|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PRUEBA DE EVALUACIÓN: CONOCIMIENTOS** | | | **PRUEBA PARCIAL TEÓRICA** | | | |
| Número de curso: IFC302LOE | | | | | | |
| Denominación: **TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA** | | | | | | |
| Horas: 35 | Modalidad: presencial | | Fecha inicio: 17/09/2024 | | Fecha fin:  24/06/2025 | |
| Centro: AFA FORMACION CONTINUA | | | | | **Prueba Tema 5** | |
| Código y denominación del módulo formativo:  **0489: PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES** | | | | | | |
| Profesor: Tomás Huerta Menéndez | | | | | | **CALIFICACIÓN** |
| Alumno/a: | | | | | |
| **Tiempo de realización:** 110 minutos | | | | | | |
| **Descripción de la prueba:** Prueba relativa al módulo 0489 Programación Multimedia y Dispositivos Móviles que evalúa los contenidos del tema 5. | | | | | | |
| **Sistema de calificación:**   * Preguntas para valorar los conocimientos. Cada ejercicio tiene señalado su valor tras el enunciado.   **Restricciones:**   * Si la base de datos no funciona de forma remota, la calificación será de 0. * Si las imágenes no se cargan de forma remota, la calificación será de 0. | | | | | | |
| **Instrucciones:**  La prueba consiste en tres ejercicios.  La prueba se superará con un 5.  Tiempo para la realización de la prueba: 110 minutos. | | | | | | |
| Firma del Profesor: | | Firma del Alumno/a: | | Fecha: | | |

**Objetivo**

El objetivo final es terminar la SecondActivity y crear una tercera en la que se muestren 4 imágenes subidas *a mano* a Cloudinary.

**Ejercicio 1. Rematar la SecondActivity (2 puntos)**

Añade tres botones en la SecondActivity:

**1.1. Primer botón (1 punto)**

Su texto es "Logout" y su función es borrar las preferencias guardadas (usando el editor), para después ir a la FirstActivity mediante un intent.

**1.2. Segundo botón (0.5 puntos)**

Su texto es "Exit" y cierra la aplicación de forma completa, empleando para ello:

finishAffinity();

System.exit(0);

**1.3. Tercer botón (0.5 puntos)**

Su texto es "Galería" y lleva a la ThirdActivity mediante un intent (sin extras ni nada más).

**Ejercicio 2. Cloudinary (1 punto)**

*A MANO* (sin programar): crea una carpeta de "assets" en Cloudinary con el nombre Apellido1\_Apellido2\_Nombre y sube las 4 imágenes adjuntas en la tarea.

Adjunta una captura de pantalla en la que se vea el nombre de la carpeta y las imágenes subidas.

**A screenshot of a computer

AI-generated content may be incorrect.**

**Ejercicio 3. ThirdActivity (7 puntos)**

El objetivo será crear una actividad con 4 ImageView cuyo contenido se cargará desde Cloudinary.

|  |  |
| --- | --- |
| **3.1. Crea una ThirdActivity (0.5 puntos)**  Crea una activity vacía, con un layout asociado.  **3.2. Modifica el layout (1.2 puntos)**  Incluye en el layout 4 ImageView con una distribución similar a la que se muestra a la derecha. Se recomienda nombrarlos de forma clara como imageView1, imageView2…  **3.3. Lógica de la activity (5.3 puntos)**  *3.3.1. Constantes y atributos (0.5 puntos)*  Incluye:  - Una constante estática de tipo String con el nombre de la carpeta que has hecho en el ejercicio 2.  - 4 atributos de tipo ImageView diferentes (puedes nombrarlos como imageView1, imageView2…) | Posible vista del xml |

*3.3.2. Método onCreate (0.2 puntos)*

Incluye llamadas a los métodos void:

- mapeaAtributosAVista.

- cargaImágenesCloudinary.

*3.3.3 Métodos adicionales (0.2 puntos)*

Añade los métodos:

- ejecutaEnHiloPrincipal

- setImagen. Refactoriza este último con un argumento adicional para poder añadirle a qué ImageView se le cambia el bitmap.

*3.3.4. Método mapeaAtributosAVista (0.4 puntos)*

Tiene el comportamiento *habitual*.

*3.3.5. Método cargaImágenesCloudinary (4 puntos)*

Su lógica es la siguiente:

* Crea una lista de ImageView y carga los 4 atributos.
* Crea un nuevo hilo (no te olvides de arrancarlo) que:
  + Obtenga los assets de la carpeta (usando la constante estática anterior)
  + Recorre cualquiera de estas listas (la de atributos o la de assets) por posiciones (no un for-each) y para cada elemento:
    - Saca el ImageView que corresponda con la posición.
    - Saca el public\_id que corresponda con la posición.
    - Genera la cadena de URL asociada al public\_id.
    - Crea el objeto URL asociado a la cadena.
    - Crea el bitmap asociado al objeto URL.
    - Usa el método refactorizado de setImagen para asociar el bitmap al atributo.

**Entrega**

En un archivo .zip adjunta:

· Este documento convertido a pdf y con tu nombre (y la captura pedida)

· La carpeta del proyecto de Android.