



# HLC

2° DAW

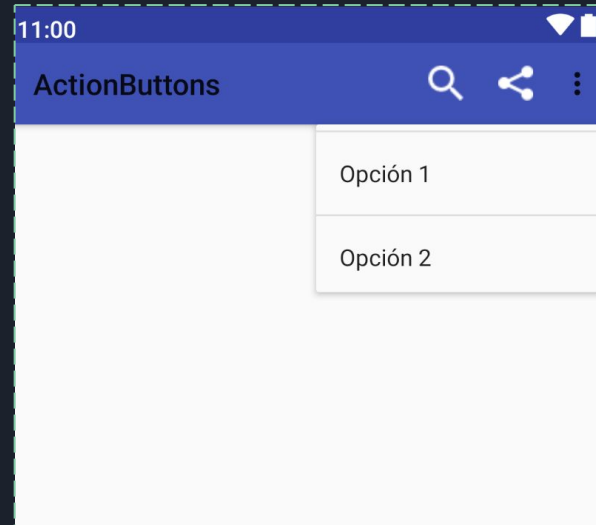
# MENÚS E ICONOS EN ACTION BAR

## CONTENIDOS DE LA UNIDAD

ACTION BAR

MENU OVERFLOW EN ACTION BAR

ACTION BUTTONS





# MENÚS E ICONOS EN ACTION BAR

## ACTION BAR - BARRA DE ACCIÓN

La **barra de acción** o también conocida como **Action Bar**, es uno de los elementos de diseño más importantes de una aplicación, ya que proporciona una **estructura visual** y **elementos interactivos** a nuestra activity.

Entre los elementos interactivos que se pueden agregar a la barra de acción se encuentran:

1. Botones de acción
2. Menú OverFlow

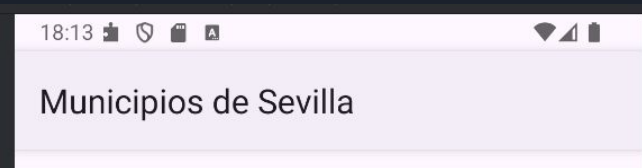
# MENÚS E ICONOS EN ACTION BAR

## ¿CÓMO HACER QUE APAREZCA EL ACTION BAR?

Nos dirigimos al archivo `res/values/themes.xml`.  
Modificamos nuestro tema para que permita Action Bar

Por ello, en esta línea debemos eliminar el último punto para que quede:

```
<style name="Base.Theme.PruebasLayout" parent="Theme.Material3.DayNight">
```





# MENÚS E ICONOS EN ACTION BAR

## CAMBIAR EL COLOR DE LA BARRA

Si queremos cambiar el color de dicha barra, agregamos este código en el onCreate()

```
if (getSupportActionBar() != null) {  
    getSupportActionBar().setBackgroundDrawable(new ColorDrawable(Color.parseColor("#FF6200EE")));  
}
```



# MENÚS E ICONOS EN ACTION BAR

## AÑADIR EL ICONO DE LA APP A LA BARRA

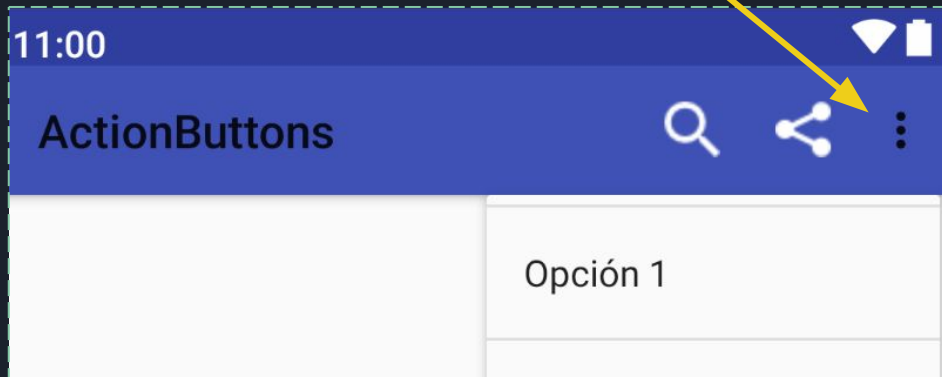
Si queremos agregar el icono, agregamos este código al if anterior

```
if (getSupportActionBar() != null) {  
    ...  
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
    getSupportActionBar().setIcon(R.mipmap.ic_launcher);  
}
```

# MENÚS E ICONOS EN ACTION BAR

## MENÚ OVERFLOW

El **menú OverFlow**, es un menú desplegable que **muestra acciones u opciones** disponibles dentro de nuestra aplicación, pero que no tienen espacio para aparecer dentro del **actionbar**, o bien, no son lo suficientemente importantes para mostrarse en primera instancia.

