

## Trabajando con Mapas

¿ Qué aprenderé ?.....	1
Diseñar los componentes.....	2
Configurar las propiedades de ActivityStarter.....	3
PRUEBA!!!!.....	5
Abrir los mapas con una búsqueda.....	5
PRUEBA!!!!.....	7
Configurar una ruta virtual.....	7
PRUEBA!!!!.....	11
Variaciones.....	12

En este capítulo construiremos un **guía personalizada para hacer un viaje a París**. Como nuestros mejores amigos no se han podido venir, crearemos una aplicación adicional que les permita hacer un recorrido virtual por la ciudad.

A primera vista, puede parecer complicado elaborar un programa que trabaje con los mapas de tu teléfono pero **AppInventor** nos dejará utilizar el componente **ActivityStarter** para abrir un mapa en **Google Maps** con cada ubicación virtual.

Primero, generaremos una herramienta que abra los mapas para la Torre Eiffel, el Louvre y la catedral de Notre Dame, todo pulsando un botón. Luego, modificaremos la aplicación para crear un paseo virtual por los mapas satélite que también están disponibles en Google Maps.

### ¿ Qué aprenderé ?

Este capítulo presenta los siguientes conceptos y componentes de **AppInventor**:

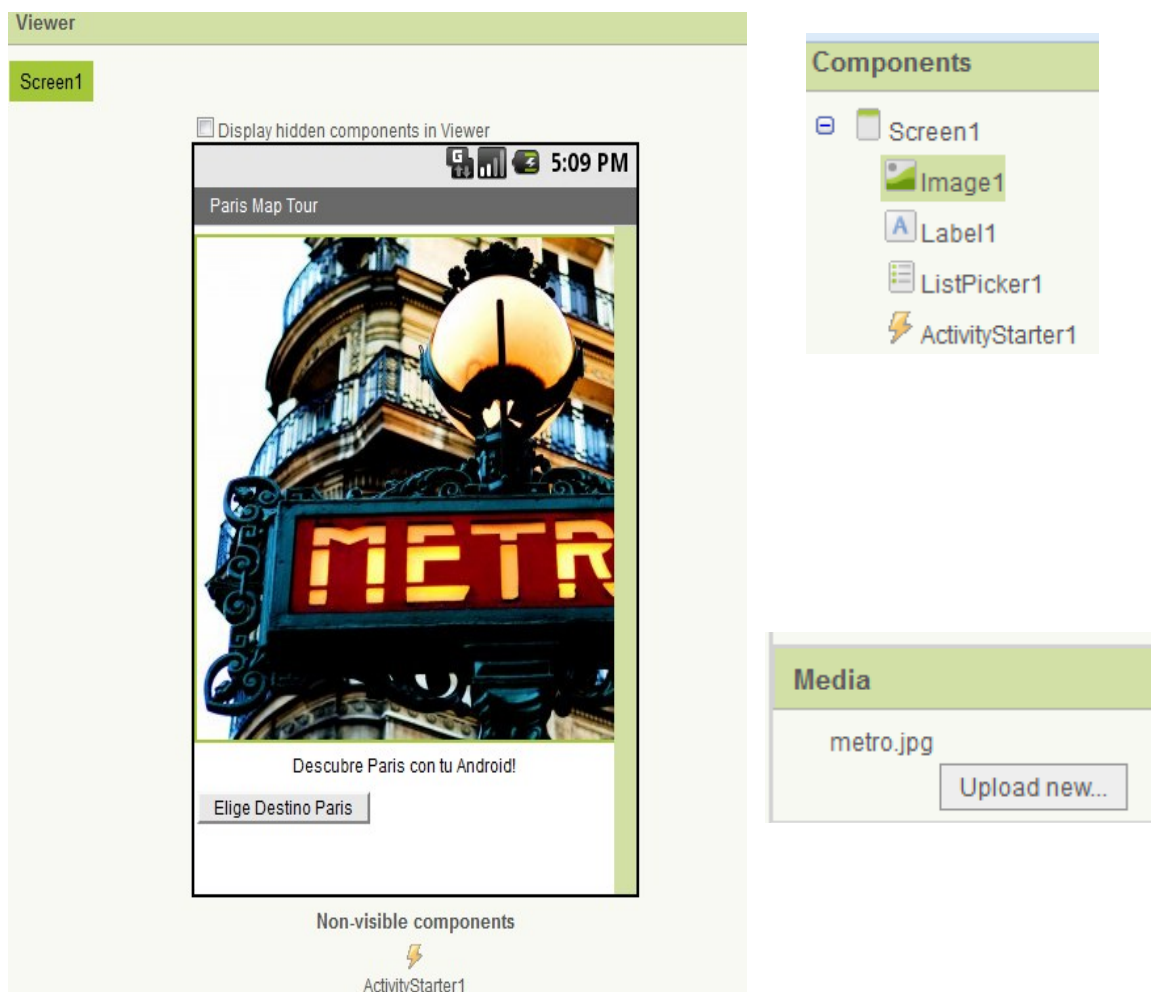
- El componente **ActivityStarter**, responsable de abrir otras aplicaciones Android. Lo usarás para recurrir a **Google Maps** con diferentes parámetros.
- El componente **ListPicker**, que permitirá al usuario seleccionar una ubicación de una lista.

