

VINCIBOT

-MANUAL-



ÍNDICE

01

Introducción

Descubre VinciBot e as súas funcionalidades.

02

Conexión co ordenador

Información sobre a conexión do robot co ordenador para a súa posterior programación.

03

Software de programación

Coñecer o software de programación de VinciBot e as categorías dos bloques de programación.

04

Intelixencia Artificial

Coñecer as extensións de IA e como adestrar un modelo para utilizalo en proxectos de robótica.

Para calquera consulta ou asesoramento
escribe a **educacion@allknoweducation.es**

01

Introdución

O robot VinciBot de MatataStudio é unha innovadora ferramenta educativa deseñada para fomentar a aprendizaxe da programación e a robótica en nenos/as e mozos/as.

Cun enfoque no ensino práctico e a interacción lúdica, VinciBot permite ao alumnado explorar conceptos complexos de STEAM (ciencia, tecnoloxía, enxeñería, arte e matemáticas) dunha maneira accesible e atractiva. Equipado cunha variedade de sensores e módulos programables, este robot facilita a creación de proxectos interactivos que desenvolven habilidades como a resolución de problemas, o pensamento lóxico e a creatividade.

A continuación enuméranse os diferentes compoñentes que integran o robot:

- Actuadores:
 - Matriz de LEDs 16x8
 - 6 LED RGB programables
 - Altofalante
 - 2 motores
- Sensores:
 - Comunicación por infravermellos
 - Sensor de luz
 - Sensor de son
 - Sensor sigue-liñas
 - Sensor de cor
 - Botóns programables
 - Sensor de obstáculos (distancia)

No manual incluído na caixa do robot pódense atopar as imaxes de onde se atopa cada un dos compoñentes do VinciBot, así como os primeiros pasos para poñelo en funcionamento.

Outros datos interesantes relativos a VinciBot son:

- Ten un espazo no medio para colocar un rotulador, o que permite utilizar o **robot para pintar**.
- É **compatible con pezas tipo LEGO**, o que amplía posibilidades á hora de desenvolver proxectos. Nos eixes laterais podemos conectar pezas e crear diferentes proxectos de robótica.
- O seu software de programación conta cunha extensión de **Intelixencia Artificial** e outra de **IoT** para desenvolver proxectos nesta índole.

02

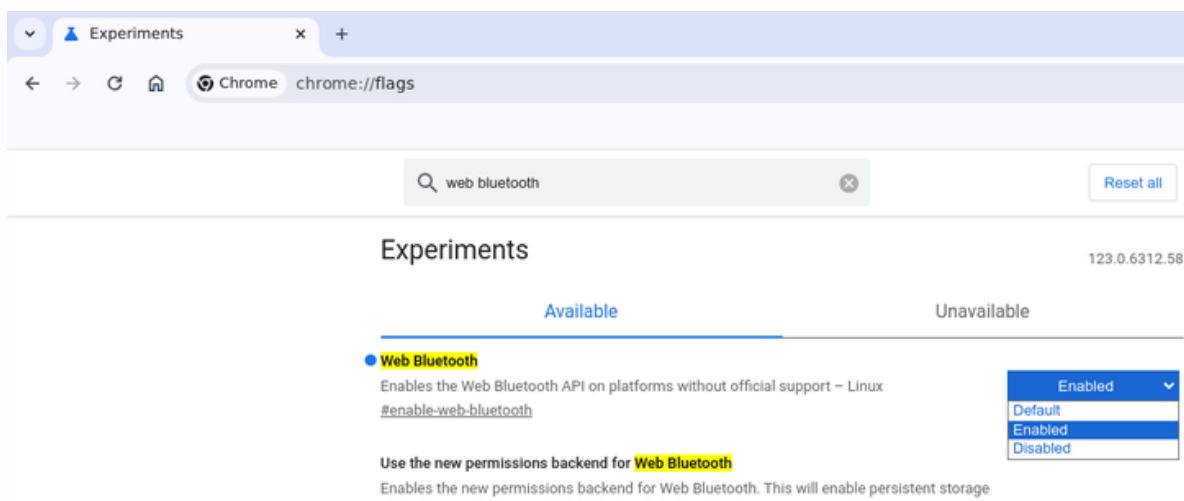
Conexión co ordenador

O robot VinciBot pode programarse a través dun software online compatible con calquera sistema operativo (entre eles Linux) e Google Chrome, a través dun software offline compatible polo momento unicamente con Windows e tamén a través dunha APP (MakeCode VinciBot) dispoñible para iOS e Android.

