

## Trabajando con el sensor de Localización

¿ Qué aprenderé ?.....	1
Diseñar los componentes.....	2
Definir el comportamiento de los componentes.....	5
Mostrar la localización actual.....	5
PRUEBA!!!!.....	8
Memorizar la localización actual.....	8
PRUEBA!!!!.....	9
Mostrar en la pantalla la dirección memorizada.....	9
PRUEBA!!!!.....	12
Almacenar la dirección memorizada.....	12
Recuperar la dirección almacenada cuando se abre la aplicación.....	14
PRUEBA!!!!.....	16
Variaciones.....	16

En este capítulo construiremos una aplicación para localizar nuestro coche después de aparcarlo. Pulsaremos un botón cuando estacionemos el vehículo y Android usará el sensor de posición para guardar las coordenadas GPS y la dirección. Más tarde, cuando volvamos a abrir la aplicación, nos dará las indicaciones necesarias para llegar hasta el coche.

### ¿ Qué aprenderé ?

En esta aplicación aprenderás:

- Determinar la ubicación del dispositivo Android con el componente **LocationSensor**.
- Utilizar **TinyDB** para guardar esta información en una base de datos.
- Emplear el componente **ActivityStarter** para abrir **Google Maps** desde la aplicación y mostrar las indicaciones para llegar hasta la ubicación almacenada en la base de datos.

## Diseñar los componentes

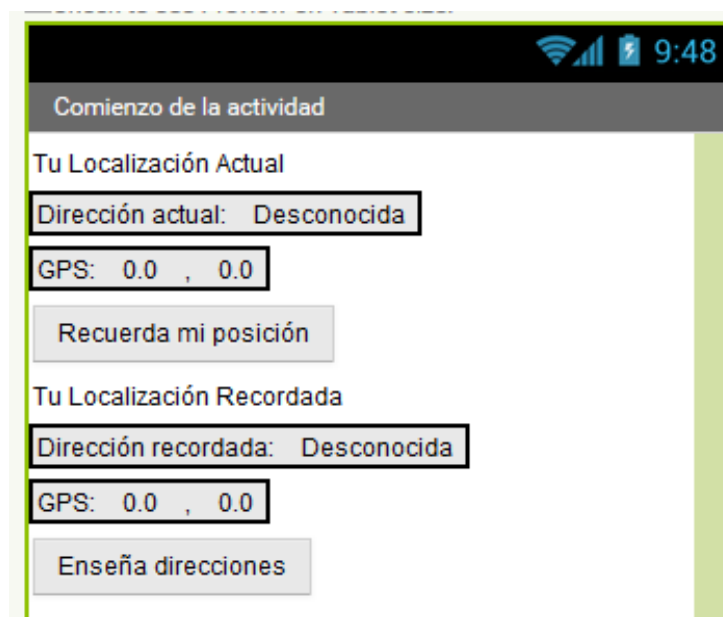
Crea un nuevo **proyecto** llamado **Localiza**, cambia el texto de la barra del título por **Comienzo de la actividad**.

La interfaz de usuario de la aplicación Android, **Localiza** está compuesta por una serie de etiquetas que mostrarán la ubicación actual y la almacenada en la memoria. Además, incluirá unos cuantos botones que permitirán guardar la localización actual e indicar las direcciones.

Necesitaremos algunas etiquetas para enseñar el texto estático. También trabajaremos con otras, como **Latitud**, que ofrecerá los datos recogidos del sensor de ubicación. A éstas les proporcionaremos un valor predeterminado (0,0), que se utilizará mientras el GPS adquiere información sobre nuestra posición.

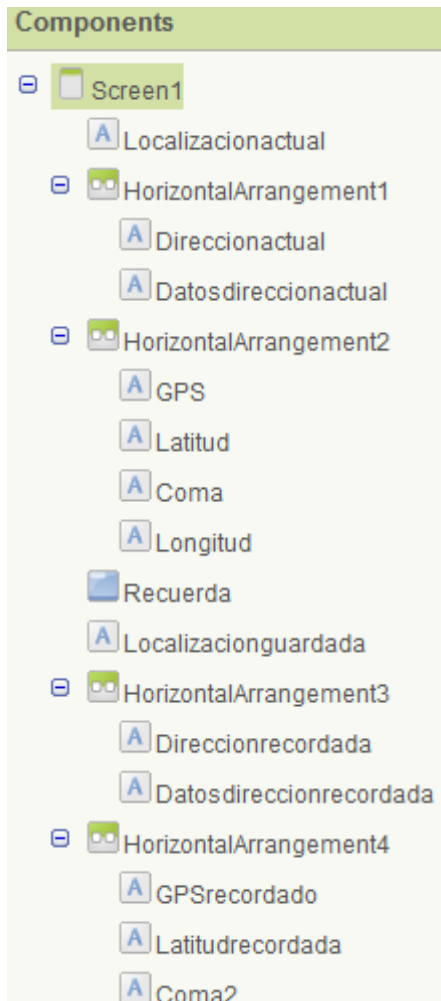
También nos hará falta 3 componentes no visibles: **LocationSensor** para obtener nuestra dirección actual, **TinyDB** para almacenar direcciones y **ActivityStarter** para abrir **Google Maps** y conseguir las indicaciones precisas para llegar desde una ubicación hasta otra.

