

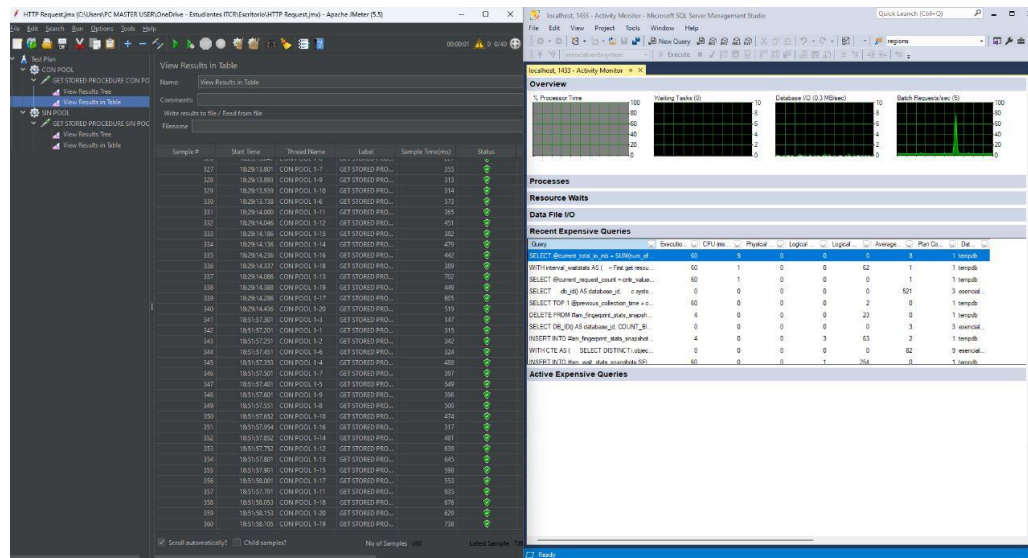
## Estudiantes:

Ariel Leyva c.2022175018

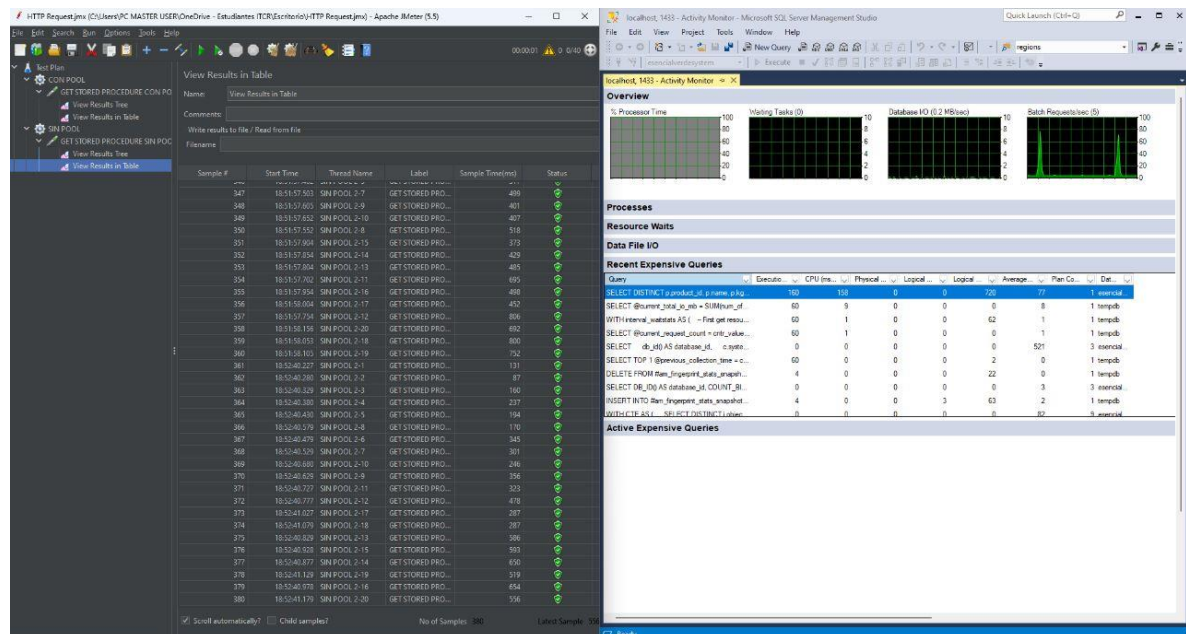
Tomás Granados c.2021579524

## Documentación de Métricas de Rendimiento REST-API / ORM

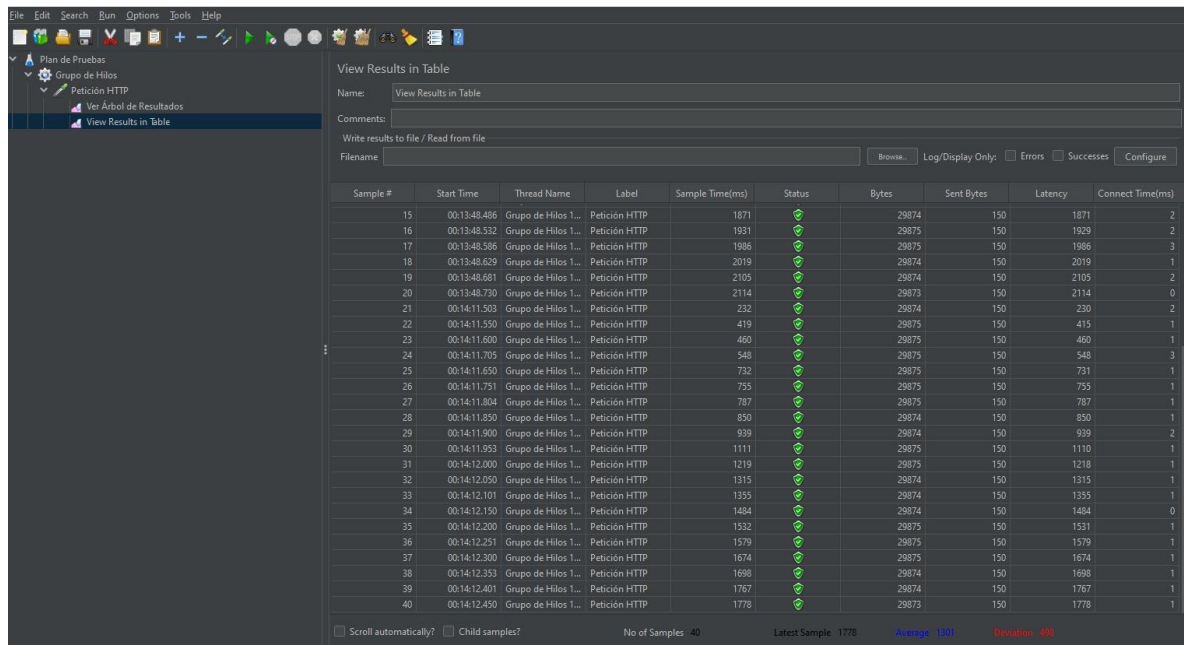
### Testeo de conexión Utilizando Pool Connection



