



P1902010 - Aplicaciones educativas como elemento dinamizador de aula

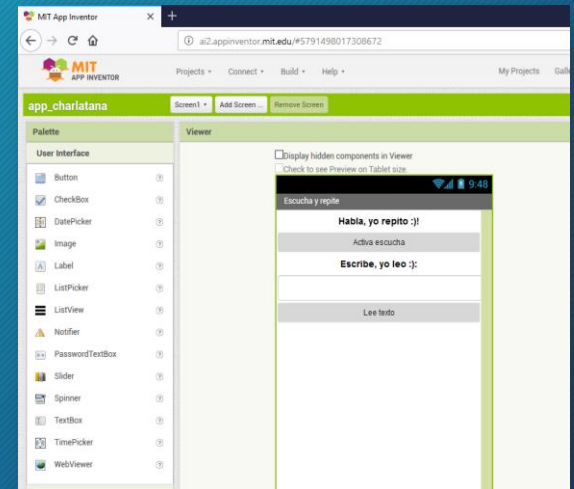
Relator: Iván Gómez Conde

Email: ivangconde@edu.xunta.es

Asesor: Manuel Abelleira Barros

Planificación do curso

- Introducción a APP Inventor (diferenza entre deseño da aplicación e programación dos bloques)
- Conexión co noso dispositivo móbil ou tableta.
- Posible instalación da nosa app no dispositivo.
- Creación de diversas apps facendo uso de distintos sensores e medios.
- Creación de apps con intelixencia artificial empregando Machine Learning for Kids (ML4K)



Introducción a APP Inventor



- Creada polo Instituto Tecnolóxico de Massachusetts (MIT) e Google cun primeiro lanzamento en 2010. (necesitamos unha conta de Google para identificarnos)

A screenshot of the MIT App Inventor website homepage. The browser address bar shows 'appinventor.mit.edu/explore/'. The page features a navigation menu with 'About', 'News & Events', and 'Resources', along with a 'Create apps!' button. A search bar is present with the text 'Google Custom'. Below the navigation, there are social media icons and a 'Donate!' button. The main content area includes a large banner for the 'MIT App Inventor Summit' on August 1-3, 2019, in Cambridge, MA. Another banner announces the 'MIT App Inventor Youth Mobile Power App Challenge!'. A video player shows a man speaking, with the caption 'Mass TLC 2017 Distinguished Leader Award'. A sidebar on the right offers 'Try our App Building Guides for the Youth Mobile Power Series' and mentions a partnership with 'YR Media'. At the bottom, a statistics bar shows: 'Active Users: This Month: 964.9K This Week: 319.1K Today: 33.6K' and 'Registered Users: 8.2M Countries: 195 Apps Built: 34.0M'. The footer states 'App Inventor code is open source.'

<http://appinventor.mit.edu/explore/>

Características de APP Inventor 2



- Ferramenta Web de desenvolvimento para dispositivos móveis.
- Desenvolvimento sinxelo de apps.
- Creación de apps para Android
(para iOS lanzouse en 2019 unha versión beta)
- Baseado en programación por bloques.
- Orientado a eventos.
- Alternativas: Thunkable (permite apps para iOS), AppyBuilder, Kodular...



Ser *beta tester* para App Inventor con iOS



- Lanzamento: 25 de Setembro de 2019
- Versión 0.9 de MIT APP Inventor companion para iOS.
<https://appinventor.mit.edu/blogs/evan/2019/09/25/mit-app-inventor-ios-version-0-9>
- Formulario a cubrir para ser beta tester en iOS:
<https://docs.google.com/forms/d/e/1FAIpQLScUvWy9KyFFwYgA99h6J3yrwHLYorhW48MaEPqlrVpD4z2Fig/viewform>
- No caso de resultar seleccionado é preciso instalar a app TestFlight de Apple dende APP Store.
- O obxectivo desta versión é buscar erros e informar deles.



Diseñador da interface



The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=es_ES#6558793857630208`. The interface is divided into several panels:

- Paleta (User Interface):** A list of UI components including Botón, CasillaDeVerificación, SelectorDeFecha, Imagen, Etiqueta, SelectorDeLista, VisorDeLista, Notificador, CampoDeContraseña, Deslizador, Desplegable, CampoDeTexto, SelectorDeHora, and VisorWeb.
- Visor (Preview):** A central area showing a preview of the app. It includes a status bar with the time 9:48 and a toolbar with buttons for 'Negro', 'Rojo', 'Verde', and 'Azul'. Below the preview are buttons for 'Limpiar', 'Selector Imagen', and 'Guardar'.
- Componentes (Components):** A tree view showing the hierarchy of components on the screen, including 'Screen1', 'MarcoHorizontal', 'Negro', 'Rojo', 'Verde', 'Azul', 'DisposiciónHorizontal2', 'Deslizador1', 'Lienzo1', 'DisposiciónHorizontal1', 'Limpiar', 'SelectorDeImagen1', 'Guardar', and 'Etiqueta1'.
- Propiedades (Properties):** A panel on the right showing the properties for the selected component, 'Screen1'. Properties include 'PantallaAcercaDe', 'AccentColor', 'DispHorizontal', 'DispVertical', 'AppName', 'ColorDeFondo', 'ImagenDeFondo', 'AnimaciónCierreDePantalla', 'Icono', and 'AnimaciónAlAbrirPantalla'.

A red arrow points to the 'Diseñador' tab in the top right corner of the interface.

Programación por bloques



The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=es_ES#6558793857630208`. The interface is divided into several sections:

- Top Bar:** Includes navigation icons (back, forward, refresh, home) and a search bar. The MIT App Inventor logo is on the left, and user information (Español, ivangomezcondelp...) is on the right.
- Project Management:** A green bar with buttons for "Dibujar", "Screen1", "Añadir ventana", and "Eliminar ventana". On the right, there are buttons for "Diseñador" and "Bloques". A red arrow points to the "Bloques" button.
- Bloques (Left Panel):** A sidebar with a tree view showing categories like "Integrados", "Control", "Lógica", "Matemáticas", "Texto", "Listas", "Colores", "Variables", "Procedimientos", and "Screen1". Under "Screen1", there are sub-categories for "MarcoHorizontal" (with color options: Negro, Rojo, Verde, Azul) and "DisposiciónHorizontal" (with "Deslizador1").
- Visor (Main Area):** The central workspace for block programming. It contains several event-driven blocks:
 - cuando Lienzo1 . Arrastrado:** Triggers "ejecutar" blocks: "llamar Lienzo1 . DibujarLinea" (with parameters x1, y1, x2, y2 and "tomar XPrevio", "tomar YPrevio", "tomar XActual", "tomar YActual") and "draggedAnySprite".
 - cuando Deslizador1 . PosiciónCambiada:** Triggers "ejecutar" blocks: "poner Lienzo1 . AnchoDeLinea" and "tomar posiciónDePulgar".
 - cuando SelectorDeImagen1 . DespuésDeSelección:** Triggers "ejecutar" block: "poner Lienzo1 . ImagenDeFondo" and "SelectorDeImagen1".
 - cuando Negro . Clic:** Triggers "ejecutar" block: "poner Lienzo1 . ColorDePintura" and "Negro".
 - cuando Rojo . Clic:** Triggers "ejecutar" block: "poner Lienzo1 . ColorDePintura" and "Rojo".
 - cuando Verde . Clic:** Triggers "ejecutar" block: "poner Lienzo1 . ColorDePintura" and "Verde".
 - cuando Azul . Clic:** Triggers "ejecutar" block: "poner Lienzo1 . ColorDePintura" and "Azul".
 - cuando Limpiar . Clic:** Triggers "ejecutar" block: "llamar Lienzo1 . Limpiar".
 - cuando Guardar . Clic:** Triggers "ejecutar" block: "poner Etiqueta1 . Texto" and "llamar Lienzo1 . Guardar".
- Bottom Bar:** Includes a "Mostrar avisos" button and a "Cambiar nombre" button.

¿Que é un programa?



- A RAG define un programa como: “Conxunto de instrucións e datos que se introducen nun ordenador para poder realizar unha serie de operacións”
 1. Programación textual
 2. Programación visual ou por bloques

1. Programación textual



```
public class Aclaracion
{
    private int atributo1;
    private int atributo2;
    private String atributo3;

    //Declaramos un constructor
    public Aclaracion(int attr1, int attr2, String attr3)
    {
        atributo1 = attr1;
        atributo2 = attr2;
        atributo3 = attr3;
    }

    public static void main(String[] args)
    {
        Aclaracion ac = new Aclaracion(5, 10, "x");//Creamos un objeto enviando p

        System.out.println(ac.atributo1 + ", " + ac.atributo2 + ", " +ac.atributo
        //Imprime '5, 10, x'
    }
}
```

Programación textual (seguido o léxico, sintaxe e semántica de un lenguaje)

2. Programación por bloques



```
como iniciar_juego
ejecutar
  poner global acabado a falso
  poner global tiempo_limite a 15
  poner valor_tiempo . Texto como unir "Tiempo: "
  tomar global tiempo_limite
  poner global objetos_restantes a 5
  poner Temporizador_x_segundo . TemporizadorHabilitado como cierto
  Llamar colocar_objetivos

cuando Reiniciar_Juego .Clic
ejecutar Llamar iniciar_juego
```

Programación visual (construimos los nuestros programas encaixando bloques que corresponden coas instrucións)

Programación por bloques en App Inventor



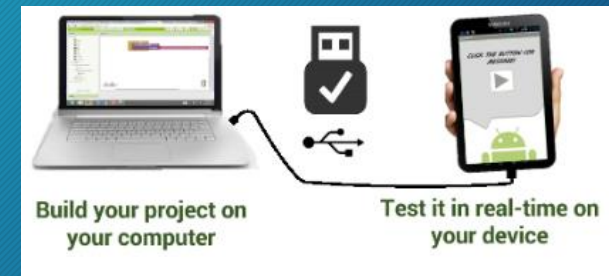
The screenshot shows the MIT App Inventor interface for a project named 'app_charlatana'. The top navigation bar includes 'Projects', 'Connect', 'Build', and 'Help'. Below the project name, there are buttons for 'Screen1', 'Add Screen...', and 'Remove Screen'. The interface is divided into two main sections: 'Blocks' on the left and 'Viewer' on the right. The 'Blocks' section contains a palette of categories: Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under 'Screen1', several components are listed: ET_Repetir, BT_Escucha, ET_Texto, CT_Texto_Usuario, BT_Lee, TextoAVoz1, ReconocimientoDeVoz, and Acelerómetro1. The 'Viewer' section displays a sequence of blocks: 'initialize global name to', 'get ReconocimientoDeVoz1', 'set ReconocimientoDeVoz1', 'initialize local name to', and 'initialize local name to'. A 'when Acelerómetro1 Shaking do' block is also visible, containing a 'call ReconocimientoDeVoz1' block. At the bottom of the Viewer, there are warning icons (0 and 1) and a 'Show Warnings' button.

This screenshot shows the same MIT App Inventor interface for 'app_charlatana', but with a more complex block structure in the Viewer. The 'Blocks' palette is visible on the left, showing categories like Built-in, Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under 'Screen1', components listed include ET_Repetir, BT_Escucha, ET_Texto, CT_Texto_Usuario, BT_Lee, TextoAVoz1, ReconocimientoDeVoz, and Acelerómetro1. The 'Viewer' section shows a sequence of blocks: 'BT_Escucha Click', an equals sign block, 'ReconocimientoDeVoz1 GetText', an addition block, 'TextoAVoz1 Speak', a multiplication block, a 'Click' block, 'TextoAVoz1 Speak', a division block, 'Acelerómetro1 Shaking', a 'bitwise and' block, and 'random integer from 1 to 100'. The 'random integer from' block has '1' in the first input field and '100' in the second. At the bottom of the Viewer, there are warning icons (0 and 1) and a 'Show Warnings' button.

Métodos de conexión do móbil coa nosa app



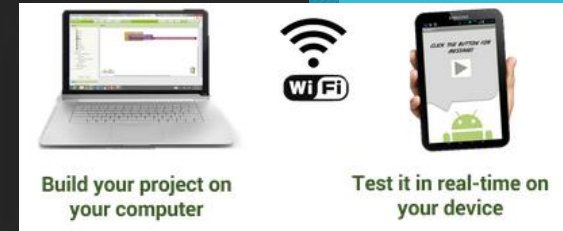
- Método 1: conexión por WiFi (opción recomendada)
- Método 2: conexión por cable USB
- Método 3: proba a través de emulador (opción non recomendable)



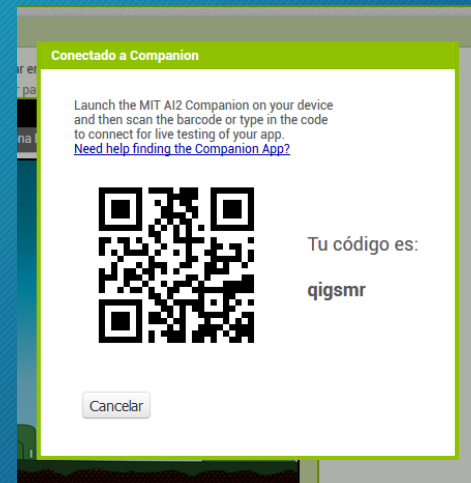
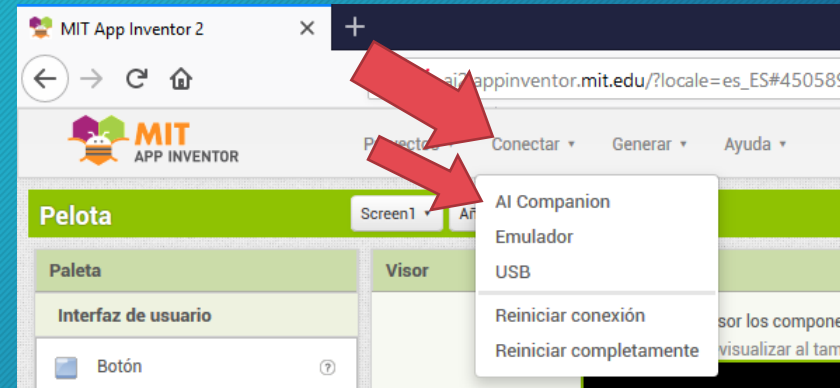
Instruccions:

<http://appinventor.mit.edu/explore/ai2/setup>

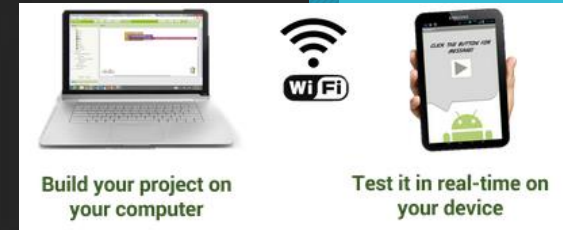
Traspaso da APP ao móbil para probas (vía WiFi)



- Paso 1: Descarga e instala a App **MIT AI2 Companion** no teu teléfono ou tableta mediante o Play Store.
- Paso 2: O móbil ou tableta e o ordenador deben estar conectados na mesma red local (mesmo rango de IPs).
- Paso 3: No teu proxecto, selecciona “Conectar” e “AI Companion”.
- Paso 4: Abre AI Companion no móbil e escanea o código QR

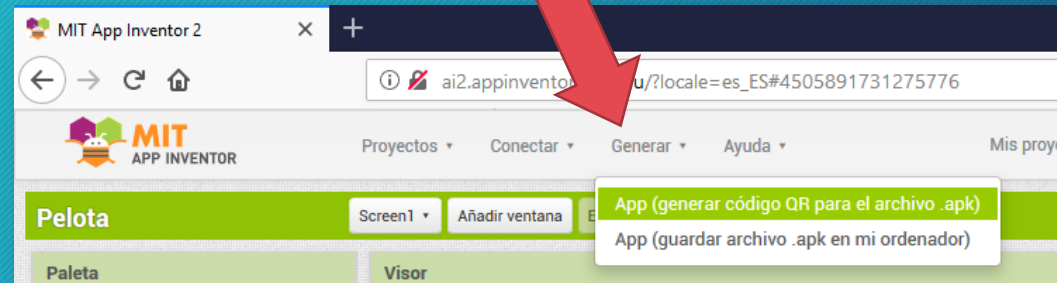


Instalación da APP no noso móbil ou tableta (vía WiFi)

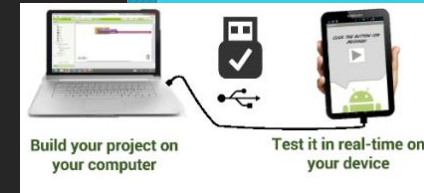


- Paso 1: Opción “Generar” -> APP (código QR)
- Paso 2: APP Inventor 2 preparará e compilará a aplicación
- Paso 3: Escanear o código QR dende o móbil

* O móbil ten que ter habilitadas as opción de desenvolvedor para poder instalar



Traspaso da APP ao móvil para probas (vía cable USB)



- Paso 1: Instalación no ordenador de App Inventor 2 Setup 2 Starter (Windows, Linux, Mac)

Archivo de instalación para Windows:

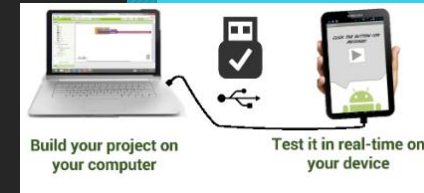
http://appinv.us/aisetup_windows

- Paso 2: Instalar MIT AI2 Companion App no noso móbil dende o Play Store.
- Paso 3: Lanzar aiStarter no noso ordenador.



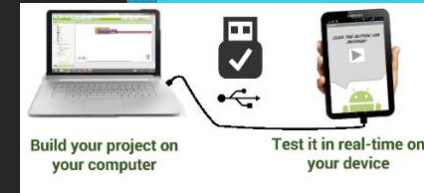
```
aiStarter
Platform = Windows
Appinventor tools located here: "C:\Program Files (x86)"
Bottle server starting up (using WSGLRefServer())...
Listening on http://127.0.0.1:8084/
Hit Ctrl-C to quit.
```

Traspaso da APP ao móvil para probas (vía cable USB)



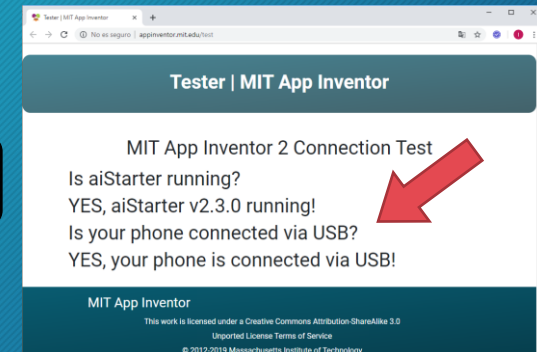
- Paso 4: Habilitar no noso dispositivo Android as opción como desenvolvedor
 - En Android 4.2 ou superiores:
Axustes -> Información do teléfono ->
(Sobre o número de compilación pulsamos 7 veces para que nos habilite as opcións de desenvolvedor)
- Paso 5: Habilitar depuración USB (USB Debugging ON)
 - En Android 4.2 ou superiores:
Axustes -> Sistema -> Avanzado -> Opcións para desenvolvedores -> Depuración por USB (ACTIVO)

Traspaso da APP ao móbil para probas (vía cable USB)

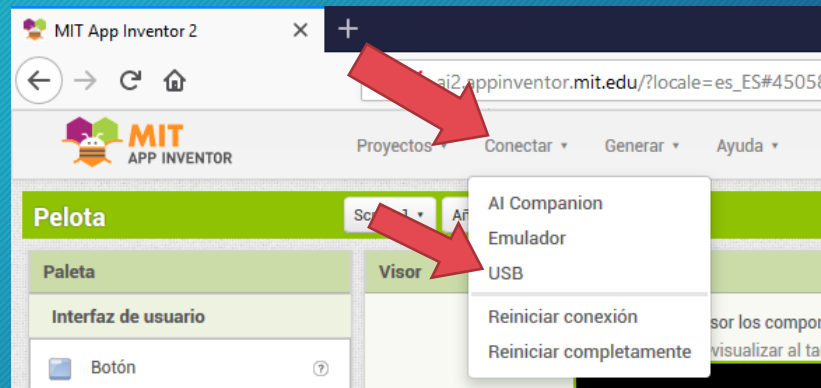


- Paso 5: Conectar o móbil e comprobar o test de conexión que todo é correcto

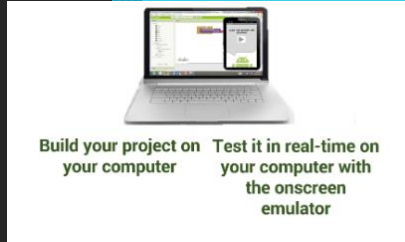
Test de conexión: <http://appinventor.mit.edu/test>



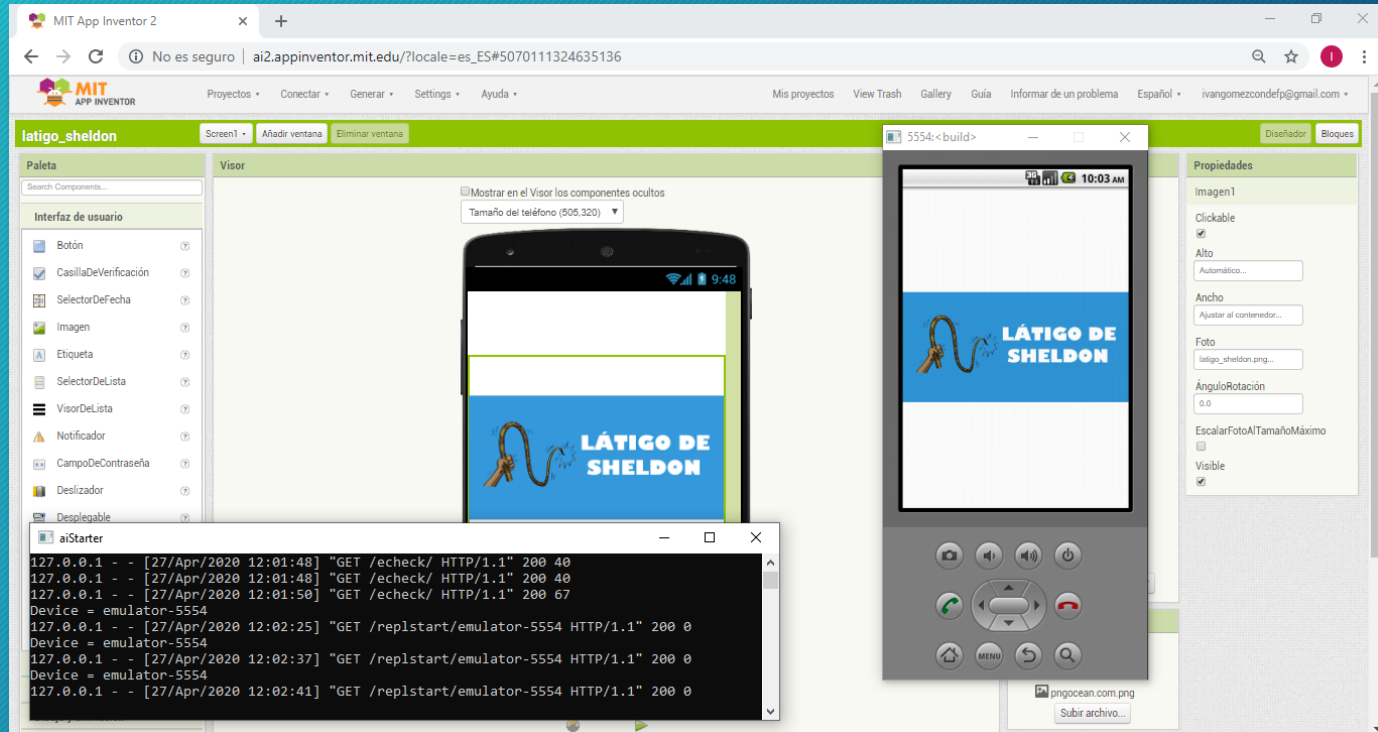
- Paso 6: Conectar coa nosa aplicación e probala no noso móbil.



