

TALLER AVANZADO DE APPINVENTOR

6, 7 y 8 de julio del 2015 en Ubrique, Cádiz

1. Introducción

En este tutorial se va a describir cómo utilizar la herramienta online *AppInventor* para construir una aplicación para dispositivos móviles Android. Esta aplicación hará uso de varias de las capacidades que ofrecen los dispositivos móviles.

2. Construcción y despliegue de la aplicación base

En primer lugar vamos a crear un nuevo proyecto para Android, el cual consistirá en una pantalla de bienvenida desde la cual tendremos acceso a las funcionalidades descritas en posteriores secciones.

2.1. Componentes utilizados

Sobre la pantalla principal *Screen1* necesitamos incluir botones de acción (*Button*) que nos permitan acceder a cada una de las partes de nuestra aplicación. En la siguientes figuras podemos observar los componentes utilizados y su disposición visual sobre la pantalla.



2.2. Desarrollo

Una vez creado el proyecto, los pasos a seguir son:

1. Configurar aspectos básicos de la aplicación, mediante la modificación de las siguientes propiedades de la pantalla principal:
 - a. Icono: *aulablog.jpeg* [subir fichero]
 - b. AppName: *AulaBlogApp*
 - c. Title: *AulaBlog Ubrique*
 - d. AboutScreen: *App creada en el evento AulaBlog*
 - e. BackgroundImage: *cartel.jpeg* [subir fichero]
 - f. AlignHorizontal: *center*
 - g. AlignVertical: *center*
2. Añadir un botón con nombre *ButtonFotos* y modificar las siguientes propiedades:
 - a. FontSize: 30
 - b. Text: *Fotos*
 - c. Shape: *rounded*
3. Añadir un botón con nombre *ButtonCuestionarios* y modificar las siguientes propiedades:
 - a. FontSize: 30
 - b. Text: *Encuestas*
 - c. Shape: *rounded*
4. Añadir el comportamiento necesario para que cuando el usuario haga clic en el botón *ButtonFotos* se abra la pantalla desde la que realizar y “retocar” imágenes.



5. Asociar clic del botón *ButtonCuestionarios* para que abra la pantalla desde la que se puede participar en cuestionarios



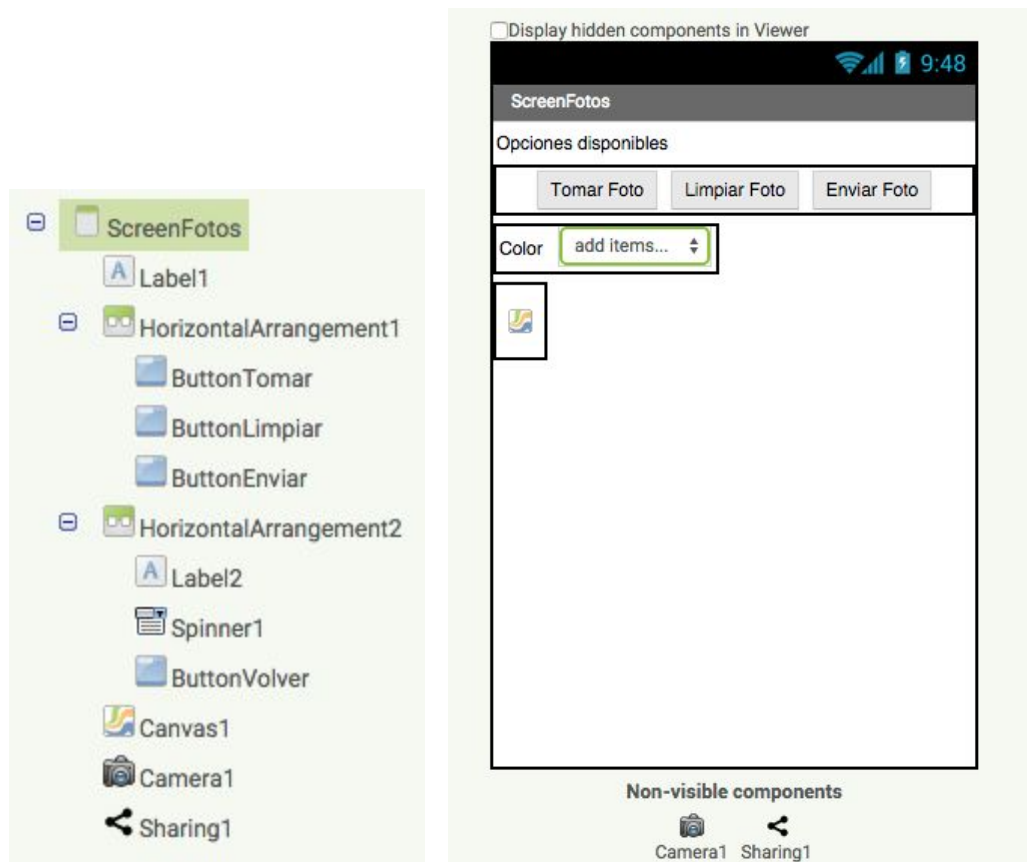
3. Editar y compartir fotos

En este punto, vamos a permitir que desde nuestra aplicación podamos tomar fotos y hacer ciertos “retoques”, para luego poder compartirlas por redes sociales.

