

Capítulo 7

Aplicación práctica II



Contenido

1. Introducción
2. Descripción de la aplicación
3. Diseño de la interfaz gráfica. Elección y configuración de componentes
4. Desarrollo de la lógica. Definir el comportamiento
5. Resumen

1. Introducción

Continuando con el desarrollo de aplicaciones *Android* mediante *MIT App Inventor 2*, y del mismo modo que ocurriera en la unidad anterior, en la presente unidad se expondrán las pautas a seguir para diseñar tanto la interfaz gráfica como todo lo necesario en cuanto a la lógica y comportamiento de una aplicación basada en baterías de preguntas de una única selección. Se trabajará, entre otros, con bloques de listas y de base de datos para el almacenamiento y recuperación de información. Al igual que ocurriera con la aplicación desarrollada en la unidad anterior, esta será una aplicación totalmente funcional que se podrá descargar y probar en el dispositivo *Android* físico.

2. Descripción de la aplicación

Se va a desarrollar una aplicación que se denominará *SportQuiz*, en la que se incluirán tres baterías de cinco preguntas cada una, relacionadas con un deporte concreto.

La aplicación contará con dos pantallas, una inicial en la que se podrá elegir un deporte en el que se basarán las preguntas seleccionando una imagen, y una segunda en la que irán apareciendo las preguntas junto con sus posibles respuestas, entre las que se deberá elegir una. Para ello, se debe marcar una caja de selección y, seguidamente, pulsar el botón Aceptar. Al hacerlo, se producirá un sonido indicando si se ha acertado o no. Al finalizar la batería de preguntas, se felicitará al usuario por haber acertado todas las preguntas o, de otro modo, se le indicarán las preguntas en las que ha errado.

Se puede observar que la interfaz gráfica de la aplicación es sencilla. En ella se incorporan una serie de elementos de imagen acordes con el deporte elegido y, además, como característica se añade en la segunda de las pantallas un componente *ImageSprite* que realizará un movimiento por un componente *Canvas*.

A modo de introducción en relación a la lógica de la aplicación, la primera de las pantallas servirá para almacenar los datos correspondientes, tanto a los enunciados de las preguntas como a posibles respuestas y soluciones. Este

mecanismo de almacenamiento de información se realizará a través del bloque TinyDB y, concretamente, con su procedimiento StoreValue. Este procedimiento recibe dos parámetros, uno con el nombre o etiqueta, *tag* en inglés, mediante el que se identificará el valor almacenado, y otro con dicho valor. Del mismo modo, permitirá en la segunda de las pantallas la recuperación de los datos almacenados a través del procedimiento GetValue, cuyo funcionamiento es en parte similar al anterior. Primero, se especificará el nombre de la etiqueta a recuperar y, segundo, el valor por defecto que aparecerá en caso de no existir el valor del primer parámetro especificado.

Otro elemento con el que se trabajará serán las listas. Las preguntas serán almacenadas a modo de listas de cuatro elementos. El primero de ellos corresponderá al enunciado, y los tres siguientes, a las tres posibles respuestas.

Para realizar el cambio de la primera pantalla a la segunda se deberá elegir uno de los tres posibles botones, a los cuales se asignará una imagen relacionada con el deporte en cuestión. Al seleccionarlo, se abrirá la segunda pantalla y se pasará por parámetro información referente al deporte elegido a través de una variable.

Desde aquí se planteará una posible solución en cuanto a interfaz gráfica y lógica de aplicación se refiere. Una vez se finalice la práctica y se dominen los diferentes elementos que en ella se incluyen, se recomienda alterarlos y modificarlos con el fin de practicar planteando otras posibles soluciones.

Para empezar el desarrollo de la aplicación, se deberá crear el nuevo proyecto para tenerlo todo listo al comienzo del siguiente paso. A continuación, se resume el proceso.

Se debe acceder desde el navegador a la dirección de *MIT App Inventor 2* que, como ya se sabe, es <http://appinventor.mit.edu/>, elegir la opción **Create** e introducir los datos de usuario de la cuenta de Google. Una vez dentro de la aplicación, hacer clic sobre la opción de menú **Projects** y elegir **Start new projects...** Como nombre para la aplicación, se puede elegir *unidad7*.

El proyecto contiene inicialmente una pantalla y se añadirá una segunda. Para ello, hacer clic sobre el botón **Add screen...** y dejar el nombre **Screen2**

que tiene por defecto. Así se diferenciará a **Screen1** como la pantalla inicial de selección del deporte y a **Screen2** como la pantalla donde se realizarán las preguntas.

3. Diseño de la interfaz gráfica. Elección y configuración de componentes

Una vez creado el proyecto, se comenzará a desarrollar el aspecto que poseerá la aplicación. Se especificarán cada uno de los componentes que constituyen las pantallas adaptándose a las exigencias de los requisitos. Seguidamente, se determinarán qué valores poseerán inicialmente cada uno de los componentes elegidos y, por último, se establecerá la lógica de la aplicación para que el comportamiento de la misma sea el esperado. Esta última labor no se realizará en este sino en el próximo apartado.



Nota

Para el diseño de la interfaz gráfica de la aplicación se hará uso de una serie de recursos de imagen que se podrán descargar de la URL <<http://www.recursosadicionales.com/mit/ud7.rar>>. Con esto se conseguirá adaptar el desarrollo de la forma más fiel a la explicación.

Se deberán tener claros los requisitos de la aplicación leyendo al detalle la descripción de la misma con el fin de desglosarla y extraer la máxima información para seleccionar los componentes adecuados. Es lógico pensar que, cuanta más información se posea sobre los requisitos de la aplicación, más fácil será determinar qué componentes son necesarios para desarrollarla con garantías. En la elección de componentes se observará que se seleccionará algún componente no visible, que no formará parte de la interfaz gráfica pero que será igual de necesario como los que sí lo son, para el correcto funcionamiento de la aplicación.

3.1. Pantalla de selección de deporte

Al abrir la aplicación aparecerá a modo de pantalla de inicio la pantalla de selección de deporte. Esta pantalla que se podrá identificar como **Screen1** en el panel **Components** mostrará en la parte superior el título de la aplicación y, a su derecha, un dibujo decorativo. Además, debajo se colocarán tres botones dispuestos verticalmente, uno para cada deporte, como son fútbol, baloncesto y tenis.