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



Tipo de componente	Grupo de Palette	Cómo lo llamaremos	Finalidad
<i>Label</i>	<b>UserInterface</b>	Localizacionactual	Muestra “Tu Localización Actual”.
<i>Horizontal Arrangement</i>	<b>Layout</b>	<b>Horizontal Arrangement1</b>	Organiza la información.
<i>Label</i>	<b>UserInterface</b>	Direccionactual	Muestra el texto “Dirección”
<i>Label</i>	<b>UserInterface</b>	Datosdireccionactual	Enseña en la pantalla la dirección actual usando datos dinámicos.
<i>Horizontal Arrangement</i>	<b>Layout</b>	<b>Horizontal Arrangement2</b>	Estructura la información del GPS.
<i>Label</i>	<b>UserInterface</b>	GPS	Muestra el texto “GPS:” .
<i>Label</i>	<b>UserInterface</b>	Latitud	Muestra la latitud actual.
<i>Label</i>	<b>UserInterface</b>	Coma	Muestra en la pantalla “ , “.
<i>Label</i>	<b>UserInterface</b>	Longitud	Muestra la longitud actual.
<i>Button</i>	<b>UserInterface</b>	Recuerda	Haz clic en este botón para guardar la ubicación actual. Muestra “Recuerda mi posición”.
<i>Horizontal Arrangement</i>	<b>Layout</b>	<b>Horizontal Arrangement3</b>	Organiza la información sobre la dirección guardada.
<i>Label</i>	<b>UserInterface</b>	Localizacionrecordada	Muestra el texto “Tu localización recordada”.
<i>Label</i>	<b>UserInterface</b>	Direccionrecordada	Muestra el texto”Dirección recordada:”.
<i>Label</i>	<b>UserInterface</b>	Datosdireccionrecordada	Ofrece la dirección memorizada.
<i>Horizontal Arrangement</i>	<b>UserInterface</b>	<b>Horizontal Arrangement4</b>	Organiza la información sobre la dirección guardada.
<i>Label</i>	<b>UserInterface</b>	GPSrecordado	Muestra el texto”GPS:”.
<i>Label</i>	<b>UserInterface</b>	Latitudrecordada	Muestra la latitud memorizada.
<i>Label</i>	<b>UserInterface</b>	Coma2	Muestra en la pantalla “ , “.
<i>Label</i>	<b>UserInterface</b>	Longitudrecordada	Muestra la longitud memorizada.
<i>Button</i>	<b>UserInterface</b>	Direcciones	Haz clic para mostrar el mapa.
<i>Sensor</i>	<b>Sensors</b>	LocationSensor1	Obtiene información del GPS.

Tipo de componente	Grupo de Palette	Cómo lo llamaremos	Finalidad
<i>TinyDB</i>	<i>Storage</i>	TinyDB	Guarda en base de datos la dirección.
<i>ActivityStarter</i>	<i>Conectivity</i>	ActivityStarter1	Abre la aplicación Maps.



Vamos a configurar las propiedades de los componentes:

- La propiedad **Text** de las etiquetas (**Label**) que deban incluir un texto fijo se configura como dice la tabla anterior.
- La propiedad **Text** de las etiquetas que usen datos dinámicos del **GPS** deberá mostrar **0.0** .
- La propiedad **Text** de las etiquetas que utilicen datos dinámicos para las direcciones (“**Direccion**”) deberá mostrar el texto “**Desconocida**”.
- **Desactiva** la propiedad **Enabled** de los botones **Recuerda** y **Direcciones**.



- Configura las propiedades de **ActivityStarter** para que **ActivityStarter.StartActivity** abra **Google Maps**.

