

Tecnologías y Desarrollo en Dispositivos Móviles

Apartado 7: Log, Toast y Dialogs

Autores:

Víctor M. Rivas Santos
(Antonio Rueda Ruiz)

La clase Log

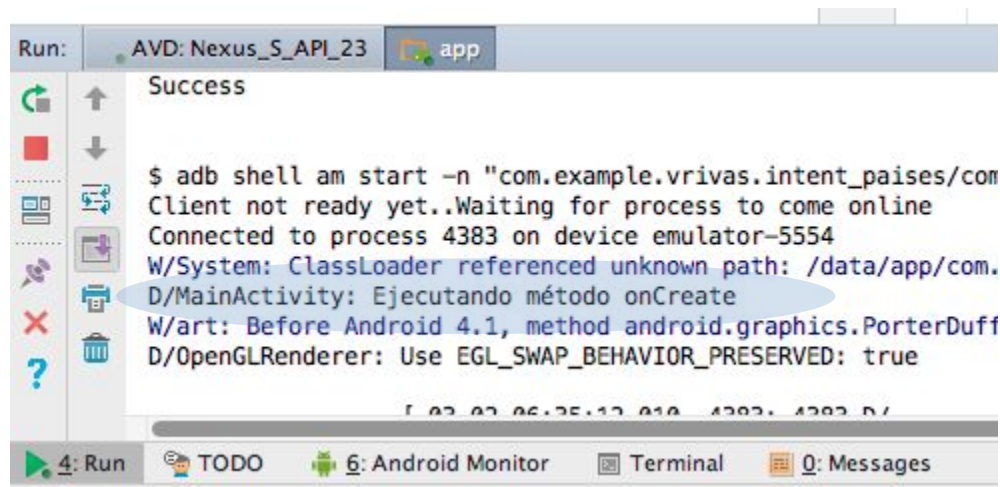
- Permite mostrar en el *LogCat* (dentro de la pestaña Android Monitor) mensajes para depuración.
- El formato es:

`Log.método("Etiqueta", "Mensaje");`

`private static final String LOG_TAG = "MyActivity";`

`@Override`

`protected void onCreate(Bundle savedInstanceState) {
 Log.d(LOG_TAG, "Ejecutando método onCreate");`



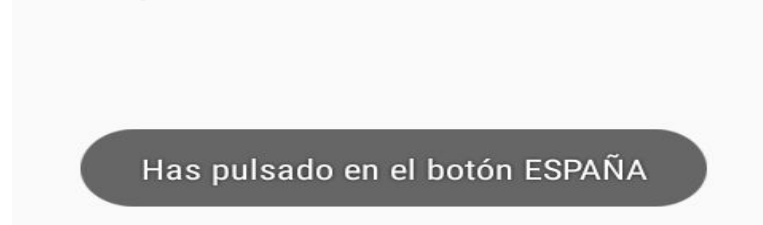
Métodos de la clase Log

- Log.**e**: Cuando comunicamos que un error ha ocurrido y que, en principio, podía ser esperable
- Log.**wtf**: Cuando comunicamos un error que pensábamos que nunca se produciría (*What a Terrible Failure*)
- Log.**w**: Se usa cuando detectamos una incidencia que, sin ser un error, quizá no debería haber ocurrido (esto es, un *warning*)
- Log.**i**: Se utiliza cuando deseamos comunicar que algo se ha llevado a cabo de forma satisfactoria.
- Log.**d**: Son los típicos mensajes que sirven de “marcas” en nuestro código y nos permiten saber hasta dónde se está ejecutando y/o qué secuencia de ejecución va siguiendo la app.
- Log.**v**: Si realmente deseamos poner muchos mensajes en la consola, haciendo que tenga mucha verborrea acerca de lo que va haciendo nuestro programa.

La clase Toast

- Pequeños mensajes de corta duración en pantalla

```
Toast.makeText(this, "Has pulsado en el botón ESPAÑA", Toast.LENGTH_SHORT).show();
```

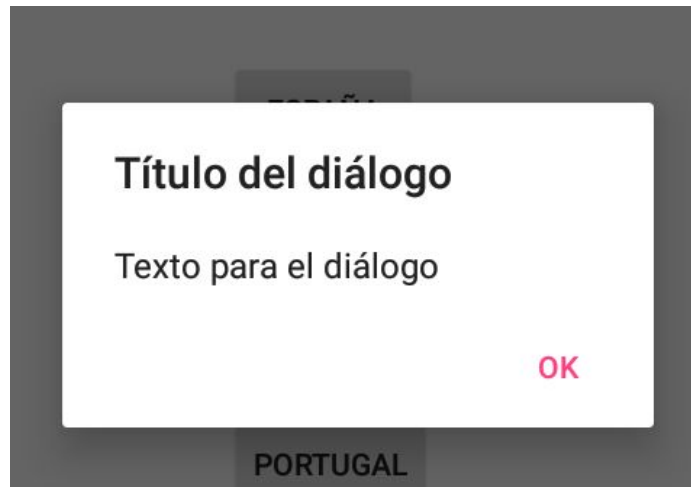


```
Toast.makeText(this, R.string.text_toast, Toast.LENGTH_LONG).show();
```



