

## NOTIFICACIONES CON CUADROS DE DIÁLOGO

### Cuadros de diálogo básicos en Android

- Usamos un Builder para crear el AlertDialog:

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

- Introducimos los parámetros del dialogo

```
builder.setMessage("Esto es un mensaje de alerta.")  
    .setTitle("Información")  
    .setPositiveButton("OK", (dialog, id) -> {dialog.cancel();});
```

- Construimos el AlertDialog a partir del Builder

```
AlertDialog dialogo= builder.create();
```

- Una vez creado, lo podemos mostrar:

```
dialogo.show();
```

### Cuadros de diálogo con listas en Android

- Creamos una lista en el código,

```
String[] items = {"Español", "Inglés", "Francés"};
```

- o bien la extraemos de un archivo de recursos
  - Creamos el recurso dentro de res/arrays (o cualquier otro nombre) con values resources file

```
<resources>  
    <string-array name="profesores">  
        <item> Enrique</item>  
        <item> Pedro </item>  
        <item>M José</item>  
        <item>Adolfo </item>  
        <item>Clara </item>  
    </string-array>  
</resources>
```

- Accedemos a ella mediante:

```
String[] items=getResources().getStringArray(R.array.profesores);
```

- Cambiamos setMessage del builder por método para acceder a listas, en este caso con RadioButtons:

```
builder.setSingleChoiceItems(items, -1, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialogo, int item) {

        //gestionamos el evento, se le pueden pasar objetos externos
        profesor.setNombre(items[item]);

    }}
)
```

También la gestión anónima puede ser función lambda

```
builder.setSingleChoiceItems(items, -1, (dialogo, item) -> {
    valor[0] =item;
    profesor.setNombre(items[item]);})
```

- El resto de elementos del builder se mantienen

```
builder
.setTitle("Información")
.setPositiveButton("OK", (dialogo, id) -> {

    //código ;

});
```

## Cuadros de diálogo personalizados

- Creamos un LinearLayout en el archivo res/layout con elementos de la vista. Ejemplo:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/profesor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="Nombre"/>
</LinearLayout>
```

- Creamos la vista desde el código:

```
LayoutInflater inflater = getLayoutInflater();  
View view=inflater.inflate(R.layout.activity_formulario_add_profesor,null);
```

- Lo incorporamos en el builder.

```
builder  
    .setView(view)  
    .setTitle("Información")  
    .setPositiveButton("OK",(dialog,id)->{  
        // findViewById () se usa de la view creada  
        EditText editText=(EditText)(view.findViewById(R.id.profesor));  
        Editable texto=editText.getText();  
        //profesor es un objeto previo que hemos utilizado para guardar la información  
        profesor.setNombre(texto.toString());  
        validar(profesor);  
    })  
    .setNegativeButton("Cancelar",(dialog,id)->{dialog.cancel();});
```