

# Laboratorio 4.- Los números muertos

---

## Contenido:

1	INCLUIR ACCIONES EN LAS NOTIFICACIONES .....	2
2	LOS NÚMEROS MUERTOS.....	3
2.1	MECÁNICA.....	3
2.2	PROTOTIPO DE LA INTERFAZ.....	4

**Objetivos:** Realizar una aplicación completa con los conocimientos adquiridos hasta ahora usando notificaciones y diálogos.

## 1 Incluir acciones en las notificaciones

Las notificaciones de Android permiten, a través de las acciones, incluir varios posibles comportamientos (hasta un máximo de 3) de manera que el usuario pueda, desde la propia notificación llevar a cabo distintas acciones.

Para añadir una acción a la notificación hay que usar el método `addAction(...)` de la clase `NotificationCompat.Builder`. Las acciones se ven distintas en función de la versión del sistema operativo ya que desde Android N no se muestra el icono, aunque es obligatorio indicarlo (ver Fig. 1 y Fig. 2).

```
elBuilder.setLargeIcon(BitmapFactory.decodeResource(getResources(), R.drawable.tarta))
    .setSmallIcon(android.R.drawable.stat_sys_warning)
    .setContentTitle("Mensaje de Alerta")
    .setContentText("Ejemplo de notificación en DAS.")
    .setSubText("Información extra")
    .setVibrate(new long[]{0, 1000, 500, 1000})
    .setAutoCancel(true)
    .addAction(android.R.drawable.ic_input_add, "Añadir", intentEnNot)
    .addAction(android.R.drawable.star_on, "Favorito", intentEnNot2)
    .addAction(android.R.drawable.sym_action_call, "Llamar", intentEnNot3);
```

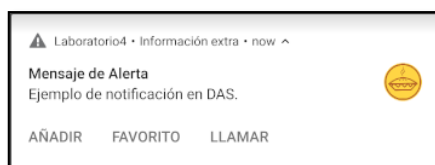


Fig. 1.-Acciones en Android P

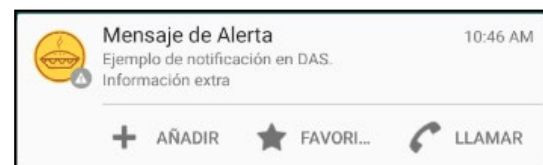


Fig. 2.- Acciones en Android L

⚠ Al pulsar una acción la notificación no desaparece y sigue apareciendo como pendiente. Para que la notificación desaparezca hay que controlar qué notificación es la que se quiere cancelar (mediante el id utilizado al crearla) y llamar al método `cancel(...)` de la clase `NotificationManager`.

Al crear y lanzar la notificación

```
Intent i = new Intent(this, otraActividad.class);
i.putExtra("id", 1);
PendingIntent intentEnNot = PendingIntent.getActivity(this, 0, i,
    PendingIntent.FLAG_UPDATE_CURRENT |
    PendingIntent.IMMUTABLE);
...
elManager.notify(1, elBuilder.build());
```

```
Bundle extras=getIntent().getExtras();
if (extras!=null){
    int id=extras.getInt("id");
    NotificationManager elManager = (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);
    elManager.cancel(id);
}
```

En la actividad lanzada por la notificación

## Desarrollo Avanzado de Software - Curso 2023 / 2024

Desde Android 13 en adelante es necesario incluir los permisos de las notificaciones en el fichero XML AndroidManifest y la solicitud de permisos en el onCreate de la actividad principal.

En el manifiesto:

```
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

En la actividad:

Codigo="Añadir un integer";

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.POST_NOTIFICATIONS) !=
        PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.POST_NOTIFICATIONS}, Codigo);
    }
}
```

## 2 Los números muertos

Los Números Muertos es un juego para dos jugadores donde el jugador 1 piensa un número con un determinado número de dígitos y el jugador 2 tiene X intentos para intentar adivinarlo.

### 2.1 Mecánica

La mecánica del juego de Los Números Muertos es muy sencilla. El jugador 1 piensa un número, el jugador 2 tiene que intentar adivinarlo.

