

Trabajando con sonidos

¿ Qué vamos a construir ?.....	1
¿ Qué aprenderé ?.....	2
Diseñar los componentes.....	2
Crear el teclado.....	3
Agregar el componente Sound.....	4
Decirle a Android que incorpore los sonidos.....	7
PRUEBA!!!!.....	7
Agregar las notas restantes.....	8
PRUEBA!!!!.....	10
Grabar y reproducir las notas.....	10
Grabar y reproducir las notas.....	12
PRUEBA!!!!.....	16
PRUEBA!!!!.....	18
Variaciones.....	18

Hoy nos parece mentira que la primera vez que se utilizó la tecnología para reproducir y grabar música fuera en 1878, cuando **Edison** patentó el **fonógrafo**. Desde entonces, hemos avanzado mucho contando con sintetizadores de música, mezclas, CD, teléfonos capaces de desarrollar música y listas de reproducción distribuidas por Internet. En este capítulo nos adentraremos en el mundo de la música y construiremos un aplicación llamada **Xylophone** capaz de reproducir y grabar música.

¿ Qué vamos a construir ?

La aplicación es un trabajo del miembro del equipo de AppInventor **Liz Looney**. Con ella podremos:

- Reproducir **cinco notas diferentes** tocando botones de colores en la pantalla.
- Presionar el botón **Play** (Reproducir) para que la aplicación repita las notas que hemos tocado.

- Presionar el botón **Reset** (Borrar) para que el programa olvide las notas que hemos tocado y podamos reproducir una nueva canción.

¿ Qué aprenderé ?

En esta aplicación aprenderás:

- Utilizar un único componente **Sound** para reproducir distintos archivos de audio.
- Emplear el componente **Clock** para medir y forzar retardos entre las acciones.
- Decidir cuándo crear un procedimiento.
- Generar un procedimiento que se llame a sí mismo(Recursividad).
- Los usos avanzados de las listas, incluyendo la capacidad de agregar elementos, acceder a ellos y borrar su contenido.

Diseñar los componentes

Crea un nuevo proyecto llamado **Xylophone**. Escribe el mismo nombre en la barra de título de la pantalla. Abre **Blocks Editor** y conecta tu teléfono al ordenador o activa el emulador. La aplicación tiene 13 componentes, 8 de los cuales pertenecerán al teclado. Añade los componentes que aparecen en la siguiente tabla:

Tipo de componente	Grupo de Palette	Cómo lo llamaremos	Finalidad
<i>Button</i>	User Interface	Boton1	Reproducir la nota Do(C) de la primera octava.
<i>Button</i>	User Interface	Boton2	Reproducir la nota Re (D).
<i>Button</i>	User Interface	Boton3	Reproducir la nota Mi (E).
<i>Button</i>	User Interface	Boton4	Reproducir la nota Fa (F).

<i>Button</i>	User Interface	Boton5	Reproducir la nota Sol (G).
<i>Button</i>	User Interface	Boton6	Reproducir la nota La (A).
<i>Button</i>	User Interface	Boton7	Reproducir la nota Si (B).
<i>Button</i>	User Interface	Boton8	Reproducir la nota Do (C) de la segunda octava.
<i>Sound</i>	Media	<i>Sound1</i>	Reproduce las notas.
<i>Button</i>	User Interface	Play	Reproduce las canciones.
<i>Button</i>	User Interface	Reset	Borra la memoria de la aplicación.
<i>HorizontalArrangement</i>	Layout	<i>HorizontalArrangement1</i>	Coloca los botones Play y Reset juntos.
<i>Clock</i>	Sensors	Clock1	Registra los retardos que hay entre las notas.

Crear el teclado

La interfaz de usuario incluirá un teclado con 8 notas, con una escala que tiene 7 notas musicales. Nuestro xilófono irá desde el **Do** de la primera octava hasta el **Do** de la segunda. En esta sección crearemos el teclado musical. Vamos a empezar por generar las dos primeras teclas de nuestro xilófono, que implementaremos como botones.

1. Arrastra un elemento **Button** desde **UserInterface** hasta el **Viewer**. Tendrá un aspecto alargado y un color magenta, igual que la tecla correspondiente de un xilófono.

Configuraremos sus propiedades:

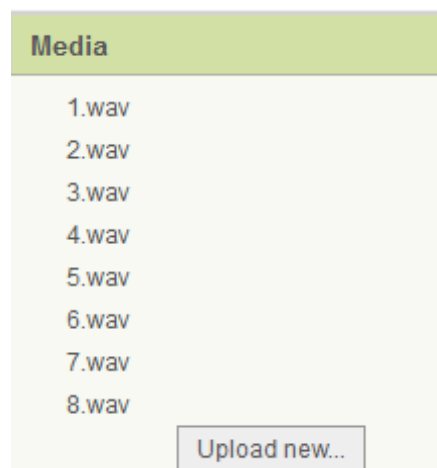
- a) Cambia el color de **BackgroundColor** a **Magenta**.
 - b) Modifica **Text** a **DO**.
 - c) Pon la propiedad **Width** a **Fill parent** para que el botón ocupe todo el ancho de la pantalla.
 - d) **Height** ponlo a **40**.
2. Repite los pasos anteriores para crear un segundo botón llamado **Boton2**. Colócalo debajo de **Boton1**. Usa los mismos valores en las propiedades **Width** y **Height** pero asigna el color **rojo** a **BackgroundColor** y escribe **RE** en su campo **Text**.

