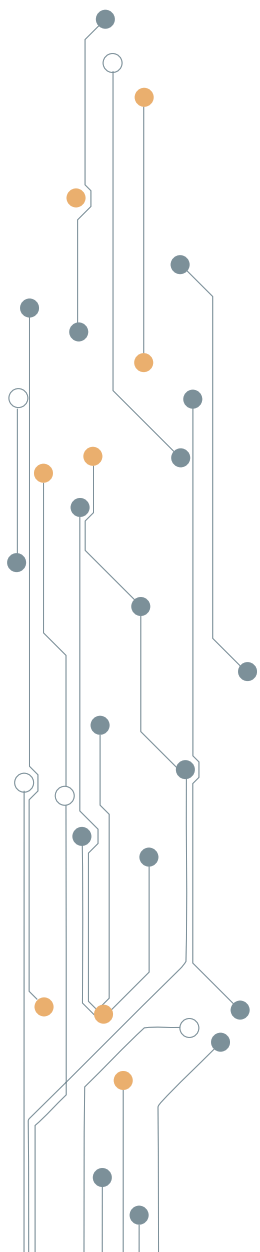




# Alertas y cuadros diálogo

# Índice



## Alertas y cuadros de diálogo

1   Alertas y cuadros de diálogo	3
1.1   Alertas	3
1.2   Cuadros de diálogo	5

# 1. Alertas y cuadros de diálogo

Además de utilizar componentes gráficos para presentar información al usuario, en muchas ocasiones resulta útil la utilización de cuadros de diálogo para informar de alguna situación al usuario, incluso para preguntarle por alguna decisión que debe de tomar.

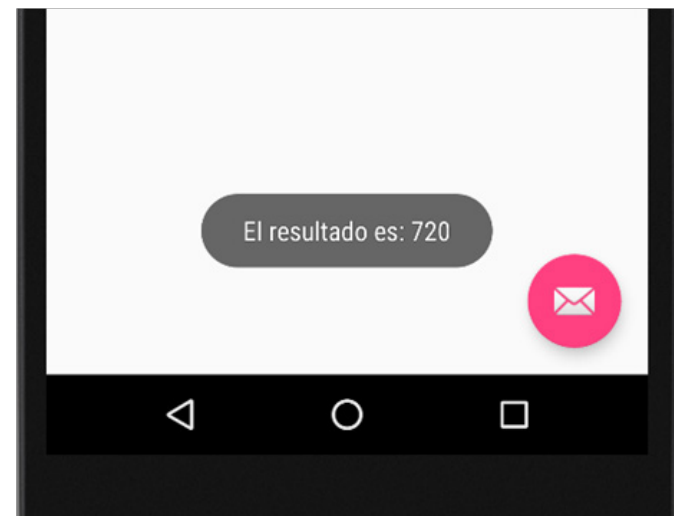
Para este tipo de acciones, Android nos proporciona dos elementos las alertas y los cuadros de diálogo.

## 1.1 | Alertas

Una alerta es un mensaje que aparece en la pantalla del terminal para informar al usuario de alguna situación y que desaparece automáticamente al cabo de unos segundos.

Para generar una alerta, Android dispone de la clase Toast del paquete android.widget. Esta clase dispone del siguiente método estático con el que podremos crear un objeto Toast:

```
makeText(Context context, CharSequence text, int duration)
```