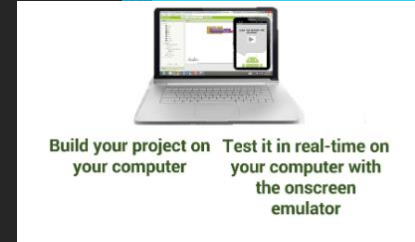
Probar a App no emulador



- Descargar e instalar aiStarter:
<https://appinventor.mit.edu/explore/ai2/setup-emulator.html>
- Ejecutar aiStarter
- Lanzar el proyecto desde la opción de emulador

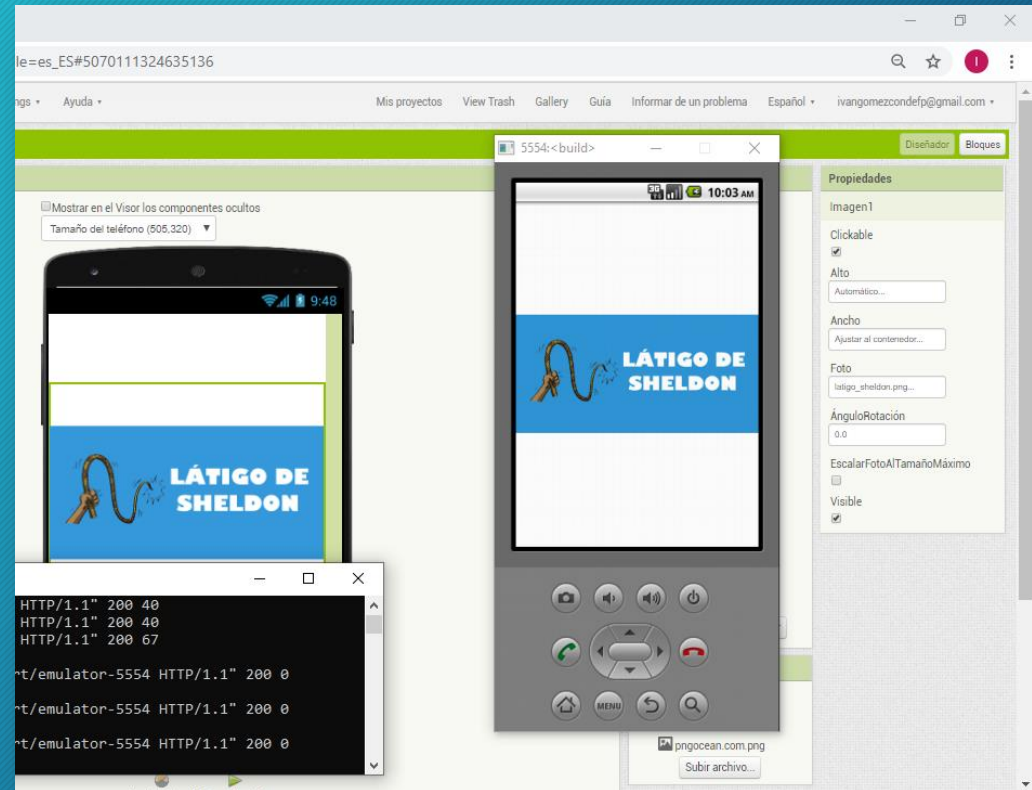


Probar a App no emulador



Problemas asociados ao uso do emulador:

- Non podemos probar sensores (acelerómetro, gps...)
- Lentitude na proba de apps.
- Imposibilidade de conectar con modelos de ML4K
- Resolucións de pantalla non reais



Aplicacións a crear no curso



- No curso elaboraremos unha serie de apps que nos permitirán desenvolver o noso aprendizaxe coa ferramenta de “App Inventor”:
 - App Látego de Sheldon
 - App Charlatana
 - App Debuxo
 - App Localización por GPS
 - App Aplicación multiventá
 - App Xogo co acelerómetro
 - App Autoavaliación dun exame con almacenamento en base de datos externa
 - App IA de recoñecemento de texto para adiviñar o destino de vacacións mais adecuado co uso de ML4K
 - APP IA de recoñecemento de cores a través de imaxes co uso de ML4K

“App Látego de Sheldon”

App
Látego de
Sheldon

- Trátase dunha app moi sinxela que servirá como toma de contacto con App Inventor.
- Esta app simplemente reproducirá o son dun látego cando se axite o móbil, baseada na famosa serie de “Big Bang Theory”



<https://www.youtube.com/watch?v=NCPiwrwZ-sg>

Compoñentes necesarios

App
Látego de
Sheldon

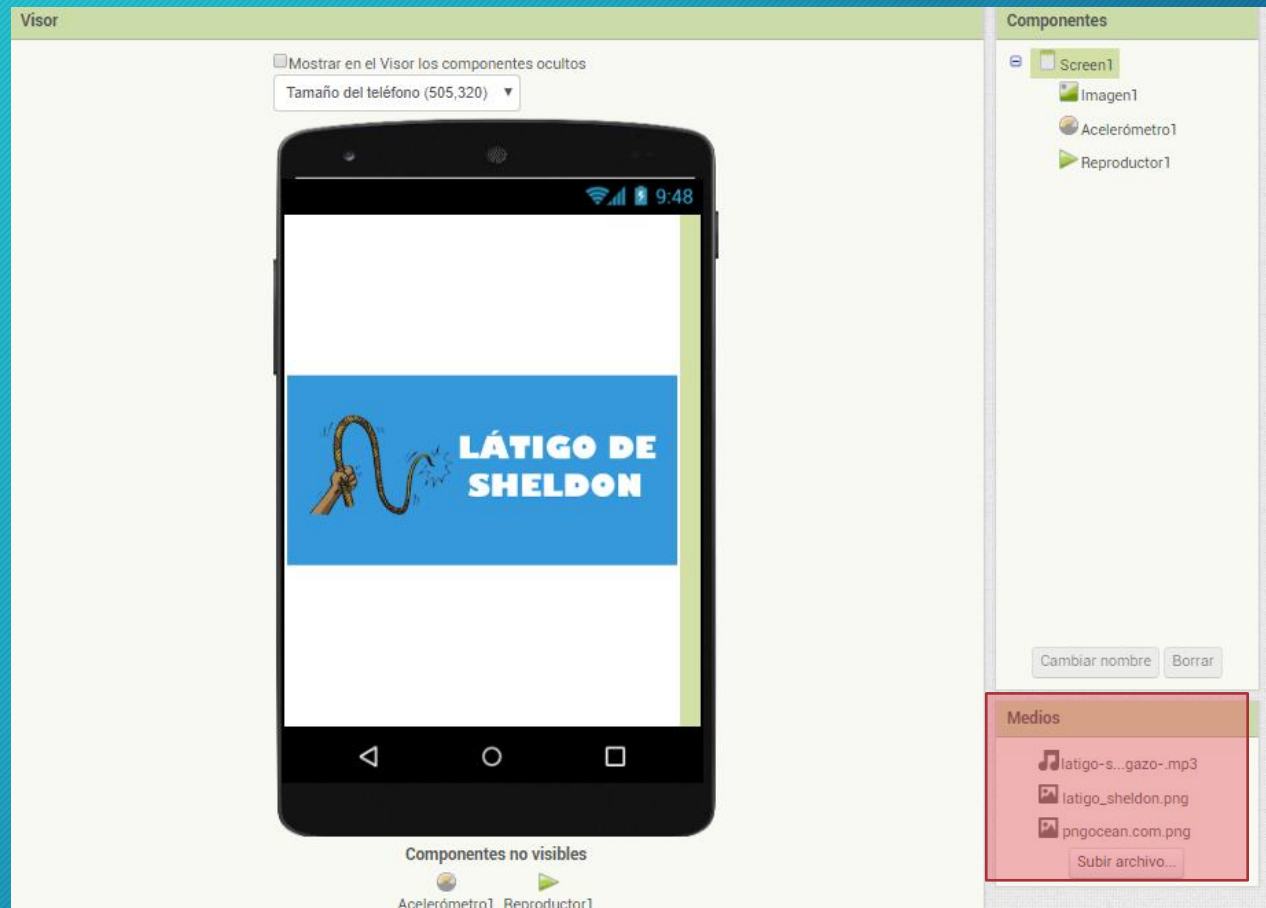
- Precisaremos soamente tres compoñentes na nosa interface:
 - Imaxe (para o fondo da nosa aplicación)
 - Acelerómetro (para detectar o movemento do noso móbil)
 - Reprodutor de son (para reproducir o son do látego)



Deseño da interface

App
Látego de
Sheldon

- Engadiremos os tres arquivos que precisamos en “Medios”: o son do látego en .mp3, a imaxe de fondo da nosa app e o icono para cando se instala no móbil



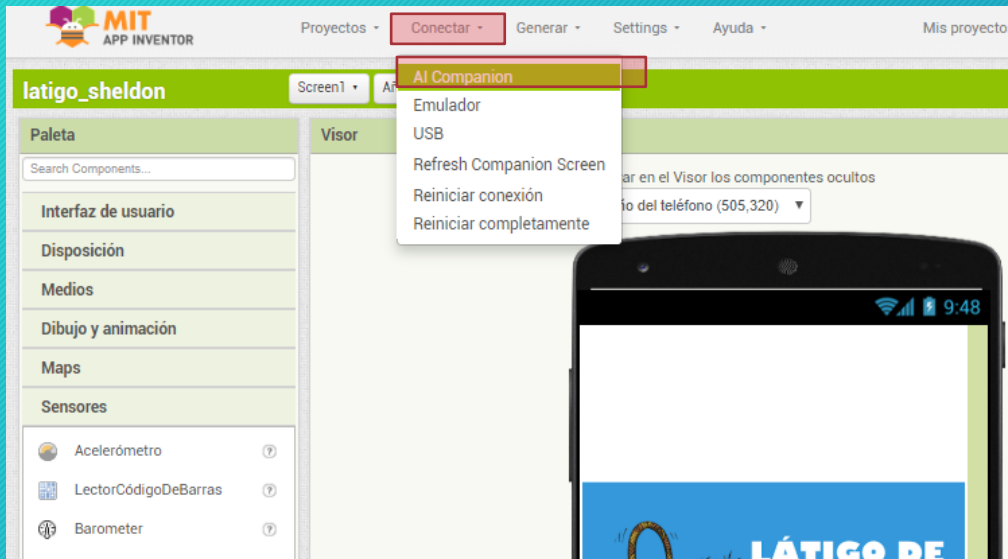
Programación

App
Látego de
Sheldon

The screenshot displays the MIT App Inventor web interface. At the top, the MIT App Inventor logo is on the left, and navigation links for 'Proyectos', 'Conectar', 'Generar', 'Settings', and 'Ayuda' are on the right. Below the navigation bar, the project name 'latigo_sheldon' is shown, along with buttons for 'Screen1', 'Añadir ventana', and 'Eliminar ventana'. The interface is divided into two main sections: 'Bloques' (Blocks) on the left and 'Visor' (Viewer) on the right. The 'Bloques' section contains a list of categories: Integrados (Control, Lógica, Matemáticas, Texto, Listas, Dictionaries, Colores, Variables, Procedimientos), Screen1 (Imagen1, Acelerómetro1, Reproductor1), and 'Cualquier componente'. The 'Visor' section shows a visual representation of the program logic: a yellow event block labeled 'cuando Acelerómetro1 .Agitar' is connected to a purple action block labeled 'ejecutar llamar Reproductor1 .Iniciar'.

Probar a nosa app “Látego de Sheldon”

App
Látego de
Sheldon



Conectado a Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
[Need help finding the Companion App?](#)



Tu código es:

phfkwp

Cancelar



“App charlatana”

App
charlatana

- Crearemos unha primeira aplicación que repita por voz o que eu lle falo ou o que eu lle escribo.
- Necesitaremos:
 1. Unha etiqueta para indicar que fale.
 2. Un botón para que repita por voz o que eu lle dixen.
 3. Outra etiqueta para indicar que escriba.
 4. Un campo de texto para que o usuario poida escribir algo.
 5. E un último botón para dicirlle a nosa aplicación que diga por voz o xa introducido no campo de texto.
 6. Ademais se movemos o noso móbil coa man tamén debe activar o recoñecemento da nosa voz.

Arrastramos os componentes

App
charlatana

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672`. The application title is `app_charlatana`. The interface is divided into several panels:

- Paleta (Palette):** Lists various UI components. Three components are highlighted with colored boxes and arrows pointing to the canvas:
 - Botón (Button):** Highlighted in a red box, with arrows pointing to the 'Escucha y repite' and 'Activa escucha' components on the canvas.
 - Etiqueta (Label):** Highlighted in a purple box, with arrows pointing to the 'Escucha y repite' and 'Escribe, yo leo :)' components on the canvas.
 - CampoDeTexto (Text Field):** Highlighted in a green box, with an arrow pointing to the 'Lee texto' component on the canvas.
- Visor (Viewer):** Shows a preview of the app interface. The visible text includes:
 - Escucha y repite
 - Habla, yo repito :)! (with a microphone icon)
 - Activa escucha
 - Escribe, yo leo :):
 - Lee texto
- Componentes (Components):** Lists the components currently on the canvas:
 - ET_Repetir
 - BT_Escucha
 - ET_Texto
 - CT_Texto_Usuario
 - BT_Lee
 - TextoAVoz1
 - ReconocimientoDeVoz1
 - Acelerómetro1
- Propiedades (Properties):** Shows the configuration for the selected component (Screen1), including settings for background color, text alignment, and animation.

Cambiamos os nomes dos elementos

App
charlatana

- Acostumamos a seguir boas prácticas de programación:
 - A cada etiqueta pónolle un nome representativo empezando por ET. Por exemplo: ET_RepetirVoz, ET_ReproducirTexto
 - A cada botón pónolle un nome representativo empezando por BT. Por exemplo: BT_Escucha, BT_Lee.
 - Ao noso campo de texto tamén lle cambiamos o nome: CT_TextoUsuario

Cambiamos nome e poñemos que ancho sexa o do contedor

App
charlatana

MIT App Inventor 2

ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672

MIT APP INVENTOR

Proyectos ▾ Conectar ▾ Generar ▾ Ayuda ▾

Mis proyectos Gallery Guía Informar de un problema Español ▾ ivangomezcondefp@gmail.com ▾

app_charlatana

Screen1 ▾ Añadir ventana Eliminar ventana

Diseñador Bloques

Paleta

Interfaz de usuario

- Botón
- CasillaDeVerificación
- SelectorDeFecha
- Imagen
- Etiqueta
- SelectorDeLista
- VisorDeLista
- Notificador
- CampoDeContraseña
- Deslizador
- Desplegable
- CampoDeTexto
- SelectorDeHora
- VisorWeb

Visor

Mostrar en el Visor los componentes ocultos

Marcar para previsualizar al tamaño de la tablet

Escucha y repite

Habla, yo repito :!

Activa escucha

Escribe, yo leo :)

Lee texto

Componentes

- Screen1
 - ET_Repetir
 - BT_Escucha
 - ET_Texto
 - CT_Texto_Usuario
 - BT_Lee
- TextoAVoz1
- ReconocimientoDeVoz1
- Acelerómetro1

Cambiar nombre Borrar

Propiedades

ET_Repetir

ColorDeFondo

Ninguno

Negrita

Cursiva

Tamaño de letra

16

TipoDeLetra

por defecto ▾

HTMLFormat

HasMargins

Alto

Automático...

Ancho

Ajustar al contenedor...

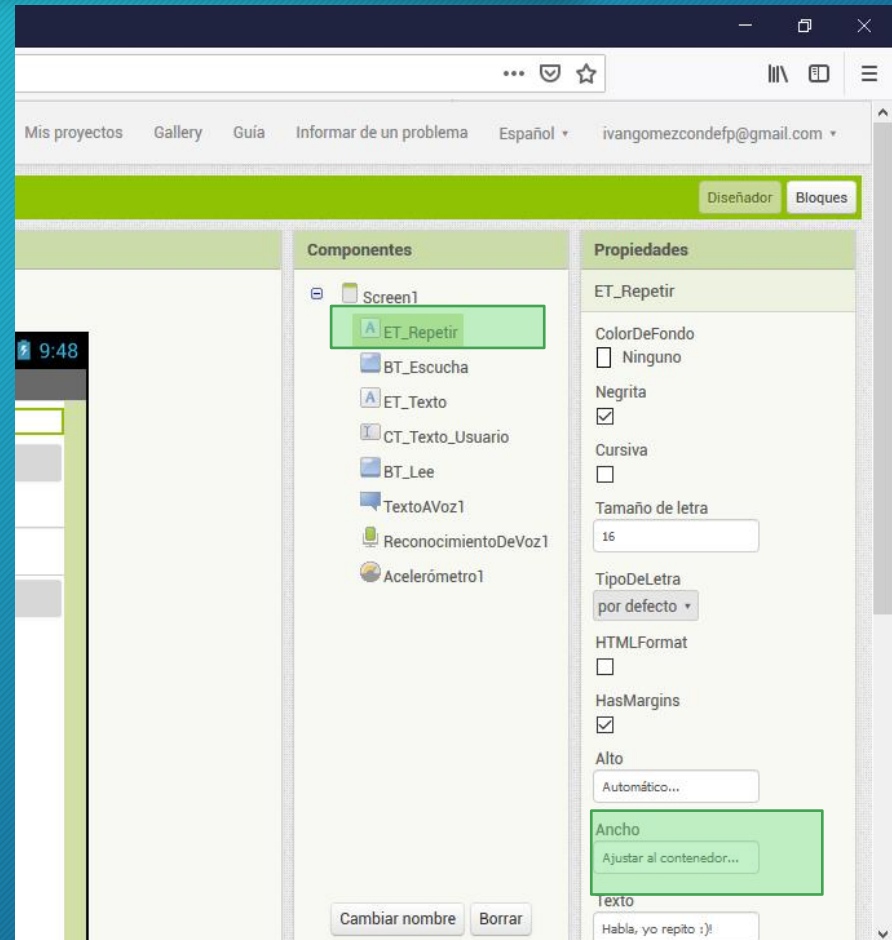
Texto

Habla, yo repito :!

Cambiamos propiedades dos elementos

App
charlatana

- Para facer a nosa aplicación algo mais simétrica podemos facer que os elementos ocupen todo o ancho do noso contedor. Para iso indicamos en propiedades de cada elemento (despois de facer clic nese elemento):
 - Posición do texto: centro (1)
 - Ancho: axustar ao contedor.



Engadimos elementos non visibles (pero necesarios)

App
charlatana

- Texto a voz, Recoñecemento de voz (no apartado medios) e Acelerómetro (no apartado sensores)

The screenshot displays the MIT App Inventor 2 web interface. On the left, the 'Medios' (Media) category is highlighted in red, containing 'ReconocimientoDeVoz' and 'TextoAVoz'. A red arrow points from these items to the 'Componentes no visibles' (Hidden components) section at the bottom of the design canvas, which lists 'TextoAVoz1', 'ReconocimientoDeVoz1', and 'Acelerómetro1'. The central design canvas shows a mobile app interface with the following text and buttons: 'Escucha y repite', 'Habla, yo repito :)', 'Activa escucha', 'Escribe, yo leo :)', 'Lee texto', and 'Acelerómetro1'. On the right, the 'Propiedades' (Properties) panel for 'E1_Hepetir' is visible, with 'ReconocimientoDeVoz1' highlighted in red. The 'Medios' section at the bottom right of the design canvas also has a 'Subir archivo...' button.

Engadimos elementos non visibles (pero necesarios)

App
charlatana

- Texto a voz, Reconocemento de voz (no apartado medios) e Acelerómetro (no apartado sensores)

The screenshot displays the MIT App Inventor 2 web interface. On the left, the 'Sensores' (Sensors) category is highlighted in red, with the 'Acelerómetro' (Accelerometer) component selected. A red arrow points from this component to the 'Acelerómetro1' component in the 'Componentes no visibles' (Hidden components) section at the bottom of the design canvas. The design canvas shows a mobile app interface with the following visible elements: a title bar 'Escucha y repite', a text field containing 'Habla, yo repito :)', a button 'Activa escucha', another text field containing 'Escribe, yo leo :)', and a button 'Lee texto'. The right-hand side of the interface shows the 'Propiedades' (Properties) panel for the selected 'Acelerómetro1' component, with various settings like 'ColorDeFondo', 'Negrita', 'Cursiva', 'Tamaño de letra', etc. The 'Medios' (Media) section at the bottom right includes a 'Subir archivo...' button.

Programamos botón de escoita

App
charlatana

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows `ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672`. The page title is `app_charlatana`. The interface is divided into several sections:

- Header:** Includes navigation links like "Proyectos", "Conectar", "Generar", "Ayuda", "Mis proyectos", "Gallery", "Guía", "Informar de un problema", "Español", and the user email `ivangomezcondefp@gmail.com`.
- Toolbar:** Contains buttons for "Screen1", "Añadir ventana", "Eliminar ventana", "Diseñado", and "Bloques".
- Bloques (Left Panel):** A list of available blocks categorized by type (Lógica, Matemáticas, Texto, Listas, Colores, Variables, Procedimientos). The `BT_Escucha` block is highlighted in red.
- Visor (Right Panel):** Shows the visual representation of the app screen. A button labeled `BT_Escucha` is visible. The event-driven programming blocks are shown as follows:
 - When `BT_Escucha` is clicked, execute `ReconocimientoDeVoz1` to get text.
 - When `BT_Escucha` gains focus, execute `BT_Escucha`.
 - When `BT_Escucha` is clicked for a long time, execute `BT_Escucha`.
 - When `BT_Escucha` loses focus, execute `BT_Escucha`.
 - When `BT_Escucha` is pressed, execute `BT_Escucha`.
 - When `BT_Escucha` is released, execute `BT_Escucha`.
 - When `BT_Escucha` is released, execute `BT_Escucha` to change the background color.

Programamos botón de ler

App
charlatana

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL: `ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672`. The page title is "app_charlatana". The interface is divided into several sections:

- Header:** Includes navigation links like "Proyectos", "Conectar", "Generar", "Ayuda", "Mis proyectos", "Gallery", "Guía", "Informar de un problema", "Español", and a user email "ivangomezcondefp@gmail.com".
- Toolbar:** Contains "Screen1", "Añadir ventana", "Eliminar ventana", "Diseñador", and "Bloques".
- Bloques (Left Panel):** A list of components categorized by type: Lógica, Matemáticas, Texto, Listas, Colores, Variables, and Procedimientos. Under "Screen1", components like "ET_Repetir", "BT_Escucha", "ET_Texto", and "CT_Texto_Usuario" are listed. The "BT_Lee" component is highlighted with a red box.
- Visor (Right Panel):** Shows the visual representation of the app. A button labeled "BT_Lee" is visible. A red box highlights the "cuando BT_Lee .Clic" event block. The corresponding logic block is: "ejecutar" -> "llamar TextoAVoz1 .Hablar" -> "mensaje (CT_Texto_Usuario . Texto)".

