

el servidor Django o que la gestión de sesiones no es relevante para el funcionamiento de la aplicación.

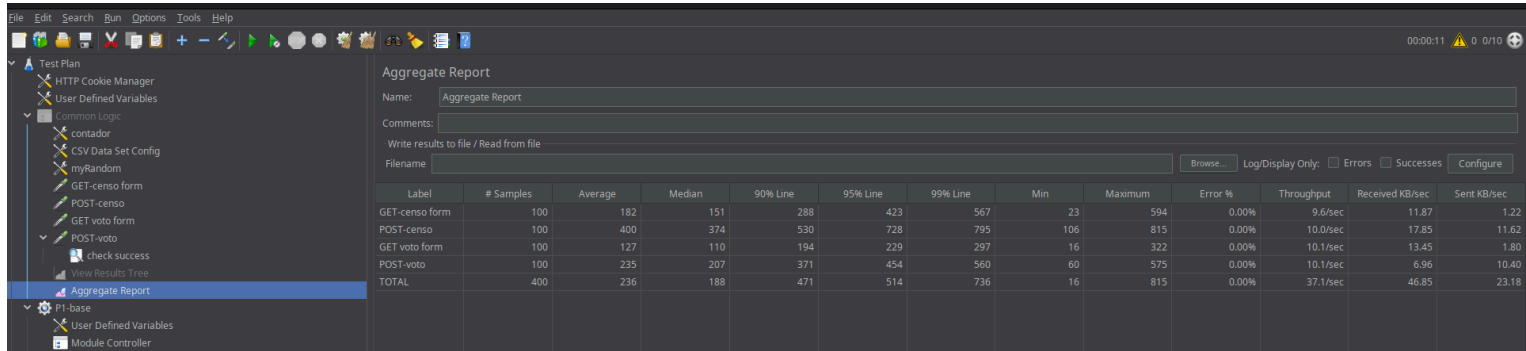
Memoria de Prácticas

Práctica 2

Sistemas Informáticos II

Autores: Ignacio González Porras y Daniel Wentworth Fernández

Ejercicio 1.



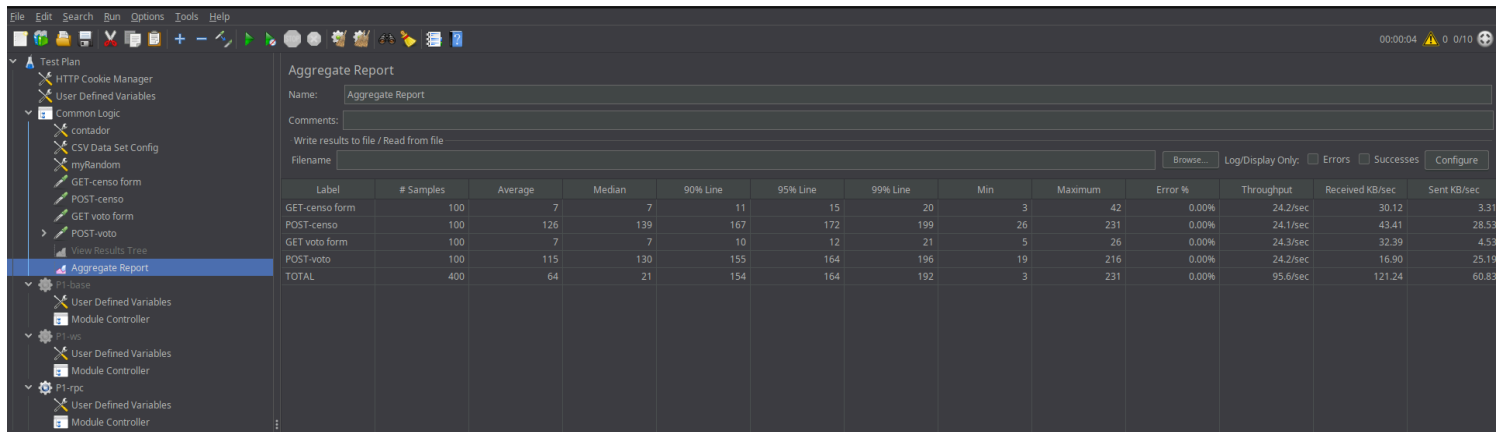
The screenshot shows the JMeter Aggregate Report for a Test Plan. The report is titled 'Aggregate Report' and displays performance metrics for various test elements. The table below summarizes the data shown in the report.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
GET-censo form	100	182	151	288	423	567	23	594	0.00%	9.6/sec	11.87	1.22
POST-censo	100	400	374	530	728	795	106	815	0.00%	10.0/sec	17.85	11.62
GET voto form	100	127	110	194	229	297	16	322	0.00%	10.1/sec	13.45	1.80
POST-voto	100	235	207	371	454	560	60	575	0.00%	10.1/sec	6.96	10.40
TOTAL	400	236	188	471	514	736	16	815	0.00%	37.1/sec	46.85	23.18