Éstas son las únicas visibles en la interfaz de usuario. La siguiente tabla te muestra las propiedades de **ActivityStarter** que has de rellenar:

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

### **Definir el comportamiento de los componentes**

- Cuando **LocationSensor** lea los datos del GPS, guardará esta información en las etiquetas apropiadas de la interfaz del usuario. Así, el usuario sabrá que el sensor ha obtenido la ubicación y que la aplicación está lista para memorizarla.
- Cuando el usuario presione el botón **Recuerda**, el programa copiará la información actual en las etiquetas para memorizarla. También necesitaremos almacenar la localización que se debe recordar para que no se pierda cuando el usuario cierre la aplicación. Cuando vuelva a abrirla, seguirá estando disponible.
- Cuando el usuario pulse en **Direcciones**, el programa abrirá **Google Maps** para mostrar la dirección de la posición memorizada.
- Cuando se vuelva a abrir la aplicación, abrirá la ubicación guardada desde la base de datos.

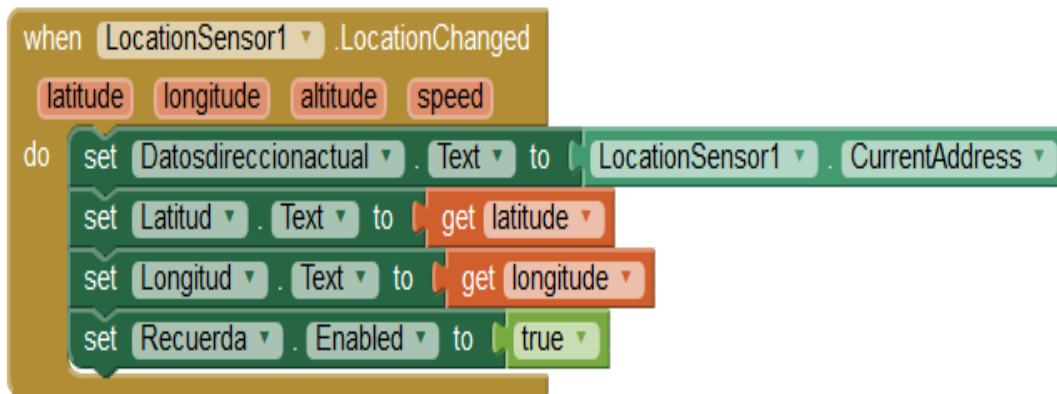
### **Mostrar la localización actual**

El evento **LocationSensor.LocationChanged** se lanzará no sólo cuando cambie la ubicación del dispositivo, sino también la primera vez que el sensor deba obtener información del GPS. A veces, esta primera lectura puede tardar varios segundos y, en ocasiones, no se obtiene porque no se consigue leer la señal de los satélites (dependerá de los ajustes del dispositivo).

Cuando tengamos información sobre la posición en la que nos encontramos, la aplicación deberá guardar estos datos en las etiquetas adecuadas. La siguiente tabla muestra los bloques que necesitaremos para ello:

Tipo de bloque	Cajón	Finalidad
<i>LocationSensor   LocationChanged</i>	<b>LocationSensor</b>	Se lanzará cuando el teléfono reciba una nueva lectura del GPS.
<i>set Datosdireccionactual.Text to</i>	<b>DatosDireccionActual</b>	Coloca la nueva información en las etiquetas destinadas a albergar la posición actual del dispositivo.
<i>LocationSensor   CurrentAddress</i>	<b>LocationSensor</b>	Esta propiedad obtendrá la dirección.
<i>set Latitud.Text to</i>	<b>Latitud</b>	Coloca la latitud en la etiqueta adecuada.
<i>value latitude</i>	<b>My Definitions</b>	Se deberá conectar a <b>set Latitud.Text to</b>
<i>set Longitud.Text to</i>	<b>Longitud</b>	Coloca la longitud en la etiqueta adecuada.
<i>value longitude</i>	<b>My Definitions</b>	Se deberá conectar a <b>set Longitud.Text to</b>
<i>set Recuerda.Enabled to</i>	<b>Recuerda</b>	Recuerda la información de la dirección actual.
<i>true</i>	<b>Logica</b>	Se deberá conectar a <b>set Recuerda.Enabled to</b>

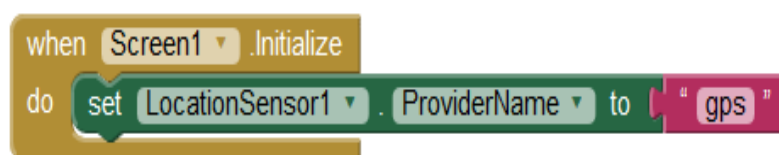
Cómo puedes ver en la figura de abajo, **latitude** y **longitude** son argumentos del evento **LocationChanged**, por lo que arrastraremos hasta el área de trabajo las referencias a ellos ubicadas en el cajón **My Definitions**.



