

# NUEVO PARADIGMAS DE INTERACCIÓN

## PRÁCTICA 4 – gymkana

Jorge Chamorro Padial  
Germán Iglesias Padial

### 1 PuntoGPSVoz

La aplicación espera una entrada de audio con el siguiente formato:

[numero] grados [numero] minutos [numero] segundos [norte|sur|este|oeste]

Se decodifica, y se interpretan las coordenadas introducidas por el usuario. Acto seguido, y teniendo en cuenta una precisión prefijada anteriormente. Se indica al usuario hacia qué punto geográfico ha de avanzar.

Se pueden utilizar, como proveedores de localización, tanto la red wifi como el GPS. El sistema consta de una única clase, MainActivity.

Para este código hemos utilizado nuestro propio tutorial así como el de David Gasquez y José Miguel Morón Lopez. vamos a proceder a valorar los tres.

En primer lugar, el tutorial de David es muy completo y explica a buen nivel de detalle el código y para qué sirve. Explica, además, cosas que nosotros no hemos tenido en cuenta como el uso de diferentes proveedores de geolocalización, que nos ha sido muy útil en el desarrollo de este programa. Su principal carencia es que, al igual que nosotros, ha utilizado código que ya está catalogado como “deprecated” pero suponemos que eso se debe a que su dispositivo tiene una versión antigua (como el nuestro) y en ese momento no nos dimos cuenta de ese hecho.

<b>Tutorial analizado</b>	Brújula y GPS	
<b>Nombre del Compañero</b>	David Gasquez	
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	☺ ☺ ☺ ☺	El código es correcto, a pesar de estar “deprecated”
<b>Compleitud</b>	☺ ☺ ☺	Va al grano, aunque información adicional sobre aplicaciones directas de esto tales como la creación de localizaciones o el uso de mapas no estaría de más.

<b>Calidad de la documentación</b>	☺ ☺ ☺	
<b>Utilidad para el desarrollo de la aplicación propia</b>	☺ ☺ ☺ ☺ ☺	Todo lo que hemos necesitado se explica aquí
<b>Nota global</b>	☺ ☺ ☺ ☺	

<b>Tutorial analizado</b>	Geolocalización	
<b>Nombre del Compañero</b>	José Miguel Morón Lopez	Alfonso Ruiz-Bravo Jiménez
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	☺ ☺ ☺	El código es correcto y está bien explicado.
<b>Compleitud</b>	☺ ☺ ☺ ☺ ☺	Explica tanto la geolocalización como el uso de mapas
<b>Calidad de la documentación</b>	☺ ☺ ☺ ☺	Explica lo que hace, línea por línea y paso a paso
<b>Utilidad para el desarrollo de la aplicación propia</b>	☺ ☺ ☺ ☺	Igual que el anterior, todo lo aquí explicado lo hemos necesitado
<b>Nota global</b>	☺ ☺ ☺ ☺	

<b>Tutorial analizado</b>	Brújula y GPS (Nuestro tutorial)	
<b>Nombre del Compañero</b>	Jorge Chamorro Padial	Germán Iglesias Padial
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	☺ ☺ ☺	El código es correcto, aunque depreciaed
<b>Compleitud</b>	☺ ☺ ☺ ☺	Explicamos tanto la geolocalización como el uso de mapas. Pero nos han faltado detalles como el uso de proveedores
<b>Calidad de la documentación</b>	☺ ☺ ☺	La documentación del código es francamente mejorable. Pero la del pdf creemos que es adecuada.
<b>Utilidad para el desarrollo de la aplicación propia</b>	☺ ☺ ☺ ☺ ☺	
<b>Nota global</b>	☺ ☺ ☺	Leyendo nuestro propio tutorial, se nos ha hecho largo para esta práctica, que requiere más bien poco tiempo de “aprendizaje teórico”.

## 2 PuntoGestoQR

Se lee un gesto que se ha guardado previamente en una base de datos de gestos y luego se utiliza una biblioteca QR (requiere la instalación de un lector QR).

<b>Tutorial analizado</b>	QRReader	
<b>Nombre del Compañero</b>	Hugo Mario Lupión Fernández	Antonio Doncel Campos
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	😊 😊 😊 😊	El codigo presentado es correcto
<b>Compleitud</b>	😊 😊 😊	Se utiliza una biblioteca QR y se toma la imagen. Quizás se podría hablar de cómo reconocer otro tipo de códigos que también reconoce la biblioteca Metaio
<b>Calidad de la documentación</b>	😊 😊 😊	Estaría bien algo más de explicación tanto en comentarios teóricos como prácticos.
<b>Utilidad para el desarrollo de la aplicación propia</b>	😊 😊	<p>El uso de la biblioteca Metaio frente a Zxin no nos ha parecido acorde a la licencia que hemos escogido para nuestro proyecto (GNU GPL, que es viral, y por lo tanto incompatible con licencias shareware) así que hemos optado por usar la biblioteca libre frente a la escogida por los compañeros.</p> <p>Pero entendemos que esto es solo una decisión de diseño y no supone una valoración del código de los compañeros.</p>
<b>Nota global</b>	😊 😊 😊	

### 3 PuntoFotoBrújula

Existen dos brújulas, una que indica la posición objetivo y otra que indica la posición actual.

Este programa se ha probado enteramente con un emulador. Los tutoriales utilizados son exactamente los mismos que en PuntoGPSVoz así que poco más queda por añadir en esta sección salvo el detalle de que ningún tutorial hace una explicación sobre la implementación de la brújula en versiones recientes de Android. Para lo cual se hace necesario combinar dos sensores, el de gravedad y el acelerómetro (anteriormente se utilizaba solamente `Sensor.TYPE_ORIENTATION`, que está “deprecated”).