En primer lugar, se ha de seleccionar el componente **Screen1** y definir las propiedades **AlignHorizontal** y **AlignVertical** con el valor **Center** para que los componentes que se añadan a continuación en ella se alineen en el centro del eje horizontal y vertical. Para habilitar la segunda de las opciones, se deberá desmarcar primero la opción **Scrollable**, que a su vez hará que no aparezca una barra de desplazamiento vertical en la pantalla. Se otorgará a la aplicación un icono para identificarla entre las apps que se tengan instaladas en el dispositivo *Android*, eligiendo el archivo “icon.png” desde la propiedad **Icon**. Además, se debe establecer la propiedad **ScreenOrientation** a **Portrait** para que mantenga la orientación vertical de pantalla. Por último, añadir como **Title** el valor **Unidad 7**.

Para definir estos componentes se añadirá primero un componente **VerticalArrangement** que los englobará a todos. Hay que comenzar definiendo para las dimensiones **Fill parent** en la propiedad **Width** y **Height**, para que se ajuste al ancho y alto de la pantalla del dispositivo, y a **Center** tanto la propiedad **AlignHorizontal** como **AlignVertical**.

Dentro de este, se deben incluir cuatro componentes **HorizontalArrangement** cuyas propiedades serán iguales. Entre ellas, hay que establecer tanto **AlignHorizontal** como **AlignVertical** con valor **Center**, además de **Width** y **Height** con valor **Fill parent**. Con esto, cada uno de ellos ocupará la misma cantidad de pantalla. Dentro del primer **HorizontalArrangement** incluir un componente **Label**, para el título de la aplicación, y otro **Image**, en el que se colocará una imagen descriptiva. Para la etiqueta, marcar la propiedad **FontBold**, establecer **26** como valor de **FontSize** y **SportQuiz** de **Text**. Además, elegir **Center** para la propiedad **TextAlign**, **Fill parent** para **Width** y **Automatic** para **Height**. Por otro lado, en cuanto al componente imagen se elegirá la imagen “inicio.png”

para la propiedad **Picture**. Las demás se dejarán por defecto. Seguidamente, y para el segundo, tercer y cuarto componente **HorizontalArrangement**, añadir un botón para cada uno de ellos. Su función será la de determinar el deporte de las preguntas a mostrar en función del botón elegido. Para diferenciar cada uno de los botones, se pueden renombrar como btnFutbol, btnBaloncesto y btnTenis, respectivamente. Además hay que elegir, para el atributo **Image** de cada uno de los botones, la imagen “futbol.png”, “baloncesto.png” y “tenis.png” para el botón correspondiente. A continuación, eliminar el contenido de la propiedad **Text** y especificar, tanto en la propiedad **Width** como en **Height** de cada botón, un tamaño de 60 px para ambos.

Aunque no sean componentes visibles en la interfaz gráfica, se van a añadir unos componentes que posteriormente se utilizarán desde el panel **Editor de bloques**. Habrá que insertar los componentes **TinyDB** y **Player** desde la paleta de componentes hasta el visor. Obsérvese que se colocarán debajo en la zona de componentes no visibles. En principio, y por el momento, solo se modificará para estos componentes desde el diseñador la propiedad **Source** del componente **Player**, eligiendo el archivo de sonido “inicio.mp3”.

En la siguiente tabla se podrán observar todos los componentes junto con sus nombres. Esto es importante y facilitará posteriormente su identificación a través de los bloques que se utilizarán en el desarrollo de la lógica de la aplicación. Compruébese que todos los componentes concuerdan con los nombres que aparecen a continuación y renómbrense en el proyecto si no fuera así.

Componentes de la pantalla de Selección de deporte	
Componente	Uso
VerticalArrangement1	Distribución vertical de los componentes.
HorizontalArrangement1	Distribución horizontal de los componentes.
lblTitulo	Etiqueta que muestra el título.
imgTitulo	Imagen descriptiva del título.
HorizontalArrangement2	Distribución horizontal de los componentes.

Continúa en página siguiente >>

<< Viene de página anterior

Componentes de la pantalla de Selección de deporte	
btnFutbol	Botón para elegir preguntas de fútbol.
HorizontalArrangement3	Distribución horizontal de los componentes.
btnBaloncesto	Botón para elegir preguntas de baloncesto.
HorizontalArrangement4	Distribución horizontal de los componentes.
btnTenis	Botón para elegir preguntas de tenis.
TinyDB1	Almacena la información de las preguntas.
Player1	Reproduce un sonido al iniciar la aplicación.

3.2. Pantalla de preguntas

La segunda pantalla tiene como finalidad mostrar las diferentes preguntas de las que se compone la batería de la opción elegida. Se comprobará a continuación que su interfaz gráfica será sencilla de definir. En ella se distingue inicialmente un componente **Canvas** por el que se moverá un componente **ImageSprite**, que representa a la pelota con la que se juega en el deporte elegido. Seguidamente, se diferenciará una zona en la que se muestra tanto el enunciado como las posibles respuestas representadas a modo de componentes **CheckBox**. Además, se mostrará un botón que confirmará la elección del usuario y continuará con la siguiente pregunta si la hubiera, junto con una etiqueta en la que se mostrará el resultado al finalizar la batería de preguntas. Por último, una imagen que ilustra el campo en el que se practica el deporte elegido.

Comenzar, al igual que ocurriera con la pantalla anterior, seleccionando el componente **Screen2** en este caso y establecer las propiedades **AlignHorizontal** y **AlignVertical** con el valor **Center** para que los componentes que se añadan a continuación en ella se alineen en el centro del eje horizontal y vertical. Para habilitar la segunda de las opciones se deberá desmarcar primero la opción **Scrollable** que a su vez hará que no aparezca una barra de desplazamiento vertical en la pantalla. Además, se debe establecer la propiedad **ScreenOrientation** a **Portrait** para que se mantenga fija la orientación vertical de pantalla.



Recuerde

Para que las pantallas se adapten a la resolución del dispositivo, se debe desmarcar la opción `Scrollable` del componente `Screen`. Con esto no se mostrará la barra de desplazamiento vertical.

