

# Tareas Android Básico Noviembre 25

Se obtendrá un 0 por los siguiente motivos:

- La app no compila y ejecuta de primeras (sí se permiten crasheos durante su ejecución).
- Plugins de IA del IDE activados
- Usar cualquier método de comunicación con el exterior, incluidos sistemas de mensajería y sincronización de archivos (dropbox, drive, github)
- No registrar tu examen a través de OBS o similar, o registrarlo con CORTES. El archivo generado se dejará en el escritorio con el siguiente formato *PMDM\_nombre\_apellidos*

<b>Toolbar (0.5 punto)</b> .....	<b>1</b>
<b>Actividad Ajustes (2 puntos)</b> .....	<b>2</b>
<b>MainActivity (1.75 puntos)</b> .....	<b>3</b>
<b>TabLayout+ViewPager2 (0.75 punto)</b> .....	<b>4</b>
<b>Creación de los fragmentos (3.75 puntos)</b> .....	<b>4</b>
<b>Creación de layout para Landscape (1 punto)</b> .....	<b>5</b>
<b>Internacionalización (0'25 puntos)</b> .....	<b>5</b>

Bloque A (2.5 puntos)[[min. 1 punto](#)]: Toolbar+Actividad Ajustes

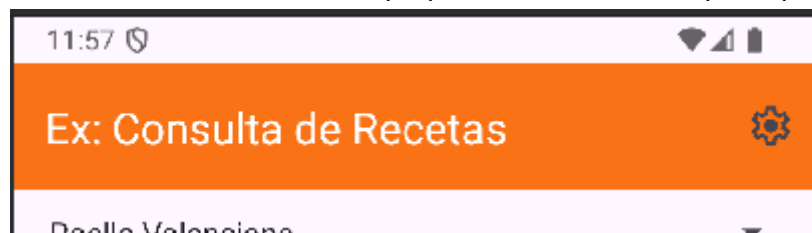
Bloque B (3 puntos) [[min. 1 punto](#)]: MainActivity + layout Landscape + internacionalización

Bloque C (4.5 puntos)[[min. 1.5 punto](#)]: TabLayout + ViewPager2 + Fragmentos

## Toolbar (0.5 punto)

Incorpora una toolbar en *activity\_main.xml* y crea un menú que solo tendrá un ítem “Ajustes”.

En MainActivity sobrescribe los métodos necesarios y configura esa Toolbar para que actúe como barra de acciones. Modifica las propiedades necesarias para que se vea así:



Al pulsar el ítem de ajustes se lanzará la actividad *AjustesActivity* (ya creada)

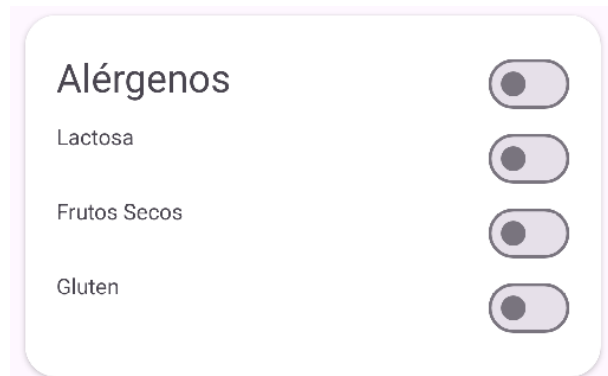
No se permiten posiciones relativas en el layout

*¿Qué hacer en caso de no saber integrar la Toolbar? Sustituye la Toolbar por un botón que realizará la misma funcionalidad que el botón de ajustes de la Toolbar.*

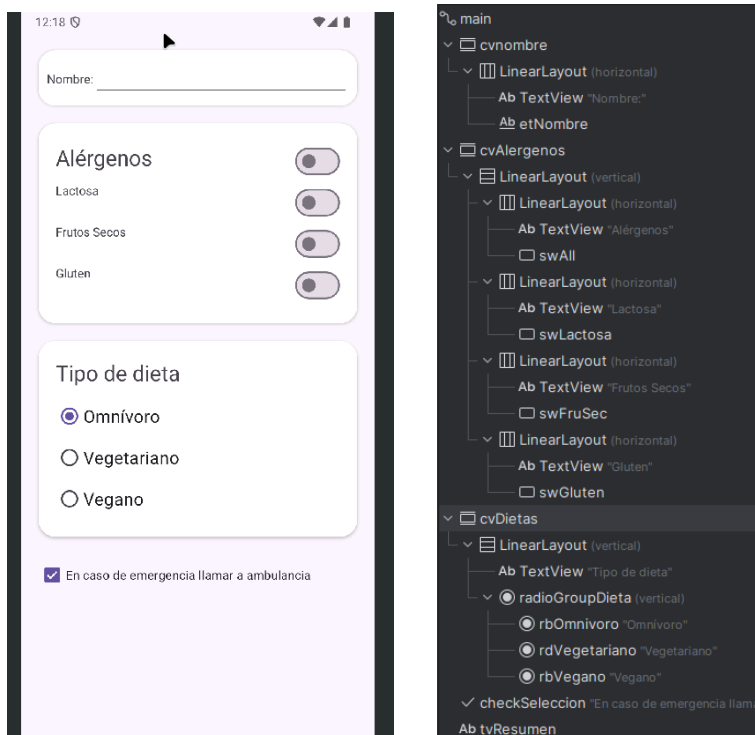
# Actividad Ajustes (2 puntos)

Se proporciona *activity\_ajustes.xml* y *AjustesActivity.kt*. Se requiere:

- MaterialSwitch con id swAll tendrá un oyente de tipo **onClick**. Este oyente llamará el método marcarTodos()
- Termina de implementar marcarTodos() para que se marquen/desmarquen todos los MaterialSwitch a la vez.



