



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Pruebas de desempeño para aplicaciones
móviles y sitios web en logística
automotriz**

INFORME DE ACTIVIDADES PROFESIONALES

para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Jesús Manuel Lima Gallegos

ASESOR DE INFORME

M. en A. Luis Yair Bautista Blanco



Ciudad Universitaria, Cd. Mx., 2022

INDICE

Introducción	2
1. Empresa de soluciones en logística y transporte	3
1.1 Impacto empresarial	3
1.2 Misión de la empresa	3
1.3 Visión de la empresa	3
1.4 Valores de la empresa	3
1.5 Organigrama organizacional	4
1.6 Perfil de A.Q.A.E.	5
2. Capacitaciones	7
2.1 <i>Boot Camp</i>	7
2.1 Habilidades humanas	7
2.2 Habilidades tecnológicas	8
2.3 Habilidades organizacionales	10
3. Labores Empresariales	12
3.1 Generación de <i>Scripts</i>	12
3.2 Generación de <i>Datasets</i>	13
3.3 Pruebas de desempeño	13
3.3.1 Pruebas Manuales	13
3.3.2 Pruebas Automatizadas	14
4. Estrategia de Proyectos	16
4.1 Preparación	16
4.2 Pruebas	19
4.3 <i>Bugs</i>	24
4.4 Análisis de datos	26
5. Conclusiones	28
6. Bibliografía	29

Introducción

“La vida es y siempre seguirá siendo una ecuación incapaz de resolver, pero tiene ciertos factores que conocemos” - Nikola Tesla -

Tomando en cuenta las palabras del inventor, ingeniero y físico Nikola Tesla el inicio del trabajo en el campo laboral se asemeja mucho a este concepto, desde mi punto de vista, ya que nos incorporamos a un nuevo lugar sin algunos factores iniciales como la cultura, nuestras responsabilidades a desempeñar, gente con la que colaboraremos, etc.

Al iniciar mi camino por el campo laboral empecé a utilizar todos los factores conocidos que había adquirido en la Facultad de Ingeniería y a lo largo de mi vida, para desempeñar un papel exitoso dentro de la empresa generando utilidades de valor positivo para la misma.

Los factores no conocidos empecé a aprenderlos y a desarrollarlos para empezar a tener mejores cimientos, esto lo llevé a cabo mediante diferentes certificaciones y prácticas de equipo en la empresa.

Al tener esta ecuación que no podía resolver de manera individual en su totalidad, pero si podía resolver la parte con los factores conocidos, desarrollé diferentes pruebas de desempeño en las cuales me encargaba del rendimiento de diferentes servicios y proyectos dentro de la empresa generando utilidades y ahorro para la empresa.

Empresa de soluciones en logística y transporte.

Impacto Empresarial

Omnitracs es una compañía internacional con más de 30 años en Latinoamérica, líder en soluciones de tecnología con enfoque en seguridad y logística para la industria del transporte, ofreciendo soluciones de telemática, rastreo y ruteo para la administración de flota, a través de aplicaciones de software, hardware y servicios logísticos. Con más de 1,000 empleados, Omnitracs, ofrece soluciones para ayudar a más de 12,000 clientes a administrar 1,100,000 activos en más de 70 países. Omnitracs transforma la industria del transporte a través de tecnología y conocimiento experto, brindando las mejores soluciones de cumplimiento, seguridad y protección, productividad, telemática y seguimiento, gestión de transporte (*Transportation Management Solutions TMS*), planificación, entrega, análisis de datos y servicios profesionales. [1]

Omnitracs es una empresa que siempre apoya a diferentes organizaciones sin fines de lucro con una gran empatía hacia las personas; por ejemplo, durante la crisis de salud pública de COVID-19 Omnitracs ha apoyado a múltiples organizaciones de múltiples maneras, ya sea económicamente, a conductores de camiones de América del Norte y a sus familias en tiempos de crisis o ayuda a los conductores de semirremolques regionales y a sus familias cuando se produce una enfermedad o lesión. [2]

Misión de la empresa

Colaborar con empleados, clientes y la comunidad para ofrecer soluciones de tecnología de alta calidad, innovadoras, seguras y unificadas para maximizar la experiencia, la eficiencia, la seguridad del transporte y la protección del ecosistema logístico. [3]

Visión de la empresa

Ser el líder mundial en tecnología de transporte y logística a través de innovación disruptiva, proporcionando soluciones integradas, dinámicas e inteligentes. (Esta información es conocimiento interno ya estando dentro de la empresa)

Valores de la empresa

Integridad: Se trabaja con respeto, rectitud, bondad y honradez, todos dentro de la empresa son sinceros.