El fichero P2_P1-base.jmx se ha adjuntado con el resto de “tests plans” y código.

Ejercicio 2.

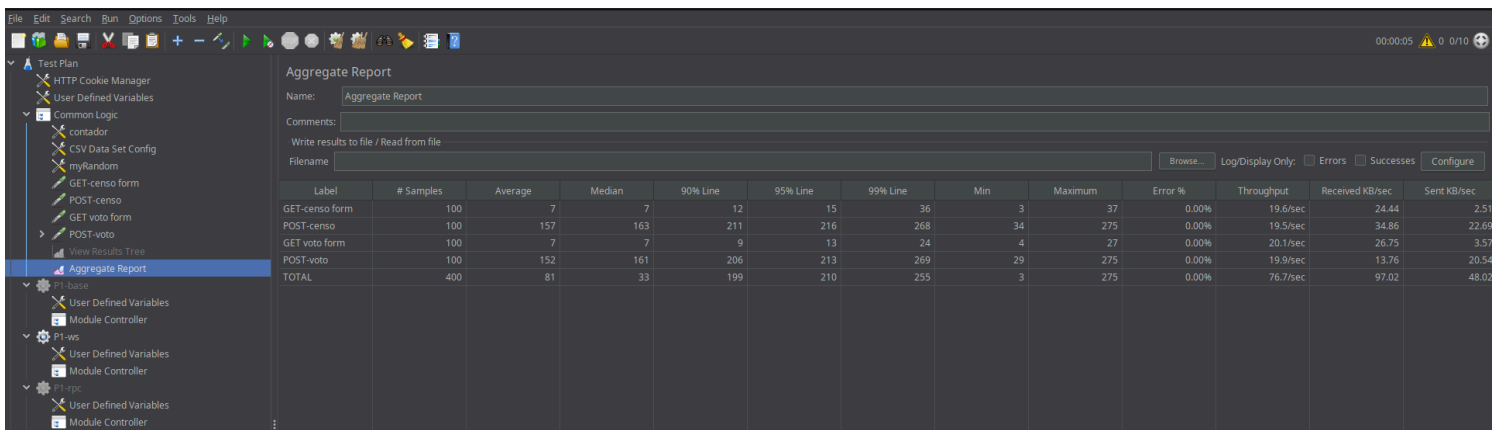
Test plan de P1-rpc:



The screenshot shows the JMeter Aggregate Report for a Test Plan. The report is titled 'Aggregate Report' and displays performance metrics for various test elements. The table below summarizes the data shown in the report.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
GET-censo form	100	7	7	11	15	20	3	42	0.00%	24.2/sec	30.12	3.31
POST-censo	100	126	139	167	172	199	26	231	0.00%	24.1/sec	43.41	28.53
GET voto form	100	7	7	10	12	21	5	26	0.00%	24.3/sec	32.39	4.53
POST-voto	100	115	130	155	164	196	19	216	0.00%	24.2/sec	16.90	25.19
TOTAL	400	64	21	154	164	192	3	231	0.00%	95.6/sec	121.24	60.83

Test plan de P1-ws:



The screenshot shows the JMeter Aggregate Report for a Test Plan. The report is titled 'Aggregate Report' and displays performance metrics for various test elements. The table below summarizes the data shown in the report.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
GET-censo form	100	7	7	12	15	36	3	37	0.00%	19.6/sec	24.44	2.51
POST-censo	100	157	163	211	216	268	34	275	0.00%	19.5/sec	34.86	22.68
GET voto form	100	7	7	9	13	24	4	27	0.00%	20.1/sec	26.75	3.57
POST-voto	100	152	161	206	213	269	29	275	0.00%	19.9/sec	13.76	20.54
TOTAL	400	81	33	199	210	255	3	275	0.00%	76.7/sec	97.02	48.02

Cuestión 1.

