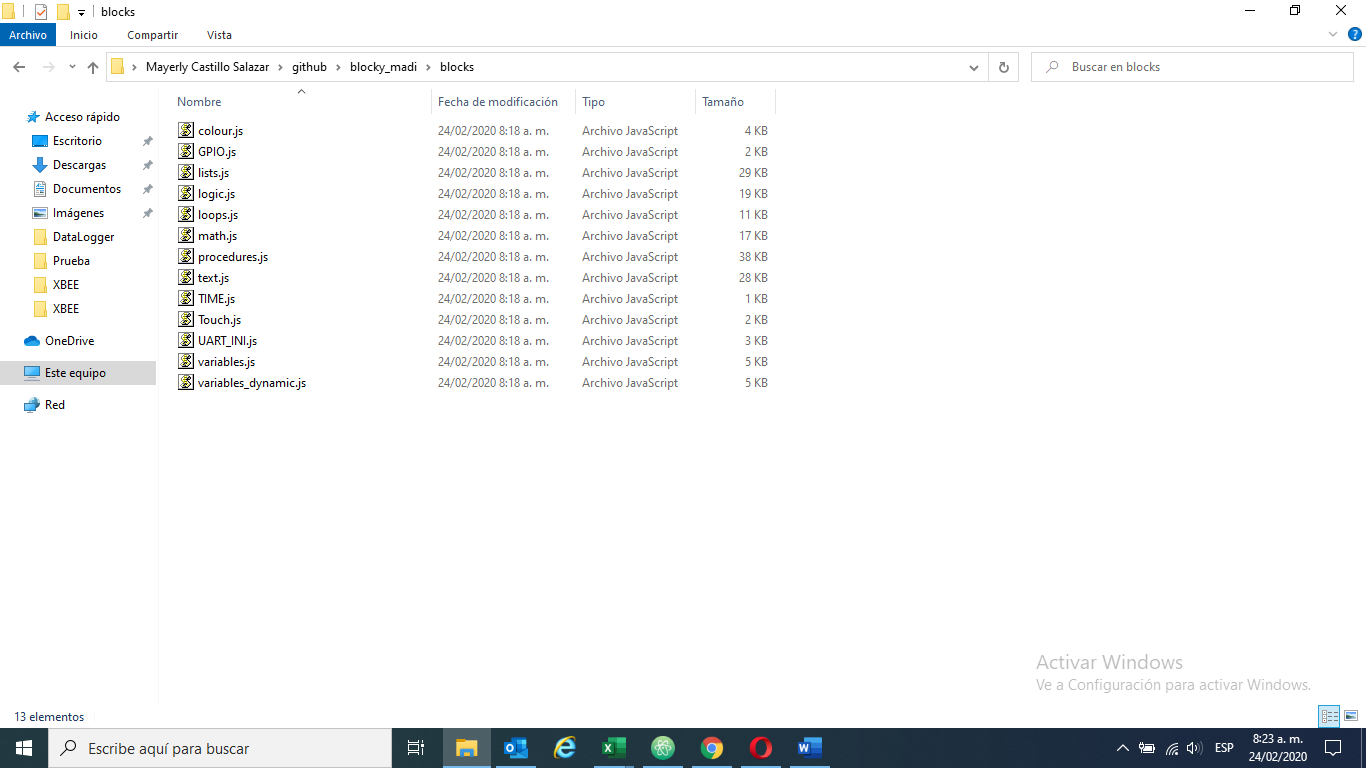


1. Condicionales, bucles y funciones: estos pueden evaluar alguna condición y realizar algunas acciones en específico, tales como repetir algún código, contener funciones entre otras. En la figura (tal) se presenta un bloque de tipo función que puede contener en su interior otros bloques.