Cada vez que el jugador 2 dice un número, el jugador 1 le dirá cuántos números muertos y cuántos heridos hay en la cifra que ha dado. Un número muerto es aquel dígito que está en la misma posición que en el número a adivinar. Un número herido es aquel dígito que aparece en el número a adivinar, pero está en una posición distinta.

**Ejemplo:** Si el número a adivinar es el **4314**

- Si el jugador 2 dice **8765**, el jugador 1 le tiene que responder: 0 muertos y 0 heridos.
  - Si el jugador 2 dice **1098**, el jugador 1 le tiene que responder: 0 muertos y 1 herido (el 1 está en el número a adivinar, pero en una posición distinta).
  - Si el jugador 2 dice **4531**, el jugador 1 le tiene que responder: 1 muerto (el 4) y dos heridos (el 3 y el 1). Cada dígito es una "instancia" distinta, por eso el 4 está sólo muerto, no puede estar muerto y herido a la vez.
  - Si el jugador 2 dice **8733**, el jugador 1 le tiene que responder que 0 muertos y 2 heridos (cada uno de los 3).
- Implementad una aplicación con todas las actividades que consideréis necesarias y donde el jugador que "piense" el número sea el propio dispositivo (usad la clase *Random*).

```
Random generador =new Random();
int numero=generador.nextInt(10000);
```

← Limite

## Desarrollo Avanzado de Software - Curso 2023 / 2024

- Ponedle varios niveles de dificultad que varíen en el número de dígitos del número a adivinar o en el número de intentos para hacerlo. Seguid el prototipo de interfaz que se describe a continuación.

### 2.2 Prototipo de la interfaz

La pantalla principal (Fig. 3) tendrá dos botones, uno que permitirá comenzar el juego y otro donde se mostrarán las instrucciones en forma de ventana de diálogo (Fig. 4).



Fig. 3.- Interfaz principal

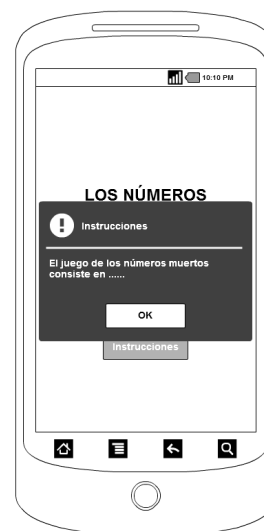


Fig. 4.- Diálogo de instrucciones

Para comenzar el juego habrá que elegir, a través de un diálogo, el nivel de dificultad con el que se quiere jugar (Fig. 5). Una vez elegido, se pasará a la pantalla de juego donde se mostrará el nivel en el que estamos jugando, cuántos intentos nos quedan y un historial de los intentos que hemos hecho durante esa partida y su resultado (número de muertos y heridos).



Fig. 5.- Selección de nivel



Fig. 6.- Cálculo del resultado

## Desarrollo Avanzado de Software - Curso 2023 / 2024

Si el número NO se ha acertado, se mostrará un toast con el resultado (Fig. 7). Si el número SÍ se ha acertado, se nos mostrará una notificación en la barra de tareas indicando el número de intentos en el que se ha conseguido (Fig. 8). Dicha notificación incluirá dos acciones, una que nos permitirá finalizar el juego y otra que nos permitirá jugar de nuevo comenzando una nueva partida.

Cuando el número de intentos se termine, se mostrará un diálogo al usuario indicándoselo (Fig. 9) y cuando pulse el botón OK se le llevará al menú inicial de la aplicación.



Fig. 7.- Mensaje de información

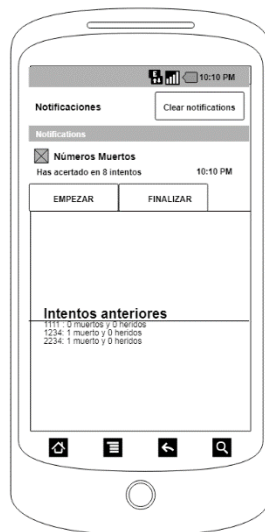


Fig. 8.-Notificación de éxito



Fig. 9.-Fin del juego