

Taller 2: conceptos básicos y entorno  
de programación App Inventor 2.0



educador digital

# TecnoTIC

Diplomado para docentes de tecnología e informática en el uso pedagógico  
de las TIC con impacto en los aprendizajes de los estudiantes

# Contenido

---

Presentación - ¿Qué es Android?

---

¿Para aprender a programar el Android necesito tener un teléfono móvil con Android?

---

¿Qué programa se utiliza para programar el Android?

---

Historia

---

¿Cómo es el código?

---

¿De dónde me bajo el App inventor?

---

¿En qué dirección entro para programar?

---

Interface

---

Ejecución de nuestro código.

---

App inventor 2 en español offline.



---

Resumen de instalación.

---

Ejecución de nuestro código.

---

Componentes (básicos, multimedia, animación, sensores, alineamientos de pantalla)

---

ACTIVIDAD 2. Personaliza la interfaz gráfica de la app.

---

---



## - ¿Qué es Android?

Android es un sistema operativo gratuito que utiliza muchos teléfonos móviles actuales (teléfonos inteligentes = smartphone), está basado en LINUX. Android pertenece a Google.

Cada cierto tiempo aparecen nuevas versiones: 1.5 (Cupcake), 1.6 (Donut), 2.0 (Eclair), 2.2 (Froyo), 2.3 (Gingerbread), 3.2 (Honeycomb), 4.0 (Ice Cream Sandwich), 4.1 (Jelly Bean), 4.2 Jelly Bean (Gummy Bear), 4.3 (Jelly Bean), 4.4 (KitKat), 4.5 (kit kat), 5.0 (Lollipop) ...

El Android lo podemos programar fácilmente y realizar programas de fórmulas matemáticas, juegos, ubicación en los mapas de Google, mover un robot mediante Bluetooth, contestación automática de SMS,...



¿Para aprender a programar el Android necesito tener un teléfono móvil con Android?

No. Solo necesitas un ordenador y conexión a Internet. En el ordenador puedes hacer los programas y probarlos en un emulador instalado en el mismo ordenador.

Si tienes un teléfono móvil o tablet con Android puedes pasar los programas realizados al móvil para ver cómo funcionan y utilizarlos.



## - ¿Qué programa se utiliza para programar el Android?

Indico tres formas muy utilizadas de programación:

1.- Con Android Studio y Java . Se realizan los programas escribiendo códigos en Java. Es la forma que utilizan los buenos programadores.

2.- Mediante una aplicación de pago (50 € aproximadamente) llamada (B4A) BASIC4Android, para realizar programas de manera parecida a como se hace con Visual Basic

3.- Mediante App inventor. Es una forma intuitiva de hacer programas situando convenientemente las instrucciones como si fueran piezas de un puzzle. Utilizaremos App inventor para realizar nuestros programas. Lo que sí necesitas es estar conectado a Internet para poder utilizar el App inventor.

## - Historia

- App Inventor apareció el 12 de julio de 2010, y se abrió a todo el público en 15 de diciembre de 2010.

- Lo desarrollaron entre Hal Abelson de MIT y los ingenieros de Google Ellen Spertus y Liz Looney.

- A principio de agosto de 2011 Google anunció que ya no mantendría esta aplicación, pero que la haría código libre destinado a la educación.

- Una semana después el Instituto Tecnológico de Massachusetts (MIT), una institución de educación superior privada situada en Cambridge, Massachusetts (EE.UU.), anunció que se haría cargo del proyecto.

- El 31 de diciembre de 2011 App Inventor de Google dejó de funcionar.

- El 4 de marzo de 2012, el Instituto Tecnológico de Massachusetts (MIT) volvió a poner el proyecto en Internet.

- En diciembre de 2013, MIT saca AI 2, una nueva versión de App Inventor: <http://ai2.appinventor.mit.edu>

- En noviembre de 2014 sale la opción de idioma, con lo cual podemos ver la mayoría de los elementos en español.

- El 15 de julio de 2015 desaparece App inventor (el antiguo). Existe una aplicación web para pasar los proyectos .zip del App inventor al .aia del App inventor 2. Aplicación online.

