



GUIA PARA PRINCIPIANTES

Español

(Traducción Joaquim Martínez - INS PALAMOS)

Joaquim Martínez Rodríguez (INS Palamós)

Esta guía es uno de los muchos productos creados por el proyecto Erasmus+ "Transmedia: a new narrative that involves communication and education towards the future classroom". Código: 2014-1-ES01-KA201-004478. De septiembre de 2014 a Agosto de 2016, el trabajo compartido de las siguientes Instituciones lo hizo posible:

COORDINADOR DEL PROYECTO: Josep Antoni Blasco Fernández (jblasco@xtec.cat)

2

- **INSTITUCIÓN COORDINADORA :** INSTITUT DE PALAMÓS (Cataluña, España)
<http://inspalamos.cat/>
Coordinadores: Joaquim Martínez Rodríguez, Magda Soler Pedrós & Josep Ciberta i Tirado

INSTITUCIONES ASOCIADAS:

- CENTRO DE FORMAÇÃO DE ASSOCIAÇÃO DE ESCOLAS CENTRO-OESTE (Caldas da Rainha, Portugal) <http://www.cfaecentro-oeste.pt/>
Coordinadores: Nicolau Joao Gonçalves Borges & Carla de Jesus
- DIMOTIKO SCHOLEIO DROSIA K.A' (Larnaca, Cyprus)
<http://dim-drosia-ka-lar.schools.ac.cy/>
Coordinadora: Stella Eliadou
- COLEGIUL NATIONAL 'B.P. HASDEU' (Buzau, Romania)
<http://bphasdeu.ro/>
Coordinadora: Viorica Raicu
- PAGKYPRIO LYKEIO LARNACA (Larnaca, Cyprus)
<http://lyk-pagkyprio-lar.schools.ac.cy/>
Coordinadoras: Filitsa Filipou Masonou & Konstantia Mala
- ISTITUTO TECNICO INDUSTRIALE STATALE 'GALILEO GALILEI' (Livorno, Italy)
<http://www.galileilivorno.gov.it/>
Coordinadores: Federico Agen & Maddalena Degan

Indice

3

Capítulo 1.	Introducción	4
Capítulo 2.	Uso de la interfaz	4
Capítulo 3.	Diseñador de App Inventor	5
Capítulo 4.	Editor de bloques de App Inventor	7
Capítulo 5.	Como probar una App	9
Capítulo 6.	Como instalar una App	11

Capítulo 1. Introducción

App Inventor, <http://aiz.appinventor.mit.edu/>, ofrece todo lo necesario para aprender a desarrollar tus propias aplicaciones para Android. Es una herramienta basada en la nube, lo que permite desarrollar las aplicaciones en el propio navegador de Internet. Solo es necesario disponer de una cuenta de Google para disponer de una cuenta de App Inventor en la cual crear, almacenar y exportar les tus apps.

4

La herramienta ha sido desarrollada por el MIT y utiliza el mismo lenguaje basado en bloques usado por Scratch <https://scratch.mit.edu>.

Capítulo 2. Uso de la interfaz

Una aplicación Android o app es una aplicación gráfica. Para desarrollarla se necesita diseñar la interfaz gráfica y el código que ejecutará. Por ello App Inventor se divide en el Diseñador, para diseñar la interfaz, y los Bloques, para editar el código de la aplicación construido con bloques.



Capítulo 3. Diseñador de App Inventor



5

Con el diseñador se pueden añadir componentes gráficos que verá el usuario dentro de la pantalla o bien otro tipo de componentes fuera de ella.