3.1. Componentes utilizados

Para implementar esta funcionalidad necesitaremos una serie de componentes para presentar texto en pantalla (*Label*), organizar elementos horizontalmente

(*HorizontalArrangement*), botones de acción (*Button*), área de dibujo (*Canvas*), selector de opciones (*Spinner*), cámara de fotos (*Camera*) y compartir ficheros (*Sharing*). En la siguientes figuras podemos observar los componentes utilizados y su disposición visual sobre la pantalla.

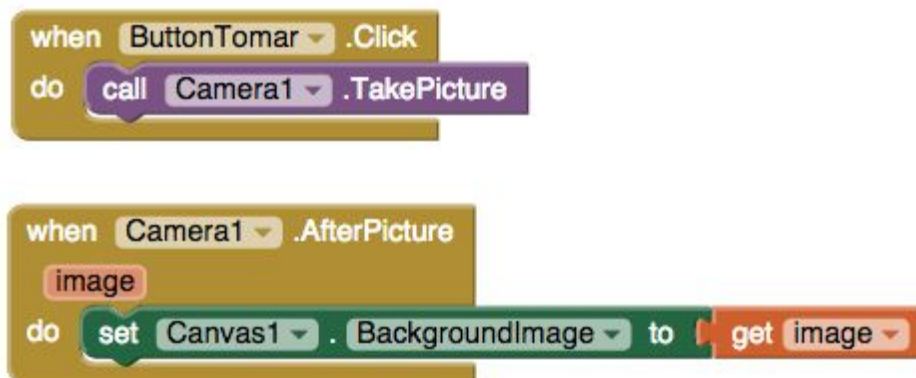


3.2. Desarrollo

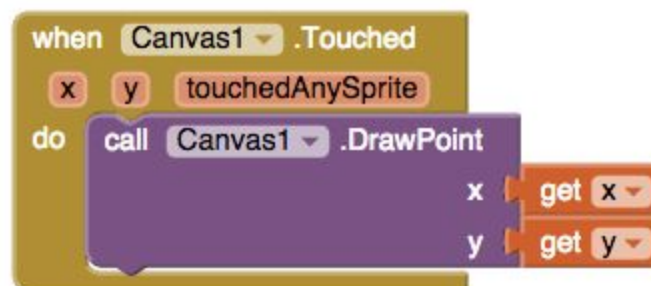
A continuación, se resumen los pasos a seguir para diseñar la pantalla y definir el comportamiento asociado:

1. Crear una nueva pantalla con nombre *ScreenFotos*
2. Crear una etiqueta con nombre *Label1* y configurar la siguiente propiedad:
 - a. Text: *Opciones disponibles*
3. Añadir un contenedor con nombre *HorizontalArrangement1* (para organizar horizontalmente unos botones) y configurar las siguientes propiedades:
 - a. AlignHorizontal: *Center*
 - b. Width: *Fill parent...*
4. Añadir un botón con nombre *ButtonTomarFoto* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. Text: *Tomar foto*
5. Añadir un botón con nombre *ButtonLimpiarFoto* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. Text: *Limpiar foto*
6. Añadir un botón con nombre *ButtonEnviarFoto* dentro del contenedor anterior y configurar la siguiente propiedad:

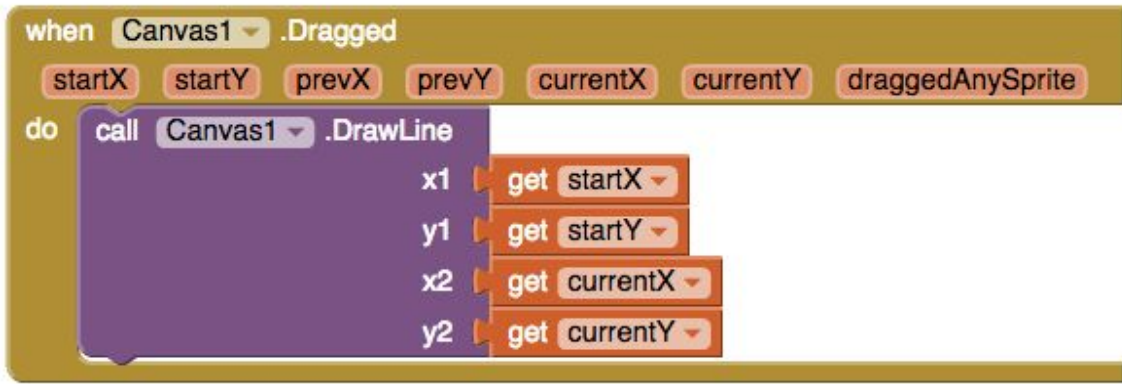
- a. Text: *Enviar foto*
7. Añadir un contenedor con nombre *HorizontalArrangement1* (para organizar horizontalmente unos botones) y configurar las siguientes propiedades:
 - a. AlignHorizontal: *Center*
 - b. Width: *Fill parent...*
8. Añadir una etiqueta con nombre *Label2* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. Text: *Color*
9. Añadir un selector de opciones con nombre *Spinner1* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. ElementsFromString: *Negro, Rojo, Verde, Azul*
10. Añadir un botón con nombre *ButtonVolver* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. Text: *Volver*
11. Añadir un área de dibujo con nombre *Canvas1*
12. Añadir un componente para manejar la cámara de fotos con nombre *Camera1*.
13. Añadir un componente para compartir ficheros con nombre *Sharing1*.
14. Añadir el comportamiento necesario para que cuando el usuario haga clic en el botón con texto *ButtonTomarFoto* se abra la aplicación de la cámara de fotos y al retornar la foto se ponga la instantánea como imagen de fondo del área de dibujo *Canvas1*.