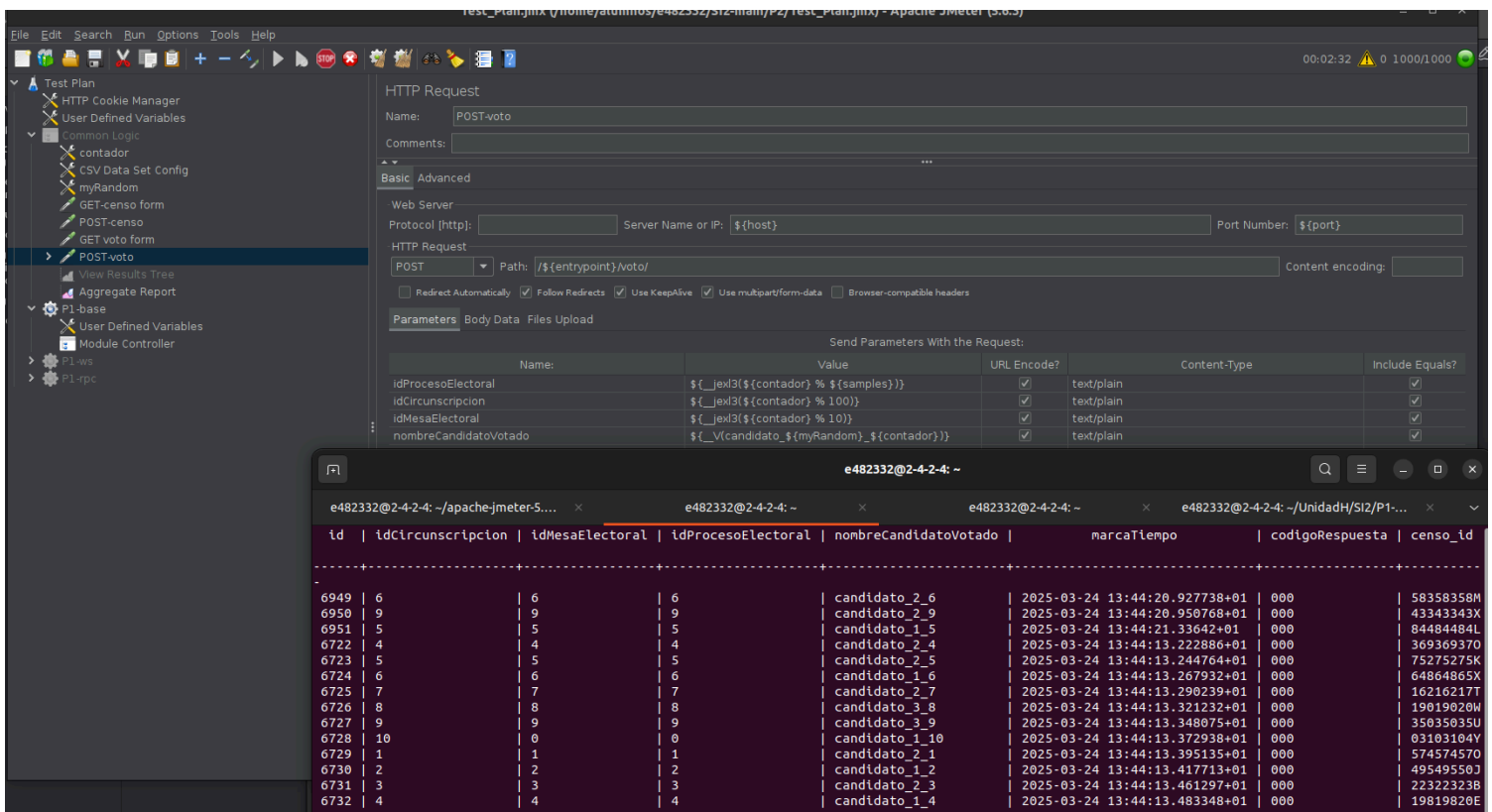
Las diferencias principales entre los dos “Thread Groups” son los entypoints que se utilizan para P1-rpc y P1-ws (que son diferentes al de P1-base) y el puerto usado, que se cambia al 38000, a diferencia del puerto 28000 que se utiliza en P1-base.