Calidad: Se trabaja siempre con la mayor eficacia, se tiene todo preparado para cuando va a salir un lanzamiento en el ambiente de pruebas y, en caso de encontrar algún defecto, se soluciona siempre lo antes posible, para que en la fecha de lanzamiento al público no exista ningún defecto.

Trabajo en Equipo: Si necesitas apoyo, ya sea por falta de conocimiento interno o que necesites ayuda con una herramienta, el personal interno de la empresa está para apoyarte en todo momento, se trabaja en conjunto para la solución de algunos defectos y para mantener todo el ambiente estable.

Innovación: Siempre se buscan nuevas herramientas para mejorar las pruebas y se actualiza todo cada vez que hay un lanzamiento. Se busca siempre generar actualizaciones con base en las opiniones de los clientes y al avance tecnológico.

Comunidad: Todos tienen un espíritu positivo enfocado a obtener los mejores resultados, somos proactivos con una mentalidad de éxito.

Organigrama organizacional



Dentro de la empresa yo tengo el cargo de *Associate Quality Assurance Engineer (A.Q.A.E)* como se muestra a continuación.

Como *Associate Quality Assurance Engineer* estoy encargado de que los diferentes ambientes cuenten con un buen rendimiento para que estos no tengan defectos a la hora de ser usados por los clientes mediante la generación de pruebas manuales y automáticas en los diferentes sitios web o aplicaciones móviles existentes.

Perfil de A.Q.A.E.

Actividades que se realizan en el trabajo como *Associate Quality Assurance Engineer*.

- Proporcionar el nivel de pruebas de estimación de esfuerzo: Se realizan diferentes pruebas de desempeño de manera automatizada en ambientes aislados con herramientas como *LoadRunner (VirtualUserGenerator)*, *Jmeter* o *Gatling* con códigos generados específicamente para cada ambiente.
- Comprender los requerimientos del producto y los casos de prueba y después ejecutar las pruebas y documentar los resultados: Inicialmente se nos capacita internamente para conocer los ambientes donde trabajaremos en la empresa, y una vez teniendo el dominio de la prueba, generamos *Scripts* para cada ambiente simulando diferentes escenarios que hacen los usuarios frecuentemente. Después en una página interna de la empresa subimos los resultados de las pruebas ya sean bugs, errores en los sitios web, tiempos de respuesta con pruebas automatizadas o manuales.
- Participar en las actividades de resolución de problemas relacionados con productos y sistemas de *Omnitracs*: Se revisa la documentación de los resultados en diferentes reuniones, para esto utilizamos la metodología *Agile*, la reunión principal es cada 30 días, en la cual asiste el *Senior Manager*, el *Manager II* y *Quality Assurance Engineers*, donde en equipo revisamos a detalle los diferentes problemas o actualizaciones que hay que generar. Después de esa reunión, tenemos breves reuniones de 15 minutos para discutir cómo vamos con las tareas asignadas.
- Documentar e informar defectos y problemas relacionados con productos y sistemas de *Omnitracs*: En algunas pruebas de desempeño encontramos algunos bugs o problemas relacionados con las actualizaciones de cada lanzamiento, esto se documenta internamente y, en una página interna de la empresa, se genera una solicitud de atención donde se describe a detalle la situación. Cada *ticket* se asigna por responsabilidades.

Lo que se necesita saber para realizar el trabajo como *Associate Quality Assurance Engineer*.

- Habilidades intermedias de comunicación escritas: Se requiere poder escribir en idioma inglés para describir problemas en las páginas de la empresa, entender algunos manuales y, mandar correos al personal.
- Habilidades intermedias de comunicación verbales: Se requiere poder hablar en idioma inglés para tener facilidad de expresión durante las reuniones de la empresa y poder entender sin ningún problema los videos de capacitación.
- Conocimientos básicos en metodologías de desarrollo de software: Ser capaz de poder entender códigos y desarrollar códigos al menos en un lenguaje de programación, de preferencia R y C#.
- Habilidades básicas para pruebas: Poder realizar pruebas de desempeño manualmente y de manera automatizada.
- Habilidades básicas de solución de problemas: Tener la capacidad de resolver problemas de programación interna, como, por ejemplo, en Scripts realizados con C#.
- Conocimientos técnicos básicos: Se debe tener un entendimiento de qué es una base de datos, lenguajes de programación (preferentemente R o C#), cálculo integral, álgebra, ecuaciones diferenciales y estadística.
- Conocimientos básicos en base de datos: Saber cómo funciona una base de datos y poder generar códigos SQL.

