

UF1. [PAC01] Solución

Para la realización de las siguientes actividades deberéis ayudaros del libro de texto, capítulos 4, 5, 6, 11, 12, 14 y 15, y consultar, si lo creéis necesario, internet. Aunque esta PAC se base en esos capítulos, es muy recomendable leer los capítulos 1, 2 y 3 para empezar a programar para Android.

Debéis subir un único archivo comprimido que contenga **UN ÚNICO DOCUMENTO PDF** con las respuestas de la parte teórica, y el proyecto creado para la parte práctica.

Los ejercicios sobre la parte teórica tienen un valor de 0,5 puntos, y el ejercicio práctica vale 5 puntos.

Se valorará positivamente el correcto uso del nombre en los nombres de variables, métodos y clases.

Recordad que la fecha límite para la entrega de esta PAC es el 14 de OCTUBRE.

Sugerencia: Para la realización de estos ejercicios es necesario utilizar Android Studio.

Información: Cualquier PAC copiada y/o en la que se haya utilizado “copy-paste” de código ya escrito será puntuada con un 0.

Actividades

PARTE TEÓRICA

1. ¿Qué es una actividad?

Una actividad es un componente de la aplicación Android que proporciona interacción con el usuario. Se puede interpretar como una ventana tradicional en la que se desarrolla una acción simple.

2. ¿Cuál es el evento obligatorio de todas las actividades? ¿Cuál es su funcionalidad?

onCreate(). Sirve para inicializar los componentes gráficos.

3. ¿Cuáles son los pasos que debemos seguir para añadir una nueva actividad a nuestra aplicación?

En primer lugar extender de la clase Activity, e implementar al menos el evento onCreate(). Definir la actividad en el fichero de manifiesto y por último lanzar la actividad, bien por parte del Sistema operativo, o por nosotros mediante la llamada a los métodos startActivity() o startActivityForResult().

4. Explicar el ciclo de vida de una actividad con sus eventos.

Las actividades tienen 3 estados:

Resumed: está ejecutándose.

Paused: parada pero aún visible.

Stopped: parada y no visible.

Cuando la aplicación empieza llama al método onCreate() donde se establecen sus componentes, después se llama al método onStart() para que sea visible. El siguiente método es onResume(), y empieza a correr la actividad. Si se lanza la actividad se pausa ésta, que después volverá al método onResume() para continuar con su ejecución. Aunque también puede darse el caso de que se pause por otras aplicaciones, que entonces necesitaremos volver a cargar los componentes. En vez de pausar, se puede detener llamando al método onStop(), la forma de volver a utilizar esta actividad es llamando al método onRestart(), a no ser que se haya detenido por otras aplicaciones, que entonces, al igual que en el caso anterior, necesitaremos volver a cargar los componentes. Por último, el método onDestroy() libera los recursos de la actividad y se termina.

5. ¿Qué evento utilizarías para grabar el estado de una aplicación?

onPause y onSaveInstanceState ya que los eventos onStop y onDestroy no se garantiza que se llamen.

6. ¿Qué es un fragmento?

Un fragmento es na parte constituyente de una actividad. Formaremos las actividades a partir de uno o varios fragmentos.

7. ¿Cómo añadimos un fragmento a una actividad?

Mediante las transacciones, que se encargan de la gestión para añadir, remplazar o borrar un fragmento de una actividad.

```
FragmentManager fragmentManager = _getFragmentManager();
```

```
FragmentTransaction fragmentTransaction = fragmenManager.beginTransaction();
```

```
Fragment newFragment = new ExampleFragment();
```

```
FragmentTransaction transaction = getFragmentManager().beginTransaction();  
  
transaction.replace(R.id.fragment_container, newFragment);  
  
transaction.addToBackStack(null);  
  
transaction.commit();
```

8. ¿En qué recurso añadiremos las cadenas de texto? ¿De qué forma declararías una cadena?

Values/strings.xml

9. ¿En qué recurso añadiremos los iconos de la aplicación?

Drawable

10. ¿Cómo hacemos en Android para tener otras configuraciones de idiomas?

Para cada traducción crearemos un directorio con el calificador del idioma correspondiente (values-calificador).