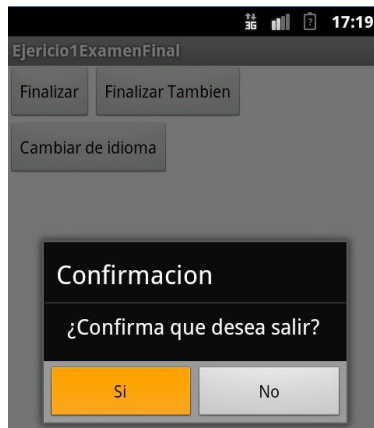


## 1. EJERCICIOS

Ejercicio 9.	. Menús de aplicación y contextuales. Y Diálogo .....	2
Ejercicio 10.	Menús de aplicación y contextuales. Diálogo confirmación. ....	2
Ejercicio 11.	ListView no personalizados, Activitys y distribución GridView .....	2
Ejercicio 12.	ListViewPersonalizado .....	3
Ejercicio 13.	Distribuciones TableLayout y GridLayout .....	4
Ejercicio 14.	.....	4
Ejercicio 15.	.....	5
Ejercicio 16.	.....	6
Ejercicio 17.	Repaso Activitys. ....	6
Ejercicio 18.	Shared Preference , SQLite y fragments .....	7
Ejercicio 19.	SQLite .....	7
Ejercicio 20.	. Fragments (ejercicio de examen).....	8
Ejercicio 21.	Fragments (ViewPager).....	9
Ejercicio 22.	Fragments (ViewPager).....	9
Ejercicio 23.	.....	11
Ejercicio 24.	.....	11
Ejercicio 25.	.....	11
Ejercicio 26.	.....	12
Ejercicio 27.	.....	13
Ejercicio 28.	.....	13

### Ejercicio 9. Menús de aplicación y contextuales. Y Diálogo



Deseamos hacer la siguiente aplicación:

- Menú de aplicación con las opciones Finalizar y Finalizar También (*acaban con la ejecución de la aplicación*).
- Al pulsar en los botones **Finalizar** y en **Finalizar También** harán exactamente lo mismo (optimizar el código, no vale copiar/pegar).
- Al pulsar en **Finalizar** y en **Finalizar También** (*cualquiera de las diferentes opciones que hay de hacerlo*) mostrará un diálogo al usuario para preguntar si desea finalizar la aplicación o no. Si pulsa en Si se verá un mensaje de despedida

cerrando la aplicación, si pulsa en No volverá a la pantalla principal.

- El botón **Cambiar de Idioma** modificará el idioma por defecto de la aplicación, colocándolo en inglés (como mínimo el texto de los tres botones).
- Cumplir con la distribución de componentes que se ve en la captura de pantalla.

Nota.- Ten en cuenta para el cambio de idioma las siguientes líneas:

```
Resources res = MainActivity.this.getResources(); DisplayMetrics dm =  
res.getDisplayMetrics();  
android.content.res.Configuration conf = res.getConfiguration(); conf.locale  
= new Locale("en"); res.updateConfiguration(conf, dm);
```

### Ejercicio 10. Menús de aplicación y contextuales. Diálogo confirmación.

Crear un menú de aplicación con al menos tres nombres de colores: rojo, amarillo y verde. Según elija uno u otro el panel de fondo de la aplicación se colocará del color elegido. Cuando se pulse durante un tiempo en el layout de fondo, se limpiará a través de un menú contextual el layout quedándolo en blanco.

Para todas las opciones pedir al usuario confirmación: ¿Está seguro sí/no?

Centro de Enseñanza Concertada  
Gregorio Fernández

### Ejercicio 11. ListView no personalizados, Activitys y distribución GridView

Realizar una lista de la compra a la cual se pueden añadir productos que aparecerán en el ListView interior. Al pulsar sobre un elemento de la lista se rellenarán los campos Nombre y Precio , que seguidamente si pulsamos en Borrar se eliminará de la lista. Finalmente Visualizar generará un listado de todos los productos en un **GridView de un segundo activity**.