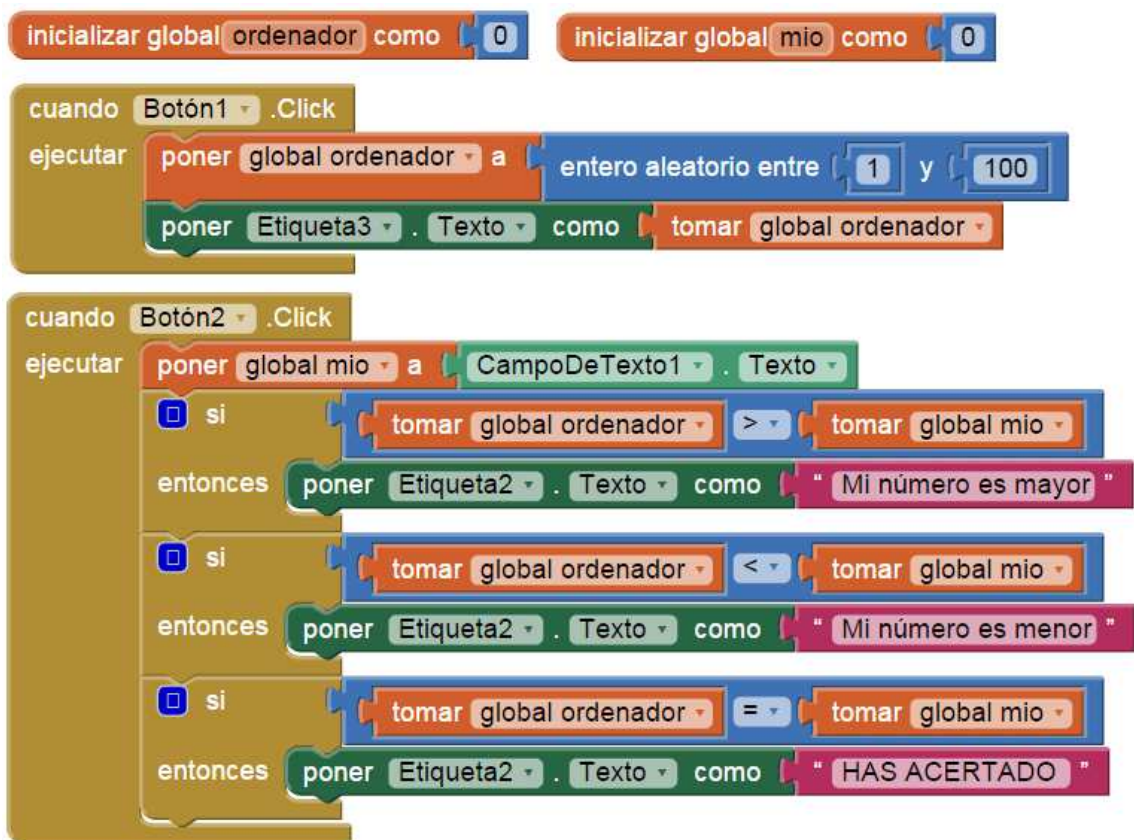
- En julio de 2015 crean una mochila en la parte de Bloques para poder copiar y pegar.



- En diciembre de 2015 se crea el componente FirebaseDB.
- El 15 de junio de 2016 agrega el componente podómetro.
- A principio de 2016 comienza a funcionar los servicios en modo de prueba.

## - ¿Cómo es el código?

- En forma de Bloques...