### Testeo de conexión Utilizando Sin Pool Connection



## Testeo de conexión Utilizando ORM en Ruby



The screenshot shows a test runner interface with a sidebar on the left containing a tree view of test results. The main area displays a table titled 'View Results in Table'. The table has columns for Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, Sent Bytes, Latency, and Connect Time(ms). The data shows 40 samples of HTTP requests, all with a status of 'Success' (indicated by a green checkmark). The sample times are relatively low, ranging from 1871 to 1778 ms. The latency is consistently 150 ms, and the connect time is consistently 2 ms. The interface also includes a 'Write results to file / Read from file' section with a 'Filename' field and a 'Browse...' button. At the bottom, there are checkboxes for 'Scroll automatically?' and 'Child samples?', and summary statistics: 'No of Samples: 40', 'Latest Sample: 1778', 'Average: 1301', and 'Deviation: 498'.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
15	00:13:48.486	Grupo de Hilos 1...	Petición HTTP	1871	Success	29874	150	1871	2
16	00:13:48.532	Grupo de Hilos 1...	Petición HTTP	1931	Success	29875	150	1929	2
17	00:13:48.586	Grupo de Hilos 1...	Petición HTTP	1986	Success	29875	150	1986	3
18	00:13:48.629	Grupo de Hilos 1...	Petición HTTP	2019	Success	29874	150	2019	1
19	00:13:48.681	Grupo de Hilos 1...	Petición HTTP	2105	Success	29874	150	2105	2
20	00:13:48.730	Grupo de Hilos 1...	Petición HTTP	2114	Success	29873	150	2114	0
21	00:14:11.503	Grupo de Hilos 1...	Petición HTTP	232	Success	29874	150	230	2
22	00:14:11.550	Grupo de Hilos 1...	Petición HTTP	419	Success	29875	150	415	1
23	00:14:11.600	Grupo de Hilos 1...	Petición HTTP	460	Success	29875	150	460	1
24	00:14:11.705	Grupo de Hilos 1...	Petición HTTP	548	Success	29873	150	548	3
25	00:14:11.659	Grupo de Hilos 1...	Petición HTTP	732	Success	29873	150	731	1
26	00:14:11.751	Grupo de Hilos 1...	Petición HTTP	755	Success	29873	150	755	1
27	00:14:11.804	Grupo de Hilos 1...	Petición HTTP	787	Success	29875	150	787	1
28	00:14:11.850	Grupo de Hilos 1...	Petición HTTP	850	Success	29874	150	850	1
29	00:14:11.900	Grupo de Hilos 1...	Petición HTTP	939	Success	29874	150	939	2
30	00:14:11.953	Grupo de Hilos 1...	Petición HTTP	1111	Success	29873	150	1110	1
31	00:14:12.000	Grupo de Hilos 1...	Petición HTTP	1219	Success	29875	150	1218	1
32	00:14:12.050	Grupo de Hilos 1...	Petición HTTP	1315	Success	29874	150	1315	1
33	00:14:12.101	Grupo de Hilos 1...	Petición HTTP	1355	Success	29874	150	1355	1
34	00:14:12.150	Grupo de Hilos 1...	Petición HTTP	1484	Success	29874	150	1484	0
35	00:14:12.200	Grupo de Hilos 1...	Petición HTTP	1532	Success	29875	150	1531	1
36	00:14:12.251	Grupo de Hilos 1...	Petición HTTP	1579	Success	29875	150	1579	1
37	00:14:12.300	Grupo de Hilos 1...	Petición HTTP	1674	Success	29875	150	1674	1
38	00:14:12.353	Grupo de Hilos 1...	Petición HTTP	1698	Success	29874	150	1698	1
39	00:14:12.401	Grupo de Hilos 1...	Petición HTTP	1767	Success	29874	150	1767	1
40	00:14:12.450	Grupo de Hilos 1...	Petición HTTP	1778	Success	29873	150	1778	1

En breves palabras podemos decir que utilizar una conexión con pool permite la reutilización de conexiones existentes, lo que resulta en un mejor rendimiento y una gestión más eficiente de los recursos. En contraste, una conexión sin pool establece y cierra conexiones individuales para cada operación, lo cual puede ser menos eficiente y consumir más recursos del sistema.

Lo podemos ver reflejado en esta serie de pruebas en donde la que contiene el request utilizando la conexión sin pool hay un mayor de numero de solicitudes en el servidor, además de ser un poco más tardadas que las que fueron realizadas utilizando el pool de conexiones. Estas fueron realizadas de una manera más eficiente debido a que utilizando las conexiones ya establecidas con el servidor, sus consultas tienen mayor velocidad conexión y al hacer el mismo número de consultas en cada tipo podemos ver el sample time en la conexión con pool es mucho más bajo que sin pool.