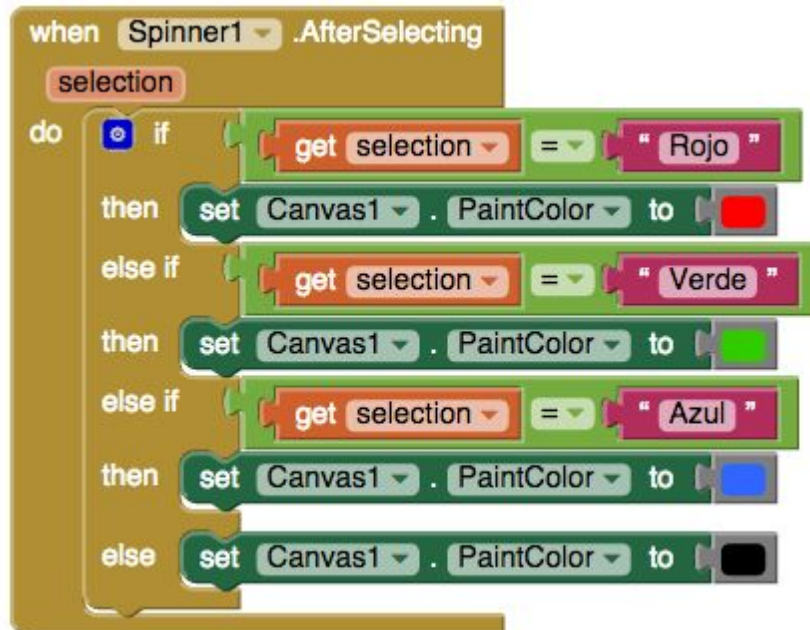
15. Añadir el comportamiento necesario para que cuando el usuario toque cualquier lugar de la fotografía del *Canvas1* aparezca un punto de color negro.



16. Añadir el comportamiento necesario para que cuando el usuario arrastre el dedo sobre la imagen del *Canvas1* se pinte una línea.



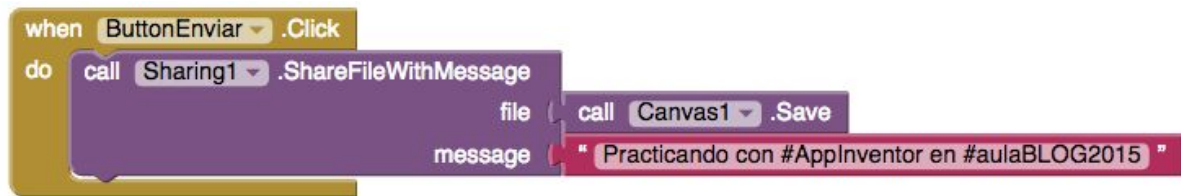
17. Añadir el comportamiento necesario para que el color de las líneas y puntos a pintar en el *Canvas1* corresponda con el elegido por el usuario mediante el selector *Spinner1*.



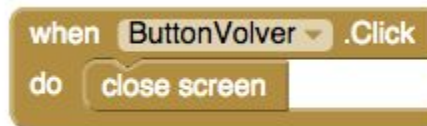
18. Añadir el comportamiento requerido para que cuando el usuario pulse el botón *ButtonLimpiar* se borren los retoques realizados.



19. Añadir el comportamiento necesario para que cuando el usuario pulse el botón *ButtonEnviar* se envíe la fotografía, junto con los retoques añadidos por alguna red social.



20. Añadir el comportamiento necesario para que cuando el usuario pulse el botón *ButtonVolver* se cierre la pantalla y se vuelva al menú principal.