Neste manual centrarémonos exclusivamente no software online que é posible utilizar coa maqueta Abalar, ao que se accede a través da seguinte ligazón: <https://vinci.matatastudio.com/>.

Antes de explicar o funcionamento do software, comentar que o robot pode conectarse ao ordenador por cable ou Bluetooth, pero para esta última opción é imprescindible levar a cabo os seguintes pasos:

1. Abrir Google Chrome e buscar na barra do navegador: chrome://flags
2. Unha vez aberto, no buscador poñer "web Bluetooth"
3. No despregable asociado a Web Bluetooth, cambiar a opción a "Enabled"
4. Reiniciar o navegador.



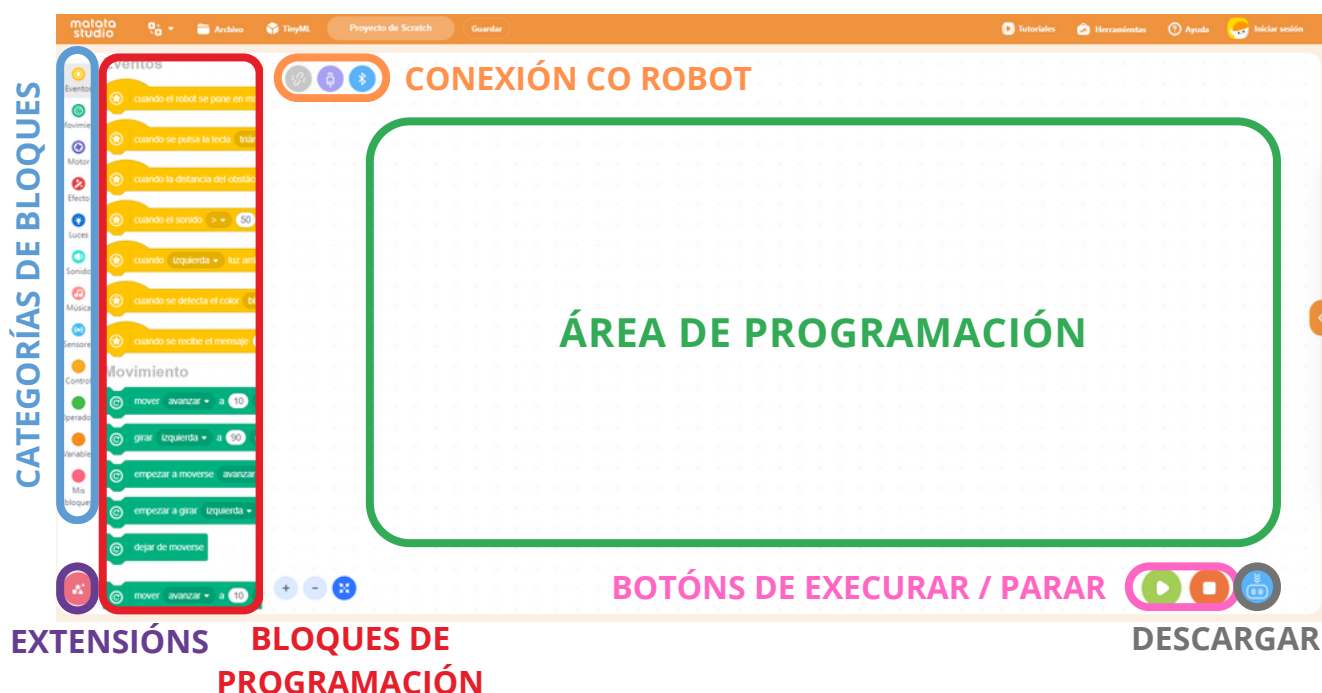
Agora si que estamos listos/as para descubrir o software de programación e coñecer as diferentes formas de utilizar o VinciBot.

03

Software de programación

Podemos acceder ao software online de programación a través da seguinte ligazón: <https://vinci.matatastudio.com/>. É un software inspirado en Scratch e polo tanto realízase unha programación visual por bloques, se ben permite tamén ver o código en linguaxe Python.

Coñezamos primeiro a súa interface:



Área de programación: espazo onde se pode arrastrar e soltar bloques de código. A programación, como en Scratch, é intuitiva e realízase combinando bloques para crear secuencias e estruturas lóxicas.