## Diseñar los componentes

Crea un nuevo proyecto y llámalo **ParisMapTour**. La interfaz de usuario que empleará la aplicación será un componente **Image** con una fotografía de **París**. Agrega desde **Media** la imagen “*metro.jpg*”.

Añade un **Label** con el texto “**Descubre Paris con tu Android!**”(centrado y con la anchura de la pantalla del móvil, como ya hemos hecho en otros capítulos), un **ListPicker** que tendrá asociado un botón, todos ellos situados en la Paleta **Basic**. Agrega también un componente no visible llamado **ActivityStarter**, desde la paleta **OtherStuff**.

Cambia el **texto del Listpicker** y escribe “**Elige Destino Paris**” con anchura de pantalla del móvil igualmente. Cambia el **title de screen1** y escribe “**Paris Map Tour**”. El diseño de la pantalla será algo parecido a lo siguiente:



La siguiente tabla contiene todos los componentes que emplearemos. Arrástralos desde las secciones de **Palette** (paleta) hasta el **Viewer** (Visor) y asíñales el nombre que indicamos:

Tipo de componente	Grupo de Palette	Cómo lo llamaremos	Finalidad
<i>Image</i>	<b>Basic</b>	Image1	Muestra una imagen estática con el metro de París en la pantalla.
<i>Label</i>	<b>Basic</b>	Label1	Muestra el texto “Descubre Paris con tu Android!”.
<i>ListPicker</i>	<b>Basic</b>	ListPicker1	Muestra la lista de destinos posibles.
<i>ActivityStarter</i>	<b>Other stuff</b>	ActivityStarter1	Abre la aplicación Maps cuando se selecciona un destino.

## Configurar las propiedades de ActivityStarter

**ActivityStarter** es un componente que permite abrir en cualquier aplicación Android: un navegador, Google Maps o cualquier otra que tengamos instalada en el teléfono. Cuando un usuario abre una desde otro programa, puede regresar a la primera mediante el botón **Atrás**. Nosotros vamos a construir **Paris Map Tour** para que abra la aplicación **Maps** y muestre un mapa basado en la elección del usuario. Luego, utilizará el botón **Atrás** para volver a **Paris Map Tour** y seleccionar otro destino.

Para desarrollar esta aplicación debemos recurrir a las siguientes **propiedades** de **ActivityStarter** (escríbelas tal cual aparecen respetando las mayúsculas y minúsculas):

Propiedad	Valor
<i>Action</i>	android.intent.action.VIEW
<i>ActivityClass</i>	com.google.android.maps.MapActivity
<i>ActivityPackage</i>	com.google.android.apps.maps

Deberemos configurar varias propiedades desde **Blocks Editor**. **DataUri** permite abrir un mapa concreto de Google Maps. Esta propiedad se debe estructurar desde **Blocks Editor** en vez de hacerlo desde **Component Designer** porque necesita ser dinámica. Su valor cambiará en función de

la elección del usuario: *Torre Eiffel, el Louvre y la catedral de Notre Dame.*

Debemos definir una lista con los destinos y los comportamientos:

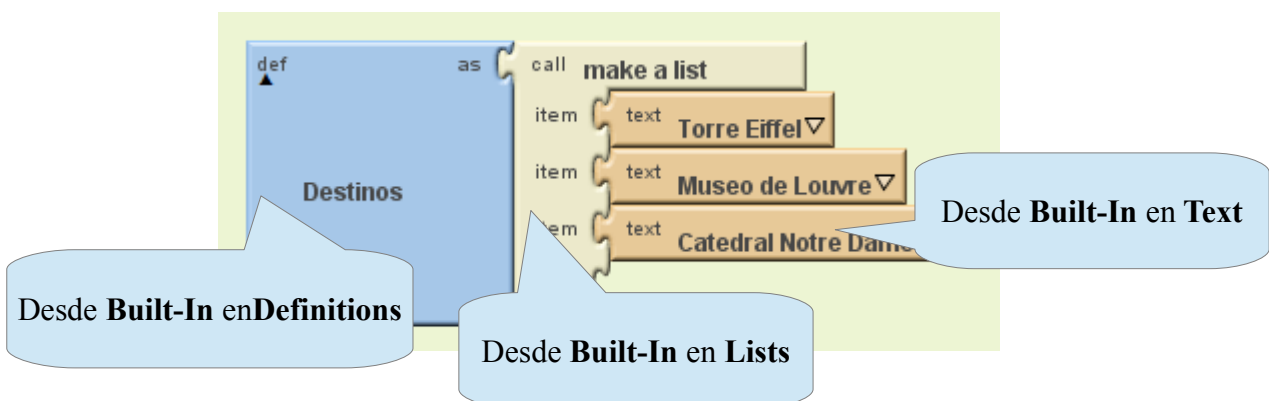
- Cuando se abra la aplicación, incorporarás los destinos que estén almacenados en el componente **ListPicker** para que el usuario elija uno de ellos.
- Cuando lo hayas elegido, se abrirá la aplicación **Maps** y mostrará la ubicación de dicho destino. En la primera versión de la aplicación abriremos **Maps** y le pediremos que busque el destino seleccionado.

Ahora, vamos a crear una lista de destinos...

Abre **Blocks Editor** y crea una variable con los destinos de **Paris**. Utiliza la siguiente tabla:

Tipo de bloque	Cajón	Finalidad
<i>def variable (“Destinos”)</i>	<b>Definitions</b>	Crea una lista con los destinos.
<i>make a list</i>	<b>Lists</b>	Agrega los elementos a la lista.
<i>text (“Torre Eiffel”)</i>	<b>Text</b>	El primer destino.
<i>text (“Museo de Louvre”)</i>	<b>Text</b>	El segundo destino.
<i>text (“Catedral Notre Dame”)</i>	<b>Text</b>	El tercer destino.

La variable **destinos** llamará a la función **make a list**. Dentro de ella habremos colocado el texto de los tres lugares disponibles, como se muestra a continuación:

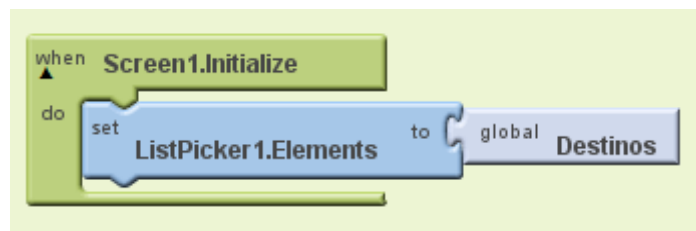


Ahora permitiremos que **el usuario seleccione un destino...**

Para incorporar los destinos dentro de **ListPicker** configuraremos la propiedad **Elements** como una lista. Haremos que el valor de la propiedad **Elements** de **ListPicker** sea la lista **destinos** que acabamos de crear.

Como queremos que se muestre al iniciarse la aplicación, debemos definir este comportamiento en el evento **Screen1.Initialize**. La siguiente tabla muestra los componentes que vamos a utilizar:

Tipo de bloque	Cajón	Finalidad
<i>Screen1.Initialize</i>	<b>Screen1</b>	Este elemento se iniciará cuando se abra la aplicación.
<i>set ListPicker1.Elements to</i>	<b>ListPicker1</b>	Asignaremos la lista a esta propiedad.
<i>global destinos</i>	<b>My Definitions</b>	La lista con los destinos.