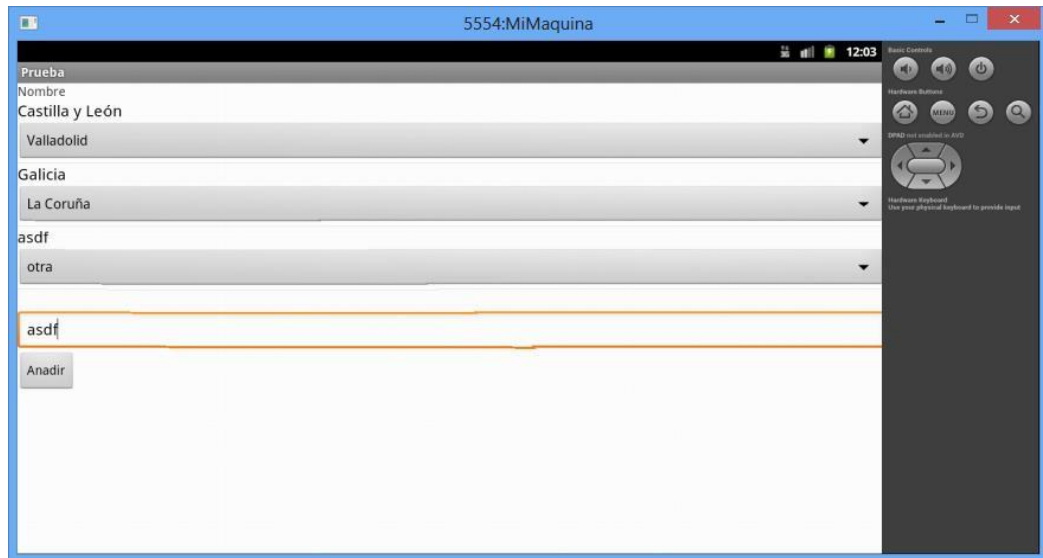
### *Ejercicio 12. ListViewPersonalizado*

Tenemos un ListView personalizado que muestra la comunidad autónoma, y las provincias de esa comunidad dentro de un Spinner. Tendremos una casilla en la parte inferior, que permitirá añadir al mismo tiempo Comunidad y Provincias de esa comunidad. El formato del texto a añadir será: Castilla y León;Valladolid;Zamora;Burgos;.....

Es decir, los valores están separados por ; siendo el primero de ellos la comunidad.

Al pulsar sobre el nombre de la comunidad del ListView se borrará ese elemento compuesto.





### Ejercicio 13. Distribuciones TableLayout y GridLayout

Realizar este interfaz según las distribuciones indicadas debajo de los mismos:

Table Layout	GridLayout

### Ejercicio 14.

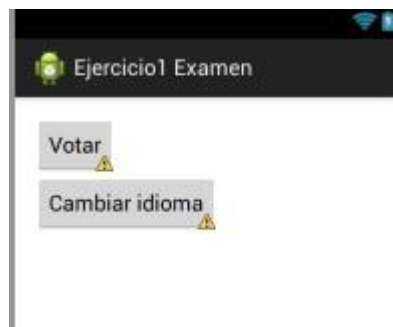
Centro de Enseñanza Concertada  
Gregorio Fernández

Creamos un activity inicial que llamará a través de un botón a un segundo activity con dos botones, Contar y Finalizar. El botón de Contar incrementará un contador cada vez que se pulse sobre él, después de pulsar en finalizar pasará al primer activity el valor contado y se mostrará en un textview.

Además en el activity inicial habrá un Layout donde según vayamos moviendo el dedo se irá viendo en un TextView la posición X,Y donde tenemos el dedo en ese momento.

### *Ejercicio 15. Activitys – Ministerio de Robología*

El Ministerio de Robología del gobierno de España encarga al centro Gregorio Fernández la realización de una aplicación móvil para la recogida de votos para las próximas elecciones.



- El botón Cambiar idioma permitirá cambiar de idioma del botón de Votar y del botón de Cambiar idioma, sólo por tanto de esta primera activity y sólo de estos dos botones. Desde el botón de votar se activará una segunda ventana, donde aparecerá un desplegable con los partidos políticos existentes: PSOE, Podemos, IU y PP. Tras seleccionar un partido y pulsar en Votar, se volverá al primer activity donde en un TextView inferior se irán mostrando las estadísticas de los votos realizados:

PSOE: x

Podemos: w

IU: y

PP: z

Centro de Enseñanza Concertada  
Gregorio Fernández

```
Resources res = MainActivity.this.getResources();  
DisplayMetrics dm = res.getDisplayMetrics();  
android.content.res.Configuration conf =  
res.getConfiguration(); conf.locale = new Locale("en");  
res.updateConfiguration(conf, dm);
```

### *Ejercicio 16.*

Generar un número aleatorio haciendo que constantemente se vaya mostrando en pantalla un número que simula el movimiento de un dado hasta finalmente para. Utilizar un progressbar hasta que finalmente se genera el número final:



Ejer10\_NumeroAsync

### *Ejercicio 17. Repaso Activities.*

Queremos realizar una aplicación con 4 botones como se ve en la captura, cada vez que pulse en los botones abrirá un segundo activity en donde se escribirá en un TextView el número del botón en el que se pulsó.

