



IMPLEMENTANDO BASES DE DATOS EN TIEMPO REAL

POR: ANDRES FELIPE RAMIREZ CAICEDO

Especialización de Android – Semana 3 entrega 1 –
Curso 4

CONTENIDO

DESCRIPCIÓN DE LOS ENDPOINTS	2
PLATAFORMA DE SERVIDOR PARA LA COMUNICACIÓN CON LA BASE DE DATOS Y LAS NOTIFICACIONES	4
PANTALLAZOS Y GENERALIDADES DE LA APLICACIÓN.....	5

DESCRIPCIÓN DE LOS ENDPOINTS

En la aplicación se utilizaron cuatro EndPoints, distribuidos en una clase; “getRecentMediaUserID”, “getUsuarioBusqueda”, “getUsuarioFollows” y “getRecentMediaTimeline”, basados en las correcciones de jose.petagramws y mi proyecto original ya que no poseía sincronía por lo tanto no cumplía con algunas especificaciones:

- getRecentMediaUserID: el cual se encarga de la manipulación de los datos del usuario que se liga a la cuenta, en el caso de la aplicación se usa para los datos que se visualizan en la Figura 2.

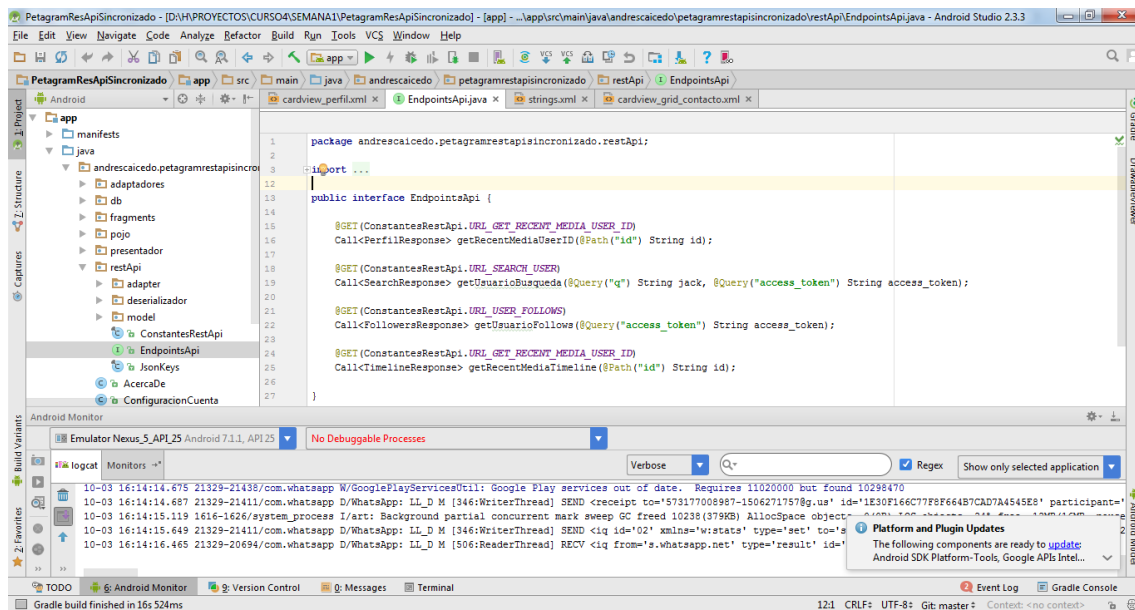


Figura 1. Primer EndPoint

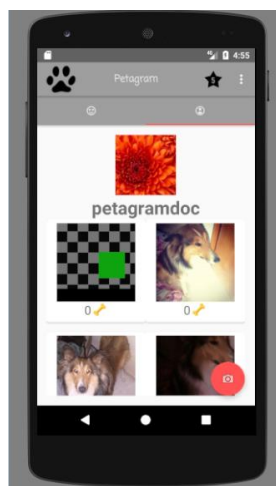


Figura 2. Datos del EndPoint getRecentMediaUserID

- El segundo EndPoint es “getUsuarioBusqueda”, el cual usa una URL para completar el String de conexión y poder manipular la información, en este caso poder manipularla

dependiendo de un ID tal y como se muestra en la Figura 3, esto con el fin de traer la información de un usuario en específico.

