

Desafío - Mars Rover Images

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta y sube el `.zip`

Descripción

En el siguiente desafío se deberá construir una aplicación para dispositivos android, que al iniciar comience un proceso de descarga de cuatro imágenes del Mars Land Rover de la nasa, en dónde se deben mostrar cuatro progressBar y cuatro imageView dentro de un constraintLayout ordenados en cuatro secciones simétricas y centradas.

Al iniciar la aplicación se debe activar el progressBar uno, en su sección correspondiente (imagen uno), y acción seguida se debe ejecutar una corrutina que realice la descarga de la imagen uno a través de la url string dada. Al completarse la corrutina se debe mostrar la imagen descargada en el imageView número uno, posteriormente se debe deshabilitar la progressBar uno y activar la progressBar dos, para dar curso a la próxima corrutina que mostrará la imagen 2, y así sucesivamente hasta completar las cuatro tareas.

Adicionalmente se deberá crear un botón al final de la pantalla que muestre un AlertDialog con el siguiente mensaje: "Este mensaje demuestra el funcionamiento de las corrutinas en background" y debe ser cancelable. Esto para visualizar la acción en el thread UI de android mientras las corrutinas se ejecutan en segundo plano.

Instrucciones

1. Capturar las cuatro primeras imágenes del resultado de la api siguiente:

```
{
  - photos: [
    - {
      id: 102693,
      sol: 1000,
      - camera: {
        id: 20,
        name: "FHAZ",
        rover_id: 5,
        full_name: "Front Hazard Avoidance Camera"
      },
      img_src: "http://mars.jpl.nasa.gov/msl-raw-images/proj/msl/redops/ods/surface/sol/010
      earth_date: "2015-05-30",
      - rover: {
```

Nota: Se recomienda crear una variable final de tipo Array en la actividad para el trato de estas url de imagenes.

2. Crear las vistas requeridas para el funcionamiento de la aplicación. 4 ProgressBar, 4 ImageView y 1 Button, estos dentro de un constraintLayout bien alineados modelando un grid 2x2 .

Tip: Para los imageView se pueden utilizar los siguientes atributos para lograr modelar un grid 2x

```
android:layout_width="0dp"
android:layout_height="0dp"
app:layout_constraintDimensionRatio="H,1:1"
```

3. Implementar el botón junto a la acción de desplegar un AlertDialog con el mensaje solicitado y que se pueda cancelar, utilizando solo wrap_content para el ancho y largo de la vista. El texto del botón será "SHOW ALERT THREAD UI".
4. Se debe crear un método "downloadImages" que sea llamado en el método onStart() de la actividad y que realice las siguientes acciones:
 - Hacer visible la progressBar uno.
 - Recorrer un arreglo de strings que contiene los string de url de las imágenes, utilizando el método withContext de las corrutinas indicando que se desea ejecutar en un proceso background el método downloadImagesNasaRover(url) que devolverá un bitmap. Dentro de nuestro método downloadImages() se deberá validar si el bitmap no es nulo para posteriormente crear condicionales que permitan mostrar la imagen en el imageView correspondiente y también habilitar/deshabilitar el progressBar que corresponda.
5. Se ha de crear un método downloadImagesNasaRover(url) que recibe un string y devuelve un bitmap para la descarga de la imagen desde el api de la nasa indicado. Este método como ya se indicó ha de ser utilizado en el método downloadImages().

6. De manera opcional, solo si tienen el tiempo, realizar una descarga adicional siguiendo los mismos pasos anteriores, pero en esta ocasión utilizando promesas en vez de coroutines.

Nota**: agregar en el archivo manifest dentro del tag application el siguiente parámetro en caso de tener algún tipo de error asociado al tráfico de los datos:

```
android:usesCleartextTraffic="true"
```

Pasos

- 1) Crear proyecto android, api mínima 19, targetSdk 26.
- 2) Modificar diseño layout por defecto activity_main.xml para incluir las vistas requeridas.
- 3) Alinear las vistas dentro del ConstraintLayout para que estén centradas y donde se esperan.
- 4) Agregar acción del botón dentro del método onCreate() para que crear un AlertDialog que muestre los mensajes solicitados.
- 5) Crear el método downloadImages().
- 6) Crear el método downloadImagesNasaRover(url).
- 7) Agregar permisos de internet al manifest y parámetros que sean necesarios.
- 8) Compilar.
- 9) Correr aplicación en emulador o dispositivo android con usb.