Capacitaciones

Boot Camp

Al ingresar a la empresa se llevan a cabo diferentes cursos intensivos internos para conocer los diferentes ambientes dentro de la empresa, al conjunto de los cursos intensivos los llamamos internamente *Boot Camp*.

Al inicio del *Boot Camp* tenemos una cálida bienvenida por parte del CEO y el *Senior Director* donde nos dan una descripción general de la empresa y cómo ha evolucionado a través de los años.

A continuación, tenemos una presentación con la Directora de Recursos Humanos que nos enseña los diferentes programas internos de recursos humanos que realiza la empresa y los puntos de contacto internos para saber dónde dirigirnos ante cualquier situación.

Posteriormente a la presentación con recursos humanos el *Senior Manager, Product Marketing* de la empresa nos da una descripción general de los que hace la empresa a nivel industrial y de los diferentes tipos de clientes con los que colaboramos, mostrándonos los diferentes servicios con los que contamos y la variedad de soluciones que le podemos dar a cada tipo de cliente.

Después un *Talent Development Partner* nos enseña a utilizar las diferentes plataformas para levantar tickets, registrar horas de trabajo, tomar capacitaciones en diferentes plataformas, registrar días por enfermedad o vacaciones, ver las noticias internas de la empresa, consultar el programa de salud, entre otras funciones.

Finalmente, la *VP, Product Management* nos enseña cómo se maneja el desempeño interno dentro de la empresa y la visión de ésta a un futuro más prometedor donde podemos apreciar claramente la innovación que se ha desarrollado a lo largo de la existencia de la empresa con el paso del tiempo dentro de la industria de logística y transporte.

Habilidades humanísticas

Como protocolo interno de la empresa se nos solicita tomar los siguientes cursos para desarrollar nuestras habilidades humanísticas, generando un mejor ambiente de trabajo en la empresa, y tener mejores relaciones interpersonales entre todos los

equipos para que de esta manera tengamos el mejor desempeño posible en nuestras actividades laborales y personales.

- *Goal Settings at Work:* Curso diseñado para que podamos crear un plan real con metas inteligentes conocidas internamente como *SMART Goals (Specific, Measurable, Attainable, Realistic, and Timely)* con marcos de referencia reales y aprender a comunicar efectivamente nuestros *SMART Goals* al equipo.
- *Feedback is Fuel:* Desarrollar una mentalidad de crecimiento asociado en recibir y dar retroalimentación, aprender los diferentes tipos de retroalimentación, reducir el miedo y ansiedad de recibir y dar retroalimentación identificando el elemento clave de dónde, cuándo y cómo.
- *Communication and Social Skills:* El curso está diseñado para ayudar a los profesionistas a tener más confianza y efectividad al comunicarse de forma verbal y escrita.
- *Growth Mindset:* Tiene el objetivo de generar mayor confianza en los profesionistas para tomar riesgos y generar una mentalidad de crecimiento en la empresa para generar mayor impacto positivo en aspectos culturales, innovación y compromiso para salir de nuestra zona de confort.

Posteriormente, al haber concluido los cursos de habilidades humanísticas, me di cuenta de que mis habilidades personales mejoraron en la creación de metas laborales y personales, mi forma de dar y recibir retroalimentación y mi mentalidad de crecimiento con un enfoque de éxito.

Habilidades tecnológicas

Una vez concluido el Boot Camp junto con las capacitaciones de habilidades humanísticas, se me integró al área de Performance donde se me asigna un copiloto de cargo Quality Assurance II, que es la persona encargada de mostrarme todos los detalles y actividades internas de Performance dentro de la empresa.

Después de aprender todo lo relacionado al área de Performance, me doy cuenta de que puedo reforzar mis habilidades ingenieriles para mejorar mi desempeño en mi equipo de Performance, especializándome en el uso de las herramientas que se utilizan en la empresa.

Empiezo tomando las capacitaciones para tener el dominio de los programas de cómputo LoadRunner, Jmeter, Blazemeter y Gatling, que son las principales

herramientas que utilizamos en el equipo de Performance para la generación de Scripts, con estas herramientas podemos replicar las acciones que generan nuestros clientes en los sitios web o en las aplicaciones móviles.

