UNIVERSITAT DE BARCELONA

Software Integrated Project (2019-2020)

Trial Exam

Karim Lekadir, PhD
Universitat de Barcelona
Dpt. Matemàtiques & Informàtica

1. Software Mock-up

Question 1

¿Cuál es el objetivo principal de un mock-up?

A. Presentar la estructura del código

B. Definir todas las funcionalidades de la app

C. Mostrar todas las interficies de la app de manera genérica

D. Ninguna de las anteriores

Answer 1

¿Cuál es el objetivo principal de un mock-up?

A. Presentar la estructura del código

B. Definir todas las funcionalidades de la app

C. Mostrar todas las interficies de la app de manera genérica

D. Ninguna de las anteriores

Question 2

¿Que debe incluir un mock-up?

A. Interficies gráficas detalladas

B. Intents en formato de flechas

C. Descripción de las clases

D. Todas las anteriores

Answer 2

¿Que debe incluir un mock-up?

A. Interficies gráficas detalladas

B. Intents en formato de flechas

C. Descripción de las clases

D. Todas las anteriores

2. LAB questions

Question 1

¿Qué tres métodos son llamados cuando se inicia una actividad desde cero?

A. OnStart(), OnResume()

B. OnCreate(), OnResume()

C. OnCreate(), OnResume(), OnStart()

D. OnCreate(), OnStart(), OnResume()

Answer 1

¿Qué tres métodos son llamados cuando se inicia una actividad desde cero?

A. OnStart(), OnResume()

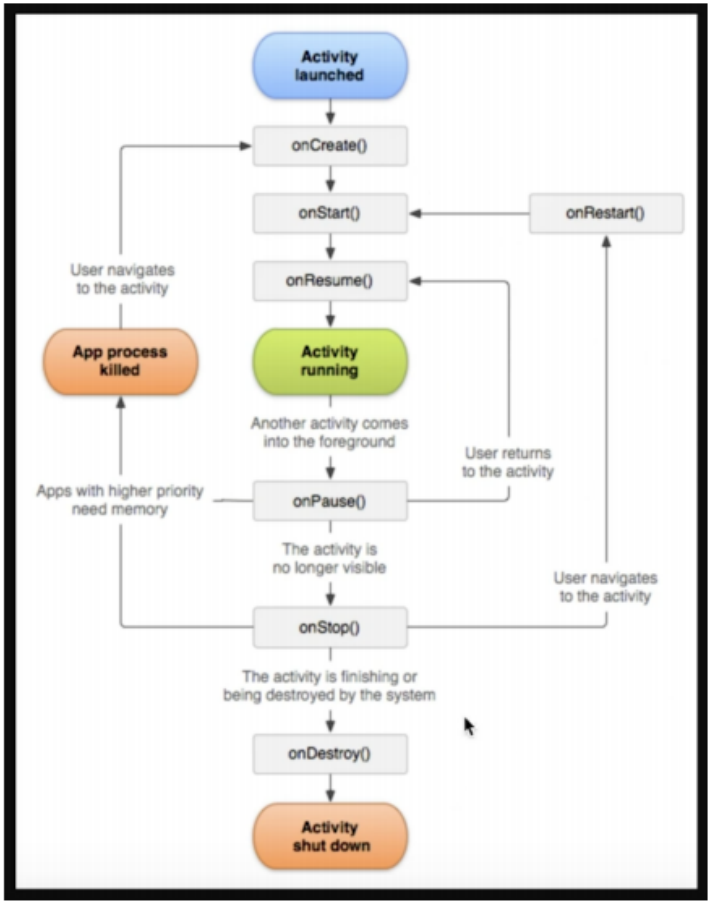
B. OnCreate(), OnResume()

C. OnCreate(), OnResume(), OnStart()

D. OnCreate(), OnStart(), OnResume()

Answer 1

Cicle de vida d'una aplicació



Les apps tenen diferents processos. Es important definir un mock-up inicial i ordenar tots els processos amb els seus respectius layouts.

