Solucionario de

ejercicios de autoevaluación

Ejercicios de autoevaluación Unidad de Aprendizaje 1

1. Java es:

a. Un lenguaje de programación.
b. Una plataforma completa de tecnologías, herramientas y librerías que proporcionan un entorno de ejecución en cualquier plataforma.
c. Un sistema que ejecuta programas en distintas plataformas
d. Un conjunto de librerías.

2. Determina si la siguiente oración es verdadera o falsa: "Los programas que escribimos en Java son multiplataforma porque el entorno Java incluye una máquina virtual que interpreta el código *bytecode".*

- **Verdadero**
- Falso

3. Indica, entre las siguientes opciones, cuál es la que define mejor que "Java es un lenguaje de alto rendimiento":

a. Ideal para pequeñas compañías.
b. Mueve masas.
c. Ideal para grandes compañías.
d. Llega a todo tipo de público.

4. La *JVM,* o *Java Virtual Machine,* se encarga de:

a. Contener un número de librerías para la escritura de Java.
b. Codificar el código nativo a código intermedio o *bytecode.*
c. Facilita el mantenimiento de *software.*
d. Interpreta el código *bytecode* en tiempo de ejecución.

5. Determina si la siguiente oración es verdadera o falsa: "Para que un entorno de desarrollo integrado sea considerado como tal, debe estar compuesto por un editor de texto y herramientas de automatización, nada más".

- Verdadero
- **Falso**

6. La legibilidad de un programa informático es importante porque:

a. Evita errores a la hora de compilar.
b. Te permite corregir errores con rapidez, aunque no facilita el mantenimiento en el tiempo de una aplicación.
c. Reduce el coste de mantenimiento del *software* permitiendo corregir errores a menor esfuerzo.
d. Así toda la comunidad informática está de acuerdo.

7. El recolector de basura de Java*:*

a. Comprueba que todos los elementos sean palabras válidas.
b. Optimiza el rendimiento del código.
c. Verifica la estructura sintáctica de un programa.
d. Referencia las variables que se hayan quedado sin referenciar.

8. Los *IDE* deben cumplir algunas características, entre las que no se encuentra:

a. Ejecutar un programa informático.
b. Ser compatible con más de una plataforma.
c. Reconocer la sintaxis en la que se escribe.
d. Soportar varios lenguajes de programación.

9. Determina si la siguiente oración es verdadera o falsa: "Java es un lenguaje que diferencia entre mayúscula y minúscula".

- **Verdadero**
- Falso

10. Establece el orden adecuado si quisiéramos realizar una conversión automática de tipos.

a. *byte*
b. *double*
c. *long*
d. *int*

1. *byte*
2. *int*
3. *long*
4. *double*

Ejercicios de autoevaluación Unidad de Aprendizaje 2

1. ¿Qué descripción explica mejor el concepto de clase de Java?

a. Es un tipo especial de variable.
b. Es un modelo según el cual ordenamos los datos de las clases.
c. Es una variable muy parecida al *Array.*
d. Son una clase de datos ordenador de manera secuencial.

2. ¿De qué distintas partes se compone una clase?

a. Sus funciones y variables.
b. El *software* y el *hardware.*
c. Su cardinalidad y su tipo.
d. Sus métodos y atributos.

3. Determina si la siguiente oración es verdadera o falsa: "Las características principales de un objeto en Java son la identidad, su estado y su comportamiento".

- **Verdadero**
- Falso

4. Indica, entre las siguientes opciones, qué código tiene que ver con la herencia:

a. Public class Deportista extends Participante
b. Public class Deportista inherit Participante
c. Public class Deportista implements Participante
d. Public class Deportista belong to Participante

5. ¿Cuál es el resultado de instanciar a una clase?

a. Tendremos una clase duplicada.
b. La clase se eliminará.
c. Se conecta la clase con otra.
d. Se crea un objeto del tipo clase elegido.

6. Determina si la siguiente oración es verdadera o falsa: "La capacidad de crear clases abstractas en Java no es necesaria dentro de la POO".