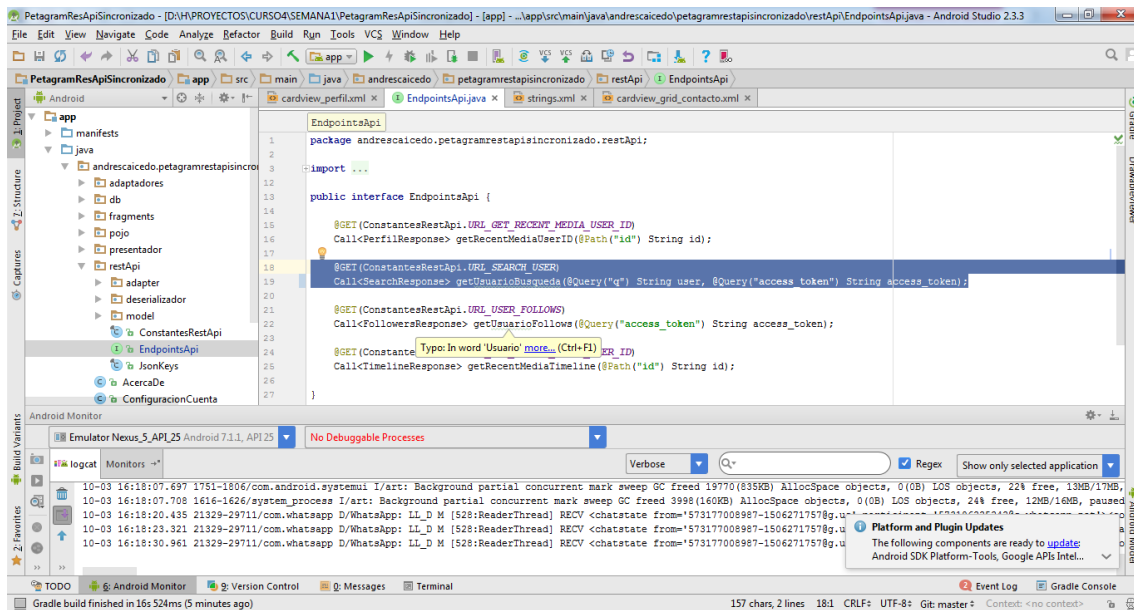


Figura 3. EndPoint con URL

El Endpoint getUsuarioFollows se encarga de traer los follows de la cuenta específica.

El EndPoint “getRecentMediaTimeline” se encarga de traer en un arreglo las fotos del usuario del cual ya tenemos el ID, tal y como se muestra en la Figura 4.

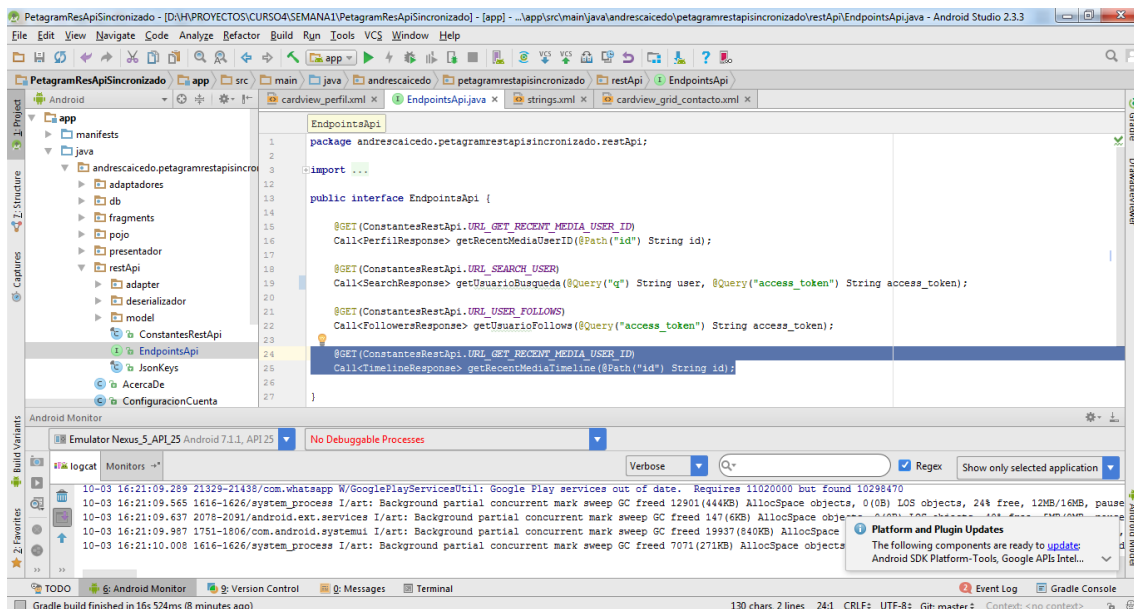


Figura 4. Antepenultimo EndPoint

El endPoint de registrar usuario se encarga de recolectar el id (token) y el id de usuario de instagram los cuales realizaran las acciones pertinentes con retrofit y las configuraciones de firebase realizadas.

