

Pruebas de perfilamiento – Sprint 3

MISW 4203 – Ingeniería de software para aplicaciones móviles
Grupo 1

Metodología

Para la ejecución de la prueba de perfilamiento se realizó la siguiente prueba:

<https://youtu.be/aLtLPnS8wmQ>

Resultados

El análisis de los resultados se enfoca en las 2 variables principales relacionadas con el rendimiento de la aplicación, consumo de CPU y consumo de memoria RAM. En particular, desde la perspectiva del usuario, no se detectaron interrupciones en el uso de la aplicación, la cual respondió de forma fluida a los inputs generados durante la prueba. Tampoco se encontraron errores de tipo OOM ni ANR

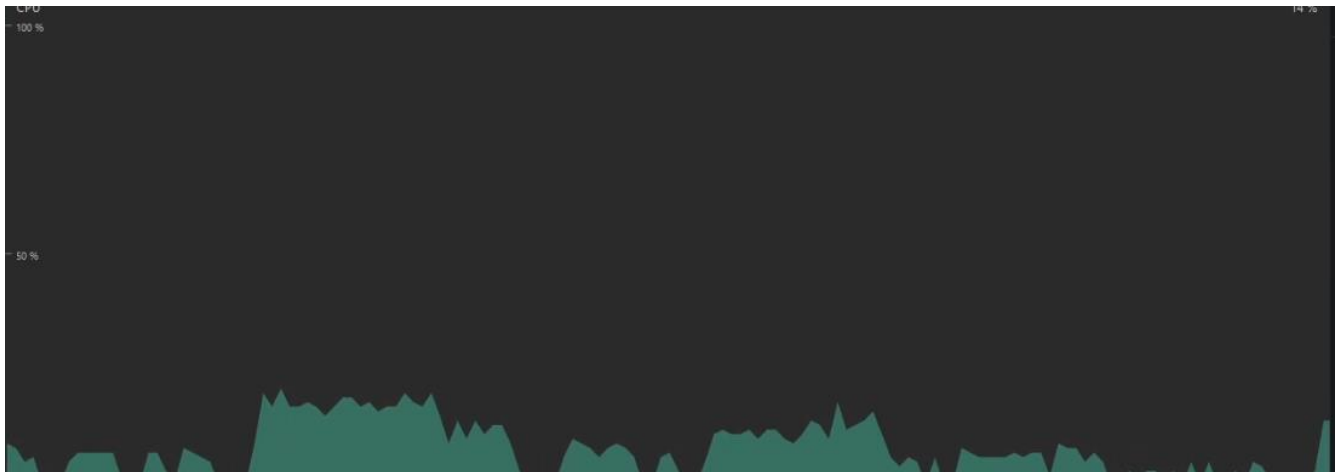


Figura 1 - Máximo de uso de CPU en la prueba ejecutada

En cuanto al consumo de CPU, se observó un pico de aproximadamente 30% en el uso de CPU durante la renderización de la vista de artista y álbum al cambiar rápidamente la aplicación. Esto es explicado por el uso de las corrutinas para realizar la carga de los elementos que no cargan adecuadamente sus imágenes al hacer invocaciones a la API REST. Sin embargo, gracias a las corrutinas y a la API remota, se observa un comportamiento adecuado de la aplicación.