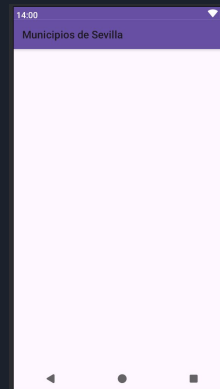
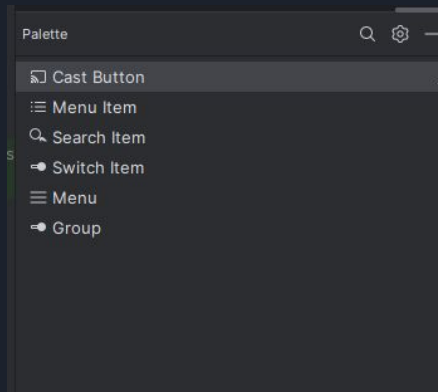


# MENÚS E ICONOS EN ACTION BAR

## MENÚ OVERFLOW

Para ponerlo en práctica

- Crearemos un nuevo archivo haciendo click derecho en **app>res> new Android Resource File**
- En File name: overflow (*da igual el que le pongamos, pero debe ser en minúsculas*)
- Resource type: Menu
- Se nos ha creado **menu>overflow.xml**







# MENÚS E ICONOS EN ACTION BAR

## MENÚ OVERFLOW

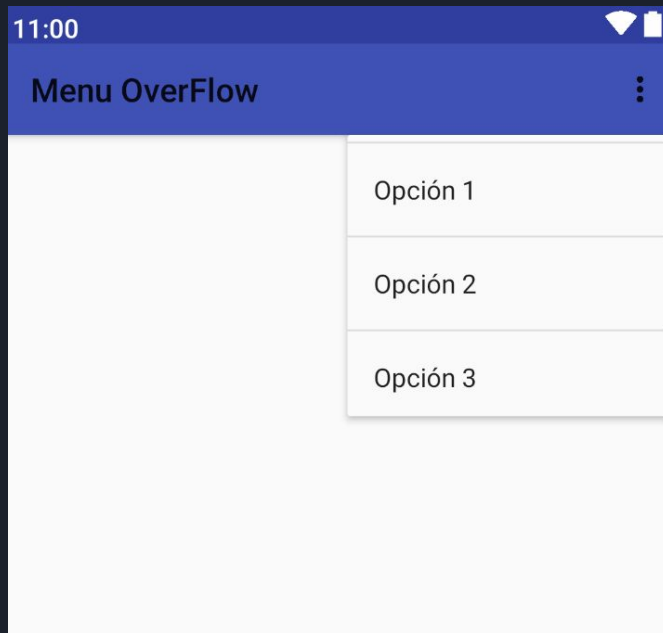
- Nos aparece una nueva paleta de componentes
- Dentro de **overflow** arrastramos dentro del actionBar un **Menu Item**
- Arrastramos dos más hasta tener tres ítems.

# MENÚS E ICONOS EN ACTION BAR

## MENÚ OVERFLOW

Para ponerlo en práctica

- Añadimos las **ids**:
  - item1
  - item2
  - item3
- En nuestro archivo **strings.xml** añadimos:
  - `<string name="item1">Opción 1</string>`
  - `<string name="item2">Opción 2</string>`
  - `<string name="item3">Opción 3</string>`
- Y ponemos las referencias en **title**





# MENÚS E ICONOS EN ACTION BAR

## MENÚ OVERFLOW

Vamos a la parte lógica. Programaremos un método para mostrar y ocultar el menú:

```
//Método para mostrar y ocultar el menú
@Override
public boolean onCreateOptionsMenu(Menu menu){
    getMenuInflater().inflate(R.menu.overflow, menu);
    return true;
}
```



# MENÚS E ICONOS EN ACTION BAR

## MENÚ OVERFLOW

Y ahora un método para asignar funciones a cada uno de los elementos del menú:

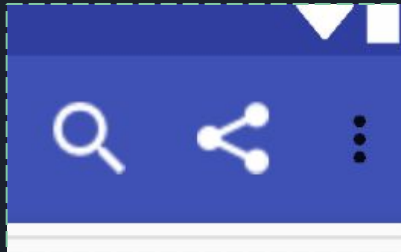
```
//Método para asignar las funciones correspondientes a las opciones.  
@Override  
public boolean onOptionsItemSelected(MenuItem item){  
    int id = item.getItemId();  
  
    if(id == R.id.item1){  
        Toast.makeText(this, "Opción 1", Toast.LENGTH_SHORT).show();  
    } else if(id == R.id.item2){  
        Toast.makeText(this, "Opción 2", Toast.LENGTH_SHORT).show();  
    } else if(id == R.id.item3){  
        Toast.makeText(this, "Opción 3", Toast.LENGTH_SHORT).show();  
    }  
    return super.onOptionsItemSelected(item);  
}
```

# MENÚS E ICONOS EN ACTION BAR

## BOTONES DE ACCIÓN

Los botones de acción o **action buttons**, aparecen directamente en el **ActionBar** como un **icono** y/o texto.

Al igual que el menú **overFlow**, los botones de acción se utilizan en las acciones que no tienen espacio para aparecer dentro del **ActionBar**, o simplemente no son lo suficientemente importantes.



# MENÚS E ICONOS EN ACTION BAR

## BOTONES DE ACCIÓN

Continuaremos el proyecto anterior añadiendo un par de botones, los cuales usarán iconos.

- En el actionBar arrastramos ahora dos **Menu Item** más, con **ids**:
- Y con **icon** que queramos
- En **showAsAction** establecemos: always