Interfaz de usuario contiene objetos como etiquetas para escribir texto en la pantalla, cajas de texto con texto editable por el usuario o botones para ejecutar código:

<http://ai2.appinventor.mit.edu/reference/components/userinterface.html>

Diseño contiene objetos que permiten organizar los objetos gráficos en pantalla de forma horizontal, vertical o en una tabla:

<http://ai2.appinventor.mit.edu/reference/components/layout.html>

Media contiene objetos que pueden aparecer en las pantallas o no y que permiten al usuario reproducir música y vídeos:

<http://ai2.appinventor.mit.edu/reference/components/media.html>

Dibujo y animación contiene componentes como el lienzo o el spriteimage que permiten mover y editar imágenes de forma sencilla:

<http://ai2.appinventor.mit.edu/reference/components/animation.html>

Sensores permite recoger los valores de los diferentes sensores del dispositivo:

<http://ai2.appinventor.mit.edu/reference/components/animation.html>

El reloj de esta sección permite ejecutar instrucciones cada cierto intervalo de tiempo.

Social contiene componentes que permiten llamar por teléfono, enviar mensajes, compartir datos con otras aplicaciones del mismo dispositivo o crear aplicaciones clientes de Twitter:

<http://ai2.appinventor.mit.edu/reference/components/social.html>

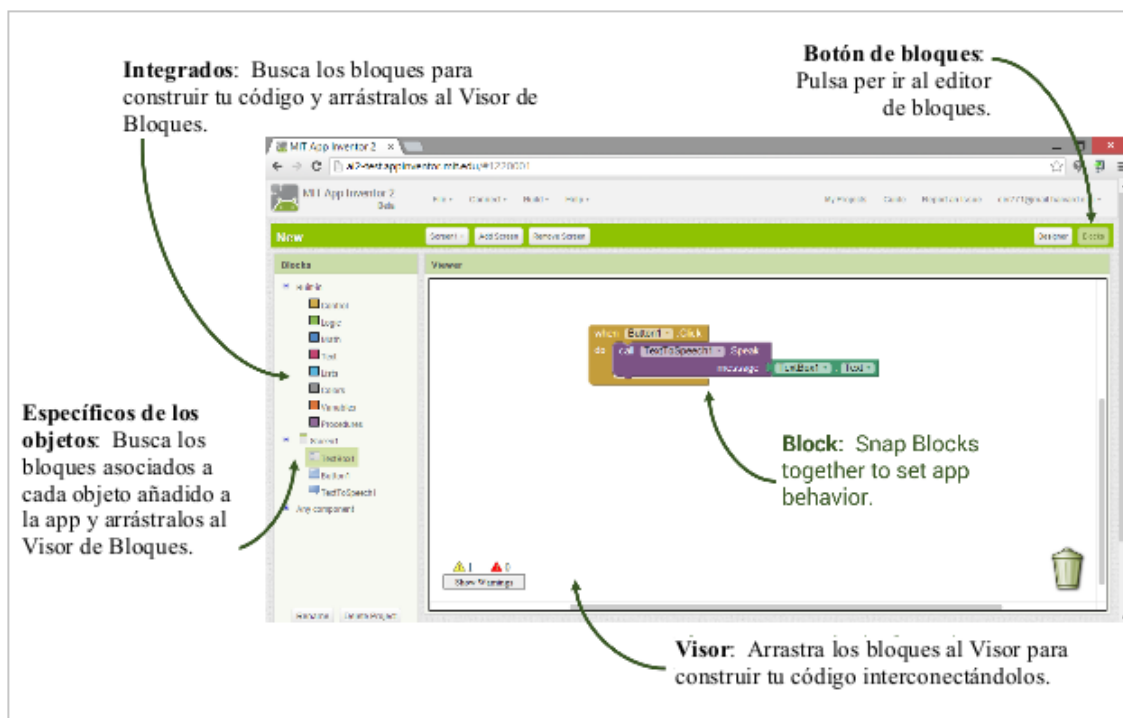
Conectivitat contiene los componentes necesarios para conectar un cliente con un servidor bluetooth o bien pedir páginas web:

<http://ai2.appinventor.mit.edu/reference/components/connectivity.html>

Lego Mindstorms para interactuar con este tipo de robots:

<http://ai2.appinventor.mit.edu/reference/components/legomindstorms.html>

Capítulo 4. Editor de bloque de App Inventor



7

El editor de bloques es donde se escribe el código de programación de la app. En este caso no es un lenguaje compuesto por sentencias de texto sino que las sentencias están representadas por bloques y el código es se construye encajando estos bloques de manera similar a como se hace un rompecabezas.