El segundo activity sólo se abrirá si el usuario pulsa primero en Aceptar en un diálogo de confirmación de Aceptar/Cancelar. Este segundo activity tendrá un botón de Cerrar para finalizar el activity y volver al primero.





### *Ejercicio 18. Shared Preference , SQLite y fragments*

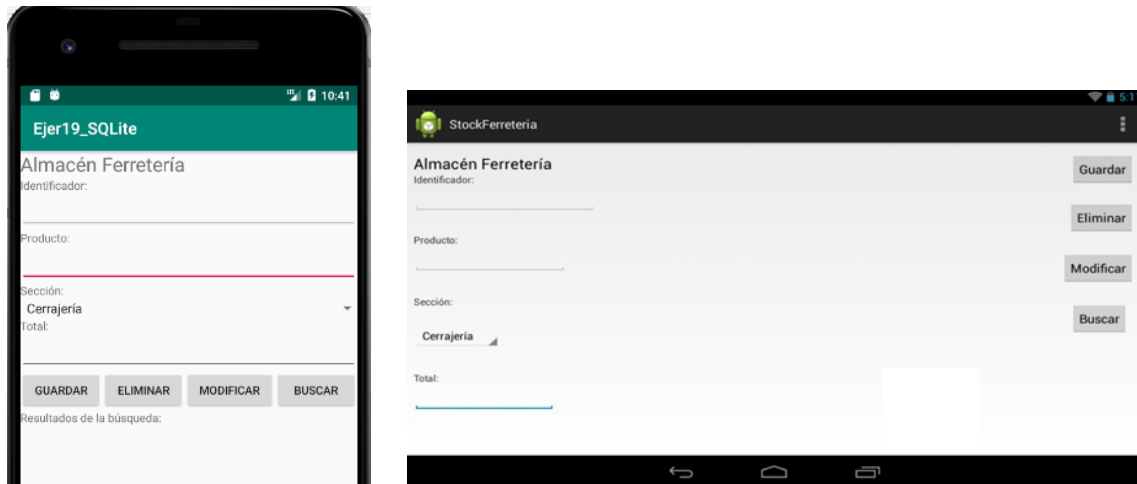
Se mostrará una aplicación, en la que se solicita el nombre y el número de teléfono del usuario. Ambos datos se almacenarán en el terminal móvil para que las próximas veces que inicie la aplicación se cargue en ambas casillas los datos anteriormente almacenados (SharedPreferences). Para guardarlos habrá un botón Guardar y para finalizar la aplicación un botón Finalizar.

Finalmente colocaremos un botón de Histórico (con Base de datos SQLite), para mostrar todos los nombres y teléfono en un listado dentro de un TextView de todos los usuarios almacenados (Base de datos), permitiendo a través de un botón Borrar Histórico, borrar todos los usuarios almacenados hasta el momento. Ese TextView estará incrustado en un Fragment.

### *Ejercicio 19. SQLite*

Realiza la siguiente aplicación que gestiona los artículos de una ferretería, tendrá dos interfaces diferentes según esté en horizontal o en vertical:

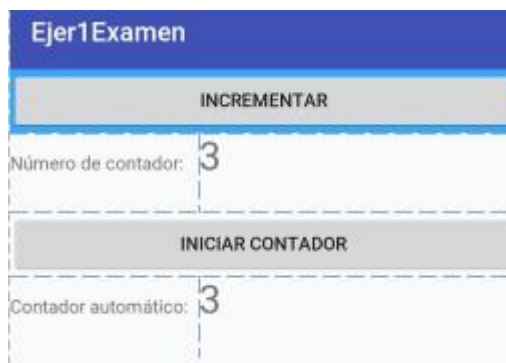




Se supone que todas las operaciones se realizarán a partir del identificado (*eliminar, modificar y buscar en función del identificador colocado*).