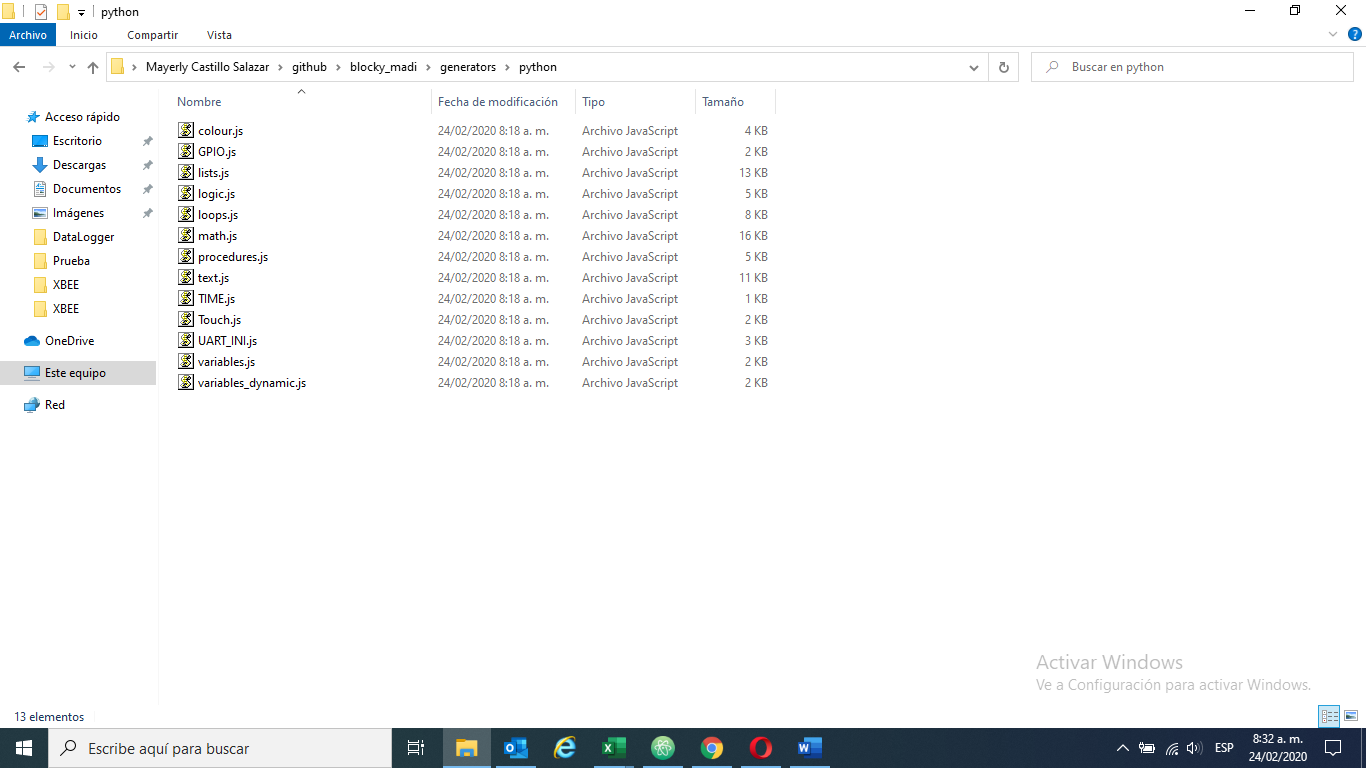
Ya con los tipos de bloques definidos se procede a programarlos, para lo cual primero se crea un archivo JavaScript y en este se ingresa la definición del bloque que Blockly Developer Tools genera, como se aprecia en la Figura (Tal)