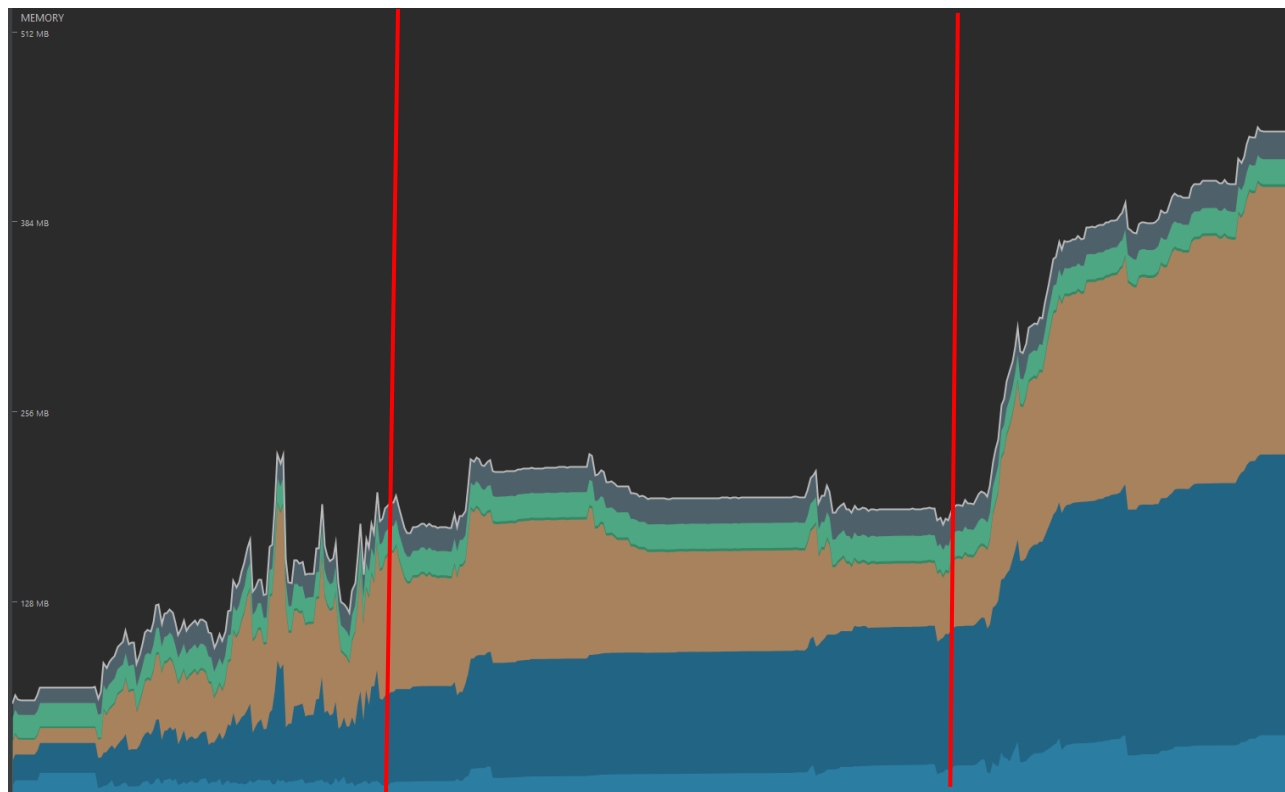


Figura 2 - Consumo de memoria RAM ejecutado durante la prueba

Durante la ejecución de la prueba se identificaron tres periodos importantes. En la figura 2 se representan separados por líneas rojas. La región izquierda, o inicial, representa la carga de todos los elementos de la aplicación. Es posible notar la carga de elementos y la ejecución de eventos de Garbage Collection. Sin embargo, es considerable que la memoria RAM promedio incrementa sostenidamente, lo que puede indicar la presencia de elementos con referencias fuertes

En la sección intermedia se observa un comportamiento muy estable. Esta región representa el uso de la vista de coleccionistas y los formularios de creación de álbum y asociación de canciones a un álbum. Esto indica que, aunque se use una librería como Glide para manejar el caché de las imágenes de la aplicación, uno de los mayores ofensores en el consumo de memoria RAM es la carga de imágenes desde el servidor. Es importante tener en cuenta que el coleccionista tiene una imagen de placeholder y no hace peticiones para carga de imagen ya que el modelo original no tiene este atributo.

Finalmente, la última región muestra un crecimiento acelerado en el consumo de memoria RAM en el momento en que se hace el cambio de fragmentos por medio de la barra de navegación inferior. Esto indica que no se están eliminando las referencias de forma oportuna al momento de destruir una vista

Conclusiones y siguientes pasos

Por una parte, las pruebas permitieron exponer que tener un gran número de información cargada en una sola vista impacta significativamente el rendimiento de una aplicación. Si esta, además, presenta información incompleta u orígenes de imagen dañados, la aplicación intentará continuamente realizar solicitudes hasta poder almacenar una imagen valida en el caché implementado.

Sin embargo, es importante destacar la estabilidad del proceso en escenarios de uso normales. A pesar del consumo excesivo de RAM durante la ejecución de la vista de artistas y en la vista de álbumes. Durante las nuevas vistas creadas se observa un comportamiento adecuado en cuanto al consumo de CPU y memoria RAM, mostrando la eficiencia de las corrutinas, las microoptimizaciones y el caché implementado en la aplicación