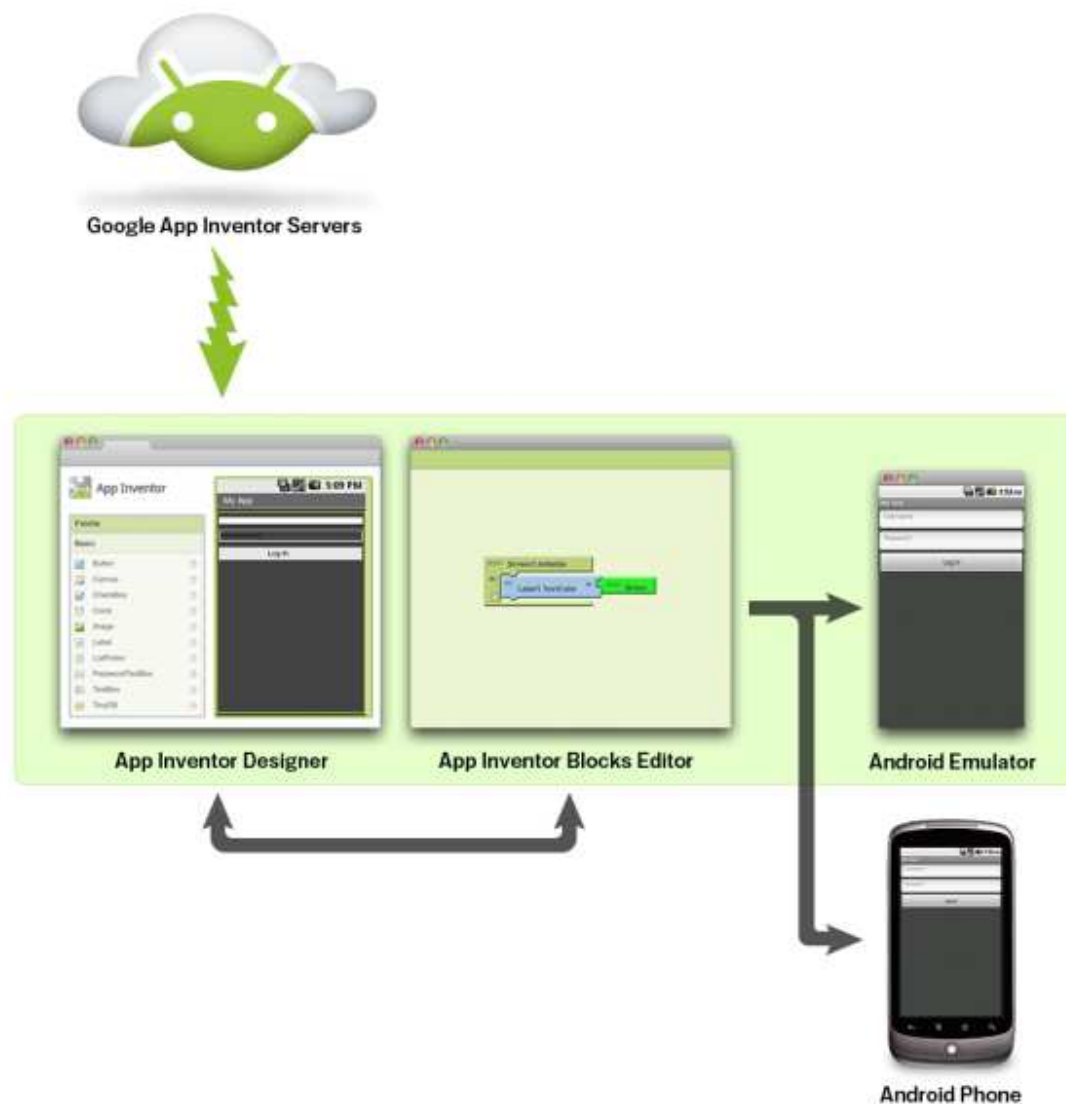


## - ¿De dónde me bajo el App inventor?

El App inventor no es un programa que te tengas que bajar al computador. Funciona en cloud computing, el programa funciona mediante la conexión a Internet, tu trabajas con el programa conectado directamente al servidor de App inventor. Es como una página web y en ella realizas las operaciones. Solo necesitas una cuenta de gmail.

NOTA: hay versiones para poderlas instalar en el PC y trabajar sin estar conectado a internet (offline).





Para que funcione necesitas tener instalado JAVA. También necesitas una cuenta de Google Gmail.

- ¿En qué dirección entro para programar?

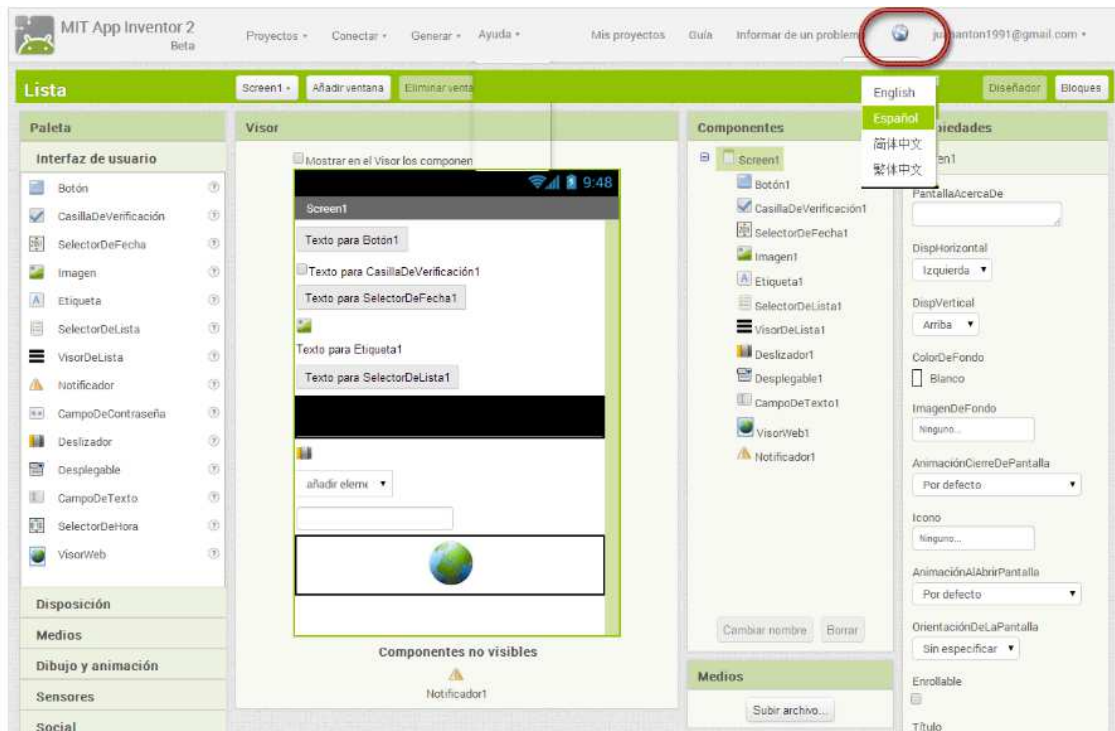
Si vas a estudiar App inventor 2, entra en esta dirección y te autentica con tu dirección de correo de gmail...

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



## - Interface.

Podemos trabajar en español, en inglés, francés, portugués, italiano...



## - App inventor 2 en español offline.

- Hemos visto que para trabajar con AI2 debemos estar conectados a Internet, ¿se podría trabajar de alguna manera offline, es decir sin estar conectado?

- Pues sí, para ello un equipo distinto al MIT ha creado un AI2 Ultimate con Personal Server. Esto instala el AI2 en nuestro computador, utilizando JAVA. Luego en un navegador escribimos <http://localhost:8888> y podemos trabajar directamente con nuestro computador sin estar conectado a Internet.

- He preparado un pequeño tutorial de instalación de App inventor 2 Ultimate, es interesante que lo leas y lo realices:

[http://kio4.com/appinventor/125app\\_inventor\\_2\\_ultimate.htm](http://kio4.com/appinventor/125app_inventor_2_ultimate.htm)

[http://kio4.com/appinventor/125app\\_inventor\\_2\\_ultimate.htm](http://kio4.com/appinventor/125app_inventor_2_ultimate.htm)





## - Resumen de instalación.

- Podemos trabajar online y offline.

