

## Descubriendo AppInventor

¿ Qué aprenderé ?.....	4
Instalando java.....	5
Instalando AppInventor.....	6
Creando nuestra primera aplicación.....	7
Comenzando un proyecto.....	11
Crear una etiqueta.....	11
Agregar una botón.....	16
Definir el Comportamiento de los botones.....	19
Agregar un ronroneo.....	20
Agitar el teléfono.....	21
Empaquetar la aplicación para su descarga.....	25
Configurando el emulador.....	27
Variaciones.....	30

Android es un sistema operativo para teléfonos móviles y tablets, de la misma forma que Microsoft Windows es un sistema operativo para los PCs. El sistema Android es mantenido por Google y viene con un conjunto de características muy útiles.

Las características del sistema Android incluyen Google Search y Google Maps, lo que significa que puedes encontrar información fácilmente en la web y encontrar direcciones desde tu móvil, como si estuvieras en tu ordenador. Hay un gran rango de aplicaciones disponibles desde Google Play Store.

Android es un sistema operativo de código abierto, construido con Linux, también de código abierto, lo que significa que puede ser fácilmente ampliable para incorporar nuevas tecnologías tan pronto como aparezcan. La plataforma Android continuara desarrollándose mientras la comunidad de desarrolladores trabajen juntos para construir aplicaciones de móviles innovadoras y tu puedes ser parte de esta innovación excitante con AppInventor para Android.

AppInventor te permite desarrollar aplicaciones para teléfonos Android usando un navegador

web y bien un emulador o un teléfono conectado, sin escribir una sola línea de código.

Es una herramienta basada en la web, desarrollada por los laboratorios Google en la que los servidores de AppInventor almacenan tu trabajo y te ayudan a mantener tus proyectos. Cuando la aplicación esta finalizada te permite “empaquetarla” para producir un fichero **.apk** de Android que puede ser compartida e instalada en un teléfono Android, como otras aplicaciones Android. AppInventor es soportado por un amplio rango de sistemas operativos y navegadores web, con unas especificaciones mínimas:

Sistemas operativos requeridos:

- **Windows:** WindowsXP, Windows Vista, Windows 7+.
- **Linux:** Ubuntu 8+, Debian 5+.
- **Mac:** Mac OS X 10.5+

Navegadores requeridos:

- **Internet Explorer:** 7.0+.
- **Mozilla Firefox:** 3.6+.
- **Google Chrome:** 4.0+.
- **Apple Safari:** 5.0+.

Adicionalmente AppInventor necesita dos elementos de software para ser instalados en tu computador:

- **Java:** Para ofrecer librerías de software para el entorno de desarrollo de AppInventor basado en la web.
- **Appinventor Setup:** Para ofrecer soporte en el lado del cliente para el entorno de desarrollo de AppInventor basado en la web.

A continuación veremos ambos procesos de instalación.

## ¿ Qué aprenderé ?

- Instalar Java

- Instalar AppInventor.
- Desarrollar una aplicación.
- Añadir componentes y comportamientos.
- Preparar al dispositivo Android y configurar el Emulador.
- Administrar Proyectos.

## Instalando java

Para usar el entorno de desarrollo de AppInventor tu computador necesitará las librerías de soporte de **Java**. Sigue estos pasos para instalar fácilmente Java en tu sistema:

- Abre el navegador web y accede a <http://www.java.com>
- Haz clic en el botón **“Descarga gratuita de Java”** para comenzar el proceso de Instalación.

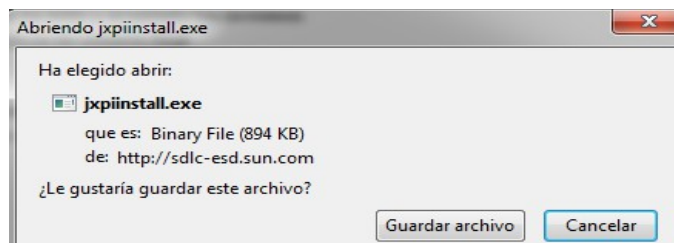
### Descargar Java para Windows

Recomendado Version 7 Update 45 (Tamaño de archivo: 893 KB)

**Aceptar e iniciar descarga gratuita**

Al descargar Java, confirma que ha leído y aceptado los términos del [acuerdo de licencia de usuario final](#)

Aparecerá esta ventana.



Haz clic en **Guardar archivo**.

- Accede a la carpeta Descargas y ejecuta el archivo **jxpiinstall.exe**.

Aparecerá esta ventana:



- Haz clic en el botón **Instalar** y acaba el proceso de Instalación.

