IMPORTANTE: LEER LAS NOTAS AL FINAL DE ESTE DOCUMENTO.

1. ¿Qué es Android?
2. ¿Qué es un kernel y cuál es su función?
3. ¿A qué se refiere que Android está basado en Linux?
4. ¿Quién fue el creador de Android?
5. ¿Cuál es la peculiaridad de los nombres de Android?
6. ¿Cuál fue el primer teléfono con Android, dónde se desarrolló y cuántos se vendieron?
7. ¿Cuál es el *easter egg* en las últimas 3 versiones de Android?
8. Realiza una tabla con el nombre de cada una de las versiones de Android, su fecha de lanzamiento y las características más importantes de cada una.
9. Escribe el nombre de al menos 20 dispositivos que usen Android.
10. Menciona 5 diferencias que existan entre Android y iOS.
11. Escribe el nombre de al menos 4 sistemas operativos hechos para dispositivos móviles.
12. Describe el concepto "Write once, run everywhere" y cuáles son sus implicaciones en el desarrollo de aplicaciones.
13. Define ampliamente (más de un párrafo) los siguientes conceptos.
    a. Código fuente
    b. Compilador
    c. Intérprete
    d. Máquina virtual
    e. Código máquina
    f. Bytecode
14. Describe el proceso de compilación de un programa de Java.
15. Explica detalladamente la arquitectura de Android.
16. ¿Existe alguna diferencia en compilar un programa cualquiera en Java a compilar una aplicación para Android? En caso de contestar afirmativamente, explicar cuál es.

17. Describe detalladamente el concepto "Ahead of Time compilation (AoT)". ¿Desde qué versión de Android se utiliza?
18. ¿Qué es el Android Runtime (ART)?
19. ¿Qué fue Dalvik?
20. Define el concepto "Just in Time (JIT) compilation".
21. ¿Qué diferencias hay entre ART y Dalvik?
22. Android 2.2 "Froyo" introdujo una mejora a la JIT llamada "Trace-based Just in Time compilation". Explica brevemente el concepto y por qué se considera una mejora.
23. ¿Todas las bibliotecas estándar de Java están disponibles en Android? En caso de contestar que no, dar algunos ejemplos de qué bibliotecas no están en Android.
24. ¿Qué es una actividad (activity) en Android? Explica brevemente la función de la clase 'Activity'.
25. ¿Qué es la clase 'R' en Android?
26. ¿Qué es un modificador de acceso?
27. ¿Qué es la cohesión y el acoplamiento (coupling)? Utiliza las definiciones que se usan en ingeniería de software.
28. Declarar atributos de una clase como públicos provoca acoplamiento fuerte (tight-coupling). ¿Verdadero o falso? ¿Por qué?
29. ¿Qué es una interfaz en Java?
30. ¿Para qué sirven las interfaces?
31. ¿Qué son las anotaciones y para qué sirven? Nota: Para que no se confundan, `@Override` es un ejemplo de una anotación.
32. Da ejemplos de anotaciones utilizadas comúnmente en Android y explica brevemente su función.
33. Instalar Android Studio 3 en el sistema operativo de su preferencia. Hacer un manual a computadora sobre cómo instalarlo. El manual deberá subirse a tu carpeta en GitHub (detalles al final de esta tarea) con el nombre "`01_manual.pdf`". Aceptaré el *pull request* a más tardar 15 minutos después de haber comenzado la clase.

34. Investiga la historia de Android hasta la actualidad. Se breve y conciso, pero no olvides mencionar fechas, nombres, mascotas y compañías. Longitud mínima: 2 páginas, longitud máxima 3 hojas.
35. ¿Qué es Android Developers? ¿Cuál es su importancia?
36. Define detalladamente los siguientes conceptos:
    i. API (Application Programming Interface)
    ii. SDK (Software Development Kit)
    iii. Framework
    iv. IDE (Integrated Development Environment)
37. ¿Qué es Android Studio?
38. ¿Por qué es necesario instalar el JDK para utilizar Android Studio?
39. ¿Cuáles son las ventajas y desventajas de usar Android Studio? ¿Es posible usar otro IDE? ¿Cuál?
40. ¿Qué es el SDK de Android, qué es nivel de API y cómo se relacionan estos con la versión de Android?
41. ¿Qué es el Android SDK Manager?
42. ¿Qué lenguajes de programación o de etiquetas utiliza Android para el desarrollo de sus aplicaciones? ¿Cuál o cuáles recomendarías utilizar y por qué?
43. ¿Qué es un paradigma de programación?
44. Define el paradigma de programación orientado a eventos.
45. ¿Qué es un patrón de diseño? ¿Por qué son importantes patrones de diseño para el desarrollo de software?
46. Explica detalladamente el patrón de diseño MVC (Model-View-Controller) y su funcionamiento. ¿Qué tiene que ver con Android?
47. Investiga y describe otros patrones de diseño que se utilicen en la actualidad. Por lo menos investigar 3 más.
48. ¿Cuál es la estructura de un proyecto de Android? ¿Qué contienen y cuál es la función de los directorios "manifests", "java" y "res" de un proyecto de Android?
49. ¿Qué es GUI (Graphical User Interface)?