Activamos reconocimiento de VOZ

App
charlatana

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672`. The page title is `app_charlatana`. The interface includes a top navigation bar with options like `Proyectos`, `Conectar`, `Generar`, and `Ayuda`. Below this, there are buttons for `Diseñador` and `Bloques`. The `Bloques` panel on the left lists various categories: `Lógica`, `Matemáticas`, `Texto`, `Listas`, `Colores`, `Variables`, and `Procedimientos`. Under `Procedimientos`, the `ReconocimientoDeVoz` block is highlighted with a red box. The `Visor` panel on the right shows a visual programming workspace with several code blocks. A red box highlights a block: `cuando ReconocimientoDeVoz1 .DespuésDeObtenerTexto` with a `Resultado` variable and an `ejecutar` block. Another yellow box highlights a block: `cuando ReconocimientoDeVoz1 .AntesDeObtenerTexto` with an `ejecutar` block containing `llamar ReconocimientoDeVoz1 .ObtenerTexto`. A third yellow box highlights a block: `cuando ReconocimientoDeVoz1 .DespuésDeObtenerTexto` with a `Resultado` variable and an `ejecutar` block containing `llamar TextoAVoz1 .Hablar` with `mensaje` and `tomar Resultado`. The interface also shows a `Mostrar avisos` button at the bottom.

Activamos o acelerómetro

App
charlatana

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL: `ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672`. The page title is "app_charlatana".

Bloques (Left Panel):

- Categories: Lógica, Matemáticas, Texto, Listas, Colores, Variables, Procedimientos.
- Screen1 components: ET_Repetir, BT_Escucha, ET_Texto, CT_Texto_Usuario, BT_Lee, TextoAVoz1, **ReconocimientoDeVoz**, **Acelerómetro1**.
- Search: "Cualquier componente".
- Buttons: "Cambiar nombre", "Borrar".

Visor (Right Panel):

- Block 1: `cuando Acelerómetro1 . CambioEnAceleración` (when acceleration changes).
 - Inputs: `xAccel`, `yAccel`, `zAccel`.
 - Action: `ejecutar` (execute).
- Block 2: `cuando Acelerómetro1 . Agitar` (when shaken).
 - Action: `ejecutar` (execute).
- Block 3: `Acelerómetro1 . Disponible` (available).
- Block 4: `Acelerómetro1 . Habilitado` (enabled).
- Block 5: `poner Acelerómetro1 . Habilitado como` (set enabled).
- Block 6: `Acelerómetro1 . IntervaloMínimo` (minimum interval).
- Block 7: `poner Acelerómetro1 . IntervaloMínimo como` (set minimum interval).
- Block 8: `Acelerómetro1 . Sensibilidad` (sensitivity).
- Block 9: `poner Acelerómetro1 . Sensibilidad como` (set sensitivity).
- Block 10: `Acelerómetro1 . XAccel` (x-axis acceleration).

Probamos ou instalamos a nosa app no móbil ou tableta

App
charlatana

No teu proxecto, selecciona “Conectar” e “AI Companion”.

The screenshot shows the MIT App Inventor 2 web interface. The browser address bar displays `ai2.appinventor.mit.edu/?locale=es_ES#5791498017308672`. The navigation menu includes 'Proyectos', 'Conectar', 'Generar', and 'Ayuda'. The 'Conectar' button is highlighted with a red box. Below the navigation bar, the project name 'app_charlatana' is visible, along with buttons for 'Screen1', 'Añadir ventana', and 'Eliminar ventana'. The main workspace is divided into four panels: 'Paleta' (left), 'Visor' (center), 'Componentes' (right), and 'Propiedades' (right). The 'Paleta' panel lists various sensors like 'Acelerómetro', 'LectorCódigoDeBarras', 'Reloj', 'GyroscopeSensor', 'SensorDeUbicación', 'NearField', 'SensorDeOrientación', 'Pedometer', and 'ProximitySensor'. The 'Visor' panel shows a preview of the app interface with a dialog box titled 'Conectado a Companion'. The dialog box contains the following text: 'Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app. [Need help finding the Companion App?](#)'. Below the text is a QR code and the text 'Tu código es: yuvwzx'. A 'Cancelar' button is located at the bottom of the dialog box. The 'Componentes' panel shows a list of components for 'Screen1', including 'ET_Repetir', 'BT_Escucha', 'ET_Texto', 'CT_Texto_Usuario', 'BT_Lee', 'TextoAVoz1', 'ReconocimientoDeVoz1', and 'Acelerómetro1'. The 'Propiedades' panel shows the properties for the selected 'ET_Repetir' component, including 'ColorDeFondo', 'Negrita', 'Cursiva', 'Tamaño de letra', 'TipoDeLetra', 'HTMLFormat', 'HasMargins', 'Alo', 'Ancho', and 'Texto'.

“APP Debuxo”

App
debuxo

- Esta app consistirá en incorporar un lenzo na miña app para que o usuario poida debuxar o que queira en diferentes cores. Poderá limpar a pantalla ou debuxar sobre unha foto.
- Necesitaremos por tanto:
 1. Catro botóns para (negro, vermello, azul e verde)
 2. Un deslizador para cambiar o grosor do meu pincel.
 3. Un lenzo.
 4. Un botón de limpar, outro botón de seleccionar imaxe e outro de gardar.
 5. Ademais, nesta app empregaremos Marcos horizontais para agrupar elementos.

Selección dos elementos necesarios (botóns, deslizador)

App
debutxo

The screenshot displays the MIT App Inventor 2 web interface. The central 'Visor' (Preview) area shows a mobile app design with a header 'Dibujar' and four colored buttons: Negro, Rojo, Verde, and Azul. Below the buttons is a large white canvas area, and at the bottom are three buttons: Limpiar, Selector Imagen, and Guardar. The interface is divided into four main panels:

- Paleta (Palette):** Lists various UI components. 'Botón' and 'Deslizador' are highlighted with red boxes.
- Componentes (Components):** A tree view showing the app's structure. A red box highlights the 'BT_Azul' component under 'MarcoHorizontal1'.
- Propiedades (Properties):** Shows the configuration for the selected 'BT_Azul' component, including 'ColorDeFondo' (Azul), 'Habilitado' (checked), and 'Tamaño de letra' (14.0).
- Medios (Media):** Includes a 'Subir archivo...' button.

Selección de tres marcos horizontais (disposición horizontal)

App
debutxo

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL: `ai2.appinventor.mit.edu/?locale=es_ES#6558793857630208`. The interface is divided into several panels:

- Paleta (Left):** A sidebar containing various component categories. The "Disposición" (Layout) category is highlighted in red. Within this category, the "DisposiciónHorizontal" component is selected, also highlighted in red. Red arrows point from this selection to the visualizer.
- Visor (Center):** A central workspace showing a mobile app preview. The app has a title bar "Dibujar" and a status bar with the time "9:48". The main content area contains four buttons: "Negro" (black), "Rojo" (red), "Verde" (green), and "Azul" (blue). Below these buttons is a large white canvas with a drawing tool icon. At the bottom of the app are three buttons: "Limpiar", "Selector Imagen", and "Guardar".
- Componentes (Right):** A list of components added to the app. It includes "Screen1", "MarcoHorizontal" (with sub-components "BT_Negro", "BT_Rojo", "BT_Verde", "BT_Azul"), "MarcoHorizontal2" (with "Deslizador1"), "Lienzo1", and "MarcoHorizontal3" (with "BT_Limpiar", "BT_SelectorDelImagen", "BT_Guardar", and "ET_RutaArchivo").
- Propiedades (Far Right):** A panel showing the properties of the selected "BT_Azul" component. Properties include "ColorDeFondo" (set to "Azul"), "Habilitado" (checked), "Negrita", "Cursiva", "Tamaño de letra" (14.0), "TipoDeLetra" (por defecto), "Alto" (Automático...), "Ancho" (20 percent...), "Imagen" (Ninguno...), and "Forma" (redondeado).

Selección dos elementos necesarios (lenzo)

App
debutxo

The screenshot displays the MIT App Inventor 2 web interface. The central canvas shows a mobile app design with a drawing toolbar and a canvas area. The 'Componentes' panel on the right is highlighted with a red box, showing a tree view of components. The 'Propiedades' panel on the far right shows the properties for the selected 'BT_Azul' component. The 'Medios' panel at the bottom right has a 'Subir archivo...' button.

Paleta

- Interfaz de usuario
- Disposición
- Medios
- Dibujo y animación
- Pelota
- Lienzo
- Spritemagen
- Maps
- Sensores
- Social
- Almacenamiento
- Conectividad
- LEGO® MINDSTORMS®
- Experimental
- Extension

Visor

Mostrar en el Visor los componentes ocultos
 Marcar para previsualizar al tamaño de la tablet

Dibujar

Negro Rojo Verde Azul

Limpia Selector Imagen Guardar

Componentes

- Screen1
 - MarcoHorizontal
 - BT_Negro
 - BT_Rojo
 - BT_Verde
 - BT_Azul
 - MarcoHorizontal2
 - Deslizador1
 - Lienzo1
 - MarcoHorizontal3
 - BT_Limpiar
 - BT_SelectorDelImagen
 - BT_Guardar
 - ET_RutaArchivo

Propiedades

BT_Azul

ColorDeFondo
Azul

Habilitado

Negrita

Cursiva

Tamaño de letra
14.0

TipoDeLetra
por defecto

Alto
Automático...

Ancho
20 percent...

Imagen
Ninguno...

Forma
redondeado

MostrarPulsación

Texto
Azul

Medios

Subir archivo...

Selección dos elementos necesarios (selector de imaxe)

App
debuxo

The screenshot displays the MIT App Inventor 2 web interface. The central canvas shows a mobile app design with a header 'Dibujar' and four colored buttons: Negro, Rojo, Verde, and Azul. Below the buttons is a large white canvas labeled 'Lienzo1'. At the bottom of the app are three buttons: 'Limpiar', 'Selector Imagen', and 'Guardar'. The interface is divided into several panels:

- Paleta (Left):** Contains categories like 'Interfaz de usuario', 'Disposición', 'Medios', 'Reproductor', 'Sonido', 'GrabadorDeSonidos', 'ReconocimientoDeVoz', 'TextoAVoz', 'ReproductorDeVideo', and 'TraductorYandex'. The 'Medios' category is highlighted in red, and 'SelectorDelImagen' is selected within it.
- Componentes (Middle-Right):** A tree view showing the app's structure. Under 'Screen1', there are two 'MarcoHorizontal' containers. The first contains 'BT_Negro', 'BT_Rojo', 'BT_Verde', and 'BT_Azul'. The second contains 'Deslizador1', 'Lienzo1', and another 'MarcoHorizontal3' containing 'BT_Limpiar', 'BT_SelectorDelImagen', and 'BT_Guardar'. The 'BT_Azul' component is highlighted in yellow.
- Propiedades (Right):** Shows the properties for the selected 'BT_Azul' component, including 'ColorDeFondo' (Azul), 'Habilitado' (checked), 'Negrita', 'Cursiva', 'Tamaño de letra' (14.0), 'TipoDeLetra' (por defecto), 'Alto' (Automático...), 'Ancho' (20 percent...), 'Imagen' (Ninguno...), 'Forma' (redondeado), and 'MostrarPulsación' (checked).

Programación dos botóns de color (negro, vermello, verde e azul)

App
debuxo

The image shows a visual programming environment with a project tree on the left and a code editor on the right. The project tree includes:

- Screen1
 - MarcoHorizontal
 - BT_Negro
 - BT_Rojo
 - BT_Verde
 - BT_Azul
 - MarcoHorizontal2
 - Deslizador1
 - Lienzo1
 - MarcoHorizontal3
 - BT_Limpiar

The code editor displays four event-driven blocks:

- cuando BT_Negro .Clic
ejecutar poner Lienzo1 . ColorDePintura como [negro]
- cuando BT_Rojo .Clic
ejecutar poner Lienzo1 . ColorDePintura como [rojo]
- cuando BT_Verde .Clic
ejecutar poner Lienzo1 . ColorDePintura como [verde]
- cuando BT_Azul .Clic
ejecutar poner Lienzo1 . ColorDePintura como [azul]

Programación do debuxado

App
debuxo

The image shows a programming environment interface. On the left is a project browser with a tree structure:

- Screen1
 - MarcoHorizontal
 - BT_Negro
 - BT_Rojo
 - BT_Verde
 - BT_Azul
 - MarcoHorizontal2
 - Deslizador1
 - Lienzo1

The main workspace displays a script for the 'Lienzo1' object:

```
cuando Lienzo1 .Arrastrado
  XInicial YInicial XPrevio YPrevio XActual YActual draggedAnySprite
ejecutar
  llamar Lienzo1 .DibujarLinea
    x1 tomar XPrevio
    y1 tomar YPrevio
    x2 tomar XActual
    y2 tomar YActual
```

Programación de limpiar e guardar

App
debuxo

The screenshot displays a visual programming environment with two main panels: 'Bloques' (Blocks) on the left and 'Visor' (Viewer) on the right.

Bloques Panel: Lists various components and methods. Under 'DisposiciónHorizontal', there are buttons for 'BT_Negro', 'BT_Vermello', 'BT_Azul', and 'BT_Verde'. Below that is a 'Deslizador1' (slider) and a 'Lienzo1' (canvas). Another 'DisposiciónHorizontal' section contains 'BT_Limpar' (clean), 'SI_Selector' (if-selector), and 'BT_Gardar' (save). At the bottom is 'ET_Ruta' (path).

Visor Panel: Shows two event-driven code blocks:

- Top Block:** Triggered by 'cuando BT_Limpar .Clic' (when BT_Limpar is clicked). The execution sequence is: 'ejecutar llamar Lienzo1 .Limpiar' (execute call Lienzo1.Limpiar) followed by 'poner Lienzo1 .ImagenDeFondo como' (set Lienzo1.ImagenDeFondo as) with a small white square icon.
- Bottom Block:** Triggered by 'cuando BT_Gardar .Clic' (when BT_Gardar is clicked). The execution sequence is: 'ejecutar poner ET_Ruta .Texto como' (execute set ET_Ruta.Texto as) followed by 'llamar Lienzo1 .Guardar' (call Lienzo1.Guardar).

At the bottom of the Visor panel, there are warning and error indicators (a yellow triangle with '0' and a red circle with '0') and a button labeled 'Mostrar avisos' (Show warnings).

Programación de selector de imaxe e do deslizador

App
debutxo

The screenshot displays the Android Studio IDE with the following components:

- XML Layout (Left Panel):**
 - Screen1
 - MarcoHorizontal
 - BT_Negro
 - BT_Rojo
 - BT_Verde
 - BT_Azul
 - MarcoHorizontal2
 - Deslizador1
 - Lienzo1
 - MarcoHorizontal3
 - BT_Limpiar
 - BT_SelectorDelmag
 - BT_Guardar
- Kotlin Code (Main Area):**

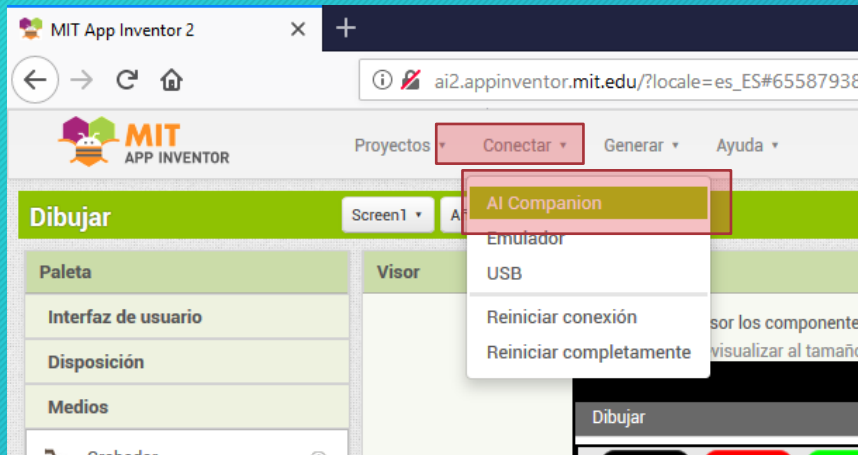
```
cuando Deslizador1 .PosiciónCambiada
    posiciónDelPulgar
    ejecutar poner Lienzo1 . AnchoDeLínea como tomar posiciónDelPulgar

cuando BT_SelectorDelimagen .DespuésDeSelección
    ejecutar poner Lienzo1 . ImagenDeFondo como BT_SelectorDelimagen . Selección
```

Probamos ou instalamos a nosa app no móbil ou tableta

App
debuxo

No teu proxecto, selecciona “Conectar” e “AI Companion”.



“APP Localización por GPS”

Localización
por GPS









- Esta app permitirá obter a lonxitude e latitude na que nos encontramos, grazas ao GPS e automaticamente mostraranos a situación nun mapa mediante Leaflet.
- Precisaremos:
 - Un marco horizontal con catro etiquetas.
 - Sensor de ubicación.
 - Mapa “Map”.
 - Localizador “Marker”.



Sensor de ubicación

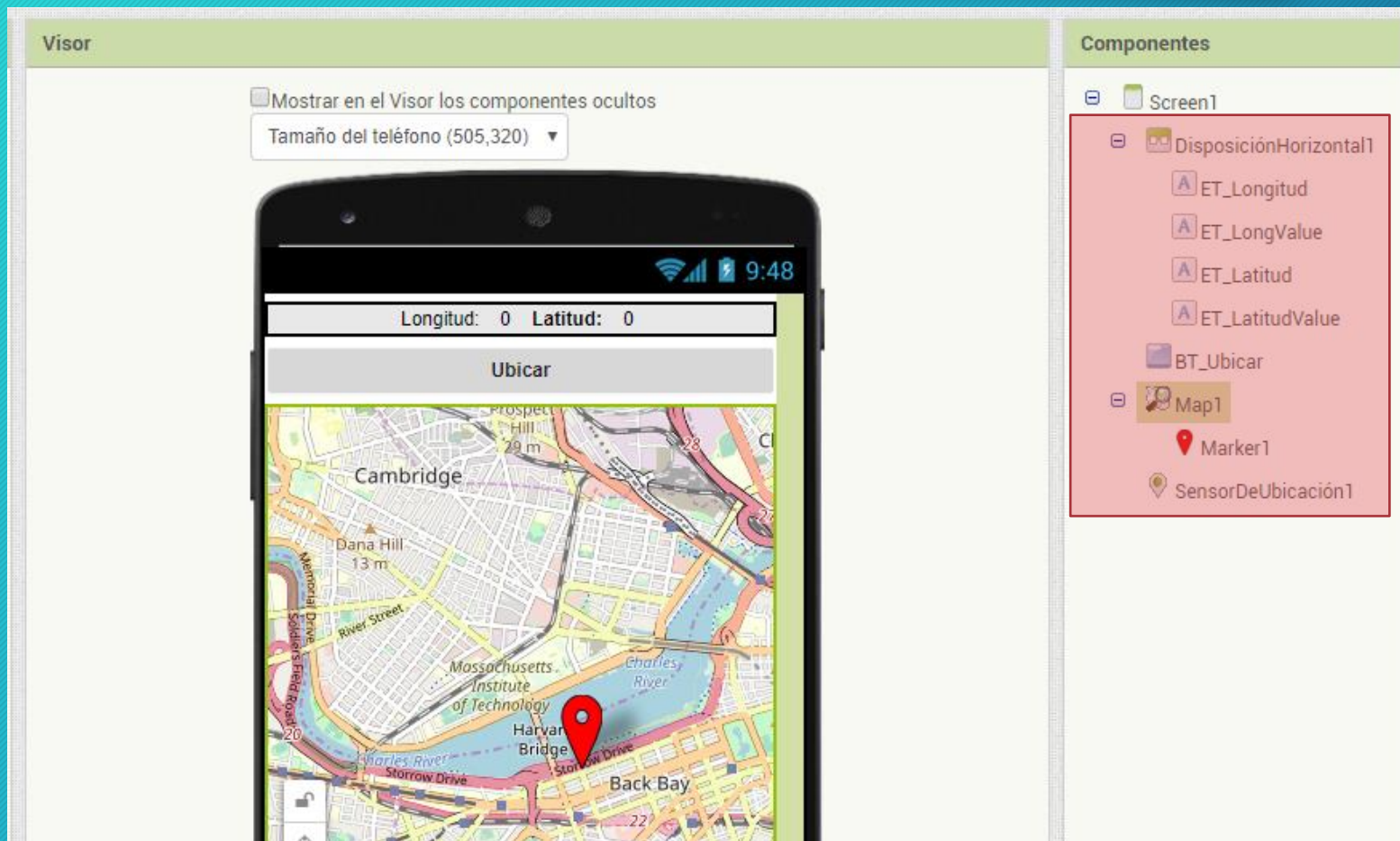
Localización
por GPS

- Información oficial del sensor:
 - El componente tiene que tener el valor Verdadero en la propiedad Habilitado, y debe ser viable determinar la ubicación, ya sea activando la conectividad a través de redes Wifi, o habilitando en el dispositivo la función GPS (si se está en el exterior).
 - Es posible que la información sobre la ubicación no esté disponible en el momento de iniciar la aplicación. Normalmente hay que esperar un poco para que el proveedor de la localización la encuentre y la utilice, o bien utilizar el evento `CambioEnUbicación`

Sensores		
	Acelerómetro	?
	LectorCódigoDeBarras	?
	Barometer	?
	Reloj	?
	GyroscopeSensor	?
	Hygrometer	?
	LightSensor	?
	SensorDeUbicación	?