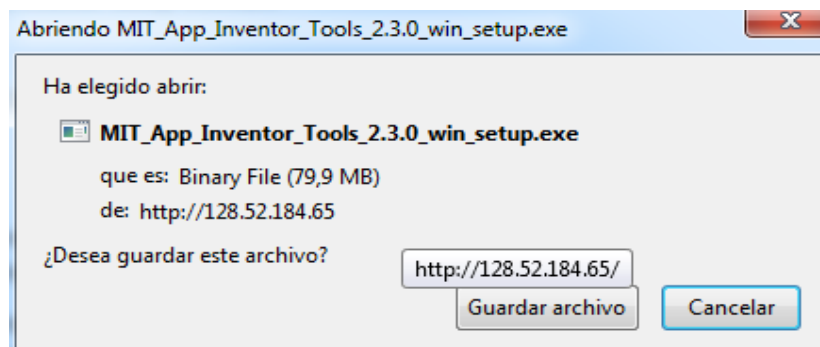
## Instalando AppInventor

Una vez hemos realizado el paso anterior, el ordenador necesita ofrecer el soporte del lado del cliente para el entorno de desarrollo de **AppInventor**. Sigue estos pasos para instalarlo fácilmente en tu sistema.

- Abre el navegador y teclea la localización de la instalación de **AppInventor2**

<http://appinventor.mit.edu/explore/ai2/windows.html>

Desde aquí vamos a descargar y ejecutar el instalador



Haz clic en el botón **Guardar Archivo**.

- Accede a la carpeta Descargas y ejecuta el fichero de instalación de AppInventor **MIT\_App\_Inventor\_Tools\_2.3.0\_win\_setup**



Haz clic en el botón **Next**. Acepta las opciones por defecto que te ofrezca el proceso de instalación.

## Creando nuestra primera aplicación

La aplicación que hoy iniciamos se llama **AppInventor2**, el enlace a día de hoy es:

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

Para poder crear aplicaciones necesitamos tener una cuenta de **google** ya creada, en caso contrario deberemos crearla. Si es así veremos esta pantalla.



La aplicación MIT AppInventor Experimental solicita autorización para acceder a su cuenta de Google.

Seleccione la cuenta que desea utilizar.

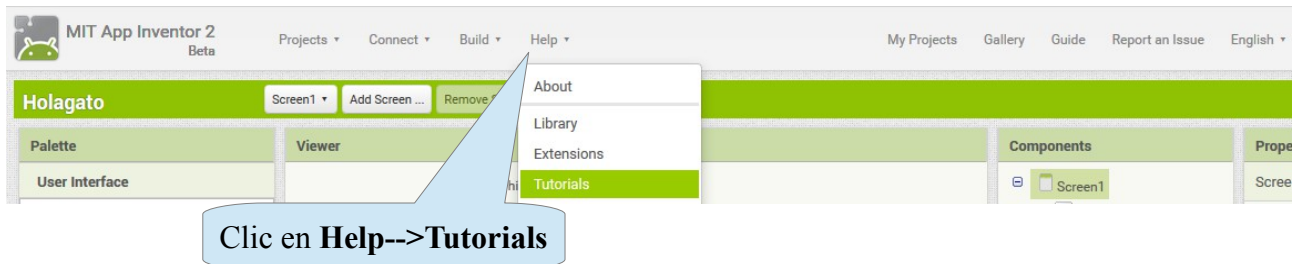
sdestruels@gmail.com

Google no está afiliado al contenido de MIT AppInventor Experimental ni a sus propietarios. Si accedes a tu cuenta, Google compartirá tu dirección de correo electrónico con MIT AppInventor Experimental, pero no compartirá ni la contraseña ni los datos personales.

[Accede a otra cuenta](#)

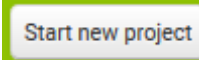
Recordar esta aprobación durante los próximos 30 días

Tras hacer clic en **Permitir**, veremos esta otra pantalla.



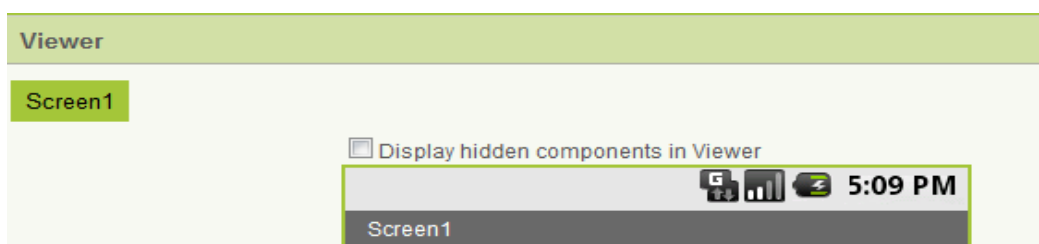
Haciendo clic en **Tutorials** puedes acceder a una serie de aplicaciones que te van introduciendo en el uso de APPInventor2.

Vamos a crear una pequeña aplicación.

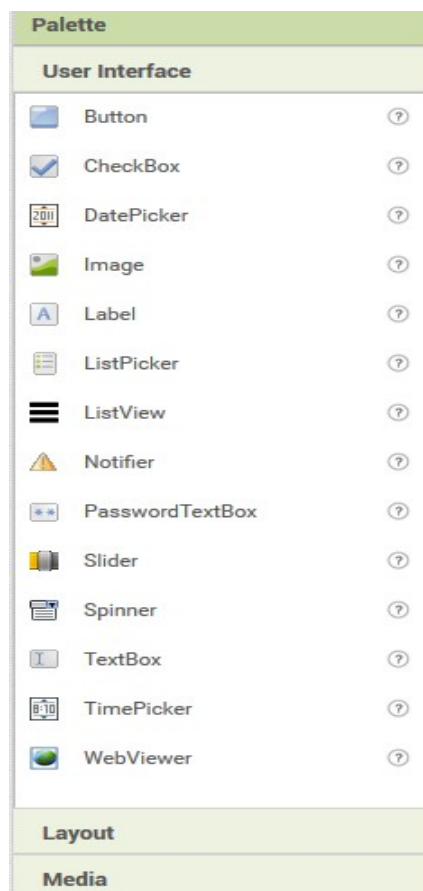
Haz clic en Projects y elegiremos  y dale como nombre “**holagato**”.