**CurrentAddress** no es un argumento, sino una propiedad de **LocationSensor**, que encontraremos en el cajón **LocationSensor**. Éste, además, efectúa cierto trabajo adicional, ya que se encarga de llamar a la aplicación **Google Maps** para obtener la dirección postal de la posición GPS. Este controlador de eventos también activa el botón **Recuerda**.

En **Component Designer** hicimos que este botón se iniciase desactivado porque el usuario no tenía ninguna información que memorizar y el sensor todavía no había conseguido la posición GPS. Así pues, ha llegado el momento de programar su comportamiento.

Para indicarle al Localizador que vamos a usar el **GPS** necesitamos usar una propiedad de **LocationSensor1**.



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

## PRUEBA!!!!

### Memorizar la localización actual

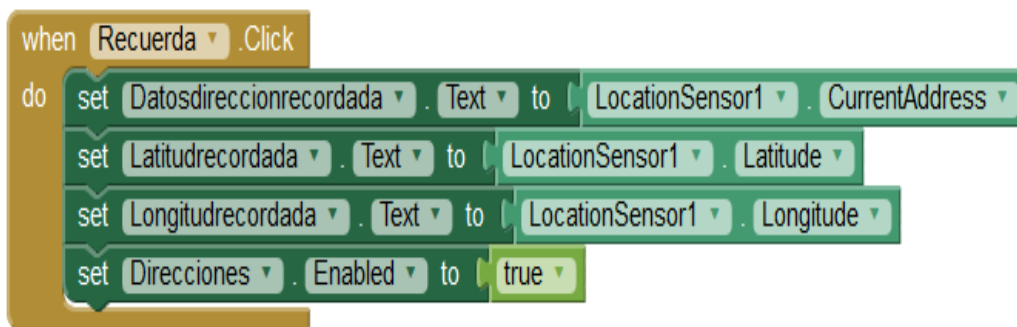
Cuando el usuario toque el botón **Recuerda**, la mayor parte de la información sobre la ubicación del dispositivo se deberá colocar en las etiquetas para que la aplicación muestre en la pantalla los datos memorizados. La siguiente tabla muestra los bloques necesarios para configurar esta funcionalidad:

Tipo de bloque	Cajón	Finalidad
<i>Recuerda.Click</i>	<b>Recuerda</b>	Se inicia cuando el usuario toca el botón con la etiqueta <b>Recuerda mi posición.</b>
<i>set Datosdireccionrecordada.Text to</i>	<b>Datosdireccionrecordada</b>	Guarda en las etiquetas los datos almacenados obtenidos por el sensor GPS con la dirección donde se encuentra el dispositivo.
<i>LocationSensor1.CurrentAddress</i>	<b>LocationSensor</b>	Esta propiedad obtendrá la dirección postal.
<i>set Latitudrecordada.Text to</i>	<b>Latitudrecordada</b>	Coloca la latitud en la etiqueta <b>Latitudrecordada.</b>
<i>LocationSensor.Latitude</i>	<b>LocationSensor</b>	Se deberá conectar a <b>set RememberedLatLabel.Text to</b>
<i>set Longitudrecordada.Text to</i>	<b>Longitudrecordada</b>	Coloca la longitud en la etiqueta <b>Longitudrecordada.</b>
<i>LocationSensor.Longitude</i>	<b>LocationSensor</b>	Se deberá conectar a <b>set RememberedLongLabel.Text to</b>
<i>set Direcciones.Enabled to</i>	<b>Direcciones</b>	Obtiene el mapa de la dirección memorizada.



Tipo de bloque	Cajón	Finalidad
<i>true</i>	<b>Logic</b>	Se deberá conectar a <b>Direcciones.Enabled to</b>

Cuando el usuario presiona el botón **Recuerda**, la información conseguida por el sensor de ubicación se colocará en las etiquetas **recordadas**, como muestra la siguiente figura:



Es posible que te hayas fijado en que el botón **Direcciones** está activado. Puede ser un problema porque si el usuario lo toca antes de que se registre la nueva lectura del GPS, memorizará la dirección actual, por lo que el mapa que aparecerá en la aplicación no servirá para nada. Pero no parece lógico que la gente vaya a actuar así. Después de que el usuario camine un poco, las direcciones serán distintas.