Categorías de bloques: os bloques de código están organizados en diferentes categorías en función da utilidade dos bloques, por exemplo, movemento, eventos, motor, efectos, etc.

Bloques de programación: os programas constrúense utilizando bloques de código de forma visual. Cada bloque representa unha acción ou comando específico, como mover un motor, mostrar algo na matriz LED ou reproducir un son.

Extensiones: neste apartado podemos cargar novas categorías de bloques para ampliar as posibilidades do programa. Aquí atoparemos as extensións de IA e IoT, así como as correspondentes aos diferentes compoñentes de kits compatibles con VinciBot.

Conexión co robot: neste apartado temos dúas posibilidades de conexión co robot, por cable ou Bluetooth. Isto resulta moi práctico para ir facendo modificacións nas programacións e ver os cambios en tempo real, sen necesidade de descargar.

Botóns de executar / parar: estes botóns podémolos utilizar para activar o código cando utilizamos o evento de “Cuando el robot se pone en marcha” e para parar o robot en calquera momento, independentemente das accións que estea levando a cabo.

Para poder utilizar estes botóns é imprescindible estar conectado por cable ou Bluetooth.

Descargar: o botón de descarga permítenos cargar o programa no propio robot para poder utilizalo en calquera momento. Se ben para poder descargalo o software ten que estar conectado ao robot de calquera das maneiras, unha vez que lle descargamos o programa podemos utilizalo sen necesidade de conexión.

A continuación explícanse en detalle as diferentes categorías de bloques que se atopan no programa:



Eventos: esta categoría inclúe bloques relacionados con eventos que desencadean a execución do programa.



Movimientos: neste apartado atopamos tódolos bloques para controlar o movemento do robot a través das rodas.



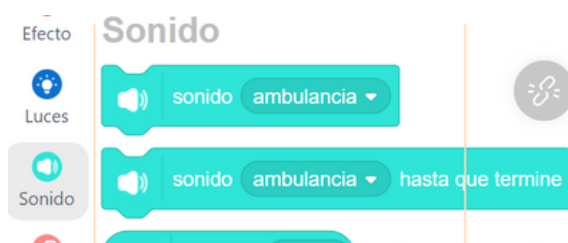
Motor: neste apartado atopamos tódolos bloques para controlar o movemento dos eixes laterais do robot. Debemos ter en conta que o motor é o mesmo que o da roda, polo tanto, programar os eixes é como programar as rodas. Se por necesidades do proxecto queremos mover os eixes e non as rodas podemos poñer o robot ao revés, así aínda que se movan os motores o robot non se despraza.



Efecto: nesta categoría atopamos bloques para programar efectos no robot que implican son, luces e movemento.



Luces: nesta categoría atopamos tódolos bloques para programar a matriz de LEDs frontal para mostrar iconas, texto, etc... Ademais dos bloques que permiten programar os 6 LEDs RGB que atopamos na parte de arriba do robot.



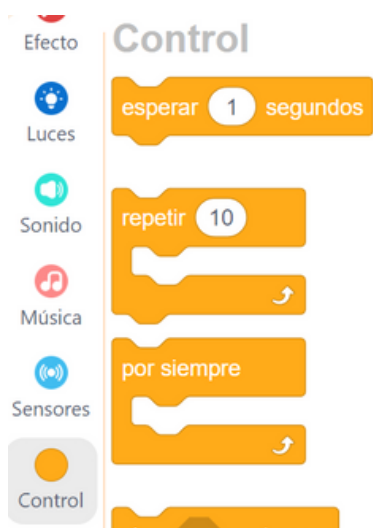
Sons: nesta categoría atopamos tódolos bloques para programar os sons que pode emitir o robot, podendo escoller entre moitas opcións.



Música: nesta categoría atopamos tódolos bloques para programar sons que pode emitir o robot, pero relacionados con instrumentos musicais e notas. Nesta categoría podemos crear melodías propias.