- Es pot dividir en 3 procesos cíclics (onResume, onStart and onCreate) depenent de la interficie i els procesos en marxa.

Question 2

¿El atributo *orientation* a que layout se aplica?

A. Relative layout
B. Linear layout
C. Grid layout
D. Frame layout

Answer 2

¿El atributo *orientation* a que layout se aplica?

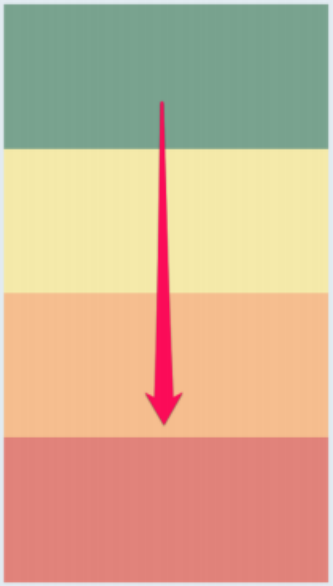
A. Relative layout
B. Linear layout
C. Grid layout
D. Frame layout

Answer 2

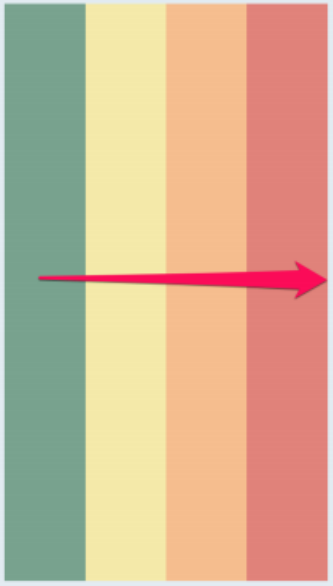
Tipos de Layout - Linear Layout

Linear layout distribuye todas los contents (Views) dentro del ViewGroup a lo largo de la dimensión establecida. La orientación la puedes escoger a través del atributo **android: orientation**.

android:orientation="vertical"



android:orientation="horizontal"



Question 3

¿Dónde se deben declarar los permisos de internet para vuestra app?

A. En app/build.graddle

B. En el Manifest.xml

C. No se debe declarar. Android 9.0 ya declara el acceso a internet por defecto

D. En settings.graddle de la carpeta graddle.scripts

Answer 3

¿Dónde se deben declarar los permisos de internet para vuestra app?

A. En app/build.graddle

B. En el Manifest.xml

C. No se debe declarar. Android 9.0 ya declara el acceso a internet por defecto

D. En settings.graddle de la carpeta graddle.scripts

Answer 3

```
<manifest xlmns:android...>
 ...
 <uses-permission android:name="android.permission.INTERNET" />
 <application ...
</manifest>
```

Question 4

¿Qué datos no es necesario definir en un intent implicito?

A. Data
B. Component Name
C. Action
D. Ninguna de las anteriores

Answer 4

¿Qué datos no es necesario definir en un intent implicito?

A. Data

B. Component Name

C. Action

D. Ninguna de las anteriores

Answer 4

Cómo crear un Intent

Un objeto **Intent** tiene información que el sistema Android usa para determinar qué componente debe iniciar (como el nombre exacto del componente o la categoría que debe recibir la intent), además de información que el componente receptor usa para realizar la acción correctamente (por ejemplo, la acción que debe efectuar y los datos en los que debe actuar).
Los campos a crear para definir una intent són los siguientes:

1. **Nombre del componente**. Componente que se debe iniciar. Opcional. Lo que define una intent como explícita.
2. **Acción** String que define la acción genérica a realizar. (View, Send, etc)
3. **Datos** Tipo de datos a manejar por la aplicación que genera y/o recibe
4. **Categoría** String con información sobre los datos a manejar.