### 4 PuntoMovimientoSonido

Tutorial analizado	Acelerómetro	
Nombre del Compañero	Pedro Fernández Bosch	José Francisco Bravo Sánchez
	Puntuación	Comentario
Corrección	☺ ☺ ☺ ☺ ☺	El código es correcto y funciona perfectamente.
Compleitud	☺ ☺ ☺ ☺ ☺	Se explica, paso a paso, la implementación y qué hace cada cosa
Calidad de la documentación	☺ ☺ ☺ ☺	El PDF es muy completo, pero faltaría comentar algo más el código
Utilidad para el desarrollo de la aplicación propia	☺ ☺ ☺ ☺ ☺	Nuestra aplicación se basa, prácticamente entera, en la biblioteca de clases que han realizado los compañeros.
Nota global	☺ ☺ ☺ ☺ ☺	Es un tutorial que nos ha ayudado muchísimo con este tema.

Tutorial analizado	Gyroscope
--------------------	-----------

<b>Nombre del Compañero</b>	Carla Marisa Lobo Simões	Ana Catarina Ribeiro Lobo
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	😊 😊 😊 😊	El código presentado es correcto
<b>Compleitud</b>	😊 😊 😊 😊 😊	El trabajo realizado es completo en todos los sentidos, y el código, aunque algo complicado, ilustra sobre el uso del giroscopio en diferentes ámbitos, dispositivos, etc....
<b>Calidad de la documentación</b>	😊 😊 😊 😊	El código es tedioso y se podría comentar más. El PDF es completísimo, además teniendo en cuenta que las compañeras son Erasmus tiene un mérito extra expresarse tan claramente y con un lenguaje tan amplio. Explicando desde los fundamentos teóricos hasta la implementación a nivel práctico.
<b>Utilidad para el desarrollo de la aplicación propia</b>	😊 😊 😊	A nivel práctico no hemos usado su código pero a nivel teórico nos hemos enterado muy bien de qué estábamos haciendo. A pesar de que al final usamos más el acelerómetro que el giroscopio.
<b>Nota global</b>	😊 😊 😊 😊	

<b>Tutorial analizado</b>	Tutorial Acelerómetro	
<b>Nombre del Compañero</b>	Oliver Sánchez Marín	
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	😊 😊 😊 😊	El código presentado es correcto
<b>Compleitud</b>	😊 😊 😊	Esta práctica muestra un funcionamiento simple del acelerómetro. Como la extracción de movimientos por cada eje. Pero no va más allá de eso como los otros compañeros si han hecho.

<b>Calidad de la documentación</b>	😊 😊 😊 😊	<p>La documentación es muy didáctica y utiliza ejemplos y otros recursos.</p> <p>El código no cumple la convención javadoc y estaría bien al menos comentar lo que hace cada método. Sin embargo se hace un comentario línea a línea bastante clarificador.</p>
<b>Utilidad para el desarrollo de la aplicación propia</b>	😊 😊 😊	Al igual que con la valoración anterior, nos ha servido para aprender conceptos teóricos que nos han permitido no perdernos demasiado al programar el acelerómetro..
<b>Nota global</b>	😊 😊 😊 😊	

<b>Tutorial analizado</b>	Reproductor de música con el acelerómetro	
<b>Nombre del Compañero</b>	José Arcos Aneas	
	<b>Puntuación</b>	<b>Comentario</b>
<b>Corrección</b>	😊 😊 😊 😊 😊	El código es correcto.
<b>Complejidad</b>	😊 😊 😊 😊 😊	La aplicación no se limita simplemente a probar el acelerómetro sino que además da un ejemplo práctico y relativamente original del mismo.
<b>Calidad de la documentación</b>	😊 😊 😊 😊	La documentación explica paso a paso qué se ha de realizar y para qué funciona cada cosa. A veces algunos comentarios no definen mucho lo que está pasando con el código pero en general son ilustrativos.
<b>Utilidad para el desarrollo de la aplicación propia</b>	😊 😊	<p>No hemos utilizado el código del compañero porque ya tenemos un “exceso” de tutoriales de este tipo. Lo empezamos a consultar para ver si había algo nuevo que pudiéramos extraer.</p> <p>Lo que no significa que su solución sea mala</p>
<b>Nota global</b>	😊 😊 😊 😊	No hemos tenido en cuenta la utilidad en nuestra práctica por los motivos

		anteriormente comentados. Es un tutorial muy bueno y la idea de tener un reproductor de música con el acelerómetro motiva a leerlo.
--	--	-------------------------------------------------------------------------------------------------------------------------------------

## 5 PuntoExtra

Se han probado tres sensores:

- El de gravedad
- El de luminosidad.
- Ruído

Este último no es un sensor propiamente dicho, pero nos ha parecido que un medidor de dB es una gran utilidad de nuestro móvil y que con él se pueden hacer cosas a la altura de lo que se podrían hacer con los sensores aquí estudiados.

## 6 Problemas y otras consideraciones

Nuestros móviles son demasiado antiguos y hemos tenido problemas para probar algunos sensores. Principalmente el de **movimientos, geolocalización y brújula**. Los dos primeros funcionan, pero no son demasiado precisos (existe un error altísimo entre la ubicación estimada y la real). Y el de la brújula directamente no nos ha funcionado. Instalando aplicaciones de la tienda de Google hemos obtenido los mismos resultados.

Hemos implementado y probado las soluciones por medio de emuladores, pero hasta el día del examen no vamos a ver el funcionamiento en un entorno real.