Seguidamente, se debe añadir un componente `VerticalArrangement`, en el que se incluirán todos los componentes visibles de la pantalla. Establecer, para las dimensiones, el valor `Fill parent` en la propiedad `Width` y `Height`, para que se ajuste al ancho y alto de la pantalla del dispositivo, y a `Center` tanto la propiedad `AlignHorizontal` como `AlignVertical`.

En su interior, incluir primero un componente `Canvas` y, dentro de este, otro `ImageSprite`. Para el primero de ellos, establecer los siguientes valores a las propiedades: `None` para `BackgroundColor`, `Right` para `TextAlignment` y `Fill parent` para la propiedad `Width`. En cuanto al componente `ImageSprite` se debe establecer `0` como valor para `X` e `Y`, además de `50 pixels` tanto para `Width` como para `Height`.

En la siguiente zona se muestran tanto enunciado como preguntas. Dichos componentes irán dentro de otro del tipo `VerticalArrangement`, para el que se establecerá a `Fill parent` el valor de las propiedades `Width` y `Height`, y a `Center`, tanto la propiedad `AlignHorizontal` como `AlignVertical`, con lo que los componentes internos se alinearán en el centro de la pantalla en los ejes horizontal y vertical, ocupando el espacio disponible a lo ancho y alto de la pantalla, de manera proporcional en el último caso. Dentro de este componente de distribución vertical, se encuentra, en primer lugar, una etiqueta o componente `Label`, en el que se mostrará el enunciado de la pregunta. Entre sus propiedades a modificar, se debe eliminar el texto contenido en la propiedad `Text` y dejarlo vacío, establecer el valor `Center` en la propiedad `TextAlignment` y un ancho de `280 pixels` para la propiedad `Width`. Se puede probar a establecer el ancho más adecuado al dispositivo con el fin de que no haya desbordamiento debido a que el elegido sea similar al total de píxeles del ancho de la pantalla, con esto

se tiene que es conveniente definir algunos píxeles menos. Seguidamente hay que añadir tres componentes **CheckBox**, para los que se deberán establecer las mismas propiedades para cada uno de ellos. Entre ellas, eliminar el texto de la propiedad **Text** y dejarlo vacío y, del mismo modo que para la etiqueta del enunciado, establecer **280 pixels** para la propiedad **Width**.

Para terminar, se añadirán tres componentes más: el botón que confirmará la elección del usuario, una etiqueta para indicar el resultado final y la imagen que muestra el campo donde se desarrolla el deporte elegido. Para el botón, establecer el valor **Aceptar** para la propiedad **Text**. En cuanto a la etiqueta, simplemente se eliminará el texto que contiene la propiedad **Text** y se dejará vacía. Por último, para la imagen, se cambiará a **Fill parent** los valores tanto de **Width** como de **Height**, para que se ocupe todo el sitio que falte para rellenar la página.

Se podría pensar que, durante la creación de esta pantalla, se han olvidado definir las imágenes a mostrar tanto para la pelota como para el campo o pista deportiva en el que se practica cada deporte, y no es así. En este caso, se trata de una definición de imágenes que se establecerán en tiempo de ejecución, es decir, en función de la elección del usuario cuando la aplicación se esté ejecutando, y que será definida en el siguiente apartado desde el **Editor de bloques**. Aunque, por otro lado, no hay que olvidar incluirlas al proyecto en *MIT App Inventor 2*: estas son “campoF.png”, “campoB.png” y “campoT.png”.

Al igual que ocurriera con la primera pantalla, se van a insertar algunos componentes no visibles que no se muestran en la interfaz pero que posteriormente se utilizarán desde el panel **Editor de bloques**. Insértense los componentes **TinyDB**, **Player** y **Clock** desde la paleta de componentes hasta el visor. Obsérvese que se colocarán debajo en la zona de componentes no visibles. En principio, y por el momento, solo se modificará, para estos componentes, desde el diseñador la propiedad **TimeInterval** del componente **Clock** dotándole del valor **100**.

A continuación, se muestra una tabla con los nombres de todos los componentes junto con sus definiciones. Hay que asegurarse de que los nombres de todos los componentes se corresponden con los nombres que aparecen a continuación y renombrarlos en el proyecto si no fuera así.

Componentes de la pantalla de Selección de deporte	
Componente	Uso
VerticalArrangement1	Distribución vertical de los componentes.
Canvas1	Zona por la que se moverá la pelota.
isLogoDeporte	Representa una pelota que realizará un movimiento.
VerticalArrangement2	Distribución vertical de los componentes.
lblEnunciado	Muestra el enunciado de la pregunta.
cbOpcionA	Muestra la respuesta a.
cbOpcionB	Muestra la respuesta b.
cbOpcionC	Muestra la respuesta c.
btnAceptar	Botón para confirmar la elección.
lblResultado	Muestra el resultado final.
imgCampo	Imagen del campo o pista deportiva.
TinyDB1	Recupera la información de las preguntas.
Player1	Reproduce un sonido para notificar si se ha acertado.
Clock1	Reloj para determinar el movimiento de la pelota.

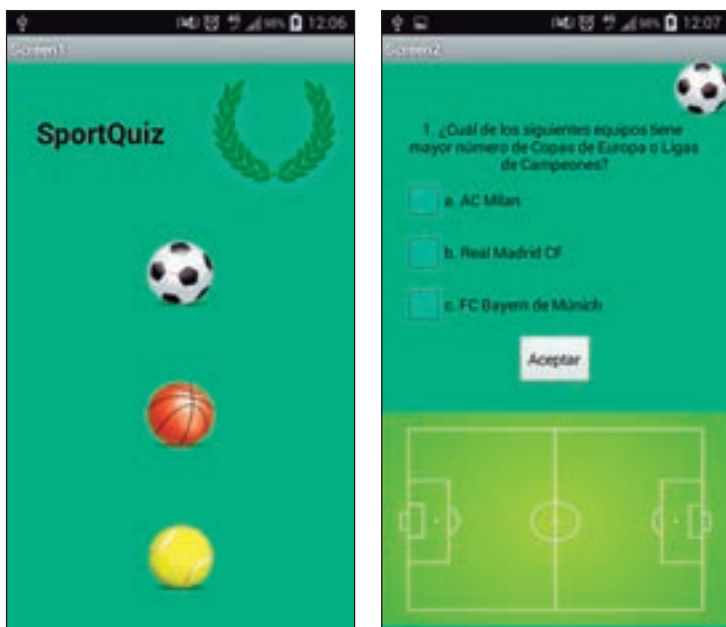


Actividades

1. Modifique la orientación de cada pantalla para observar el comportamiento de la interfaz gráfica en modo paisaje.

Del mismo modo que no se han definido las imágenes a mostrar en la pantalla de las preguntas, hay otra serie de características, como por ejemplo el color de fondo o el texto contenido en las preguntas, que serán también establecidas en tiempo de ejecución. Con esto se puede afirmar que en este momento no se dispone de la información suficiente para mostrar una previsualización del aspecto de las pantallas. En las siguientes imágenes se muestra un adelanto