Más tarde repetiremos estos pasos para añadir 6 botones más. El contenido de **Component Designer** debería parecerse a la siguiente figura:



Agregar el componente **Sound**

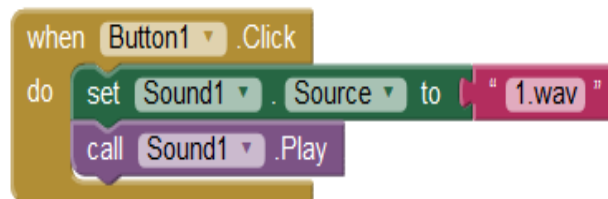
Añade desde la sección **media** todos los archivos de sonido suministrados para este capítulo. Después arrastra un componente **Sound** al visor y deja el nombre predeterminado: **Sound1**. Cambia su propiedad **MinimumInterval** (500 milisegundos) por **0**.



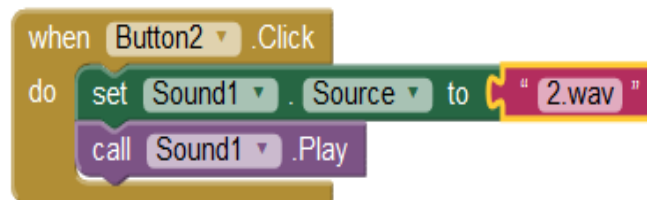
Tenemos que programar un comportamiento que reproduzca un sonido cuando el usuario toque un botón de la pantalla. Si se pulsa el **Boton1**, reproduciremos el sonido “**1.wav**”. Si se pulsa el **Boton2**, se reproducirá el sonido “**2.wav**”. Y así sucesivamente. Para ello, desde **Blocks**, haremos lo siguiente:

1. Abre el cajón **Boton1** y arrastra el bloque **Button1.Click** hasta el área de trabajo.
2. Abre el cajón **Sound1** y arrastra el bloque **Sound1.Source** hasta el área de trabajo, colocándolo dentro del bloque **Button1.Click**.

3. Arrastra un elemento de texto hasta el área de trabajo. Asigna **1.wav** al valor **text** y colócalo en el bloque **Sound1.Source**.
4. Añade un bloque **Sound1.Play**.



Haz lo mismo con el **Button2**, debería quedarte algo parecido a la figura siguiente:

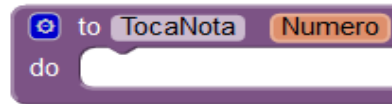


Da la sensación que estamos repitiendo código, ¿verdad?. Deberíamos generar un procedimiento, parecido al que vimos en los capítulos 3 y 5. Ahora, vamos a generar uno que tome los números como argumentos, asigne el archivo de sonido adecuado de cada fichero y lo reproduzca. Podemos utilizar los bloques de **unión** del cajón **Text** (una alternativa a **make text**) para **combinar el número** (por ejemplo, 1) y el **texto .wav**. Estos son los pasos que debemos dar para generar el procedimiento que necesitamos:

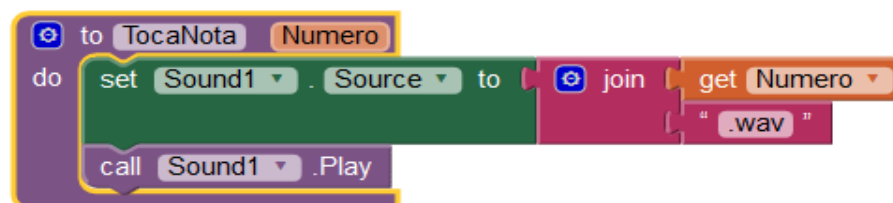
1. Abre el cajón **Procedures** y arrastra el bloque **to procedure** hasta el área de trabajo.
2. Haz clic sobre la rueda dentada azul y arrastra un bloque **input** dentro del bloque **inputs**. Este bloque representa los parámetros del procedimiento.



3. Haz clic sobre el parámetro **x** y cambia su nombre por **Numero**.
4. Pulsa sobre **procedure** para modificar su nombre por **TocaNota**.



5. Arrastra el bloque **Sound1.Source** hasta la ranura **do** de **TocaNota**. Coloca debajo un bloque **Sound1.play**.
6. Abre el cajón **Text** y arrastra el bloque **join** hasta la ranura de **Sound1.Source**.
7. Situate sobre el parámetro **numero** y desplaza una bloque **get** hasta la ranura superior del bloque **join**.
8. Abre el cajón **Text** y arrastra un bloque **text** hasta la ranura inferior del bloque **join**.
9. Cambia el valor **text** por **.wav**. Habrás obtenido:

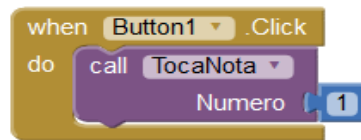


Ahora cambiaremos los eventos **Button1.clic** y **Buttton2.clic** que hicimos previamente.

