|  |  |  |  |
| --- | --- | --- | --- |
| **Alumno(s):** | Daniel Francisco Franco Ortiz | **Fecha:** |  |

|  |
| --- |
| OBJETIVO |
| Comprender la función de la librería Volley, sus métodos y su estructura mediante la creación de una aplicación móvil que pueda hacer consultas a una pagina web. |

|  |
| --- |
| MARCO TEÓRICO |
| Volley es una biblioteca HTTP que hace que las redes para las aplicaciones de Android sean más fáciles y, lo que es más importante, más rápidas.  Volley ofrece los siguientes beneficios:   * Programación automática de solicitudes de red. * Múltiples conexiones de red simultáneas. * Almacenamiento en memoria caché transparente de memoria y disco con coherencia de caché HTTP estándar. * Soporte para la priorización de solicitudes. * Solicitud de cancelación API. Puede cancelar una sola solicitud o puede establecer bloques o ámbitos de solicitudes para cancelar. * Facilidad de personalización, por ejemplo, para reintento y rebobinado. * Una ordenación sólida que facilita el llenado correcto de su IU con datos obtenidos de forma asíncrona de la red. * Herramientas de depuración y rastreo.   Volley sobresale en las operaciones de tipo RPC utilizadas para llenar una interfaz de usuario, como recuperar una página de resultados de búsqueda como datos estructurados. Se integra fácilmente con cualquier protocolo y sale de la caja con soporte para cadenas sin procesar, imágenes y JSON.  Volley no es adecuado para operaciones de descarga o transmisión de gran tamaño, ya que Volley mantiene todas las respuestas en la memoria durante el análisis.  Un RequestQueue necesita dos cosas para hacer su trabajo: una red para realizar el transporte de las solicitudes y un caché para manejar el almacenamiento en caché.  BasicNetwork es la implementación de red predeterminada de Volley. Una BasicNetwork debe inicializar con el cliente HTTP que usa su aplicación para conectarse a la red. Por lo general, esta es una HttpURLConnection .  Si la aplicación hace un uso constante de la red, probablemente sea más eficiente configurar una única instancia de RequestQueue que dure la vida útil de la aplicación.  El enfoque recomendado es implementar una clase singleton que encapsula RequestQueue y otras funcionalidades de Volley.  En un nivel alto, usa Volley creando una RequestQueue y pasándola Request objects. RequestQueue gestiona los hilos de trabajo para ejecutar las operaciones de red, leer y escribir en el caché y analizar las respuestas. Las solicitudes hacen el análisis sintáctico de las respuestas en bruto y Volley se encarga de enviar la respuesta analizada nuevamente al hilo principal para la entrega.  En un nivel alto, usa Volley creando una RequestQueue y pasándola Request objects. RequestQueue gestiona los hilos de trabajo para ejecutar las operaciones de red, leer y escribir en el caché y analizar las respuestas. Las solicitudes hacen el análisis sintáctico de las respuestas en bruto y Volley se encarga de enviar la respuesta analizada nuevamente al hilo principal para la entrega.  Volley siempre entrega respuestas analizadas en el hilo principal. Ejecutar en el hilo principal es conveniente para poblar los controles UI con los datos recibidos, ya que puede modificar libremente los controles UI directamente desde su controlador de respuestas, pero es especialmente crítico para muchas de las semánticas importantes proporcionadas por la biblioteca, particularmente relacionadas con la cancelación de solicitudes.  Para enviar una solicitud, simplemente construya una y añádala a RequestQueue con add(). Una vez que agrega la solicitud, se mueve a través de la canalización, se le da servicio y se analiza y entrega su respuesta en bruto.  Para cancelar una solicitud, llame a cancel() en su objeto Request . Una vez cancelado, Volley garantiza que nunca se llamará a su controlador de respuesta. Lo que esto significa en la práctica es que puede cancelar todas sus solicitudes pendientes en el método onStop() su actividad y no tiene que ensuciar sus controladores de respuesta con comprobaciones de getActivity() == null , si se ha llamado a onSaveInstanceState() ya, u otra repetición defensiva. |

|  |
| --- |
| DESARROLLO |
| *Primero se agrega el permiso para poder acceder a internet debido a que volley lo utilizara para conectarse a la página.*    *Para la utilización de volley se debe agregar la dependencia correspondiente el la clase app del Gradle.*    *Se define la URL de la pagina a la cual se accedera, se revisa si la imagen que se consultara estará a color o en blanco y negó, posteriormente se tomaran los valores de width y height para definir las dimensiones de la imagen.*  *Se crea un objeto RequestQueue, se declara el objeto que contendrá la imagen y se asignara al contenedor correspondiente, por ultimo se hace la petición a la pagina con el método add(), del RequestQueue.* |

|  |
| --- |
| CONCLUSIONES |
| Volley es una librería que permite hacer consultas web de manera fácil de entender y con una estructura simple, lo que permite una mayor velocidad de respuesta que al utilizar otros tipos de objetos.  Sin embargo, Volley no es recomendado para las consultas o descargas de archivos muy pesados. |

|  |
| --- |
| BIBLIOGRAFÍA |
| <https://developer.android.com/training/volley/index.html>  <https://developer.android.com/training/volley/requestqueue.html>  <https://developer.android.com/training/volley/simple.html> |