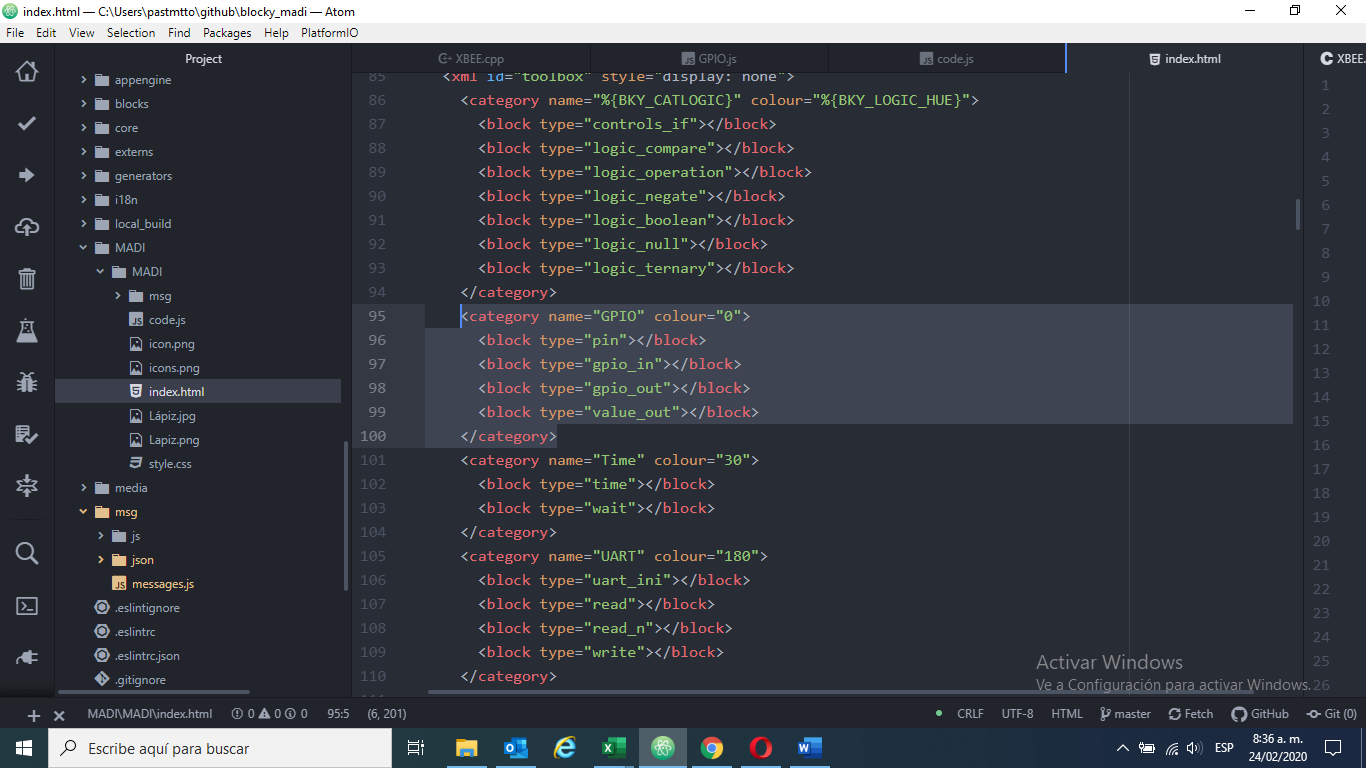


 Se guarda junto con los demás archivos .js (JavaScript) que vienen predeterminados en Blockly como en la Figura (Tal), esto con el fin de que al ser llamadas esas funciones no generen inconvenientes con la ruta de llamado.

Cuando ya se tenga este archivo guardado, lo siguiente es crear el archivo generador para el código en Python. Como se evidencia en la Figura (tal), se crea el archivo y se agrega el código generado en Blockly Developer Tools, y en la parte “ code = ‘…’ ” se agrega el código correspondiente en Python. La Figura (tal) es un ejemplo de generador Python.

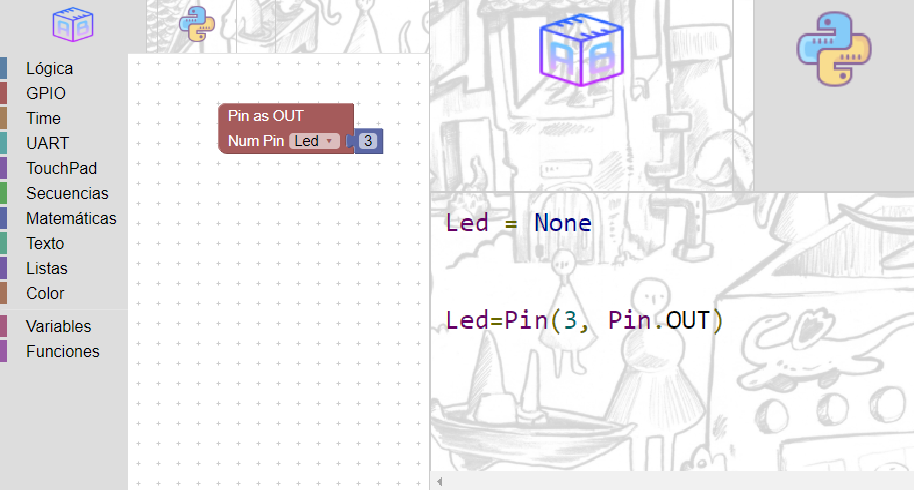


De igual forma que el archivo .js (Javascript) del bloque, este archivo generador se guarda en su ruta correspondiente de la carpeta de Blockly como en la Figura (tal).

Luego de haber realizado este proceso para agregar el bloque nuevo a Blockly, lo siguiente a realizar es el llamado desde la interfaz web, en el archivo HTML 5 como en la Figura (tal).