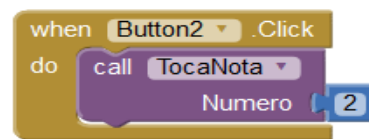
Borra su contenido y rellenalos de la siguiente forma:

10. Desde **Procedures** arrastra el bloque **call TocaNota** hasta el cuerpo vacío de **Button1.Click**.
11. Arastra un bloque numerico desde **Math** y colócalo en la ranura **Numero** de **callTocaNota**. Cambia el valor a **1**.

De esta manera, cuando se presione el **Boton1**, se llamará al procedimiento **TocaNota**. Su argumento **numero** tendrá el valor **1**. Así pues, debería asignar el valor **1.wav** a la variable **Sound1.Source** y reproducir el sonido **1.wav**.



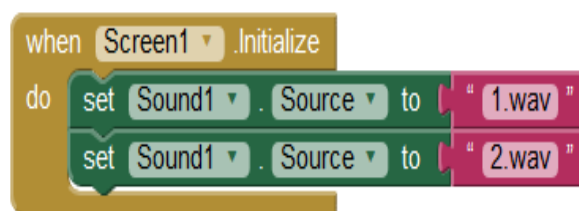
Ahora, crea un bloque **Button2.Click** de forma similar:



Como habrás observado la forma de llamar a los ficheros de audio: 1.wav, 2.wav, etc. tiene su sentido para crear un procedimiento genérico.

Decirle a Android que incorpore los sonidos

Si has probado la llamada anterior a **TocaNota**, habrás encontrado con que la aplicación no reproduce el sonido esperado o que cuando suena, aparece un retardo inesperado. La razón es que Android necesita abrir los sonidos durante la ejecución. Así pues, hay que abrir los archivos explícitamente durante el arranque del programa:



Guarda primero el proyecto con el nombre **Xylophone**, se generará el fichero **“Xylophone.apk”** que es el que debes copiar a tu teléfono. Verás que el sonido se reproduce con algo de retardo. Si no llegas a escuchar nada, comprueba el volumen de tu teléfono.

PRUEBA!!!!

Agregar las notas restantes

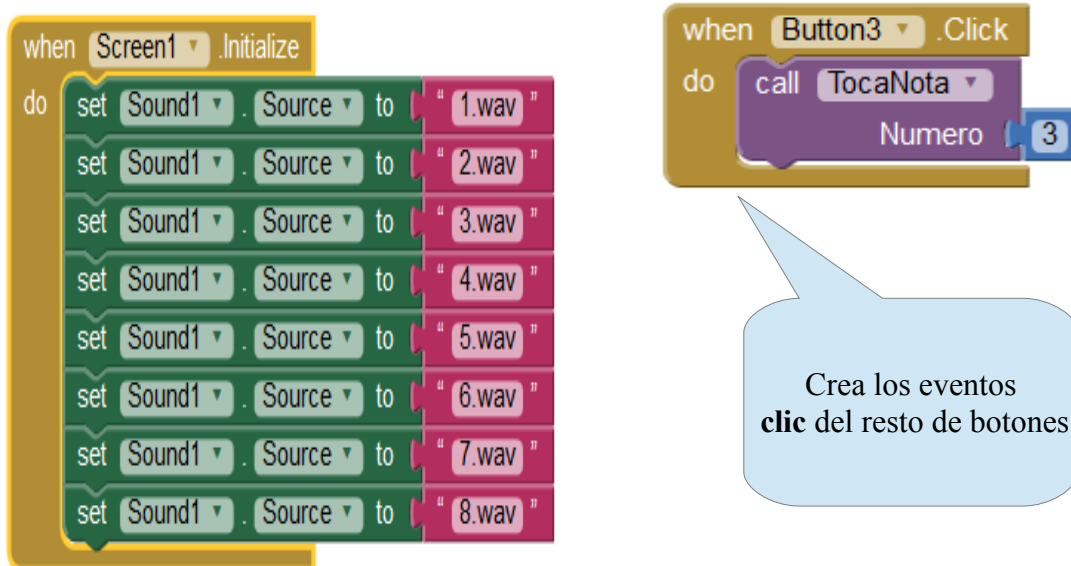
Añadimos las 6 notas restantes en sus correspondientes botones, como hemos hecho con los dos primeros. Luego modificamos el valor de sus propiedades **Text** y **BackgroundColor**:

- **Button3** (“MI”, Pink)
- **Button4** (“FA”, Orange)
- **Button5** (“SOL”, Yellow)
- **Button6** (“LA”, Green)
- **Button7** (“SI”, Cyan)
- **Button8** (“DO”, Blue)

