

Examen Final Práctico (A)

Alumno: _____ Hora inicio: _____ entrega: _____ Fecha: 12/1/2024

ADVERTENCIA: Puede consultarse cualquier tipo de información propia o disponible a través de Internet. No obstante, queda terminantemente prohibido la comunicación con cualquier persona durante la realización del examen. El simple hecho de tener abierta una pág. o aplicación de este tipo supondrá el suspenso inmediato del examen. También queda prohibido el uso de I.A. tipo Copilot o GTP. Tendrás que desactivarlas antes de iniciar el examen. No se permite usar aplicaciones que permita el envío de texto al móvil (como Google Docs). Las consultas solo se pueden hacer desde el PC, nunca desde el móvil. Este solo se podrá usar para verificar el funcionamiento de la aplicación.

DESCRIPCIÓN: Dispones de un máximo de 2 horas para realizar el mayor número de pasos y entregar el resultado en una tarea de Poliformat. Ver instrucciones de entrega al final.

IMPORTANTE: Inicia la captura con el programa de Proctoring antes de comenzar el examen.

0. Crea un nuevo proyecto con los siguientes datos (reemplaza **Apellido_Nombre** por tu nombre y **usuario_upv** por tu usuario de correo sin @)

Project: Phone and Tablet / Empty Views Activity
Name: Examen **Apellido_Nombre**
Package name: com.example.**usuario_upv**.examen

1. JAVA: Ha de funcionar, aunque se cambie las constantes de entrada. (1,5 pts, 20 min)

- a) Añade las siguientes constantes en la clase **MainActivity**. Representan los horarios y sueldos de varios controladores aéreos de AENA a lo largo de un turno de 24 horas. Rosa comienza el primer turno a las 00:00. Es sustituido por Juan a las 02:00, y así sucesivamente hasta el último turno, que lo hace Juan de 23:00 a 24:00. La lista **SUELDO** indica lo que cobrará cada trabajador por hora expresado en miles de €. (0.19 -> 190€).

```
static long FINAL = 24;  
static long INICIO[] = { 0, 2, 5, 9, 15, 20, 21, 23 };  
static String NOMBRE[] = { "Rosa", "Juan", "Rosa", "Pere", "Rosa", "Juan", "Pere", "Juan" };  
static double SUELDO[] = { 0.22, 0.19, 0.2, 0.18, 0.2, 0.19, 0.2, 0.3 };
```

- b) Crea una clase de tipo POJO con las propiedades de las listas anteriores. Ha de tener un constructor que inicialice todas las propiedades, el método **toString()** y un método que devuelva el valor de sueldo en euros, expresado en un entero.

- c) Crea un método dentro del POJO que reciba como parámetro las 3 listas que hemos creado. El método debe crear tantos objetos de la clase POJO, como elementos haya en la lista. El método devolverá una lista con los 8 objetos creados. Muestra el resultado en el **LogCat**.

2. ALGORITMO: Has de mostrar los resultados por el LogCat. (2 pts, 20 min)

- a) Crea una variable de tipo **Map<>** donde se almacene el sueldo por hora más bajo cobrado por cada trabajador. **SOLUCIÓN:** Juan=0.19, Rosa=0.2, Pere=0.18

- b) Calcula el coste laboral que tendrá que abonar AENA en esta jornada. Recuerda que el último turno acaba a las **24**. **SOLUCIÓN:** 4.78 ó 4780

CONEXIÓN A FIREBASE: Conecta la aplicación creada a un proyecto Firebase. Puedes crear un nuevo proyecto o conectar la aplicación a un proyecto previamente creado. (5 min)

3. ESCRITURA DE DATOS: (1,5 pts 15 min)

- a) Da acceso al proyecto de Firebase al usuario **jtomas00@gmail.com**.

- b) Crea una nueva colección utilizando Firestore, donde se almacene la lista con los 8 POJOs creados en el punto 1. Usa como clave (nombre de los documentos) valores enteros crecientes. 0 para el primer documento, 1 para el segundo y así sucesivamente.

4. RECICLERVIEW: Añade un `RecyclerView` en el layout. En el `RecyclerView` se mostrará toda la información de cada POJO almacenado. (2 pts 20 min)
5. M5STACK: Conéctate a un bróker MQTT. Suscríbete al token `examen/ApellidoN`. Reemplaza `Apellido` por tu primer apellido y `N` por la inicial de tu nombre. Cada vez que este token cambie su valor se mostrará en la pantalla del M5Stack. Puedes usar el bróker `broker.hivemq.com`. (1,5 pts 20 min)
6. LINUX: Conéctate por `ssh` a una máquina Linux en la dirección `65.20.96.154`. Utiliza tu usuario de UPV y como contraseña tu DNI (o NIE) sin la letra del final.
 - a) Dentro de la carpeta `home` de tu usuario, crea una carpeta cuyo nombre coincida con el de tu primer apellido sin acentos.
 - b) Copia en esta carpeta el fichero `AndroidManifest.xml` del proyecto Android creado en este examen. (1,5 pts 10 min)

Entrega de la práctica:

NOTA: Se bajará la nota si entregas fuera de hora o por no seguir las instrucciones.

7. Verifica que todo funciona correctamente antes de realizar el siguiente punto.
8. Selecciona la opción `Build / Build Bundel(s) APK(s) / Build APK`. Así generamos el apk.
9. Selecciona la opción `File / Export to Zip file...` para guardar el código del proyecto.
10. Para entregar usa la opción `Tareas` del menú de poli[Format]. Adjunta los ficheros:
 - apk, con el ejecutable
 - zip, con el código
11. Pega en el cuadro de texto el fragmento de código realizado en los puntos 1.JAVA solo c), 2. ALGORITMO y 5. M5STACK.
12. Envía la tarea.
13. Para facilitar la corrección en el cuadro de texto, que encontrarás al entrar en la entrega de la tarea, indica los ejercicios realizados de la siguiente lista introduciendo una X entre los paréntesis. Añade los comentarios oportunos:

 - ☐ 1.- JAVA
 - ☐ 2.- ALGORITMO
 - ☐ 3.- ESCRITURA DE DATOS
 - ☐ 4.- RECICLERVIEW
 - ☐ 5.- M5STACK
 - ☐ 6.- LINUX
14. Ejecuta el proyecto en un terminal o emulador, abre la consola de Firebase y un cliente MQTT. Avisa al profesor para que evalúe el examen.
15. Detén la captura en el programa de Proctoring.