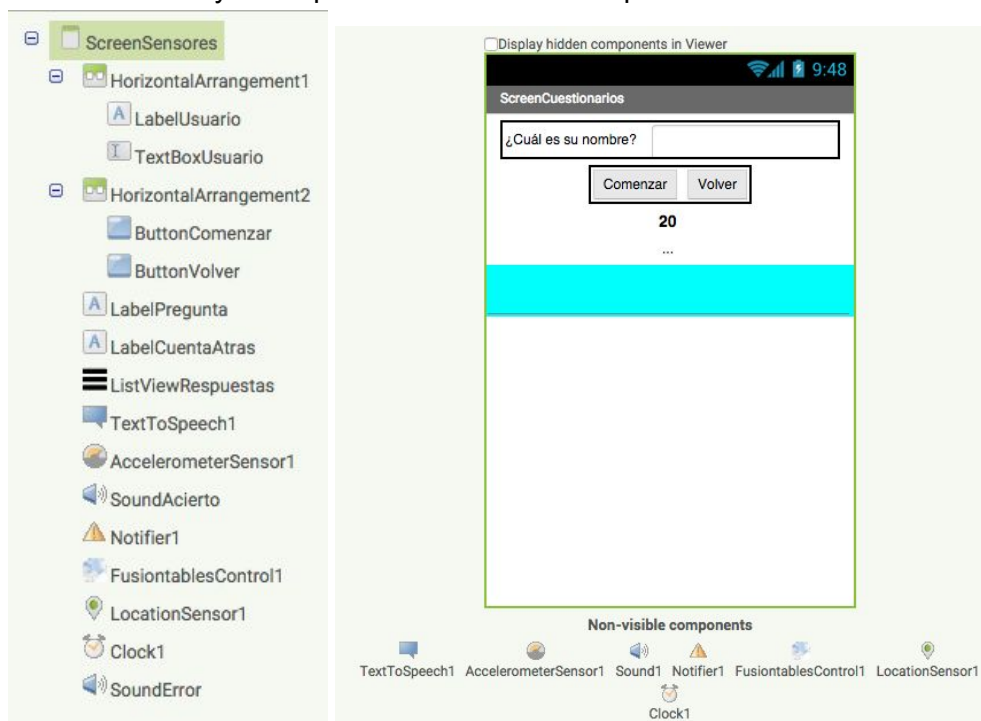


4. Cuestionarios

En este punto, vamos a permitir que desde nuestra aplicación podamos participar en cuestionarios interactivos.

4.1. Componentes utilizados

Para implementar esta funcionalidad necesitaremos una serie de componentes para presentar texto en pantalla (*Label*), organizar elementos horizontalmente (*HorizontalArrangement*), botones de acción (*Button*), listado de opciones (*ListView*), conversor de texto a voz (*TextToSpeech*), sensores de aceleración (*Accelerometer*), vibración con el componente (*Sound*), etc.. En la siguientes figuras podemos observar los componentes utilizados y su disposición visual sobre la pantalla.



3.2. Desarrollo

A continuación, se resumen los pasos a seguir para diseñar la pantalla y definir el comportamiento asociado:

1. Crear una nueva pantalla con nombre *ScreenCuestionarios*
2. Añadir un contenedor con nombre *HorizontalArrangement1* (para organizar horizontalmente unos botones) y configurar las siguientes propiedades:
 - a. *AlignHorizontal*: *Center*
 - b. *Width*: *Fill parent...*
3. Añadir una etiqueta con nombre *LabelUsuario* dentro del contenedor anterior y configurar la siguiente propiedad.
 - a. *Text*: *¿Cuál es su nombre?*
4. Añadir una caja de texto con nombre *TextBoxUsuario* dentro del contenedor anterior y configurar la siguiente propiedad
 - a. *Hint*: *Escriba su nombre de usuario*
5. Añadir un contenedor con nombre *HorizontalArrangement2* (para organizar horizontalmente unos botones) y configurar las siguientes propiedades:
 - a. *AlignHorizontal*: *Center*
 - b. *Width*: *Fill parent...*
6. Añadir un botón con nombre *ButtonComenzar* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. *Text*: *Comenzar*
7. Añadir un botón con nombre *ButtonVolver* dentro del contenedor anterior y configurar la siguiente propiedad:
 - a. *Text*: *Volver*
8. Añadir una etiqueta con nombre *LabelPregunta* y configurar la siguiente propiedad.
 - a. *Text*: *(vacío)*
 - b. *FontBold*: *true*
9. Añadir una etiqueta con nombre *LabelCuentaAtras* y configurar la siguiente propiedad:
 - a. *Text*: *(vacío)*
 - b. *FontItalic*: *true*
10. Añadir una componente vista de listado de opciones con nombre *ListViewRespuesta* y configurar la siguiente propiedad:
 - a. *BackgroundColor*: *Cyan*
11. Añadir mediante la vista de bloques una lista de cuestiones, compuestas cada una por su correspondiente pregunta, 4 respuestas posibles y un número indicando cuál de las respuestas es la correcta.



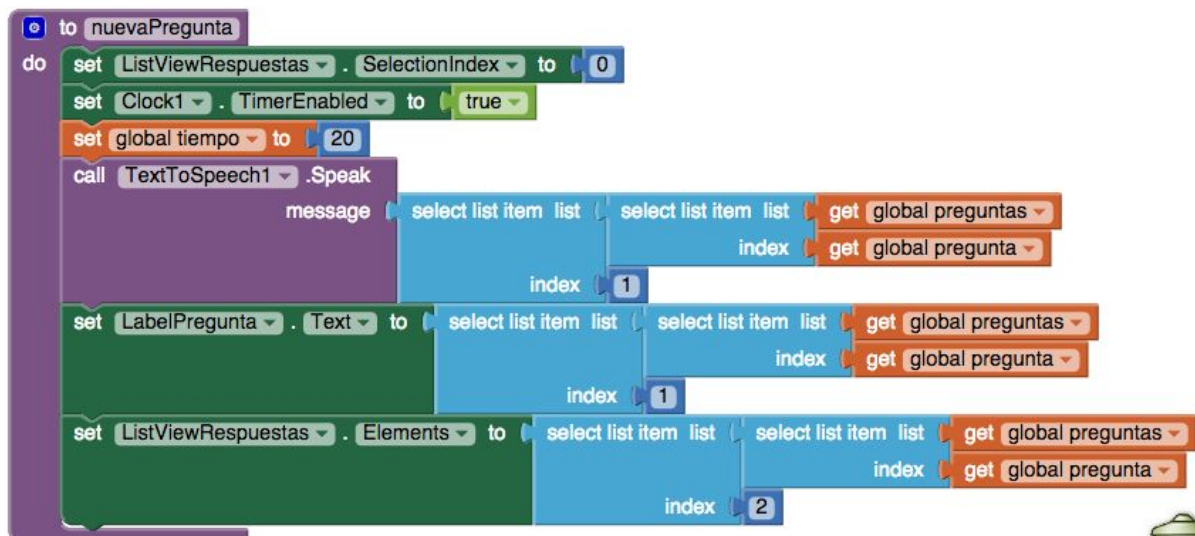
12. Inicializar varias variables globales para controlar el número de la pregunta actual, la cuenta atrás, el número de aciertos y el número de fallos acumulados.