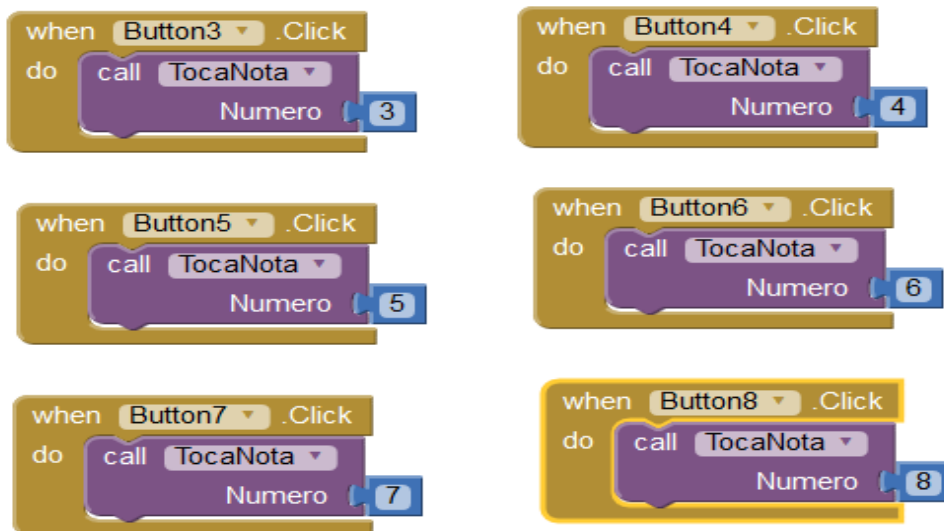
Si lo deseas, puedes cambiar el **TextColor** de **Button8** a **White** para que se lea mejor su contenido.



De vuelta a **Blocks** , crearemos bloques **Click** para cada uno de los nuevos botones y estableceremos las llamadas apropiadas a **TocaNota**. Del mismo modo, agregaremos cada archivo de sonido a **Screen.Initialize** como se ve en la siguiente figura:



Obteniendo:



Guarda primero el proyecto con el nombre **Xylophone2**, se generará el fichero “**Xylophone2.apk**” que es el que debes copiar a tu teléfono. Prueba todas las notas.

PRUEBA!!!!

Grabar y reproducir las notas

Es divertido utilizar nuestro xilófono virtual para tocar música pero sería mejor si nuestra aplicación fuese capaz de grabar las melodías que tocamos y reproducirlas posteriormente. Para que pueda hacer esto, debe recordar las notas que hemos tocado. Además de acordarse de las teclas que pulsamos, deberá reconocer el tiempo que dejamos entre dos notas. Si no actuamos así, no seremos capaces de distinguir cuándo se tocan dos notas muy rápidamente o se deja un tiempo de 10 segundos entre ellas.

Nuestra aplicación trabaja con dos listas. Cada una de ellas tendrá una entrada por cada una de las notas que se toquen:

- **Notas:** contendrá los nombres de los archivos de sonido que se han de reproducir, respetando el orden en que los tocó el usuario.
- **Tiempos:** Registrará el momento en el que se ha de reproducir una nota.

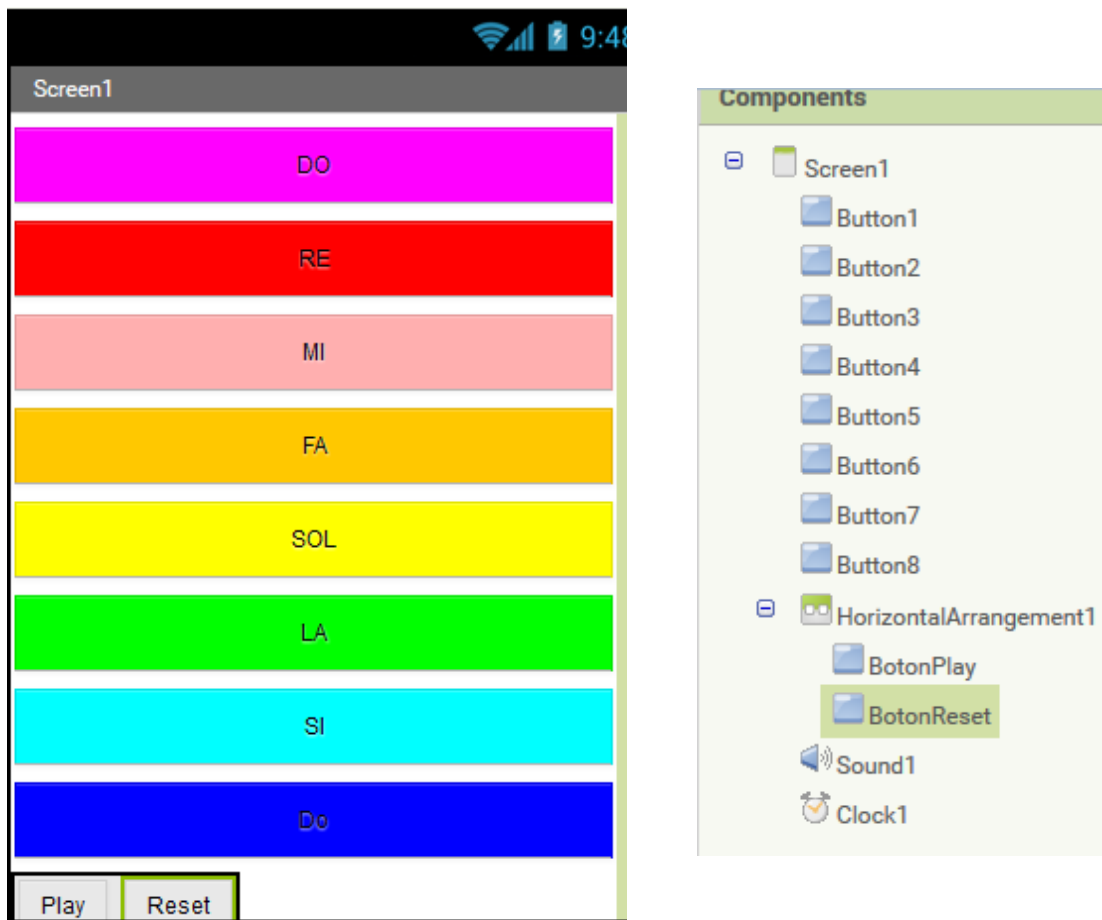
Podemos controlar el tiempo por medio del componente **Clock**, que también utilizaremos para determinar el tiempo de reproducción de las notas.

