



## **BÁSICOS APP INVENTOR**

# **Manual de Introducción a AppInventor**





## Contenido

1. ¿Qué es AppInventor? .....	2
2. ¿Qué tipo de aplicaciones pueden crearse con AppInventor? .....	3
3. ¿Cómo se construye una aplicación en AppInventor? .....	4
3.1. Componentes .....	5
3.2. Comportamiento .....	6
3.2.1. Eventos .....	9
3.2.2. Manejadores de Eventos .....	13
4. Requisitos Previos .....	15
5. El Entorno de Desarrollo .....	17
5.1. Diseñando los Componentes .....	19
5.2. Añadiendo Comportamientos .....	21
6. Referencias .....	24



## 1. ¿Qué es AppInventor?

AppInventor es una aplicación web de Google que permite crear aplicaciones para el sistema operativo de dispositivos móviles Android. Utiliza un editor Drag and Drop (Arrastrar y soltar) para la generación de interfaces gráficas y un sistema de bloques para gestionar el comportamiento de la aplicación. Los proyectos generados a través de esta herramienta se almacenan automáticamente en los servidores de App Inventor, permitiendo llevar en todo momento un seguimiento y control de todo nuestro trabajo.

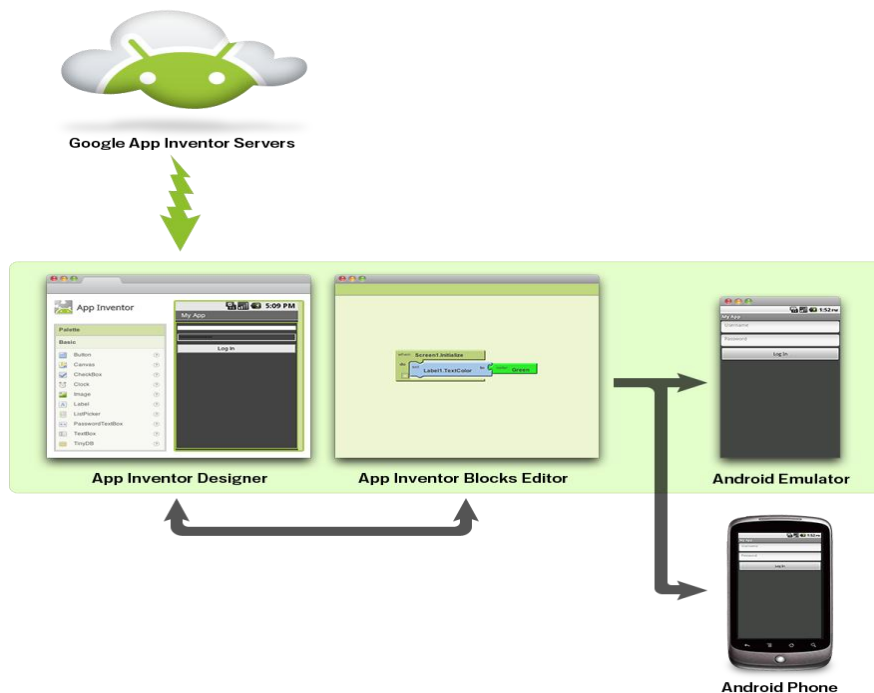


Figura 1. Visión global de AppInventor

La principal característica de *AppInventor* es que no es necesario tener ningún conocimiento de programación para desarrollar las aplicaciones. Simplemente basta con disponer de un navegador web y una cuenta de usuario de Google.



## 2. ¿Qué tipo de aplicaciones pueden crearse con AppInventor?

*AppInventor* ofrece un amplio abanico de posibilidades para los desarrolladores:

### **Juegos**

Puede crearse juegos sencillos haciendo uso incluso del acelerómetro incluido en el dispositivo móvil.

### **Aplicaciones educativas**

Es posible desarrollar aplicaciones útiles para educación, como por ejemplo tests de respuestas múltiples o preguntas directas.

### **Aplicaciones de geolocalización**

Puede hacerse uso del dispositivo GPS incluido en el móvil para crear aplicaciones de geolocalización.

### **Aplicaciones multimedia complejas**

Pueden crearse aplicaciones que van desde reconocimiento de códigos de barras hasta reproducir vídeos y música o tomar fotografías.

### **Aplicaciones orientadas a la Web**

Pueden desarrollarse aplicaciones que se comuniquen con la web (Twitter, RSS, etc.).



### 3. ¿Cómo se construye una aplicación en AppInventor?

Las aplicaciones construidas mediante AppInventor están compuestas por los elementos que se muestran en el siguiente diagrama:

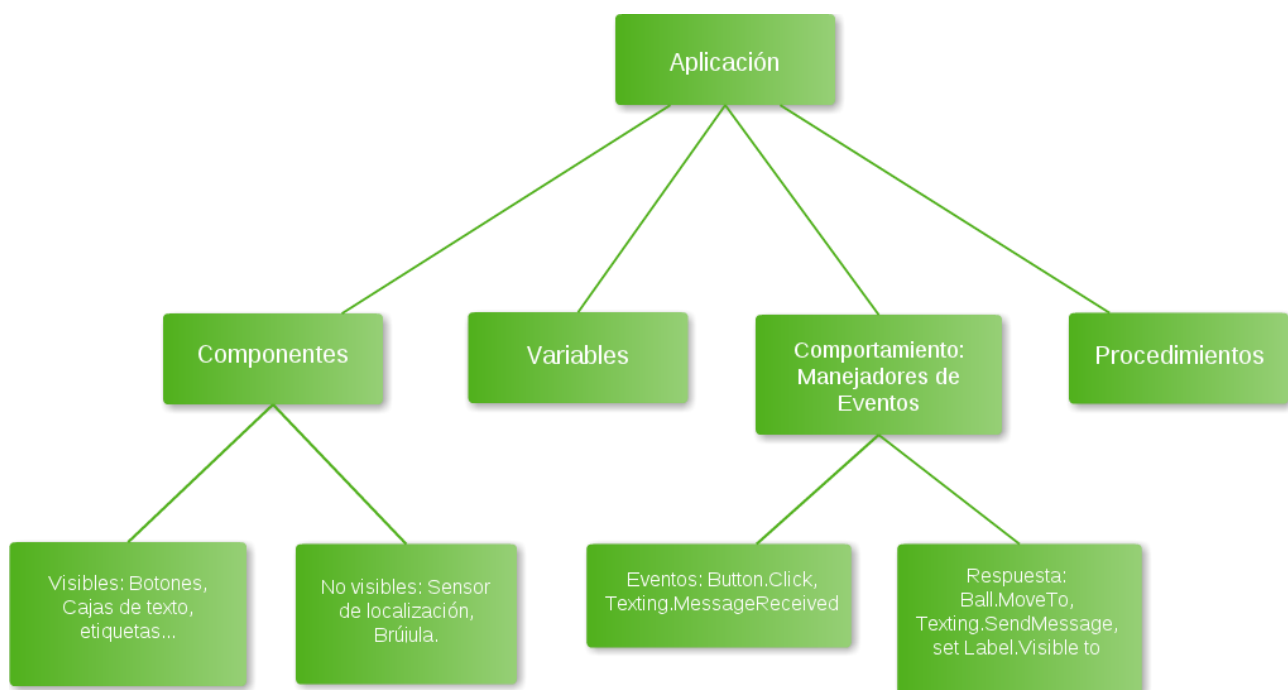


Figura 2. Arquitectura interna de una aplicación creada con AppInventor

Una buena manera de entender una aplicación, es descomponerla en dos partes, por un lado los componentes y por otro los comportamientos.



### 3.1. Componentes

Hay dos tipos de componentes principales en cualquier aplicación: los visibles y los no visibles.

Los componentes visibles son aquellos que podemos ver una vez hemos ejecutado nuestra aplicación (botones, cajas de texto, etiquetas, etc.). El conjunto de estos elementos se denomina comúnmente como la interfaz de usuario de la aplicación.

Por otro lado, los componentes no visibles son aquellos que no podemos ver en la aplicación, ya que no son parte de la interfaz de usuario. Proporcionan acceso a la funcionalidad interna de los dispositivos; por ejemplo, el componente *Texting* permite enviar y procesar mensajes de texto, y el componente *LocationSensor* permite determinar la localización del dispositivo.

Ambos componentes están definidos mediante una serie de propiedades. Las propiedades son fragmentos de memoria que permiten almacenar información relativa al componente al que referencian. Los componentes visibles, por ejemplo, disponen de propiedades relativas a su posición, altura y anchura, y alineación, que definen conjuntamente su aspecto dentro de la aplicación global. Todas estas propiedades se definen dentro del diseñador de componentes de AppInventor.



Figura 3. Ejemplo de un componente y sus propiedades

## 3.2. Comportamiento

El comportamiento define como una aplicación debe responder ante una serie de eventos, los producidos por la interacción del usuario (un clic de botón) y los externos (un SMS recibido en nuestro dispositivo). En este punto es donde reside la mayor complejidad en el desarrollo de aplicaciones. Afortunadamente, AppInventor proporciona un lenguaje visual de bloques que nos permite definir comportamientos de una forma muy precisa.

Normalmente, podemos identificar el desarrollo de aplicaciones con la elaboración de una “receta”, es decir, siguiendo una secuencia lineal de instrucciones.





Sin embargo, la mayoría de las aplicaciones actuales, no cumple estrictamente este tipo de paradigma. No se ejecutan una serie de instrucciones en un orden predeterminado, sino que, la aplicación reacciona a una serie de eventos, normalmente iniciados por el usuario final de la aplicación. Por ejemplo, si el usuario hace clic sobre un botón, la aplicación responde realizando alguna operación (enviar un mensaje de texto, confirmar una determinada operación, etc.).

Este tipo de aplicaciones se pueden interpretar como un conjunto de componentes que reaccionan ante unos determinados eventos. Las aplicaciones incluyen una serie de “recetas” (secuencias de instrucciones), las cuales se ejecutan cuando se producen los eventos asociados a las mismas.