Ejercicio 3.

Según lo que podemos observar en Jmeter al ejecutar el plan de pruebas, Jmeter no genera todas las peticiones a al vez visualmente, pero vemos que si ha sido capaz de crearlas, simplemente el procesamiento de todas es costoso, y no es inmediato, luego van apareciendo las solicitudes progresivamente en la GUI de Jmeter.

Ejercicio 4.

A la hora de tratarlo dentro del bucle, sucede lo mismo. Las solicitudes se pueden crear sin problema, incluso antes de que sean atendidas la anterior iteración, respaldando nuestra respuesta en el anterior apartado de la independencia de las capacidades del sistema para realizar ambas tareas. Esto lo vemos ya que en la salida en Jmeter, vemos completamente intercaladas los ids de las solicitudes lo que nos lleva a pensar que hay disparidad entre cuando se crean y cuando se procesan, ya que el id puede haber sido creado después que otro y aparecer procesado antes que un id menor.



The screenshot displays the Apache JMeter interface. On the left, the 'Test Plan' tree shows a 'POST-voto' request under a 'P1-base' thread group. The main panel shows the configuration for this request, including the path '/\${entrypoint}/voto/' and parameters for 'idProcesoElectoral', 'idCircunscripcion', 'idMesaElectoral', and 'nombreCandidatoVotado'. Below the configuration, a terminal window shows the output of the test execution, displaying a table of results with columns for id, idCircunscripcion, idMesaElectoral, idProcesoElectoral, nombreCandidatoVotado, marcaTiempo, codigoRespuesta, and censo_id.