- *Performance Testing Course with Jmeter and Blazemeter:* Establecer los cimientos de conocimiento para crear códigos y realizar pruebas de desempeño de manera automatizada con *JMeter* y *BlazeMeter*.
- *HP LoadRunner 12.55:* Generación de códigos para pruebas de desempeño automatizadas en sitios web o en aplicaciones móviles.
- *Gatling Fundamentals for Stress, Load & Performance Testing:* Se aprende cómo grabar y escribir código para poder generar diferentes escenarios de prueba y poder analizar los resultados de estas mediante la herramienta de Gatling.

Posteriormente me dispongo a aprender a utilizar de manera apropiada la aplicación llamada Gitlab, que es una herramienta que utilizamos en el equipo de *Performance* para poder respaldar datos de las pruebas de desempeño y para compartir nuestros resultados con otros equipos de la empresa.

- *Gitlab:* Se obtiene una descripción general de la arquitectura de Gitlab enseñando cómo generar código dentro de la herramienta para crear pruebas compatibles con otras herramientas creando un registro de estas.

A continuación, al darme cuenta que, de manera interna, muchos de los datos que tenemos para nuestras pruebas de rendimiento los podemos visualizar con las aplicaciones de Grafana y Datadog, decido enrolarme en la capacitación de Grafana para poder aprender a crear nuevos gráficos para la visualización de nuestras pruebas.

- *Grafana:* Aprender el funcionamiento de Grafana con servidores SQL para poder crear una visualización gráfica de los datos obtenidos en las pruebas de desempeño.

Finalmente, la decisión de tomar el curso de Grafana me motiva a mejorar mis habilidades de programación, por lo que inicio capacitaciones en el lenguaje de programación R para poder utilizar los datos generados en cada prueba de desempeño o carga para así poderlos graficar y presentar de mejor manera visual en los reportes solicitados por la empresa.

- *R Programming A-Z™ & Advanced Analytics:* El objetivo es aprender a escribir código en R y trabajar con grandes cantidades de datos de manera sencilla para la generación de su análisis y creación de gráficos para presentaciones empresariales.

Al término de mis capacitaciones en el ámbito tecnológico me doy cuenta que mis habilidades de programación mejoran notablemente a pesar de ser un lenguaje diferente al aprendido en la universidad, también se nota una marcada diferencia en mi desempeño profesional al utilizar las principales herramientas para la generación de Scripts.

Habilidades organizacionales

Inspirado por la notable mejoría en mis habilidades tecnológicas, humanísticas y en mi preparación como estudiante de la Facultad de Ingeniería, en un proyecto donde aprendí a utilizar metodologías de diseño e ingeniería, decido emprender mi ampliación de conocimientos a nivel organizacional para mejorar en mi crecimiento y desempeño profesional por lo que me inscribo en el curso de *Mastering Agile Scrum Project Management* que es la metodología utilizada por mi empresa.

- *Mastering Agile Scrum Project Management*: Dominar desde sus principios la metodología Agile aprendiendo cambios de administración, priorización, expectativas de todos los participantes, empoderamiento de equipo y niveles de planeación para obtener las habilidades necesarias para poder medir los estándares de calidad y entender el rol de cada uno de los participantes con la concentración en las necesidades y metodologías de detección de errores, problemas o cualquier cosa considerada fuera de lo aceptable.

Posteriormente, tomo la iniciativa de capacitarme en CTFL para aprender a visualizar y tener mejor toma de decisiones para el desarrollo de código y metodología utilizada para las pruebas de desempeño, asegurando que se tengan resultados más exitosos en cada prueba realizada.

- *Certified Tester ISTQB Foundation Level (CTFL)*: Aprender a diseñar pruebas adecuadamente con los requerimientos de calidad necesarios en la industria, entendiendo los problemas clave en pruebas de software para aplicaciones.

Con la motivación de los cursos anteriores, tomo la iniciativa de inscribirme en un curso para tener el dominio en la aplicación Microsoft Teams poder comunicarme mediante la misma de manera más eficaz en mi empresa, ya que esta es la principal herramienta de comunicación dentro del equipo y la organización.

- *Mastering Microsoft Teams*: Se tiene como objetivo aprender a utilizar todas las funciones de la aplicación más reciente de *Microsoft Teams* para tener mejor comunicación interna en la empresa.