Online, con conexión a Internet es la forma "oficial", nos creamos una cuenta de correo, por ejemplo de gmail y creamos el código mediante la conexión a un sitio web.

Podremos ver el resultado de nuestro código en un emulador instalado en el ordenador, en el móvil/tablet mediante Wifi o en el móvil/tablet mediante cable USB.

Para entrar en esa web vamos a... <http://ai2.appinventor.mit.edu/>

Offline, se trata de instalar en nuestro Computador el "programa" App Inventor 2 y trabajar con él sin necesidad de estar conectado a Internet.

Podremos ver el resultado de nuestro código en un emulador instalado en el computador, en el móvil/tablet mediante Wifi o en el móvil/tablet mediante cable USB.

Para bajar el "programa" App Inventor 2 vamos a... <http://sourceforge.net/projects/ai2u/files/>

- Hay otra versión offline llamada AiLiveComplete!:  
<http://sourceforge.net/projects/ailivecomplete/files/>

<https://amerkashi.wordpress.com/page/3/>

Pulsamos en WinStartBuildServer.cmd y luego escribimos en un navegador localhost:8888

Portable, lo ponemos en un directorio o en un USB y trabajamos sin necesidad de tenerlo instalado.

## - Ejecución de nuestro código.

Tanto si estamos trabajando Offline como Online, hay varias formas de comprobar cómo está funcionando nuestro código:

- Instalando en el móvil/tablet una aplicación bajada de la Play de Google, llamada: MIT AI2 Companion, luego necesitamos que la red donde trabajamos tenga Wifi. Es decir, que el móvil tenga Wifi (siempre lo tiene) y nuestro ordenador esté conectado a la misma red Wifi. (NO es necesario que el ordenador esté conectado a Wifi, puede estar conectado con cable a la red, lo que quiero decir es que tanto el móvil como el ordenador estén conectados a la misma red y que ésta tenga Wifi).



Pulsaremos en App Inventor AI Companion. Nos saldrá en la pantalla del ordenador un código QR. Arrancamos MIT AI2 en nuestro móvil y escaneamos ese QR para cargar la aplicación que hemos realizado.

- Instalando en nuestro computador un Emulador, es decir una especie de teléfono móvil que actúe como nuestro móvil. Para ello bajamos:

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

Lo instalamos en Windows, nos saldrá un icono Ai2 Starter. Pulsamos ese icono para arrancar este servidor. Luego en App Inventor pulsamos en: Emulador.

- Conectando el móvil al ordenador por cable USB. Instalamos en nuestro ordenador el driver correspondiente a nuestro modelo de móvil.

Después instalamos <http://appinventor.mit.edu/explore/ai2/setup-device-usb.html>

Arrancamos el Ai2 Starter. Luego en App Inventor pulsamos en: USB

## Componentes Básicos

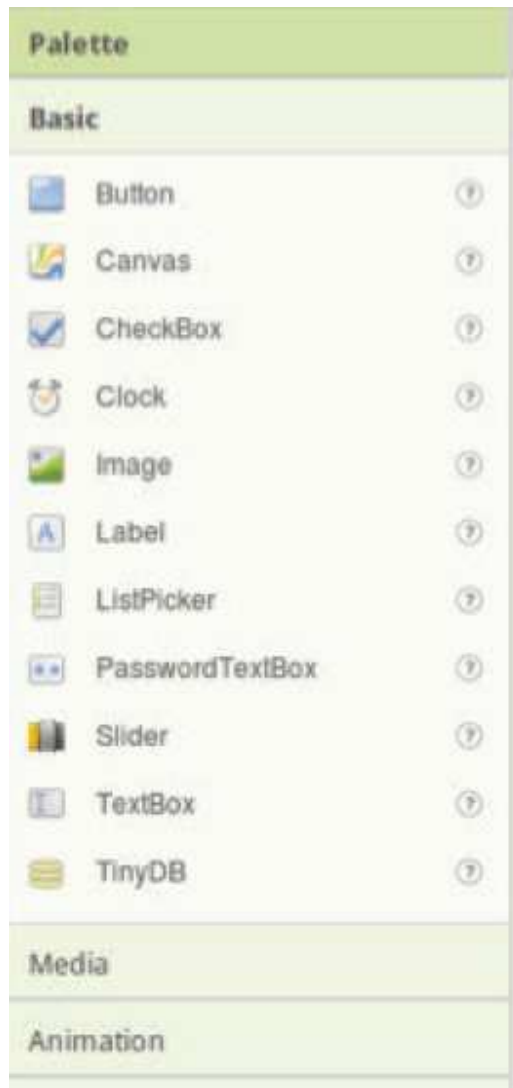
Introducción:

Al crear un nuevo proyecto Android, Google App Inventor nos abrirá la ventana de diseño de la aplicación:



En la parte izquierda tendremos los componentes visuales y no visuales que podremos añadir a nuestra aplicación con tan solo arrastrar y soltar:

A continuación mostramos la lista de componentes disponibles en la paleta (Palette) agrupados por tipo. Hay que mencionar que dicha lista puede variar pues Google App Inventor 2.0 está en constante actualización y por tanto será susceptible de modificaciones futuras añadiendo nuevos componentes y mejorando los actuales:



- **Image (Imagen)**

Se usa el componente de imagen para representar imágenes que los usuarios seleccionan y manipulan.