50. ¿Qué es un widget en Android? Nota: me refiero a Widget como subclase de View, no confundir con App Widget.
51. ¿Qué es un contenedor (layout) en Android? Menciona por lo menos 3 ejemplos de ellos y explica brevemente cómo funcionan y cómo modifican la GUI.
52. ¿Qué es un Android Virtual Device (AVD)?
53. ¿Qué son las opciones de desarrollador de Android y cómo se activan?
54. ¿Qué es un emulador y cuáles existen para Android? ¿Para qué me sirve, como desarrollador, un emulador de Android? ¿Como desarrollador, es preferible utilizar un emulador o un dispositivo físico?
55. ¿Qué son los archivos .JAR, .DEX, .APK y cómo se relacionan entre ellos?
56. ¿Qué son Gradle y Maven Central y para qué los utiliza Android Studio?
57. Basándose en la respuesta de la pregunta 40, instala la versión del SDK y cualquier componente extra que necesites para probar tus aplicaciones en un dispositivo físico o emulado.
58. Describe detalladamente los pasos que se necesitan seguir para activar el modo de desarrollador en un dispositivo con Android.
59. Crear un AVD. Si cuentas con un dispositivo móvil con el cual probar tus aplicaciones, crea un AVD con una versión distinta de Android a la de tu dispositivo.

60. Hacer el "Hola mundo" de Android. Mostrarás tu aplicación cuando pases a entregar tu tarea.

PREGUNTAS EXTRA (NO CUENTAN PARA LAS REFERENCIAS NI PARA LA CALIFICACIÓN DE LA TAREA, PERO SE TOMARÁN COMO PUNTOS EXTRA PARA LA CALIFICACIÓN FINAL DE SU CURSO)

61. Si sabes dibujar o tienes la habilidad suficiente, hacer UN fanart de algún personaje de las siguientes series: “Gochuumon wa usagi desu ka”, “Shoujo Shuumatsu Ryokou”, “Urara Meirochou”. Si el dibujo es bueno, podrás tener de 1 a 2 puntos extras sobre tu calificación.

62. Dibuja el logotipo de Android con Crayolas (Sé creativo; el dibujo más abstracto tendrá un premio).
63. Escribe tu nombre en japonés utilizando katakana.
64. ¿Crees que eres feliz? Argumenta tu respuesta.
65. Escribe un breve comentario por cada uno de los cursos que has tomado como prebecario del PROTECO. ¿Qué te han parecido los instructores? ¿Cumplieron tus expectativas del curso? ¿Alguna recomendación que quisieras darles? Agrega cualquier otra cosa que quieras decir. Sólo yo leeré estos comentarios, mándalos al correo al final de este documento.

Notas

- **Todas las tareas requieren de bibliografía o lista de referencias por pregunta.**
  - **Si conocen la respuesta de alguna pregunta deberán dejarlo indicado en su lista.**
  - **Si utilizaron más de una fuente, deberán colocarlas en forma de lista.**
  - **Este apartado NO DEBERÁ SER ESCRITO A MANO. Es válido hacer su lista en un archivo de texto plano (.txt), en un archivo de Word (.docx) o en cualquier formato de archivo que pueda ser abierto fácilmente.**
  - **No es necesario que su lista siga un formato específico como APA. Es suficiente colocar el enlace a una página web o bien, si usaron un libro, colocar el título, autor y edición.**
- **Para este curso usaremos la forma de trabajo colaborativa basada en *pull requests* de GitHub, por lo que todos deberán tener una cuenta en esa plataforma.**
  - **Repositorio: `https://github.com/Angel5215/Android-Gen36`**
  - **Crearán un fork del repositorio `Android-Gen36` y desarrollarán sus tareas en el fork. Si la tarea es en equipos basta con que uno de ustedes la haga, pero no olviden documentar claramente quiénes son los miembros del equipo.**
  - **Los commits que realicen deberán hacerlos desde línea de comandos. Cualquier commit hecho con una aplicación gráfica no será tomado en cuenta y restará puntos a su calificación.**

- **Para su proyecto se tomará en cuenta que haya un avance visible reflejado en varios commits, por lo que deberán ir registrando el desarrollo de su proyecto.**
- **Cuando estén contentos con el desarrollo de alguna tarea o su proyecto deberán hacer un pull request desde la página de GitHub. Únicamente eso contará como la entrega de una tarea o proyecto.**

- **Para sus listas de referencias deberán crear una carpeta dentro de la carpeta `Referencias`. El nombre que tendrá esta carpeta es `AA_XXXX_YYYY`**
  - **`AA` corresponde a su número de prebecario.**
  - **`XXXX` corresponde a su apellido paterno escrito con minúsculas**
  - **`YYYY` corresponde a su apellido materno escrito con minúsculas**
  - **Dentro de esta carpeta colocarán el archivo de sus referencias. Deberá estar nombrado de la siguiente manera: `NN.EXT`**
    - **`NN` es el número de la tarea**
    - **`EXT` es la extensión de su archivo.**

Cualquier duda sobre su tarea, enviar un mensaje de correo a:

`angel.vazquez.proteco@gmail.com`