Desde **Component Designer** añadiremos un componente **Clock** y dos botones **Play** (Reproducir) y **Reset** (Borrar), que colocaremos dentro de un elemento **HorizontalArrangement**:

1. Arrastra un componente **Clock**. Desactive su propiedad **TimerEnabled** porque no queremos que su contador llegue a cero antes de que tengamos que utilizarlo durante la reproducción.
2. Accede al cajón **Layout** y arrastra un componente **HorizontalArrangement** hasta colocarlo debajo del último botón. Modifica su propiedad **Width** a **Fill parent**.
3. Desde el cajón **User Interface**, arrastra un elemento **Button**. Cambia su nombre a **BotonPlay** y su propiedad **Text** a **Play**.

4. Arrastra otro **Button** y colócalo a la derecha de **Play**. Cambia su nombre a **BotonReset** y asigna **Reset** a su propiedad **Text**.

El aspecto debería ser similar al siguiente:



Grabar y reproducir las notas

Deberemos tener una lista con las notas y los tiempos y agregar nuevos elementos a la misma siempre que el usuario presione un botón.

1. Crea una variable llamada **Notas**.
2. Abre el cajón **List** y arrastra un bloque **make a list**, colocándolo en la ranura de **def Notas**.

Estos pasos definirán una nueva variable **notas** que, inicialmente, estará vacía. Repite los pasos para crear una segunda variable llamada **tiempos**:

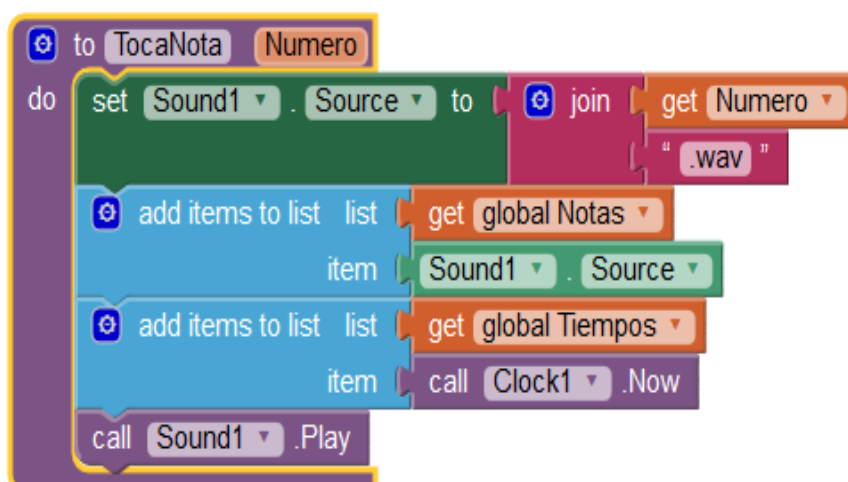


Siempre que se toca una nota, tenemos que guardar el nombre del archivo de sonido que se ha utilizado (dentro de la lista **notas**) y el instante en el que se tocó (en la lista **tiempos**).

Para tener en cuenta el instante en el que se toca la nota emplearemos el bloque **Click1.Now**, que devolverá el momento exacto (por ejemplo, March 12, 2011, 8:33:14 AM) con una precisión de milisegundos. Estos valores, que se obtienen a través de los bloques **Sound1.Source** y **Click1.Now**, se deberán añadir a las listas *notas* y *tiempos* respectivamente, como se ve en la siguiente figura. Por ejemplo, si en el xilófono tocamos las teclas **DO,DO,DO,RE,MI**, las listas tendrán cinco entradas:

- **Notas:** 1.wav, 1.wav, 1.wav, 2.wav, 3.wav.
- **Tiempos:** (omitimos las fechas): 12:00:01, 12:00:02, 12:00:03, 12:00:03.5, 12:00:04.