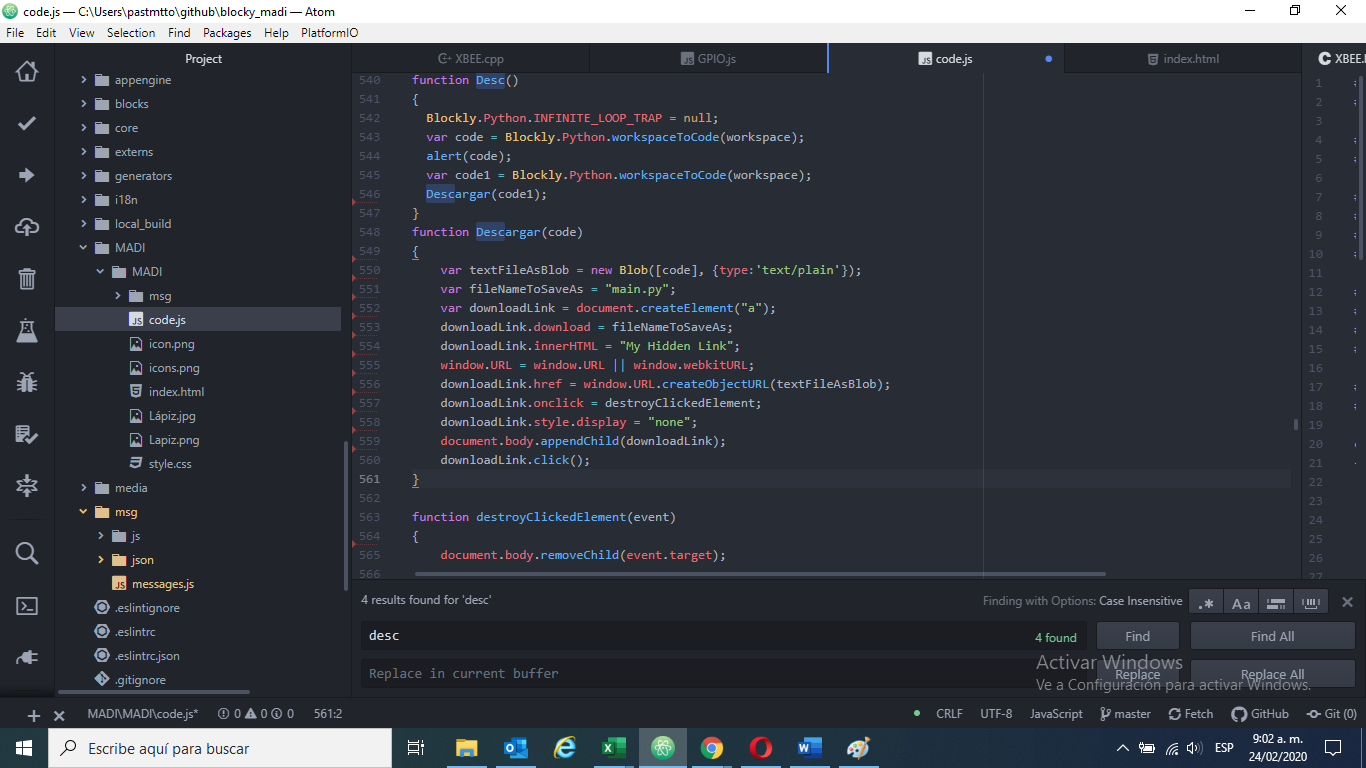
Ya realizado todo lo anterior, se procede a compilar todo el programa, para lo cual se abre la terminal (LINUX), se ingresa a la carpeta del proyecto y se ejecuta el archivo build.py como en la Figura (tal).

//Se agrega la foto de compilación en Linux.

Lo siguiente es comprobar que el bloque funcione y genere el código en Python correspondiente como en la Figura (tal).

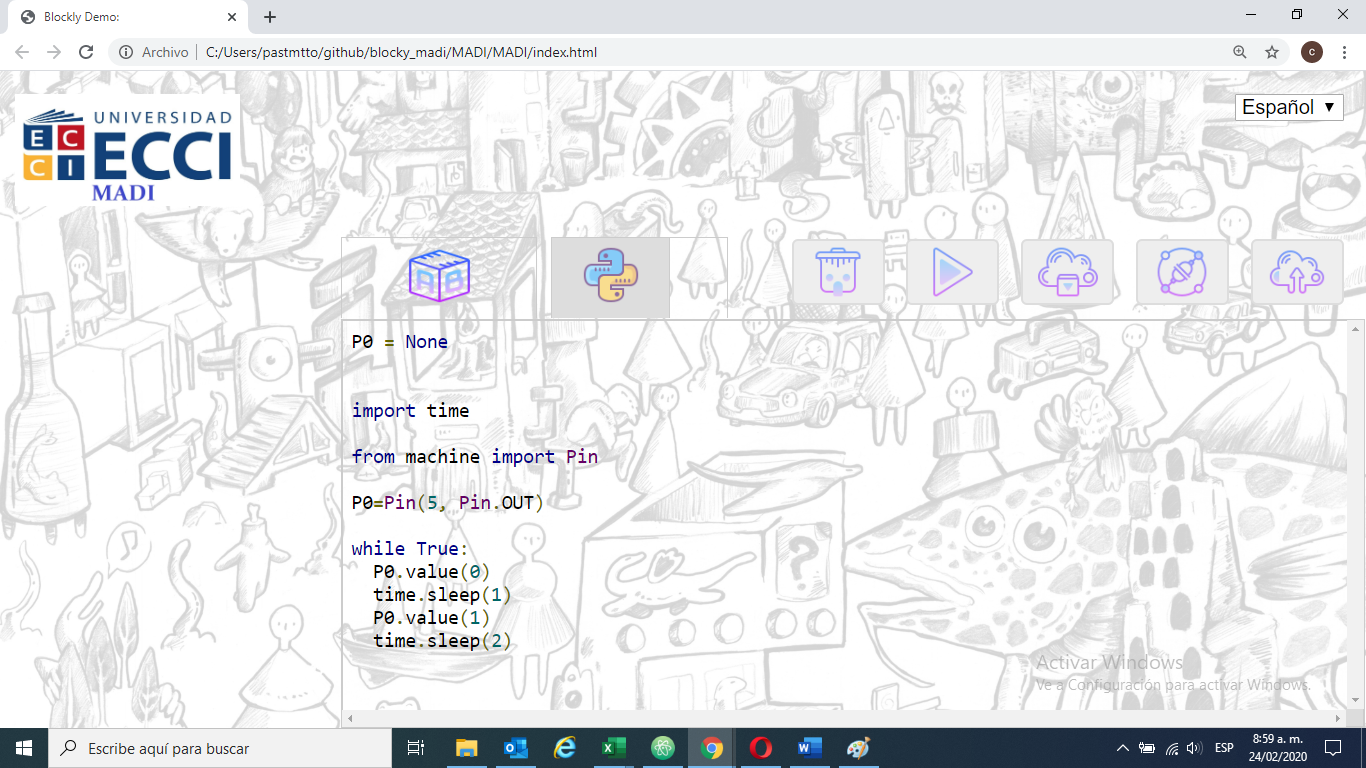
* Descargar Código generado en Python.

