



Programming for mobile devices

2021-2022

Administrador de noticias

1 Introducción

El objetivo de este proyecto es desarrollar una aplicación cliente para visualizar noticias de un periódico. Esta aplicación tendrá las funcionalidades necesarias para leer / publicar / administrar artículos del periódico. En concreto, mostrar los artículos en la pantalla principal, mostrar los detalles de un artículo determinado, filtrarlos según su categoría, y además, poder actualizar el contenido y las imágenes de los artículos y recibir actualizaciones sobre los nuevos artículos publicados.

Este proyecto se implementará como una aplicación de Android en Java. La tarea debe desarrollarse en grupos de hasta 4 miembros.

En las siguientes secciones describiremos tanto las funcionalidades que se deben desarrollar como las características opcionales de la aplicación.

2 Componentes básicos

En esta sección, se explicará el comportamiento que debe tener la aplicación, así como la información básica que la aplicación cliente de Android debe poder administrar. En núcleo de la aplicación serán los artículos y las imágenes que existen dentro de cada artículo.

- Un artículo está compuesto por título, subtítulo, categoría, resumen y cuerpo.
- El título, categoría y resumen son obligatorios, es decir, cuando el usuario está llenando el formulario para agregar / modificar el artículo, debe ser validado por los diferentes formularios.
- Además, el servidor devuelve un campo `update_date` con la marca de tiempo de la última modificación del artículo dado.
- La imagen de los artículos se devuelve también como parte del objeto artículo, pero dependiendo de la solicitud, la imagen asociada a cada artículo se devuelve de dos formas diferentes:
 - cuando se solicita un artículo concreto, la imagen de tamaño completo se devuelve en los atributos: `image_data` y `image_media_type`
 - cuando se solicita una lista de artículos, se devuelve una miniatura de la imagen para cada artículo en los atributos: `thumbnail_data` y `thumbnail_media_type`
- Las categorías disponibles son: Nacional, Economía, Deportes y Tecnología.

Los datos relativos a los artículos se obtienen de los servicios REST desplegados en el servidor:

<https://sanger.dia.fi.upm.es/pmd-task/>

Las llamadas a los servicios deben modelarse estructurando el código de forma modular y cohesionada.

Deben crearse las clases de modelo Artículo e imagen en java.



3 Características obligatorias (hasta 5 puntos)

En esta sección enumeraremos las características obligatorias para la aplicación de Android.

3.1 Métodos para conexión a los servicios web (hasta 2,5 puntos)

Se crearán los métodos necesarios para hacer login y descargar los artículos e imágenes, así como para convertir el resultado recibido en json a objetos Java.

3.2 Pantalla principal (hasta 1,5 puntos)

La pantalla principal mostrará una lista de artículos disponibles en el servidor como usuario anónimo (ordenados por fecha), en esta pantalla principal una barra de navegación o menú lateral debe mostrar todas las opciones de categorías de noticias en el periódico, para poder filtrar el listado de una forma rápida. Las posibles categorías son: Nacional, Economía, Deportes, tecnología o 'Todas'.

- Todos los artículos de la página principal deben incluir:
 - su título,
 - su resumen (abstract),
 - su imagen en miniatura (si existe), y
 - la categoría a la que pertenece la noticia.

El cuerpo y el texto del subtítulo del artículo no se muestran en la página principal y se mostrarán en la página de detalles del artículo.

La lista se implementará usando un ListView o un RecyclerView, y para las filas de la lista se implementará un diseño personalizado. Cada elemento (artículo) de la lista debe estar vinculado (al seleccionar una noticia) a la página de detalles del artículo.

La pantalla principal opcionalmente incluirá un botón de inicio de sesión. (ver la parte de características opcionales). El usuario debe saber si está conectado (logado) o no en cada momento. Si el usuario está conectado, los artículos se pueden crear, editar y eliminar. De lo contrario, el usuario no podrá editar el contenido del periódico.