- Verdadero
- **Falso**

7. ¿Qué código asociarías a un interfaz de Java?

a. Masajista ms = new Masajista (InterfazPrueba)
b. Masajistas ms =new Masajista. InterfazPrueba
c. public class Masajista implements InterfazPrueba
d. pblic class Masajista extends InterfazPrueba

8. ¿Qué implica para un método el hecho de sobrecargarlo?

a. Editar el contenido para cambiar su comportamiento.
b. Crear un método de mismo nombre y distintos atributos.
c. Añadir funcionalidades nuevas.
d. Cambiarle el nombre dejando la misma funcionalidad.

9. Determina si la siguiente oración es verdadera o falsa: "Java es un lenguaje que diferencia entre mayúscula y minúscula".

- **Verdadero**
- Falso

10. Relaciona los conceptos con las definiciones:

a. Programación declarativa.
b. Programación funcional.
c. Programación con restricciones.
d. Programación imperativa.
e. Programación estructurada.
f. Programación procedimental.

b. Se basa en el uso de funciones.
f. Lo usan lenguajes como C++ o Pascal.
d. Establece su ejecución en forma de sentencias.
a. Se declaran los problemas de manera inmutable.
e. Se ejecuta de principio a fin la lectura del fichero.
c. Expresa la relación entre variables con restricciones.

Ejercicios de autoevaluación Unidad de Aprendizaje 3

1. **¿Qué descripción explica mejor el concepto de clase *Exception* de Java?**

    a. *Exception* tiene como subclase la clase *Throwable.*
    b. *Throwable* tiene como subclase la clase *Exception.*
    c. Es una clase que sirve para parar la ejecución.
    d. Son una clase de datos ordenador de manera secuencial.

2. **La llamada a *finally* se produce siempre al final del bloque *try/ catch,* excepto cuando...**

    a. ... se captura un error.
    b. ... solo exista un bloque *finally* en la aplicación.
    c. ... el *finally* está vacío.
    d. ... hay un fallo o un *crash* primero.

3. **Determina si la siguiente oración es verdadera o falsa: "La palabra reservada *throws* del lenguaje de programación Java sirve para lanzar una excepción dentro del código".**

    ■ Verdadero
    ■ **Falso**

4. **Indica, entre las siguientes frases, qué afirmación es cierta si hacemos referencia a la clase *Collection:***

    a. Esta puede ser recorrida.
    b. Debe instanciar un iterador.
    c. Tiene que ser recorrida obligatoriamente con un bucle *for.*
    d. No es implementada sobre ninguna clase.

5. **El bucle *forEach* de Java:**

    a. Es llamado así porque se convoca escribiendo *forEach.*
    b. Necesita una variable para su inicialización.
    c. Se debe usar siempre en vez del bucle *for* simple.
    d. Es una alternativa al uso de los clásicos iteradores.

6. **Determina si la siguiente oración es verdadera o falsa: "Las variables de clase se llaman de esta manera porque el valor que contengan está asociado a todas las instancias de esta misma clase".**

   - **Verdadero**
   - Falso

7. **Los métodos de clase no pueden acceder a los valores que contenga la clase que no sean de clase, esto es debido a que:**

   a. Los valores de instancia van asociados a la clase y no al objeto.
   b. Estos no pueden invocar el objeto.
   c. Los valores de instancia van asociados al objeto y no a la clase.
   d. Dependen del método *main* para ejecutarse.

8. **Los tipos enumerados son considerados como:**

   a. Similares a los atributos de una clase.
   b. Paralelos a las clases y las interfaces del lenguaje de programación Java.
   c. Grupos de valores dentro de una clase.
   d. Funcionalidades recogidas en una variable.

9. **La interfaz Serializable de Java sirve para:**

   a. Incluir en su interior otras clases.
   b. Trabajar el traspaso de la información de un objeto.
   c. Reconocer la sintaxis en la que se escribe el objeto.
   d. Soportar varios lenguajes de programación dentro de un mismo programa.