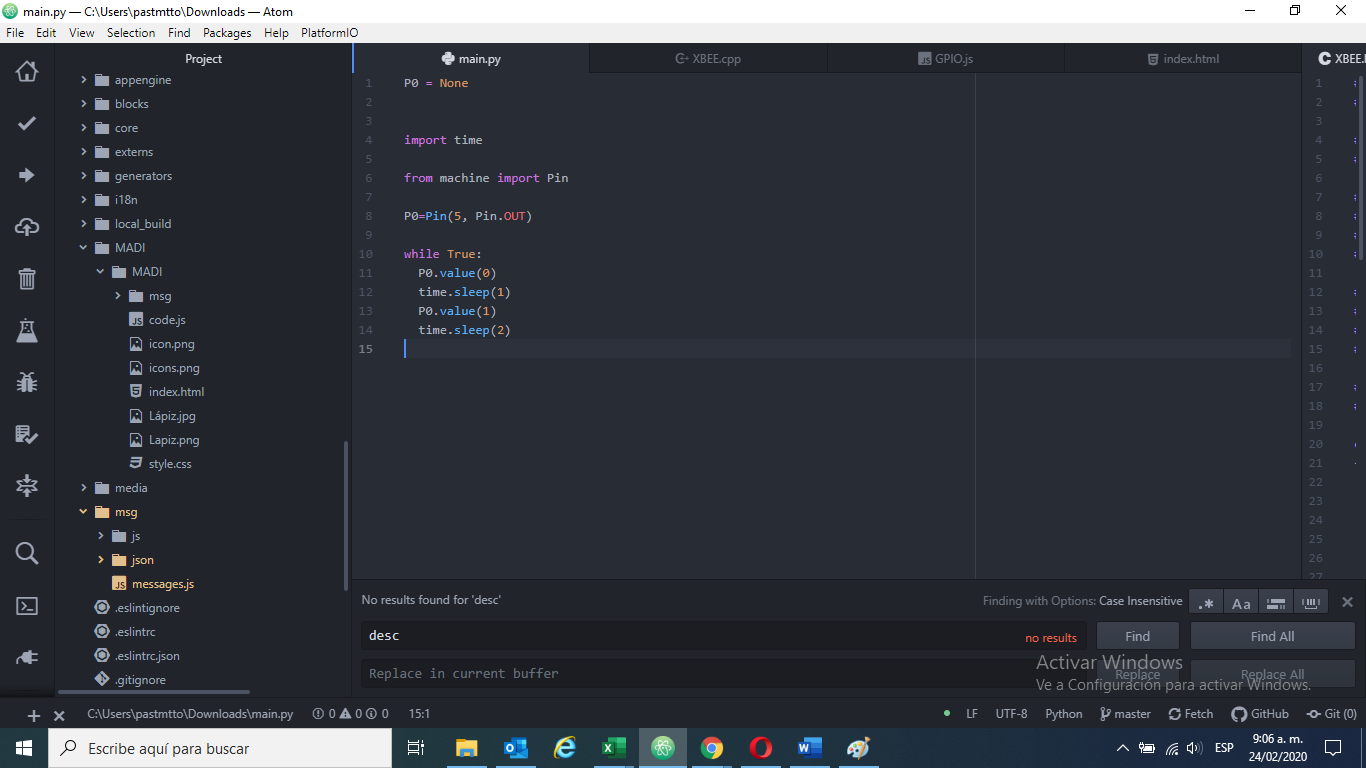
Se le agrega el botón de descarga del archivo .py (Python) y verificar que el programa realizado por el usuario sea el correcto. Ver Figura (tal)



Por tanto, se realiza la prueba realizando un programa tipo Blink (parpadeo) en Blockly como en la Figura (Tal).