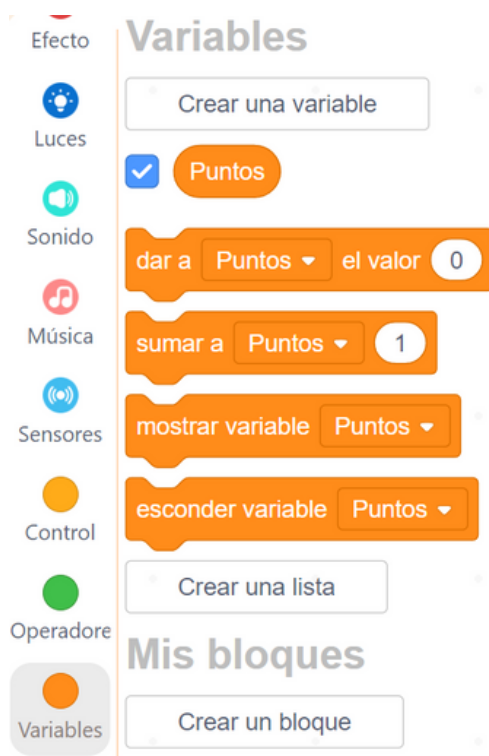
Sensores: nesta categoría atopamos tódolos bloques para poder utilizar os valores que recolle o robot a través dos diferentes sensores. Ademais, tamén aqueles que permiten programar un temporizador.



Control: esta categoría contén bloques que permiten controlar o fluxo de execución do programa, como bucles e estruturas de decisión (condicionais).



Operadores: estes bloques inclúen operadores matemáticos e lóxicos que permiten realizar cálculos e comparacións nos programas.



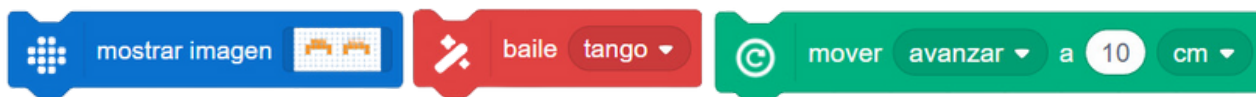
Variables: esta categoría inclúe bloques para crear e manipular variables, permitindo almacenar e recuperar información durante o programa. Por defecto os bloques aparecen ocultos, debemos crear unha variable para que se mostren.

Cando creamos unha variable ábrese una pantalla no lateral dereito do programa onde podemos ir vendo o seu valor.

Podemos ver que dentro de cada categoría atopamos bloques con distintas formas. Esas formas axudan a identificar rapidamente que tipo de instrución ou función está a utilizarse no código, o que facilita a escritura e a comprensión do programa. Os bloques están deseñados para encaixar entre si de forma lóxica e coherente e combinándoos é como se crean os programas que executará o robot VinciBot. En xeral podemos diferenciar os seguintes tipos de bloques:



Eventos: bloques que indican cando se van a levar a cabo as accións. É por iso que unicamente podemos enganchar outros bloques por abaixo.



Acción: bloques que se poden unir entre eles para crear unha secuencia e, por suposto, sempre teñen que ir despois dun bloque de evento para que a acción (ou accións) saiba cando debe executarse. Tamén se utilizan como consecuencia nunha estrutura condicional.



Estructuras condicionais: bloques para programar condicionais. Permiten introducir unha ou varias condicións (bloques booleanos), así como a consecuencia de que se cumpra ou non dita condición (bloques de accións). Utilizando este tipo de bloques enriquecemos os nosos programas creando proxectos de robótica máis complexos.



Control: bloques que permiten controlar o fluxo de execución do programa. Non son accións en si mesmas, senón que lles din ás accións que ten dentro como comportarse.



Booleanos: bloques que internamente gardan dous tipos de datos: 1 ou 0, é dicir, verdadeiro ou falso. Utilízanse sempre como condicións en bucles e condicionais e teñen forma triangular.



Variables: bloques redondos que gardan datos, habitualmente numéricos. Utilízanse para mostrar información, facer operacións con elas ou crear comparativas que funcionan como condicións nas condicionais ou bucles.

04

Intelixencia Artificial

Imos explicar neste apartado como funciona a extensión de Intelixencia Artificial, pero antes de analizar os bloques de programación da extensión, vexamos paso a paso como adestrar modelos para desenvolver proxectos nesta liña.

Paso 1: importante! Antes de empezar debemos crearnos unha conta para poder **iniciar sesión** e tamén ter o **robot conectado** ao programa, ben sexa por cable ou Bluetooth.