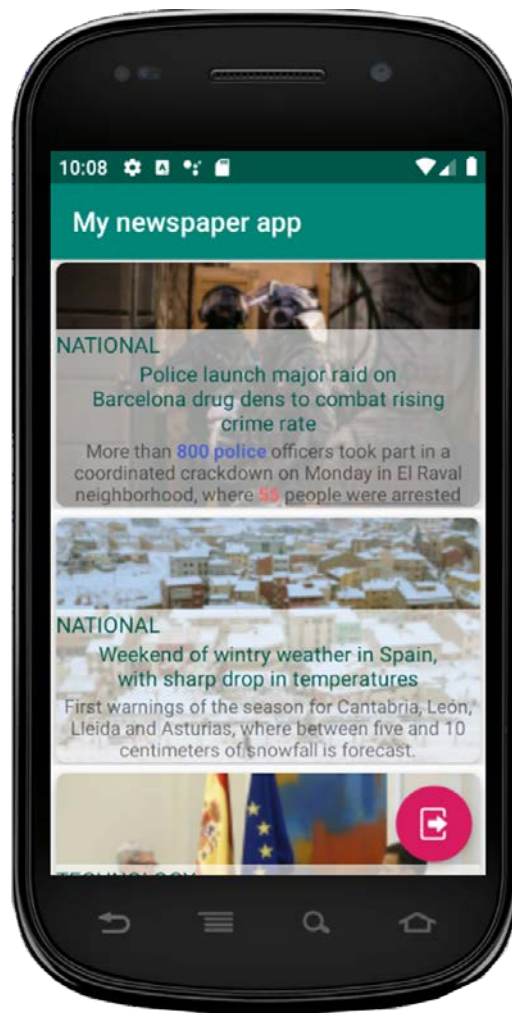


Figura 1. Ejemplo de pantalla principal de la aplicación

3.3 Pantalla de detalles de un artículo (hasta 1 punto)

- Los detalles del artículo deben mostrarse con su título, subtítulo, resumen, categoría, cuerpo e imagen (si tiene imagen).
- Además, se mostrará la fecha de modificación y el nombre de usuario que han modificado el contenido, bajo la imagen del artículo.
- Los artículos no se pueden modificar es esta pantalla.



4 Características opcionales

4.1 Formulario de acceso (login) (hasta 0,75 puntos)

Se pide agregar un formulario de inicio de sesión usando un nombre de usuario y contraseña. Para implementar el formulario de inicio de sesión, se debe agregar una nueva actividad de Android o un formulario. Si se implementa esta opción, el usuario podrá guardar la sesión (opción "recuérdame"). De esta manera, la clave API se almacenará en las preferencias de la aplicación, por lo que no será necesario iniciar sesión en las próximas ejecuciones de la aplicación.

En la pantalla principal, el botón de inicio de sesión debe cambiarse a un botón de "cerrar sesión" cuando el usuario ya ha iniciado sesión.

4.2 Uso de floating buttons en el interfaz (hasta 0,75 puntos)

La navegación por las categorías de noticias, así como el login se realizan mediante floating buttons.

4.3 Creación y edición de artículos (hasta 2,5 puntos)

Pantalla principal

- La aplicación seguirá el patrón CRUD típico, es decir, los usuarios pueden crear, leer, actualizar y eliminar artículos.
 - La aplicación no tiene que preocuparse por modificaciones simultáneas del mismo artículo. Si dos usuarios están editando el mismo artículo, la última modificación se guardará sobrescribiendo las modificaciones realizadas por el otro usuario.
- Si el usuario está registrado, los siguientes botones deben aparecer en la pantalla principal en cada noticia:
 - un botón por cada artículo para editarlo, este botón redirigirá al formulario de edición.
 - un botón para cada artículo para eliminarlo.
- La eliminación de artículos requiere la confirmación del usuario antes de aplicar la eliminación.
- La eliminación de artículos debe dar alguna retroalimentación al usuario con el resultado de la operación.
- Un botón para crear un nuevo artículo que redirige a un formulario de edición de artículo vacío.

Pantalla de edición y creación de artículos

- El formulario debe incluir todos los campos para editar / crear el artículo, que son: título, categoría, subtítulo, resumen, cuerpo y permitir la selección de imagen (de la galería).
- Las categorías deben seleccionarse en un campo limitado con los valores: Nacional, Economía, Deportes y Tecnología.
- Todas las entradas del formulario (excepto la imagen) son obligatorias y los valores del formulario deben estar validados.
- Este formulario debe incluir un botón para volver a la pantalla principal (cancelar) y otro botón para guardar / crear el artículo usando la información agregada por el usuario.
- Guardar información debe dar alguna retroalimentación al usuario con el resultado de la operación.
- Cuando se guarda el artículo, la aplicación volverá a la pantalla principal (actualizando la lista de artículos).