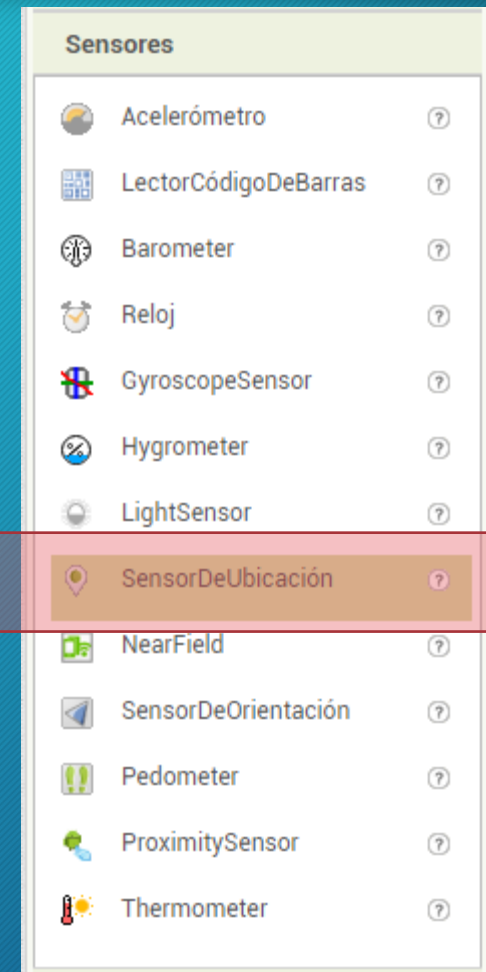
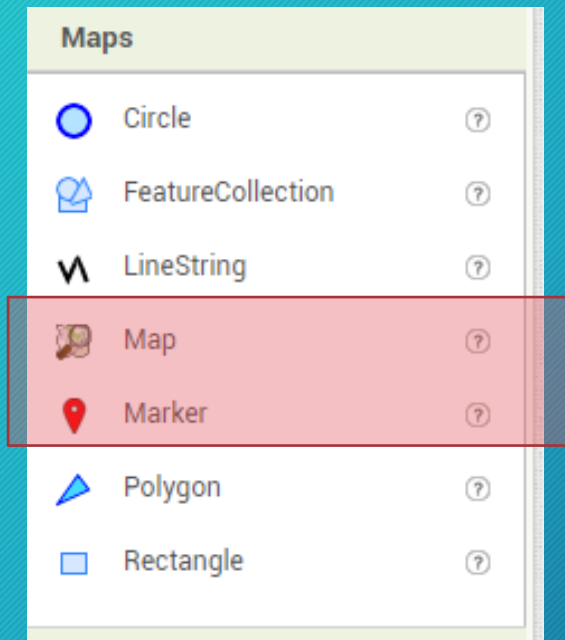
Diseño da interface

Localización por GPS



Incorporación dos elementos non visibles pero necesarios

Localización por GPS



Programación do botón de ubicar

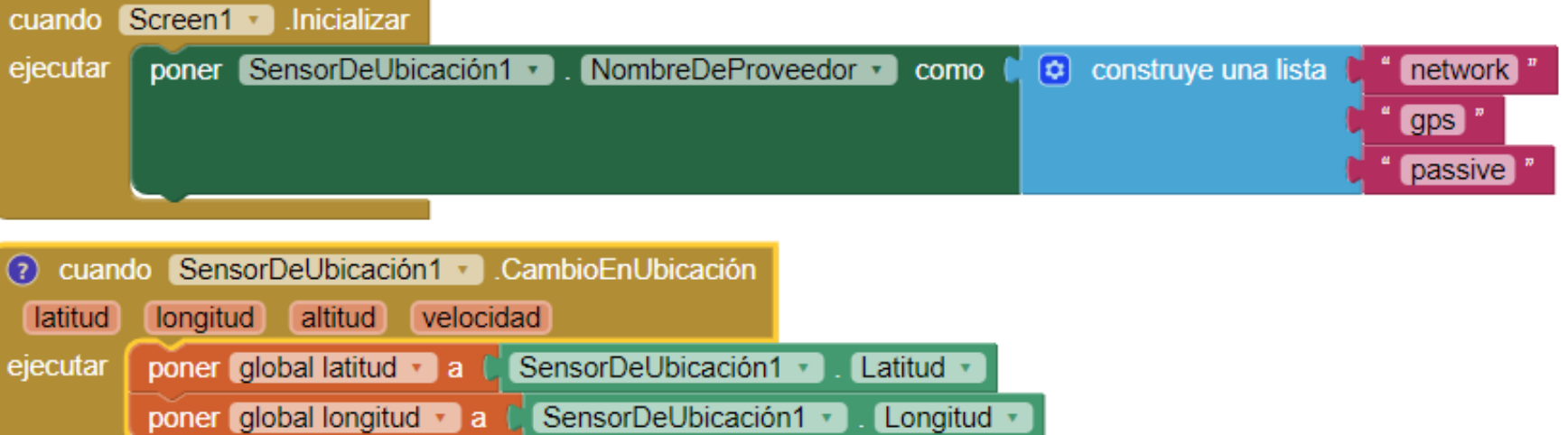
Localización por GPS

```
inicializar global latitud como 0
inicializar global longitud como 0

cuando BT_Ubicar .Clic
ejecutar
  poner ET_LongValue . Texto como tomar global longitud
  poner ET_LatitudValue . Texto como tomar global latitud
  llamar Map1 .PanTo
    latitud tomar global latitud
    longitud tomar global longitud
    zoom 13
  llamar Marker1 .SetLocation
    latitud tomar global latitud
    longitud tomar global longitud
```

Programación do proveedor do sensor de ubicación e cambio de ubicación

Localización por GPS



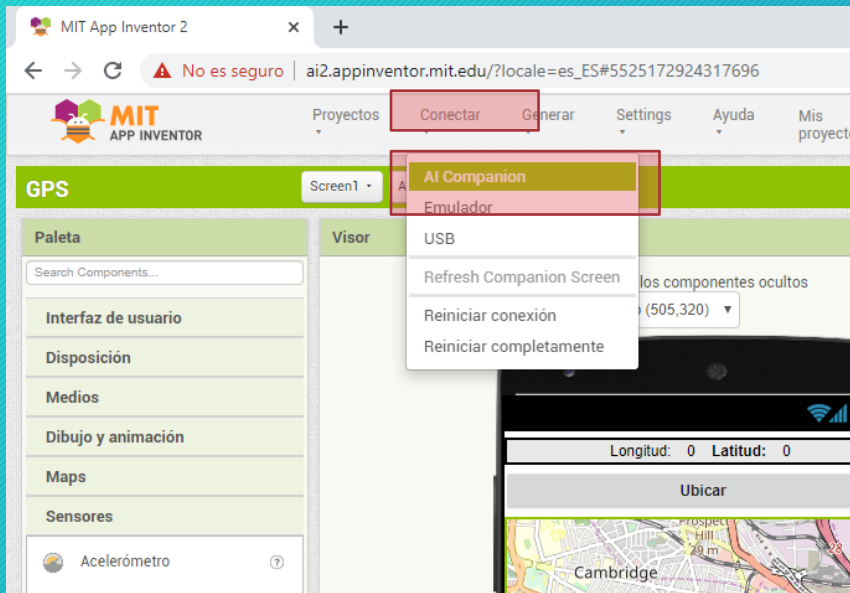
The image shows two Scratch code blocks for programming a location sensor. The first block is a 'when screen initializes' block. It contains an 'execute' block with the text 'poner SensorDeUbicación1 . NombreDeProveedor como' followed by a 'construye una lista' block. The 'construye una lista' block has three items: 'network', 'gps', and 'passive'. The second block is a 'when location changes' block. It has input fields for 'latitud', 'longitud', 'altitud', and 'velocidad'. It contains two 'execute' blocks: 'poner global latitud a SensorDeUbicación1 . Latitud' and 'poner global longitud a SensorDeUbicación1 . Longitud'.

```
cuando Screen1 .Inicializar
ejecutar
  poner SensorDeUbicación1 . NombreDeProveedor como
    construye una lista
      " network "
      " gps "
      " passive "

cuando SensorDeUbicación1 .CambioEnUbicación
  latitud
  longitud
  altitud
  velocidad
ejecutar
  poner global latitud a SensorDeUbicación1 . Latitud
  poner global longitud a SensorDeUbicación1 . Longitud
```

Probar a nosa app “Localización por GPS”

Localización por GPS



Conectado a Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
[Need help finding the Companion App?](#)



Tu código es:
phfkwp

Cancelar

“APP Multiventá”

App multi
ventá

- Esta app permitirá familiarizarnos con varias ventás para movernos entre elas. Algo común nunha app na que nos movemos entre ventás cando pulsamos en diferentes botóns.
- Precisaremos:
 - Un marco horizontal
 - Tres botóns no marco horizontal para moverse entre as tres ventás (ventá 1, ventá 2 e ventá 3)
 - Dúas novas ventás a maiores da que xa nos crea por defecto (Screen1).

Incorporar elementos: tres ventás, tres botóns e marco horizontal.

App multi
ventá

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=es_ES#5311287488282624`. The top navigation bar includes links for 'Proyectos', 'Conectar', 'Generar', 'Ayuda', 'Mis proyectos', 'Gallery', 'Guía', 'Informar de un problema', 'Español', and the user email 'ivangomezcondefp@gmail.com'. The main workspace is titled 'Multiventana' and features a 'Diseñador' (Designer) tab. On the left, the 'Paleta' (Palette) shows various UI components under the 'Interfaz de usuario' (User Interface) category, including Botón, CasillaDeVerificación, SelectorDeFecha, Imagen, Etiqueta, SelectorDeLista, VisorDeLista, Notificador, CampoDeContraseña, Deslizador, Desplegable, CampoDeTexto, SelectorDeHora, and VisorWeb. A red box highlights the 'Añadir ventana' (Add window) button in the top bar, which has opened a dropdown menu with options for 'Screen1', 'Ventana2', and 'Ventana3'. The central canvas shows a preview of 'Screen1' with a dark header bar containing three buttons labeled 'Ventana1', 'Ventana2', and 'Ventana3'. Below the canvas, there are checkboxes for 'Mostrar en el Visor los componentes ocultos' and 'Marcar para previsualizar al tamaño de la tablet'. On the right, the 'Componentes' (Components) panel shows a tree view with 'Screen1' containing a 'DisposiciónHorizontal1' (Horizontal Disposition) component, which includes three sub-components: 'BT_Ventana1', 'BT_Ventana2', and 'BT_Ventana3'. The 'Propiedades' (Properties) panel on the far right shows settings for 'Screen1', such as 'PantallaAcercaDe', 'AccentColor', 'DispHorizontal', 'DispVertical', 'AppName', 'ColorDeFondo', 'ImagenDeFondo', 'AnimaciónCierreDePantalla', 'Icono', and 'AnimaciónAlAbrirPantalla'. At the bottom of the components panel, there are 'Cambiar nombre' (Change name) and 'Borrar' (Delete) buttons.

Programación (bloque control, “abrir pantalla”)

App multi
ventá

The screenshot displays the MIT App Inventor 2 web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/?locale=es_ES#5311287488282624`. The interface is in Spanish and shows a project titled "Multiventana".

Left Panel (Bloques): A sidebar containing a "Bloques" section with categories: Integrados (Control, Lógica, Matemáticas, Texto, Listas, Colores, Variables, Procedimientos), Screen1, and Cualquier componente. Under "Screen1", there is a "DisposiciónHorizontal1" block containing three button components: "BT_Ventana1", "BT_Ventana2", and "BT_Ventana3".

Right Panel (Visor): A workspace for programming logic. It contains several event-driven blocks for the buttons:

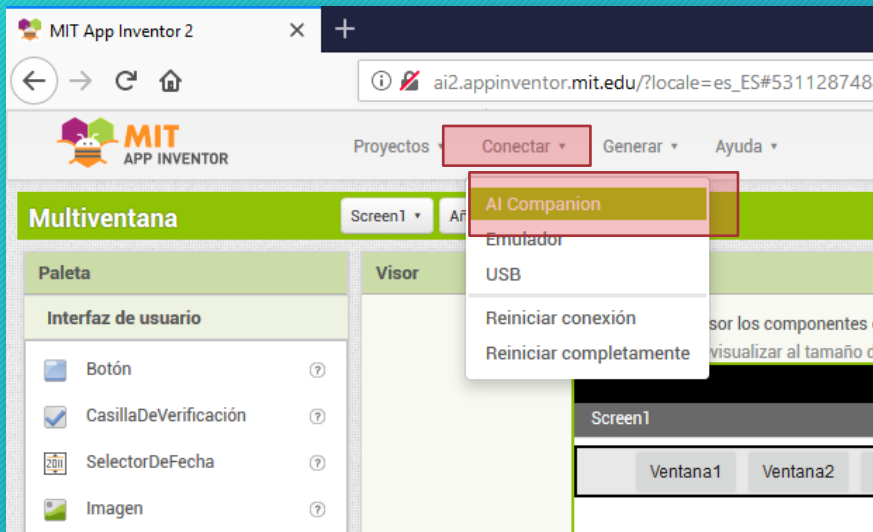
- BT_Ventana1 .Clic:** ejecutar [bloque vacío]
- BT_Ventana1 .ObtenerFoco:** ejecutar [bloque vacío]
- BT_Ventana1 .ClicLargo:** ejecutar [bloque vacío]
- BT_Ventana1 .PerderFoco:** ejecutar [bloque vacío]
- BT_Ventana1 .Presionar:** ejecutar [bloque vacío]
- BT_Ventana1 .Soltar:** ejecutar [avisos]
- BT_Ventana1 .Clic:** ejecutar abrir otra pantalla Nombre de la pantalla "Screen1"
- BT_Ventana2 .Clic:** ejecutar abrir otra pantalla Nombre de la pantalla "Ventana2"
- BT_Ventana3 .Clic:** ejecutar abrir otra pantalla Nombre de la pantalla "Ventana3"

At the bottom left, there are buttons for "Cambiar nombre" and "Borrar". On the right side of the workspace, there are icons for zooming in (+), zooming out (-), and a trash can.

Introduce algo en cada ventá e proba a túa app

App multi
ventá

No teu proxecto, selecciona “Conectar” e “AI Companion”.



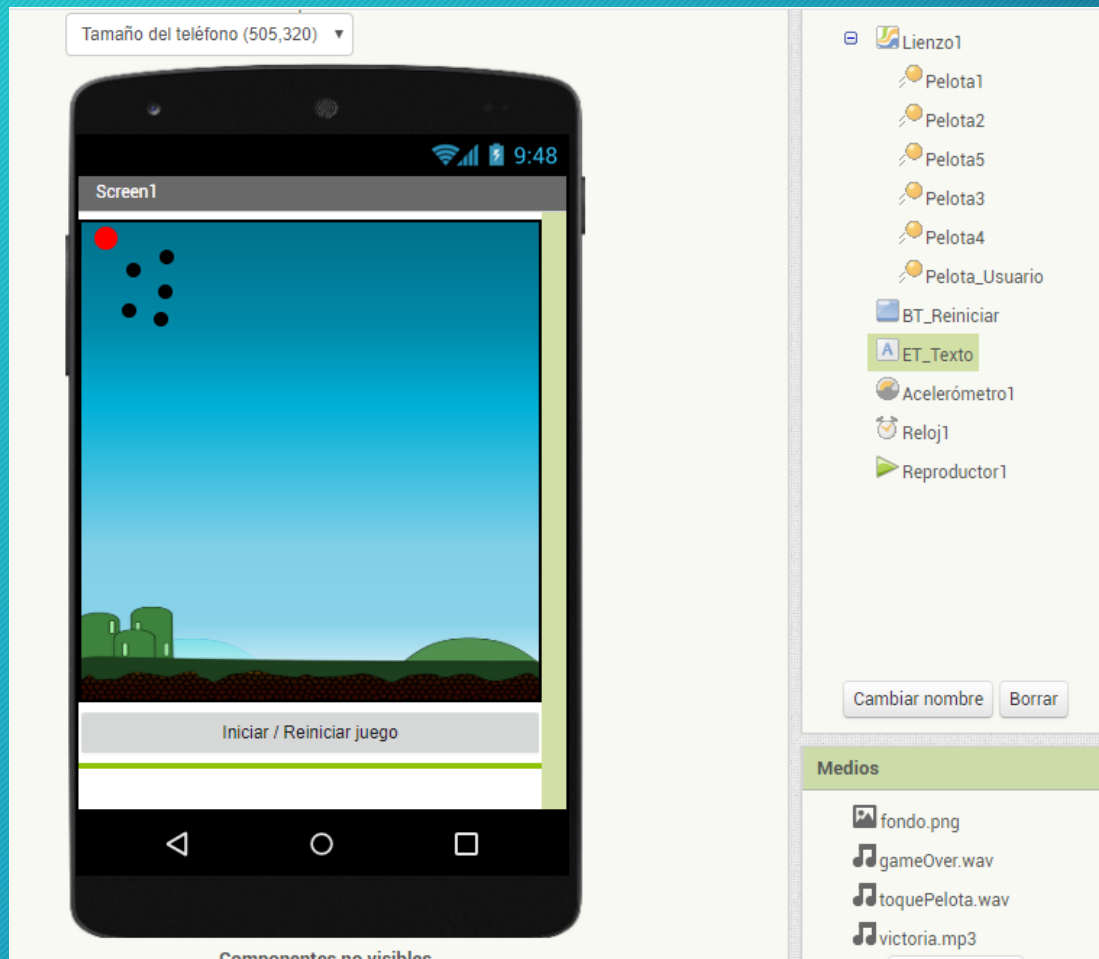
“APP Xogo co acelerómetro”

App xogo

- Imos crear o noso primeiro xogo que consiste en dirixir unha pelota para que toque certos obxectivos en función de como movamos coa man o noso teléfono.
- Precisaremos:
 - Un lenzo (cunha imaxe de fondo que tedes na aula virtual).
 - Un botón para reiniciar o xogo.
 - Un temporizador.
 - Un acelerómetro.
 - Tres audios (victoria, perda no xogo e toque da nosa pelota cun obxectivo).
 - Imaxe de fondo

Xogo co acelerómetro

App xogo



Variables no xogo

App xogo

- Una variable es un espacio en memoria que permite almacenar un valor. Para hacer referencia a dicha variable se utilizan un nombre simbólico también llamado identificador. En nuestro caso, los identificadores son: “enJuego”, “tiempo”, “objetivos”

inicializar global enJuego como falso

inicializar global tiempo como 15

inicializar global objetivos como 5

Inicialización da pantalla e ocultar as pelotas

App xogo

```
cuando Screen1 .Inicializar
ejecutar
  Llamar ocultarPelotas
  poner Reloj1 . TemporizadorHabilitado como falso
  poner global enJuego a falso
```

```
como ocultarPelotas
ejecutar
  poner Pelota1 . Visible como falso
  poner Pelota2 . Visible como falso
  poner Pelota3 . Visible como falso
  poner Pelota4 . Visible como falso
  poner Pelota5 . Visible como falso
  poner Pelota_Usuario . Visible como falso
```

Control do temporizador

App xogo



Posición das pelotas no inicio

App xogo

The image shows a Scratch script titled "colocarPelotas" with a "como" icon. The script is enclosed in a "ejecutar" loop. It contains 17 blocks: 10 "poner" blocks for setting the X and Y coordinates of five balls (Pelota1 to Pelota5) and one "Pelota_Usuario" ball, and 7 "poner" blocks for setting the "Visible" property of each of these balls to "cierto".

```
como colocarPelotas
ejecutar
  poner Pelota1 . X como 0.2 x Lienzo1 . Ancho
  poner Pelota1 . Y como 0.2 x Lienzo1 . Alto
  poner Pelota2 . X como 0.75 x Lienzo1 . Ancho
  poner Pelota2 . Y como 0.2 x Lienzo1 . Alto
  poner Pelota3 . X como 0.2 x Lienzo1 . Ancho
  poner Pelota3 . Y como 0.75 x Lienzo1 . Alto
  poner Pelota4 . X como 0.75 x Lienzo1 . Ancho
  poner Pelota4 . Y como 0.75 x Lienzo1 . Alto
  poner Pelota5 . X como 0.45 x Lienzo1 . Ancho
  poner Pelota5 . Y como 0.45 x Lienzo1 . Alto
  poner Pelota_Usuario . X como 0.1 x Lienzo1 . Ancho
  poner Pelota_Usuario . Y como 0.1 x Lienzo1 . Alto
  poner Pelota1 . Visible como cierto
  poner Pelota2 . Visible como cierto
  poner Pelota3 . Visible como cierto
  poner Pelota4 . Visible como cierto
  poner Pelota5 . Visible como cierto
  poner Pelota_Usuario . Visible como cierto
```


Iniciar / Reiniciar xogo

App xogo

```
cuando BT_Reiniciar - .Clic
ejecutar
  Llamar colocarPelotas -
  poner Reloj1 - . TemporizadorHabilitado - como cierto -
  poner global enJuego - a cierto -
  poner global tiempo - a 15
  poner global objetivos - a 5
  poner ET_Texto - . Texto - como unir " Tiempo restante: "
  tomar global tiempo -
```

Movimiento da pelota do usuario

App xogo

```
cuando Acelerómetro1 . CambioEnAcceleración
  xAccel yAccel zAccel
ejecutar
  poner Pelota_Usuario . X como Pelota_Usuario . X + (-1) × tomar xAccel
  poner Pelota_Usuario . Y como Pelota_Usuario . Y + tomar yAccel
```

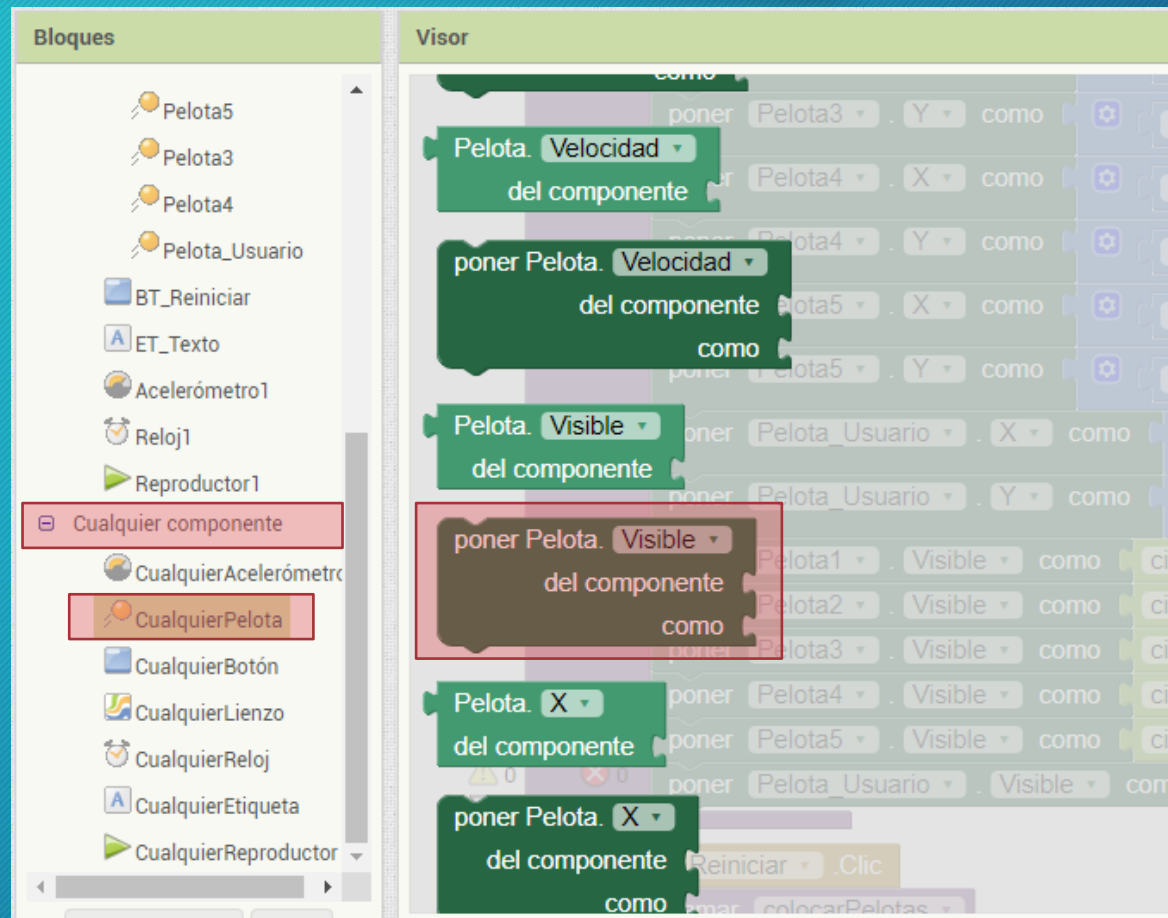
Exemplo en Y: -10 (cando o dispositivo está vertical e apuntando hacia o chan)
+10 (cando o dispositivo está vertical, e hacia arriba). Para

Exemplo en X: +10 (cando está completamente inclinado coa pantalla hacia a esquerda)
-10 (cando está completamente inclinado coa pantalla hacia a dereita).