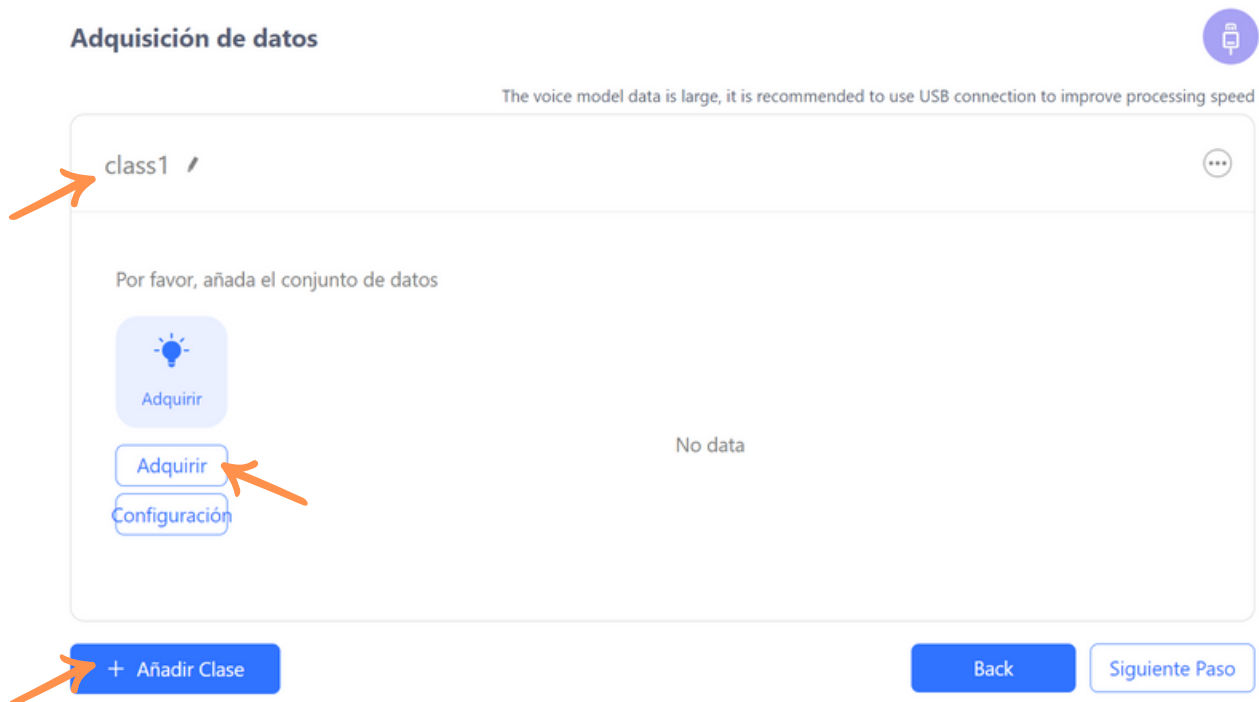
Paso 2: acceder a **TinyML** para **crear un modelo** utilizando tecnoloxía Machine Learning e seleccionar entre as diferentes posibilidades dispoñibles: recoñecemento de imaxes (non dispoñible para usar simplemente co robot), recoñecemento de palabras ou recoñecemento de obxectos. Unha vez seleccionado terás que poñerlle un nome ao proxecto para despois poder acceder e editar o modelo.

Os seguintes pasos son iguais independentemente da opción que escollamos, polo que o explicaremos utilizando o recoñecemento de palabras.



Paso 3: crear as diferentes clases. As clases son a forma en que definimos e clasificamos as diferentes cousas que queremos que o programa recoñeza. Para crealas seguimos os seguintes pasos:

- Editar o nome das clases en función da información que van a gardar e crear tantas clases como sexa necesario.
- Dentro de cada clase facer clic no botón de adquirir para recoller os distintos tipos de mostras, ven sexa falando ou mostrando un obxecto ao robot. Tamén podemos pulsar en configuración para cambiar o tempo de recollida da mostra ou o retardo na recollida.



Paso 4: unha vez lista a fase de adquisición de datos debemos pasar ao adestramento e a implementación. Para iso pulsamos no botón de “siguiente paso”.

Debemos ter en conta que cantas máis mostras teñamos en cada clase mellor funcionará a nosa IA. Igualmente, os modelos quedan gardados ao iniciar sesión e sempre podemos volver a eles para melloralos.