### *Ejercicio 20. . Fragments (ejercicio de examen)*

Desarrolla el siguiente interfaz teniendo en cuenta que el componente que tiene el botón de incrementar es un fragment.



Por otro lado el TextView numérico (*Número de contador*) está en el activity\_main.xml y se incrementará en una unidad por cada pulsación del botón INCREMENTAR. El listener del botón, la gestión de la pulsación, estará dentro de la clase Fragment.

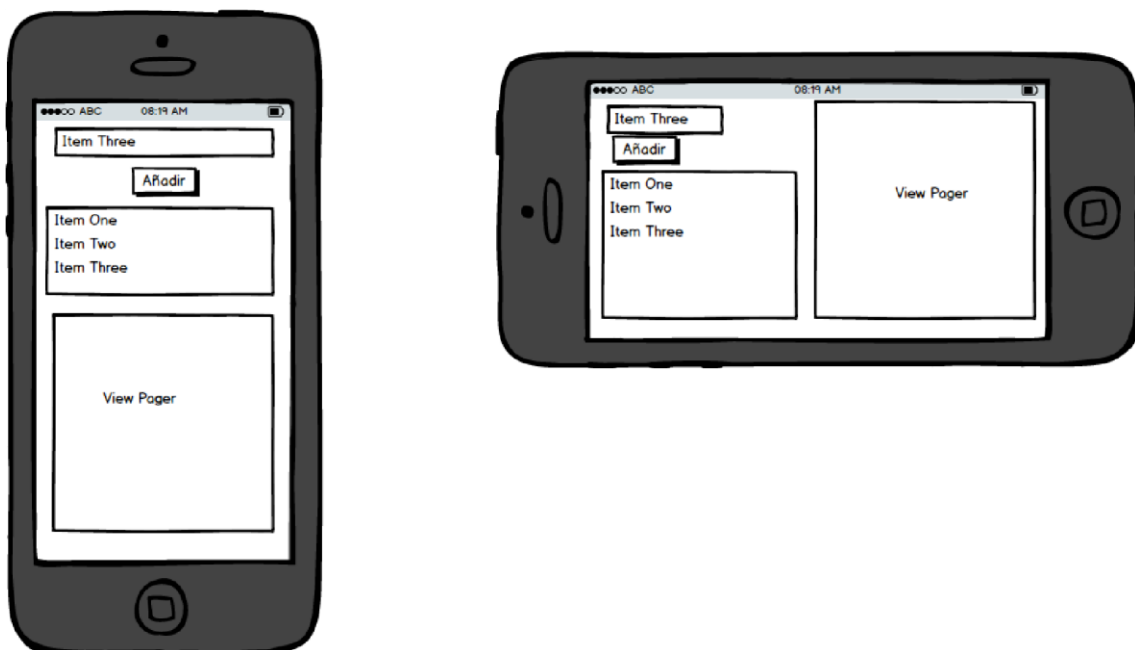
Centro de Enseñanza Concertada  
Gregorio Fernández



El botón de INICIAR CONTADOR hará que el contador inferior comience a incrementarse desde 0 hasta 20 incrementando el valor de 1 en 1 cada segundo. Debajo de ese último contador existirá una barra de progreso que llegue también de 0 a 20.

### *Ejercicio 21. Fragments (ViewPager)*

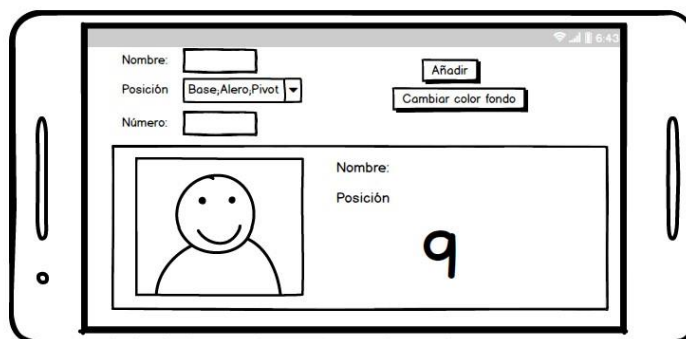
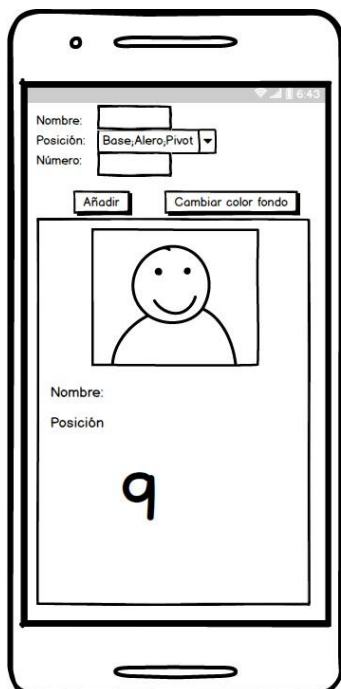
Insertamos desde Añadir el elemento de la casilla superior en un ListView inferior. Al mismo tiempo, tendremos un fragment que se muestra según esté en horizontal o vertical según la imagen:



Dentro de cada uno de los Fragments meteremos el texto de los elementos de la lista.

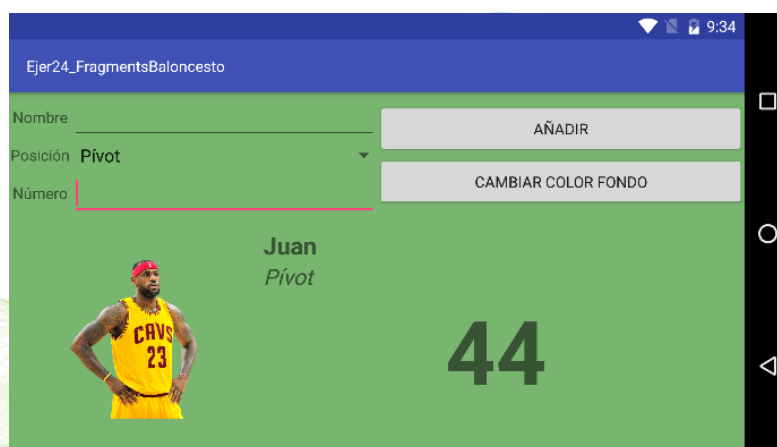
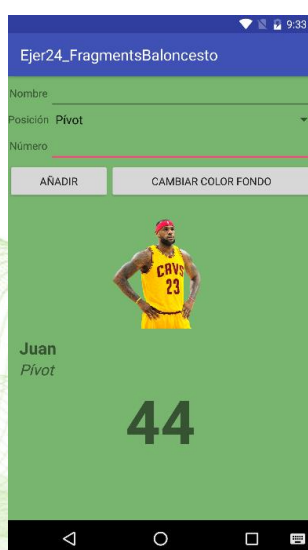
### *Ejercicio 22. Fragments (ViewPager)*

Queremos visualizar según la posición del móvil estos dos interfaces:



Debemos implementar las zonas (arriba, abajo, en horizontal o vertical) con fragments. Cuando pulsar en Añadir, se añade en BD los datos de nombre, posición y número. Esos datos se asociarán a un fragment en la parte inferior (recoge por tanto toda la info de la BD).

Por último el botón de Cambiar color fondo, cambiar el color de fondo de la aplicación, para que siempre se inicie por defecto con el color elegido.



### Ejercicio 23.

Construir un editor de texto, con una casilla de texto y tres botones debajo: Negrita, Cursiva y Subrayado. Al pulsar sobre ellos, se colocara el texto en el formato apropiado.



Esta primera parte debes hacerla con formatos HTML y en una segunda versión con formato específico Android. (Ejercicio5)

### Ejercicio 24.



### Ejercicio 25.

Realizar un formulario con controles personalizados:

IZALOS. Enseñanza Concertada  
Gregorio Fernández

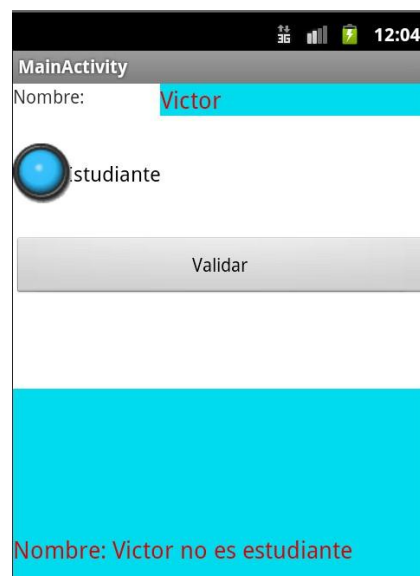




Soltero/Casado:

Nombre:

CheckBox que al ser activado, en el estado pressed, aparecerá una imagen, otra cuando está desactivado. El color de todos los textos serán rojos. Las casillas del formulario tendrán fondo azul.