de cómo van a quedar definidas ambas pantallas después de aplicar la configuración que en el apartado siguiente se desarrollará a través del **Editor de bloques**. Es posible que el dispositivo tenga una resolución o densidad de pantalla diferente y no se muestre tal cual se aprecia en las siguientes imágenes.



Interfaz gráfica



Recuerde

Al iniciar el desarrollo de una aplicación se han de establecer en primer lugar todos los componentes gráficos de la interfaz antes de emprender el desarrollo de la lógica de la aplicación. Así se evitará tener que hacer y rehacer gran cantidad del trabajo.

4. Desarrollo de la lógica. Definir el comportamiento

Tras diseñar la parte dedicada a la interfaz gráfica desde la ventana del **Diseñador**, se va a continuar desarrollando la lógica de la aplicación a través del **Editor de bloques**. Dichos bloques determinarán el comportamiento de la misma a través de la interacción del usuario con los componentes que se han añadido a la interfaz gráfica a lo largo del apartado anterior.

4.1. Pantalla de selección de deporte

La sencillez de esta primera pantalla se pudo observar durante el diseño de la interfaz gráfica. La interactividad que el usuario puede tener con ella se reduce a la elección de una de las tres opciones que son accesibles al pulsar uno de los botones. En este apartado se va a definir la lógica de ese comportamiento, se establecerán las estructuras de almacenamiento de los datos correspondientes a las preguntas, además de otros pequeños detalles como la reproducción de un sonido al iniciar la aplicación y la elección del color de fondo de la pantalla.

Lo primero sería añadir un bloque **Initialize** que actuará sobre el componente **Screen1** al iniciarse este. Dentro del mismo, se debe incluir el método **BackgroundColor** aplicado al componente **Screen1** y conectar a este bloque la función **make color**, que se encontrará desde la categoría **Colors**, permitiendo así definir el color mediante un código RGB. Establecer (76, 178, 125) como valores del código. De este modo se podrá establecer cualquier color que se necesite y no solo los predefinidos que se proponen desde el panel **Properties** del **Diseñador** o del panel **Blocks** desde el **Editor de bloques**.

Seguidamente, se incluirá el bloque necesario para que se reproduzca un sonido cuando la aplicación sea iniciada. Para ello, seleccionar el componente **Player1** desde el panel **Bloques** y añadir el bloque correspondiente al procedimiento **Start** a continuación. Para este componente ya se configuró la propiedad **Source** con el sonido en cuestión durante el apartado anterior.

A continuación, se va a mostrar cómo almacenar datos haciendo uso del bloque **TinyDB**, concretamente mediante su procedimiento **StoreValue**. A través

Crea tus aplicaciones Android con App Inventor 2

de él se podrán guardar los enunciados, posibles respuestas y soluciones, todo ello gracias al uso de listas.

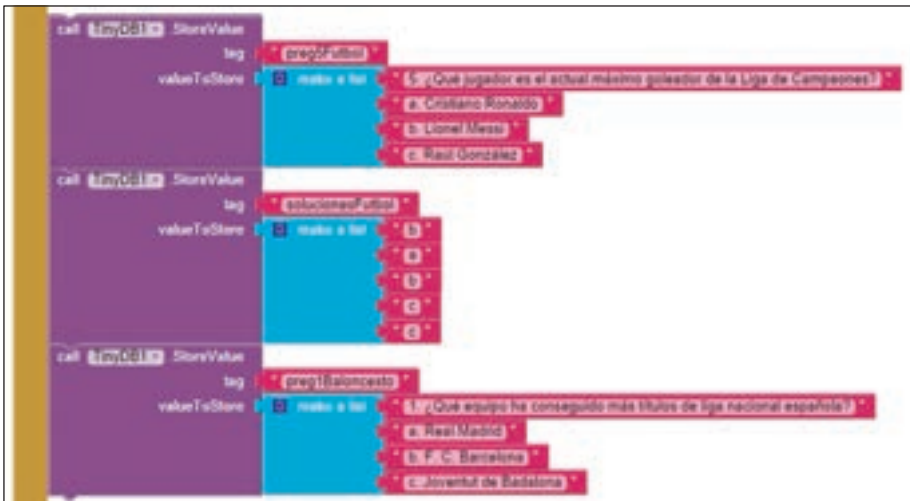
En la siguiente imagen se muestran los bloques incluidos hasta el momento. En cuanto a las preguntas, solo se muestran dos para no hacer redundante el contenido. Los enunciados y posibles respuestas que se sugieren se podrán encontrar en el documento “preguntas.txt”, que se incluye en la documentación a través del enlace de descarga indicado anteriormente. Si se desea elegir otras se puede hacer, siempre teniendo especial cuidado en establecer correctamente las soluciones.



Evento *Initialize* del componente *Screen1*

Como se habrá podido observar, cada valor almacenado consta de un **tag** y del valor a almacenar. El **tag** está comprendido por el prefijo **preg** más el número de la pregunta, que oscilará desde uno hasta cinco, y el nombre del deporte. Según esta nomenclatura, los **tag** de cada pregunta serán **preg1Futbol**, **preg2Futbol**, **preg3Futbol**, **preg4Futbol**, **preg5Futbol**, **preg1Baloncesto**, **preg2Baloncesto**, **preg3Baloncesto**, **preg4Baloncesto**, **preg5Baloncesto**, **preg1Tenis**, **preg2Tenis**, **preg3Tenis**, **preg4Tenis** y **preg5Tenis**. Estos **tag** servirán más adelante para recuperar la información contenida en cada valor almacenado que se corresponde con una lista, cuyos elementos son cuatro: el primero se corresponde al enunciado y los sucesivos, a las posibles respuestas.