4.4 Notificaciones de nuevos artículos (hasta 2 puntos)

La aplicación debe implementar un sistema de sondeo (polling) para verificar si algún artículo se ha actualizado en el servidor. Este sondeo se implementará mediante un trabajo periódico de verificación de artículos a partir de una fecha. Esta fecha (de la última noticia obtenida) se usará para verificar las actualizaciones, y debe almacenarse para las próximas conexiones del servicio. En caso de existir artículos nuevos en el servidor, la aplicación debe mostrar una notificación indicando el número de artículos nuevos encontrados (incluyendo alguna imagen y/o descripción) y al presionar la notificación se abrirá la pantalla principal de la aplicación.

El servidor proporciona las últimas actualizaciones usando el método:

`https://sanger.dia.fi.upm.es/pmd-task/articlesFrom/yyyy-mm-dd hh:ii:ss`

La fecha debe usar el formato “yyyy-mm-dd hh:ii:ss”, el servidor devolverá como resultado de la ejecución del método una lista de artículos con fecha de actualización mayor a la fecha dada como.

4.5 Almacenamiento y navegación offline de las noticias e imágenes (hasta 2 puntos)

La aplicación utilizará una base de datos interna SQLite para almacenar la información descargada con el fin de reducir las conexiones al servidor. Todos los artículos e imágenes (con sus atributos) descargados se almacenan en esa base de datos SQLite.

La información se debe actualizar periódicamente de forma similar a la de la Sección 4.4, actualizando periódicamente las últimas noticias actualizadas y almacenándolas en la BBDD de la aplicación.

La vista de lista de la pantalla principal utiliza datos de la base de datos interna siempre.

5 Consideraciones finales

Se tendrá MUY en cuenta la modularidad y limpieza del código para la calificación de cada parte de la aplicación, así como los comentarios dentro del código.

Todas las constantes de cadena en la aplicación deben declararse como recursos del proyecto (strings.xml), para facilitar la traducción a otros idiomas de la aplicación.

Todos los miembros del grupo deben participar en todas las partes del desarrollo; así serán capaces de contestar las preguntas realizadas durante la presentación del proyecto.

6 Entrega

La fecha límite del proyecto será el 16 de diciembre de 2021. Para ello, se creará una tarea en el moodle de la asignatura que estará disponible hasta las 23:59 horas del 16 de diciembre de 2021.

El envío debe incluir un archivo comprimido (zip, rar o 7z) que contenga la carpeta main del proyecto (conteniendo el código fuente y los recursos del proyecto de Android), así como el fichero de configuración de gradle del módulo (que contiene las dependencias). No se deben incluir los binarios ni el archivo apk generado.



Además, se deberá entregar el fichero (PPT o PDF) que se usará en la presentación que se realizará el día 17 de diciembre en horario de clase. La presentación deberá durar un máximo de 10 minutos y cada grupo tiene que explicar qué funcionalidades ha implementado y cómo la han hecho (sin necesidad de mostrar código en la presentación).

7 Anexo

Se recomienda crear una clase de utilidad para serializar y deserializar las imágenes desde Android para su comunicación con el servidor:

El código necesario para convertir las imágenes que vienen como cadenas de texto en base64 es el siguiente:

```
@RequiresApi(api = Build.VERSION_CODES.FROYO)
public static Bitmap base64StringToImg(String input)
{
    try {
        byte[] decodedBytes = Base64.decode(input, Base64.NO_WRAP);
        return BitmapFactory.decodeByteArray(decodedBytes, 0,
decodedBytes.length);
    } catch (Exception e) {
        e.getMessage();
        return null;
    }
}
```

Para convertir un Bitmap a una cadena de texto en base64:

```
@RequiresApi(api = Build.VERSION_CODES.FROYO)
public static String imgToBase64String(Bitmap image)
{
    Bitmap.CompressFormat compressFormat = Bitmap.CompressFormat.PNG;
    int quality = 100;
    ByteArrayOutputStream byteArrayOS = new ByteArrayOutputStream();
    image.compress(compressFormat, quality, byteArrayOS);
    return Base64.encodeToString(byteArrayOS.toByteArray(), Base64.NO_WRAP);
}
```

Importando las clases necesarias:

```
import androidx.annotation.RequiresApi;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Base64;
import java.io.ByteArrayOutputStream;
```