La primera herramienta que vas a utilizar es el diseñador de Componentes (**Component Designer**), que se divide en varias partes:

En la parte central hay una zona en blanco que se llama visor(**Viewer**), hasta aquí arrastraremos los controladores de la aplicación. De todas formas, para comprobar el verdadero aspecto que tendrán cuando estén en funcionamiento deberemos abrirla en el teléfono o visualizarla desde el emulador que viene con AppInventor.

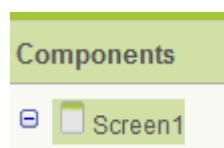


A la izquierda se encuentra la paleta (**Palette**), que lista todos los componentes disponibles.

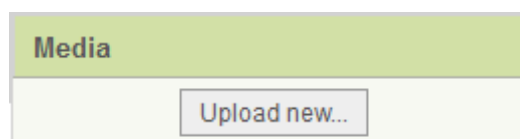


A la derecha del visor se halla la lista de componentes (**Components**). En ella se muestra todos los utilizados en el proyecto.

Cualquiera que se arrastre hasta el visor se añadirá a esta relación. De momento sólo hay uno: **Screen 1**, que representa la pantalla del teléfono.



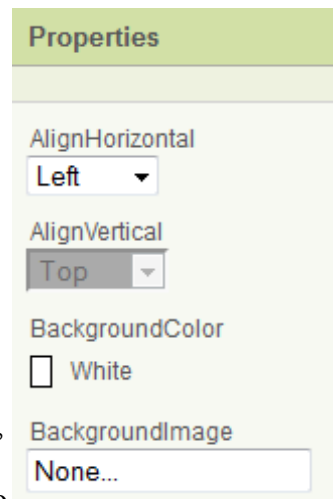
Debajo de la lista de componentes tenemos un área donde se verán todos los contenidos multimedia (imágenes y sonido) que se empleen en el proyecto (**Media**). Éste todavía no tiene ninguno.



En la parte derecha de la pantalla hay una sección que muestra las propiedades de los componentes (**Properties**). Cuando hacemos clic sobre uno situado en *Viewer*, sus propiedades aparecen aquí. No son más que detalles de los componentes que podemos modificar.

Ahora muestra las propiedades de la pantalla del teléfono (*Screen1*). Aparece el color de fondo (**backgroundcolor**), la imagen de fondo (**backgroundimage**) y el título (**title**).

En la **aplicación “holagato”** trabajaremos con dos componentes visibles: **Label**, que mostrará el texto “*Toca a Kitty*” y **Button**, que tendrá una imagen de un gato. También existirá uno no visible (**sound**) que reproducirá los sonidos del gato y un **Accelerometer** que detectará los movimientos que se le hagan al teléfono.