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

**PRUEBA!!!!**

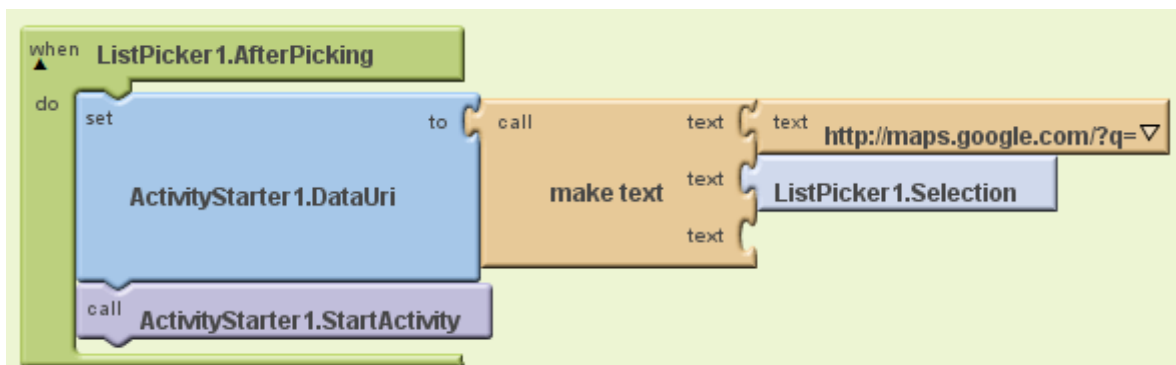
## Abrir los mapas con una búsqueda

Ahora programaremos la aplicación para que, cuando el usuario elija uno de los destinos, **ActivityStarter** abra **Google Maps** y lo busque allí. Cuando se escoge un elemento del componente **ListPicker**, se lanza el evento **ListPicker.AfterPicking**. Dentro del controlador de eventos **AfterPicking** deberemos configurar **DataUri** del componente **ActivityStarter** para que sepa qué mapa debe abrir.

Luego, utilizaremos **ActivityStarter.StartActivity** para abrir **Google Maps**. La siguiente tabla muestra todos los bloques que se emplearán para programar esta funcionalidad:

Tipo de bloque	Cajón	Finalidad
<i>ListPicker1.AfterPicking</i>	<b>ListPicker1</b>	Este evento se lanzará cuando el usuario seleccione un elemento de <b>ListPicker</b> .
<i>set ActivityStarter1.DataUri to</i>	<b>ActivityStarter1</b>	DataUri le dice a Maps qué mapa debe abrir.
<i>make text</i>	<b>Text</b>	Construye el contenido de <b>DataUri</b> a partir de dos segmentos de texto.
<i>text (“geo:0,0?q=”)</i>	<b>Text</b>	La primera parte de DataUri que debe recibir Maps.
<i>ListPicker1.Selection</i>	<b>ListPicker1</b>	El elemento que selecciona el usuario.
<i>ActivityStarter1.StartActivity</i>	<b>ActivityStarter1</b>	Abre Maps.

Cuando el usuario selecciona un elemento **ListPicker**, éste se guarda en **ListPicker.Selection** y se inicia el evento **AfterPicking**. Como muestra la figura de abajo, la propiedad **DataUri** se configura como un objeto de texto que combina la cadena <http://maps.google.com/?q=> con el elemento que ha seleccionado el usuario. Así pues, si elige el primer elemento de la lista, *Torre Eiffel*, el valor de DataUri sería `http://maps.google.com/?q=TorreEiffel`.



Como ya hemos definido las otras propiedades de **ActivityStarter** para que abra **Maps**, el

bloque **ActivityStarter1.StartActivity** abrirá dicha aplicación y solicitará que se efectúe la búsqueda almacenada en **DataUri**.

Guarda primero el proyecto con el nombre **ParisMapTour2**, se generará el fichero “**ParisMapTour2.apk**” que es el que debes copiar a tu teléfono.