10. **Determina si la siguiente oración es verdadera o falsa: "Para realizar la deserialización de un objeto, haremos uso de la clase *ObjectInputStream*".**

    - **Verdadero**
    - Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 4

1. ¿Cuál de las siguientes opciones no es parte de la instalación necesaria para poder desarrollar aplicaciones *Android?*

a. *Android Studio.*
b. *C++ Compiler.*
c. *Java Runtime Envoriment.*
d. *Android SDK.*

2. ¿De qué distintas partes se compone una clase?

a. Sus funciones y variables.
b. El *software* y el *hardware.*
c. Su cardinalidad y su tipo.
d. Sus métodos y atributos.

3. Determina si la siguiente oración es verdadera o falsa: "Las características principales de un objeto en Java son la identidad, su estado y su comportamiento".

- **Verdadero**
- Falso

4. ¿Cuál de los códigos tiene que ver con la herencia?

a. Public class Deportista extends Participante
b. Public class Deportista inherit Participante
c. Public class Deportista implements Participante
d. Public class Deportista belong to Participante

5. ¿Cuál es el resultado de instanciar a una clase?

a. Tendremos una clase duplicada.
b. La clase se eliminará.
c. Se conecta la clase con otra.
d. Se crea un objeto del tipo clase elegido.

6. **Determina si la siguiente oración es verdadera o falsa: "La capacidad de crear clases abstractas en Java no es necesario dentro de la POO".**

   - Verdadero
   - **Falso**

7. **¿Qué código asociarías a una interfaz de Java?**

   a. Masajista ms = new Masajista (InterfazPrueba)
   b. Masajistas ms =new Masajista. InterfazPrueba
   c. public class Masajista implements InterfazPrueba
   d. pblic class Masajista extends InterfazPrueba.

8. **¿Qué implica para un método el hecho de sobrecargarlo?**

   a. Editar el contenido para cambiar su comportamiento.
   b. Crear un método de mismo nombre y distintos atributos.
   c. Añadir funcionalidades nuevas.
   d. Cambiarle el nombre dejando la misma funcionalidad.

9. **Los *IDE* deben cumplir algunas características, entre las que no se encuentra:**

   a. Ejecutar un programa informático.
   b. Ser compatible con más de una plataforma.
   c. Reconocer la sintaxis en la que se escribe.
   d. Soportar varios lenguajes de programación.

10. **Determina si la siguiente oración es verdadera o falsa: "Java es un lenguaje que diferencia entre mayúscula y minúscula".**

    - **Verdadero**
    - Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 5

1. ¿Qué vías usaremos para comprobar los niveles de *API* que soporta nuestra aplicación?

a. Dispositivo físico y emulador.
b. En este caso, con el emulador podemos comprobarlo.
c. Dispositivo físico.
d. Todas las opciones son incorrectas.

2. ¿En qué carpeta de nuestra aplicación guardaremos los recursos visuales y diferentes configuraciones sobre diseño?

a. Directorio *manifests*.
b. Carpeta Java.
c. Raíz del proyecto.
d. El directorio res.

3. Determina si la siguiente oración es verdadera o falsa: "Los recursos que incluimos dentro de las bibliotecas que creamos deben ser privados para que no sean accesibles por desarrolladores externos".

- **Verdadero**
- Falso

4. Indica, entre las siguientes opciones, qué frase es incorrecta:

a. El *SDK* de *Android* es un emulador de dispositivos móviles.
b. El *SDK* es ofrecido a los desarrolladores *Android* de manera gratuita.
c. Dentro de las herramientas del *SDK* encontramos *QEMU*.
d. Las aplicaciones *Android* se escriben en lenguaje Java sobre el *SDK*.

5. ¿Cuál es el resultado de instanciar a una clase?

a. Tendremos una clase duplicada.
b. La clase se eliminará.

c. Se conecta la clase con otra.
d. Se crea un objeto del tipo clase elegido.

6. Determina si las siguientes oraciones son verdaderas o falsas:

a. Un módulo de biblioteca solo puede ser usado una vez en otro módulo, y en ningún otro.

