

Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

Performance



Grado en Ingeniería Informática – Ingeniería del Software

Diseño y Pruebas II

Curso 2019 – 2020

Miembros del equipo

Jorge Andrea Molina

Juan Carlos Cortés Muñoz

María Elena Molino Peña

Alejandro Muñoz Aranda

Mario Ruano Fernández

Fernando Ruíz Robles

<https://github.com/dp2-g3-7/petclinic.git>

Performance

Se ha probado el rendimiento de la aplicación sobre un ordenador de sobremesa con Windows 10, una memoria RAM de 16 GB y una Intel Core i7-3770. Se ha decidido realizar estas pruebas en una misma máquina para que el resultado sea lo más realista posible. Se presentan, a continuación, los datos para cada una de las 29 historias de usuario. Tras ellos, se añaden algunas consideraciones.

Historia de usuario	Óptimo	Mínimo Ruptura
ConcertarCita	1500	4500
CambiarFechaCita	1500	4500
CancelarCita	1500	4500
BuscarMascotaAdopcion	90	
AdoptarMascota	90	
SolicitarEstancia	2800	7000
ActualizarEstancia	3000	7000
CancelarEstancia	4800	9000
AceptarRechazarEstancia	3500	7000
BajaMascota	2600	6500
RegistrarMedicamento	4200	5000
ActualizarMedicamento	7000	9000
RegistrarMedialTest	500	3000
ActualizarPruebasMedicas	500	3000
RegistrarPetType	5000	7000
ActualizarPetType	7000	8500
CrearTratamiento	100	1500
ActualizarTratamiento	100	1500
EliminarHitorialTratamiento	200	1500
ListarTratamientos	1500	5000
RegistrarBanner	300	2500
EliminarBanners	300	2500
ActualizarDatosDeUsuario	3000	7000
SolicitarRegistroMascota	50	300
AceptarRechazarSolicitudMascota	10	500
RegistrarVeterinario	5000	7000
EditarVeterinario	4000	7000
RegistrarVisita	150	1000
EditarVisita	150	1000

Se observan dos HU sin número mínimo de clientes concurrentes para que se produzca el cuello de botella. Son las relacionadas con las peticiones a una API externa, cuyo mínimo de usuarios viene definido por la limitación de 1000 peticiones al día y 50 al segundo. Por ello, puesto que cualquier mejora se enfrentaría a estas limitaciones, se decidió no tratarlas con profiling.

El rendimiento es claramente inferior en las historias SolicitarRegistroMascota, AceptarRechazarSolicitudMascota, RegistrarVisita, EditarVisita, RegistrarBanner, EliminarBanners, CrearTratamiento, ActualizarTratamiento y EliminarHistorialTratamiento. Esas 9 historias pertenecen a 4 familias de historia de la aplicación: Mascotas, Visitas, Tratamientos y Banners. Por ello, para el siguiente apartado de profiling se van a escoger: SolicitarRegistroMascota, AceptarRechazarSolicitudMascota, RegistrarVisita, CrearTratamiento y RegistrarBanner. De esta forma, se conseguirá una mejora en el rendimiento de la otra historia relacionada con cada una de ellas.

Se debe mencionar que, en el caso de banners, se va a optimizar un método que está vigente en todas las peticiones de la aplicación; mostrar un banner publicitario aleatorio en cada vista de la aplicación. Por lo tanto, mediante la aplicación de esta mejora se conseguirá un mayor rendimiento global.