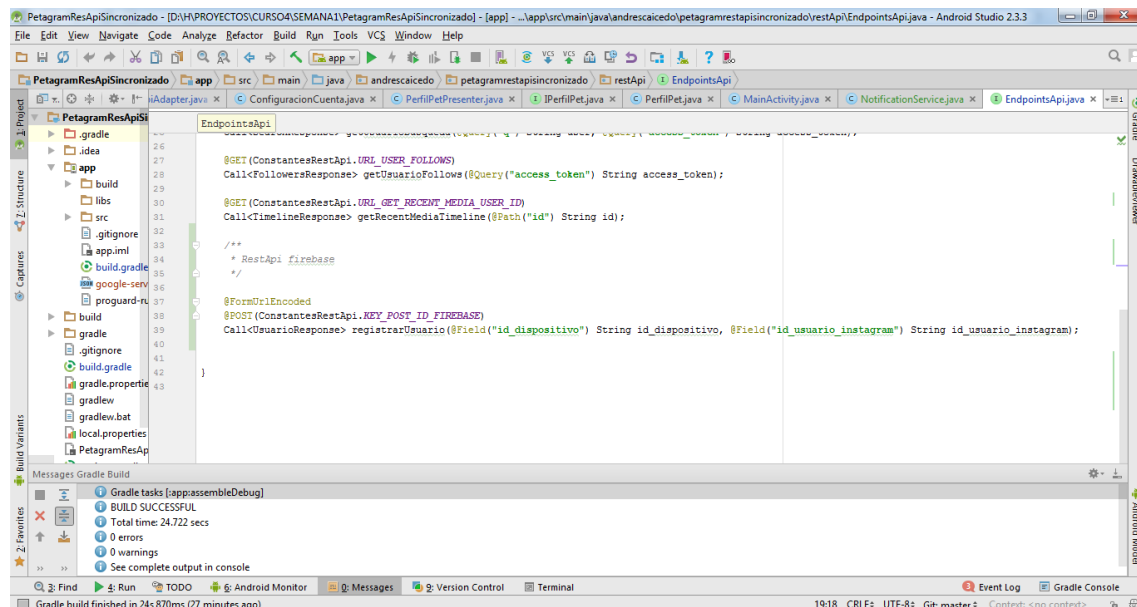
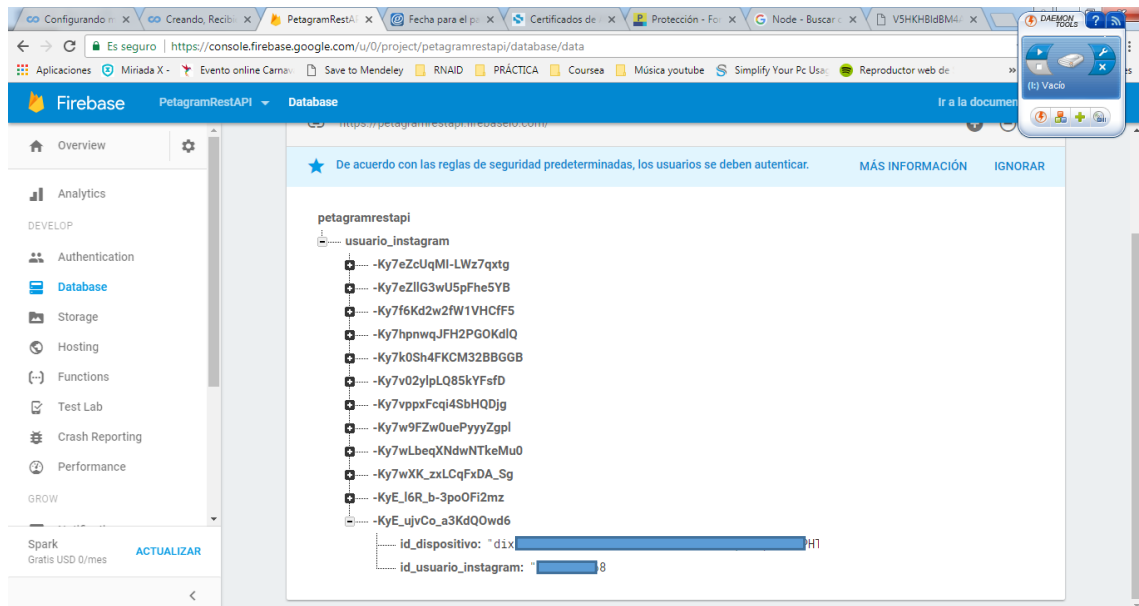