Tras cada ronda de preguntas de un deporte concreto se incluirán sus soluciones, de modo que todos los bloques de procedimientos **StoreValue** irán encadenados unos tras otros hasta completar todas las preguntas y soluciones respectivas. En la siguiente imagen se podrá observar la concatenación entre la última de las preguntas de **Fútbol** con sus soluciones y con la primera de las preguntas del siguiente deporte, que es **Baloncesto**. Insértense todos los bloques necesarios para definir las quince preguntas con sus respectivos solucionarios. Se propone este orden para facilitar la organización, pero realmente no influye en la ejecución de la aplicación.



Procedimientos **StoreValue** del evento **Initialize** del componente **Screen1**

Por último, y desde fuera del bloque **Initialize**, se añadirán los bloques correspondientes a los eventos de “Click” que el usuario podrá realizar sobre los botones de cada deporte. Para añadir dichos eventos, hay que elegir desde el panel **Bloques** cada uno de los botones y seleccionar el bloque correspondiente al evento “Click”. La finalidad de cada bloque es detectar la pulsación de un botón para dirigir al usuario a la batería de preguntas que corresponda a su elección. Esta acción se realizará con el bloque de control **open another screen with start value**, al que habrá que pasarle primero el nombre de la pantalla a la que se desea dirigir y, segundo, un parámetro que se recibirá en la pantalla de

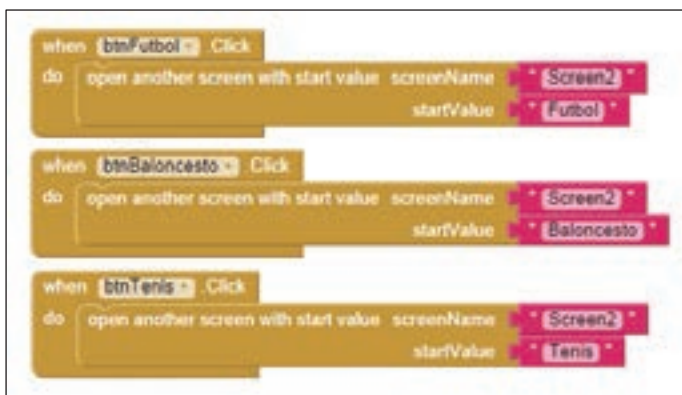
destino al iniciarse. Este parámetro determinará posteriormente la batería de ejercicios que se mostrarán.



Nota

El paso de parámetros solo funcionará en *MIT APP Inventor 2* en caso de exportar el fichero “apk” para su posterior instalación en un dispositivo *Android* real. De otro modo, al probarlo a través de *AI Companion*, el emulador o mediante el cable USB no tendrá efecto.

Los bloques de evento “Click” de los tres botones quedarán de la forma que muestra la siguiente imagen.



Eventos “Click” para elección de la batería de preguntas

4.2. Pantalla de preguntas

Finalizado el desarrollo de la lógica de la primera de las pantallas, se continúa con la segunda y última. Esta pantalla contiene una lógica y comportamiento más complejo que la anterior y por ello se va a separar y dividir su desarrollo lo más posible con el fin de facilitar su comprensión.

Inicialmente se van a declarar diferentes variables que se utilizarán a lo largo del desarrollo de la lógica de la pantalla, algunas de tipo **Math** o numéricas, que indicarán por ejemplo el total de preguntas, el número de pregunta o errores actuales; y otras de tipo **Text** o cadena de texto, que mostrarán por ejemplo un mensaje para el resultado o almacenarán el deporte elegido para las preguntas.

En la siguiente tabla se muestra una relación de las variables empleadas junto con su tipo y valor inicial, además de una imagen con el resultado.

Variables para la pantalla Screen2		
Variable	Tipo	Valor inicial
total_preguntas	Math	5
num_pregunta	Math	1
indice	Math	1
errores	Math	0
respuesta	Text	"" (cadena vacía)
resultado	Text	"No ha acertado las preguntas:"
deporte	Text	"" (cadena vacía)

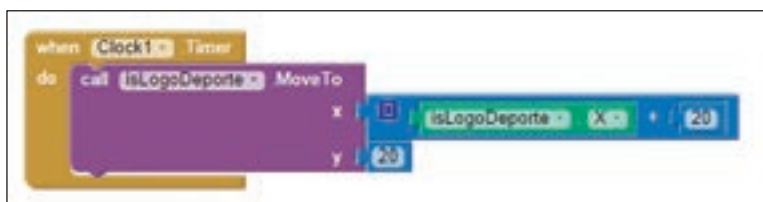


Variables para la pantalla Screen2

Tras definir las variables e inicializarlas, la primera acción que realiza la pantalla **Screen2** sin la intervención del usuario es el desplazamiento de una imagen cuyo aspecto corresponde a la pelota con la que se practica el deporte

elegido. Esta imagen, como se definió en el apartado anterior, es un componente **ImageSprite** que se mueve por la pantalla dentro de la zona que ocupa un componente **Canvas**. Para definir el movimiento, se utilizará el evento “Timer” del componente no visible **Clock1**, añadido durante el diseño de la interfaz gráfica. Añádase al editor de bloques e inclúyase dentro del mismo el procedimiento **MoveTo** del componente **isLogoDeporte**, que es el nombre que se le dio al **ImageSprite**. El procedimiento recibe dos parámetros que se corresponden a las coordenadas de la posición del punto en la que se sitúa el componente **isLogoDeporte**. El punto **x** tendrá como valor la posición **x** inicial del componente definida en el panel **Properties** del **Diseñador**, que es 0 y a la que se irán sumando 20 cada vez que sea ejecutado el procedimiento, el valor de **y** será un valor fijo de 20 para que el movimiento sea en sentido horizontal y hacia la derecha. Recuérdese que la propiedad **TimeInterval** se estableció con valor **100**, lo que quiere decir que, a cada décima de segundo, se ejecutará el procedimiento, que se traducirá en el correspondiente desplazamiento de la pelota y se parará al llegar al borde derecho del **Canvas** que se corresponde con el borde de la pantalla.