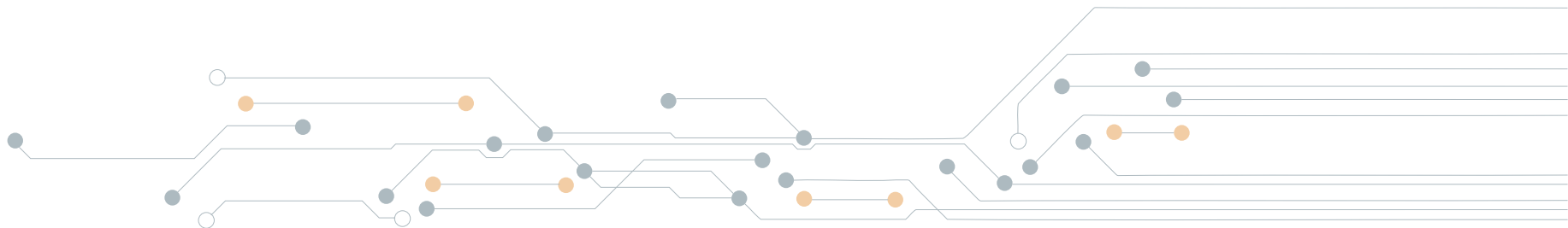


El primer parámetro representa el objeto Context. Ya hemos visto el uso de este objeto durante la creación de Intents. El segundo parámetro representa el texto que queremos mostrar en la alerta, mientras que el tercer y último parámetro es la duración de la alerta.

Una vez creado el objeto Toast, la llamada a su método show() provocará que se muestre la alerta en pantalla.

Por ejemplo, si en el ejercicio ejemplo02 queremos presentar también el resultado del factorial mediante una alerta, añadiríamos la siguiente instrucción al final del método calcular():

```
Toast.makeText(this,"El resultado es "+res, Toast.LENGTH_LONG).show()
```



## 1.2 | Cuadros de diálogo

Cuando además de mostrar un mensaje necesitamos pedir al usuario la toma de una decisión, utilizaremos un cuadro de diálogo con botones para la interacción con el mismo.

La gestión de cuadros de diálogo en Android se lleva a cabo a través de la clase `AlertDialog.Builder` del paquete `android.app`. Dicha clase dispone de un constructor para crear el objeto a partir del contexto

**`AlertDialog.Builder(Context context)`**

Una vez creado el objeto, la configuración del cuadro de diálogo que vamos a generar se realiza a través de una serie de métodos de `AlertDialog.Builder`, entre los que destacamos:

- **`setMessage(CharSequence texto)`**. Establece el texto de aviso que se mostrará en el cuadro de diálogo.
- **`setTitle(CharSequence titulo)`**. Establece el texto de la barra de título del cuadro de diálogo
- **`setPositiveButton(int id, DialogInterface.OnClickListener listener)`**. Establece las propiedades del botón de respuesta positiva (aceptar, si, etc.). El primer parámetro representa el identificador de recurso `String` asociado al texto del botón, identificadores que se encuentran en la clase pública `android.R.string`, como por ejemplo `android.R.string.yes` que establece el texto de respuesta afirmativa, o `android.R.string.ok` que establece el texto "ok" o "aceptar", según el idioma. El segundo parámetro es el **objeto escuchador** donde se define el método a ejecutar tras la pulsación del botón. Si no se quiere ejecutar ninguna acción específica, este segundo parámetro se establecerá a `null`. Existe otra versión de este método en la que el primer parámetro es directamente el texto que se quiere mostrar en el botón.



- **setNegativeButton(int id, DialogInterface.OnClickListener listener).** Tiene la misma función que el anterior, pero para el botón de respuesta negativa (no, cancelar, etc.).
- **setIcon(int iconId).** Permite definir un icono para el cuadro de diálogo. Como parámetro se proporciona el identificador del icono. Los iconos son archivos de imagen localizados en la carpeta `drawable` de la sección de recursos del proyecto. Cuando se dispone de un archivo de imagen en esta carpeta, se crea automáticamente un identificador para esa imagen en la clase **R.drawable** con el nombre del archivo.

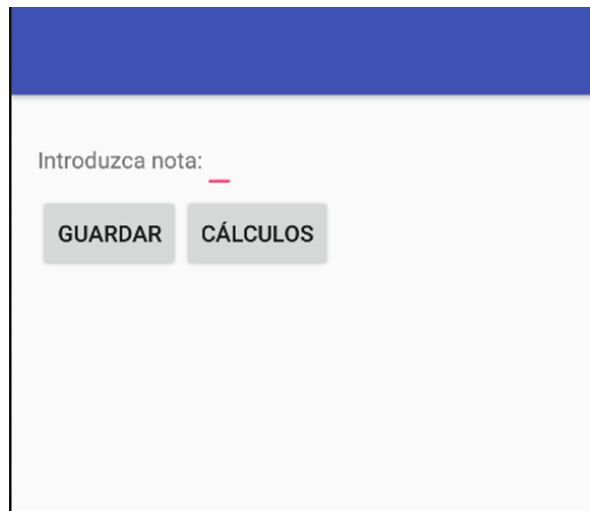
Una vez configurado el objeto `AlertDialog.Builder`, la llamada al método `show()` del mismo provoca la creación del cuadro de diálogo y que se muestre en pantalla.