Modifica **TocaNota** tal que:



Cuando el usuario emplee el botón **Reset**, las dos listas deberán volver a su estado original, es decir, quedarán vacías. Como no se verá ningún cambio, estaría bien añadir un pequeño bloque **Sound1.Vibrate** para saber que realmente se ha hecho algo:



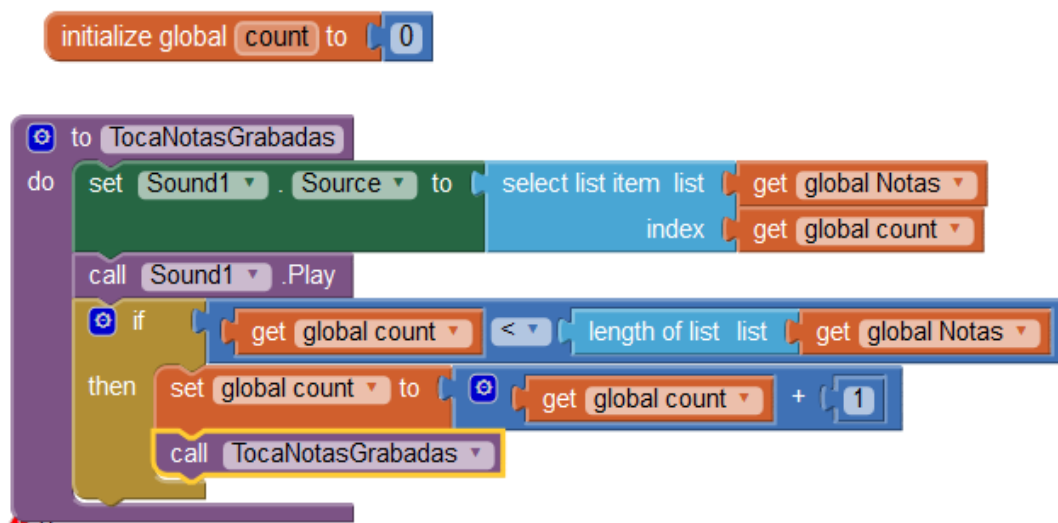
Para reproducir las notas, empezaremos centrándonos en cómo lograr que la herramienta reproduzca los sonidos, sin preocuparnos de momento por el tiempo. Utilizaremos una **variable count** que sabrá en qué nota nos encontramos (inicialmente tiene un valor 0), un procedimiento **TocaNotasGrabadas** (que se encargará de reproducir esa nota y de pasar a la siguiente) y le aplicaremos **recurrencia**. Cuando se presione el botón **Play**, el código se ejecutará, asignará el valor **1** al **contador** e invocará a **TocaNotasGrabadas**, a no ser que no se haya grabado ninguna nota.

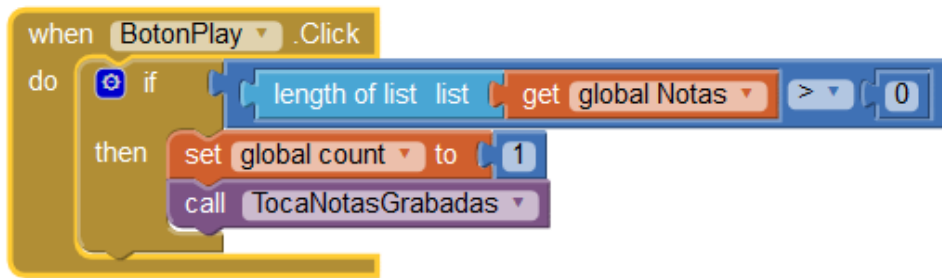
Para hacernos una idea de cómo funciona vamos a ir paso a paso, revisando todo lo que ocurrirá cuando el usuario toque **3 notas (1.wav, 3.wav, y 6.wav)** y luego presione el botón **Play**. Lo primero que pasa es que se pone en marcha **Play.Click**. Como la longitud de la lista es 3, que es mayor que 0, el contador tomará el valor 1 e invocará a **TocaNotasGrabadas**:

1. La primera vez que se llama a **TocaNotasGrabadas**, el valor de **count** será **1**:
 - a) Se configura **Sound1.Source** con el primer elemento de la lista **notas**, que es **1.wav**.
 - b) Se llama a **Sound1.Play** para que reproduzca esta nota.