Figura 5. EndPoint registrarUsuario

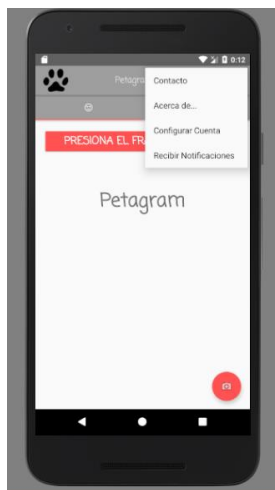
PLATAFORMA DE SERVIDOR PARA LA COMUNICACIÓN CON LA BASE DE DATOS Y LAS NOTIFICACIONES

Node.js, fue el entorno en tiempo de ejecución multiplataforma escogido para el servidor que se contruyo en heroku el cual contiene:

1. La librería firebase-admin puesto que firebase actualizo sus importaciones para estos tipos de proyectos y ya no se utilizan los que aparecen en el video de clase.
2. Se configuro var tokenDeviceURI="token-device", tal cual aparece en los videos de clase.
3. Se añadio var usuario_instagramURI="usuario_instagram", para manejar lo relacionado a la actividad realizada, se usa para registrar los usuarios, solicitando el id_dispositivo y el id_usuario_instagram para almacenarlo en la base de datos en tiempo real de firebase, tal y como se puede visualizar en la siguiente figura:



PANTALLAZOS Y GENERALIDADES DE LA APLICACIÓN



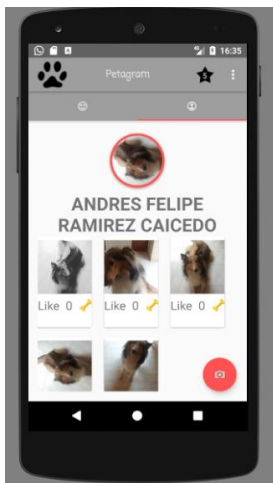
Recibir notificaciones añadido



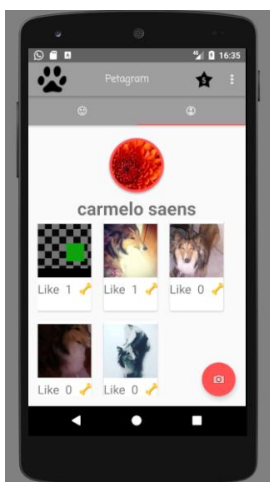
Id de usuario enviado, se muestra cuando se cambia de usuario pero se envía cuando se presiona en recibir notificación.



Interfaz inicial.



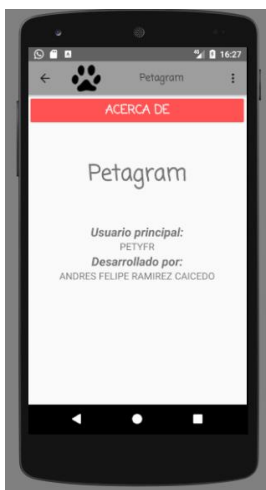
Apartado que muestra las fotos del contacto que le dio permiso a petagramdoc, que es petyfr