Integrados contiene sentencias, operaciones y valores del lenguaje:

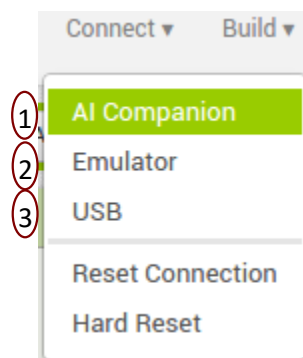
- **Bloques de control:** condicionales o bucles:
<http://appinventor.mit.edu/explore/ai2/support/blocks/control.html>
- **Bloques lógicos:** valores y operaciones lógicas:
<http://appinventor.mit.edu/explore/ai2/support/blocks/logic.html>

- **Bloques de matemáticas:** valores, operaciones y funciones para trabajar con valores numéricos:
<http://appinventor.mit.edu/explore/ai2/support/blocks/math.html>
- **Bloques para texto:** valores y funciones para trabajar con cadenas de texto:
<http://appinventor.mit.edu/explore/ai2/support/blocks/text.html>
- **Bloques para listas:** funciones para trabajar con listas. Una lista es una cadena compuesta de valores de cualquier tipo de datos:
<http://appinventor.mit.edu/explore/ai2/support/blocks/lists.html>
- **Colores:** para trabajar con colores:
<http://appinventor.mit.edu/explore/ai2/support/blocks/colors.html>
- **Variables:** para guardar o leer el valor de una variable:
<http://appinventor.mit.edu/explore/ai2/support/blocks/variables.html>
- **Procedimientos:** para encapsular código dentro de un procedimiento que puede ser llamado y ejecutado varias veces:
<http://appinventor.mit.edu/explore/ai2/support/blocks/procedures.html>

Bajo el nombre de la pantalla se encuentran todos los objetos añadidos con el Diseñador. Cuando se selecciona uno de estos objetos aparecen todos los bloques relacionados con él. Por ejemplo cuando se selecciona un botón se pueden ejecutar instrucciones o cuando se selecciona un sensor se puede leer el valor del sensor.

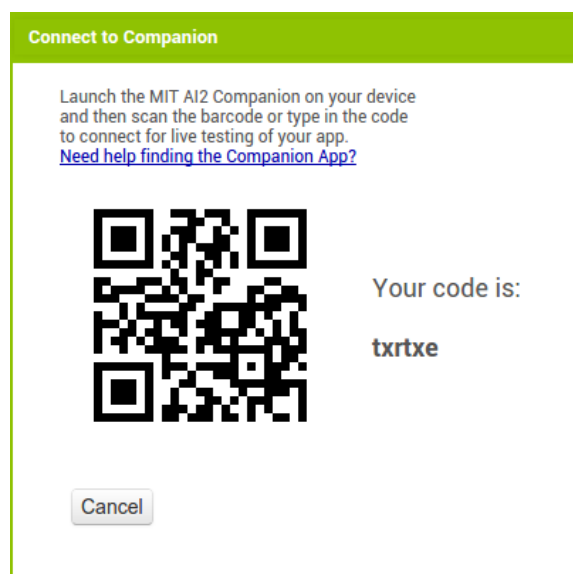
Capítulo 5. Como probar una App

Existen tres maneras diferentes de probar una app que se encuentra en desarrollo.