Guarda primero el proyecto con el nombre **Localiza2**, se generará el fichero “**Localiza2.apk**” que es el que debes copiar a tu teléfono.

**PRUEBA!!!!**

## Mostrar en la pantalla la dirección memorizada

Cuando el usuario toque el botón **Direcciones**, queremos que nuestra aplicación abra Google Maps utilizando la dirección actual y la almacenada, es decir, aquella donde aparcamos el coche.

El componente **ActivityStarter** puede abrir cualquier programa **Android**, incluido **Google**

**Maps.** Para usarlo será necesario configurar algunos datos pero para abrir una navegador web o un mapa no hay que determinar demasiadas cosas.

Para abrir un mapa deberemos configurar la propiedad **ActivityStarter.DataUri**. Admite las mismas URL que escribimos en el navegador Web. Si quieres probarlo abre <http://maps.google.com> en el ordenador y pide que muestre las direcciones de Madrid y Barcelona. Cuando las veas en pantalla, haz clic en el botón **Enlazar** y comprueba la URL que aparece. Este tipo de URL será la que utilizaremos con nuestra aplicación.

La diferencia es que las direcciones que crearemos se generarán a partir de las coordenadas GPS, no del nombre de una ciudad. La URL deberá tener el siguiente formato:

**<http://maps.google.com/maps?saddr=37.82557,-122.47898&daddr=37.81079,-122.47710>**

Escríbela en tu ordenador. ¿Reconoces a dónde apunta?.

En nuestra aplicación tendremos que construir la URL y definir dinámicamente los parámetros de la dirección de origen (**saddr**) y la de destino (**daddr**). En los capítulos anteriores ya hemos sido capaces de concatenar cadenas de texto. Ahora haremos lo mismo, combinando la información del GPS de las direcciones memorizada y actual.

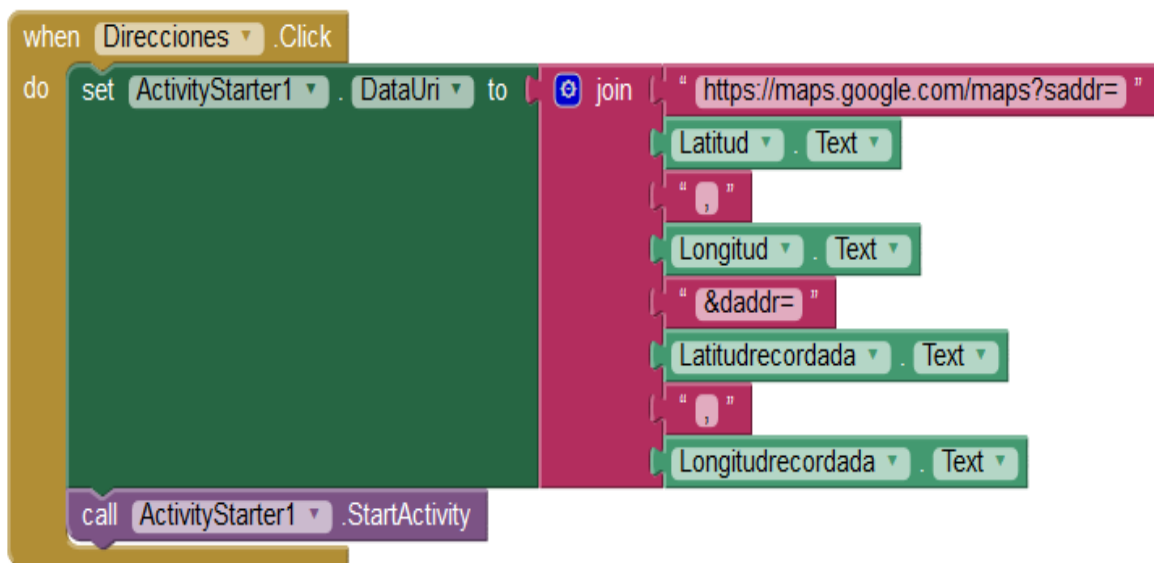
La que generaremos con estas cadenas la guardaremos en la propiedad **ActivityStarter.DataUri**. A continuación, llamaremos a **ActivityStarter.StartActivity**. La siguiente tabla muestra los bloques que precisamos:

Tipo de bloque	Cajón	Finalidad
<i>Direcciones.Click</i>	<b>Direcciones</b>	Se lanza cuando el usuario toca el botón <b>Directions</b> .
<i>set ActivityStarter.DataUri to</i>	<b>ActivityStarter</b>	Determina la URL del mapa que queremos recuperar.
<i>make text</i>	<b>Text</b>	Construye una URL a partir de varios elementos.