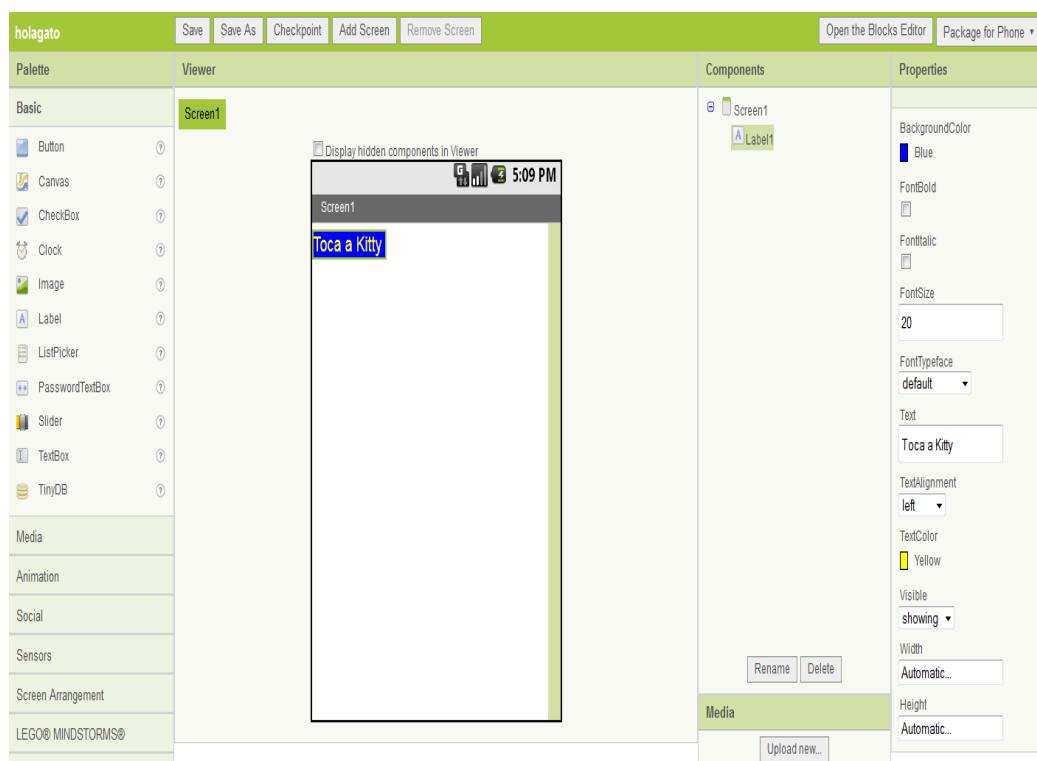
## Comenzando un proyecto

### Crear una etiqueta

El primer componente a añadir es el **Label**:

1. En **Palette** (Paleta) haga clic sobre **Label**, dentro de la sección Basic y arrástrelo sobre el Visor.
2. En sus propiedades asociadas a la derecha hay una llamada **Text**. Sustituye el texto que haya por “*Toca a Kitty*”. Y presione Intro. El nuevo mensaje aparecerá en el visor.
3. Pulse en el cuadro blanco que se encuentra en la propiedad de **BackgroundColor**. Selecciona el *azul (blue)*. Cambia el tono de la fuente desde la propiedad **TextColor** y selecciona *Yellow*. Por último, modifica el tamaño de la fuente **FontSize** al valor *20*.

**Content Designer** mostrará el contenido de la Figura 1.1:



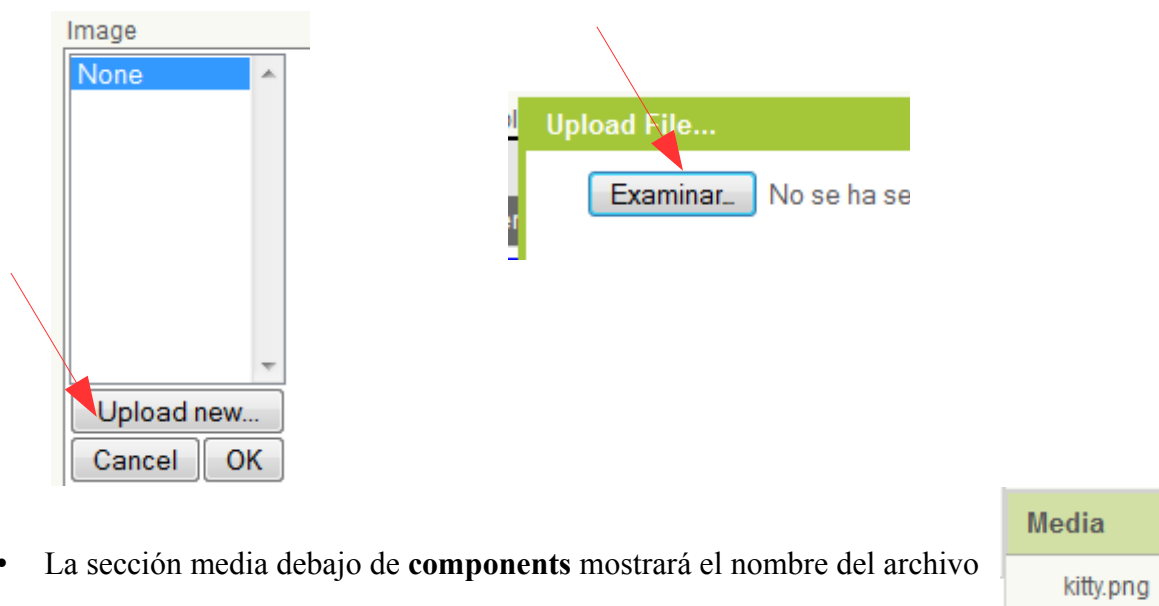
**Figura 1.1.** La aplicación ya tiene su primera etiqueta.

## Agregar una botón

Vamos a utilizar un componente **Button** para incorporar la foto de un gato a nuestra aplicación. Empezaremos creando un botón normal y después cambiaremos su imagen de fondo por la del gato.

En **Palette** (Paleta) haz clic sobre **Button**, y arrástralo sobre el Visor, colocándolo debajo de la etiqueta (Label). Ya tenemos el botón que emplearemos para lanzar algún efecto sonoro cuando se toque el teléfono. Ahora colocaremos la imagen del gato y quitaremos el texto que aparece en la pantalla:

- Localizamos la imagen “*kitty.png*” y el sonido “*meow.mp3*”
- Vamos a las propiedades del button. En la propiedad **image** (imagen) que es *None*, buscamos “*kitty.png*”. Y pinchamos en OK.