La imagen siguiente contiene los bloques que definen el movimiento de la pelota por el componente **Canvas**.

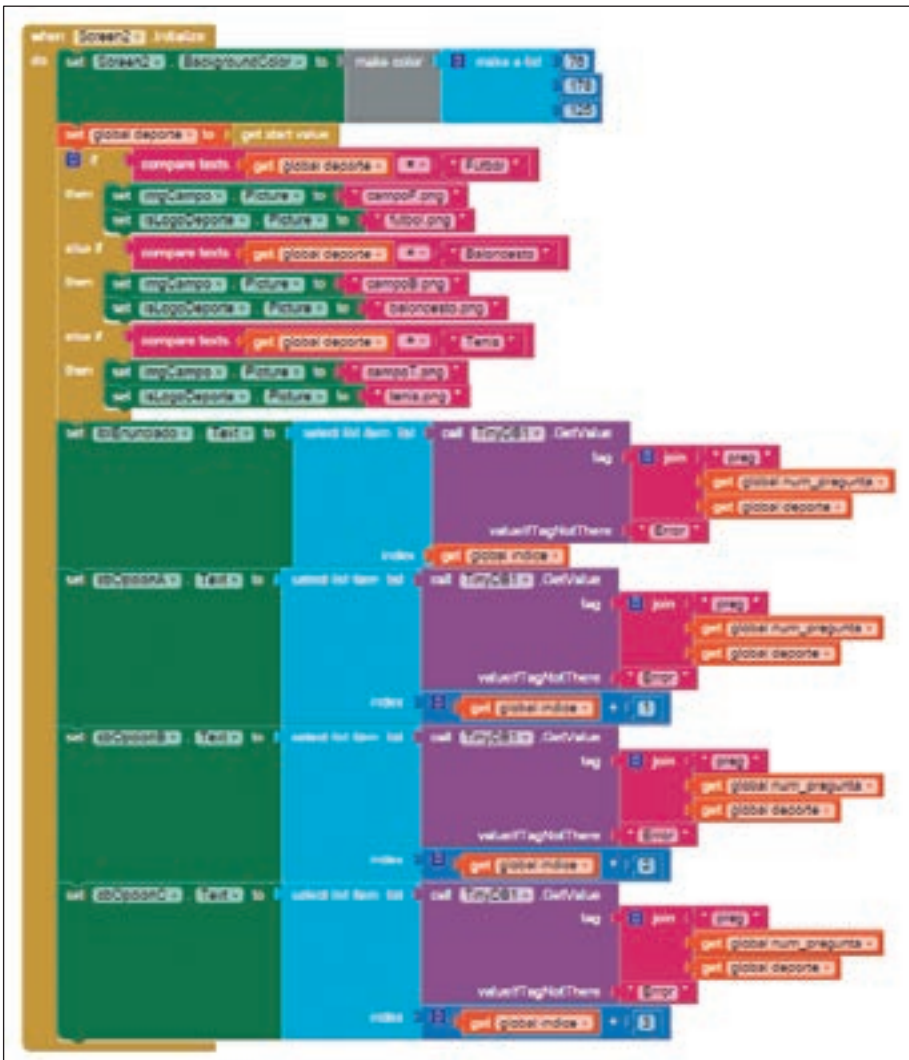


Bloques para el movimiento de la pelota

A continuación se añadirá el evento “Initialize” sobre el componente **Screen2** para definir una serie de acciones que se realizarán sobre la pantalla cada vez que se inicie. Entre estas acciones se puede encontrar la de asignar el color de fondo para la aplicación, recuperar el valor de la variable pasada como parámetro desde la pantalla **Screen1**, y asignarla a la variable **deporte**, para posteriormente poder realizar comparaciones con el objetivo de establecer las imágenes del balón y el campo del deporte elegido en la imagen inferior y en el

ImageSprite superior. Además de esto, se incluirán los bloques para recuperar primero y mostrar después, en etiquetas y cajas de selección, las preguntas almacenadas a través del componente **TinyDB1**.

Todas las acciones definidas para el evento “Initialize” se muestran en la siguiente imagen.



Evento “Initialize” del componente **Screen2**



Recuerde

Para que el movimiento del componente *ImageSprite* sea uniforme hacia la derecha se establecerá un valor fijo de 20 para la coordenada y.



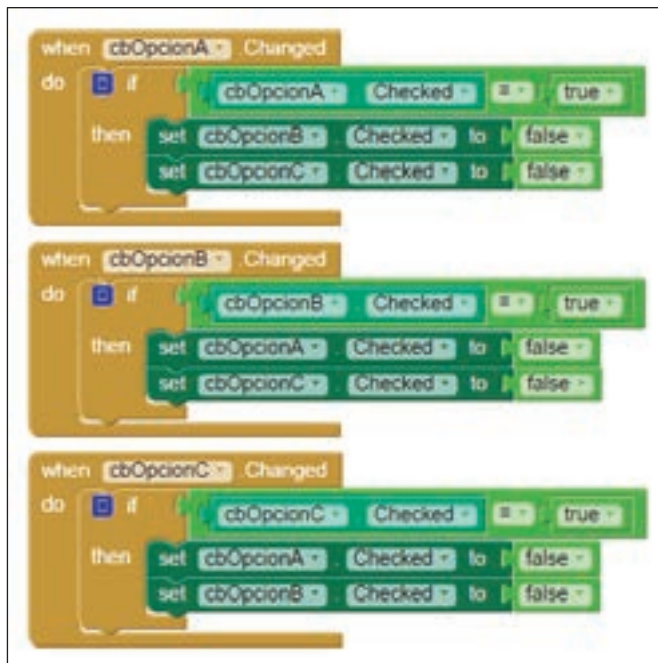
Actividades

2. ¿Qué ocurrirá si para el bloque *select list item*, en lugar de definir como index el valor de la variable *indice* más un valor incremental, se definiera el mismo valor para cada uno de ellos?
-