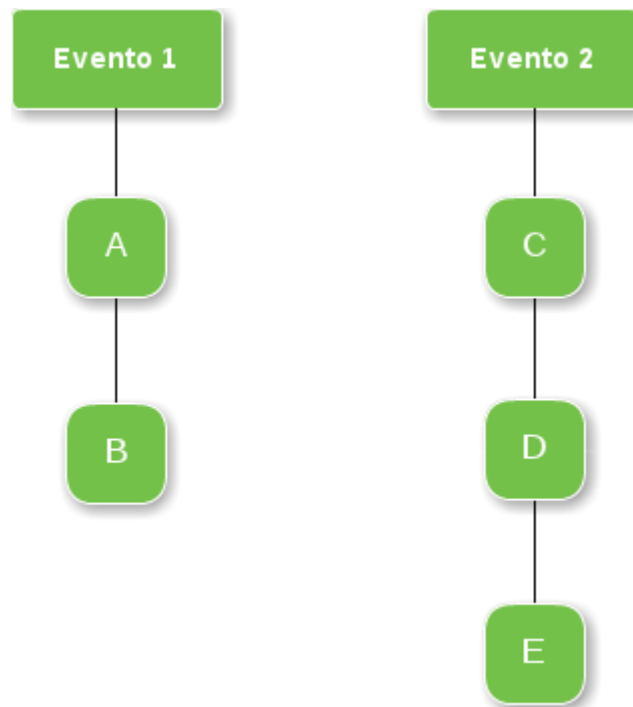


Figura 4. Ejemplo secuencial de eventos e operaciones

La mayoría de los eventos son generados por el usuario final de la aplicación, salvo en algunos casos. Una aplicación puede reaccionar a eventos que suceden en el propio dispositivo, como por ejemplo: cambios en el sensor de orientación, eventos generados en otros dispositivos, datos que llegan desde la web, etc.

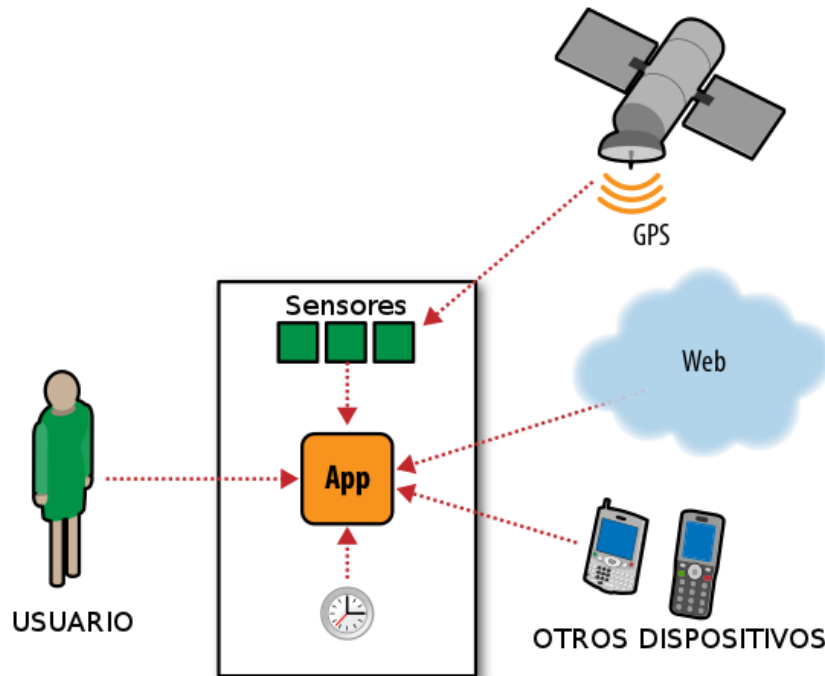


Figura 5. Una aplicación puede responder tanto a eventos internos, externos o generador por el usuario.

El motivo principal por el cual podemos afirmar que las aplicaciones desarrolladas mediante AppInventor son intuitivas, es porque están basadas en el paradigma evento-respuesta.

### 3.2.1. Eventos

A continuación, explicaremos brevemente los diferentes tipos de eventos que pueden desencadenar diferentes acciones en este tipo de aplicaciones.

Tipo de Evento	Ejemplo
Evento iniciado por el usuario	<i>Cuando el usuario hace clic en el boton_1, hacer...</i>
Evento de inicialización	<i>Cuando la aplicación se inicia, hacer...</i>
Evento de temporización	<i>Cuando han pasado 20 milisegundos, hacer...</i>
Evento de animación	<i>Cuando dos objetos colisionan, hacer...</i>
Evento externo	<i>Cuando el teléfono recibe un SMS, hacer...</i>

#### Eventos iniciados por el usuario

Son el tipo más común de eventos. Reflejan la interacción del usuario final con la aplicación. Por ejemplo, en la siguiente figura podemos observar como se ha definido que cuando el usuario hace clic sobre el botón “SpeakItButton” se reproduce oralmente el contenido escrito en la caja de texto “TextBox1”.



Figura 6. Ejemplo de evento iniciado por la interacción del usuario



### Eventos de inicialización

En algunas ocasiones nuestras aplicaciones requieren realizar ciertas funciones en el momento en el que se inician. AppInventor considera este inicio de la aplicación como un evento.

En el siguiente ejemplo, podemos observar como en el momento en el que se inicia la aplicación realizamos una determinada acción, en este caso, invocar a la función “MoveMole”.

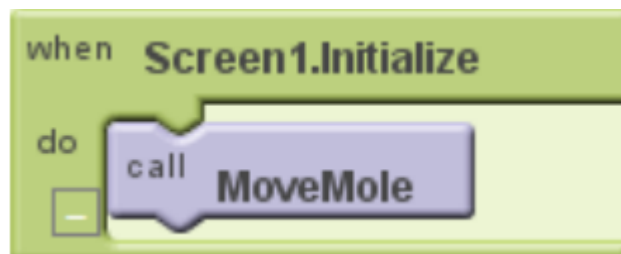


Figura 7. Ejemplo de evento de inicialización

### Eventos de temporización

Podemos necesitar en algunos casos que cierta actividad de nuestra aplicación se realice en un cierto momento. AppInventor dispone de un componente denominado “Clock”, mediante el cual podremos programar la ejecución de determinadas acciones en un determinado momento.