Tipo de bloque	Cajón	Finalidad
<i>text</i> (“ <a href="http://maps.google.com/maps?saddr=">http://maps.google.com/maps?saddr=</a> ”)	<b>Text</b>	La parte fija de la URL, la dirección del origen.
<i>Latitud.Text</i>	<b>Latitud</b>	La latitud de la posición actual.
<i>text</i> (“,”)	<b>Text</b>	Coloca una coma entre la latitud y la longitud.
<i>Longitud.Text</i>	<b>Longitud</b>	La longitud de la posición actual.
<i>Text</i> (“&daddr=”)	<b>Text</b>	El segundo parámetro de la URL, la dirección del destino
<i>latitudrecordada.Text</i>	<b>Latitudrecordada</b>	La latitud memorizada.
<i>text</i> (“,”)	<b>Text</b>	Coloca una coma entre la latitud y la longitud.
<i>Longitudrecordada.Text</i>	<b>Longitudrecordada</b>	La longitud memorizada.
<i>ActivityStarter.StartActivity</i>	<b>ActivityStarter</b>	Abre Maps.

Cuando el usuario pulsa el botón **Direcciones**, el controlador de eventos construye una URL para el mapa y llama a **ActivityStarter** para que se abra la aplicación **Maps** con el mapa, tal y como muestra la siguiente figura. Se utiliza el bloque **make text** para construir la URL que se enviará a Maps.

La URL final estará compuesta por el dominio de Maps (<http://maps.google.com/maps>) junto con los dos parámetros de la URL, *saddr* y *daddr*, que especificarán las direcciones de origen y destino. En este programa asignamos la latitud y la longitud de la dirección donde se encuentra el coche.



Guarda primero el proyecto con el nombre **Localiza3**, se generará el fichero “**Localiza3.apk**” que es el que debes copiar a tu teléfono

## PRUEBA!!!!

Cuando tengas los datos de la posición GPS, haz clic en el botón **Recuerda** y camina un poco. Al pulsar en **Direcciones** ¿aparece un mapa en la pantalla con indicaciones para volver a su punto de partida? Después de observarlo, emplea el botón Atrás unas cuantas veces. ¿Has conseguido regresar a la aplicación?.

## Almacenar la dirección memorizada

Llegados a este punto, tendremos una aplicación que funciona perfectamente bien. Es capaz de recordar la ubicación de un punto de origen y buscar un mapa que nos lleve desde el lugar en el que nos encontremos hasta dicha localización. Pero si el usuario quiere memorizar una dirección y luego cierra la aplicación, cuando vuelva a abrirla los datos se habrán perdido. Nuestro cliente desea que el programa recuerde dónde aparcó el coche. Para ello hemos de hacer algo parecido a lo que desarrollamos en la aplicación “**No escribas mientras conduces**”, es decir, guardar la información en la base de datos del teléfono mediante **TinyDB**:

- Cuando el usuario toque el botón **Recuerda**, la aplicación guardará la ubicación en la base de datos.

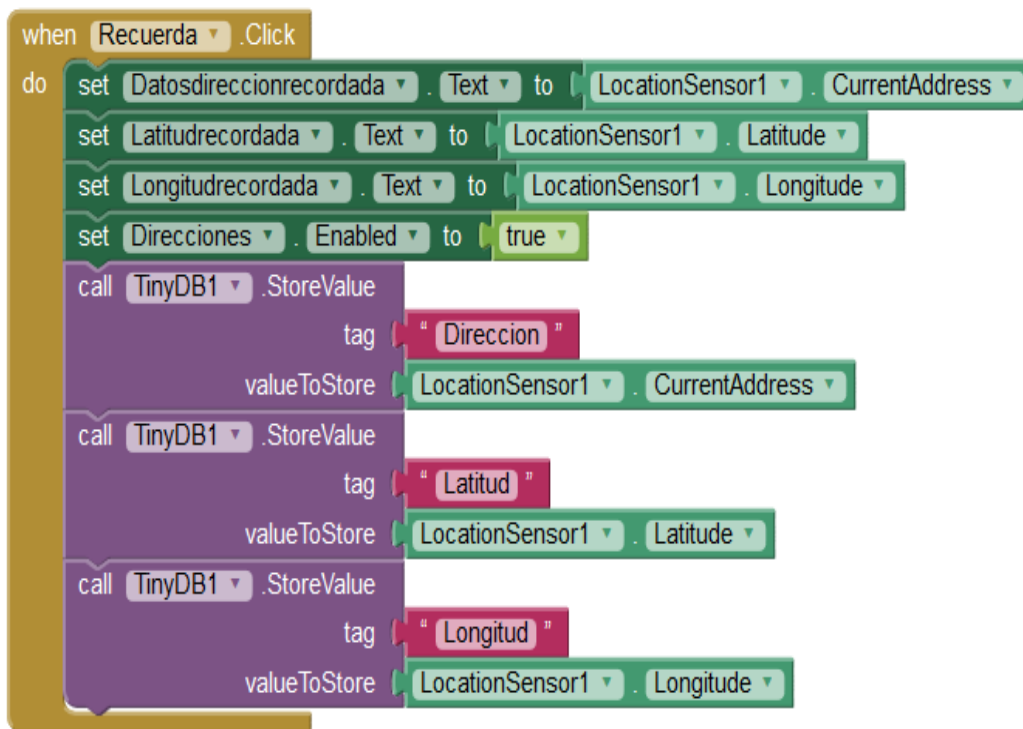
- Al volver a abrirla, tomará la posición almacenada y la abrirá en una variable o en una propiedad.