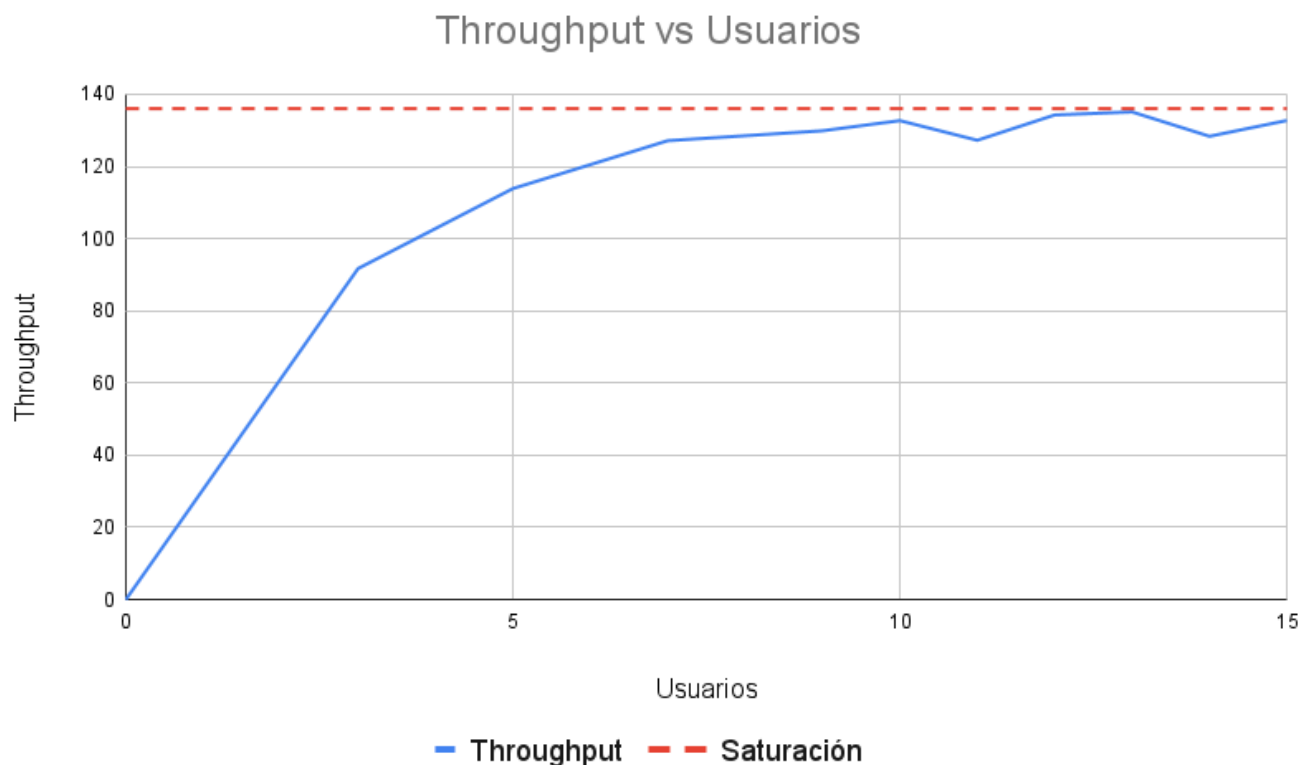
id	idCircunscripcion	idMesaElectoral	idProcesoElectoral	nombreCandidatoVotado	marcaTiempo	codigoRespuesta	censo_id
6949	6	6	6	candidato_2_6	2025-03-24 13:44:20.927738+01	000	58358358M
6950	9	9	9	candidato_2_9	2025-03-24 13:44:20.950768+01	000	43343343X
6951	5	5	5	candidato_1_5	2025-03-24 13:44:21.33642+01	000	84484484L
6722	4	4	4	candidato_2_4	2025-03-24 13:44:13.222886+01	000	369369370
6723	5	5	5	candidato_2_5	2025-03-24 13:44:13.244764+01	000	75275275K
6724	6	6	6	candidato_1_6	2025-03-24 13:44:13.267932+01	000	64864865X
6725	7	7	7	candidato_2_7	2025-03-24 13:44:13.290239+01	000	16216217T
6726	8	8	8	candidato_3_8	2025-03-24 13:44:13.321232+01	000	19019020W
6727	9	9	9	candidato_3_9	2025-03-24 13:44:13.348075+01	000	35035035U
6728	10	0	0	candidato_1_10	2025-03-24 13:44:13.372938+01	000	03103104Y
6729	1	1	1	candidato_2_1	2025-03-24 13:44:13.395135+01	000	574574570
6730	2	2	2	candidato_1_2	2025-03-24 13:44:13.417713+01	000	49549550J
6731	3	3	3	candidato_2_3	2025-03-24 13:44:13.461297+01	000	22323232B
6732	4	4	4	candidato_1_4	2025-03-24 13:44:13.483348+01	000	19819820E

Ejercicio 5.