Adquisición de datos

The voice model data is large, it is

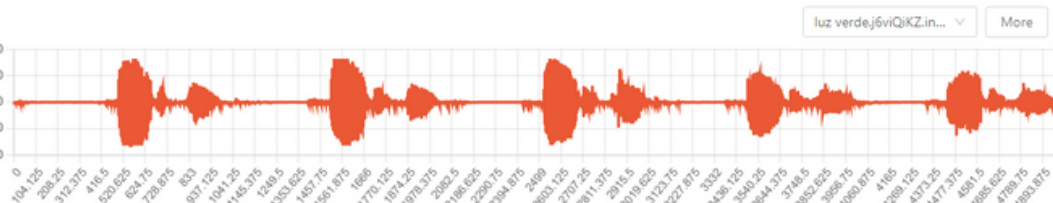
luz vermella /

Total 4 Samples



luz verde /

Total 4 Samples



Back

Siguiente Paso

Paso 5: iniciar o adestramento. Pode tardar uns minutos e dependerá do número de clases e de mostras que teñamos.

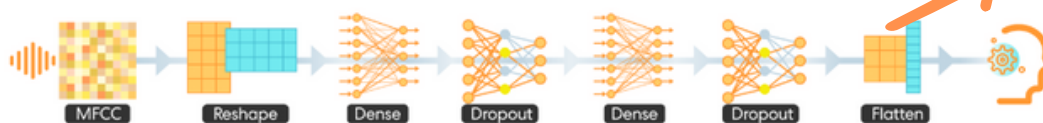


Entrenamiento e implementación

Informe de entrenamiento del modelo

Class: 3 Sample: 13

Selección de la escala de la red neuronal: **Pequeño**



Número de ciclos de entrenamiento: Tasa de aprendizaje: Índice de confianza mínimo:

window size(ms): window increase(ms):

Informe de entrenamiento del modelo

Registro

Paso 6: implementar o modelo. Unha vez realizado o adestramento aparécenos un informe que nos indica a precisión das mostras. Se a precisión é baixa podemos volver para incluír máis mostras, pero se conseguimos que sexa do 100% ou cerca podemos pulsar en “implementación do modelo”.

Isto nos vai a permitir ter acceso ao modelo dende os bloques de programación da extensión TinyML.

Informe de entrenamiento del modelo

Rendimiento del último entrenamiento



Precisión
100.0%



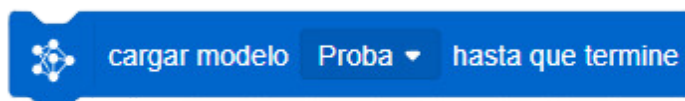
Pérdida
0.02

Implantación del
modelo

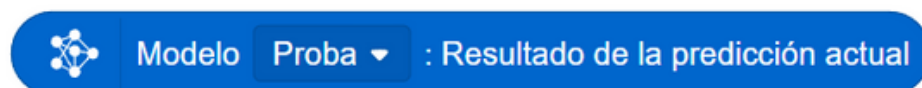
Matriz de confusión

	luz vermella	luz verde	silencio
luz vermella	100%	0%	0%
luz verde	0%	100%	0%
silencio	0%	0%	100%
F1 Score	1.00	1.00	1.00

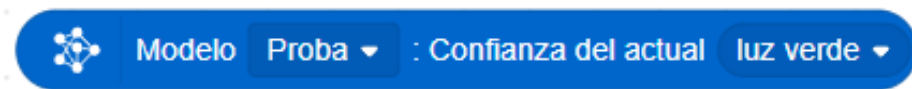
Vexamos a continuación os bloques de programación que podemos utilizar para programar unha IA en base ao modelo creado:



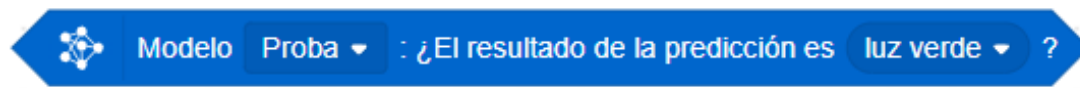
Este bloque permítenos cargar o modelo. Pinchando no despregable aparecerán os modelos que temos creados (ou outros de exemplo) para escoller o que queremos utilizar en cada programa.



Este bloque garda información, neste caso a clase detectada das do modelo que se está comprobando.



Bloque de variable que guarda o índice de confianza en relación ao recoñecemento que está a facer o programa da clase seleccionada no desplegable dentro do modelo seleccionado. Utiliza valores dende o 0 ata o 1, onde o 1 indica que hai un 100% de coincidencia. Podemos utilizar este bloque para facer a nosa IA máis precisa.



Bloque booleano para comprobar se se está detectando unha clase concreta (seleccionar no desplegable) dentro dun modelo concreto (seleccionar no desplegable). As opcións son dúas, falso/non (0) ou verdadeiro/si (1).

Estes bloques utilízanse nas concidionais, como condición para que algo suceda.