Empezaremos modificando el evento **Recuerda.Click** para que almacene en la base de datos la dirección memorizada. Para guardar la latitud, la longitud y la dirección deberemos efectuar 3 llamadas a **TinyDB.StoreValue**. La siguiente tabla muestra los bloques que añadiremos:

Tipo de bloque	Cajón	Finalidad
<i>TinyDB1.StoreValue (3)</i>	<b>TinyDB</b>	Guarda la información en la base de datos del dispositivo.
<i>text (“Direccion”)</i>	<b>Text</b>	Conecta este bloque a la ranura tag de <b>TinyDB1.StoreValue</b> .
<i>LocationSensor.CurrentAddress</i>	<b>LocationSensor</b>	Es la dirección que se deberá guardar en la base de datos. Conéctalo a la ranura value de <b>TinyDB1.StoreValue</b> .
<i>text (“Latitud”)</i>	<b>Text</b>	Conecta este bloque a la ranura tag del segundo <b>TinyDB1.StoreValue</b> .
<i>LocationSensor.CurrentLatitude</i>	<b>LocationSensor</b>	La latitud que se deberá guardar en la base de datos. Conéctalo a la ranura value del segundo <b>TinyDB1.StoreValue</b> .
<i>text (“Longitud”)</i>	<b>Text</b>	Conecta este bloque a la ranura tag del tercer <b>TinyDB1.StoreValue</b> .
<i>LocationSensor.CurrentLongitude</i>	<b>LocationSensor</b>	La longitud que se deberá guardar en la base de datos. Conéctalo a la ranura value del tercer <b>TinyDB1.StoreValue</b> .

Como se ve en la figura siguiente, **TinyDB1.StoreValue** copia los datos sobre la ubicación que están en la propiedad **LocationSensor** y los almacena en la base de datos del dispositivo. Como recordarás, la función **StoreValue** tiene dos argumentos: **tag** y **value**. El primero identifica los datos que vamos a guardar y el segundo contiene la información que se va a almacenar, en nuestro caso,

la de **LocationSensor**.