El gran estímulo de concluir exitosamente estas capacitaciones me apoya a generar mejores vínculos organizacionales con mejores herramientas para transmitir los resultados y conocimiento del equipo de *Performance* a las personas que lo soliciten además de tener mejor toma de decisiones en las herramientas y metodología a utilizar para cada situación.

Labores empresariales

Generación de *Scripts*

Previamente a la generación de *Scripts* nos reunimos como equipo todos los miembros del proyecto para analizar la mejor solución para cada escenario y tomamos la decisión de qué herramienta vamos a utilizar.

La herramienta principal para generar los *Scripts* es HP LoadRunner, después Jmeter o Gatling, cuando utilizamos HP LoadRunner para generar el *Script* hay dos maneras de hacerlo:

1. Obtener el código grabando HARs.files: En esta metodología grabamos los movimientos individuales de cada transacción paso a paso en nuestros sitios web o aplicaciones para nuestros diferentes servicios, se administran los archivos HAR mediante el *Development Tools* de Google, cuando se tienen todos los archivos HAR los abrimos uno por uno verificando las diferencias entre archivos paso a paso por transacción para poder generar el código.
2. Utilizamos una *skill* de la herramienta HP LoadRunner que nos permite grabar todos los movimientos a nivel html que realicemos, de esta manera replicamos todos los movimientos de cada transacción y al terminar la herramienta nos da un código basando en nuestros movimientos paso a paso, aunque este está incompleto.

Una vez obtenido el código completo, replicando paso a paso las transacciones que se hayan hecho en la página, se hace simultáneamente una verificación de código que consiste en reproducir el *Script* aunque esté incompleto y sabiendo que va a fallar, cuando tenemos visibilidad y conocimiento de qué falla el código, empezamos a crear código con llaves internas o expresiones regulares para que el *Script* funcione.

Finalmente, se comprime el código y se organiza por funciones, para el entendimiento del equipo de *Performance* y otros equipos de la empresa; a continuación, se parametrizan y encriptan todos los datos confidenciales por la seguridad de los datos empresariales que se maneja internamente.

Para los *Scripts* generados en Jmeter y Gatling se puede utilizar la metodología de uso de archivos HAR o construyendo el código completo insertando cada llamada al sitio web o aplicación nosotros mismos para controlar la carga deseada y que esta carga vaya directo a las APIs (Interfaz de Programación de Aplicaciones), sin embargo el

utilizar estas herramientas no siempre es nuestra primera opción debido a que las contraseñas y códigos de seguridad pueden quedar expuestos y vulnerables.

Generación de *Datasets*

Cuando hemos finalizado el Script con HP LoadRunner en algunos casos necesitamos generar un conjunto de datos tabulados en nuestros sistemas almacenados en nuestras bases de datos, a esto le llamamos internamente *Datasets*; para la creación de nuestros *datasets* creamos códigos en C#, lo cual nos permite tener acceso a diferentes datos y variables que son requeridos para el análisis en las diferentes pruebas realizadas en el equipo de Performance.

Posteriormente cargamos los *Datasets* en servidores remotos para ser utilizados por otras herramientas internas de la empresa y puedan estar a la disposición del equipo de Performance y listos para las pruebas que sean necesarias en cada nueva versión con datos actualizados de nuestros servidores.

Pruebas de Desempeño

Las pruebas de desempeño son una clase de pruebas que utilizamos para conocer el rendimiento de un software o una aplicación; en estas pruebas podemos evaluar características como perfiles de tiempo, flujo de ejecución, tiempos de respuesta, fiabilidad, límites operativos, escalabilidad, estabilidad, entre otros.

Pruebas Manuales

Inicialmente, antes de hacer la programación para automatizar nuestras pruebas hacemos pruebas manuales que consisten en entrar en nuestros diferentes sitios web y aplicaciones haciendo diferentes movimientos estratégicos para simular diferentes posibles escenarios que pueden suceder en cada ambiente.

Utilizamos herramientas como Datadog y Grafana, utilizamos Dashboards creados por el equipo de Performance utilizando los API's necesarios para tener una visualización completa de las respuestas al procesar por el algoritmo y entender que está sucediendo durante el proceso para los usuarios; de esta manera también podemos notar de manera tangible exactamente qué sucede durante cada evento y cada transacción.