Facer invisible as pelotas xa colisionadas

App xogo

- Para esta tarefa empregaremos unha instrucción particular que se encontra no grupo de instrucións de “calqueira compoñente” e dentro dela en “calqueira pelota”



Colisión das pelotas

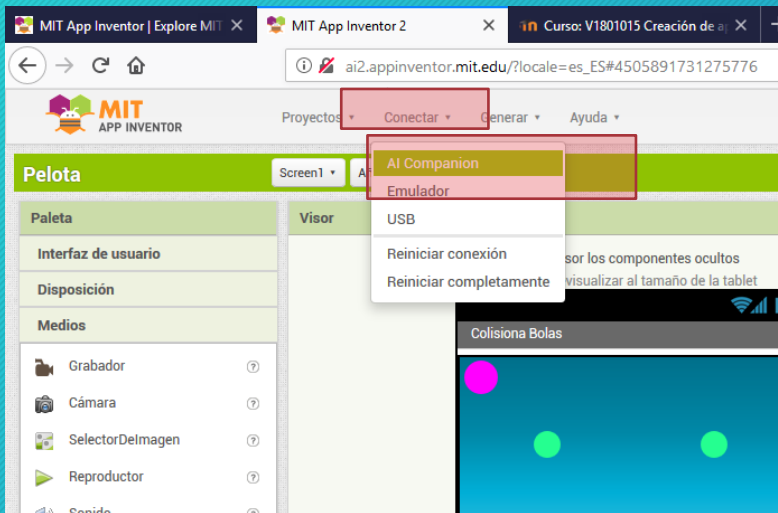
App xogo

```
cuando Pelota_Usuario .EnColisiónCon
  otro
  ejecutar
    si
      entonces
        poner Pelota.Visible del componente como falso
        poner global objetivos a tomar global objetivos - 1
        poner Reproductor1. Origen como "toquePelota.wav"
        llamar Reproductor1. Iniciar
      si
        entonces
          poner Reproductor1. Origen como "victoria.mp3"
          llamar Reproductor1. Iniciar
          poner ET_Texto. Texto como unir "ENHORABUENA! (Tiempo: "
            tomar global tiempo
            " )"
          poner global enJuego a falso
          poner global tiempo a 15
          poner global objetivos a 5
```

Probemos o noso xogo

App xogo

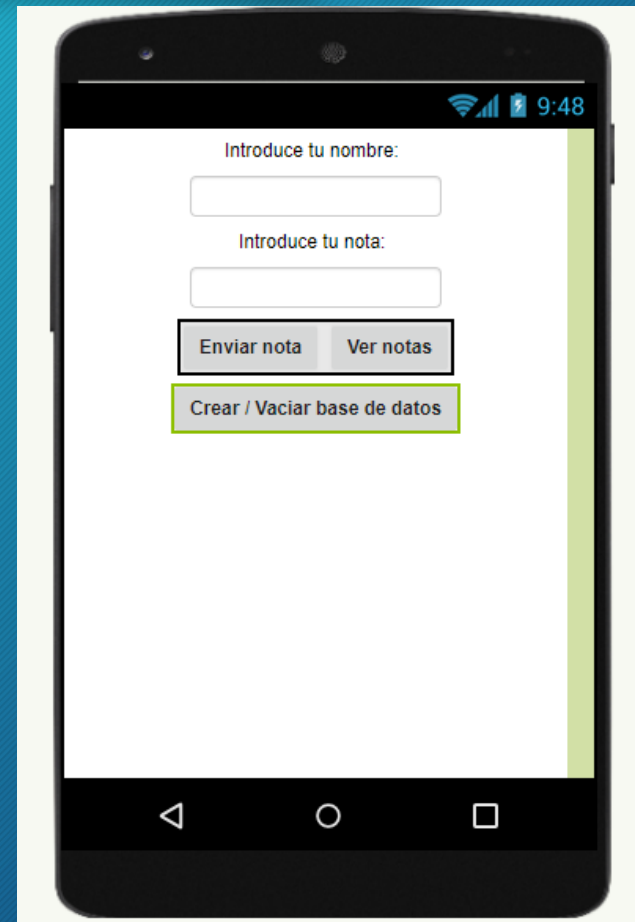
No teu proxecto, selecciona “Conectar” e “AI Companion”.



“App autoavaliación dun exame con almacenamento en base de datos externa”

Auto-
avaliación

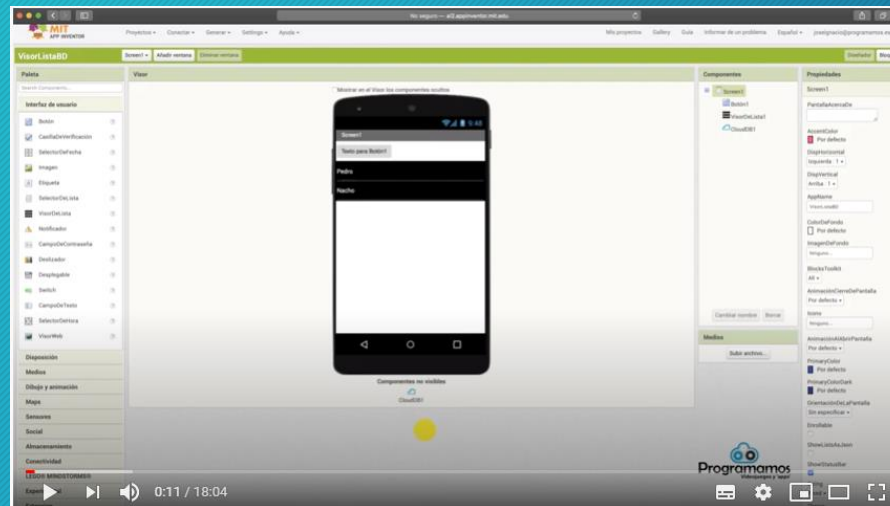
- Esta app permite a un alumno poñer o seu nome (ou usuario que lle queira dar o profesor) e indicar a nota que cree ter no exame. Tódalas notas almacénanse nunha base de datos externa que pode ser consultada para ver tódalas notas dos alumnos (incluímos unha opción de baleirado da base de datos que soamente debería ser no profesor)
- Precisaremos:
 - Catro etiquetas.
 - Dous campos de texto.
 - Dous botóns.
 - MiniWebDB



Alternativas a MiniWebDB: CloudDB

Auto-
avaliación

- App Inventor proporciona otra alternativa a MiniWebDB que é CloudDB cunhas instruccións de uso moi similares. Indico un enlace da canle de “Programamos” que mostra como facer uso dela xunto con un visor de lista.



<https://www.youtube.com/watch?v=pUBJmMKE3k0>

Elementos na parte de diseño

Auto-
avaliación

The image shows a design tool interface for a mobile application. On the left, a smartphone mockup displays the app's UI. At the top, there is a checkbox labeled "Mostrar en el Visor los componentes ocultos" and a dropdown menu for "Tamaño del teléfono (505,320)". The app screen contains the following elements:

- Header: "Introduce tu nombre:" followed by a text input field.
- Header: "Introduce tu nota:" followed by a text input field.
- Buttons: "Enviar nota" and "Ver notas" (highlighted with a red box).
- Button: "Vaciar base de datos" (disabled).

On the right, a "Componentes" (Components) palette lists the following elements:

- Screen1
 - ET_Nombre
 - CT_Alumno
 - ET_Nota
 - CT_Nota
- DisposiciónHorizontal1
 - BT_EnviarNota
 - BT_VerNotas
 - BT_VaciarBD
 - ET_NOTAS
 - MiniWebDB1

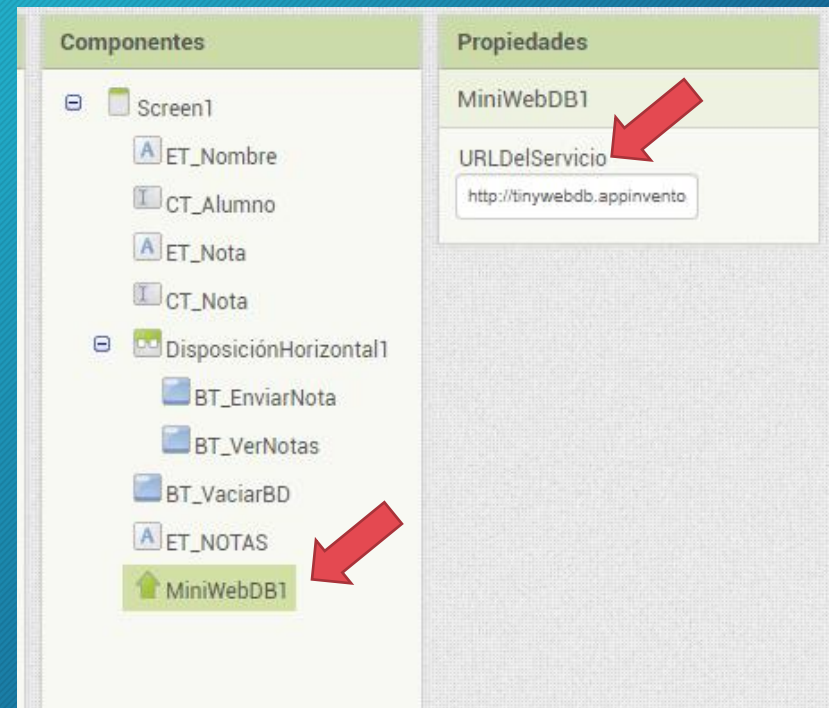
Configuración da URL da base de datos externa

Auto-
avaliación

- Empregaremos a base de datos externa que nos ofrece APP Inventor para facer os nosos test.

* **Advertencia:** outro usuario podería modificar os datos que almacenamos e ademais temos límites na cantidade de información que podemos gardar nesa base de datos.

<http://tinywebdb.appinventor.mit.edu/>

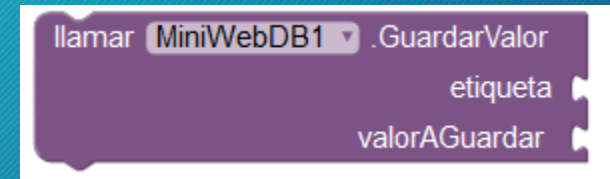


Envío e recuperación de datos da BD externa

Auto-
avaliación

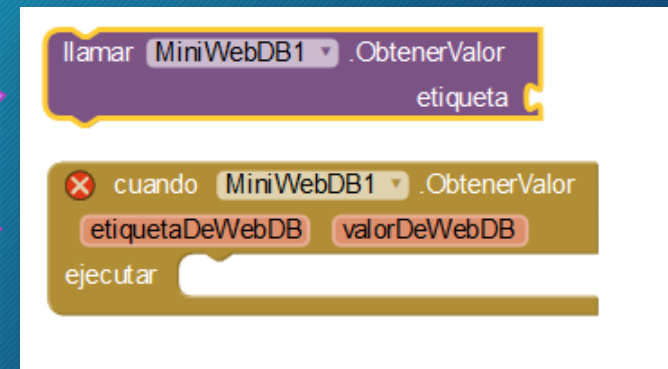
Envío mediante un par de valores:

1. Dato que quero enviar
2. Nome da variable que quedará na BD externa (para despois poder acceder a el)



Recuperación dos valores en dous pasos:


1. Solicitud
2. Evento que detecta que o noso móbil xa conseguiu os datos solicitados




Variables globais empregadas


Auto-
avaliación

- Dúas listas para gardar as notas e os nomes de alumnos e os nomes das variables onde gardamos os nosos datos na base de datos externa.

inicializar global listaAlumnos como  crear una lista vacía

inicializar global listaNotas como  crear una lista vacía

inicializar global alumnosBD como  " alumnos "

inicializar global notasBD como  " notas "

Descargar os valores da base de datos

Auto-
avaliación

- No caso de que non existan, deberemos dar ao botón de “baleirar a nosa base de datos” para que cree as listas baleiras.

cuando Screen1 .Inicializar

ejecutar Llamar descargarValoresBD

como descargarValoresBD

ejecutar

llamar MiniWebDB1 .ObtenerValor

etiqueta

tomar global alumnosBD

llamar MiniWebDB1 .ObtenerValor

etiqueta

tomar global notasBD

Evento de selección da información que chega da base de datos externa

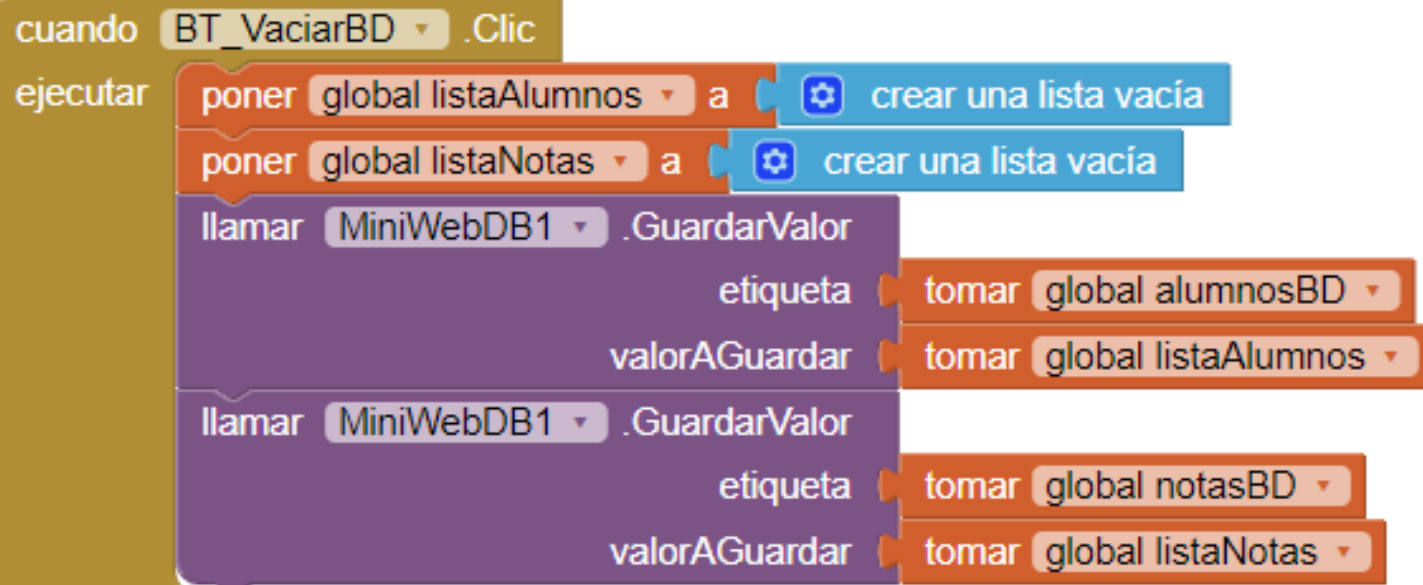
Auto-
avaliación

- Temos dous valores almacenados na base de datos (lista de alumnos e lista de notas na mesma orde). Cando chegan os datos a nosa aplicación, o que facemos e gardalos nas correspondentes variables globais da nosa app.

```
cuando MiniWebDB1 .ObtenerValor
  etiquetaDeWebDB valorDeWebDB
  ejecutar
    si
      tomar etiquetaDeWebDB = tomar global alumnosBD
    entonces
      poner global listaAlumnos a tomar valorDeWebDB
    si
      tomar etiquetaDeWebDB = tomar global notasBD
    entonces
      poner global listaNotas a tomar valorDeWebDB
```

Baleirar a base de datos ou crear os valores de novo

Auto-
avaliación



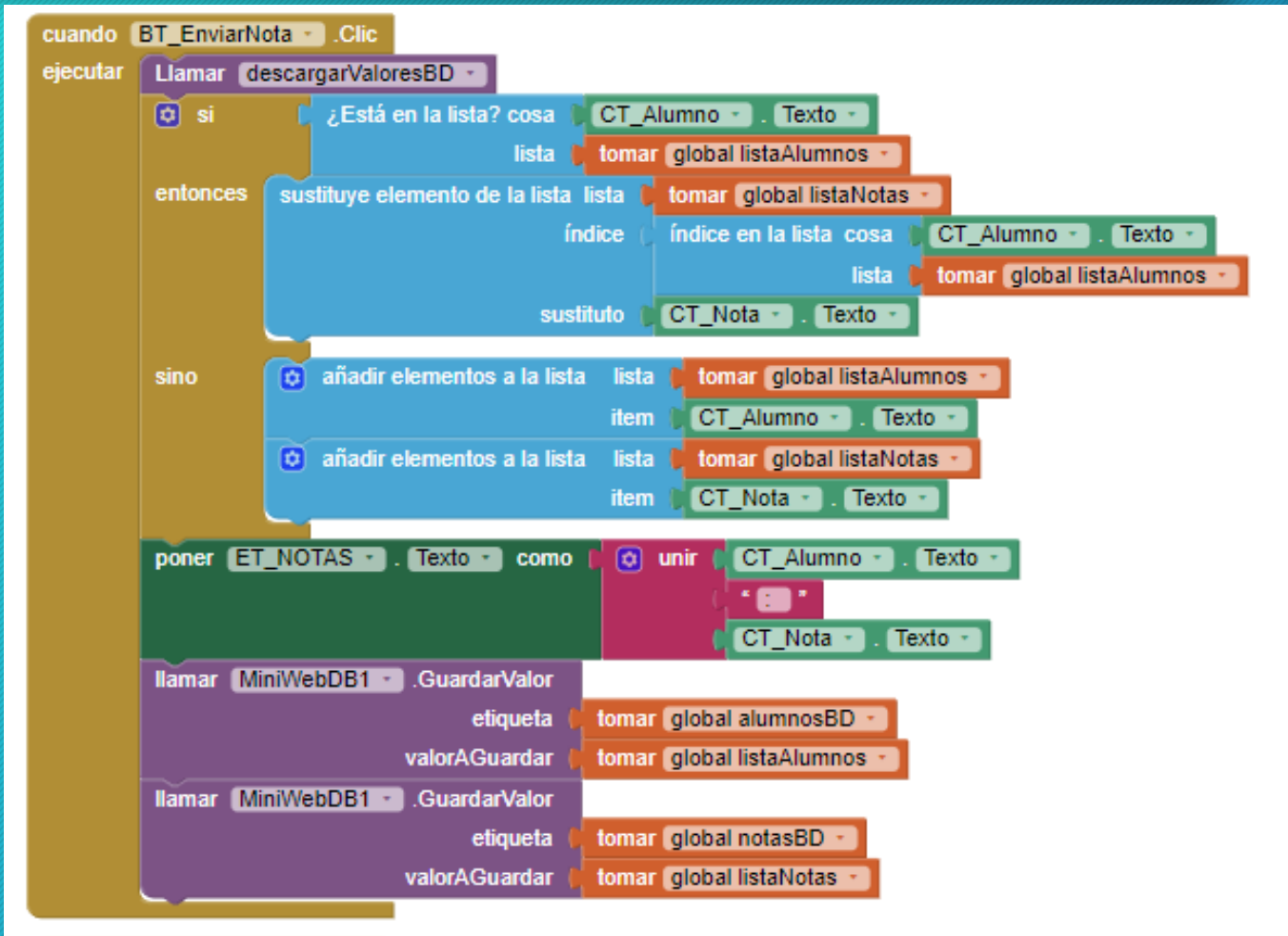
Botón “Enviar nota”

Auto-
avaliación

1. Descargamos a lista de valores da base de datos.
2. Insertamos o novo alumno coa nova nota nas correspondentes listas (se xa existe simplemente modifica a nota)
3. Enviamos a base de datos externa as novas listas.
4. Mostramos nunha etiqueta o nome do alumno e a nota que enviou.

Botón “Enviar nota”

Auto-
avaliación



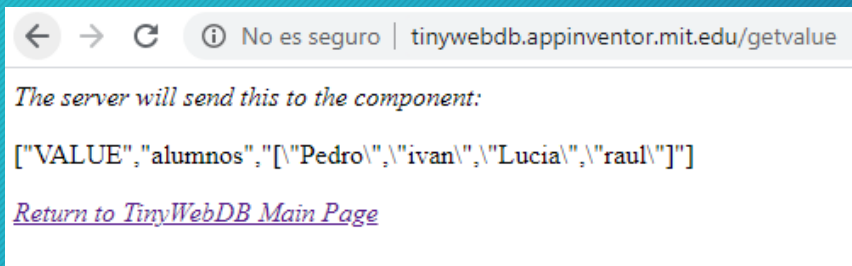
The image shows a Scratch script for a button labeled "BT_EnviarNota". The script is triggered by a ".Clic" event and is enclosed in an "ejecutar" block. The logic is as follows:

- Llamar** `descargarValoresBD`
- si** `¿Está en la lista?` `cosa` is `CT_Alumno` `. Texto`
 - lista** is `tomar global listaAlumnos`
 - entonces**
 - sustituye elemento de la lista** `lista` is `tomar global listaNotas`
 - índice** is `índice en la lista cosa` `CT_Alumno` `. Texto`
 - lista** is `tomar global listaAlumnos`
 - sustituto** is `CT_Nota` `. Texto`
 - sino**
 - añadir elementos a la lista** `lista` is `tomar global listaAlumnos`
 - item** is `CT_Alumno` `. Texto`
 - añadir elementos a la lista** `lista` is `tomar global listaNotas`
 - item** is `CT_Nota` `. Texto`
- poner** `ET_NOTAS` `. Texto` **como** `unir` `CT_Alumno` `. Texto`, `" : "`, and `CT_Nota` `. Texto`
- llamar** `MiniWebDB1` `. GuardarValor` with **etiqueta** `tomar global alumnosBD` and **valorAGuardar** `tomar global listaAlumnos`
- llamar** `MiniWebDB1` `. GuardarValor` with **etiqueta** `tomar global notasBD` and **valorAGuardar** `tomar global listaNotas`

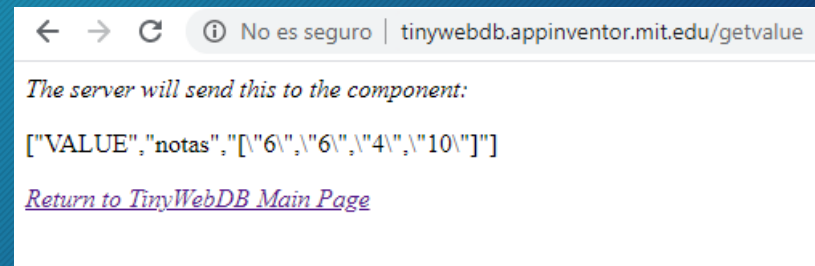
Comprobar os valores na Web

Auto-
avaliación

- Sempre podemos comprobar que os datos se gardan correctamente na nosa base de datos empregando un navegador Web e accedendo a correspondente URL: <http://tinywebdb.appinventor.mit.edu/>
 1. Seleccionamos /storeavalue para almacenar nos manualmente un valor.
 2. Seleccionamos /getvalue para comprobar que ten almacenado na base de datos coa correspondente etiqueta que lle escriba. Por exemplo se consultamos “alumnos” e “notas” obtemos 4 alumnos e as 4 notas:



```
← → ↻ ⓘ No es seguro | tinywebdb.appinventor.mit.edu/getvalue  
The server will send this to the component:  
["VALUE", "alumnos", ["Pedro", "ivan", "Lucia", "raul"]]  
Return to TinyWebDB Main Page
```



```
← → ↻ ⓘ No es seguro | tinywebdb.appinventor.mit.edu/getvalue  
The server will send this to the component:  
["VALUE", "notas", ["6", "6", "4", "10"]]  
Return to TinyWebDB Main Page
```