Question 5

Selecciona la opción verdadera:

A. PutExtra() es un método que se utiliza para enviar información a una actividad lanzada por nosotros

B. El fichero res/values/styles.xml sirve para definir los estilos

C. OnPause() es el método más fiable para guardar el estado de una actividad.

D. Todas las opciones son verdaderas.

Answer 5

Selecciona la opción verdadera:

A. PutExtra() es un método que se utiliza para enviar información a una actividad lanzada por nosotros

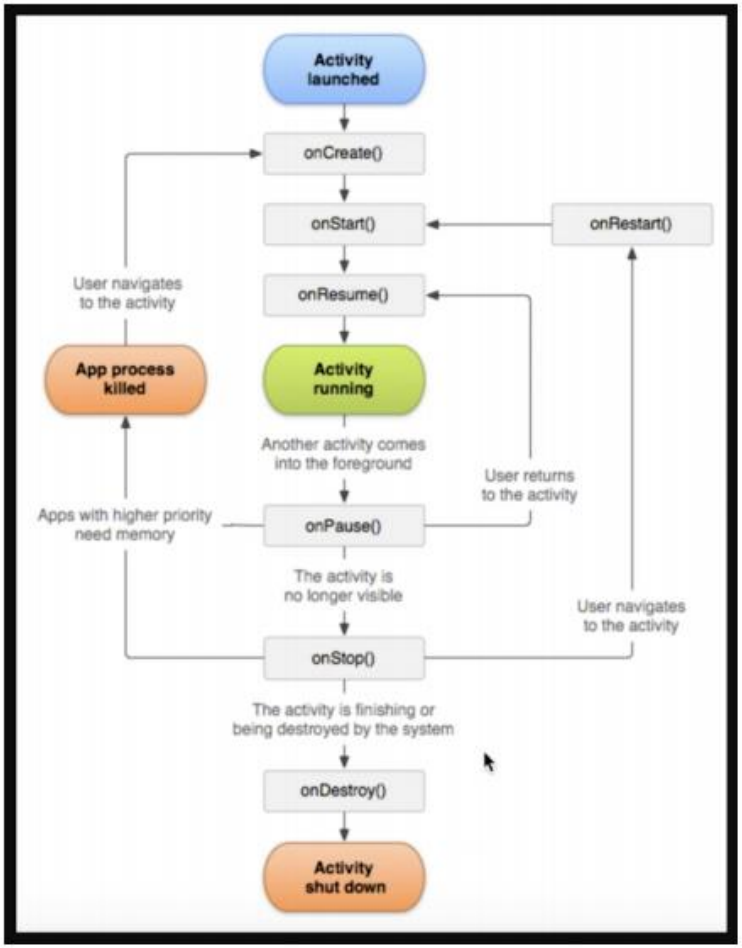
B. El fichero res/values/styles.xml sirve para definir los estilos

C. OnPause() es el método más fiable para guardar el estado de una actividad.

D. Todas las opciones son verdaderas.

Answer 5

Cicle de vida d'una aplicació



Question 6

¿Qué tipo de archivos se guardan en la caché (SharedPreferences)?

A. Todos los datos del usuario.

B. Tan sólo el último dato introducido por el usuario (LastSharedPreference)

C. Pequeños volúmenes de datos, como datos de usuario o el último estado de la app

D. Todos los anteriores.

Answer 6

¿Qué tipo de archivos se guardan en la caché (SharedPreferences)?

A. Todos los datos del usuario.

B. Tan sólo el último dato introducido por el usuario (LastSharedPreference)

C. Pequeños volúmenes de datos, como datos de usuario o el último estado de la app

D. Todos los anteriores.

Answer 6

SHARED PREFERENCES

Android proporciona otro método alternativo diseñado específicamente para administrar la serielización de datos: las preferencias compartidas o shared preferences. Cada preferencia se almacenará en forma de clave-valor, es decir, cada una de ellas estará compuesta por un identificador único (p.e. email) y un valor asociado a dicho identificador (p.e. prueba@email.com). Además, los datos se guardarán en ficheros XML, formato interpretable de la misma forma que JSON.Este método es apto para guardar **pequeños volúmenes de información**, como por ejemplo el estado de la aplicación.