Un componente de la imagen muestra una imagen. Se puede especificar la imagen para mostrar y otros aspectos de la apariencia de la imagen en el diseñador o en el Editor de bloques.



- **Label (Etiqueta)**

Las etiquetas son componentes que se utilizan para mostrar texto.

Una etiqueta muestra el texto especificado en la propiedad Text. Otras propiedades, que se pueden establecer tanto en el diseñador como en el editor de bloques, controlan la apariencia y la colocación de ese texto.

- **ListPicker (Lista de selección)**

Los usuarios pueden tocar un componente de lista de selección para elegir un elemento de una lista de cadenas de texto.

Cuando un usuario toca el botón selector, se muestra una lista de elementos de texto para que el usuario pueda elegir. Los elementos de texto se puede especificar a través del diseñador o el Editor de bloques mediante el establecimiento de la propiedad ElementsFromString mediante la concatenación de los valores separados por comas (por ejemplo, selección1, selección2, selección3 ) o mediante el establecimiento de la propiedad Elements de una List (Lista) en el editor de bloques.

Otras propiedades, incluyendo TextAlignment y BackgroundColor , afectan a la apariencia del botón y si puede ser desplegado ( Enabled ).

- **Screen (Pantalla)**

La pantalla no aparece en la paleta igual que los restantes componentes, pero aparece automáticamente en el proyecto. Cada proyecto tiene una pantalla, llamada Screen1. Este nombre no puede ser cambiado.

- **PasswordTextBox (cuadro de texto para contraseña)**

Los usuarios introducen las contraseñas en un componente de cuadro de texto para contraseña, que oculta el texto que se ha escrito en él.

Un cuadro de texto para contraseña es el mismo que el componente TextBox ordinario, excepto en que no muestra los caracteres tecleados por el usuario.

Se puede obtener o establecer el valor del texto en la caja con la propiedad Text. Si Text está en blanco, puede utilizar la propiedad Hint para proporcionar al usuario una sugerencia de lo que puede escribir. La Hint aparece como texto débil en el cuadro.

El componente de cuadro de texto para contraseña se utiliza generalmente con un componente de botón. El usuario pulsa el botón después de la introducción de texto.

- **TextBox**

Los usuarios pueden introducir texto en un componente de cuadro de texto.

El valor del texto inicial o el que el usuario haya introducido en un componente de cuadro de texto está en la propiedad Text. Si Text está en blanco, puede utilizar la propiedad Hint para proporcionar al usuario una sugerencia de lo que escribir. La sugerencia aparece como texto débil en el cuadro.

La propiedad MultiLine determina si el texto puede tener más de una línea. Para una caja de una única línea de texto, el teclado se cerrará automáticamente cuando el usuario



presiona la tecla Listo. Para cerrar el teclado para cuadros de texto de varias líneas, la aplicación debe utilizar el método HideKeyboard o confiar en que el usuario pulse la tecla Atrás.

La propiedad NumbersOnly restringe al teclado numérico, de modo que sólo puedan introducirse números.

Otras propiedades afectan a la apariencia del cuadro de texto (TextAlignment , BackgroundColor , etc) y si se puede utilizar (Enabled).

Los cuadros de texto se suelen utilizar con el componente Button , para que el usuario haga clic en el botón cuando se haya completado la entrada del texto.

Si el texto introducido por el usuario no se debe mostrar, usar PasswordTextBox en su lugar.

- **Camera (Cámara).**

Se puede usar el componente cámara para tomar una foto con el teléfono.

La cámara es un componente no visible, que toma una fotografía con la cámara del dispositivo. Después de tomar la fotografía, la ruta al archivo en el teléfono que contiene la imagen está disponible como un argumento del evento AfterPicture. La ruta puede ser utilizada, por ejemplo, como la propiedad imagen de un componente de imagen.

- **ImagePicker (Selector de imagen).**

Use un componente selector de imágenes para elegir una imagen de la galería de imágenes.

Un selector de imágenes es una especie de botón. Cuando el usuario pulsa un selector de imágenes, aparece la galería de imágenes del dispositivo, y el usuario puede elegir una imagen. Después de que el usuario elige una imagen, la propiedad ImagePath nos proporciona la ruta de la imagen. A continuación, se puede utilizar ese resultado, por ejemplo, para establecer la imagen de un botón.

- **Player (Reproductor de audio).**

Utilizar un componente player para reproducir un archivo de audio o video, o para que vibre el teléfono.

Player es un componente no visible, que reproduce audio o vídeo y controles de vibración del teléfono. El nombre de un archivo multimedia se especifica en su propiedad origen, que se puede establecer en el Diseñador o en el Editor de bloques. La duración del tiempo de vibración se especifica en el editor de bloques en milisegundos (milésimas de segundo). Utilizar un componente player para reproducir archivos de sonido largos, archivos de vídeo y hacer vibrar el teléfono. Para reproducir archivos de sonido cortos, tales como efectos de sonido, use un componente sound en su lugar.