Botón “Ver notas” de tódolos alumnos

Auto-
avaliación

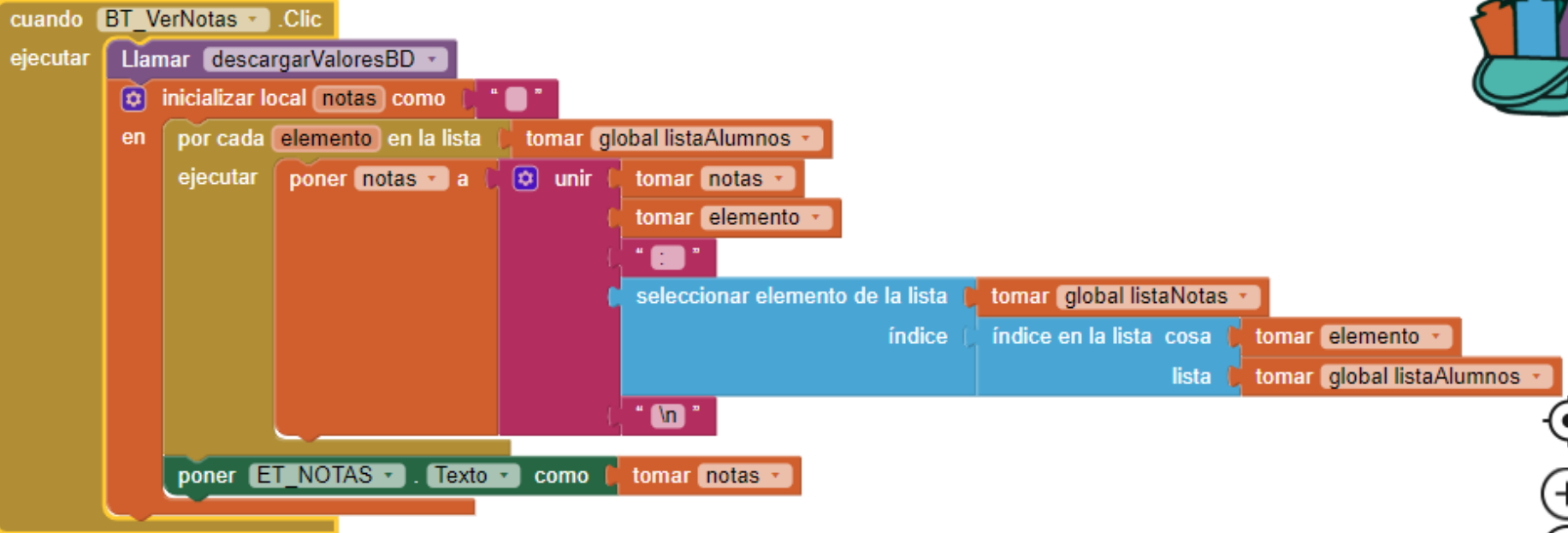
- É preciso ir recorrendo ambas listas (a de alumnos e a de notas) e ir gardando esa información nunha cadea que será a que se mostre na etiqueta correspondente ao usuario da nosa app.
- Para conseguir un salto de línea é preciso meter na nosa cadea os caracteres `\n`
- O seguinte código mostrará a lista na nosa app da seguinte maneira:

Alumno1: 10

Alumno2: 6.5

Botón “Ver notas” de tódolos alumnos

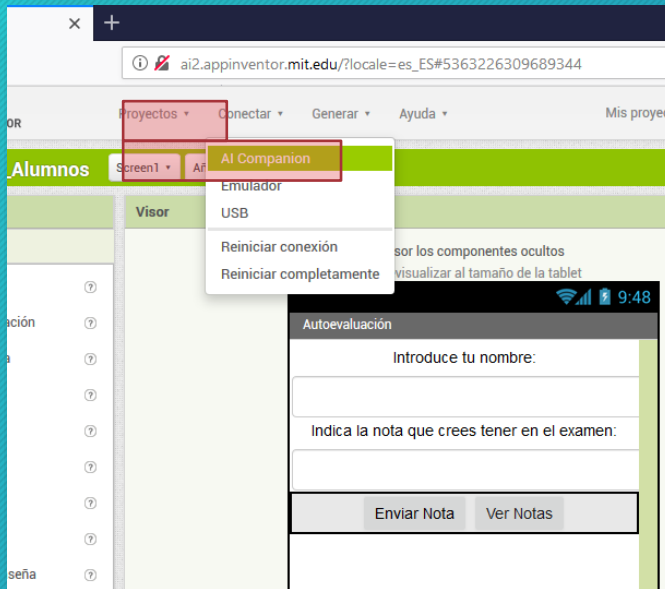
Auto-
avaliación



Probemos a nosa app

Auto-
avaliación

No teu proxecto, selecciona “Conectar” e “AI Companion”.

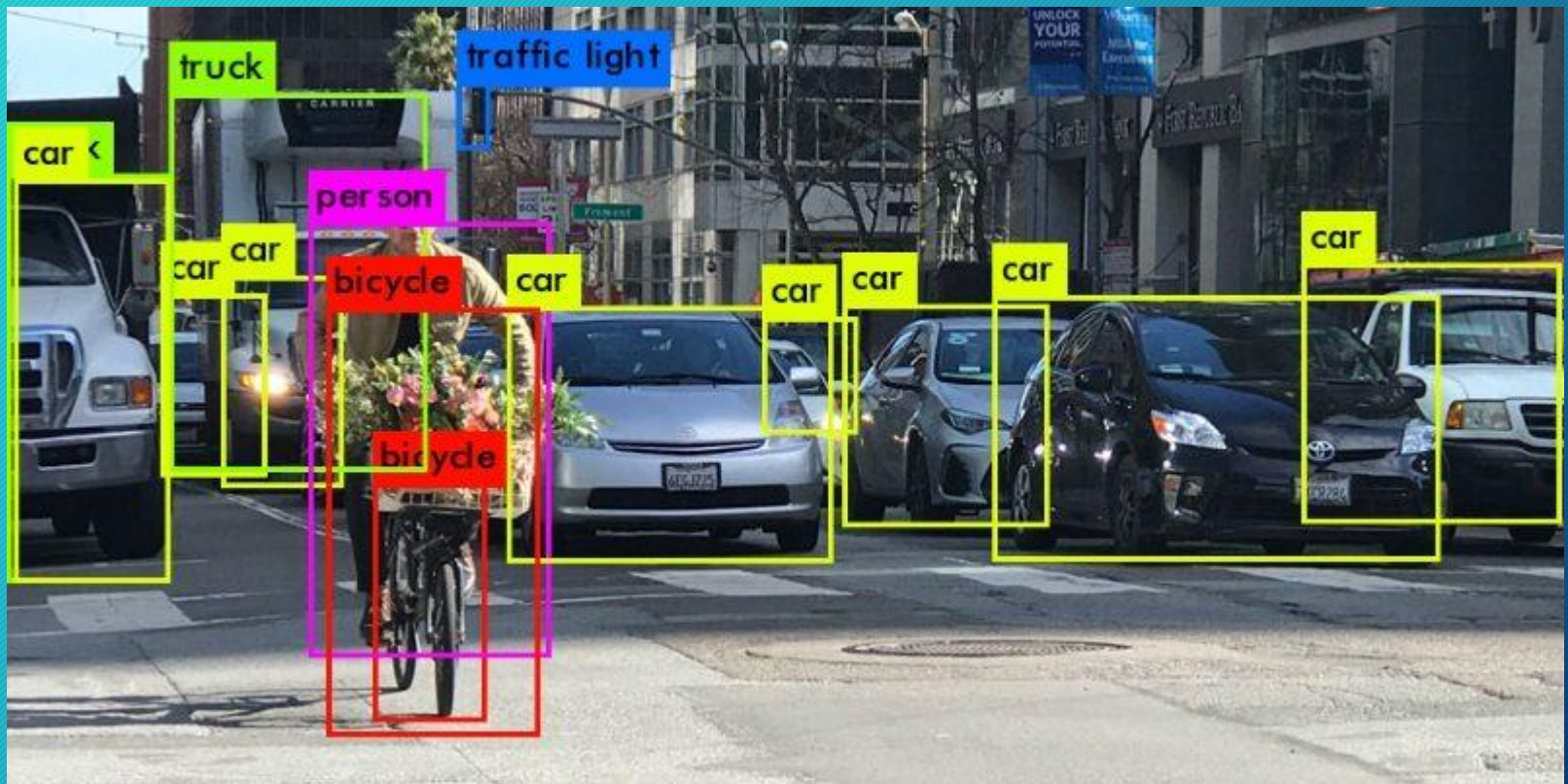


¿Que é a intelixencia artificial

A intelixencia artificial é a simulación da intelixencia humana por parte das máquinas. Trátase da combinación hardware-software capaz de realizar accións emulando o modo de pensar e razoar dos seres humanos.



Visión artificial

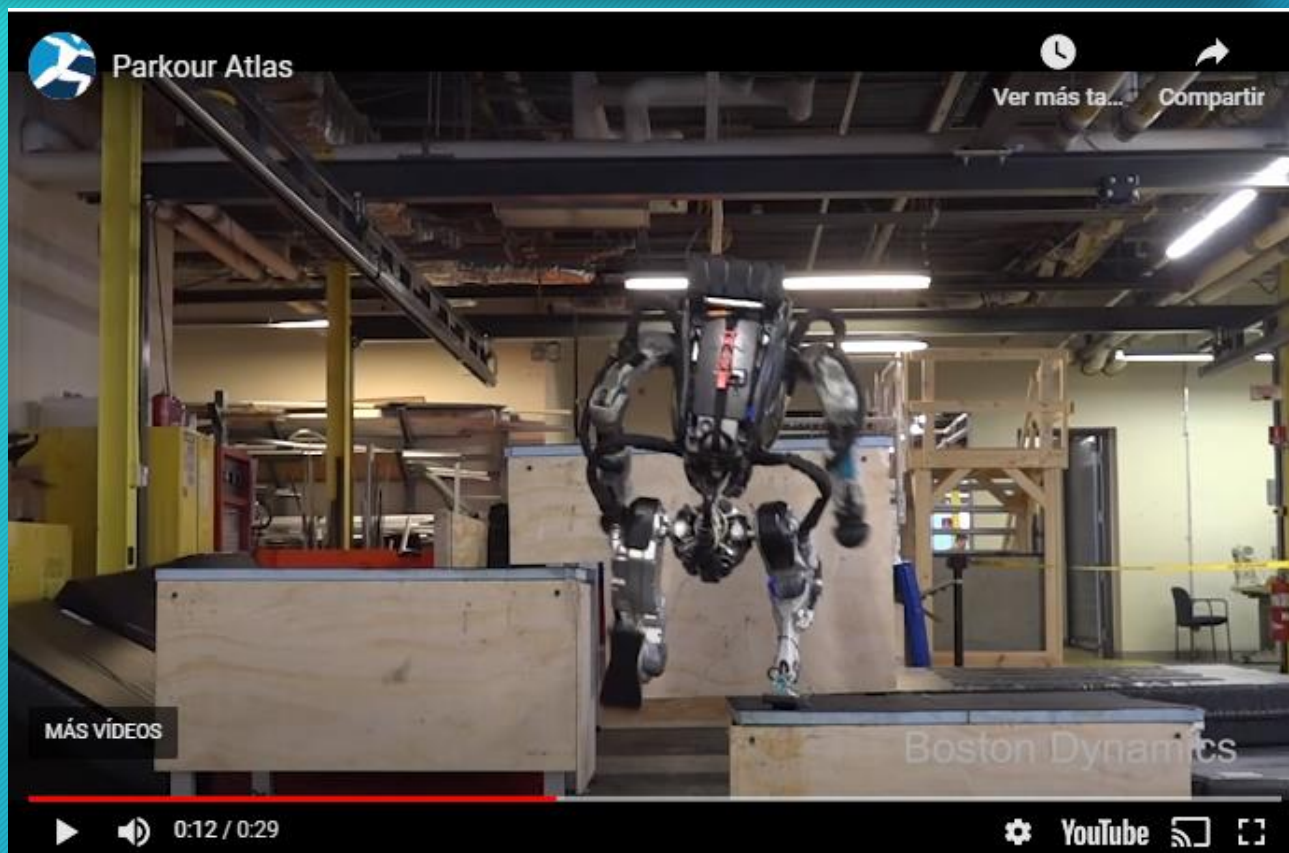


Recoñecemeto de imaxes, detección de obxectos, recoñecemento, clasificación...

Realidade virtual



Robótica



<https://youtu.be/LikxFZZO2sk>

Recoñecemento de voz e interpretación



¿Que é o aprendizaxe automático (*machine learning*)?

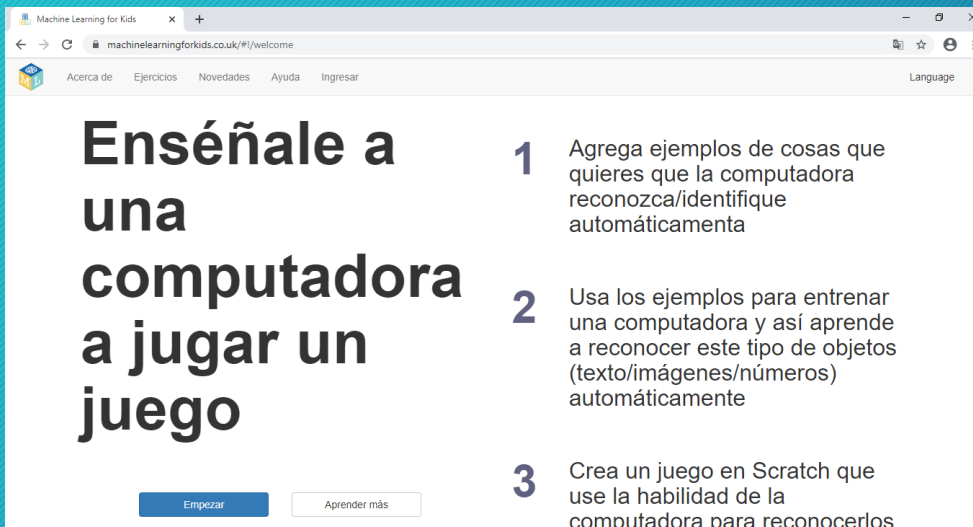
É unha das ramas de da intelixencia artificial na que se proporcionan datos ás máquinas e a través de diferentes algoritmos son elas as que aprendan e extraen as relación entre eses datos sen intervención humana.

Un dos algoritmos máis empregados é aprendizaxe supervisado:

- Moi empregado para clasificación (etiquetar un dato, unha imaxe, un conxunto de características... nunha determinada clase)
- Primeira fase de adestramento: a base de coñecemento basease en exemplos etiquetados previamente
- Segunda fase de consulta: nesta fase incorpórase un novo dato ao modelo para consultar a clase á que pertence.

Machine learning for kids (ML4K)

Ferramenta educacional para introducir aos nosos alumnos no mundo do aprendizaxe automático. Trátase dun contorno de aprendizaxe guiado para adestrar modelos sinxelos nos que se pode clasificar texto, números ou imaxes.



The screenshot shows the homepage of the Machine Learning for Kids website. The main heading reads "Enséñale a una computadora a jugar un juego". Below this, there are three numbered steps: 1. "Agrega ejemplos de cosas que quieres que la computadora reconozca/identifique automáticamente", 2. "Usa los ejemplos para entrenar una computadora y así aprende a reconocer este tipo de objetos (texto/imágenes/números) automáticamente", and 3. "Crea un juego en Scratch que use la habilidad de la computadora para reconocerlos". At the bottom of the main content area, there are two buttons: "Empezar" (Start) and "Aprender más" (Learn more).



<https://machinelearningforkids.co.uk/#!/welcome>

Crear conta de ML4K e conta en IBM Cloud



Empregarase unha conta de ML4K que precisa dos motores de algoritmos que lle proporciona IBM Cloud. Os pasos a seguir son:

1. Creación da conta de ML4K (como profesor)

Url da Web: <https://machinelearningforkids.co.uk/>
https://youtu.be/ukdg9pdnPGU?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp- (Videotutorial do INTEF)

2. Creación da conta de IBM Cloud

3. Obtención da clave de API para ML4K

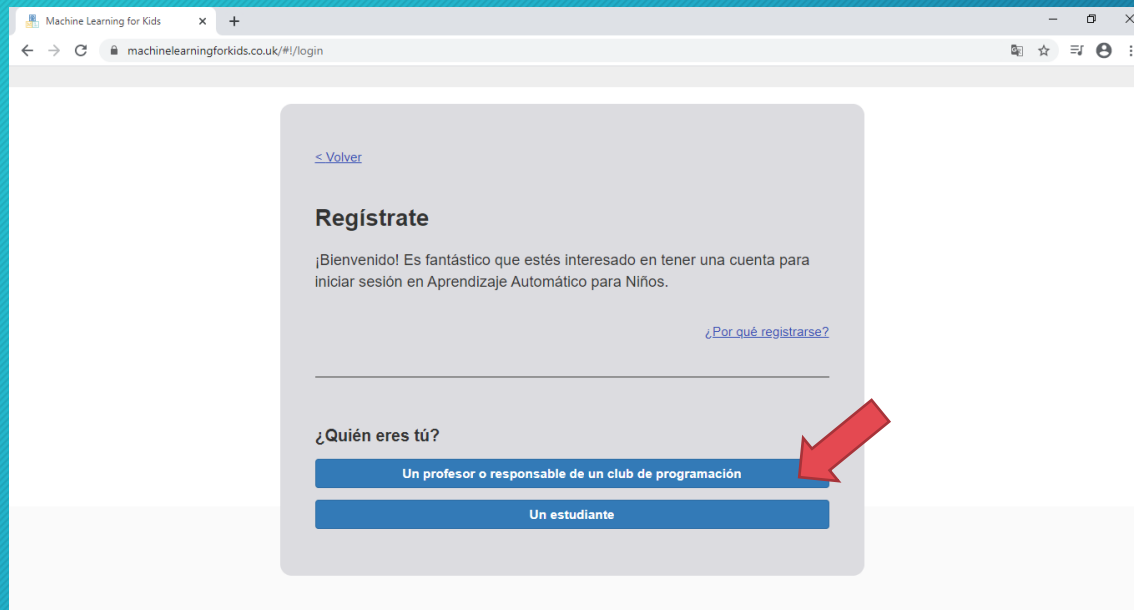
Url da Web: <https://www.ibm.com/es-es/cloud>
https://www.youtube.com/watch?v=rvhl3YClFXI&feature=youtu.be&list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-
(Videotutorial do INTEF)

Paso 1: Crear conta de ML4K



Creación da conta de ML4K (como profesor)

Url da Web: <https://machinelearningforkids.co.uk/>



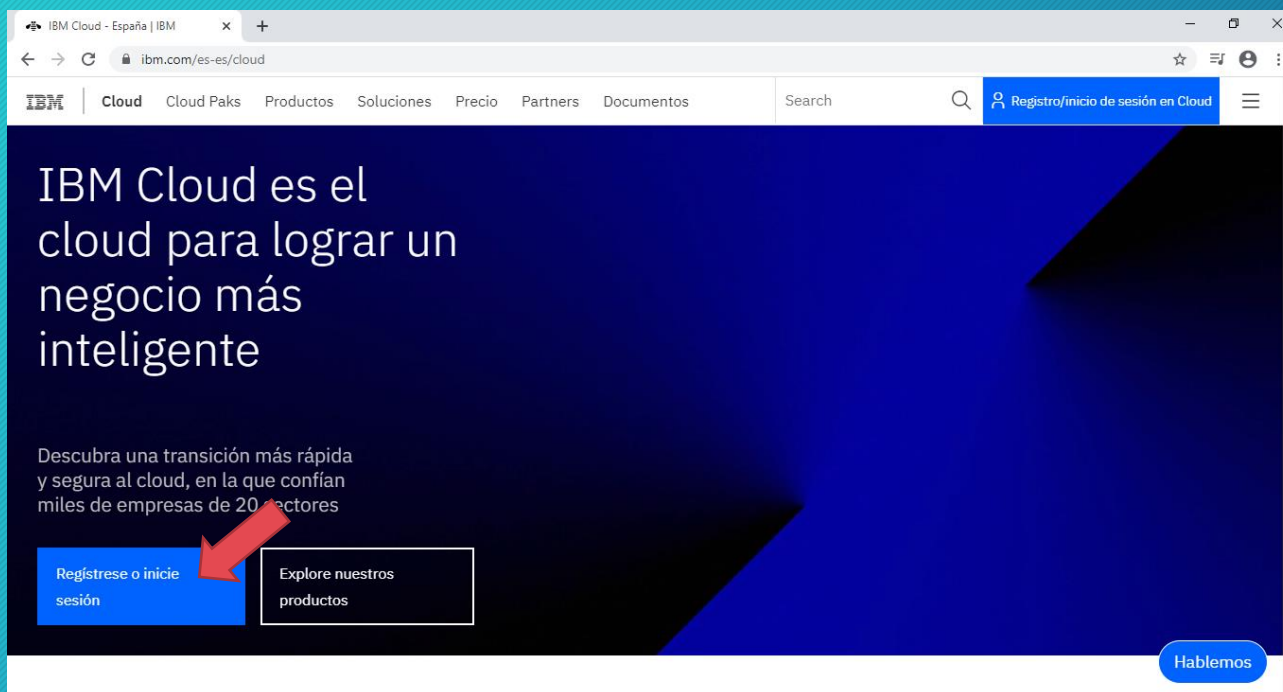
Pasos a seguir:

https://youtu.be/ukdg9pdnPGU?list=PL70-wFTtwWAYrJFR_85FShl0COrn7TOp-

Paso 2: Crear conta en IBM Cloud



Url da Web: <https://www.ibm.com/es-es/cloud>



Pasos a seguir:

https://www.youtube.com/watch?v=rvgl3YClFXI&feature=youtu.be&list=PL70-wFTtwWAYrJFR_85FShl0COrn7TOp-

Paso 3: Incorporar as clave de API (Watson Assistant e Watson Visual Recognition) en ML4K



The screenshot shows the ML4K website interface. At the top, there's a navigation bar with 'Machine Learning for Kids' and 'Página de servicio de IBM Watson Studio'. Below that, a message says: 'Si no estás seguro de qué hacer aquí, sigue la guía paso a a paso en . Si necesitas ayuda, por favor, [contacta con nosotros](#).' The main content is divided into sections for different project types:

- text**: 'El entrenamiento de modelos de aprendizaje automático para proyectos de **text** usa **Watson Assistant**'. It shows an API key: 'I1EsFCkfmQjs39p_PUPJ1IM5NR3FJ9dByzHCe_NmuYEe' and a button 'Añadir nuevas credenciales'. Below it, it says 'Número de modelos ML: 5' and 'Número textos que pueden tener los modelos de aprendizaje automático de tus alumnos.: 5'.
- imagen**: 'El entrenamiento de modelos de aprendizaje automático para proyectos de **imagen** usa **Watson Visual Recognition**'. It shows an API key: 'xOCFr1D_0bgmg88-bj_TuoxTWFL3vYee0U28Tup7R_'. Below it, it says 'Número imágenes que pueden tener los modelos de aprendizaje automático de tus alumnos.: 0'.
- numbers**: 'Training machine learning models for **numbers** projects does **not** require any API keys'.

At the bottom, there's a video player showing a progress bar at 8:23 / 10:37 and a caption: 'Training machine learning models for **sound** projects does **not** require any API keys'.

Pasos a seguir:

https://www.youtube.com/watch?v=rvgl3YClFXI&feature=youtu.be&list=PL70-wFTtwWAYrJFR_85FShl0COrn7TOp-

Exemplo de APP facendo uso de ML4K (adiviña sentimentos)



Aplicación que distingue entre 3 sentimentos (triste, alegre, indeciso) en función da frase que lle indiques.

