**Rendimiento de la API:**

Se realizará un breve análisis de cómo poder escalar nuestra API la cual, si bien funciona para unos pocos usuarios, deberá ser consumida por más y más usuarios a lo largo del tiempo. Por lo tanto, teniendo esto en mente, trataremos el problema de escalabilidad.

La prueba a realizar analizará el impacto de la base de datos sobre la disponibilidad de la API.

**LoadTest en SQLite:**

*Primero vamos a realizar un test de carga sobre la base de datos directamente.*

**Inicio**:

Se cargaron datos aleatorios sobre las tablas de la base de datos para poder estresar a la base de datos con los planes de prueba.

Las tablas fueron cargadas con la siguiente cantidad de registros:

* Cancion: 10000
* Fragmento: 40000
* Juego: 40000
* Tag: 10000
* Usuario: 14000

Se harán tanto consultas como escrituras para observar el porcentaje de error a causa de que la base de datos esté ocupada.

**Hipótesis**:

Debido a la cantidad de registros elegida se esperará que la BD presente problemas a la hora de la escritura (lockeo) y un pobre rendimiento al existir una relativamente grande cantidad de consultas simultáneas.