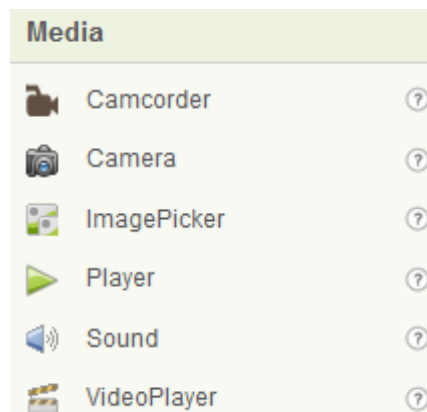
- La sección media debajo de **components** mostrará el nombre del archivo multimedia que acabamos de incorporar a la aplicación.
- Si te fijas, encima de la fotografía aparece el texto *Text for button 1*. Si no queremos que salga, modificamos el contenido de la propiedad **Text** del componente **button1**. Lo cambiamos por un mensaje que diga “*Toca al gato*” o, lo dejamos en blanco.

Ahora el aspecto de la pantalla del diseñador debería ser similar al siguiente:

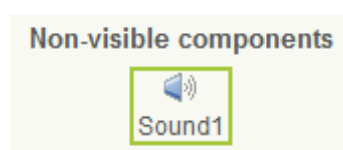


Nuestra aplicación hará que el gato maúlle cuando toquemos el botón. Para ello, agregaremos el sonido y programaremos el comportamiento del botón para que lo reproduzca al tocarlo.

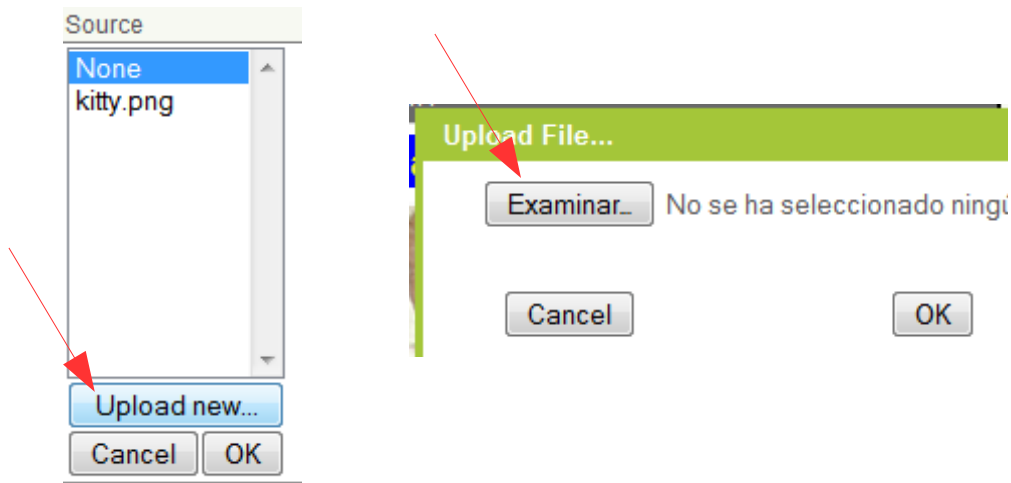
- 1) Dentro de la **paleta**, elegimos sección **Media**. Arrastra el componente **Sound** hasta la ventana del visor, no importa donde lo sueltes porque aparecerá en el área que está debajo del visor.



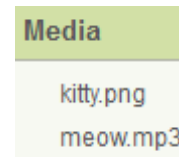
*Es un componente no visible* como te indica. Estos componentes invisibles son los objetos que hacen cosas pero que no aparecen en la interfaz del usuario.



2) Haz clic en **sound1** para ver sus propiedades. Localiza y agrega el sonido “*meow.mp3*”.



Una vez hecho se habrá añadido en la sección **Media**, debajo de **Components**.



### Definir el Comportamiento de los botones

Para ello trabajaremos desde el **Editor de Bloques**. Ábrelo desde la parte superior derecha haciendo clic sobre el botón **Blocks**



Aparecerá esta ventana:



En la parte superior izquierda tenemos entre otras cosas los objetos que habíamos definido. Haz clic sobre **button1**. Se mostrará todos los bloques disponibles para definir el comportamiento del botón.

Pulsa sobre el bloque **Button1.Click** y arrástralo hasta la zona de trabajo. Observa que el tamaño de la palabra When es más pequeño que el de Button1.click. Los bloques que la incluyen reciben el nombre de controladores de eventos. Especifican qué debe hacer un componente cuando tenga lugar un evento.



En nuestro ejemplo, queremos que tenga lugar cuando el usuario toque el gatito (que en realidad es un botón). A continuación, añadiremos unos bloques para especificar qué ocurrirá como respuesta a este suceso.

Pulsamos sobre el elemento **sound1**. Así, abriremos el cajón donde están los componentes de sonido y arrastraremos un bloque llamado **call Sound1.Play** hasta el área de trabajo.

Seguramente habrás notado que la forma del bloque call Sound1.Play coincide con la muesca del bloque Button1.Click. App Inventor se ha diseñado para que sólo ciertos bloques puedan funcionar ente sí. En este caso, los que incluyen la palabra call son los responsables de que los componentes hagan algo. Unimos los dos bloques y App Inventor emitirá un sonido.