1. Creación do modelo en ML4K:

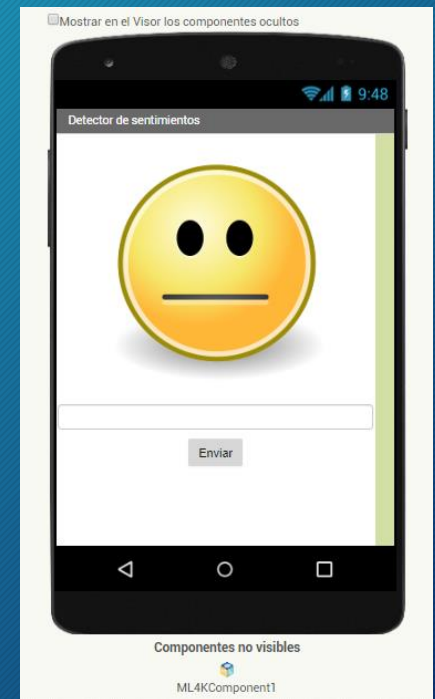
https://www.youtube.com/watch?v=VNe-QMBsY4o&feature=youtu.be&list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-

2. Xerar o modelo de aprendizaxe:

https://youtu.be/tKMwpWMg8tM?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-

3. Probar o modelo de aprendizaxe:

https://youtu.be/X_hg-7CRrnl?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-



Exemplo de APP facendo uso de ML4K (adiviña sentimentos)



4. Datos sesgados no modelo:

https://youtu.be/bvSPcqLOmnE?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-

5. Integramos o modelo na nosa aplicación:

https://youtu.be/qhL7rqGejSM?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-

6. Programamos a nosa app:

https://youtu.be/KzC6n3Yu4ks?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-

7. Probamos a versión final:

https://youtu.be/qoY-fhNGXHYY?list=PL70-wFTtwWAYrJFR_85FShI0COrn7TOp-

“APP Destino de vacaciones”

App
Destino de
vacaciones

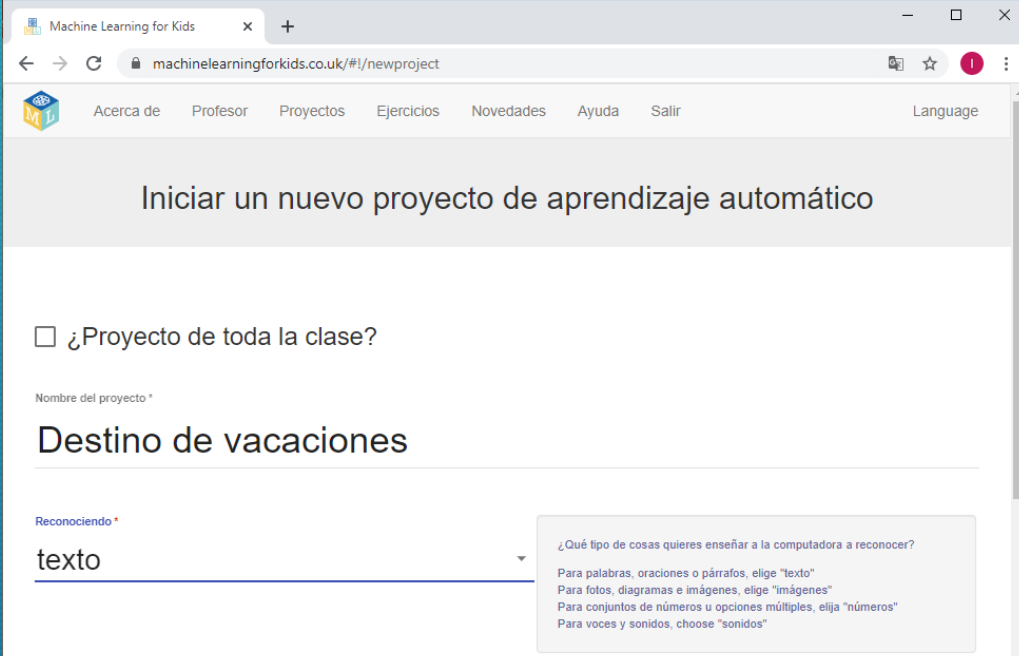
- Esta app permitirá adiviñar o destino que resulta máis adecuado para irse de vacaciones en función do que desexemos facer, que indicaremos cunha frase de texto.
- Para simplicidade escolleremos simplemente tres categorías de destinos:
 1. Praia
 2. Escapada rural ou campo
 3. Cidade



Machine Learning for Kids (ML4K). Creación proxecto

App
Destino de
vacacións

- Accedemos á páxina de ML4K:
<https://machinelearningforkids.co.uk/>
- Iniciamos sesión e creamos un novo proxecto de aprendizaxe automático:
 1. Damos nome ao proxecto
 2. Escollemos “Recoñecendo texto”

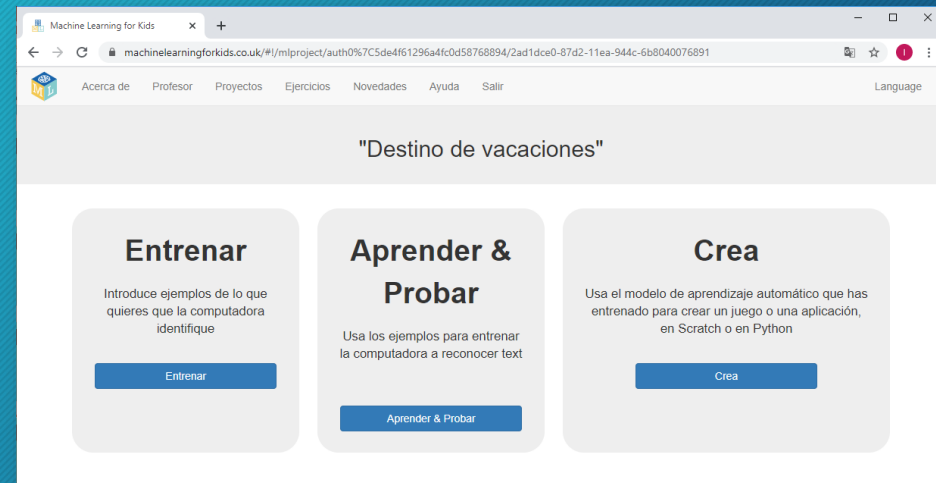


The screenshot shows a web browser window with the URL machinelearningforkids.co.uk/#/newproject. The page title is "Machine Learning for Kids". The main heading is "Iniciar un nuevo proyecto de aprendizaje automático". There is a checkbox labeled "¿Proyecto de toda la clase?". Below it, the "Nombre del proyecto" field contains the text "Destino de vacaciones". The "Reconociendo" dropdown menu is set to "texto". A tooltip is visible, explaining the options: "¿Qué tipo de cosas quieres enseñar a la computadora a reconocer?" with instructions for "texto", "imágenes", "números", and "sonidos".

Machine Learning for Kids (ML4K). Fases

App
Destino de
vacaciones

- Fase 1 (adestramento): defino as categorías e os exemplos base de cada unha delas.
- Fase 2 (aprender e probar): en base aos algoritmos de aprendizaxe crea un modelo, no cal xa podemos consultar.
- Fase 3 (crea): xa podo crear a miña aplicación facendo consultas ao modelo



Machine Learning for Kids (ML4K). Adestramento

App
Destino de
vacaciones

- Creamos categorías (Praia, Campo e Cidade) e frases que definan as actividades desas categorías.

The screenshot shows a web browser window with the URL `machinelearningforkids.co.uk/#/mlproject/auth0%7C5de4f61296a4fc0d58768894/2ad1dce0-87d2-11ea-944c-6b8040076891/training`. The page title is "Reconociendo **text** como **Playa, Ciudad or Campo**".

Below the title, there is a navigation bar with links: "Acerca de", "Profesor", "Proyectos", "Ejercicios", "Novedades", "Ayuda", "Salir", and "Language".

The main content area is titled "Reconociendo **text** como **Playa, Ciudad or Campo**". It features three columns, each representing a category:

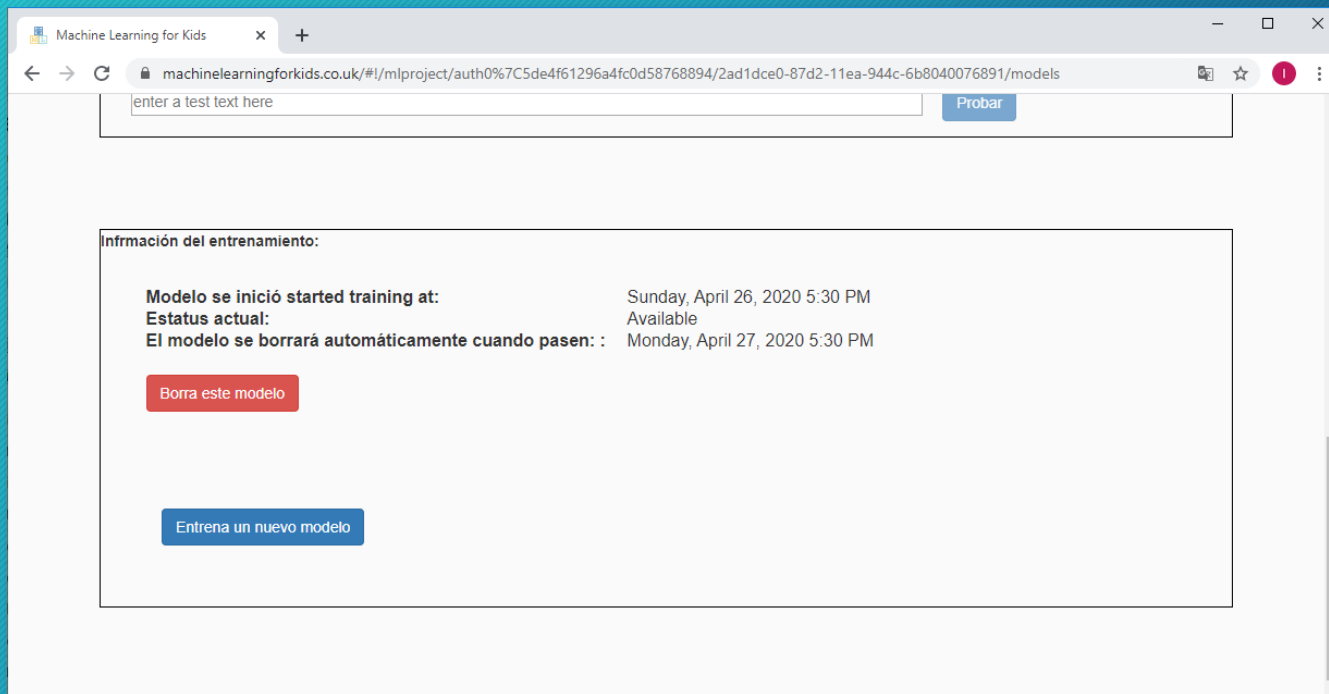
- Playa** (labeled with a circled 8):
 - Poneme bronceado
 - Ir en barco
 - Hacer surf
 - Bucear
 - Ir a un chiringuito de playa
 - + Añade un ejemplo.
- Ciudad** (labeled with a circled 4):
 - Ir a museos
 - Recorrer el casco histórico
 - Visitar catedrales
 - Ver monumentos históricos
 - + Añade un ejemplo.
- Campo** (labeled with a circled 7):
 - Dar paseos por la montaña
 - Hacer cámping
 - Ver animales salvajes por el bo...
 - Ver zonas verdes
 - + Añade un ejemplo.

At the top right of the content area, there is a button "+ Añadir etiqueta".

Machine Learning for Kids (ML4K). Aprender e probar.

App
Destino de
vacacións

- Unha vez creadas as categorías e as frases que as describen, procedemos a crear o modelo de aprendizaxe (pode tardar uns segundos en crealo). Obteremos un resultado como o seguinte (no cal xa podemos probar con frases novas):



The screenshot shows a web browser window with the URL `machinelearningforkids.co.uk/#/mlproject/auth0%7C5de4f61296a4fc0d58768894/2ad1dce0-87d2-11ea-944c-6b8040076891/models`. At the top, there is a text input field with the placeholder "enter a test text here" and a blue "Probar" button. Below this, a box titled "Infrmación del entrenamiento:" contains the following details:

Modelo se inició started training at:	Sunday, April 26, 2020 5:30 PM
Estatus actual:	Available
El modelo se borrará automáticamente cuando pasen: :	Monday, April 27, 2020 5:30 PM

Below the table, there are two buttons: a red "Borra este modelo" button and a blue "Entrena un nuevo modelo" button.

Machine Learning for Kids (ML4K). Crear aplicaciones.

App
Destino de
vacaciones

- Podemos ahora emplear o modelo para crear aplicaciones en Python, Scratch ou App Inventor. No noso caso escollemos App Inventor.

The screenshot shows the Machine Learning for Kids website interface. At the top, there is a navigation menu with links for 'Acerca de', 'Profesor', 'Proyectos', 'Ejercicios', 'Novedades', 'Ayuda', and 'Salir'. Below the menu, the main heading reads 'Usa tu modelo de aprendizaje automático para crear algo'. A link '< Volver al proyecto' is visible on the left. The main content area features four large cards, each representing a different tool for creating applications:

- Scratch**: 'Crea un Proyecto con la version antigua de Scratch'. Includes a Scratch logo and a button labeled 'Scratch'.
- Scratch 3**: 'Usa la nueva version de Scratch'. Includes a Scratch logo and a button labeled 'Scratch 3'.
- Python**: 'Programa en Python para usar tu modelo de aprendizaje automático'. Includes a Python logo and a button labeled 'Python'.
- Inventor de aplicaciones**: 'Crea una app para tu móvil o tablet'. Includes an App Inventor logo and a button labeled 'Inventor de aplicaciones'.

Each card also contains a small thumbnail image showing the respective application development environment.

Machine Learning for Kids (ML4K). Crear aplicaciones.

App
Destino de
vacaciones

- Temos que copiar a URL que identifica a extensión que empregará o noso modelo:

The screenshot shows a web browser window with the URL `machinelearningforkids.co.uk/#/mlproject/auth0%7C5de4f61296a4fc0d58768894/2ad1dce0-87d2-11ea-944c-6b8040076891/appinventor`. The page content includes:

- A button labeled "Open App Inventor".
- Text explaining that App Inventor is used to make mobile apps that run on Android phones and tablets, and that it runs in a web browser like Scratch.
- Text stating that App Inventor can be used with Machine Learning for Kids to create apps powered by artificial intelligence.
- A link to `github.com/kylecorry31/ML4K-AI-Extension` for more instructions.
- A note: "Support for App Inventor was created by Kyle Corry and Joe Mazzone".

On the right side, there is a section titled "To add your machine learning model to your App Inventor project:" with the following steps:

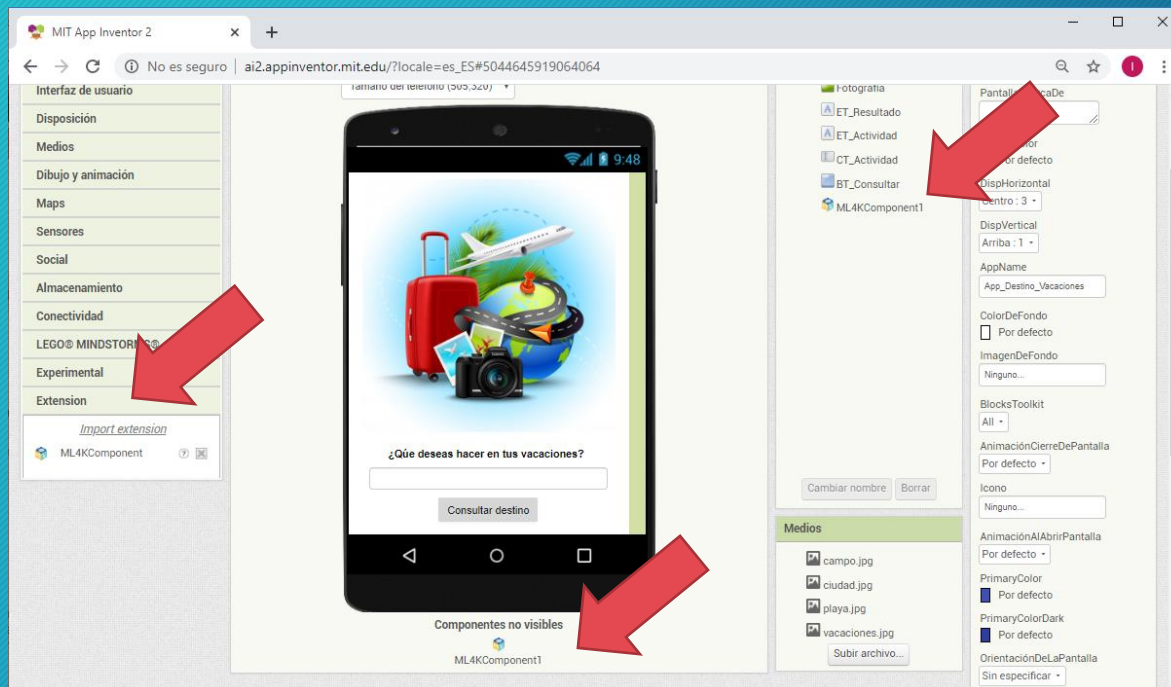
1. Click on **Import extension**
2. Click on **URL**
3. Fill in the URL for your project:

A red arrow points to the URL `https://machinelearningforkids.co.uk/api/appinventor/e686b050-87d2-11ea-944c-6b8040076891d253a7dc-2e40-4079-a1c6-de3ae47c4300/extension` which is highlighted in blue. Below this, a screenshot of the App Inventor interface shows the "Import extension" dialog box. The "URL" option is selected, and the same URL is entered into the "URL:" field. The "Import" button is visible at the bottom of the dialog.

APP Inventor importar extensión de ML4K

App
Destino de
vacaciones

- Pulsamos en importar extensión no noso proxecto de App Inventor, facemos clic en URL e copiamos (despois duns segundos aparecerá o novo elemento que podemos arrastrar ao noso proxecto):



Elementos da interface da aplicación

App
Destino de
vacacións

- Os elementos principais da app son:
 - Imaxe principal
 - Etiqueta que mostrará os resultados (ao iniciar a aplicación sen texto)
 - Etiqueta para mostrar un comentario de: “¿Que desexas facer nas túas vacacións?”
 - Campo de texto para recoller o que escriba o usuario.
 - Botón para consultar no modelo de aprendizaxe.
 - Modelo de ML4K
 - Imaxes en medios para telas dispoñibles despois (portada, praia, campo e cidade)

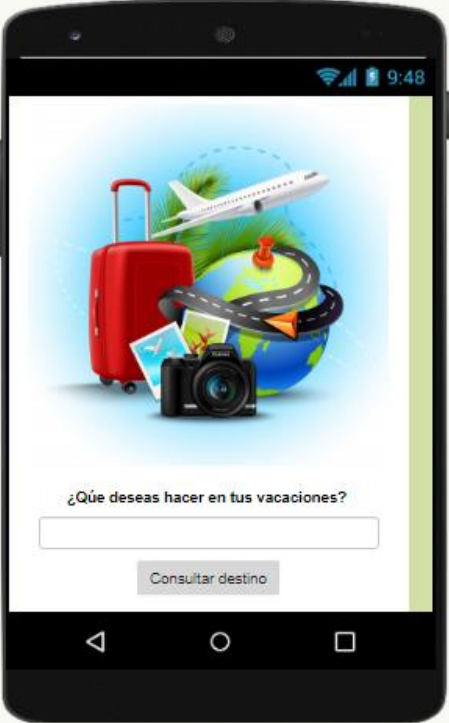
Elementos da interface da aplicación

App
Destino de
vacacións

Visor

Mostrar en el Visor los componentes ocultos

Tamaño del teléfono (505,320) ▼



¿Qué deseas hacer en tus vacaciones?

Consultar destino

Componentes no visibles

ML4KComponent1

Componentes

- Screen1
 - Fotografía
 - ET_Resultado
 - ET_Actividad
 - CT_Actividad
 - BT_Consultar
 - ML4KComponent1

Cambiar nombre Borrar

Medios

- campo.jpg
- ciudad.jpg
- playa.jpg
- vacaciones.jpg

Subir archivo...

Programación de bloques

App
Destino de
vacaciones

- Ao facer clic mandamos clasificar a nova frase.
- Cando chega a clasificación escollemos o resultado en función da % de confianza da resposta.

```
cuando BT_Consultar .Clic
ejecutar
  llamar ML4KComponent1 .ClassifyText
  data CT_Actividad . Texto

cuando ML4KComponent1 .GotClassification
  data classification confidence
  ejecutar
    si
      tomar confidence > 70
    entonces
      si
        tomar classification = "Playa"
      entonces poner Fotografía . Foto como "playa.jpg"
      si no, si
        tomar classification = "Campo"
      entonces poner Fotografía . Foto como "campo.jpg"
      sino poner Fotografía . Foto como "ciudad.jpg"
    poner ET_Resultado . Texto como unir
      " Tu destino es: "
      tomar classification
      " (Confianza del "
      tomar confidence
      " %)"
    sino
      poner Fotografía . Foto como "vacaciones.jpg"
      poner ET_Resultado . Texto como " No podemos encontrar tu destino ideal :( "
```

Exemplos de resultados

App
Destino de
vacaci3n

18:31




Tu destino es: Playa (Confianza del 100%)
¿Qué deseas hacer en tus vacaciones?

Quiero nadar y hacer castillos de arena

Consultar destino

18:31



Tu destino es: Ciudad (Confianza del 89%)
¿Qué deseas hacer en tus vacaciones?

Me gustaría visitar muchos monumentos

Consultar destino

18:32



Tu destino es: Campo (Confianza del 99%)
¿Qué deseas hacer en tus vacaciones?

Me gusta bañarme en el río y andar por el bosque

Consultar destino

18:49



No podemos encontrar tu destino ideal :(
¿Qué deseas hacer en tus vacaciones?