Entre los bloques definidos para el evento “Initialize” del componente **Screen2** se va resaltar el uso del bloque **get start value**, cuya utilidad es la de recuperar el valor pasado como parámetro desde la pantalla **Screen1** a través del bloque **open another screen with start value**. Por otro lado, para el uso del procedimiento **GetValue**, que recupera los datos almacenados con el componente **TinyDB1** en la pantalla anterior, se requiere que se especifique el nombre del **tag** a recuperar. Este **tag** será variable a lo largo de la ejecución de la aplicación y se construirá a través de la concatenación de tres cadenas de caracteres con el bloque **join** de la categoría **Text**, al que se le introducirán tres bloques **string**. El primero, **preg**, definido de manera explícita; el segundo será el resultado de recuperar el valor contenido en la variable **num_pregunta**; y el tercero, el resultado de recuperar el valor contenido en la variable **deporte**. De este modo, se obtendrá un **tag** como, por ejemplo, **preg4Tenis** o **preg2Futbol**. El valor almacenado es un valor de tipo lista, con lo que se observa además que, tras recuperar dicho valor, se deberá indicar el elemento concreto que se desea mostrar de la lista, es decir, hay que determinar qué valor se quiere recuperar de la lista de cuatro elementos, recordando que el primero es el enunciado de la pregunta y los tres siguientes las posibles respuestas. Esta

acción se lleva a cabo a través del bloque **select list ítem** de la categoría **List**, al que se le pasa como parámetro la lista y el índice del elemento a recuperar. De este modo, se irán mostrando las preguntas desde la primera hasta la última.

Una vez mostrada la primera pregunta, se requerirá la interacción del usuario para poder continuar. Este deberá, primero, seleccionar la respuesta que crea correcta, y segundo, hacer clic sobre el botón **Aceptar** para validar la respuesta y continuar a la siguiente pregunta. Para la primera de las acciones se deberá controlar que no se pueda elegir más de una respuesta a la vez. Para ello, añádase el evento “Changed” asociado a cada uno de los **CheckBox** mediante el cual se controla cada vez que una caja de selección cambia de estado, es decir, de seleccionada a no seleccionada y viceversa. Al detectar que un **CheckBox** ha sido seleccionado, se deseleccionan los otros dos. La acción que se acaba de definir se muestra en la siguiente imagen.



Eventos “Changed” de los componentes **CheckBox**

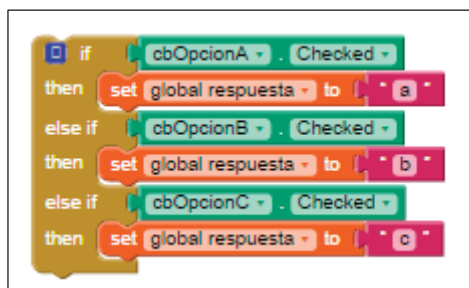
Elegida una respuesta por parte del usuario, deberá a continuación hacer click sobre el botón **Aceptar**. Este botón llevará a cabo una serie de operaciones que validarán la respuesta elegida, reproducirá un sonido anunciando el acierto o error de la respuesta, mostrará la siguiente pregunta reseteando las variables necesarias para ello y llevará el control sobre las preguntas realizadas y los errores mostrando el resultado final en caso de finalizar la batería de preguntas.



Recuerde

No se debe permitir la elección de más de una posible respuesta.

Todos los bloques correspondientes a la lógica explicada a continuación irán incluidos dentro del evento "Click" del botón **Aceptar**. En primer lugar, se comprobará la respuesta elegida y en función de ella se le asignará a la variable **respuesta** de tipo **Text** la letra en cuestión. Esto se podrá comprobar en la siguiente imagen.



Comprobación de la respuesta elegida

Seguidamente, se comparará la respuesta elegida con la respuesta correcta. Si es igual, se establecerá la propiedad **source** del componente **Player1** con el archivo de sonido “correcto.mp3”, que reproducirá el sonido “correcto”. En caso de no serlo, además de definir la propiedad **source** del componente **Player1** con el archivo de sonido “incorrecto.mp3”, que reproducirá el sonido “incorrecto”, se actualizará la variable **resultado** con el número de la pregunta errónea, además de incrementar en uno el número de errores totales. Por último, se reproducirá el sonido que corresponda a través del procedimiento **Start** del componente **Player1**. La lógica indicada se muestra en la siguiente imagen.



Elección de la locución



Actividades

3. ¿A través de qué bloque se recupera un valor pasado como parámetro?

La lógica correspondiente a la pregunta a mostrar comienza incrementando en uno el valor del número de la pregunta actual, siendo necesario para mostrar la pregunta siguiente, dado que inicialmente, al lanzar la pantalla **Screen2**, se muestra por defecto la primera de las preguntas de la batería al ser 1 el valor inicial de la variable **num_pregunta**. Tras incrementar el número de pregunta, se comprobará que no haya excedido el número total de preguntas, que es 5.