- c) **count** (1) es menor que la longitud de la lista **notas** (3), entonces se incrementa el valor de **count** a 2 y vuelve a llamar a **TocaNotasGrabadas**.
2. La segunda vez que se llama a **TocaNotasGrabadas**, el valor de **count** será **2**:
- a) Se configura **Sound1.Source** con el segundo elemento de la lista **notas**, que es **3.wav**.
- b) Se llama a **Sound1.Play** para que reproduzca esta nota.
- c) **count** (2) es menor que la longitud de la lista **notas** (3), entonces se incrementa el valor de **count** a 3 y vuelve a llamar a **TocaNotasGrabadas**.
3. La tercera vez que se llama a **TocaNotasGrabadas**, el valor de **count** será **3**:
- a) Se configura **Sound1.Source** con el tercer elemento de la lista **notas**, que es **6.wav**.
- b) Se llama a **Sound1.Play** para que reproduzca esta nota.
- c) **count** (3) ya no es menor que la longitud de la lista **notas** (3), entonces no pasará nada más y se detendrá la reproducción.

Los bloques necesarios para este comportamiento se observan en la figura siguiente:



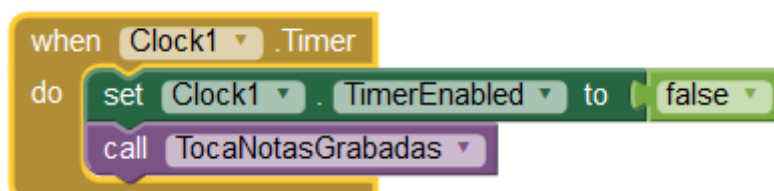


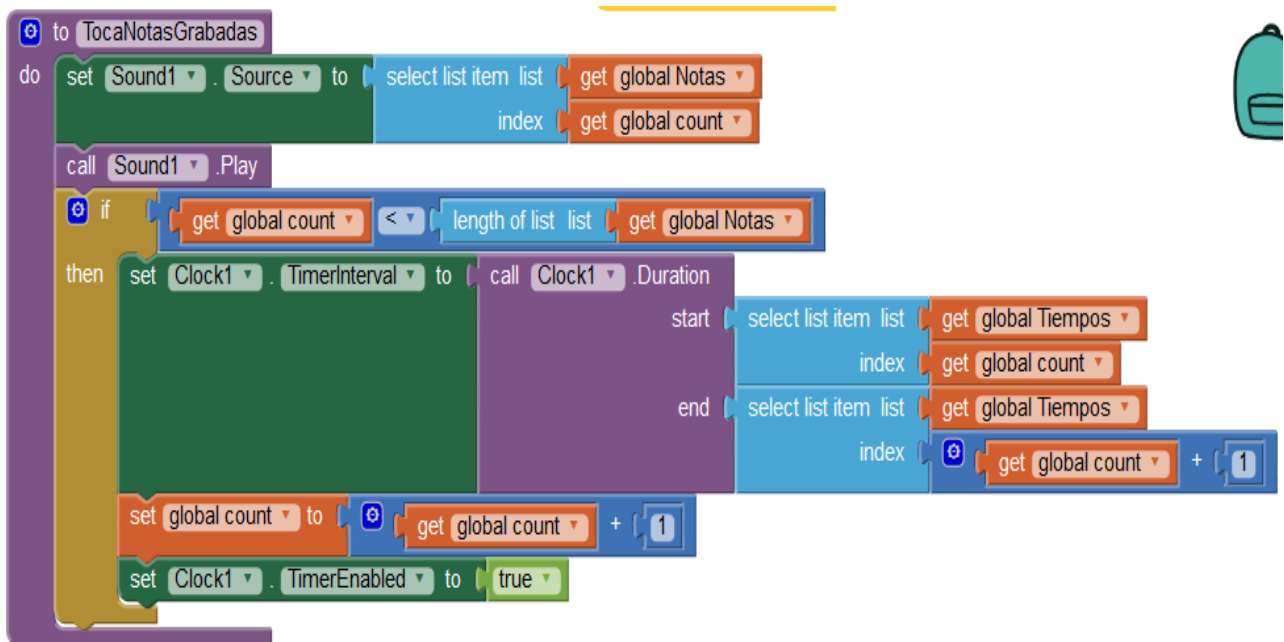
Guarda primero el proyecto con el nombre **Xylophone3**, se generará el fichero “**Xylophone3.apk**” que es el que debes copiar a tu teléfono. Prueba todas las notas.

PRUEBA!!!!

Hay un pequeño problema: prácticamente no pasa tiempo entre una llamada a **Sound1.Play** y la siguiente, de tal forma que las notas se pisan unas a las otras. A ninguna de ellas, a excepción de la última, se le permite completar su ciclo sonoro antes de que se cambie la fuente de Sound1 y se vuelva a llamar a **Sound1.Play**. Para corregir esta anomalía aplicaremos un pequeño retardo entre las invocaciones a **TocaNotasGrabadas**.