Por ejemplo, en la siguiente figura podemos observar como se ha definido un temporizador para que en un determinado tiempo mueva una bola desplazándola 10 unidades sobre el eje de las X.

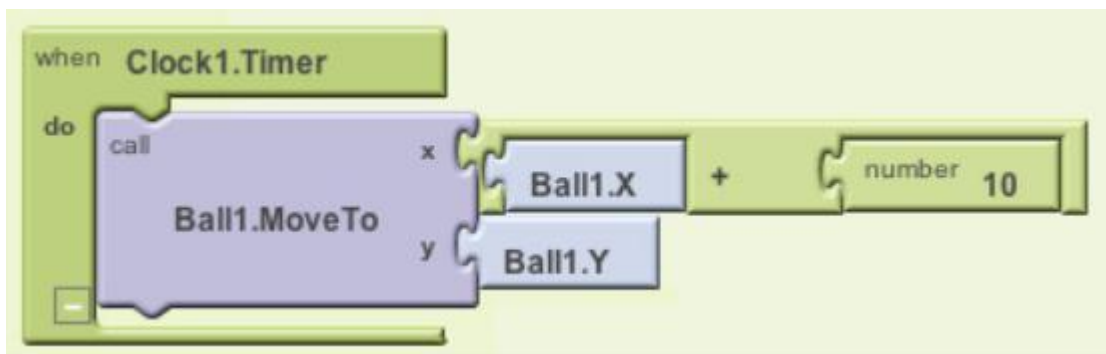


Figura 8.. Ejemplo de evento de temporización



## Eventos de animación

Las actividades en las que se ven envueltos objetos gráficos también pueden producir eventos. De esta manera podemos crear juegos o aplicaciones con animaciones interactivas, controlando en todo momento que debería ocurrir.

En el siguiente ejemplo, se ha definido un comportamiento que controla las colisiones entre el objeto "Ball1" y el objeto denominado "other", de tal modo que cuando se produce una colisión entre ambos, el objeto "Ball1" cambia su color a rosa y se reproduce un sonido de explosión.



Figura 9. Ejemplo de evento de animación

## Eventos externos

Cualquier entrada externa hacia el dispositivo es considerada también como un evento. Por ejemplo, el hecho de recibir un mensaje de texto o la posición GPS son interpretados como eventos externos.

En el siguiente ejemplo, podemos observar como se ha definido un conjunto de bloques que reaccionan al evento de recibir un mensaje de texto. Una vez recibido, se responde al emisor con el mensaje de "Stop texting me!".

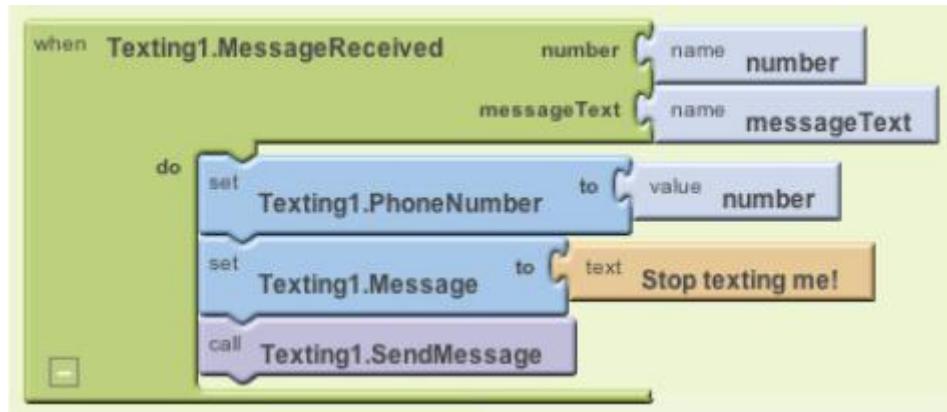


Figura 10. Ejemplo de evento externo

Para resumir, podemos decir que todas las aplicaciones que desarrollemos estarán formadas por un conjunto de eventos y sus respectivas respuestas. Nuestro trabajo será conceptualizar y definir las respuestas a los eventos que queremos manejar.



### 3.2.2. Manejadores de Eventos

Existen varios tipos de manejadores de eventos, los cuales explicaremos a continuación.

#### Condicionales

Las respuestas a eventos no son siempre secuencias lineales. En algunos casos, se pueden evaluar ciertas condiciones y elegir en función de ello que hacer.

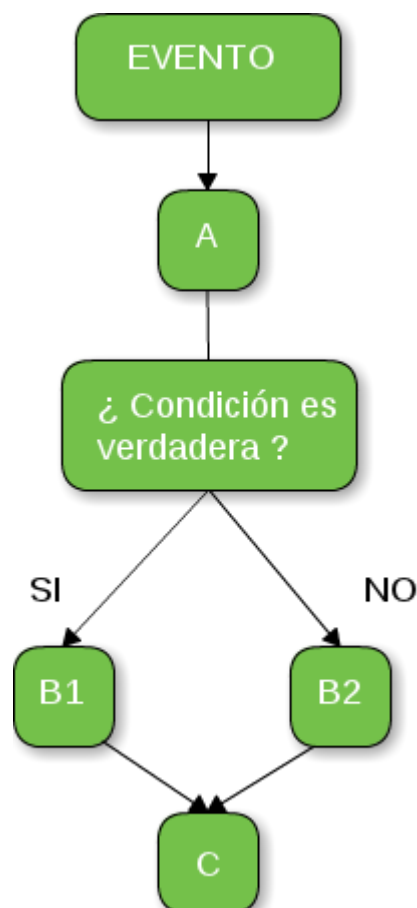


Figura 11. Ejemplo de manejador condicional



## Bloques de repetición

En algunas ocasiones podemos desear que nuestra aplicación repita ciertas operaciones. En estos casos utilizaremos los bloques de repetición proporcionados por AppInventor, que nos permitirán ejecutar ciertas tareas un número determinado de veces o mientras sea verdadera alguna condición.