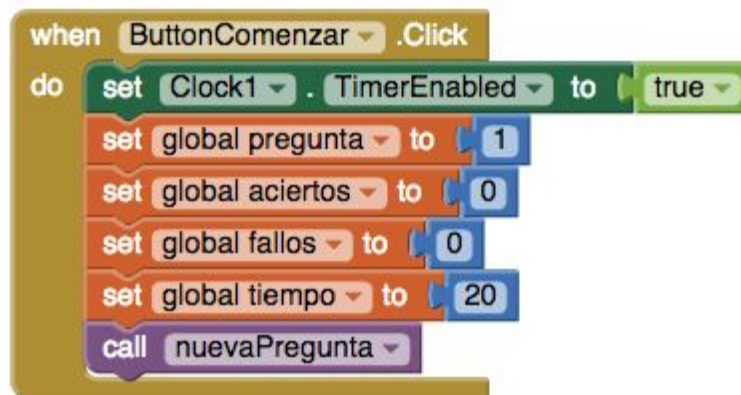
13. Añadir un componente con nombre *Clock1* para controlar la cuenta atrás y configurar la siguiente propiedad

a. *TimerInterval: 1000*

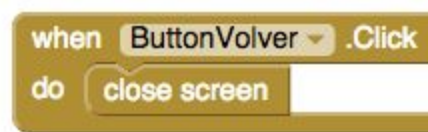
14. Crear un procedimiento denominado *nuevaPregunta*, que será invocado cada vez que haya que presentar una nueva pregunta al usuario. Sus funcionalidades son inicializar el índice seleccionado del *ListViewRespuestas*, inicializar el contador de cuenta atrás, presentar en pantalla el texto de la pregunta correspondiente, reproducir dicha pregunta mediante voz y mostrar las respuestas asociadas. Se define de la siguiente manera:



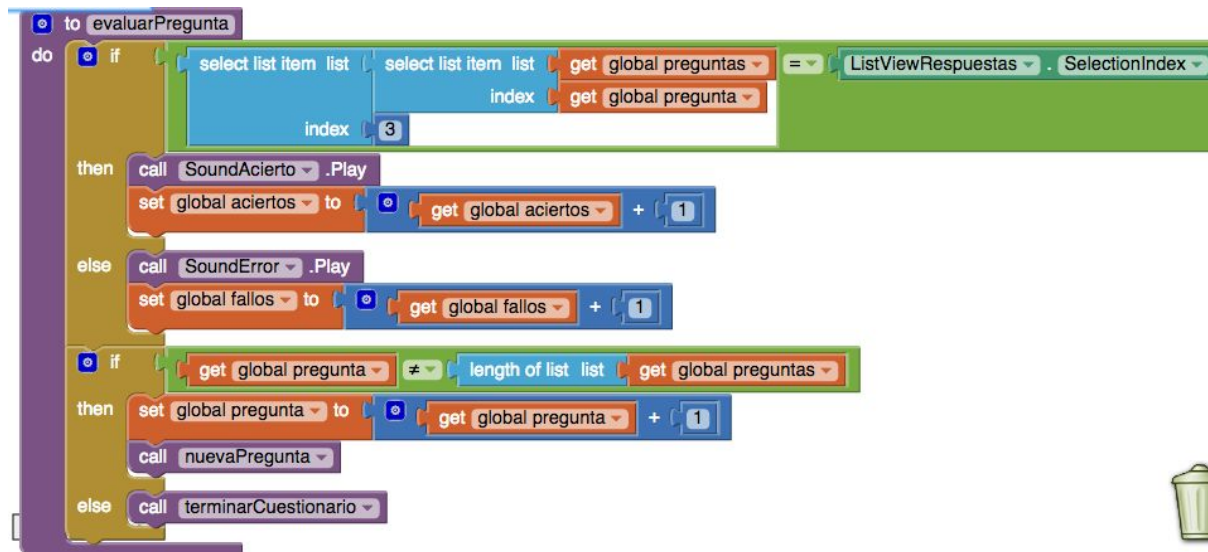
15. Añadir el comportamiento requerido para comenzar el cuestionario al pulsar el botón *ButtonComenzar*, de modo que se invoque al procedimiento creado anteriormente. Asimismo, se inicializan las variables globales y el temporizador.