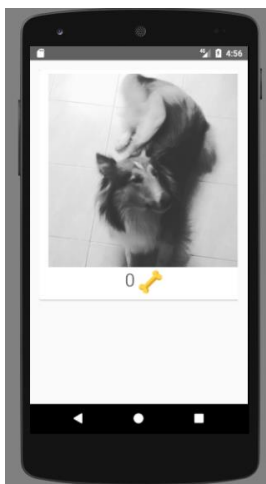
Perfil de petagramdoc, con sus fotos



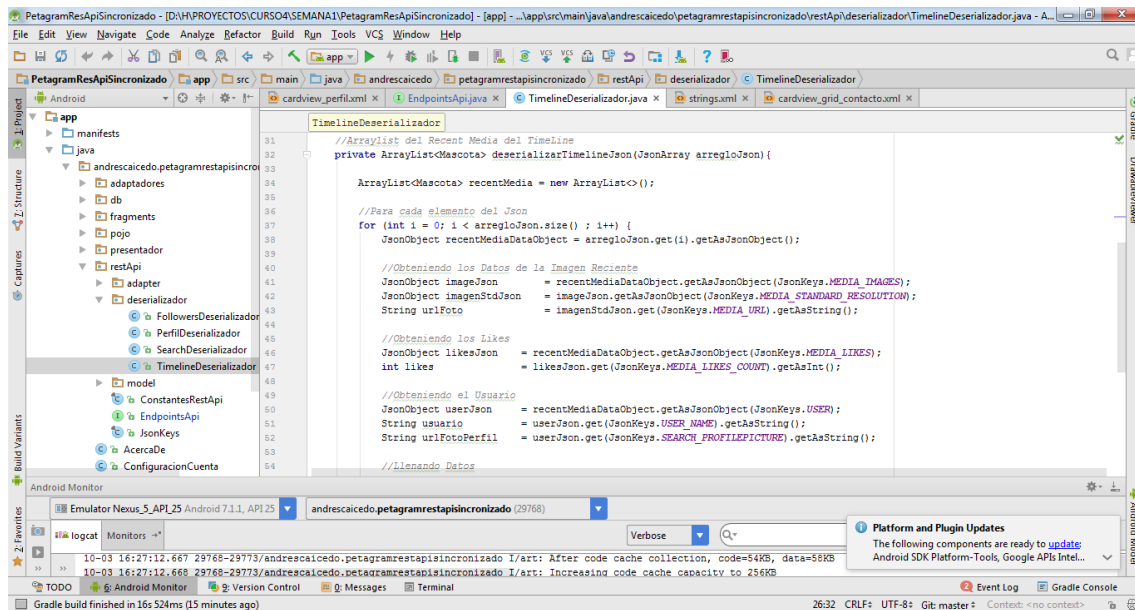
Guardado del perfil de petagramdoc



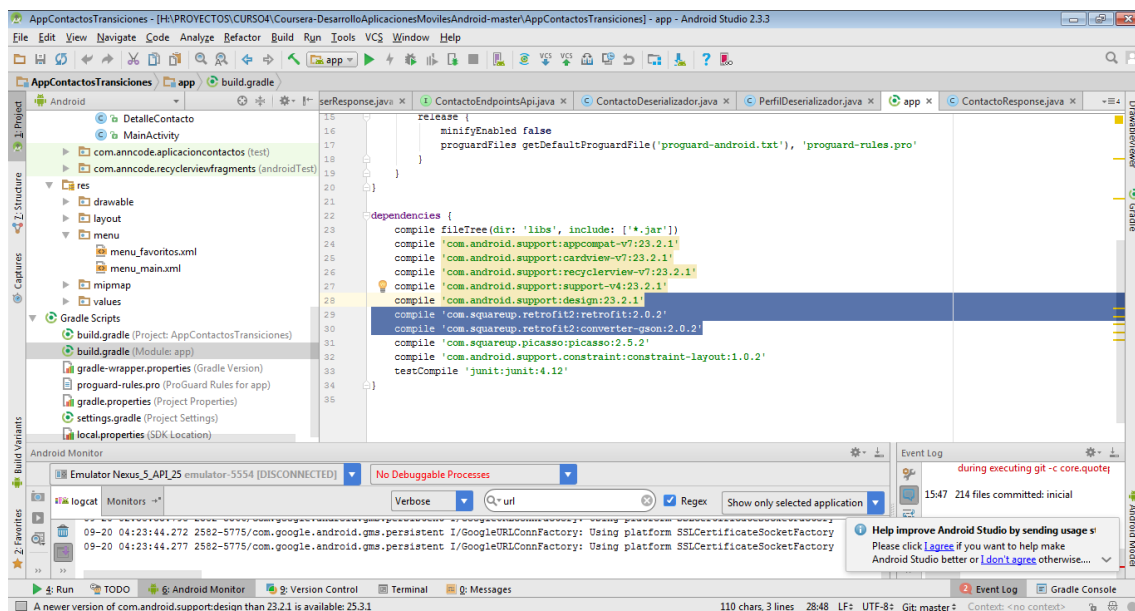
Información del desarrollador, que aparece en acerca de



Vista de una foto en general



Deserializadores y parte de las carpetas que demuestra la distribución del modelo, vista, presentador.



Implementación de la librería retrofit

URLs utilizadas:

- https://api.instagram.com/v1/users/self/follows?access_token=ACCESS-TOKEN
- https://api.instagram.com/v1/users/{user-id}/media/recent/?access_token=ACCESS-TOKEN
- https://api.instagram.com/v1/users/{user-id}/media/recent/?access_token=ACCESS-TOKEN
- https://api.instagram.com/v1/users/{user-id}/media/recent/?access_token=ACCESS-TOKEN
- https://quiet-shore-79332.herokuapp.com/usuario_instagram/