TEST CON JMETER

Todos los casos de prueba se realizan en un lapso de 2 segundos, con los nodos widfly usando 12 contenedores fijos sin auto-escalado vertical.

En estos casos de prueba se realiza una prueba de carga de la funcionalidad de agendar contra el bean que lo gestiona. Cada hilo de jmeter pasa por todo el proceso de agendar una fecha por la vacuna, quitando la parte de la interfaz web.

10 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
REST Agendar	10	610	99	1338	372,09	0,00%	5,3/sec	1,20	1,40	233,0
TOTAL	10	610	99	1338	372,09	0,00%	5,3/sec	1,20	1,40	233,0

Esta prueba modela un uso bajo del sistema, unos 10 usuarios en 2 segundos no deberían representar ningún problema para el sistema y por lo tanto sirven como punto de comparación para ver anomalías. Se observa un promedio de 0.6 segundos y un máximo de 1.3 segundos de tiempo de respuesta.

100 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
REST Agendar	100	804	382	1421	277,69	0,00%	40,8/sec	9,28	10,75	233,0
TOTAL	100	804	382	1421	277,69	0,00%	40,8/sec	9,28	10,75	233,0

Esta prueba no parece haber saturado el sistema, debido a unos tiempos de respuesta similares al de 10 usuarios.

250 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
REST Agendar	250	1402	393	1800	331,99	0,00%	69,3/sec	15,77	18,28	233,0
TOTAL	250	1402	393	1800	331,99	0,00%	69,3/sec	15,77	18,28	233,0

Esta prueba muestra una cierta degradación del sistema, debido a que no puede procesar tantos pedidos a un mismo tiempo. De todas formas el tiempo máximo de demora no varió tanto.

400 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
REST Agendar	400	3048	47	8487	2263,87	0,00%	41,1/sec	9,44	5,46	235,0
TOTAL	400	3048	47	8487	2263,87	0,00%	41,1/sec	9,44	5,46	235,0

Podemos ver que el sistema está totalmente saturado, degradando mucho sus tiempos de respuesta y empeorando el throughput comparado al caso anterior.

Visto cómo reacciona la plataforma al someterla a tales pruebas, se decidió probar escalándola horizontalmente, para ver la mejora que proporciona. Se utilizaron 3 nodos wildfly con las mismas características que el de las pruebas con 1 sólo nodo (12 cloudlets)

400 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
REST Agendar	400	1097	192	5233	1048,09	0,00%	67,1/sec	19,34	8,92	295,0
TOTAL	400	1097	192	5233	1048,09	0,00%	67,1/sec	19,34	8,92	295,0

Esta vez se tuvieron unos tiempos de respuesta prácticamente 3 veces mejores en términos de tiempo promedio de respuesta, lo cual corresponde al escalado realizado.

800 usuarios

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
REST Agendar	800	1816	185	5351	913,73	2,88%	140,3/sec	55,11	18,10	402,2
TOTAL	800	1816	185	5351	913,73	2,88%	140,3/sec	55,11	18,10	402,2

Se puede ver que esta configuración de despliegue permite más usuarios concurrentes sin mucho problema.