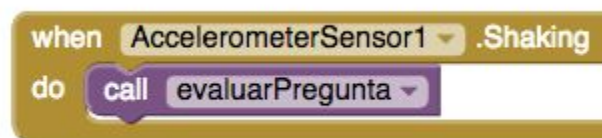
16. Añadir el comportamiento requerido para volver al menú principal cuando el usuario pulse el botón *ButtonVolver*.



17. Añadir un componente sonido con nombre *SoundAcierto* y configurar la siguiente propiedad:
- a. Source: *acierto.mp3* [subir fichero]
18. Añadir un componente sonido con nombre *SoundError* y configurar la siguiente propiedad:
- a. Source: *error.mp3* [subir fichero]
19. Añadir un componente con nombre *TextToSpeech1* para convertir texto en voz y configurar las siguientes propiedades:
- a. Country: *ESP*
 - b. Language: *ES*
20. Crear un procedimiento denominado *evaluarPregunta*, el cual se invocará cada vez que se considere como definitiva una respuesta, ya sea porque previamente el usuario sacudió el móvil o bien porque expiró el temporizador. Sus responsabilidades son:
- a. Comprobar si la respuesta marcada coincide con la correcta.
 - i. En caso afirmativo, se incrementa el número de aciertos y se emite un sonido (acierto).
 - ii. En caso negativo, se incrementa el número de fallos y se emite otro sonido (error).
 - b. Comprobar si se trata de la última pregunta del cuestionario.
 - i. En caso de que no sea la última se incrementa la variable controladora del orden y se invocará al procedimiento *nuevaPregunta*.
 - ii. En caso de que sea la última se invocará al procedimiento *terminarCuestionario* (que posteriormente será explicado).



21. Añadir un componente sensor de aceleración con nombre *AccelerometerSensor1* para detectar movimientos en el móvil.
22. Añadir el comportamiento requerido para que cuando el usuario “agite” el móvil se considere definitiva la respuesta seleccionada.