- Verdadero
- **Falso**

b. Cuando otra aplicación con mayor prioridad entra en espera por el hueco de memoria que tiene nuestra aplicación, es posible que esta no pase por un estado detenida o pausada.

- **Verdadero**
- Falso

7. ¿De qué canal se descargarán de manera predefinidas las actualizaciones del *Android SDK*?

a. Canal DEV.
b. Canal BETA.
c. Canal ESTABLE.
d. Canal CANARY.

8. El fichero AndroidManifest.xml sirve para:

a. Dar permisos a nuestras aplicaciones.
b. Guardar conjuraciones generales del uso de nuestra aplicación.
c. Registrar las actividades que contenga nuestra aplicación.
d. Todas las opciones son correctas.

9. ¿Qué quiere decir la siguiente expresión?

```
Intent intent = new Intent(MainActivity1.this, MainActivity2.class);
```

a. Mostramos la MainActivity2 desde la MainActivity1.
b. Ser compatible con más de una actividad.

c. Estamos creando a la vez las dos actividades.
d. Soportar varios lenguajes de programación.

10. Los *Fragment* son herramientas con un ciclo de vida...

a. ... propio.
b. ... heredado de *Activity*.
c. ... pensado para crear interfaces.
d. ... que soportan varias clases distintas.

Ejercicios de autoevaluación Unidad de Aprendizaje 6

1. **Cuando queremos realizar el registro de una vista, debemos dirigirnos a la carpeta del sistema:**

   a. /java
   b. /res
   c. test/
   d. androidTest/

2. **El ID que encontramos dentro de nuestra vista, al ser declarado, queda inmediatamente registrado dentro de la clase de *Android* siguiente:**

   a. *Activity*
   b. *View*
   c. *ViewGroup*
   d. R

3. **Determina si la siguiente oración es verdadera o falsa: "La clase *ListActivity* servirá para mostrar un conjunto de datos en forma de cuadrícula".**

   - Verdadero
   - **Falso**

4. **Elige entre las siguientes opciones cuál es la más adecuada para definir un elemento *ListView:***

   a. Mostrará un listado de imágenes.
   b. Mostrará un elemento de listado de cadenas de texto.
   c. Mostrará un elemento de tipo listado de valores enteros.
   d. Mostrará una lista de vistas.

5. ¿Cuál de las siguientes descripciones explicaría mejor qué es una clase de tipo Adaptador?

a. Esta sería una subclase de *BaseAdapter.*
b. Es la clase encargada de mostrar las distintas vistas de un elemento de tipo listado.
c. Es el conector entre un contenedor de datos y un listado.
d. Todas las opciones son correctas.

6. Determina si la siguiente oración es verdadera o falsa: "Cuando un evento es llamado por el sistema, el *framework Android* lo capturará través de los diferentes gestores de eventos".

- **Verdadero**
- Falso

7. Para hacer el lanzamiento de una actividad, lo haremos a través de la llamada a:

a. starActivity()
b. <intent-filter>
c. new()
d. AndroidManifest.xml

8. Dentro de los *Intents* contamos con una función que nos permitirá, desde una actividad dada, mandar información a una segunda actividad. Esta función es:

a. starActivity()
b. putExtra
c. startActivityForResult()
d. onActivityResult()

9. Cuando una actividad llama a startActivityForResult invocando el lanzamiento de una segunda actividad, esta podrá enviar el resultado a la primera a través de:

a. onActivityResult()
b. startActivityForResult()
c. delete()
d. query()

10. Determina si la siguiente oración es verdadera o falsa: "Para poder realizar el lanzamiento de un *Intent* de tipo implícito, no debemos nunca olvidar registrar el filtro adecuado dentro de nuestro fichero *AndroidManifest.xml".*

- **Verdadero**
- Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 7

1. **¿Qué nueva actividad predefinida de *Android* trae por defecto la posibilidad de realizar la navegación de manera horizontal?**

   a. Actividad de *Scroll.*
   b. Actividad con pestañas.
   c. Actividad con panel lateral.
   d. Actividad básica.