Para aplicar el retardo deberemos configurar el temporizador del reloj para que deje pasar una cantidad de tiempo entre la reproducción de una nota y la siguiente. Por ejemplo, si la que viene a continuación se reproduce 3000 ms (tres segundos) después de la actual, deberemos asignar el valor **3000** a la **propiedad Clock1.TimerInterval**. Seguidamente, se llamará de nuevo a **TocaNotasGrabadas**. Efectúa los cambios que muestra la figura de abajo en el cuerpo del bloque **if** en **TocaNotasGrabadas** y crea y rellena un controlador de eventos **Clock1.Timer** que se encargará de determinar qué ocurrirá cuando se agote el tiempo de espera.





Supongamos que tenemos el siguiente contenido en las dos listas:

- **Notas:** 1.wav, 3.wav, 6.wav
- **Tiempos:** 12:00:00, 12:00:01, 12:00:04

Play.Click asigna el valor **1** a **count** y llama a **TocaNotasGrabadas**.

1. La primera vez que se llama a **TocaNotasGrabadas**, el valor de **count** será **1**:
 - a) A **Sound1.Source** se le asigna el primer elemento de la lista **notas**, que es **1.wav**.
 - b) Se llama a **Sound1.Play** para que reproduzca esta nota.
 - c) Como **count** (1) es menor que la longitud de la lista **notas** (3), entonces:

Se asigna a **Clock1.TimerInterval** el intervalo de tiempo que transcurre entre el primer elemento (12:00:00) y el segundo (12:00:01) de la lista **tiempos**: **1 segundo**.

Se incrementa el valor de **count** a **2**.

Se activa **Clock1.Timer** e inicia la cuenta atrás.

Durante un segundo no ocurre nada más. Transcurrido este tiempo, se desactiva el temporizador y se lanza una llamada a **TocaNotasGrabadas**.

2. La segunda vez que se llama a **TocaNotasGrabadas**, el valor de **count** será **2**:
 - a) A **Sound1.Source** se le asigna el segundo elemento de la lista **notas**, que es **3.wav**.
 - b) Se llama a **Sound1.Play** para que reproduzca esta nota.
 - c) Como **count** (2) es menor que la longitud de la lista **notas** (3), entonces:

Se asigna a **Clock1.TimerInterval** el intervalo de tiempo que transcurre entre el segundo elemento (12:00:01) y el tercero (12:00:04) de la lista **tiempos**: **3 segundo**.

Se incrementa el valor de **count** a **3**.

Se activa **Clock1.Timer** e inicia la cuenta atrás.

Durante tres segundos no ocurre nada más. Transcurrido este tiempo, se desactiva el temporizador y se lanza una llamada a **TocaNotasGrabadas**.

3. La tercera vez que se llama a **TocaNotasGrabadas**, el valor de **count** será **3**:
 - a) A **Sound1.Source** se le asigna el tercer elemento de la lista **notas**, que es **6.wav**.
 - b) Se llama a **Sound1.Play** para que reproduzca esta nota.
 - c) Como **count** (3) ya no es menor que la longitud de la lista **notas** (3), no pasa nada más y se completa la reproducción.

Guarda primero el proyecto con el nombre **Xylophone4**, se generará el fichero “**Xylophone4.apk**” que es el que debes copiar a tu teléfono.

PRUEBA!!!!

Variaciones

Aquí tienes algunos escenarios alternativos para que explores:

- Ahora mismo no hay nada que impida al usuario tocar el botón **Reset** durante la reproducción, lo que provocaría un fallo en la aplicación. Modifica **Play.Click** para que desactive **Reset** durante su desarrollo. Para volver a activarlo cuando se complete la canción deberás cambiar el bloque **if** de **Play.Click**. Conviértelo en un **ifelse** y coloca la reactivación del botón **Reset** en la parte **else**.
- Igualmente, el usuario puede tocar el botón **Play** durante la reproducción, lo que provocaría un fallo en la aplicación. Modifica **Play.Click** para que desactive **Play** durante su desarrollo y cambia su etiqueta para que muestre **Playing** (Reproduciendo). Puedes utilizar un bloque **ifelse** para volver a activarlo y devolverlo a su etiqueta original.
- Agrega un botón con el nombre de la canción, por ejemplo, **Für Elise**. Si el usuario hace clic en él, la aplicación abrirá sus notas y sus tiempos en las listas correspondientes (**notas** y **tiempos**), asignará 1 a **count** y llamará a **TocaNotasGrabadas**. Para configurar los tiempos adecuados recomiendo que recurras al bloque **Clock1.MakeInstantFromMillis**.
- Si el usuario toca una nota, se marcha y no hace nada más durante varias horas, y después toca la siguiente nota, la aplicación entenderá que estas dos forman parte del mismo tema (que no será lo que el usuario tendría en mente). Mejora el programa haciendo esto: **(1) detén la grabación después de un intervalo razonable de tiempo**, por ejemplo, un minuto, o **(2) establece un límite en la cantidad de tiempo** que puede emplear **Clock1.TimerInterval** mediante el bloque **max** que encontrarás en el cajón **Math**.

- Modifica la apariencia del botón para que enseñe el nombre de la nota que está reproduciendo. Por ejemplo, puedes alterar el valor de las propiedades **Text**, **BackgroundColor** o **ForegroundColor**.