Posteriormente, utilizando herramientas internas de la empresa durante las pruebas manuales, obtenemos datos confidenciales que nos ayudarán a la generación de *Scripts* para poder iniciar las pruebas manuales.

Cabe destacar que, cuando encontramos un error en una transacción durante nuestras pruebas automatizadas, optamos por hacer pruebas manuales, ya que de esta manera siempre se tiene mayor visibilidad y podemos obtener datos más específicos de la transacción fallida; esto nos brinda la confianza de evitar que alguno de los problemas que encontramos en el equipo de *Performance* llegue a alguno de nuestros clientes cuando las nuevas versiones son liberadas.

Las ventajas de hacer las pruebas manualmente son que nos dan una idea de la experiencia de lo que viven nuestros clientes y encontrar algunos detalles; por ejemplo: la ubicación de los botones en la interfaz para facilitarle a nuestros clientes la localización de diferentes servicios, generación de caminos óptimos, etc.; este tipo de detalles siempre se documentan para su optimización y de esta manera poderle brindar una mejor experiencia de nuestros servicios a los clientes.

Pruebas Automatizadas

Una vez que hemos realizado pruebas manuales y garantizamos que todo está en orden empezamos a unir e implementar los *Datasets* y los *Scripts*, separados cada uno por servicio, a un software con la integración de Loadrunner para empezar a hacer las pruebas automatizadas en nuestras máquinas locales y analizar si se necesita hacer una adaptación de código o si todos los datos necesarios para las pruebas están completos.

Posteriormente cuando tenemos el script de manera automatizada funcionando en nuestra máquina local generamos un archivo en uno de nuestros paquetes de cómputo empresariales que nos ayuda a darle aún más detalles a la prueba, como número de usuarios virtuales que queremos utilizar por transacción, número de iteraciones, reestructurar el orden de las transacciones, *spacing*, generación de logs con datos retornados por el servidor, *think time*, velocidad de la simulación entre otros detalles específicos para mejorar cada una de nuestras pruebas.

Una vez que tenemos las pruebas con todos los detalles específicos que necesitamos para cada ambiente lo subimos a un software interno de la empresa donde todos con las credenciales y permisos adecuados tendrán acceso a la prueba para utilizarla en caso de requerirlo, de igual manera se documenta quiénes pueden asignar estos permisos para utilizar estos recursos por parte de la empresa.

Es importante que para poder hacer de manera automatizada la prueba se deben colocar los *datasets* en servidores remotos con las credenciales y requerimientos solicitados por la empresa, para que estos se vinculen automáticamente al momento de iniciar la prueba.

En el momento en el que todo está preparado para dar inicio a las pruebas automatizadas se documenta toda la información de funcionalidad en páginas de la empresa y en proyectos de Gitlab por equipo, cuando terminamos de hacer toda la documentación se inician pruebas aisladas las cuales nos permiten garantizar que la prueba está funcionando apropiadamente.

Finalmente nos ponemos de acuerdo con otros equipos para coordinar nuestras pruebas y que se lancen siempre al mismo tiempo para obtener datos más completos de los ambientes que se están poniendo a prueba. Cabe destacar que siempre, antes de cada prueba, iniciamos con una *Smoke Test*, la cual nos indica cómo está funcionando el ambiente y si todo está en orden para poder iniciar nuestras pruebas automatizadas en equipo.

Estrategia de Proyectos

Preparación

Inicialmente, para poder lanzar una prueba o programarla, es importante hacer la preparación de la misma debido a que si no seguimos todos los pasos en el orden correcto pueden fallar diferentes servicios. El primer paso es prender todos los servidores que usamos en nuestras pruebas; para llevar a cabo esto creamos múltiples Scripts con PowerShell que es una interfaz de consola con posibilidad de escritura la cual nos permite unir comandos mediante instrucciones, dentro de cada uno de estos Scripts, que se encuentran en nuestras máquinas remotas, agregamos un conjunto de comandos en lenguaje de scripting donde mandamos a llamar nuestros servidores, ya sea para encenderlos de manera individual o por grupo; primero los Linux y después los Windows, y para apagarlos primero son los Windows y después los Linux, porque si lo hacemos de manera distinta estos pueden fallar tanto en el encendido como en el apagado.

