DAM 2 – M08: Programación multimedia y dispositivos móviles

Nombre:

Fecha:

1. **¿Qué significa SDK?**
   1. Software Development Kit.
   2. Source Development Kit.
   3. SDK Development Kit.
   4. Soft Development Kit.
2. **¿Qué ventana o vista usaremos para ver la salida de debug de nuestros terminales o emuladores?**
   1. *Devices.*
   2. *Logcat.*
   3. *Output.*
   4. *Debug.*
3. **¿Qué ventana o vista usaremos para ver los mensajes derivados del funcionamiento del emulador (no de las aplicaciones)?**
   1. Devices.
   2. Logcat.
   3. Output.
   4. Debug.
4. **¿Dónde podemos probar nuestras aplicaciones?**
   1. Solo en el emulador.
   2. Solo en un terminal conectado al ordenador.
   3. En el emulador o en un terminal conectado al ordenador.
   4. En una terminal.
5. **¿Cómo se llama el fichero de configuración de las aplicaciones Android?**
   1. Android.conf.
   2. AndroidConf.xml.
   3. AndroidManifest.xml.
   4. AndroidManifest.conf.
6. **Si en el código java se encuentra una referencia a R.strings.app\_name   
   ¿a qué hace referencia?**
   1. A la cadena llamada app\_name dentro de ./res/values/strings.xml.
   2. Al fichero ./res/strings/app\_name.
   3. Al fichero ./res/values/app\_name.
   4. Al fichero ./asserts/app\_name.
7. **¿En qué carpeta colocarías la definición de pantalla de su aplicación?**
   1. ./res/.
   2. ./res/screen/.
   3. ./res/layout/.
   4. ./assets/.
8. **¿En qué carpeta se coloca el código fuente de nuestras clases?**
   1. En cualquiera que no sea la raíz.
   2. ./res/.
   3. ./source/.
   4. ./src/.
9. **¿Dónde se colocan las animaciones de imagenes?**
   1. ./res/xml/.
   2. ./res/drawable/.
   3. ./res/layout/.
   4. ./res/values/.
10. **¿Dónde se colocan se definen los colores de colors.xml?**
    1. ./res/xml/.
    2. ./res/drawable/.
    3. ./res/values/.
    4. ./res/layout/.
11. **Si queremos hacer una interfaz con una serie de objetos ordenados de arriba abajo:**
    1. Usaremos una etiqueta TableLayout con TableRow
    2. Usaremos una etiqueta LinearLayout vertical.
    3. Usaremos una etiqueta ConstraintLayout.
    4. Todas son ciertas.
12. **¿Dónde se colocan se definen los textos?**
    1. ./res/values/.
    2. ./res/drawable/.
    3. ./res/layout/.
    4. ./res/xml/.
13. **¿Dónde se colocan las imágenes que se usan en el diseño de las aplicaciones?**
    1. ./res/values/.
    2. ./res/raw/.
    3. ./res/drawable/.
    4. ./res/images/.
14. **¿Qué extensión tienen que tener los ficheros de código fuente Android?**
    1. .and.
    2. .xml.
    3. .java.
    4. .src.
15. **¿Qué es el fichero de recursos R?**
    1. Una clase que facilita el acceso a todos los ficheros colocados dentro de la carpeta ./res/.
    2. Un fichero con los nombres de ficheros usados en todo el proyecto.
    3. Un fichero que genera el compilador con el listado de las fuentes.
    4. En Android no existe ningún fichero de recursos R.
16. **Una vez identificado un objeto view de java con un recurso xml en la interfaz, ¿Qué métodos son aquellos que nos aportan sus atributos?**
    1. set’s (setText(), setTag() ,etc)
    2. make’s (makeText(), makeTag(), etc).
    3. get’s (getText(), getTag(), etc)
    4. do’s (doText(), doTag(), etc)
17. **¿Cuántas formas tenemos de hacer un botón?**
    1. Definir un objeto java botón y ejecutar el método onClick de la clase Button.
    2. Definir void’s con parámetro un View y definir los botones con el onClick.
    3. Una escucha con onClickListener desde una clase que lo implemente.
    4. b y c son ciertas.
18. **¿De qué clase tienen que heredar las clases encargadas de mostrar las pantallas que interactúan con el usuario?**
    1. MainActivity.
    2. Activity.
    3. Screen.
    4. Gui.
19. **¿Qué afirmación es la correcta?** 
    1. Cada aplicación Android corre en su propio proceso Linux.
    2. Una aplicación Android puede tener varios procesos Linux.
    3. Una aplicación Android tiene tantos procesos Linux como hilos.
    4. Android es un sistema monoproceso.
20. **¿Qué parámetros Android:layout se encargan del área que ocupan los objetos de las interficies .xml?** 
    1. height y width.
    2. weight.
    3. marginTop, marginBottom, marginStart, marginEnd .
    4. Todas son ciertas.
21. **¿Qué permite una FragmentActivity y sus derivados modernos?**
    1. Mostrar actividades haciendo fluir el ciclo de vida en los Fragment.
    2. Poder seleccionar los diferentes Fragment en una misma Activity.
    3. Fragmentar la interficie de la activity para situar los elementos de la interfaz.
    4. Facilitar la comunicación entre Activities.
22. **¿Qué método se llama al crear una Actividad?**
    1. *onCreate().*
    2. *onStart().*
    3. *onResume().*
    4. *onPreStart().*
23. **Si ya tengo una aplicación con una actividad iniciada. ¿Cuál es la secuencia que se ejecutara si bloqueo y desbloqueo la pantalla?**
    1. *onCreate()-onStart()-onResume()-onPause()-onStop()-onDestroy().*
    2. *onPause()-onStop()-onRestart()-onStart()-onResume()*