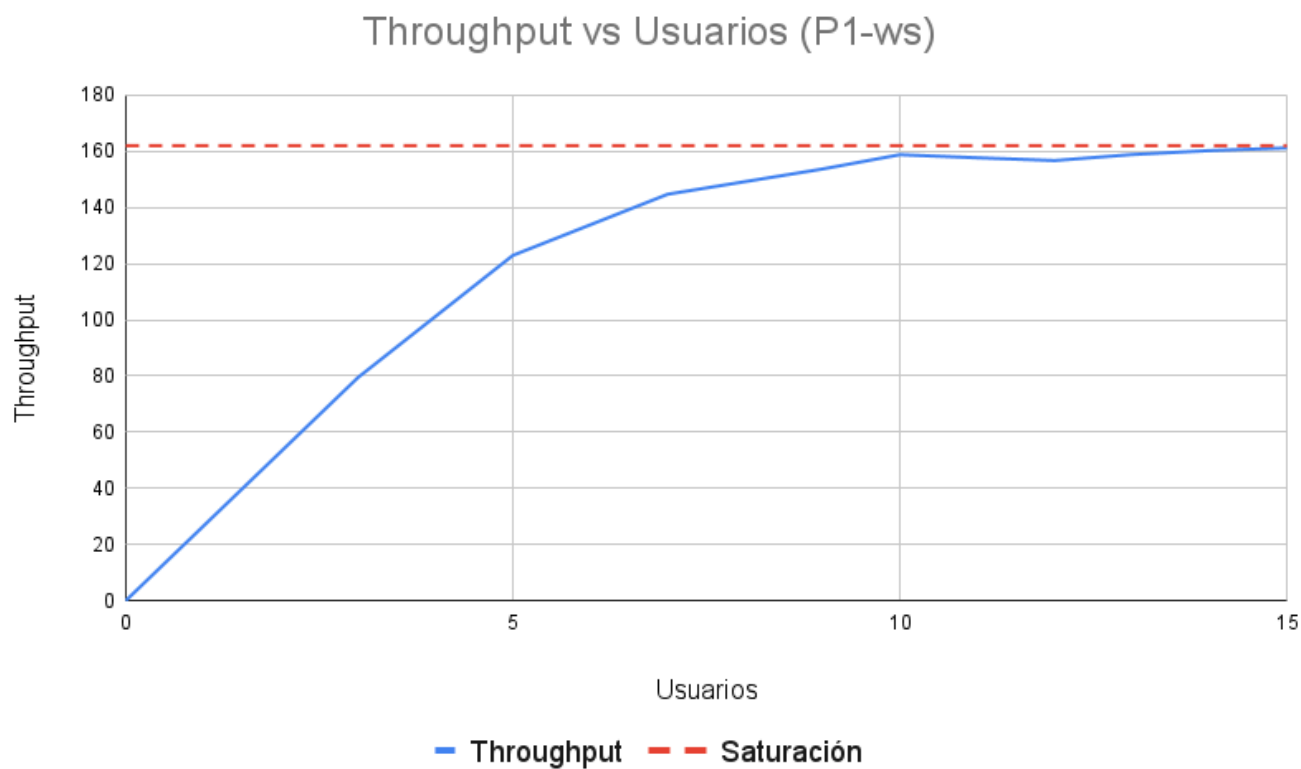
Para llevar a cabo estas medidas, aplicamos la variable $\{samples\}$ como el número de iteraciones para cada uno de los datapoints relativo al número de usuarios. Tomamos medidas similares a las vistas en el gráfico dado en el enunciado, tomando medidas distanciadas de dos en dos cuando el crecimiento era lineal, y luego de uno en uno para que la curva de saturación que se pretende mostrar sea lo más visible posible. También indicamos con una línea punteada dicho umbral de saturación, a modo visual y de manera aproximada que se tomó basándose exclusivamente en los datos experimentales.

En cuanto al rendimiento de cada uno, según los fuimos ejecutando en el orden propuesto, fuimos viendo un aumento en el throughput de salida. Respectivamente nuestras implementaciones de P1-base, P1-WS y P1-RPC, llegaron a una saturación máxima de en torno a 135, 160 y 190, siendo esta última, con RPC, la que dio mayor throughput. Esto se debe a que usa una comunicación binaria eficiente y conexiones persistentes, reduciendo la sobrecarga de serialización y latencia de red.