## Recordar valores

Debido a que los manejadores de eventos pueden ejecutar bloques, en algunas ocasiones es necesario mantener cierta información almacenada. Esta información se almacena en unas unidades de memoria denominadas variables, las cuales podremos definir en nuestras aplicaciones. Por ejemplo en una aplicación para jugar, podemos definir una variable puntuación para almacenar y modificar su valor en función de los aciertos y errores del usuario.

Podemos decir que las variables son como las propiedades , pero no están asociadas a ningún componente en particular. Las variables almacenan datos temporalmente mientras la aplicación se está ejecutando; cuando cerramos la aplicación, los datos ya no están disponibles.

En algunas ocasiones, puede ser necesario que ciertos datos estén disponibles no sólo mientras se ejecuta la aplicación, sino también cuando se cierra y vuelve a ejecutarse. Es en este caso particular, cuando necesitaremos utilizar un almacenamiento persistente, utilizando algún tipo particular de base de datos.





## Comunicaciones con la web

Algunas aplicaciones se comunican a través de internet enviando peticiones hacia servicios web.

Twitter es un ejemplo de servicio web con el cual ApplInventor puede comunicarse. Podemos crear aplicaciones que sean capaces de enviar peticiones y mostrar los tweets de nuestros seguidores e incluso actualizar nuestro estatus.

## 4. Requisitos Previos

Antes de comenzar con el desarrollo de aplicaciones para Android, necesitamos seguir una serie de pasos.

### Comprobar que cumplimos todos los requisitos para poder utilizar ApplInventor

#### Ordenador y Sistema Operativo

Macintosh (con procesador Intel): Mac OS X 10.5, 10.6

Windows: Windows XP, Windows Vista, Windows 7

GNU/Linux: Ubuntu 8+, Debian 5+

#### Navegador Web

Mozilla Firefox 3.6 o superior

Nota: Si estamos utilizando Firefox con la extensión NoScript, deberemos deshabilitarla para evitar problemas de utilización.

Apple Safari 5.0 o superior

Google Chrome 4.0 o superior

Microsoft Internet Explorer 7 o superior



Para poder empezar a utilizar el servicio de ApplInventor necesitaremos tener una [cuenta de google](#). Si no disponemos de ella, podremos crear una de manera gratuita en el siguiente [enlace](#).

### Comprobar nuestra configuración de la máquina virtual de Java

- Nuestro ordenador necesita tener instalado Java 6. Para descargar la última versión de Java visitaremos la web [www.java.com](http://www.java.com).
- Visitar la página de prueba de Java. Deberíamos observar un mensaje que indique que Java está funcionando correctamente y que la versión de Java es la 1.6 o superior.
- Ejecutar el test Java de ApplInventor haciendo clic en este [enlace](#). Este proceso comprobará que nuestro navegador está configurado apropiadamente para ejecutar Java y que nuestro ordenador puede ejecutar aplicaciones mediante Java Web Start.

ApplInventor no funcionará en nuestro ordenador si estos tests no han tenido éxito. Antes de comenzar a utilizar la aplicación debemos comprobar todos los pasos anteriores.

### Instalar el software de ApplInventor en nuestro ordenador

Antes de utilizar ApplInventor, debemos instalar el software adecuado al sistema operativo que vayamos a utilizar. En los siguientes enlaces se proporcionan instrucciones detalladas de como instalar el software:

- [Instrucciones para Mac OS X](#)
- [Instrucciones para GNU/Linux](#)
- [Instrucciones para Windows](#)



A partir de este momento, ya podremos empezar a utilizar y crear nuestras aplicaciones mediante AppInventor.

## 5. El Entorno de Desarrollo

EL entorno de programación de AppInventor tiene tres partes fundamentales:

- El *Diseñador de Componentes* se ejecuta en el navegador web. Es utilizado para seleccionar los componentes de nuestra aplicación y especificar sus propiedades. A través de él, iremos definiendo el aspecto visual.
- El *Editor de bloques* se ejecuta en una ventana independiente del *Diseñador de Componentes*. Nos permitirá crear los comportamientos necesarios de nuestra aplicación y asociarlos a sus respectivos componentes.
- Un dispositivo Android nos permitirá ejecutar y comprobar nuestras aplicaciones mientras las estamos desarrollando. Si no disponemos de ningún dispositivo adecuado, podremos probar nuestras aplicaciones utilizando el emulador de Android, el cual viene integrado dentro del sistema.

En la siguiente figura se puede visualizar cada una de las partes mencionadas anteriormente.