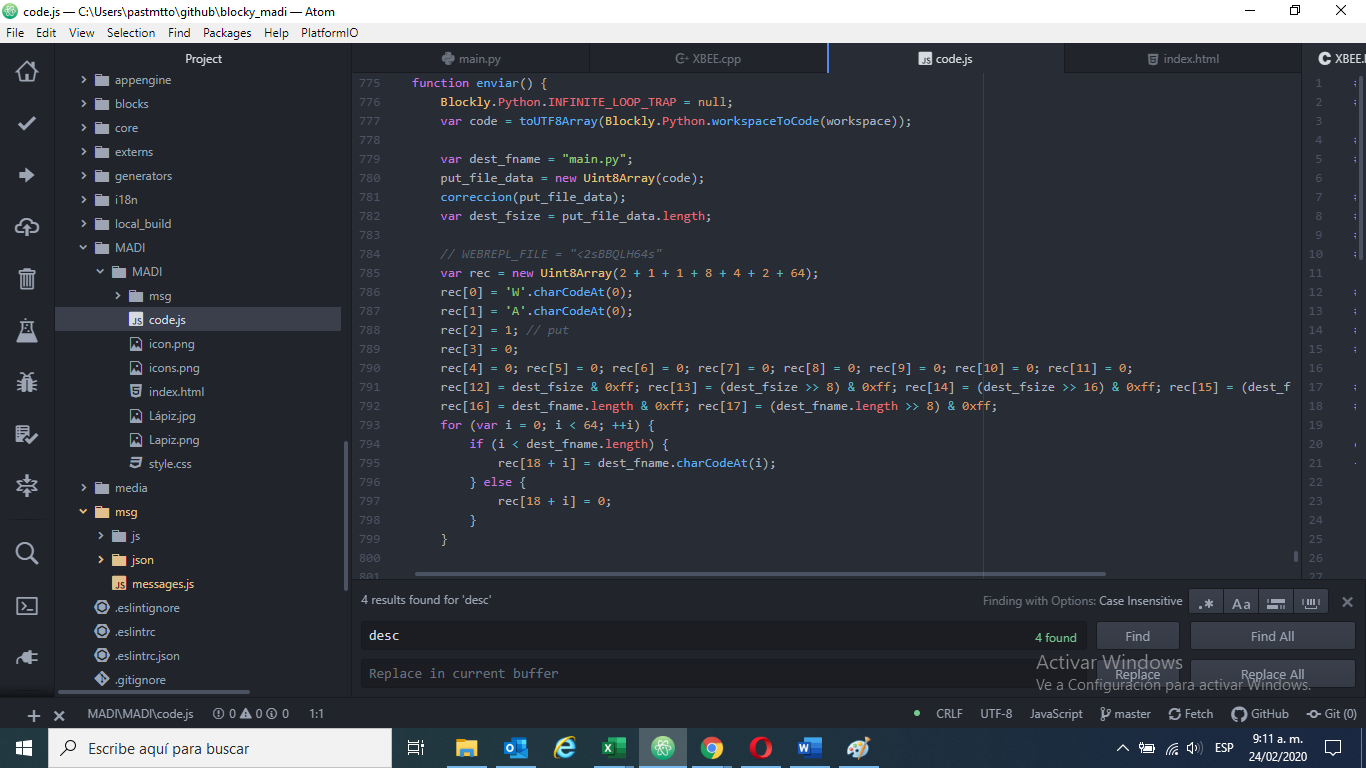


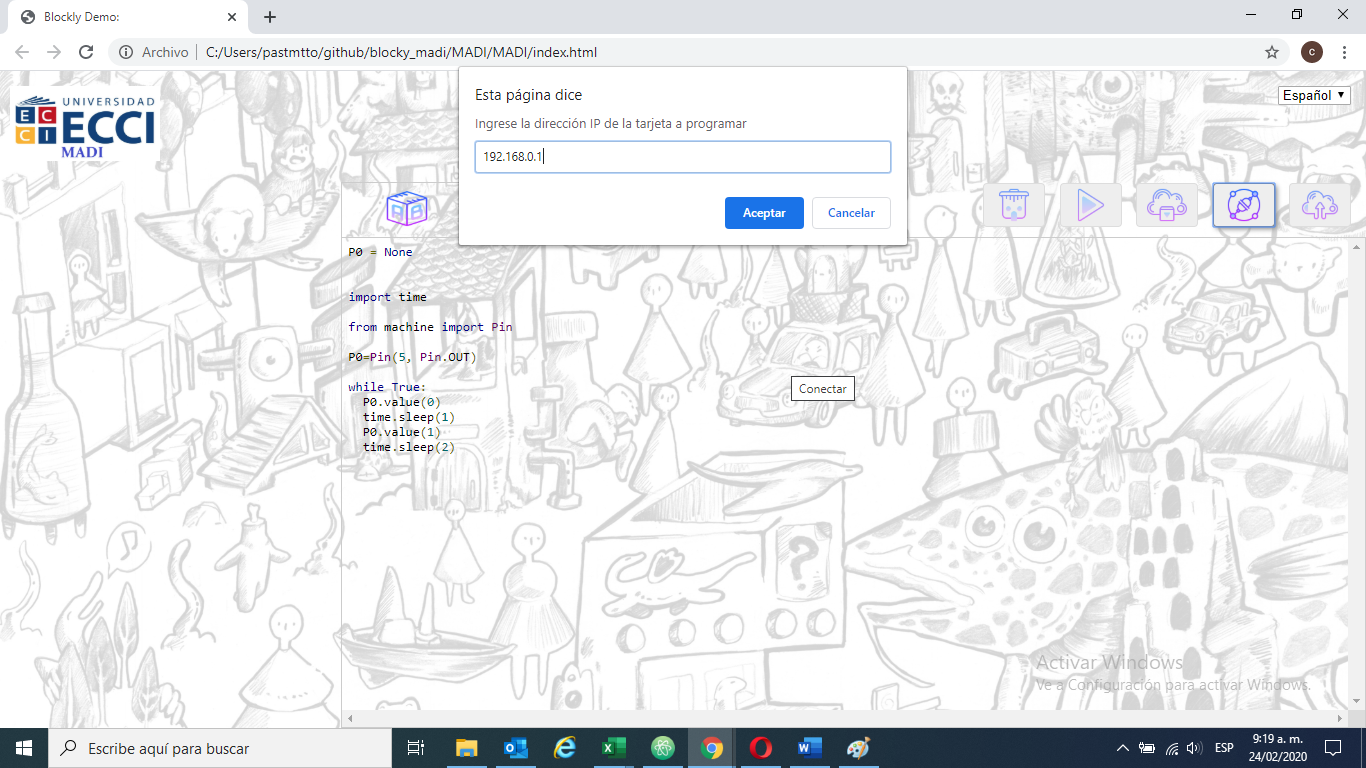
Se verifica el código en Python generado por la Web como en la Figura (tal).