A estos Scripts les damos mantenimiento cada vez que actualizamos nuestros datos en los servidores, cada 6 meses, para poder hacer pruebas siempre con los datos más recientes. La manera de darle mantenimiento a este script es removiendo o agregando instancias de servidores nuevos con una secuencia adecuada para que de esta manera quede todo de manera automatizada y podamos ahorrar tiempo en el encendido o apagado de nuestros servidores. Haciendo estos Scripts de manera automatizada generamos un ahorro de recursos para la empresa, por el trabajo que anteriormente se hacía manualmente, con un tiempo de ahorro aproximado de 45 a 60 minutos al no saturar la página principal donde tenemos todos los servidores.

Después de encender todos los servidores tenemos que monitorear de manera visual que todos se encendieron de manera correcta, para esto en el equipo de Performance creamos el siguiente Dashboard en Datadog:

- Dashboard Manage Monitors: Cumple el objetivo de verificar que todos los servidores son encendidos o apagados de manera correcta y que estos están funcionando apropiadamente. Para tener la visualización de este dashboard creamos una tabla donde mandamos a llamar cada servidor por sus tags y sus identificaciones únicas por servidor, servicio o cliente. Una vez separando todos los servidores utilizando las identificaciones y tags únicas agregamos una barra de status en la tabla que nos permite identificar si hay datos, si todo está apagado, si todo está encendido o una advertencia que nos da a entender que algo fuera de lo normal está ocurriendo.

<input type="checkbox"/>	STATUS	MUTED LEFT	NAME ↑
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF - AMQ Consumers Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF - ActiveMQ Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF - NullPointerException error is too high
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF - PgSQLException error is too high
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF - System.Exception error is too high
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF Aliveness Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF ES Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF GEO Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF HA Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF HAMG Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF HAWS Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF IFTA Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF MG Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF MP Process Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF Service Endpoint Check
<input type="checkbox"/>	OK	🔊 x ∞	RDC PERF TCP Check
<input type="checkbox"/>	OK	🔊 x ∞	[Synthetics] RDC PERF - Web Services
<input type="checkbox"/>	OK	🔊 x ∞	[Synthetics] RDC PERF - Website DriverLogEditor

Fig. 1 Datadog Monitors

Utilizar el Dashboard Manage Monitors de Datadog nos ha permitido prevenir que se hagan lanzamientos de nuevas versiones cuando todo el ambiente está apagado. Este dashboard también ha ayudado internamente a tener un monitoreo completo del ambiente de pruebas, por si se cae un servidor al momento de ejecutar una prueba, lo cual nos permite saber con mayor facilidad el momento preciso en el cual cayó y poder detener las pruebas y empezar a investigar porque ha caído, ahorrándonos tiempo en ejecuciones.

Viendo lo importante que es innovar en la empresa, recientemente cree un Script de programación en R, se utilizó el lenguaje en R debido a que internamente en el equipo de Performance tenemos Scripts en R relacionados con las bases de datos, el objetivo principal de este script es mandar a llamar todas las bases de datos que tenemos en los servidores.

Mediante la incorporación de comandos de SQL al nuevo Script en lenguaje R puedo mandar a llamar todas las bases de datos de los servidores del ambiente de pruebas para darles un mantenimiento automático. Este mantenimiento consiste en utilizar, en todas las bases de datos, los comandos Reindex (sirve para recrear los índices de las tablas en las bases de datos y que queden de manera ordenada) y Vacuum Analyze (se utiliza para realizar limpieza en cada tabla de las bases de datos, evitando que el sistema se sobrecargue de filas vacías) en todas las bases de datos lo cual nos permite que al momento de correr nuestras pruebas se realicen con mayor velocidad y para que en producción nuestros clientes puedan acceder a nuestros servicios de una manera más rápida.

Anteriormente este mantenimiento lo hacíamos manualmente en las bases de datos directamente pero al darme cuenta de que destinábamos aproximadamente 2 días en este proceso decidí crear el Script de mantenimiento el cuál tarda un total de 6 a 8 horas aproximadamente en ejecutarse completamente de manera automatizada lo cual permite al equipo de Performance realizar las pruebas más rápido y que durante el tiempo de ejecución del Script de mantenimiento la persona encargada pueda realizar otras actividades simultáneamente sin estar 100% enfocado en esta tarea, generando que los recursos del personal de la empresa estén con mayor disponibilidad para otras actividades las cuales sean necesarias que atendamos.