2. **El tipo de vista que, por defecto, nos añade una barra superior y un botón flotante sería:**

   a. Actividad de pantalla completa.
   b. Actividad de *Scroll.*
   c. Actividad vacía.
   d. Actividad básica.

3. **Determina si la siguiente oración es verdadera o falsa: "La usabilidad de nuestras aplicaciones a veces es dejada en segundo plano cuando es una valiosa cualidad para el futuro de nuestras aplicaciones".**

   - **Verdadero**
   - Falso

4. **Cuando queremos referirnos a la clase de la que heredan sus propiedades los distintos contenedores, hablamos de la clase:**

   a. *ViewGroup*
   b. *Toolbar*
   c. *Widget*
   d. *View*

5. **Para que la aplicación de un estilo surja efecto en los elementos hijos y estos atributos sean heredados, lo más recomendable es:**

   a. Crear un estilo más reducido.
   b. Enlazar los estilos a través de código.

c. Establecer los estilos en XML.
d. Colocar el estilo a nivel de toda la actividad o aplicación.

6. Determina si la siguiente oración es verdadera o falsa: "Cuando tratamos de cargar distintos estilos siempre son compatibles entre ellos y todo es visualizado de la manera esperada".

- Verdadero
- **Falso**

7. Los parámetros de diseño que se establecen de manera obligatoria son dos:

a. layout_width y layout_height.
b. layout_margin y layout_padding.
c. layout_weight y layout_height.
d. layout_padding y layout_height.

8. Cuando no sabemos mucho sobre código XML pero queremos editar las vistas de nuestras pantallas, ¿qué recurso pone a nuestra disposición *Android Studio?*

a. El editor XML.
b. La vista de diseño.
c. Código Java.
d. Creador de diseños.

9. Para realizar la modificación de las vistas, tenemos dos maneras:

a. Por código Java o XML.
b. Por código *Android* o código Java.
c. A través de XML o HTML.
d. Todas las opciones son incorrectas.

10. Determina si la siguiente oración es verdadera o falsa: "Los *layouts* son elementos que codifican la interfaz que van a tener nuestras aplicaciones".

- **Verdadero**
- Falso

Ejercicios de autoevaluación
Unidad de Aprendizaje 8

1. **Si necesitamos presentar al usuario una información concreta que debe aceptar o una cuestión sobre la que debe tomar una decisión usamos la clase...**

    a. ... *Activity.*
    b. ... *Dialog.*
    c. ... *Toast.*
    d. ... *Modal.*

2. **Dentro de los distintos objetivos que mencionamos a continuación sobre el componente *Dialog* dentro de nuestras aplicaciones, hay uno que no es cierto. ¿Cuál es?**

    a. Solicitar confirmación del usuario.
    b. Elegir una opción sobre una información.
    c. Mostrar una información al consumidor de la aplicación.
    d. Mostrar un mensaje al usuario sin necesidad de interacción.

3. **Determina si la siguiente oración es verdadera o falsa: "Para realizar la captura de la confirmación de un diálogo debemos hacer uso del método *setNegativeButton".***

    - Verdadero
    - **Falso**

4. **Si queremos mostrar un diálogo que cuente con hasta tres botones, ¿de qué clase debemos hacer uso?**

    a. *AlertDialog.Builder*
    b. *DialogInterface*
    c. *DialogFragment*
    d. *TimePickerDialog*

5. Si deseamos personalizar un dialogo con el que nos encontremos, deberemos personalizar este a través del uso del tipo de elemento:

a. *AlertDialog*
b. *Fragment*
c. *LayoutInflater*
d. *DialogFragment*

6. Determina si la siguiente oración es verdadera o falsa: "El ciclo de vida de un *DialogFragment* es parecido al de un *Fragment* común, aun teniendo eventos propios como *onCreateDialog*().".

- **Verdadero**
- Falso

7. El método *setPositiveButton,* del que hacemos uso para confirmar un diálogo, es el encargado de:

a. Establecer el nombre del botón y un *listener.*
b. Establecer el evento que este desatará.
c. Establecer la extensión de la clase *DialogFragment.*
d. Establecer el título que va a tener mi alerta.