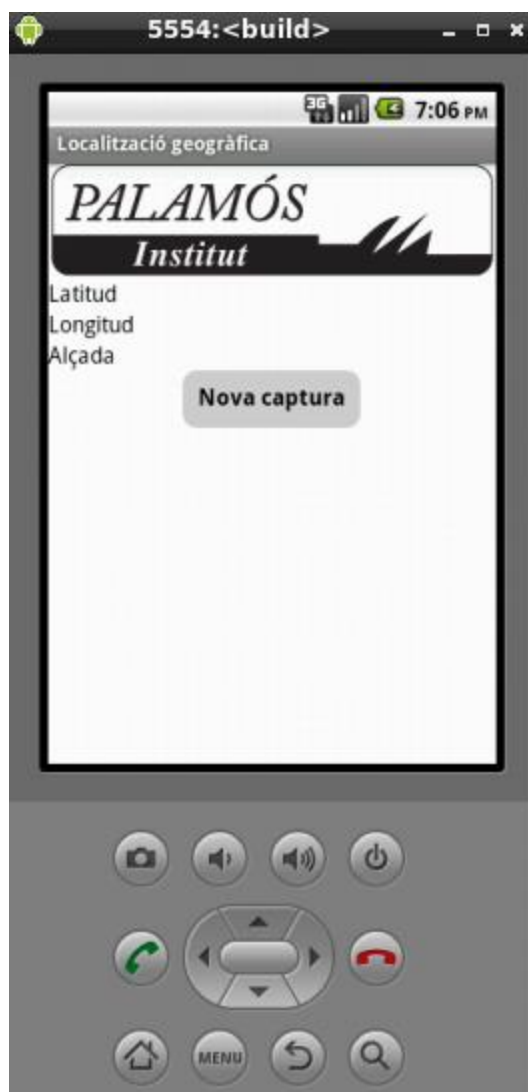


9

1. La app MIT AI Companion se puede instalar en un dispositivo Android. Cuando se ejecuta esta app es necesario introducir el código o escanear el código QR que aparece al seleccionar "Conectar" y "AI Companion".

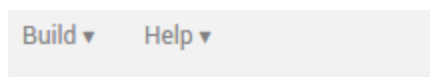


2. App Inventor incluye un Emulador Android. Las instrucciones para instalar y ejecutar el emulador se encuentran aquí: <http://appinventor.mit.edu/explore/ai2/setup-emulator.html>. Seleccionando "Conectar" y "Emulador" la app se ejecutará dentro del emulador.



3. También se puede conectar un dispositivo con un cable USB y seleccionar "Conectar" y "USB".

Capítulo 6. Como instalar una App

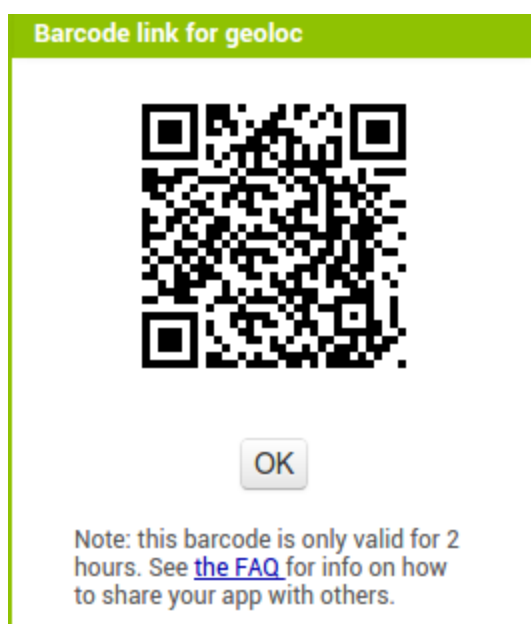


- 1 App (provide QR code for .apk)
- 2 App (save .apk to my computer)

11

Es posible instalar una app de dos formas distintas:

1. Escaneando con el dispositivo el código QR generado con "Generar" la app se descargará y se instalará automáticamente.



2. Des de "Generar" también se puede crear un archivo de instalación, apk, que se puede copiar e instalar en cualquier dispositivo Android.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International.

Permissions beyond the scope of this license may be available at educationtransmedia@gmail.com

Basado en <http://appinventor.mit.edu>