Question 7

¿Para qué sirve la serialización de datos?

A. Para agilizar las co-rutinas de la aplicación Android.

B. Para transformar objetos a flujos de bytes y poder transportarlo a través de la red, y reconstruirlos en distintas ubicaciones y/o máquinas.

C. Para conectar el Firebase con las SharedPreferences.

D. Todas las anteriores son ciertas.

Answer 7

¿Para qué sirve la serialización de datos?

A. Para agilizar las co-rutinas de la aplicación Android.

B. Para transformar objetos a flujos de bytes y poder transportarlo a través de la red, y reconstruirlos en distintas ubicaciones y/o máquinas.

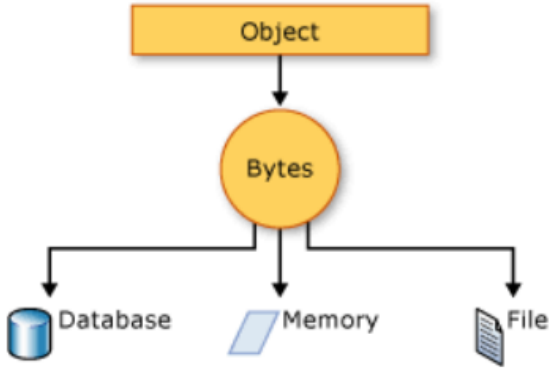
C. Para conectar el Firebase con las SharedPreferences.

D. Todas las anteriores son ciertas.

Answer 7

Serialización

La serialización (o marshalling en inglés) consiste en un proceso de codificación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible como XML o JSON, entre otros.



Question 8

¿Cuál de los siguientes atributos modificará la forma en la que se visualizará la vista?

A. Id

B. Tag

C. OnClick()

D. Ninguno

Answer 8

¿Cuál de los siguientes atributos modificará la forma en la que se visualizará la vista?

A. Id

B. Tag

C. OnClick()

D. Ninguno

Question 9

¿Dónde se declaran las características necesarias para poder recibir/interpretar un intent?

A. En l'Activity principal
B. En un filtro de intents en build.graddle
C. En un filtro de intents en Manifest.xml
D. En todas las anteriores

Answer 9

¿Dónde se declaran las características necesarias para poder recibir/interpretar un intent?

A. En l'Activity principal

B. En un filtro de intents en build.graddle

C. En un filtro de intents en Manifest.xml

D. En todas las anteriores

Answer 9

Filtros

En el proceso de recibir una Intent, debes definir que parámetros ha de tener para poder recibir en una actividad determinada. Para esto se diseñan filtros, con un elemento <intent-filter> declarados en el <manifest.xml>. Un componente de aplicación debe declarar filtros independientes para cada tarea única que puede hacer. Se deben especificar estos elementos:

- <**action**>
- <**data**>
- <**category**>

Question 10

¿Dónde se almacenan los ficheros de SharedPreferences?

A. Caché (forma local)
B. Res/preferences
C. Res/values
D. Res/strings

Answer 10

¿Dónde se almacenan los ficheros de SharedPreferences?

A. Caché

B. Res/preferences

C. Res/values

D. Res/strings

Question 11

¿Para que sirven las co-rutinas en Android?

A. Para crear hilos diferentes para la ejecución de distintos procesos asincronos en la app y agilizar la performance

B. Para conectar el Firebase con la consola

C. Para conectar la app a un servidor mediante PHP

D. Ninguna de las anteriores

Answer 11

¿Para que sirven las co-rutinas en Android?