8. Cuando queremos realizar la comunicación entre una actividad y un diálogo, ¿qué recurso pone a nuestra disposición el lenguaje de programación Java?

a. Las actividades.
b. Las interfaces.
c. Las enumeraciones.
d. La polimorfia.

9. Para realizar el lanzamiento de un *Toast,* se llamará a su método *makeText,* que no aceptará como parámetro de entrada alguno de los siguientes:

a. El tamaño del aviso.
b. El contexto en el cual nos encontremos.
c. La duración que ha de tener este.
d. El texto que mostrará.

10. Determina si la siguiente oración es verdadera o falsa: "La clase *Toast* en *Android* sirve para mostrar un mensaje breve al usuario con el cual no tiene que interactuar".

- **Verdadero**
- Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 9

1. **Los tipos de menús de *Android* coinciden en la creación de un fichero de tipo:**

   a. HTML
   b. XML
   c. JAVA
   d. VIEW

2. **El SDK de *Android* comenzó a ofrecer un botón por defecto con la única función de convocar el menú a partir de que versión:**

   a. *Android 2.0*
   b. *Android 4.3*
   c. *Android 8.0*
   d. *Android 3.0*

3. **Determina si la siguiente oración es verdadera o falsa: "Los menús en *Android* no solo tienen una funcionalidad necesaria, sino que también sirven para proporcionar una experiencia de usuario reconocida y uniforme".**

   ■ **Verdadero**
   ■ Falso

4. **Cuando queremos organizar los diferentes menús en conjuntos con características comunes, usamos la etiqueta:**

   a. <group>
   b. <item>
   c. < showAsAction >
   d. < menu >

5. **Si necesitamos crear un menú que abarque al marco contextual global de nuestra aplicación, lo más correcto sería crear un menú:**

   a. Emergente.
   b. Contextual.

c. Personalizado.
d. De opciones.

6. Determina si la siguiente oración es verdadera o falsa: "En cada versión de *Android* cambia el diseño de los menús y, por tanto, cambia también la semántica para definirlos".

- Verdadero
- **Falso**

7. Cuando queramos realizar un menú de opciones, deberemos mostrar la barra superior de la aplicación. Para que esto suceda, no debemos olvidar:

a. Extender la actividad principal de *AppCompactActivity.*
b. La creación de un menú contextual.
c. La creación de un modo de acción contextual.
d. Guardar los recursos XML que creemos.

8. Si queremos llevar a cabo la realización de un modo de acción contextual, deberemos hacerlo en versiones la API de *Android:*

a. Anteriores al API 14.
b. Posteriores a la API 11.
c. Anteriores a la API 11.
d. Anteriores a la API 5.

9. Los componentes WebView que nos ofrece *Android* se utilizarán para:

a. Mostrar contenido web dentro de la aplicación.
b. Saltar del contenido de la aplicación a contenido web.
c. La adquisición de elementos de origen web para nuestra aplicación.
d. Todas las opciones son incorrectas.

10. Determina si la siguiente oración es verdadera o falsa: "Los *WebViews* puede ser considerados como los *firewall* de nuestras aplicaciones".

- **Verdadero**
- Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 10

1. **Cuando queramos realizar el almacenamiento de información en la memoria interna del teléfono, deberemos añadir el tipo de seguridad que tendrá nuestro fichero. El más recomendado es:**

    a. Context.MODE_WRITABLE
    b. Context.MODE_PRIVATE
    c. Context.MODE_READABLE
    d. Context.MODE_WORLD_READABLE

2. **Si estamos usando la clase *PreferenceFragmentCompat,* estaremos haciendo referencia a la implementación de la opción de almacenamiento siguiente:**

    a. Memoria externa.
    b. Preferencias de compartición.
    c. Conexión a la red.
    d. Preferencias de usuario.