**PRUEBA!!!!**

## Configurar una ruta virtual

Vamos a mejorar la aplicación e incorporar algunos elementos que enseñan fotografías de las calles y los monumentos de París. Así, los amigos que no pudieron acompañarnos también disfrutarán de nuestro viaje. Para ello, empezaremos navegando por **Google Maps** para obtener URL's de los mapas. Aún utilizaremos los mismos destinos de París pero cuando el usuario seleccione uno de la lista, indexaremos su elección (determinaremos la posición que ocupa) para marcar y abrir la calle o la vista ampliada en el mapa.

Tendremos que **localizar el DataUri de mapas concretos**. El primer paso será abrir **Google Maps en el ordenador** para localizar los mapas que quieres utilizar con cada uno de los destinos:

1. Escribe en el navegador Web <http://maps.google.com>
2. Localiza el monumento (por ejemplo, la Torre Eiffel).
3. Amplía el mapa hasta el nivel deseado.
4. Selecciona el tipo de vista que quieres emplear (Satélite, Mapa o Street View).
5. Haz clic sobre el botón **Enlazar** que se encuentra en la esquina superior izquierda de la ventana de los mapas y copia la URL. Utilízala (o una parte de ella) para abrir el mapa desde la aplicación.

La siguiente tabla muestra las URL's que usaremos en nuestro programa en la visita virtual,

quizás no te salgan los mismos valores de URL (**\*consulta con el profesor**):

Monumento	URL del mapa
<i>Torre Eiffel</i>	<a href="https://maps.google.com/maps?q=torre+Eiffel&amp;hl=es&amp;ll=48.857332,2.299533&amp;spn=0.010941,0.018325&amp;hq=torre+Eiffel&amp;t=m&amp;z=15&amp;iwloc=A">https://maps.google.com/maps?q=torre+Eiffel&amp;hl=es&amp;ll=48.857332,2.299533&amp;spn=0.010941,0.018325&amp;hq=torre+Eiffel&amp;t=m&amp;z=15&amp;iwloc=A</a>
<i>Museo de Louvre</i>	<a href="https://maps.google.com/maps?q=museo+de+Louvre&amp;hl=es&amp;ll=48.860875,2.319574&amp;spn=0.043762,0.073299&amp;ll=48.858278,2.294254&amp;sspn=0.021882,0.03665&amp;hq=museo+de+Louvre&amp;t=m&amp;z=14&amp;iwloc=A">https://maps.google.com/maps?q=museo+de+Louvre&amp;hl=es&amp;ll=48.860875,2.319574&amp;spn=0.043762,0.073299&amp;ll=48.858278,2.294254&amp;sspn=0.021882,0.03665&amp;hq=museo+de+Louvre&amp;t=m&amp;z=14&amp;iwloc=A</a>
<i>Catedral de Notre Dame (Street View)</i>	<a href="https://maps.google.com/maps?q=catedral+de+Notre+Dame,+Par%C3%ADs,+Francia&amp;hl=es&amp;ie=UTF8&amp;ll=48.855906,2.340946&amp;spn=0.087533,0.146599&amp;ll=45.429657,-75.696626&amp;sspn=5.974985,9.382324&amp;oq=catedral+de+Notre+dame,+paris&amp;hq=catedral&amp;hnear=Notre+Dame,+Paris,+%C3%8Ele-de-France,+Francia&amp;t=m&amp;z=13&amp;iwloc=A&amp;layer=c&amp;cbll=48.853282,2.350126&amp;panoid=9Qzhofua3H9xb_nJHky2jA&amp;cbp=12,5.95,,0,6.67">https://maps.google.com/maps?q=catedral+de+Notre+Dame,+Par%C3%ADs,+Francia&amp;hl=es&amp;ie=UTF8&amp;ll=48.855906,2.340946&amp;spn=0.087533,0.146599&amp;ll=45.429657,-75.696626&amp;sspn=5.974985,9.382324&amp;oq=catedral+de+Notre+dame,+paris&amp;hq=catedral&amp;hnear=Notre+Dame,+Paris,+%C3%8Ele-de-France,+Francia&amp;t=m&amp;z=13&amp;iwloc=A&amp;layer=c&amp;cbll=48.853282,2.350126&amp;panoid=9Qzhofua3H9xb_nJHky2jA&amp;cbp=12,5.95,,0,6.67</a>