## Ejercicio de ejemplo

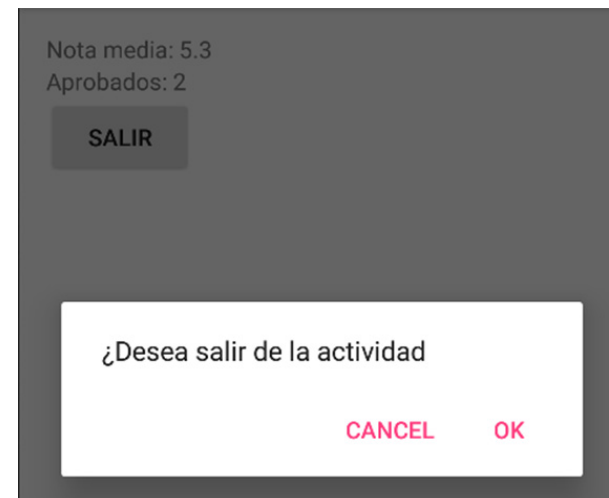
Vamos a crear un nuevo ejercicio de ejemplo en el que pondremos en práctica, no solo la creación de cuadros de diálogo, sino también de muchos de los conceptos explicados hasta ahora.

La aplicación es un gestor de notas, que permitiría a un profesor introducir notas en la aplicación y luego hacer cálculos sobre las mismas, como la obtención de la nota media y del número de aprobados:

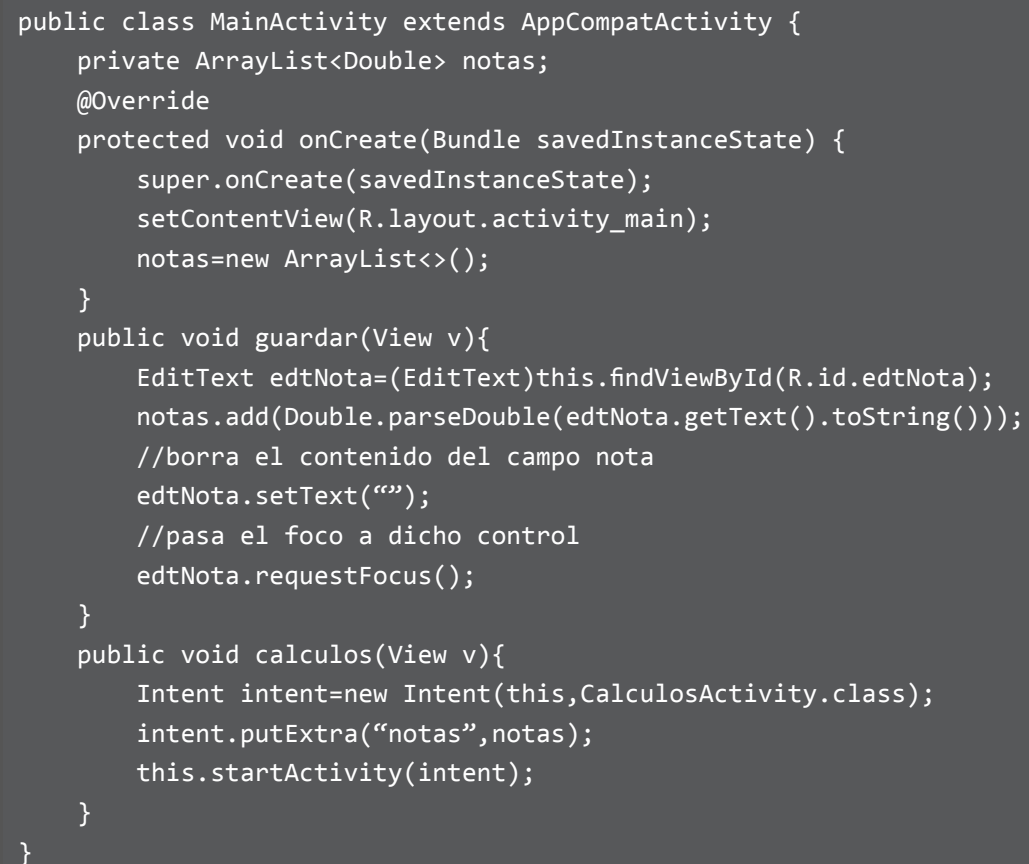


Como aún no hemos visto el tema del almacenamiento permanente en Android, las notas se quedarían en memoria, eliminándose al finalizar la aplicación.