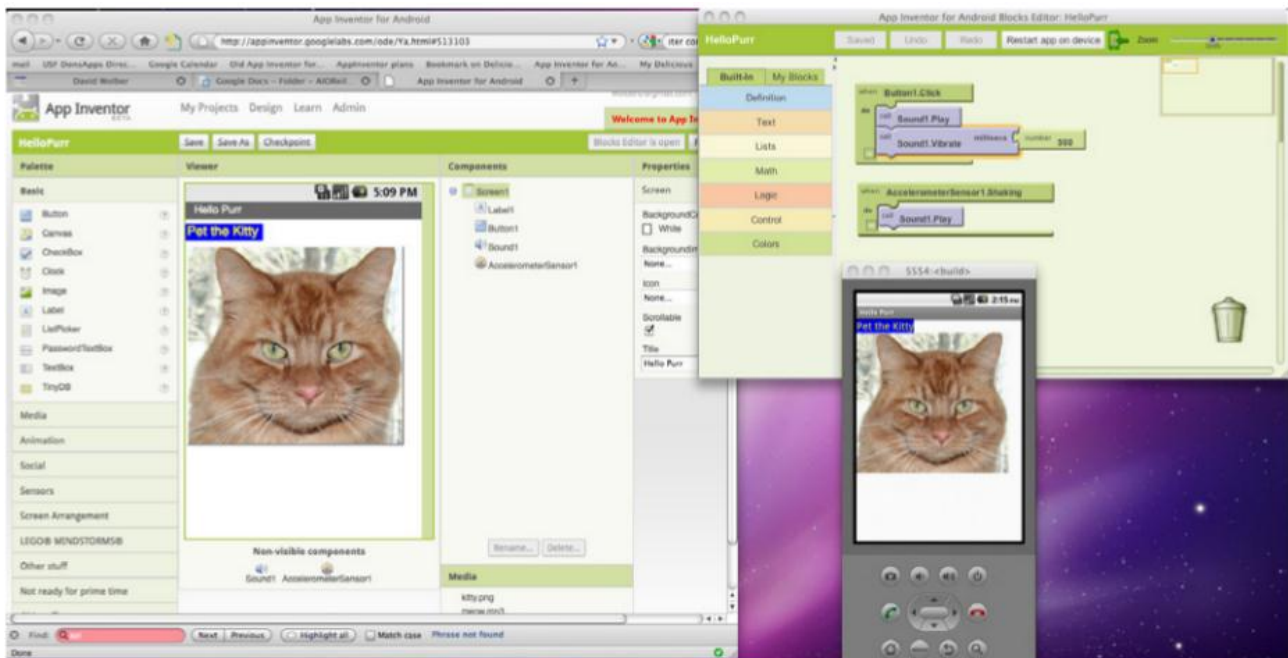


Figura12. El Diseñador de Componentes, el Editor de Bloques y el Emulador de Android.

Para empezar a utilizar AppInventor, comenzaremos introduciendo la siguiente dirección web en nuestro navegador <http://appinventor.googlelabs.com>. La primera vez que iniciemos sesión en la aplicación, podremos ver nuestra Página de Proyecto, la cual estará vacía porque aún no hemos desarrollado ninguna aplicación. Para crear un proyecto, haremos clic sobre el botón *New* situado en la parte superior izquierda de la pantalla, introduciremos el nombre que queramos para el proyecto y haremos clic en OK.

La primera ventana que nos aparecerá a continuación es el *Diseñador de Componentes*. Una vez que estamos dentro, haremos clic sobre el botón *Open Blocks Editor* situado en la parte superior derecha. En este momento, el Editor de Bloques aparecerá en una nueva ventana. Este proceso puede tardar en torno a 30 segundos.

En la ventana del Editor de Bloques podemos observar dos botones en la parte superior derecha, como muestra la siguiente figura.



Figura 13. Parte superior del editor de bloques

Si disponemos de un dispositivo Android y un cable USB, conectaremos el dispositivo al ordenador y seleccionaremos *“Connect to Device”*. Por el contrario sino disponemos de ningún dispositivo Android o si queremos probar nuestras aplicaciones utilizando un emulador, haremos clic en *“New Emulator”* y esperaremos en torno a 30 segundos hasta que el emulador termina de cargar. Cuando haya terminado el proceso, haremos clic en *“Connect to Device”* y seleccionaremos el emulador creado previamente.

## 5.1. Diseñando los Componentes

Los componentes son los elementos que combinamos para crear nuestras aplicaciones. Algunos son muy simples, como por ejemplo una etiqueta que muestra un texto en la pantalla, o un botón, mediante el cual podemos iniciar una determinada acción. Otros componentes son más elaborados; un componente *Canvas* que permite visualizar imágenes o animaciones; el acelerómetro, un sensor de movimiento que detecta cuando movemos o agitamos el teléfono; o componentes que nos permiten enviar mensajes de texto, reproducir música y video, obtener información desde páginas web, etc.

Cuando abrimos el *Diseñador de Componentes*, aparecerá la siguiente ventana.