Puedes utilizar estas URL's directamente para abrir los mapas o mostrar las coordenadas GPS que aparecen en la URL de la tabla de arriba. Por ejemplo, puedes mostrar la **Torre Eiffel** empleando sólo las coordenadas GPS que aparecen en la URL de la tabla de arriba y el **protocolo geo: de Maps**:

**geo: 48.857332,2.299533?t=h&z=19**

Al proporcionar esta cadena a **DataUri**, básicamente tendremos el mismo mapa que la URL



de donde obtuvimos las coordenadas GPS. El valor **t=h** especifica que se debe usar una **visión híbrida de las vistas Mapa y Satélite**. (**t=m** indica **vista Mapa**). El valor **z=19** determina el **nivel de zoom**. Para practicar con los dos tipos de URL's, te recomiendo que emplees el formato **geo:** con las dos primeras configuraciones de **DataUri** y la **URL** completa con la tercera.

Después de esta explicación, completamos el código de nuestra aplicación a continuación:



- Para la **Torre Eiffel** utilizaremos: **geo: 48.857332,2.299533?t=h&z=19**
- Para el **Museo del Louvre** utilizaremos: **geo:48.860875,2.319574?t=h&z=18**
- Para el la **Catedral de NotreDame** utilizaremos la URL entera de la tabla de arriba:  
[https://maps.google.com/maps?q=catedral+de+Notre+Dame,+Par%C3%ADs,+Francia&hl=es&ie=UTF8&ll=48.855906,2.340946&spn=0.087533,0.146599&sll=45.429657,-75.696626&sspn=5.974985,9.382324&oq=catedral+de+Notre+dame,+paris&hq=catedral&hnear=Notre+Dame,+Paris,+%C3%8Ele-de-France,+Francia&t=m&z=13&iwloc=A&layer=c&cbll=48.853282,2.350126&panoid=9Qzhofua3H9xb\\_nJHky2jA&cbp=12,5.95,,0,6.67](https://maps.google.com/maps?q=catedral+de+Notre+Dame,+Par%C3%ADs,+Francia&hl=es&ie=UTF8&ll=48.855906,2.340946&spn=0.087533,0.146599&sll=45.429657,-75.696626&sspn=5.974985,9.382324&oq=catedral+de+Notre+dame,+paris&hq=catedral&hnear=Notre+Dame,+Paris,+%C3%8Ele-de-France,+Francia&t=m&z=13&iwloc=A&layer=c&cbll=48.853282,2.350126&panoid=9Qzhofua3H9xb_nJHky2jA&cbp=12,5.95,,0,6.67)

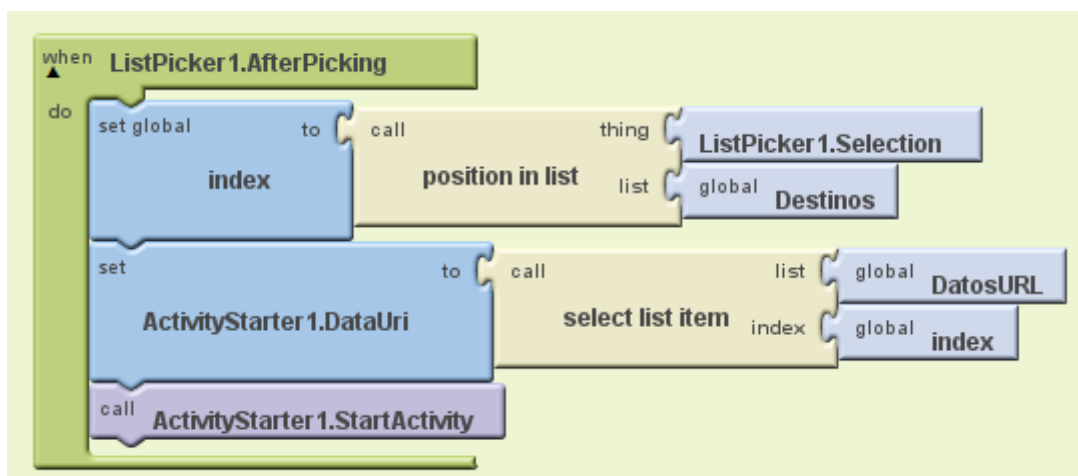
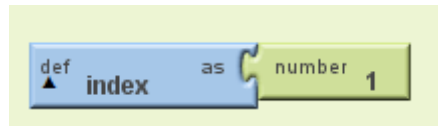
Para terminar vamos a modificar el comportamiento de **ListPicker.AfterPicking**....