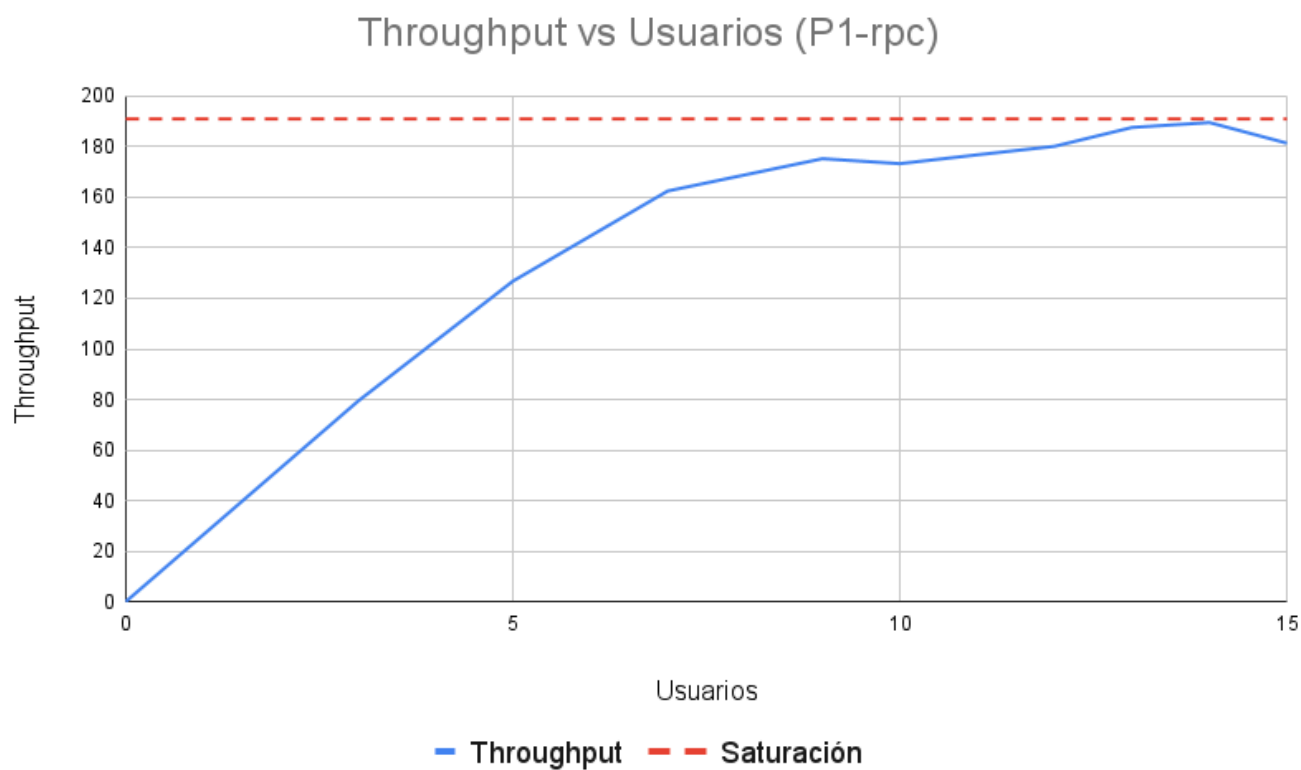
Throughput P1-base:



Throughput P1-ws:



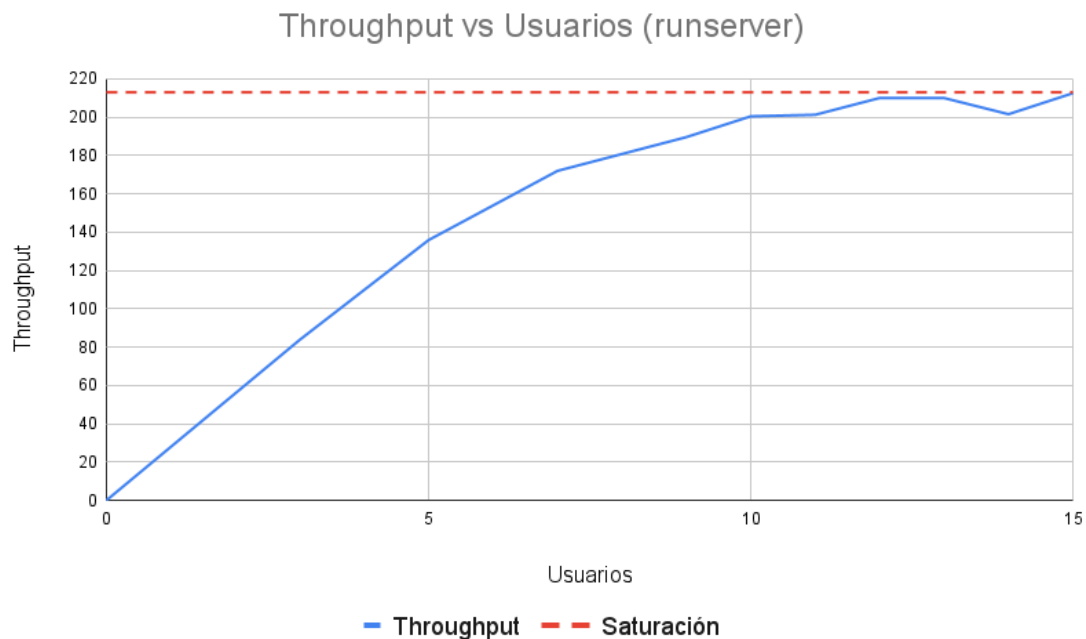
Throughput P1-rpc:



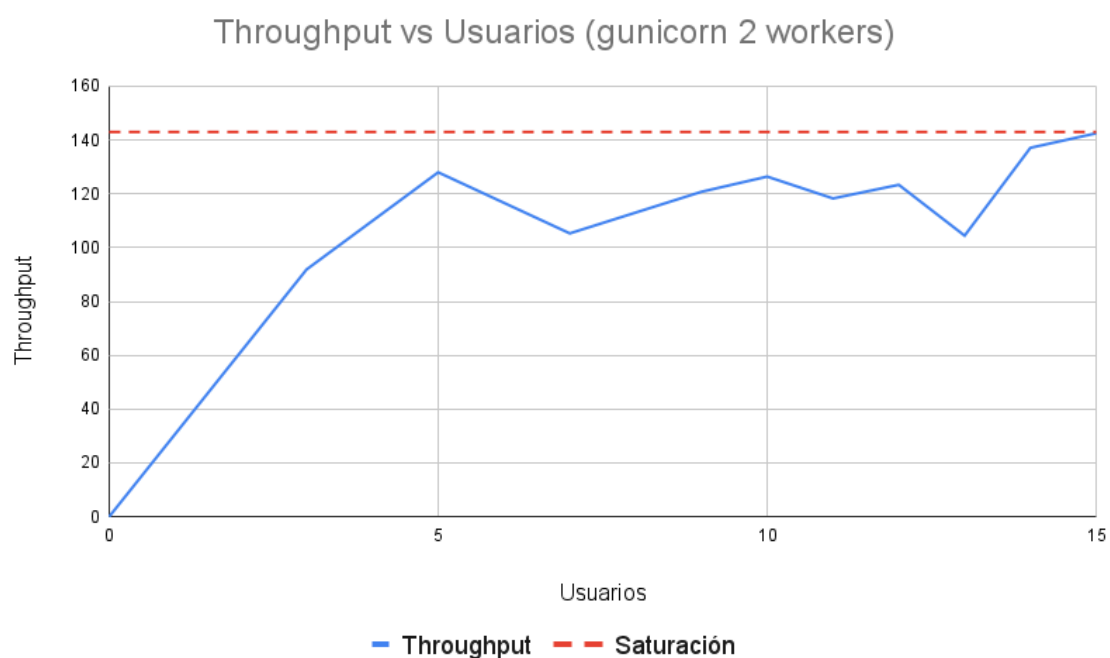
Ejercicio 6.

A la hora de comparar gunicorn, con un worker, usando P1-base como prueba común para ambos, con la ejecución del servidor de django usando runserver, obtuvimos que el P1-base como ya comentamos alcanza su saturación hacía casi un throughput de 140. Con runserver, se obtiene una medición significativamente mayor, resultando en un throughput de más de 210. El throughput de runserver es mayor porque es un servidor síncrono y ligero, sin la sobrecarga de gestión de workers. Gunicorn, aunque más robusto para producción, introduce overhead al manejar múltiples procesos concurrentes.

Throughput de runserver:



Throughput de gunicorn con un worker:



Ejercicio 7.

Cuando realizamos la medida con 2 workers de gunicorn, tras haber rebootado la máquina virtual asignándole otro core, su rendimiento aumenta sin lugar a duda. Sin embargo, nosotros esperábamos ver un aumento del 100% al ser el doble de workers ejecutándose, pero la salida real fue un throughput que se saturó hacía 145, tan solo marginalmente más que con un worker. Uno podría pensar que es la misma ejecución pero con variación experimental en los datos, ya que la diferencia no es muy significativa. Esto concluimos que se debe a que gunicorn usa un modelo pre-fork, y cada worker maneja solicitudes de forma independiente. Si el servidor no está altamente concurrido, el segundo worker no aporta mucha mejora, ya que el cuello de botella suele estar en la CPU, I/O o la base de datos.

Cuestión 2.

(d) en ninguno, ya que la gestión de sesiones en este proyecto no ha sido relevante para el funcionamiento de la aplicación