A. Para crear hilos diferentes para la ejecución de distintos procesos asincronos en la app y agilizar la performance.

B. Para conectar el Firebase con la consola

C. Para conectar la app a un servidor mediante PHP

D. Ninguna de las anteriores

Answer 11

Cómo mejorar el rendimiento de la app con las corrutinas de Kotlin

Una *corrutina* es un patrón de diseño de simultaneidad que puedes usar en Android para simplificar el código que se ejecuta de manera asíncrona. Las corrutinas se agregaron a Kotlin en la versión 1.3 y están basadas en conceptos establecidos de otros idiomas.

En Android, las corrutinas ayudan a solucionar dos problemas principales:

- Administran tareas prolongadas que, de lo contrario, podrían bloquear el subproceso principal y provocar que tu app quede inmovilizada.
- Proporcionan *seguridad del subproceso principal* o llaman de manera segura a operaciones de red o disco desde el subproceso principal.

En este tema, se describe cómo puedes usar las corrutinas de Kotlin para solucionar estos problemas, lo que te permite escribir código de apps más limpio y conciso.

Question 12

¿Cuál es la funcionalidad de la libreria GSON?

A. Convertir objetos Java a Kotlin, y viceversa
B. Convertir objetos Java a .txt, y viceversa
C. Transformar QuerySnapshots de Firebase en archivos JSON.
D. Convertir objetos Java a Json, y viceversa

Answer 12

¿Cuál es la funcionalidad de la libreria GSON?

A. Convertir objetos Java a Kotlin, y viceversa

B. Convertir objetos Java a .txt, y viceversa

C. Transformar QuerySnapshots de Firebase en archivos JSON.

D. Convertir objetos Java a Json, y viceversa

Answer 12

GSON

JSON es un formato de datos de texto muy popular que se usa para intercambiar datos en las aplicaciones web y móviles modernas. Por ejemplo, en SQL, Azure o extremos de la API Rest. Si deseamos acceder a datos de otras BD (SQL Server), necesitaremos crear esos archivos JSON con la función (añadir datos/leer datos) a través de nuestro código Kotlin. Para eso existe **GSON library**, que nos permite traducir classes Kotlin a JSON, y viceversa.

Question 13

¿Para generar un selector de apps, que función va asociada al intent?

A. CreateSelector()

B. CreateChooser()

C. CreateFilter()

D. IntentSelector()

Answer 13

¿Para generar un selector de apps, que función va asociada al intent?

A. CreateSelector()

B. CreateChooser()

C. CreateFilter()

D. IntentSelector()

Answer 13

Selector de apps

Cuando creas un intent implícito, necesariamente pueden existir varias aplicaciones que puedan ejecutar el Intent. Para ello, a veces requerimos de diseñar un selector que nos permita escoger la aplicación con la que queremos interactuar.

```
val sendIntent = Intent(Intent.ACTION_SEND)
...

// Always use string resources for UI text.
// This says something like "Share this photo with"
val title: String = resources.getString(R.string.chooser_title)
// Create intent to show the chooser dialog
val chooser: Intent = Intent.createChooser(sendIntent, title)

// Verify the original intent will resolve to at least one activity
if (sendIntent.resolveActivity(packageManager) != null) {
    startActivity(chooser)
}
```

Para mostrar el diálogo de selección, crea una Intent usando createChooser() y transfiérela a startActivity(), tal como se muestra en el siguiente ejemplo. Aquí se muestra un diálogo con una lista de apps que responden a la intent transferida al método createChooser(), con el texto proporcionado como título del diálogo.

Question 14

Define la función del siguiente código:

```
<manifest ...
  android:installLocation="preferExternal">
  ...
</manifest>
```

A. Instalar la app en la memoria física del telefono

B. Instalar la app en la memoria externa/extraíble del telefono.

C. Obtener la ubicación desde una API externa

D. Ninguna de las anteriores

Answer 14

Define la función del siguiente código:

```
<manifest ...
  android:installLocation="preferExternal">
  ...
</manifest>
```

A. Instalar la app en la memoria física del telefono

B. Instalar la app en la memoria externa/extraíble del telefono.

C. Obtener la ubicación desde una API externa

D. Ninguna de las anteriores