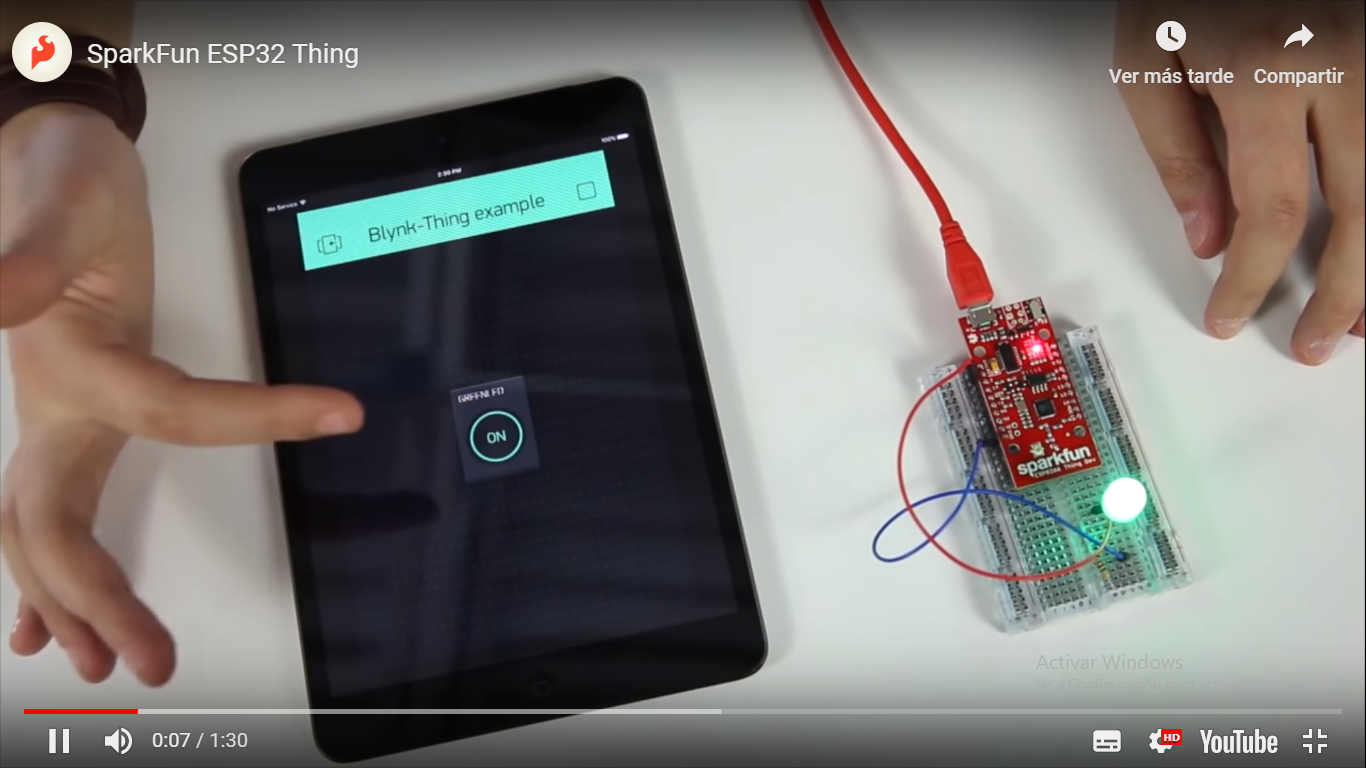
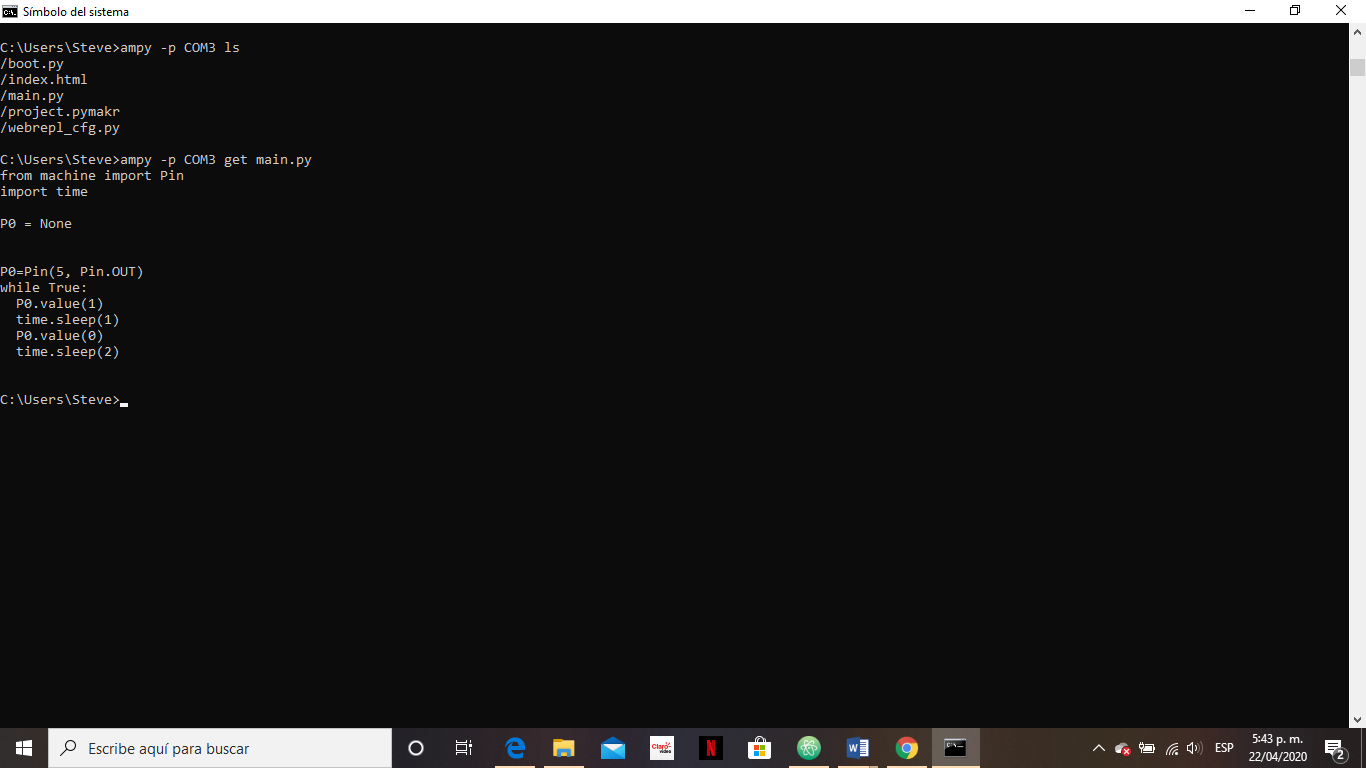
Se procede a descargar el archivo desde el botón agregado en la Figura (tal), ver Figura (tal) para ver el código descargado desde la Web.

* Envío Por WebSocket.

Las tarjetas de desarrollo ESP soportan WebSocket el cual puede ser aprovechado para poder enviar el programa generado desde la web, sin necesidad de tener conectada la tarjeta al computador, sino mediante una conexión vía red. Ver Figura (tal), código para enviar el archivo vía WebSocket.

 Primero se conecta la tarjeta ESP con la web como en la Figura (tal), esto con la dirección IP de la ESP, para este caso la dirección es 192.168.0.1 con puerto 8266.

Ya con el enlace establecido, se procede a enviar el archivo y se verifica que la tarjeta esté realizando lo programado, el cual fue el programa de la Figura (tal). Ver Figura (tal), Tarjeta realizando el Blink.

 A través de una terminal, se observa que archivos tiene la tarjeta en su memoria, y el contenido de cada archivo, Ver Figura (tal).

Hardware

* Selección de componentes eléctricos

1. Sensores
   1. HC-SR04
   2. MPU6050
   3. TouchPad
2. Actuadores

2.1) Servomotor

2.2) Motoreductor