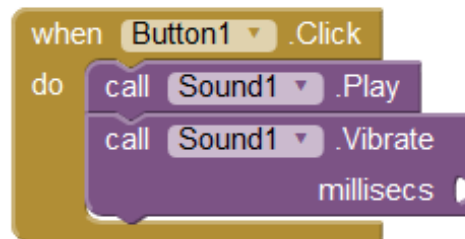


## Agregar un ronroneo

Ahora vamos a hacer que nuestro gato ronronee y maúlle cuando toquemos el botón. Simularemos el sonido haciendo que el teléfono vibre.

Bastará con añadir un segundo comportamiento al bloque que controla el botón, el cual se halla en **Blocks** :

1. Ves a **Blocks** y haz clic sobre **Sound1**.
2. Arrastra el elemento **call Sound1.Vibrate** hasta el área de trabajo, debajo del **call Sound1.Play** que teníamos.

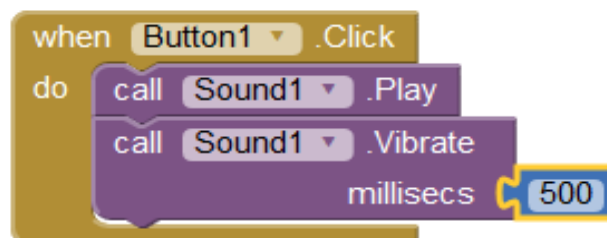


3. El nuevo bloque tiene el texto **millisecs**(milisegundos) en la parte superior derecha. Además posee una ranura que indica que aún podemos agregar algo más para definir su comportamiento. En este caso, debemos añadir al bloque **Vibrate** cuánto tiempo debe hacer que el teléfono vibre. Lo especificamos en milisegundos, así que para que el teléfono se agite medio segundo. Colocaremos un valor de 500 ms. Para ello, recurriremos a un bloque.

4. Haz clic sobre **Built-In**. Pulsa sobre el **Math**. Aparecerá



5. Cámbialo por **500**.



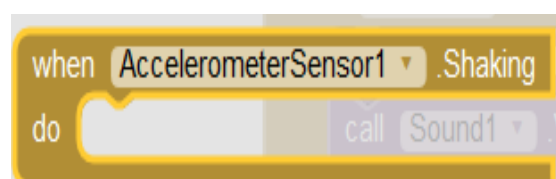
## Agitar el teléfono

Ahora vamos a añadir un elemento final que nos permitirá conocer otra de las propiedades tan interesantes de los teléfonos Android: vamos a hacer que el gato maúlle cuando agitemos el teléfono. Utilizaremos un componente llamado **AccelerometerSensor** capaz de detectar cuando el aparato se mueve.

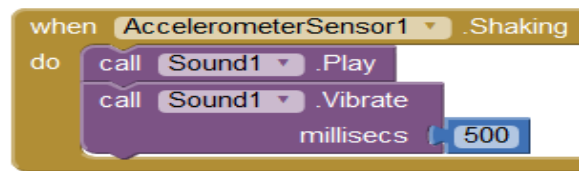
1. En la pantalla de la **Paleta**, haz clic sobre la paleta **Sensors**. Arrastre **Accelerometer Sensor** sobre el visor. No te preocupes por dónde lo sueltas porque, igual que ocurre con los componentes invisibles, AppInventor lo colocará dentro de la sección *Non-Visible components*.



2. Vamos a hacer que el evento encargado de manejar el movimiento del teléfono sea independiente del botón. Es decir, necesitaremos un nuevo controlador de eventos. Vamos a **Blocks**, haz clic en **AccelerometerSensor1**. Arrastra **AccelerometerSensor1.Shaking**.



- Al igual que hiciste con el botón1, arrastra los mismos bloques anteriores y colócalos dentro de **AccelerometerSensor1.Shaking**.



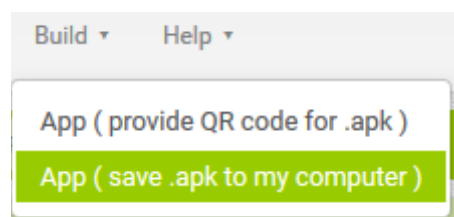
Cierra ahora el **Blocks**.

## Empaquetar la aplicación para su descarga.

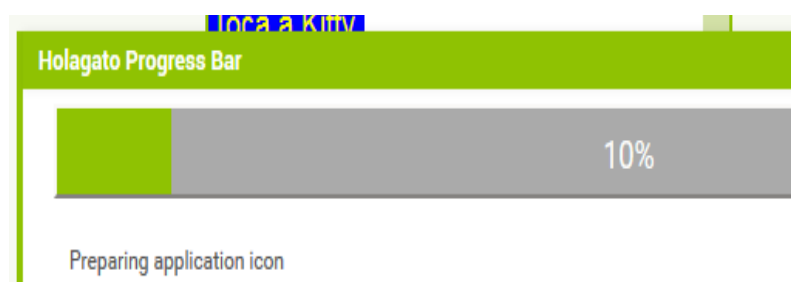
Cuando terminemos una aplicación, tendremos que empaquetarla e instalarla para poder utilizarla en cualquier teléfono, aún cuando no esté conectado a un ordenador.