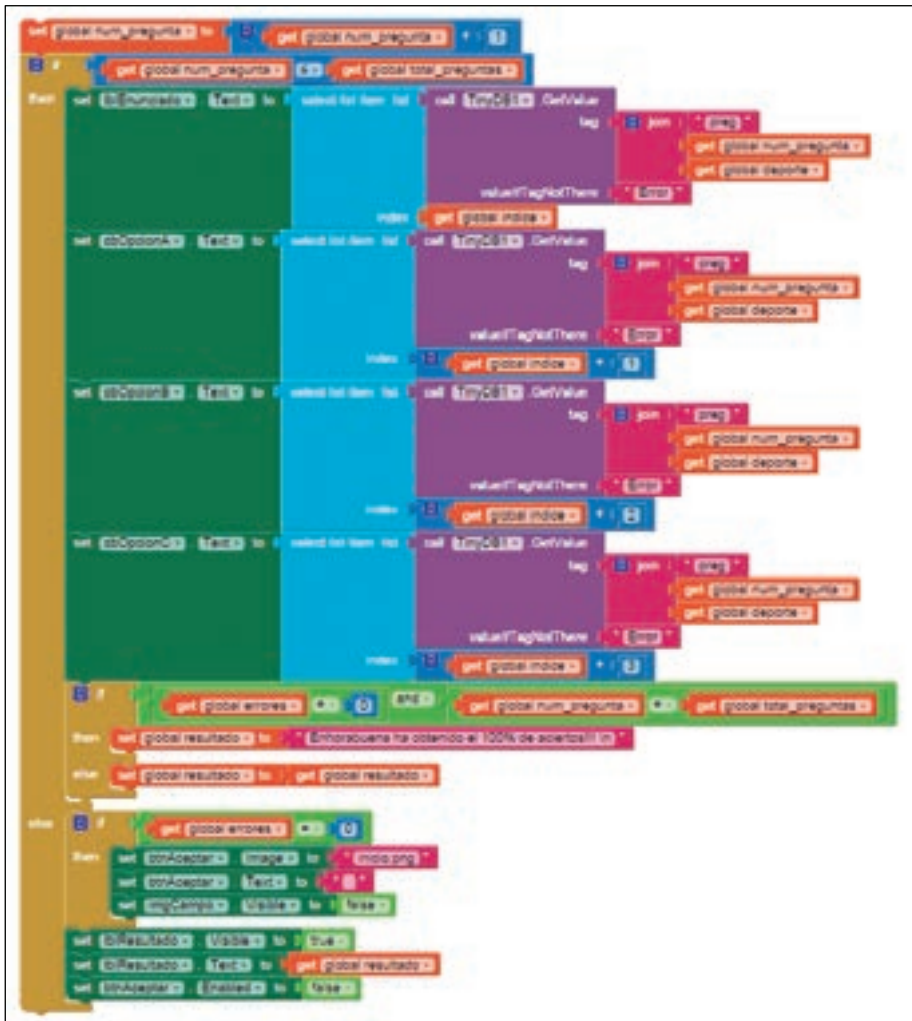
Si no se ha excedido, se pasará a mostrar la pregunta correspondiente del mismo modo que se hizo en el evento “Initialize” del componente **Screen2**. Acto seguido, se realizan dos comprobaciones que deben cumplirse ambas para almacenar, en la variable **resultado**, el mensaje “Enhorabuena, ha obtenido el 100 % de aciertos!!!”. En caso de no cumplirse alguna de ellas, se quedará igual la variable **resultado**, con los errores que tuviera. La primera de las comprobaciones es que el número de errores sea 0, y la segunda, que el número de pregunta sea igual al total de preguntas, lo que indica que se ha mostrado la última pregunta de la batería y ya se puede mostrar el resultado.

Si al incrementar en uno el valor de pregunta actual se excede el número total de preguntas, se pasa a comprobar si el número de errores es 0. En tal caso, se mostrará una imagen con la corona triunfal, sin texto, y se hará no visible el botón. Además, se hará visible la etiqueta donde se mostrará el resultado con el contenido almacenado en la variable **resultado** y, por último, se deshabilitará el botón **Aceptar** para que no se pueda usar en esta pantalla final.



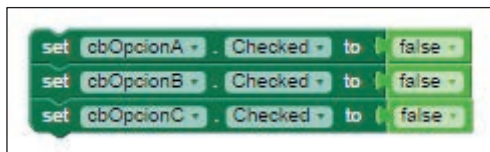
Actividades

4. ¿Qué ocurrirá si no se realiza la comprobación de que el número de una pregunta sea menor o igual que el número total de preguntas?
-



Se muestra la pregunta siguiente y se construye el resultado

Finalmente, tal y como se muestra a continuación, se establecerán con valor **false** la propiedad **Checked** de cada **CheckBox** para que se muestren desmarcadas en la próxima pregunta a mostrar si la hubiera.



Se desmarca la respuesta elegida



Nota

Desde la dirección <http://www.recursoadicionales.com/mit/ud7.apk> se puede descargar el fichero instalador de la aplicación que se acaba de realizar para comprobar en el dispositivo el resultado final.

El siguiente paso será probarla para comprobar que todo es correcto y se adapta a las especificaciones iniciales. Si no fuera así, habría que rehacer los pasos erróneos del desarrollo.

5. Resumen

La aplicación consta de dos pantallas: una primera, de selección de deporte, que hace las funciones de pantalla de inicio; y otra, que mostrará las preguntas sobre el deporte elegido. Cada una de ellas será un **Screen** diferente. Por defecto, el proyecto cuenta con uno y se deberá crear el segundo.

Para crear el proyecto se debe acceder desde el navegador a la dirección de *MIT App Inventor 2* que, como se aprendió en unidades anteriores, es <http://appinventor.mit.edu/>, elegir la opción **Create** e introducir los datos de usuario de la cuenta personal de Google. Una vez dentro de la aplicación, hacer clic sobre la opción de menú **Projects** y elegir **Start new projects...** Como nombre para la aplicación se puede elegir *unidad7*.

En la primera de las pantallas se mostrarán al usuario tres opciones, cada una con una imagen representativa de un deporte que, al tocarla, dará paso a la pantalla de preguntas del deporte elegido. En la segunda, se mostrarán una a una las preguntas definidas. Se deberá elegir una única respuesta, que se validará al pulsar sobre el botón **Aceptar**. En esta segunda pantalla se aprecia un componente **ImageSprite** con el aspecto del balón con el que se practica el deporte elegido, desplazándose por la parte superior de la pantalla, y una imagen de la pista sobre la que se practica el deporte en cuestión.

El primer paso del desarrollo de la aplicación será la definición del aspecto que tendrá la aplicación. En este se determinarán los componentes que formarán parte de ella y que serán elegidos en función de los requisitos de la misma. En segundo lugar, deberán establecerse las propiedades iniciales de cada componente con el fin de adaptarlo al diseño establecido.

Una vez diseñada la interfaz gráfica, se podrá continuar con el desarrollo de la aplicación definiendo los bloques que determinarán el comportamiento de la misma a través de la interacción del usuario con los componentes que se han añadido a la interfaz gráfica. Se recomienda que, cuando se inicie el desarrollo de una aplicación, se establezcan en primer lugar todos los componentes gráficos de su interfaz antes de continuar con el desarrollo de la lógica de la aplicación. Esto evitará tener que hacer y rehacer gran cantidad del trabajo.

Por último, se deberá probar la aplicación con el objetivo de comprobar que todo ha salido acorde a las especificaciones iniciales.