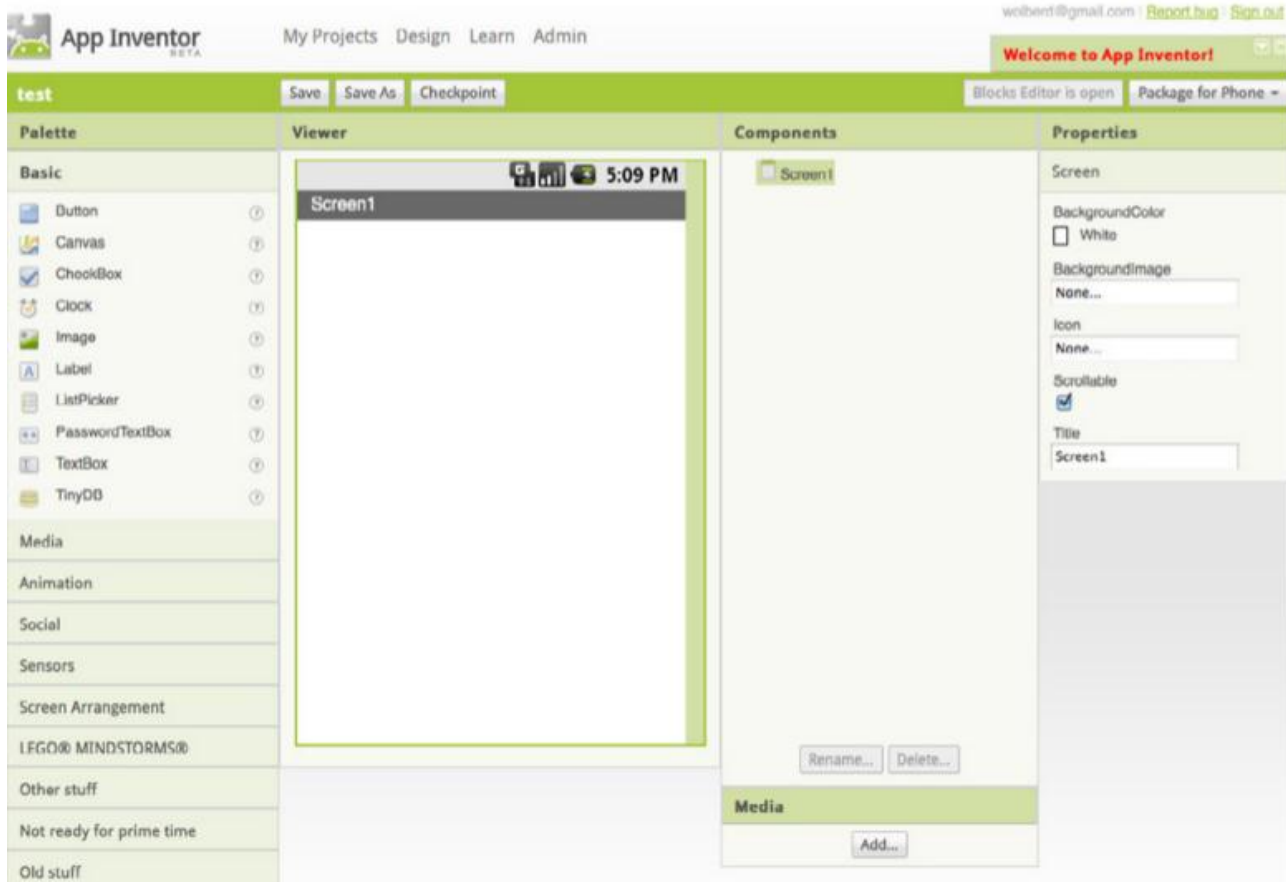


Figura 14. Ventana principal del Diseñador de Componentes

El *Diseñador de Componentes* está dividido en varias partes:

En la parte central encontramos un área denominada *Viewer*. Aquí es donde colocaremos nuestros componentes en función de cómo queremos que sea el diseño visual de nuestra aplicación. Una vez especificados y añadidos los componentes, debemos probar como se visualiza realmente nuestra aplicación en un dispositivo Android, utilizando para ello un dispositivo físico o el emulador.

A la izquierda de la aplicación encontramos un área denominada *Palette*, en la que podemos encontrar la lista completa de componentes de AppInventor. La *Palette* está dividida en varias



categorías. Haciendo clic en cada una de ellas podremos ir viendo todos los componentes disponibles.

A la derecha del *Viewer* podemos ver el listado de *Componentes*, el cual muestra una lista de todos los componentes que vamos añadiendo a nuestra aplicación. Todos los componentes que vayamos añadiendo al *Viewer*, se irán mostrando automáticamente en el listado. Por defecto, siempre aparece el componente *Screen1*, que hace referencia a la pantalla del teléfono.

En la parte inferior podemos observar un área denominada *Media* que muestra todos los componentes multimedia (imágenes y sonidos) que incluyamos en nuestro proyecto.

Por último en la parte derecha de la aplicación, encontramos la sección de propiedades, mediante la cual podremos ir modificando, según nuestras necesidades, todos y cada uno de los elementos que vayamos incluyendo en la aplicación. La lista de propiedades irá cambiando en función del elemento que estemos editando, para ello haremos clic sobre el elemento deseado en el listado y aparecerán las propiedades que podemos modificar para el elemento actual seleccionado.

## 5.2. Añadiendo Comportamientos

El *Editor de Bloques (Blocks Editor)* nos permite añadir y asignar tareas específicas a los componentes individuales de nuestra aplicación, de tal modo que podamos ir creando de manera conjunta la funcionalidad global que pretendemos que tenga nuestra aplicación.

El *Editor de Bloques* está implementado como una aplicación de *Java Web Start*, que se ejecuta en nuestro ordenador.