En primer lugar debemos asegurarnos que nuestro móvil permite descargar aplicaciones de internet de otras páginas aparte de **Android Market**. Para ello, accedemos a *Ajustes* → *Aplicaciones de nuestro teléfono* y activamos la casilla de verificación *Orígenes desconocidos*, si no lo estuviera ya, claro.

Luego volvemos a la pantalla de **Components Designer** y hacemos clic sobre **Build**, elige **save .apk**

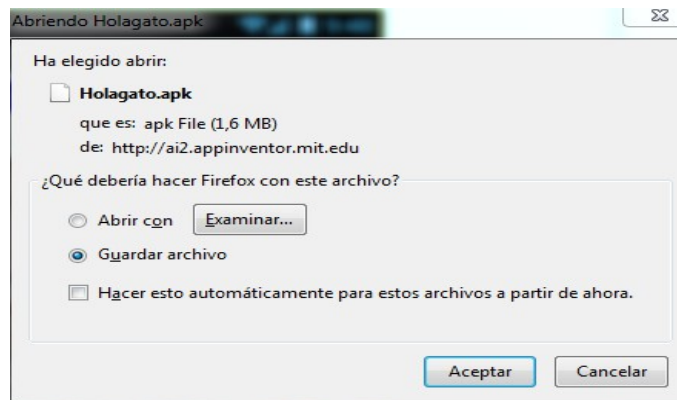


Aparecerá una barra indicando la descarga.





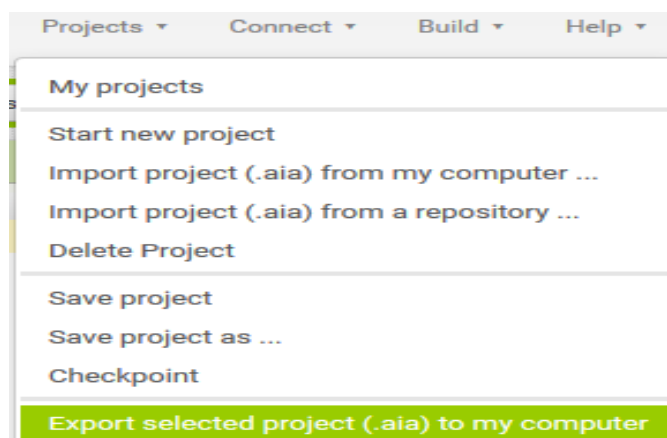
Aparecerá esta ventana, haz clic en **Guardar archivo**.



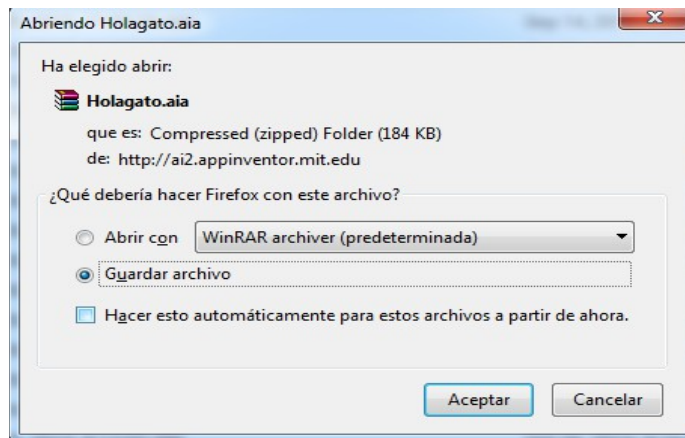
Es importante comprender que la aplicación empaquetada es independiente del proyecto que tenemos en AppInventor. Si realizamos cambios en nuestro proyecto y queremos instalarla de nuevo en nuestro teléfono, tendremos que eliminar la antigua, empaquetar la nueva, descargarla e instalarla en el móvil.

Además de compartir tu aplicación a través del fichero **.apk** que se ha generado, también podemos compartir el código fuente (los bloques) del programa para otros desarrolladores de AppInventor.

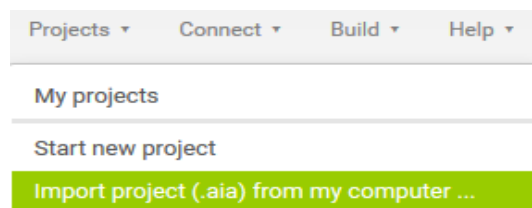
Para ello, haga clic en **My Projects**, seleccione la casilla de verificación correspondiente al proyecto que desea compartir. Pulsa ahora en **Projects** y elige **Export selected project (.aia) to my computer**.



Aparecerá esta ventana, para guardar el fichero.