- **Sound (sonido).**

Utilizar un componente de sonido para reproducir un archivo de audio, o para que vibre el teléfono.



El sonido es un componente no visible, que reproduce archivos de sonido y vibra el número de milisegundos (milésimas de segundo) que se indica en el Editor de bloques. El nombre del archivo de sonido se puede especificar en el diseñador o en el Editor de bloques.

Este componente es el mejor para archivos de sonido cortos, tales como efectos de sonido, mientras que el componente player es más eficiente para sonidos largos, tales como las canciones.

#### - **VideoPlayer (Reproductor de vídeo)**

Utilizar un componente VideoPlayer para reproducir un archivo de vídeo.

Reproductor de vídeo es un componente multimedia que reproduce vídeos. Un reproductor de vídeo aparece en su aplicación como un rectángulo. Si el usuario pulsa el rectángulo, los controles multimedia aparecen: reproducción / pausa, saltar hacia delante y saltar hacia atrás. Su aplicación puede controlar el comportamiento de la reproducción mediante una llamada a los métodos Start , Pause , y SeekTo.

Los archivos de vídeo debe estar en Windows Media Video (. Wmv), 3GPP (.3 gp), o MPEG-4 (. Mp4).

App Inventor acepta archivos de vídeo de hasta 1 MB de tamaño y limita el tamaño total de una aplicación a 5 MB, por lo que no todos los archivos de video pueden utilizarse. Si sus archivos de medios de comunicación son demasiado grandes, es posible que obtenga errores al empaquetar o instalar la aplicación, en cuyo caso se deben reducir el número de archivos multimedia o sus tamaños. Ciertos software de edición de vídeo, como Windows Movie Maker o iMovie de Apple, puede ayudar a reducir el tamaño de los vídeos, acortándolos o volviendo a codificarlos en formatos más compactos.

#### - **Ball (bola)**

Un componente de bola es un tipo particular de sprite (objeto animado) que se parece a una pelota.

Una bola es un sprite redondo que está contenido en un lienzo, puede reaccionar a toques, arrastres, interactuar con otros sprites (sprites de imagen y otras bolas) y con el borde de la tela, y moverse de acuerdo a sus propiedades. Por ejemplo, para que una bola se mueva 4 píxeles en la parte superior de un lienzo cada 500 milisegundos (medio segundo), se puede establecer la propiedad Speed a 4, la propiedad Interval a 500, la propiedad Heading a 90 (grados), y establecer la propiedad Enabled a verdadero. Puede cambiar éstas y otras propiedades para modificar el comportamiento de la bola.

La diferencia entre una bola y un sprite de imagen es que este último puede obtener su apariencia de una imagen, mientras que el aspecto de una bola sólo puede ser cambiado variando sus propiedades PaintColor y Radius.

#### - **ImageSprite (Sprite de imagen)**

Un componente sprite de imagen es un objeto animado que puede interactuar con un lienzo, las bolas, y otros objetos de imagen.



Un sprite de imagen es un objeto animado que está contenido en un lienzo, puede reaccionar a toques, arrastres, interactuar con otros sprites (sprites de imagen y otras bolas) y con el borde de la tela, y moverse de acuerdo a sus propiedades. Por ejemplo, para tener un movimiento de 10 píxeles a la izquierda cada segundo, se puede establecer la propiedad Speed a 10, la propiedad Interval a 1000 (milisegundos), la propiedad Heading a 180 (grados), y establecer la propiedad Enabled a verdadero. Puede cambiar éstas y otras propiedades para modificar el comportamiento del sprite imagen.

- **AccelerometerSensor (Sensor acelerómetro).**

Este componente recibe señales del acelerómetro del dispositivo Android, que detecta sacudidas y mide la aceleración en tres dimensiones. La aceleración se mide en unidades del SI ( $m/s^2$ ). Si el dispositivo permanece apoyado en reposo sobre su parte posterior, la aceleración Z será de aproximadamente  $9,8 m/s^2$ .

El componente produce tres valores

- XAccel : Positivo cuando el dispositivo se inclina hacia la derecha (es decir, su lado izquierdo se levanta), y negativo cuando el dispositivo se inclina hacia la izquierda (su borde derecho se levanta).
- YAccel: positivo, cuando la parte inferior del dispositivo (la del micrófono) se eleva, y negativa cuando su parte superior (la del auricular) se eleva.
- ZAccel: Positivo cuando la pantalla esté mirando hacia arriba, y negativa cuando la pantalla está mirando hacia abajo.

- **LocationSensor (Sensor de posicionamiento).**

Este componente proporciona la ubicación del dispositivo Android, utilizando el GPS si está disponible o un método alternativo en caso contrario, como pueden ser las torres de telefonía móvil o las redes inalámbricas conocidas.

LocationSensor es un componente no visible que puede proporcionar información sobre la ubicación, incluyendo longitud, latitud, altitud (si el dispositivo posee esta capacidad), y la dirección. Este componente también puede proporcionar geocodificación, la conversión de una dirección dada (no necesariamente la actual) a una latitud (con el método LatitudeFromAddress) y una longitud (con el método LongitudeFromAddress).

Para funcionar, el componente debe tener la propiedad Enabled establecida, y el dispositivo debe tener habilitada la detección de localización a través de satélites GPS (hallándose con cobertura del sistema) o un método alternativo.