Dialogs: AlertDialog

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Título del diálogo"); // Mejor usar R.string
builder.setMessage(R.string.text_dialog);
builder.setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which){
        dialog.dismiss();
    }
});
builder.show();
```



Dialogs: TimePicker

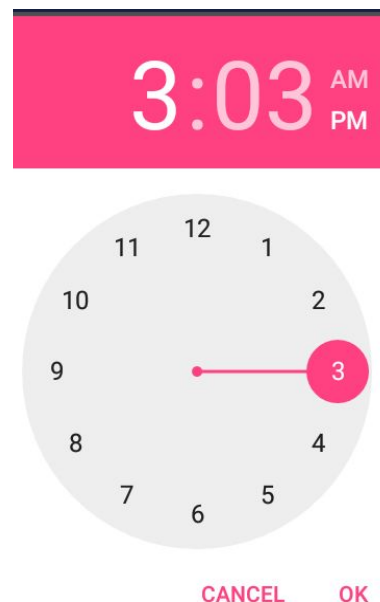
```
public static class TimePickerFragment
    extends DialogFragment
    implements TimePickerDialog.OnTimeSetListener {
```

@Override

```
public Dialog onCreateDialog(Bundle savedInstanceState) {
    // Use the current time as the default values for the picker
    final Calendar c = Calendar.getInstance();
    int hour = c.get(Calendar.HOUR_OF_DAY);
    int minute = c.get(Calendar.MINUTE);

    // Create a new instance of TimePickerDialog and return it
    return new TimePickerDialog(getActivity(), this, hour, minute,
        DateFormat.is24HourFormat(getActivity()));
}
```

```
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
    // Método que se ejecuta al fijar el usuario la hora y el minuto
    Toast.makeText(getContext(), hourOfDay+":"+minute, Toast.LENGTH_SHORT).show();
}
// ...
// Para usarlo:
DialogFragment t=new TimePickerFragment();
t.show(this.getFragmentManager(), "selector_hora");
```



Dialogs: DatePicker

```
public static class DatePickerFragment extends DialogFragment
implements DatePickerDialog.OnDateSetListener {
```

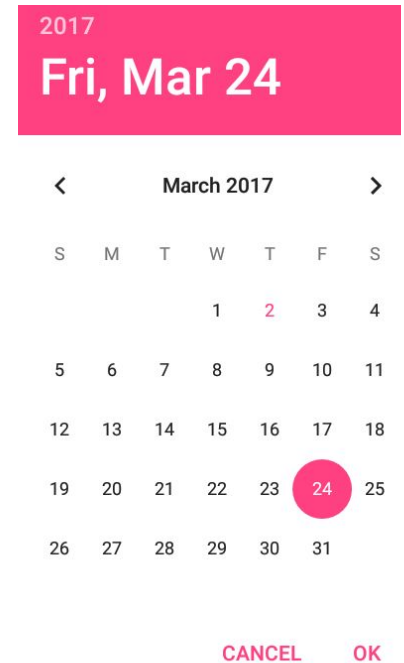
@Override

```
public Dialog onCreateDialog(Bundle savedInstanceState) {
    // Use the current time as the default values for the picker
    final Calendar c = Calendar.getInstance();
    // Create a new instance of TimePickerDialog and return it
    return new DatePickerDialog( getActivity()
        , this
        , c.get(Calendar.YEAR)
        , c.get(Calendar.MONTH)
        , c.get(Calendar.DAY_OF_MONTH) );
}
```

```
public void onDateSet(DatePicker view
    , int year
    , int month
    , int day_of_month) {
    Toast.makeText( getContext()
        , day_of_month+"/"+(month+1)+"/"+year // ¡OjO: month+1!!
        , Toast.LENGTH_SHORT).show();
}
```

// Para poder usarlo

```
DialogFragment d=new DatePickerFragment();
d.show(this.getFragmentManager(), "selector_fecha");
```



Fecha y hora de la cita

MTV: Utilización de las clases Toast, Dialog y Log

- Realiza una pequeña app en la que se pregunte al usuario por la fecha y hora de su próxima cita con el dentista
- La app deberá confirmar ambos datos (fecha y hora) al usuario una vez introducidos mediante una única Toast
 - El mensaje de la Toast deberá estar disponible en 3 idiomas: español, inglés y portugués
 - El idioma se seleccionará automáticamente atendiendo a la configuración del dispositivo
- Además, se debe usar la clase Log para informar al desarrollador de los distintos pasos que va siguiendo la app para completar su funcionalidad.
- Sube la clase MainActivity.java (en la que debe estar todo el código necesario) a la plataforma.

Referencias

- <https://developer.android.com/reference/android/util/Log.html>
- <https://developer.android.com/reference/android/widget/Toast.html>
- <https://developer.android.com/reference/android/app/AlertDialog.html>
- <https://developer.android.com/reference/android/widget/DatePicker.html>
- <https://developer.android.com/reference/android/widget/TimePicker.html>