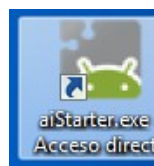


Esta vez se creará un fichero **.aia** en tu pc. Ahora puedes mandarlo por correo a tus amigos, ellos al recibirlo deberán de utilizar la opción

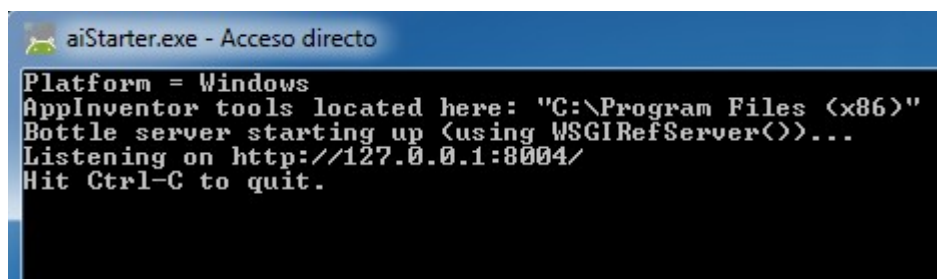


## Configurando el emulador

Lo primero de todo es que se necesita lanzar un programa llamado aistarter. Este fichero se encuentra en la carpeta de instalacion de AppInventor. Crea un acceso directo para facilitarte esta tediosa labor.



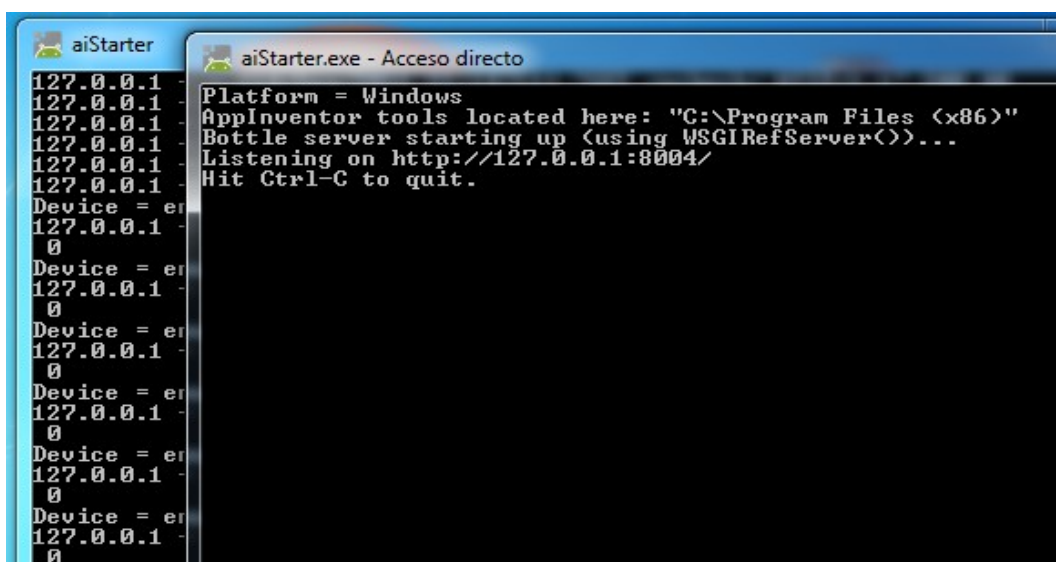
Ejecutalo desde el acceso directo. Aparecerá esta ventana:



Ahora haz clic sobre **Connect y Emulator**. Aparecerá esta ventana, representando al móvil.



Tendrás también dos ventanas de este estilo.

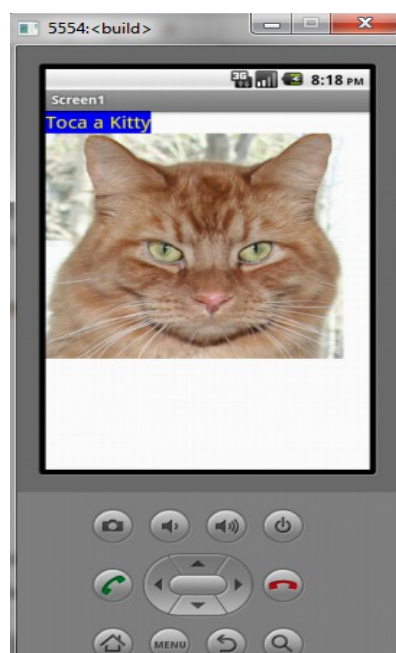


Obtendremos esta ventana, indicando que el programa se esta cargando en el emulador.



Es posible que aparezcan mensajes de actualizaciones nuevas durante este proceso, en ese caso, deberemos seguir las instrucciones dadas.

Observa ahora como la pantalla del emulador se ha cargado con la aplicación de **Hologato**.



## Variaciones

Notarás que cuando giras el teléfono el maullido suena un poco extraño, como si tuviese algún tipo de eco. Esto se debe a que el sensor de movimiento está disparando el mismo evento varias veces por segundo, por lo que los maullidos se solapan. Si te fijas en el componente **Sound** que se encuentra en **Component Designer**, verás que hay una propiedad llamada *minimum Interval*, la cual determina el tiempo que debe pasar entre la reproducción de un sonido y el siguiente. Generalmente, su valor es de 500 ms, que es menos de lo que tarda un maullido. Debes modificar este espacio para evitar que los maullidos se solapen.

Si no cierras la aplicación en tu teléfono cuando se mueva el móvil el gato cantará...Ten cuidado!!!!.