- El resto de switch tendrán un oyente de tipo OnCheckedChangeListener. Todos llamarán al método verificarSwitches(). Este método verifica si todos los switches están activados. Si todos están activados, activa el switch maestro ('swAll'). De lo contrario, lo desactiva.
- Implementar generarResumen(). Actualiza tvResumen con la información marcada. Se facilita el string de las alergias seleccionadas. Este método devolverá un String con el texto a mostrar. Por tanto, cada vez que se actualice algún componente se actualizará tvResumen.



## MainActivity (1.75 puntos)

Incorpora un Spinner entre la Toolbar anterior y El cardView(cardReceta) El spinner tendrá un oyente cuando se selecciona un elemento y lanzará el método bindearDatos que recibe la posición del spinner seleccionada (corresponde a la posición del array de las recetas).

Carga toda la información de la receta seleccionada:

- Cargará la imagen "Award Meal" renombrada como placeholder  
[https://fonts.google.com/icons?selected=Material+Symbols+Outlined:award\\_meal:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=plate&icon.size=24&icon.color=%231f1f1f](https://fonts.google.com/icons?selected=Material+Symbols+Outlined:award_meal:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=plate&icon.size=24&icon.color=%231f1f1f)
- Cargará nombre de la receta, tiempo, raciones.

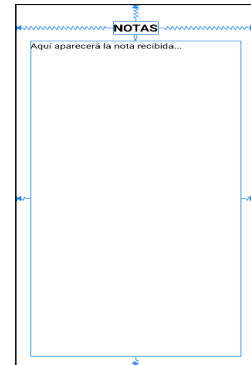


## Pasar Datos a Actividad NotasActivity (0.75 punto)

Asigna un oyente al botón para ver las notas en `bindearNotas`. Este botón lanzará una actividad que recibe esas notas como Extra para mostrarlas.

Crea una nueva actividad llamada `NotasActivity`. Cuyo layout es `activity_notas`.

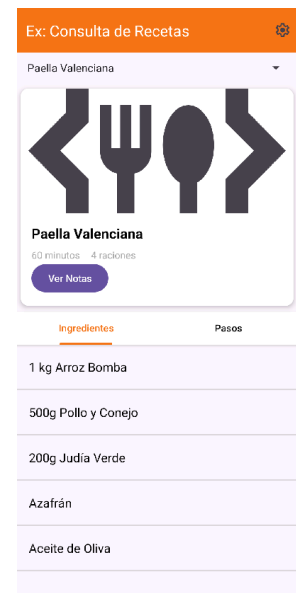
Recibe el extra con las notas desde `bindearDatos` y añadelas al elemento correspondiente.



Es OBLIGATORIO el uso de `ConstraintLayout`. No se permite ningún otro Layout Manager.

## TabLayout+ViewPager2 (0.75 punto)

Incorpora con las constraints necesarias un `TabLayout` y un `ViewPager2`. El `tabLayout` debe de estar en la parte inferior del detalle de las recetas.



## Creación de los fragmentos (3.75 puntos)

Crea dos fragmentos y todo lo necesario para que cada fragmento muestre la información.

- `IngredientesFragment` muestra la lista de ingredientes
- `PasosFragment` muestra la lista de pasos a seguir para hacer la receta
- El adaptador inicializará vacío el array (`mutableListOf<String>()`) y tendrá un método para actualizar el `Recycler View`.
- Cada fragmento también tendrá un método para actualizar la lista. Este método solo llamará a su semejante del adaptador. (`MutableList<String>`)
- Por último, modifica `bindearDatos` del `MainActivity` para utilizar el método anterior y enviarle la lista de items a cada fragmento.

`viewPagerAdapt.getInredientesFragment().setInredientes(mockRecipes[item].ingredients)`  
El fragmento recibirá el array y llamará automáticamente a un método del adaptador para actualizar la lista.

**\*\*desde la actividad obtenemos el objeto del fragmento desde el adaptador y se lo indicamos al fragmento, que a su vez informa al adaptador.**

**Nota:**podéis utilizar el mismo adaptador, viewholder y fila\_detalle para ambos fragmentos.

## Creación de layout para Landscape (1 punto)

Crea un nuevo layout para la versión landscape del activity\_main (mismos elementos, distinta distribución).

Se tiene que ver cómo continúa:



## Internacionalización (0'25 puntos)

Elimina todo el texto hardcoreado de los layouts y crea una versión en Inglés de los siguientes elementos.

- minutos
- raciones
- ingredientes
- pasos