    3. *onDestroy()-onPause()-onRestart()-onStart-onResume().*
    4. *onPause()-onStop()-onDestroy()-onCreate()-onStart()-onResume().*
24. **Dentro del ciclo de vida de una actividad, ¿cuál es el último método que se llama?**
    1. *onStop().*
    2. *onPause().*
    3. *onDestroy().*
    4. *onFinish().*
25. **¿De qué forma se organizan los objetos View en Android?**
    1. En forma lineal.
    2. En forma secuencial.
    3. En forma de árbol.
    4. No tienen un orden fijo.
26. **¿Cómo podemos definir nuestra interface de usuario?**
    1. Mediante un fichero XML.
    2. Con código Android (Java/Kotlin).
    3. Mediante un fichero XML y/o código Android(Java/Kotlin).
    4. Mediante un fichero de tipo png.
27. **¿A qué método de nuestra actividad debemos llamar para establecer nuestro objeto View principal?**
    1. setMainView.
    2. setContentView.
    3. setRootView.
    4. setView.
28. **¿Para identificar cada uno de los componentes de la vista que atributo utilizamos?**
    1. id.
    2. name.
    3. key.
    4. view.
29. **¿Que formas hay de definir una intención?**
    1. Intent(Intent.ACTION\_X) con X en {VIEW, DIAL, CALL, IMAGE\_CAPTURE, SEND, etc}
    2. checkPermission\_Intent y Intent.ACTION\_CALL.
    3. Implícita con contexto Intent y explicita con contexto especifico.
    4. Ninguna es correcta.
30. **¿Qué Intent.ACTION de los siguientes abre la aplicación del teléfono del sistema android?**
    1. Intent.ACTION\_VIEW
    2. Intent.ACTION\_CALL
    3. Intent.ACTION\_IMAGE\_CAPTURE
    4. Intent.ACTION\_DIAL
31. **¿Dónde se definen los permisos?**
    1. Build.gredle.kts (Module:App)
    2. Permission.xml
    3. AndroidManifest.xml
    4. MainActivity.java
32. **¿La configuración del constructor de Android con las dependencias y pluguins en que archivo está?**
    1. Build.gredle.kts (Module:App)
    2. Permission.xml
    3. AndroidManifest.xml
    4. MainActivity.java
33. **¿Cómo añadimos información a adicional a una intención?**
    1. intent.putExtra(Contexto,recurso)
    2. intent.addKey(key, recurso)
    3. intent.setInfo(key, recruso)
    4. Ninguna de las anteriores.
34. **Los identificadores definidos en tiempo de compilación se guardan en una clase generada ¿en cuál?**
    1. En una clase de recursos llamada *Interface.*
    2. En una clase de recursos llamada *R.*
    3. En una clase de recursos llamada *Resources.*
    4. En una clase de recursos llamada *Android.*
35. **¿Cuál de estas formas de acceder al contexto es incorrecta?**
    1. getMyContext().
    2. getContext().
    3. This.
    4. getApplicationContext().
36. **Los recursos se guardan en la carpeta “res”, pero ¿dónde se coloca esta carpeta dentro de nuestro proyecto?**
    1. Donde queramos.
    2. Donde queramos, pero por convenio se usa ./res/.
    3. En /res.
    4. En el paquete, junto a las clases que las van a usar.
37. **¿En qué subcarpeta se guardan los diseños de vista?**
    1. res/layout.
    2. res/view.
    3. res/drawable.
    4. res/xml.
38. **Si quiero traducir mi aplicación a inglés, ¿en qué carpeta colocaré sus textos?**
    1. res/values/strings-en.xml.
    2. res/values-en/strings.xml.
    3. res-en/values/strings.xml.
    4. Da igual siempre que lo defina en el manifiesto.
39. **Al archivo “res/drawable/ejemplo.png” se podrá acceder desde nuestro código java con la clase R ¿Cuál será su ruta?**
    1. R.drawable.ejemplo.
    2. R.drawable.ejemplo\_png.
    3. R.resources.drawable.ejemplo.
    4. R.getDrawable(“ejemplo.png”).
40. **La clase Bundle sirve para compartir información…**
    1. Para definir preferencias de la aplicación de forma general a todas las aplicaciones
    2. Para crear o restaurar actividades con datos de otras actividades.
    3. Para comunicar los objetos de diferentes Fragment.
    4. Todas son ciertas.
41. **¿Qué clase nos permite guardar datos asociados a una clave de forma persistente e independiente de las actividades de la aplicación?**
    1. SharedPreferences.
    2. SharedData.
    3. KeyPreferences.
    4. KeyData.
42. **¿Qué tipos de datos puede guardar la clase SharedPreferences?**
    1. Cualquier tipo de datos.
    2. Sólo los datos que implementen su interface.
    3. Datos de tipo primitivo (*boolean, float, int, long, string).*
    4. Ninguna de las anteriores.
43. **¿Qué tipo de Activity podemos usar para implementar un editor de preferencias?**
    1. Activity.
    2. ListActivity.
    3. MapActivity.
    4. PreferenceActivity.
44. **¿Qué necesita la actividad PreferenceActivity?**
    1. Un *layout* con la configuración de las preferencias.
    2. Un fichero xml con la configuración de las preferencias.
    3. Un *layout* que muestre el formulario de preferencias.
    4. Nada en especial.
45. **¿Cómo puedo recuperar una preferencia guardada mediante la clase PreferenceActivity?**
    1. Con el método getSharedPreferences de Context.
    2. Con el método getSharedPreferences de la Activity.
    3. Con el método getSharedPrefrerences de la clase Application.
    4. Creando un objeto SharedPreferences.