- **OrientationSensor (Sensor de orientación).**

Puede utilizarse el componente sensor de orientación para determinar la orientación espacial del teléfono.

El sensor de orientación es un componente no visible, que informa de los siguientes tres valores, en grados:



- Giro: 0 grados, cuando el dispositivo esté nivelado, que aumenta a 90 grados cuando el dispositivo se inclina sobre su lado izquierdo, y disminuye hasta -90 grados cuando el dispositivo se inclina sobre su lado derecho.
  - Inclinación: 0 grados, cuando el dispositivo esté nivelado, que aumenta a 90 grados cuando el dispositivo se inclina sobre su parte superior (la del auricular), aumentando aún más, hasta 180 grados, cuando se da la media vuelta. Del mismo modo, cuando el dispositivo se inclina sobre su parte inferior (la del micrófono), el ángulo disminuye a -90 grados, llegando a -180 grados, al dar la media vuelta.
  - Azimut: 0 grados, cuando la parte superior del dispositivo (la del auricular) está apuntando hacia el norte, 90 grados cuando se está apuntando al este, 180 grados cuando se está apuntando al sur, 270 grados cuando se apunta al oeste, etc.
- Estas mediciones suponen que el propio dispositivo no está en movimiento.

#### - **HorizontalArrangement (Alineamiento horizontal).**

Utilizar un componente de alineamiento horizontal para mostrar un grupo de componentes determinado de izquierda a derecha.

Este componente es un elemento de formato que permite colocar los componentes que deben mostrarse de izquierda a derecha.

Si, por el contrario, desea tener los componentes uno sobre otro, utilice VerticalArrangement en su lugar.

Con un HorizontalArrangement, los componentes están dispuestos a lo largo del eje horizontal, con un alineamiento vertical centrado.

Si la propiedad Height (altura) de un HorizontalArrangement se halla establecida en automático, su altura será determinada por el componente más alto que contenga cuya propiedad altura no esté establecida para llenar el patrón (su recuadro). Si la propiedad altura se establece en automático y sólo comprende elementos cuya altura está establecida para llenar el patrón, la altura final se calcula utilizando la altura automática de los componentes. Si la propiedad altura de un HorizontalArrangement está establecida en automático y no contiene nada, la altura será de 100.

Si la propiedad anchura de un HorizontalArrangement se establece en Automático, su anchura real estará determinada por la suma de las anchuras de los componentes que contenga. Si la propiedad anchura de un HorizontalArrangement se establece en automático, todos los componentes cuya anchura se halle establecida para llenar el patrón se comportarán como si su anchura hubiera estado establecida en automático.

Si la propiedad anchura de un HorizontalArrangement se establece para llenar el patrón o se especifica en píxeles, cualquier componente cuya anchura esté establecida para llenar el patrón ocupará la anchura no ocupada por otros componentes que compartan el mismo HorizontalArrangement.

#### - **TableArrangement (Alineamiento en tabla).**

Se puede utilizar un componente de alineamiento en tabla para mostrar un grupo de componentes de una forma tabular.





Este componente es un elemento de formato que coloca los componentes que deben mostrarse en forma de tabla.

En un `TableArrangement`, los componentes están dispuestos en una cuadrícula de filas y columnas, con no más de un componente visible en cada celda. Si varios componentes ocupan la misma celda, sólo el último será visible.

Dentro de cada fila, los componentes son alineados verticalmente en posición central.

La anchura de una columna está determinado por el mayor componente de esa columna. Al calcular la anchura de la columna, el ancho automático se utiliza para aquellos componentes cuya propiedad anchura se halla establecida para llenar la celda. Sin embargo, cada componente siempre llenará el ancho total de la columna que ocupa.

La altura de una fila es determinada por el componente más alto en la fila cuya propiedad altura no esté establecida para llenar dicha fila. Si una fila contiene sólo componentes cuya altura se halle establecida para llenar la fila, la altura de la fila se calcula utilizando la altura automática de los componentes.

#### - **VerticalArrangement (Alineamiento vertical)**

Se puede utilizar un componente de alineamiento vertical para mostrar un grupo de componentes dispuestos de arriba hacia abajo, alineados a la izquierda de la pantalla.

Este componente es un elemento de formato en el cual pueden colocarse componentes que deban mostrarse uno debajo de otro. El primer componente se muestra en la parte superior, el segundo justo debajo, y así sucesivamente. Si, por el contrario, desea que los componentes se muestren uno junto a otro, utilice un componente `HorizontalArrangement` en su lugar.

En una `VerticalArrangement`, los componentes están dispuestos a lo largo del eje vertical, y el propio `VerticalArrangement` queda alineado a la izquierda. Si la propiedad `Width` (anchura) de un `VerticalArrangement` se halla establecida en automático, su anchura será determinada por el componente más ancho que contenga, cuya propiedad anchura no esté establecida para llenar el patrón (su recuadro). Si la propiedad anchura se establece en automático y sólo comprende elementos cuya anchura está establecida para llenar el patrón, la anchura final se calcula utilizando la anchura automática de los componentes. Si la propiedad anchura de un `VerticalArrangement` está establecida en automático y no contiene nada, la anchura será de 100. Si la propiedad altura de un `VerticalArrangement` se establece en Automático, su altura final estará determinada por la suma de las alturas de los componentes que contenga. Si la propiedad altura de un `VerticalArrangement` se establece en automático, todos los componentes cuya altura se halle establecida para llenar el patrón se comportarán como si su altura hubiera estado establecida en automático.