3. **Determina si la siguiente oración es verdadera o falsa: "Haremos uso del método *getExternalStorageStatus* cuando queramos guardar alguna información y deseemos comprobar si el flujo de información está disponible para escritura".**

    - **Verdadero**
    - Falso

4. **Para poder hacer uso de la clase ConnectivityManager, debemos siempre recordar añadir el permiso correspondiente en...**

    a. ... el fichero de manifiesto de nuestra aplicación.
    b. ... la View de la cual extienda nuestra interfaz.
    c. ... la llamada al método onCreate.
    d. ... la configuración build.gradle de nuestra aplicación.

5. Si la clase en la cual nos encontremos quiere lanzar un WebView, debemos recordar añadir a nuestro archivo de manifiesto los permisos:

a. READ_EXTERNAL_STORAGE y ACCESS_NETWORK_STATE.
b. READ_EXTERNAL_STORAGE.
c. ACCESS_NETWORK_STATE.
d. INTERNET y ACCESS_NETWORK_STATE.

6. Determina si la siguiente oración es verdadera o falsa: "No tenemos manera, dentro de las clases de programación que *Android SDK* nos proporciona, de saber el estado en el cual se encuentra la conexión a la red del dispositivo".

- Verdadero
- **Falso**

7. Deseamos guardar una cantidad de datos de preferencias de nuestra aplicación. Al principio de desarrollar nuestra aplicación comenzamos a realizarla con la clase *SharedPreferences,* pero actualmente la cantidad de datos que deseamos guardar es demasiado amplia. Nos convendría cambiar al siguiente sistema de almacenamiento:

a. Memoria externa.
b. Conexión de red.
c. Base de datos SQLite.
d. Preferencias de usuario.

8. ¿En qué casos será más conveniente que realicemos la grabación del estado de una actividad?

a. Si nuestra aplicación va a mostrarse a una disposición de pantalla.
b. En caso de que sea destruida y posteriormente vuelta a crear.
c. Creador de diseños.
d. Todas las opciones son correctas.

9. Si disponemos de una aplicación en la cual tenemos distintos campos a rellenar y deseamos que nuestro formulario se guarde al destruirse la aplicación, lo haremos a través de:

a. onCreate().
b. onStop() u onPause().
c. onSaveInstanceState().
d. onDestroy().

10. Determina si la siguiente oración es verdadera o falsa: "Recogeremos la información sobre el estado en el que se encontraba la actividad a través de un objeto de tipo Bundle".

- **Verdadero**
- Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 11

1. **Dentro de las opciones de almacenamiento que *Android* nos ofrece, elegiremos implementar una base de datos con *SQLite* por varias razones. Indica cuál de las siguientes ventajas no sería una de ellas:**

   a. No necesita hacer uso de un servidor externo para funcionar.
   b. Es un gestor de bases de datos de tipo cliente-servidor.
   c. Es una manera segura de almacenar los datos de nuestra aplicación.
   d. El código del proyecto es de uso libre.

2. **Cuando queremos realizar la creación de diferentes bases de datos y organizar la estructura que tendrán esas tablas, hacemos uso de la interfaz:**

   a. Activity
   b. Tarea
   c. SQLiteOpenHelper
   d. BaseColumns

3. **Determina si la siguiente oración es verdadera o falsa: "La clase SQLiteOpenHelper será la encargada de realizar la conexión entre nuestra aplicación y nuestra base de datos".**

   - **Verdadero**
   - Falso

4. **La interfaz *BaseColumns* es usada, entre otras ventajas, porque implementa la devolución de dos datos en forma de constantes muy importantes en el manejo de nuestras tablas. Estos son:**

   a. _COUNT e _ID
   b. NOMBRE y DESCRIPCION
   c. NOMBRE e _ID
   d. _ID y DESCRIPCION

5. Para la inserción de datos dentro de nuestra base de datos, *SQLite* provee de dos métodos, *put* e *insert,* que serían sobrescritos durante el siguiente evento:

a. onRestar()
b. onStop()
c. onResume()
d. onCreate()

6. Determina si la siguiente oración es verdadera o falsa: "El método *compileStatement* es menos eficiente que el método *insert* para realizar la inserción de datos en nuestras bases de datos".

