

## Ejercicio 3 - Web Services

Detalles de la tarea de esta unidad.

### Enunciado.

#### Ejercicio 3: Retrofit y Picasso para acceder a imágenes de servicios web



Crea una aplicación que se conecte mediante Retrofit al servicio web con el que la NASA ofrece imágenes de Marte publicado en esta URL de ejemplo

🔍 Google Developers (Uso educativo nc)

[https://api.nasa.gov/mars-photos/v1/rovers/curiosity/photos?sol=1000&api\\_key=DEMO\\_KEY](https://api.nasa.gov/mars-photos/v1/rovers/curiosity/photos?sol=1000&api_key=DEMO_KEY)

Este servicio devuelve, con el parámetro indicado en la URL sol=1000 y una clave de API (api\_key), un array JSON (que puedes manejar con el convertidor GSON) con información de numerosas imágenes (actualmente 855 imágenes) incluyendo la URL de la misma y que tiene el aspecto mostrado en la primera figura.

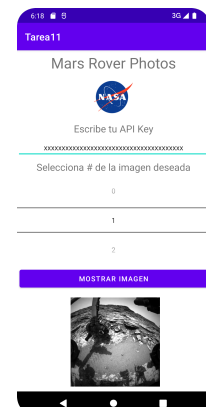
La aplicación que debes desarrollar solicitará al usuario dos parámetros y mostrará en la parte inferior la imagen mediante Picasso:

- Tu App Key que debes conseguir registrando tu correo electrónico en esta web <https://api.nasa.gov/>
- El número de imagen (entre 0 y 855) que el usuario elija.

El aspecto esperado de la aplicación al seleccionar la imagen en la posición 1 del array devuelto por el web service, es el mostrado en la captura de pantalla del móvil.

NOTA 1: Para que sea más sencilla la creación de las clases POJO usadas por Retrofit, puedes copiar desde un navegador como Firefox el texto JSON devuelto por el web service con la llamada de ejemplo indicada arriba, y mediante esta web <https://www.jsonschema2pojo.org/> (marcando JSON y GSON) ayudarte a crear la clase principal Curiosity (con la propiedad ArrayList<Photo>) y la clase Photo para definir la propiedad img\_src que es el que necesitamos para cargar la imagen con Picasso.

NOTA 2: Ten cuidado con las URLs de las imágenes porque el web service a día de hoy las devuelve con HTTP, lo cual genera un problema de seguridad. Puedes sustituir el



🔍 Google Developers (Uso educativo nc)

(#)

HTTP por HTTPS de forma programática con la función *replaceFirst()* o utilizar la clase Java URL para hacerlo.



### Criterios de puntuación. Total 2.5 puntos.

La valoración total de este ejercicio es de 2.5 puntos repartidos del siguiente modo:

- Aspecto: la aplicación se muestra de un modo similar al mostrado. 0.25 puntos.
- Funcionamiento de la aplicación:
  - ○ La aplicación se conecta y muestra la imagen seleccionada: 0.50 puntos.
  - ○ La aplicación usa correctamente Retrofit definiendo las clases POJO de forma adecuada: 1.5 puntos.
  - ○ La aplicación usa Picasso para mostrar la imagen: 0.25 puntos.

Puesto que no existe una evaluación por unidades la evaluación se realiza en base a los criterios generales del módulo.

### Recursos necesarios para realizar la Tarea.

- Ordenador con Android Studio instalado y suficientes recursos para ejecutar el emulador.
- Contenidos de la unidad, muy importantes los ejercicios resueltos de la misma.
- Páginas web de los desarrolladores de los sistemas operativos para móviles.

### Consejos y recomendaciones.

- Lee atentamente el enunciado y asegúrate de haber entendido lo que has de hacer. Especial atención a las notas al final del enunciado.
- **Intenta reproducir en tu Android Studio los ejercicios resueltos previamente y apóyate en los recursos de apoyo de los contenidos.**
- No dudes en comentarle a tu tutor o tutora cualquier duda que te pueda surgir.
- Envíasela a tu tutor o tutora a través del sistema establecido en la plataforma.
- Las capturas y el contenido de los ficheros deben aparecer en perfecto orden para que esté claro lo que deseas mostrar.
- Los diseños deben mostrarse lo más parecidos posibles a lo que se pide.
- Debe llevarse a cabo sobre una versión actual de Android.

(#)

### Indicaciones de entrega.

#### Capturas:

- Captura del aspecto de tu aplicación al arrancar la primera vez.
- Captura del aspecto de tu aplicación cuando cargues una de las imágenes.
- Captura con la estructura de los ficheros java y XML del proyecto.
- Las capturas que no sean de tu aplicación, sino de tu Android Studio, deben mostrar el fondo de pantalla y la fecha y hora de la barra de estado de tu sistema operativo para garantizar la originalidad del material.

#### Archivos:

- Ficheros importantes: clases java, archivo/s de layout, archivo de manifiesto y gradle (:app).

Obra publicada con Licencia Creative Commons Reconocimiento Compartir igual 4.0  
(<http://creativecommons.org/licenses/by-sa/4.0/>).