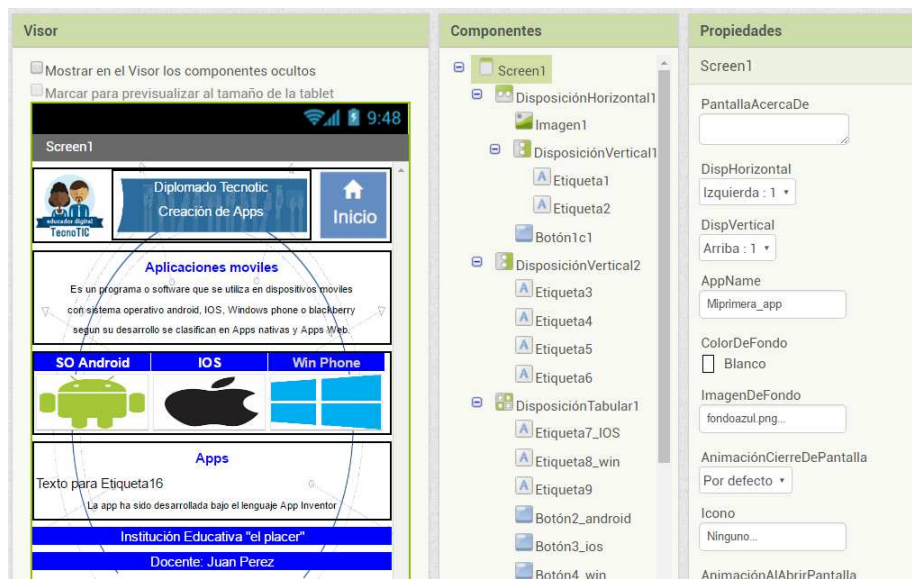
Si la propiedad altura de un `VerticalArrangement` se establece para llenar el patrón o se especifica en píxeles, cualquier componente cuya altura esté establecida para llenar el patrón ocupará la altura no ocupada por otros componentes que compartan el mismo `VerticalArrangement`.



# Actividad “personaliza la interfaz grafica de la App

Ingresa a la plantilla construida con el Formador llamada Plantilla\_app\_grupo# en ella encontrarás el siguiente diseño y bloques:

Screen1

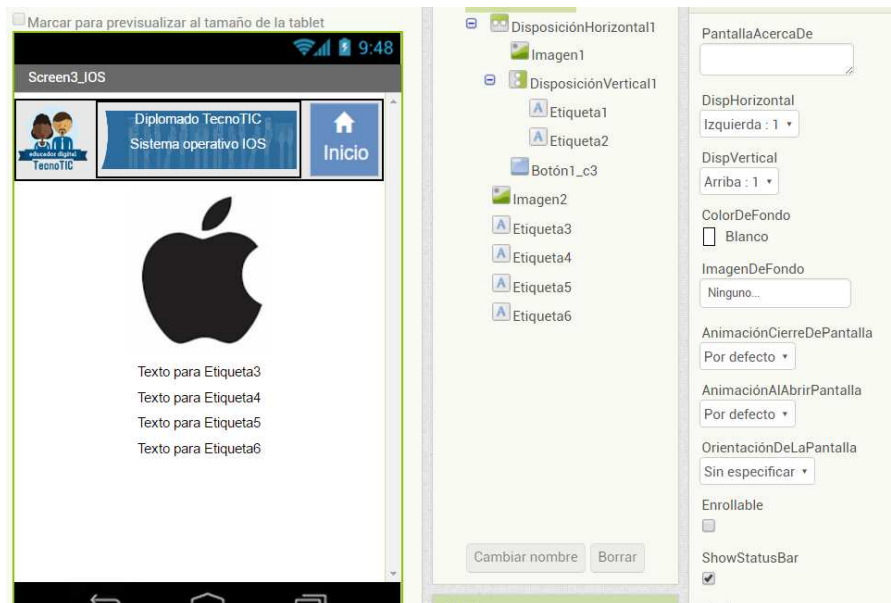


## Screen2

## Screen3



## Screen4



## Paso 1: Modifica el texto

Seleccionando los elementos Etiqueta en la sección “componentes” personaliza las características como dimensión, color, negrita, cursiva en “propiedades” de cada texto.

## Paso 2: Modifica la imagen

Carga en la sección medios las imágenes con las cuales deseas personalizar tu App y luego selecciona la imagen en “componentes” que desee modificar.

Al finalizar el taller deberás entregar la app con las modificaciones realizadas, para ello guarde la aplicación desde el menú superior “proyectos” – “Exportar a mi ordenador el proyecto (.aia) seleccionador” y suba este archivo a su carpeta del Drive.



## Desarrollo proyecto TIC - APP

En conjunto con el desarrollo de las guías para la construcción del semillero y la plantilla desarrollada ya está en la capacidad de iniciar la construcción de su aplicación móvil.

Para ello se les invita a que tome como referencia el siguiente repositorio de más de 100 aplicaciones construidas en diferentes temáticas.

<http://kio4.com/appinventor/>