### Ejercicio 26.

Crea una aplicación con un menú con dos opciones, Tablas y Salir. El primero desplegará 5 opciones de menú: 1, 2, 3, 4, 5. El segundo finalizará la aplicación. Se mostrará pulsando en cada número en un ListView la tabla de multiplicar de dicho número, teniendo en cuenta que se hará por cada línea un Layout personalizado. La opción de Salir mostrará una alerta de "Esta seguro de que desea finalizar:" Sí, No.



Solución: Ejercicio10\_menusyly

### *Ejercicio 27.*

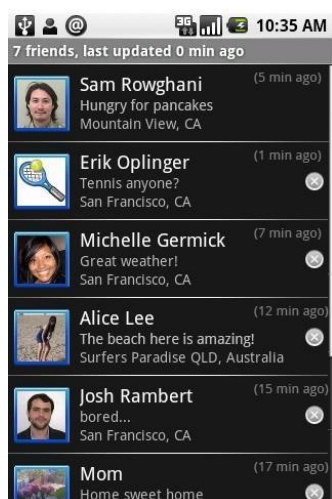
Muestra una lista de datos de tres formas distintas dependiendo de la pestaña que esté seleccionada. Añade un menú contextual sobre “Advanced Task Manager Lite” para que pueda cerrarse la aplicación.



### *Ejercicio 28.*

Generar una aplicación para hacer un ahorcado.

*gf*  
Centro de Enseñanza Concertada  
Gregorio Fernández

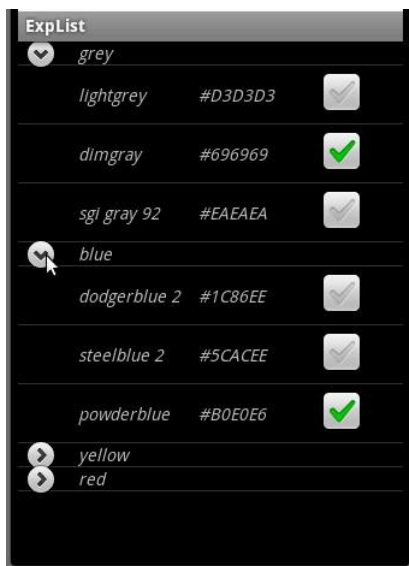


<http://www.cnblogs.com/liongname/articles/2345087.html>

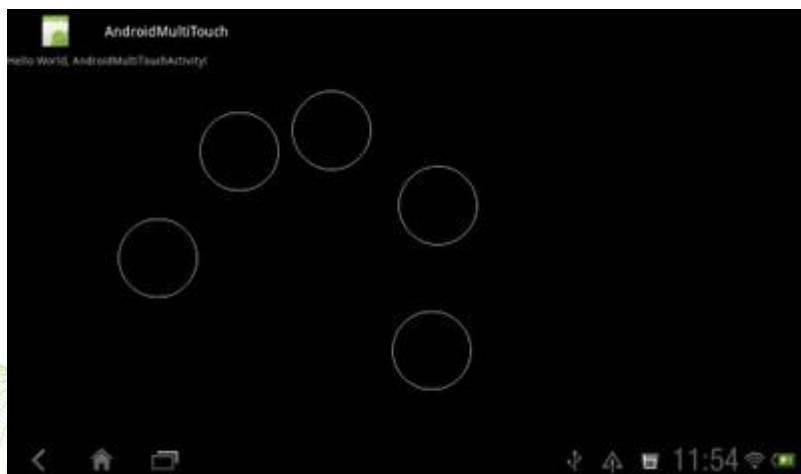
gf

Centro de Enseñanza Concertada  
Gregorio Fernández





<http://mylifewithandroid.blogspot.com.es/2010/12/expandable-list-and-checkboxes.html>



<http://android-er.blogspot.com.es/2011/12/detect-multi-touch-for-10-pointers.html>

Centro de Enseñanza Concertada  
Gregorio Fernández