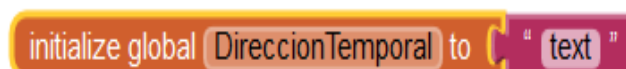


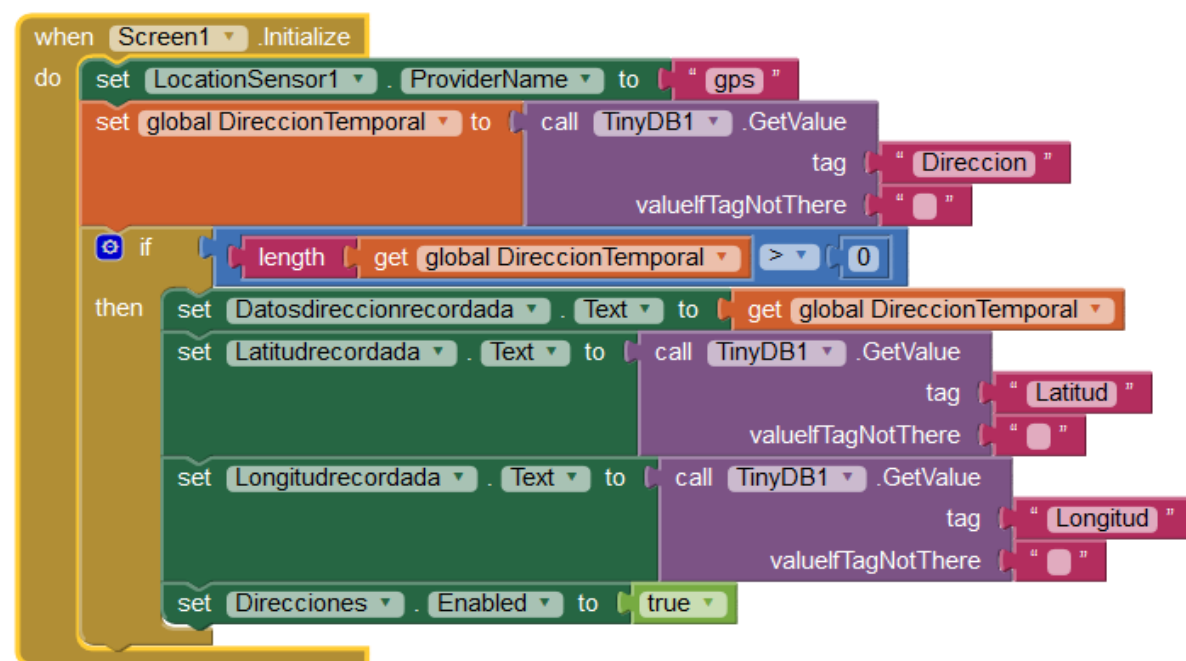
## Recuperar la dirección almacenada cuando se abre la aplicación

Cuando un usuario memoriza una ubicación y cierra el programa, queremos que estos datos estén disponibles al volver a abrirlo.

Cuando se activa la aplicación se lanza el evento **Screen.Initialize**, recuperaremos la información de la base de datos ahí. Utilizaremos la función **TinyDB.GetValue** para recuperar los datos del GPS almacenados.

Como tendremos que recuperar la dirección, longitud y latitud, necesitaremos llamar **3** veces a **GetValue**. Si es la primera vez que abrimos el programa, **TinyDB.GetValue** obtendrá un texto vacío, sin datos.





Para comprender el funcionamiento de estos bloques tenemos que ponernos en el lugar de un usuario que abre la aplicación por primera vez. Cuando esto ocurre, la base de datos del dispositivo no tiene ninguna ubicación anterior guardada, por lo que no es posible mostrar ninguna información en las etiquetas **recordada** de **Direcciones**. Después de utilizarla por primera vez, sí habrá datos disponibles en la base. Cuando se active la aplicación de nuevo deberá abrirlos. El bloque empieza llamando a **TinyDB1.GetValue** con una etiqueta de dirección (**direccion**), la que empleamos cuando almacenamos la ubicación. El valor que se recupera de la base de datos se guardará en la variable **DireccionTemporal**, donde confirmaremos si está vacía o no.

El bloque **If** es necesario porque **TinyDB** devolverá un texto vacío si no encuentra ningún dato asociado a la etiqueta que le hemos facilitado. Como ahora la variable **DireccionTemporal** alberga el valor que se ha obtenido de la base de datos, el bloque comprobará si la longitud de **DireccionTemporal** es mayor que 0. En caso afirmativo, la aplicación sabrá que **TinyDB** ha recuperado información y almacenará este valor en **Direccionrecordada**.

El programa también sabrá que, si se había obtenido una dirección postal deberá existir una latitud y longitud. Además, estos valores se podrán recuperar por medio de **TinyDB.GetValue**. Por último, si se ha rescatado alguna información de la base de datos, se activará el botón **Direcciones**.

Guarda primero el proyecto con el nombre **Localiza4**, se generará el fichero

“**Localiza4.apk**” que es el que debes copiar a tu teléfono.

## PRUEBA!!!!

Haz clic en el elemento **Recuerda** y asegúrate de que memoriza los datos sobre la ubicación. Luego, cierra el programa y vuelve a abrirlo. ¿Recuerda la última localización?

## Variaciones

- Crea la aplicación Android, **¿dónde está todo el mundo?** que permita a los usuarios saber dónde se encuentran sus amigos. Como se deberán compartir los datos que se empleen, utilizaremos una base de datos Web y el componente TinyWebDB en vez de TinyDB.
- Realiza un programa llamado **Miguitas de pan** que guarde en un registro de sus movimientos y cada cambio de ubicación en la lista. Sólo debería almacenar una miga de pan si la localización varía una cantidad mínima o si ha pasado un intervalo de tiempo determinado. Ten en cuenta que cualquier modificación de tu emplazamiento generará una lectura nueva. Deberás guardar toda la información en una lista.