- Verdadero
- **Falso**

7. Para poder poner a disposición los diferentes datos que proporciona nuestra aplicación de manera privada, y ponerlos al servicio del resto de componentes que deseen acceder a él, debemos hacer uso de la clase de *Android:*

a. ContenProvider
b. SQLiteOpenHelper
c. ContentResolver
d. BaseColumns

8. Dentro del ciclo de vida de un *ContentProvider,* cuando deseamos que este componente se destruya, debemos esperar a que sea consumido por una clase:

a. Activity
b. ContentResolver
c. ContentProvider
d. BaseColumns

9. Si deseamos actualizar la versión de nuestras tablas, lo realizaremos desde el método:

a. insert()
b. update()

c. delete()
d. query()

10. Determina si la siguiente oración es verdadera o falsa: "La parte imprescindible para establecer la comunicación entre un *ContentProvider* y un *ContentResolver* es el establecimiento correcto de los diferentes permisos a través de sus respectivos *AndroidManifest.xml*".

- **Verdadero**
- Falso

Ejercicios de autoevaluación Unidad de Aprendizaje 12

1. **De los siguientes tipos de avisos que nos proporciona *Android,* ¿cuál decidiremos desarrollar en caso de que queramos que esta se mantenga, aunque la ejecución de la aplicación haya terminado?**

   a. Diálogo.
   b. Notificación.
   c. *Toast.*
   d. Diálogo personalizado.

2. **Si deseamos identificar una de las siguientes partes como no perteneciente a una notificación *Android,* ¿qué opción deberemos elegir?**

   a. *SmallIcon*
   b. Texto
   c. Fecha
   d. Preferencias

3. **Determina si la siguiente oración es verdadera o falsa: "Haremos uso de la clase *NotificationCompat* cuando queramos construir una notificación dentro de *Android".***

   - **Verdadero**
   - Falso

4. **Para poder hacer uso de la clase *NotificationCompat,* debemos siempre recordar añadir su subclase...**

   a. ... *Builder.*
   b. ... *Notification.*
   c. ... *Action.*
   d. ... *Extender.*

5. Si deseamos lanzar una notificación que anteriormente hemos creado con *NotificationCompat,* debemos hacer uso de la clase auxiliar de Android siguiente:

a. *Builder*
b. *Notification*
c. *NotificationManager*
d. *NotificationBuilder*

6. Determina si la siguiente oración es verdadera o falsa: "Cuando creamos una instancia de la clase *NotificationManager* y pedimos al sistema el servicio *NOTIFICATION_SERVICE,* estamos pidiendo el servicio de nuestra propia aplicación".

- Verdadero
- **Falso**

7. Hemos establecido a través del uso de *NotificationCompat.Builder* la cualidad de que una notificación sea cerrada y eliminada en el momento en el cual el usuario pulse sobre una notificación. Esta funcionalidad se aporta a través del establecimiento a true de la propiedad siguiente:

a. *SmallIcon*
b. *ContentText*
c. *AutoCancel*
d. *ContentTitle*

8. La clase *NotificationCompat.DecoratedCustomViewStyle,* ¿en qué casos será más conveniente que realicemos su uso?

a. Si nuestra aplicación se ve en distintas disposiciones de pantalla.
b. En caso de que sea necesario crear un estilo personalizado.
c. Cuando la notificación solo se muestre una vez.
d. Cuando queramos personalizar una imagen.

9. Si queremos realizar una acción que debe guardarse para ejecutarse más adelante dentro de nuestro sistema, para ser lanzado ante un evento que hayamos definido, deberemos hacer uso de la clase:

a. *Intent*
b. *NotificationManger*
c. *PendingIntent*
d. *NotificationCompat*

10. Determina si la siguiente oración es verdadera o falsa: "Si deseamos realizar varios objetos *PendingIntent* al mismo tiempo, deberemos dotar estos de claves diferenciadoras para que se ejecuten por separado".

- **Verdadero**
- Falso