Para iniciarlo, debemos hacer clic sobre el botón *“Open the blocks editor”* situado en la parte superior derecha del *Diseñador de Componentes*. Una vez pulsado, el texto del botón cambiará y nos indicará que se está cargando el *Editor de Bloques*. Nuestro navegador web nos



preguntará en una nueva ventana que queremos hacer con la aplicación. Aceptamos y automáticamente se mostrará la ventana del *Editor de Bloques*.

Una vez cargado el *Editor de Bloque* aparecerá la siguiente ventana.

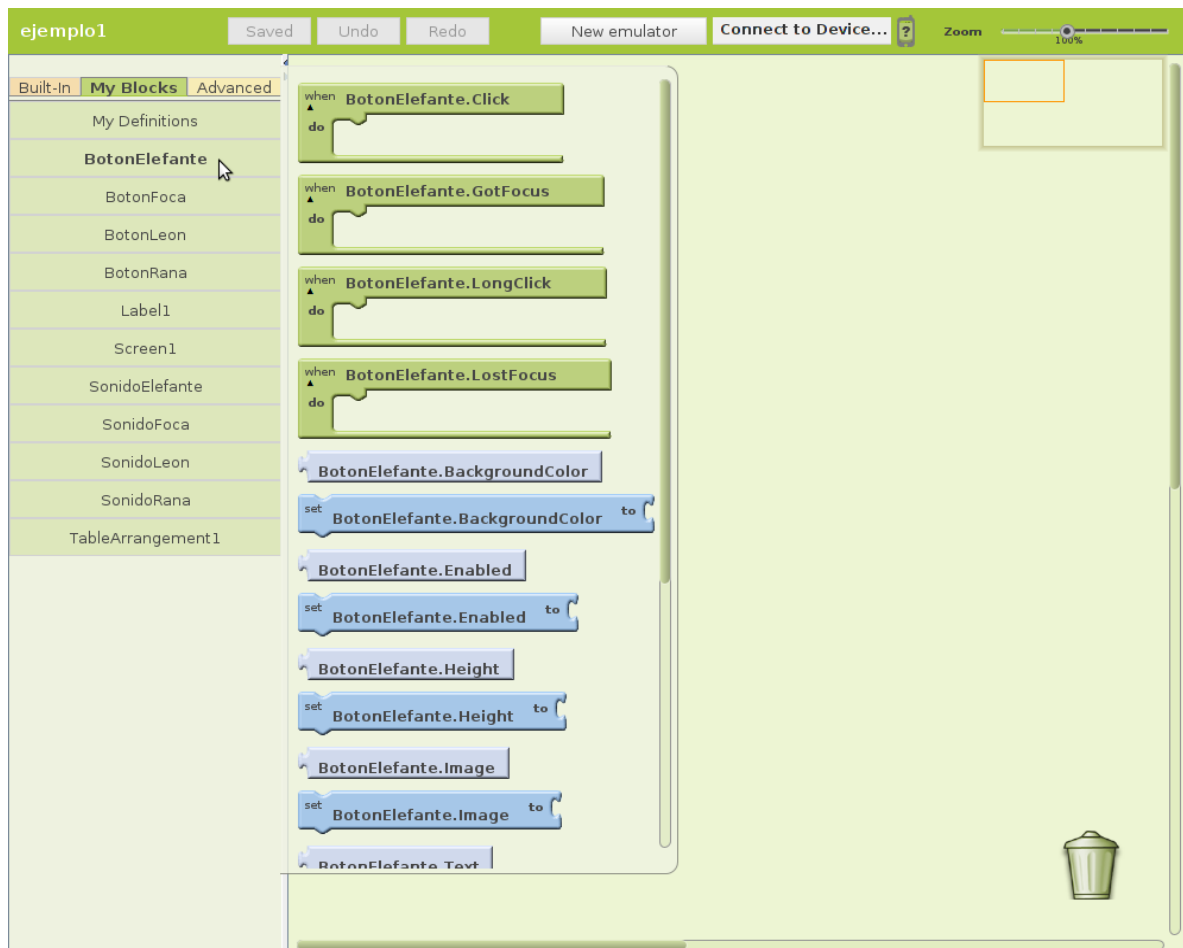


Figura 15. Ventana principal del Editor de bloques





Se divide en las siguientes partes:

- En la parte izquierda de la pantalla encontramos la zona para seleccionar bloques. A su vez, incluye tres pestañas: *Built-In*, *My Blocks* y *Advanced*. La pestaña *Built-In* contiene siete grupos genéricos de bloques: *Definition*, *Text*, *Lists*, *Math*, *Logic*, *Control* y *Colors*. Todos estos bloques estarán siempre disponibles para incluirlos en nuestras aplicaciones, independientemente de los componentes utilizados y añadidos a través del diseñador. La pestaña *My Blocks* nos permite acceder a todos los componentes definidos en el diseñador, con el fin de asignarles la funcionalidad y el comportamiento que deseemos. Por último, la pestaña de *Advanced*, nos permite aplicar comportamientos sobre grupos de componentes (botones, etiquetas, etc.) de manera global.
- En la parte central se encuentra el editor de bloques, mediante el cual definiremos las tareas y el comportamiento de los componentes de nuestra aplicación. Iremos combinando bloques según nuestras necesidades para completar el funcionamiento de nuestra aplicación.



VNIVERSIDAD  
SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL



## 6. Referencias

<http://code.google.com/p/taiic/>

<http://appinventorintro.com/>

<http://www.tuappinventorandroid.com/>

<http://www.appinventorbeta.com/forum/>

<http://appinventorblog.com/>



Oficina de Conocimiento Abierto