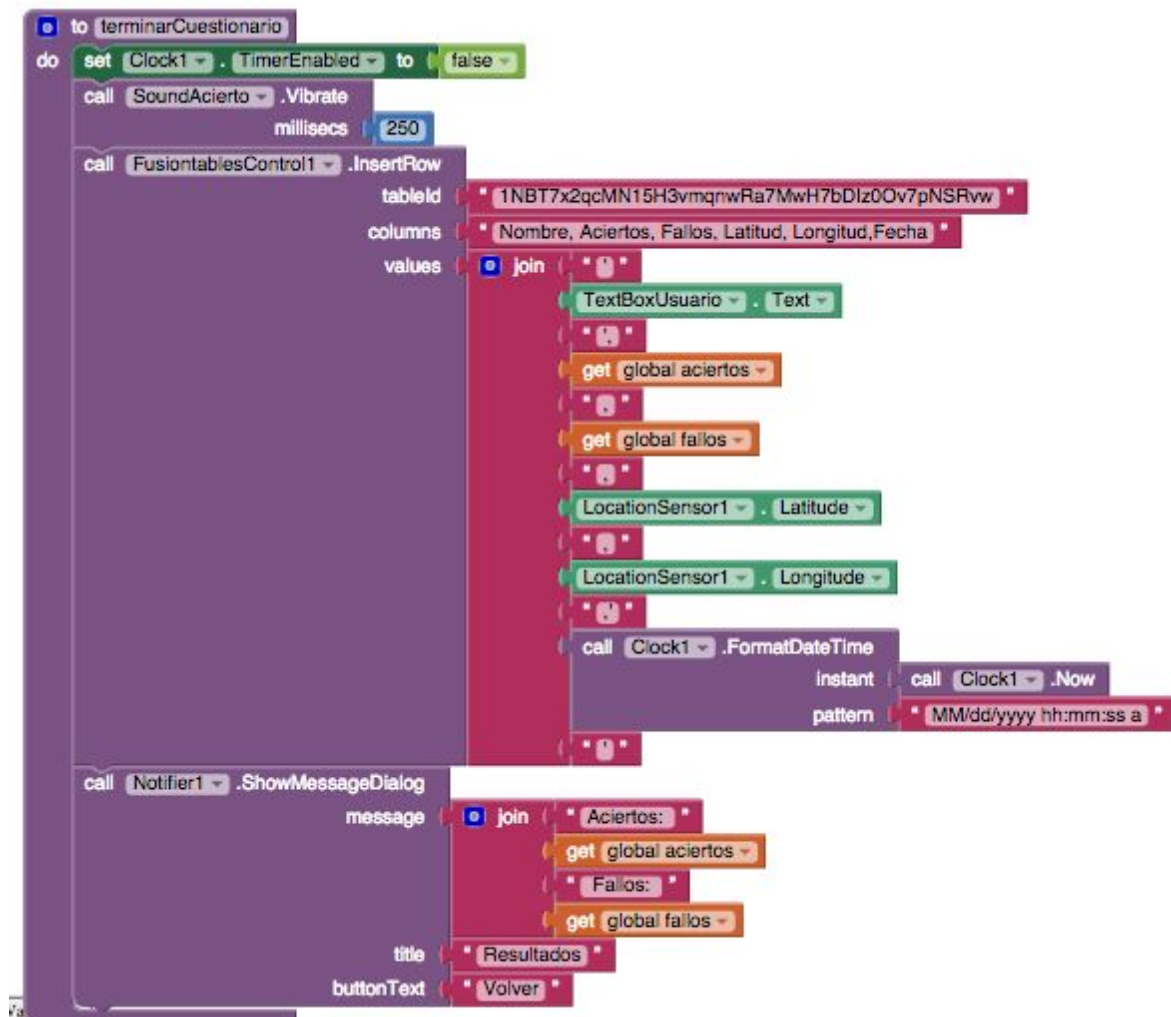


23. Añadir el comportamiento requerido para manejar el evento del componente *Clock1*. En primer lugar se actualizará la etiqueta *LabelCuentaAtras*, mostrando el valor actual del temporizador y se decrementará en una unidad. Luego, en caso que el temporizador llegue a cero, se invocará al procedimiento *evaluarPregunta*.



24. Añadir un componente para conocer la ubicación GPS con nombre *LocationSensor1*.
25. Añadir un componente con nombre *Notifier1* para abrir ventanas modales.
26. Añadir un componente con nombre *FusionTablesControl1* para enviar datos a una tabla dinámica de *Google FusionTables* (ver sección 3.3), estableciendo las siguientes propiedades:
 - a. ApiKey: *AlzaSyDL9s7r6Zlr9DN47_kNIlzRcm2JhWxy7ZU*
 - b. KeyFile: *ruizrube-cd84632c4ea8.p12* [subir fichero]

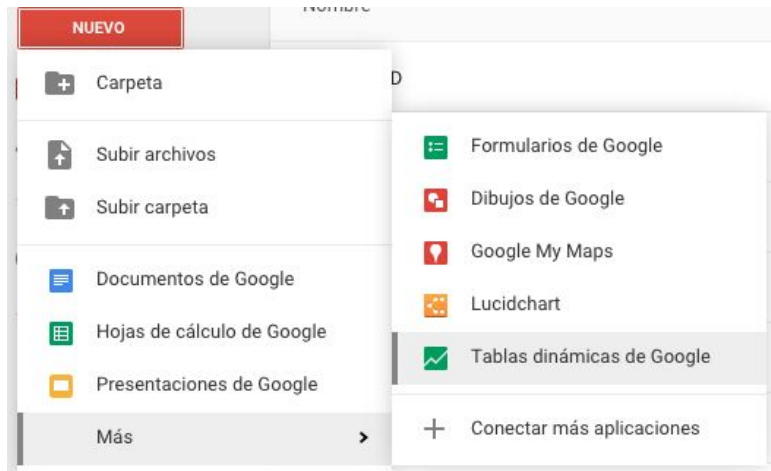
- c. ServiceAccountEmail:
1075849932338-n26pqlvqfea3dspaebf52vnacch77nhf@developer.gserviceaccount.com
 - d. UseServiceAuthentication: true
27. Crear un procedimiento denominado *terminarCuestionario*, el cual se invocará una vez que se haya evaluado la última pregunta del cuestionario. Sus responsabilidades son:
- a. Detener el temporizador
 - b. Emitir una pequeña vibración
 - c. Enviar un nuevo registro a una tabla remota del servicio *FusionTables* de Google, para almacenar el nombre del usuario, el número de aciertos, el número de fallos, la ubicación geográfica y la fecha actual.
 - i. tableID: 1NBT7x2qcMN15H3vmqnrwRa7MwH7bDlz0Ov7pNSRvw
 - d. Mostrar un mensaje modal indicando el número total de aciertos y fallos



3.3. Configuración del servicio de FusionTables

En esta sección se describen los pasos para preparar una tabla dinámica donde almacenar los datos generados desde la aplicación Android.

1. Crear nueva tabla dinámica mediante Google Drive.



2. Editar las columnas necesarias

Columns

NEW

- 1 Nombre
Text
- 2 Aciertos
Number
- 3 Fallos
Number
- 4 **Latitud**
Location
- 5 Longitud
Location
- 6 Fecha
Date/Time

Column name:

Description:

Type: **Location** ☐ Validate data [Learn more](#)

☒ Two column location

Latitude:

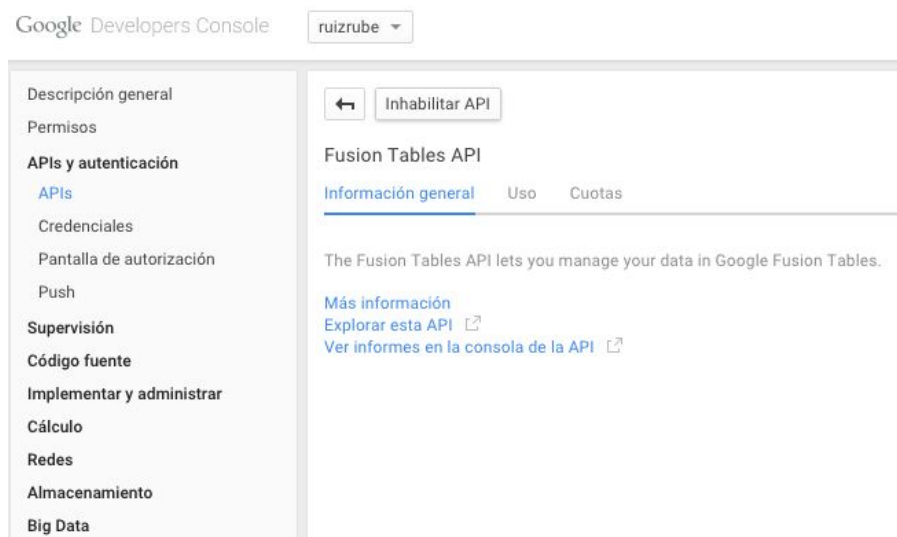
Longitude:

Format: **None**

List of drop-down items:

Custom properties JSON:

3. Entrar en la consola de desarrolladores de Google y activar el API de Fusion Tables.



4. Crear una nueva clave pública para el API

Acceso a API pública

El uso de esta clave no requiere consentimiento ni acciones por parte del usuario ni concede acceso a la información de la cuenta. Además, no se utiliza para la autorización.

[Más información](#)

[Crear clave nueva](#)

Clave para las aplicaciones Android

Clave de la API	AlzaSyDL9s7r6Zlr9DN47_kNIlzRcm2JhWxy7ZU
Aplicaciones Android	Se permite cualquier aplicación
Fecha de activación	5 jul. 2015 8:05:00
Activado por	ruizrube@gmail.com (tú)

[Editar aplicaciones Android permitidas](#)

[Volver a generar la clave](#)

[Eliminar](#)

5. Crear un nuevo certificado para autenticación

OAuth

OAuth 2.0 permite a los usuarios compartir datos concretos contigo (por ejemplo, listas de contactos), pero manteniendo la privacidad de sus nombres de usuario, contraseñas y otros datos.

[Más información](#)

[Crear ID de cliente nuevo](#)

Cuenta de servicio

ID de cliente	1075849932338-n26pqlvqfea3dspaebf52vnacch77nhf.apps.googleusercontent.com
Dirección de correo electrónico	1075849932338-n26pqlvqfea3dspaebf52vnacch77nhf@developer.gserviceaccount.com
Huellas digitales de certificados	4718dd8c5168cc32d7c5b510fce7b84c306ec9be - Eliminar cd84632c4ea86cb7b9743ac51c6e59282abd6862 - Eliminar

[Generar una clave JSON nueva](#)

[Generar una clave P12 nueva](#)

[Eliminar](#)

6. Compartir la tabla dinámica (sólo lectura) de forma pública y para la dirección de correo electrónica obtenida en el paso anterior para poder editar los datos.

Configuración de uso compartido

Enlace para compartir

<https://www.google.com/fusiontables/DataSource?docid=1NBT7x2qcMN15H3vmqnrwl>

Comparte este enlace a través de:

Quién tiene acceso

Público en la Web: cualquier usuario de Internet puede encontrar los elementos y consultarlos. [Cambiar...](#)

Ivan Ruiz Rube (tú)
ruizrube@gmail.com **Es propietario**

1075849932338-n26pqlvqfea3dspaebf52vn... [Puede editar](#)

Invitar a personas:

[Puede editar](#)

Los editores pueden añadir a personas y cambiar los permisos. [Cambiar](#)

[Ok](#)

7. Obtener el identificador público de la tabla que será necesaria a la hora de programar en AppInventor el envío de los datos a la tabla dinámica.

About this table

Name

AulaBlog

Visibility and reuse

Public

Anyone on the internet can find and access. No sign-in required

Download allowed

Reuse license

Unknown

Protected map layer

Maps client ID not set up

[Set up now](#)

Id

1NBT7x2qcMN15H3vmqnrRa7MwH7bDIz0Ov7pNSRvw

Done

[Edit table information](#)

- Configurar los parámetros necesarios en el componente *FusionTablesControl* de *AppInventor* de la siguiente manera:

Properties

FusiontablesControl1

ApiKey

AlzaSyDL9s7r6Zlr9DN47_

KeyFile

ruizrube-cd84632c4ea8.p

Query

show tables

ServiceAccountEmail

1075849932338-n26pqIvc

UseServiceAuthentication

☒

- A partir de este momento, la aplicación Android irá enviando los datos de los cuestionarios a la tabla dinámica.

Nombre	Aciertos	Fallos	Latitud	Longitud	Fecha
paco	0	2	36.60436	-6.26865	
luis	1	1	36.5063	-6.2658	
	0	2	0	0	
carlitos	1	1	36.50629	-6.2658	07/06/2015 01:58:17 a. m.
pepe	2	1	36.604	-6.26896	
juanito	0	2	36.60459	-6.26833	
juanito	0	3	36.60444	-6.26853	
juanito	0	4	36.60444	-6.26853	
ivan	0	2	36.60374	-6.26918	
sonia	1	1	36.67854	-5.44663	07/06/2015 04:00:39 p.m.
pepe	1	1	36.60408	-6.26887	
luis	2	0	36.50629	-6.2658	
carlitos	1	2	36.50628	-6.26578	07/06/2015 02:04:51 a. m.
elena	1	1	36.60414	-6.26873	