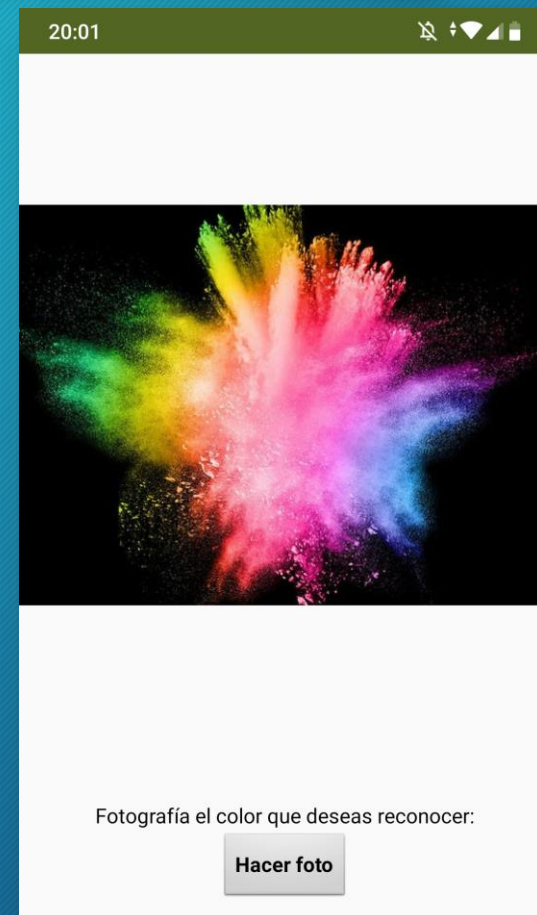
quiero descansar

Consultar destino

“APP Reconocimiento de cores”

Reconocimiento de cores

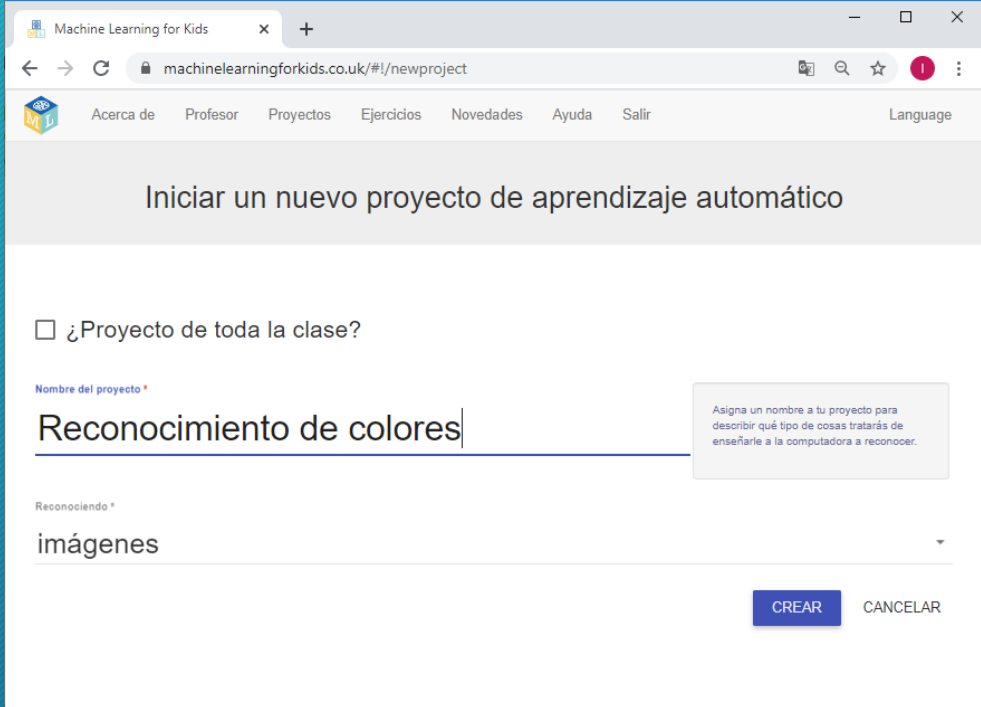
- Esta app permitirá reconocer unha cor despois de sacar unha foto.
- Para simplicidade escolleremos simplemente tres cores:
 1. Amarelo
 2. Azul
 3. Vermello



Machine Learning for Kids (ML4K). Creación proxecto

Recoñece-
mento de
cores

- Accedemos á páxina de ML4K:
<https://machinelearningforkids.co.uk/>
- Iniciamos sesión e creamos un novo proxecto de aprendizaxe automático:
 1. Damos nome ao proxecto
 2. Escollemos “Recoñecendo imaxes”

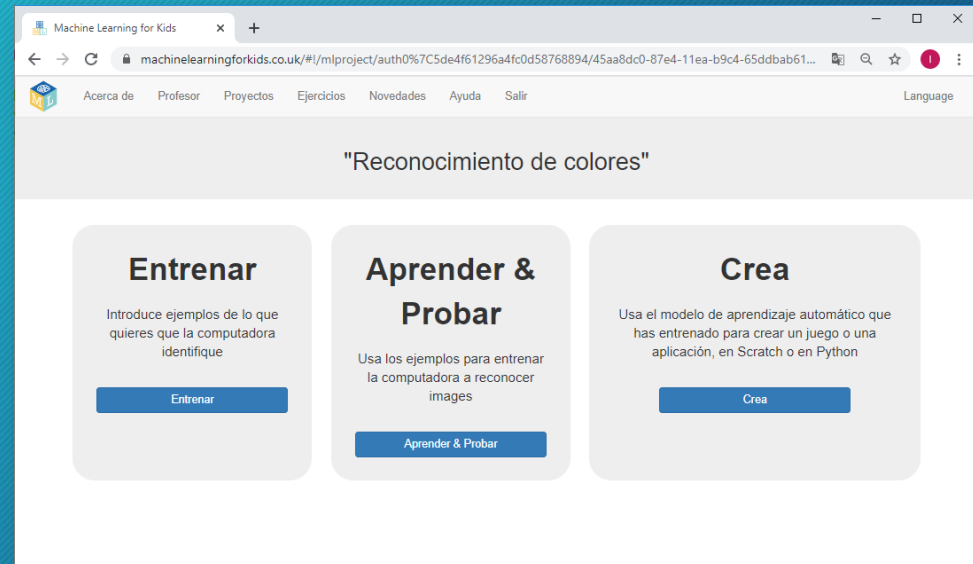


The screenshot shows a web browser window with the URL machinelearningforkids.co.uk/#/newproject. The page title is "Machine Learning for Kids". The navigation menu includes "Acerca de", "Profesor", "Proyectos", "Ejercicios", "Novedades", "Ayuda", "Salir", and "Language". The main heading is "Iniciar un nuevo proyecto de aprendizaje automático". There is a checkbox labeled "¿Proyecto de toda la clase?". Below it, the "Nombre del proyecto" field contains the text "Reconocimiento de colores". A tooltip on the right says "Asigna un nombre a tu proyecto para describir qué tipo de cosas tratarás de enseñarle a la computadora a reconocer." Below the name field, the "Reconociendo" dropdown menu is set to "imágenes". At the bottom right, there are two buttons: "CREAR" and "CANCELAR".

Machine Learning for Kids (ML4K). Fases

Recoñe-
mento de
cores

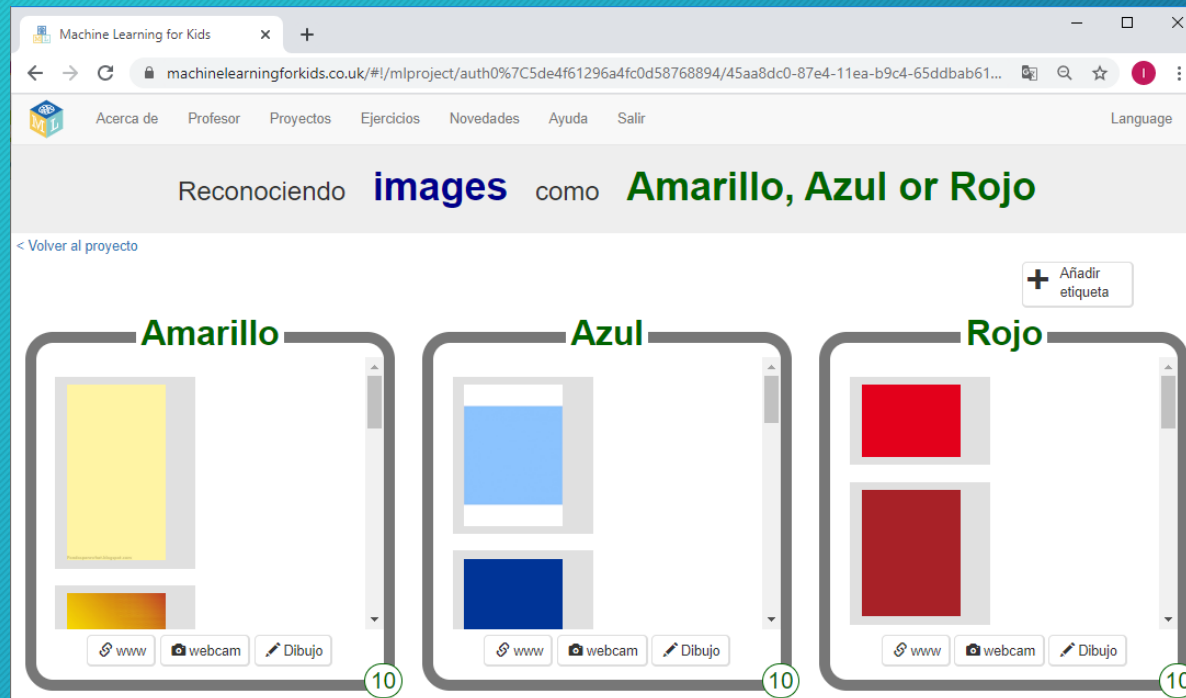
- Fase 1 (adestramento): defino as categorías e os exemplos base de cada unha delas.
- Fase 2 (aprender e probar): en base aos algoritmos de aprendizaxe crea un modelo, no cal xa podemos consultar con novas imaxes.
- Fase 3 (crea): xa podo crear a miña aplicación facendo consultas ao modelo



Machine Learning for Kids (ML4K). Adestramento

Recoñe-
mento de
cores

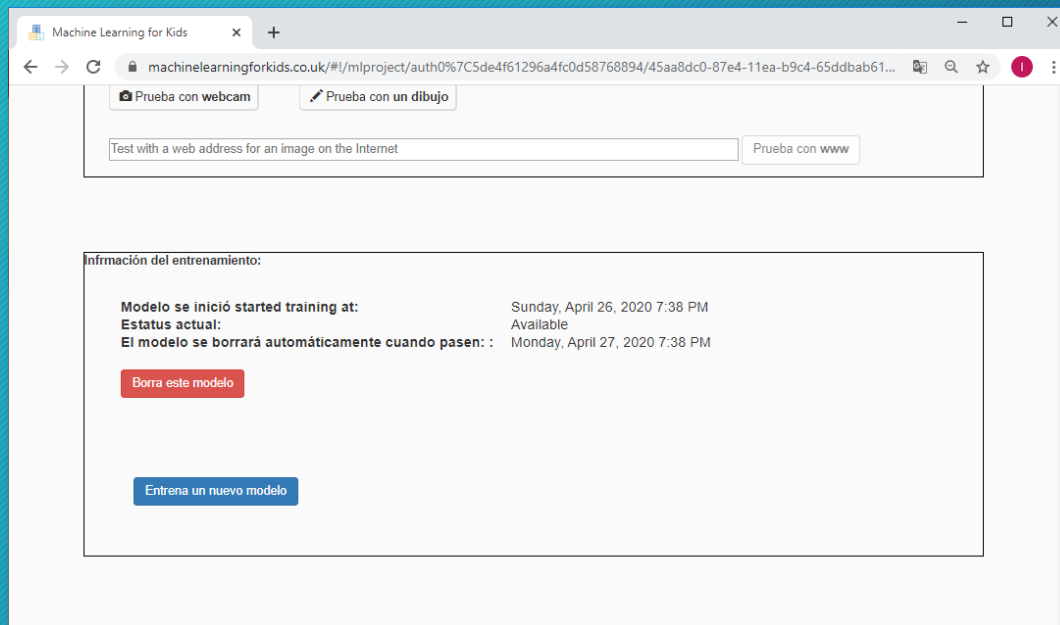
- Creamos categorías (Amarelo, Vermello, Azul) e arrastramos dende o propio navegador imaxes de cores. Temos que ter un mínimo de 10 por cada categoría.



Machine Learning for Kids (ML4K). Aprender e probar.

Recoñe-
mento de
cores

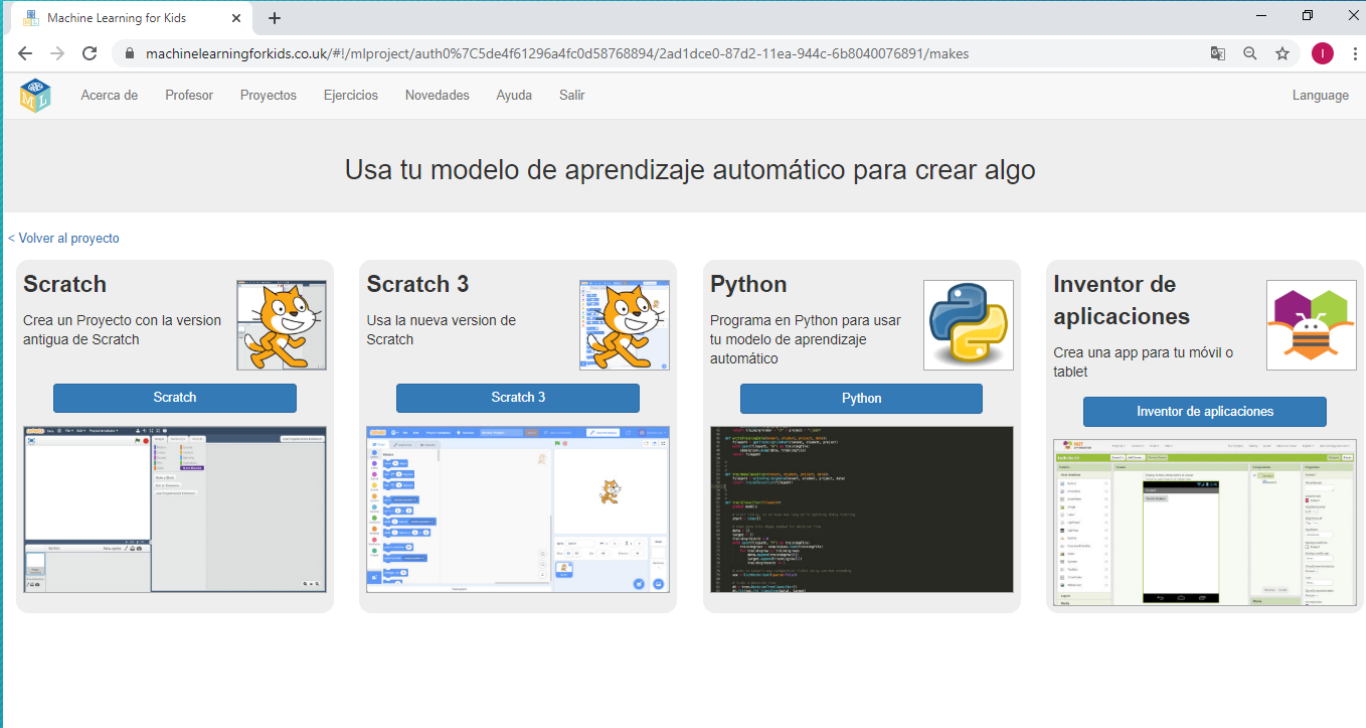
- Unha vez creadas as categorías e as imaxes que as describen, procedemos a crear o modelo de aprendizaxe (pode tardar minutos en crealo). Obteremos un resultado como o seguinte (no cal xa podemos probar con novas imaxes):



Machine Learning for Kids (ML4K). Crear aplicaciones.

Recoñe-
mento de
cores

- Podemos agora empregar o modelo para crear aplicacións en Python, Scratch ou App Inventor. No noso caso escollemos App Inventor.



The screenshot shows the Machine Learning for Kids website interface. At the top, there is a navigation menu with links for 'Acerca de', 'Profesor', 'Proyectos', 'Ejercicios', 'Novedades', 'Ayuda', and 'Salir'. Below the menu, a central heading reads 'Usa tu modelo de aprendizaje automático para crear algo'. Underneath, there are four main options, each with a title, a description, a logo, a button, and a preview image:

- Scratch**: 'Crea un Proyecto con la version antigua de Scratch'. Includes the Scratch logo and a preview of the Scratch IDE.
- Scratch 3**: 'Usa la nueva version de Scratch'. Includes the Scratch logo and a preview of the Scratch 3 IDE.
- Python**: 'Programa en Python para usar tu modelo de aprendizaje automático'. Includes the Python logo and a preview of a Python code editor.
- Inventor de aplicaciones**: 'Crea una app para tu móvil o tablet'. Includes the App Inventor logo and a preview of the App Inventor IDE.

Machine Learning for Kids (ML4K). Crear aplicaciones.

Recoñecemento de cores

- Temos que copiar a URL que identifica a extensión que empregará o noso modelo:

The screenshot shows a web browser window with the URL `machinelearningforkids.co.uk/#/mlproject/auth0%7C5de4f61296a4fc0d58768894/2ad1dce0-87d2-11ea-944c-6b8040076891/appinventor`. The page content includes:

- A button labeled "Open App Inventor".
- Text explaining that App Inventor is used to make mobile apps that run on Android phones and tablets, and that it runs in a web browser like Scratch.
- Text stating that App Inventor can be used with Machine Learning for Kids to create apps powered by artificial intelligence.
- A link to `github.com/kylecorry31/ML4K-AI-Extension` for more instructions.
- A note: "Support for App Inventor was created by Kyle Corry and Joe Mazzone".

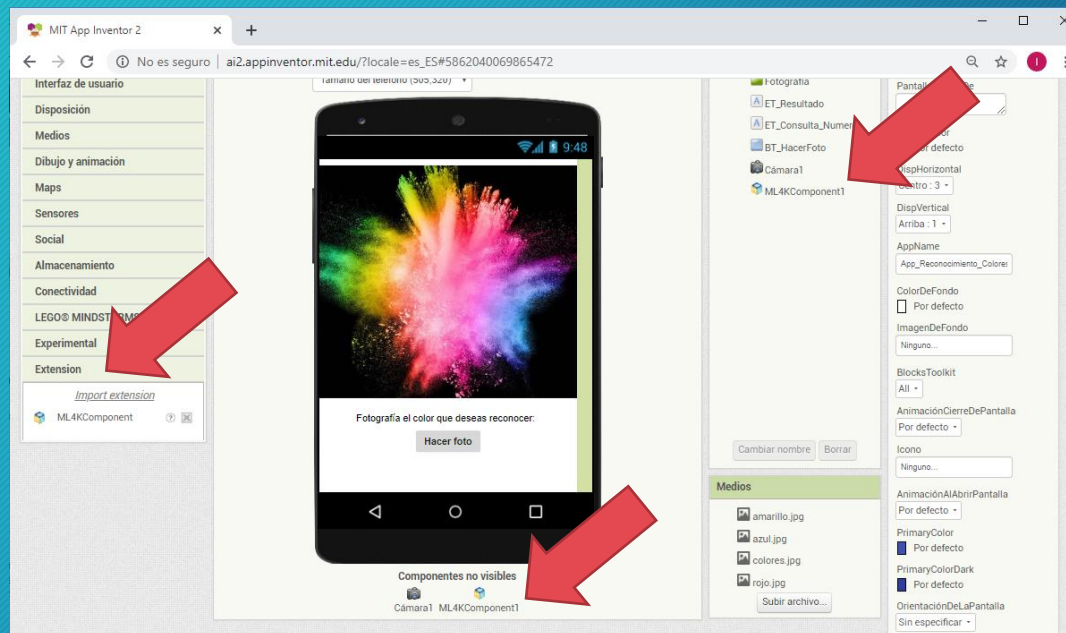
On the right side, there is a diagram of the App Inventor interface with a red arrow pointing to the "Import extension" button. The diagram includes the following steps:

1. Click on **Import extension** (indicated by a circled 1).
2. Click on **URL** (indicated by a circled 2).
3. Fill in the URL for your project: `https://machinelearningforkids.co.uk/api/appinventor/e686b050-87d2-11ea-944c-6b8040076891d253a7dc-2e40-4079-a1c6-de3ae47c4300/extension` (indicated by a circled 3).

APP Inventor importar extensión de ML4K

Recoñecemento de cores

- Pulsamos en importar extensión no noso proxecto de App Inventor, facemos clic en URL e copiamos (despois duns segundos aparecerá o novo elemento que podemos arrastrar ao noso proxecto):



Elementos da interface da aplicación

Recoñe-
mento de
cores

- Os elementos principais da app son:
 - Imaxe principal
 - Etiqueta que mostrará os resultados (ao iniciar a aplicación sen texto)
 - Etiqueta para mostrar un comentario de: “Fotografía a cor que desexas recoñecer”
 - Botón para consultar no modelo de aprendizaxe.
 - Modelo de ML4K
 - Imaxes en medios para telas dispoñibles despois (portada, amarelo, azul e vermello)


Elementos da interface da aplicação

Recoñe-
mento de
cores

Visor

Mostrar en el Visor los componentes ocultos

Tamaño del teléfono (506,320) ▼



Fotografía el color que deseas reconocer:

Hacer foto

Componentes no visibles

Cámara1 ML4KComponent1

Componentes

- Screen1
 - Fotografía
 - ET_Resultado
 - ET_Consulta_Numero
 - BT_HacerFoto
 - Cámara1
 - ML4KComponent1

Cambiar nombre Borrar

Medios

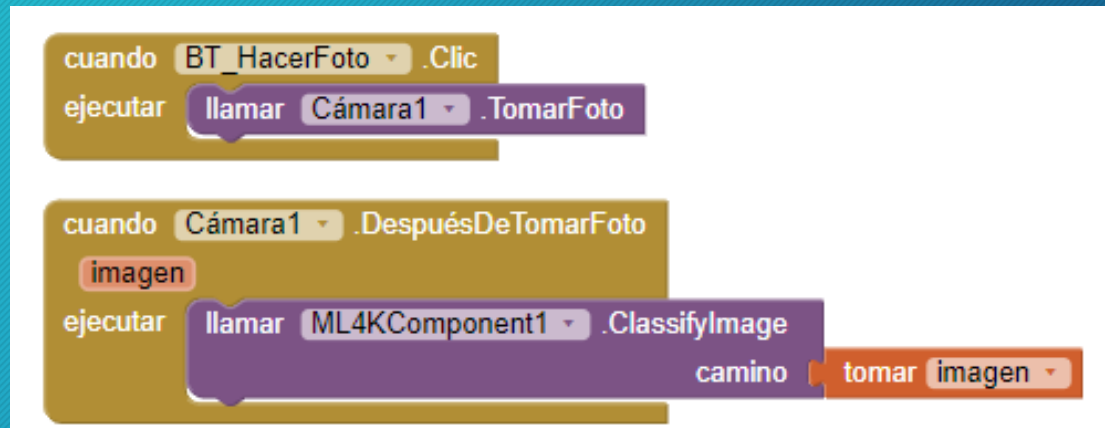
- amarillo.jpg
- azul.jpg
- colores.jpg
- rojo.jpg

Subir archivo...

Programación de bloques

Recoñece-
mento de
cores

- Ao facer clic abrírase a nosa aplicación da cámara para facer unha foto.
- Cando a foto xa foi feita, enviarase ao modelo de ML4K para que a clasifique.



Programación de bloques

Recoñecemento de cores

- Unha vez clasificada mostraremos o resultado na etiqueta de texto e coa imaxe da cor correspondente en función da % da confianza do resultado.

```
cuando ML4KComponent1 . GotClassification
  data classification confidence
  ejecutar
    si
      tomar confidence > 40
      entonces
        si
          comparar textos tomar classification = "Amarillo"
          entonces poner Fotografia . Foto como "amarillo.jpg"
          si no, si
            comparar textos tomar classification = "Azul"
            entonces poner Fotografia . Foto como "azul.jpg"
          sino
            poner Fotografia . Foto como "rojo.jpg"
        poner ET_Resultado . Texto como unir
          tomar classification
          "(Confianza del "
          tomar confidence
          "%)"
      sino
        poner ET_Resultado . Texto como "Color no reconocido :( "
        poner Fotografia . Foto como "colores.jpg"
```

Exemplos de resultados

Recoñe-
mento de
cores



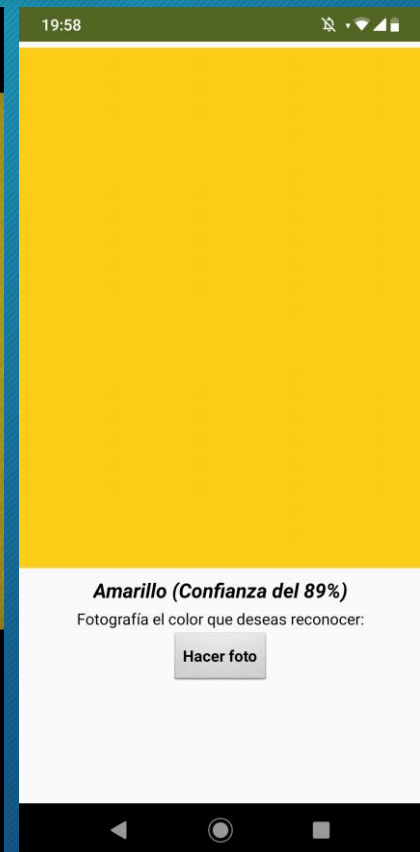
Foto



Resultado



Foto



Resultado

Aprendizaxe continuo (retroalimentación do modelo)



- Podemos facer nova funcionalidade na nosa app para que cando algo se clasifique mal, se incorpore ao noso modelo e a partir dese momento se empregue para clasificar (hai que ter en conta o tempo que se tarde en crear ese modelo)