En la primera versión de esta aplicación, el comportamiento **ListPicker.AfterPicking** asignaba a **DataUri** una cadena que se construía combinando la cadena <http://maps.google.com/?q=> y el destino que seleccionaba el usuario de la lista (por ejemplo, la *Torre Eiffel*).

En esta segunda versión, el comportamiento de **AfterPicking** deberá ser más sofisticado porque el usuario elegirá un elemento de la lista (**destinos**) pero el **DataUri** correspondiente se deberá escoger de otra lista (**DatosURL**). En breve, hablaremos más sobre la indexación. De momento, mostramos una tabla con los bloques que vamos a utilizar para seleccionar un elemento de la lista basándonos en la elección del usuario:

Tipo de bloque	Cajón	Finalidad
<i>def variable (“index”)</i>	<b>Definitions</b>	Esta variable guardará el número de la posición (valor del índice) del elemento seleccionado por el usuario.
<i>number (1)</i>	<b>Math</b>	Inicia la variable de indexación a 1.
<i>ListPicker1.AfterPicking</i>	<b>ListPicker1</b>	Cuando el usuario selecciona un elemento, se lanza este evento.
<i>set global index to</i>	<b>My Definitions</b>	Asigna a esta variable el valor de la posición del elemento seleccionado.
<i>position in list</i>	<b>Lists</b>	Obtiene la posición (índice) del elemento seleccionado.
<i>ListPicker1.Selection</i>	<b>ListPicker1</b>	El elemento seleccionado. Por ejemplo, Torre eiffel. Lo coloca dentro de la ranura <b>thing de position in list</b> .
<i>global destinations</i>	<b>My Definitions</b>	Coloca este elemento dentro de la ranura <b>list de position in list</b> .
<i>set ActivityStarter.DataUri</i>	<b>ActivityStarter</b>	Configura esto antes de iniciar la actividad que abrirá el mapa.
<i>select list item</i>	<b>Lists</b>	Selecciona un elemento de la lista <b>DataURIs</b> .
<i>global DataURIs</i>	<b>My Definitions</b>	La lista de <b>DataURIs</b> .
<i>global index</i>	<b>My Definitions</b>	Guarda la posición del elemento seleccionado.
<i>ActivityStarter.StartActivity</i>	<b>ActivityStarter</b>	Abre la aplicación Maps.

Cuando el usuario selecciona un elemento de **ListPicker**, se lanza un evento **AfterPicking**, como muestra la figura de abajo. El elemento escogido, por ejemplo, *la Torre Eiffel*, se encuentra en **ListPicker.Selection**. El controlador de eventos lo utiliza para localizar la posición del elemento elegido o su valor de **indexación** en la **lista destinos**. El índice se corresponde con la ubicación que tiene en la lista. Por eso, si el usuario selecciona *Torre Eiffel*, el índice será 1, si escoge *Museo de Louvre* el índice será 2 y si se decanta por *Catedral Notre Dame* de París, el valor será 3.



Podemos usar el índice para escoger un elemento de otra lista, en nuestro caso **dataURIs**, y para configurar dicha entrada como **DataUri** de **ActivityStarter**. Cuando lo hayamos hecho, se recurrirá a **ActivityStarter.StartActivity** para abrir el mapa.

Guarda primero el proyecto con el nombre **ParisMapTour3**, se generará el fichero “**ParisMapTour3.apk**” que es el que debes copiar a tu teléfono.

**PRUEBA!!!!**

## Variaciones

- Crea una ruta virtual de algún destino diferente, como por ejemplo, tu lugar de ocio o tu instituto.
- Haz una aplicación **Tour Virtual personalizada** que ayude al usuario a generar una guía de la localización que quiera. Para ello, deberás introducir el nombre de cada punto de destino junto con la URL de uno de sus mapas. Guarda toda esta información en una base de datos **TinyWebDB** y consigue que la aplicación trabaje con la información allí almacenada.