En la actividad que muestra los cálculos vemos un botón para salir. Ahí será donde hagamos uso del cuadro de diálogo, pues se preguntará al usuario si está seguro de que desea salir de la actividad:



Sobre la forma de colocar los controles en la actividad, ya lo hemos abordado en otros ejercicios, por lo que nos centraremos en el código de las actividades, aquí tenemos el de la actividad principal:



```
public class MainActivity extends AppCompatActivity {
    private ArrayList<Double> notas;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        notas=new ArrayList<>();
    }
    public void guardar(View v){
        EditText edtNota=(EditText)this.findViewById(R.id.edtNota);
        notas.add(Double.parseDouble(edtNota.getText().toString()));
        //borra el contenido del campo nota
        edtNota.setText("");
        //pasa el foco a dicho control
        edtNota.requestFocus();
    }
    public void calculos(View v){
        Intent intent=new Intent(this,CalculosActivity.class);
        intent.putExtra("notas",notas);
        this.startActivity(intent);
    }
}
```



Como podemos observar, las notas se guardan en un ArrayList de objetos double. Dado que ArrayList es serializable, las notas se pueden almacenar en un Intent para pasárselas a la actividad que realiza los cálculos.

Además de guardar las notas en la colección, el botón guardar se encarga de limpiar la caja de texto y **pasarle el foco** a la misma a través del método ***requestFocus()***, de manera que el usuario pueda seguir introduciendo notas.

Vemos que los cálculos son realizados por dos métodos adicionales, *media()* y *aprobados()*, que son llamados desde el *onCreate* de la actividad para rellenar el contenido de los TextView con los resultados.

Por otro lado, la gestión de la pulsación del botón de salir genera un cuadro de diálogo en el que se pregunta al usuario si desea abandonar la actividad. La respuesta al botón de afirmación se realiza mediante un objeto de una clase anónima que implementa OnClickListener.

```
public class CalculosActivity extends
    AppCompatActivity {
    @Override
    protected void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_
            calculos);
        Intent intent=this.getIntent();
        ArrayList<Double>
        notas=(ArrayList<Double>)intent.
            getSerializableExtra("notas");
        TextView tvMedia=(TextView)this.
            findViewById(R.id.tvMedia);
        TextView tvAprobados=(TextView)this.
            findViewById(R.id.tvAprobados);
        //llama a los métodos que hacen los
        cálculos y muestra
        //los resultados en los campos de texto
        tvMedia.setText("Nota media:
            "+media(notas));
        tvAprobados.setText("Aprobados:
            "+aprobados(notas));
    }
    private double media(ArrayList<Double>
        notas){
        double result=0;
        for(Double n:notas){
            result+=n;
        }
        return result/notas.size();
    }
}
```

```
private int aprobados(ArrayList<Double>
    notas){
    int total=0;
    for(Double n:notas){
        if(n>=5){
            total++;
        }
    }
    return total;
}
public void salir(View view){
    AlertDialog.Builder cuadro=new
        AlertDialog.Builder(this);
    cuadro.setMessage("¿Desea salir de la
        actividad?");
    //el escuchador del botón de afirmación
    se define como
    //una clase anónima
    cuadro.
        setPositiveButton(android.R.string.yes,
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
                    dialog, int which) {
                    //Abandona la actividad
                    CalculosActivity.this.finish();
                }
            });
    //no se define ningún escuchador para
    el botón de negación
    cuadro.
        setNegativeButton(android.R.string.no,null);
    cuadro.show();
}
```

*Telefonica*

---

EDUCACIÓN DIGITAL