Finalmente utilizamos Performance Center ya que es una herramienta de pruebas de rendimiento global habilitada para la Web de HP, Performance Center está diseñada para agilizar el proceso de pruebas y aumentarlas; dentro de Performance Center tenemos los Scripts para los diferentes servicios que maneja la empresa, para la parte de preparación creamos un Script que llamamos Smoke test el cuál creamos utilizando lenguaje de programación C, internamente para crear el Script de manera manual como lo hicimos en el equipo de Performance se necesita tener el conocimiento de

programación del lenguaje C y tener noción de cómo funciona la herramienta LoadRunner para poder mandar a llamar diferentes URL's.

El conocimiento del lenguaje C y de la herramienta LoadRunner a un alto nivel como miembro de la empresa es vital para poder establecer un orden en base a las funciones en el código de programación, poder encriptar todos los datos confidenciales por parte de la empresa y clientes además de poder actualizar manualmente el código cuando se hacen actualizaciones en los ambientes de prueba y crear un camino digital que seguirá el código para simular los escenarios deseados.

Lo que se desea obtener al hacer el Smoke test en el ambiente de pruebas por el equipo de Performance es realizar una revisión rápida de que funcionan todos los servicios con la carga mínima en el ambiente de pruebas y garantizar que no tiene ningún defecto que pueda interrumpir las mismas; para obtener éste escenario deseado en el Smoke test hay que verificar en el código que toda la construcción esta correcta; esto lo podemos notar en la estructura del código y verificando que tiene todos los parámetros correctos como son expresiones regulares para la validación de tokens que tenemos que implementar manualmente para el acceso a algunas páginas web de la empresa, encriptaciones correctas de usuarios, contraseñas y otros parámetros confidenciales por parte de la empresa y creación de funciones if/else para que los usuarios entren con las versiones correctas en los sitios web de esta manera asegurando que todos los parámetros están vigentes y no ocasionarán ningún problema a los servicios en las pruebas de desempeño.

Pruebas

Una vez terminada la etapa de preparación accedemos a nuestras máquinas remotas y utilizamos nuestros emuladores para los dispositivos móviles, se tomó la decisión de utilizar un emulador para ejecutar los programas y aplicaciones en las diferentes plataformas de la empresa ahorrando recursos de la empresa como sería tener múltiples dispositivos físicos (mínimo 16,000 equipos) ejecutando las aplicaciones y programas; por eso para nuestras simulaciones en dispositivos móviles llegamos a la conclusión de que el emulador puede representar todos los dispositivos que necesitamos a un bajo costo para la empresa.

Es importante tener el conocimiento de lenguaje C# a un nivel alto para poder comprender el código del emulador ya existente en la empresa y hacer ciertas modificaciones estratégicas que requerimos para cada cambio de versión como dónde ponemos los parámetros como son: número de conductores, driver batch size, driver batch time, driver ramp up time, start delay, message bundle interval, customer driver

list file, relay simulator files entre otros parámetros requeridos para las pruebas de desempeño; adicionalmente a esto en el código se debe calcular exactamente cuántos mensajes vamos a mandar y garantizar que se van a enviar cierto número de mensajes por usuario en un tiempo determinado.

Una vez establecidos los parámetros iniciamos el emulador para empezar a generar una carga similar a la de producción para poder emular un escenario como el que se vive día a día y poder encontrar defectos y generar ideas para mejorar la experiencia del usuario como lo es: tiempo de respuesta, acumulación de mensajes, entre otros servicios generados por los emuladores.

Al mismo tiempo que iniciamos nuestras pruebas para dispositivos móviles en nuestras máquinas remotas accedemos a Performance Center donde utilizamos otros Scripts que tenemos creados en Load Runner para el resto de los servicios de la empresa como son reportes y páginas web; a diferencia del Smoke test estos Script están configurados en mayor número de iteraciones y usuarios virtuales para aumentar la carga, un pacing que sirve para darle la cadencia a cada transacción que escribimos en nuestras líneas de código, activamos la función de log y adicionalmente a esto escribimos un código por si Performance Center falla poder visualizar todos los logs generados durante la prueba que se van guardando mientras se ejecuta la misma, un thinktime (tiempo de espera) después de cada transacción para que el comportamiento del Script sea lo más humano posible y no tan mecánico.

Todos los Scripts creados para los múltiples servicios son ejecutados al mismo tiempo generando una carga equivalente a la que se vive en producción, las pruebas de rendimiento las hacemos en un tiempo aproximado de 2 horas con un ambiente de pruebas aislado para tener datos más precisos y poder tener más confianza en los resultados; el resultado más importante de esta prueba de desempeño es que nuestras bases de datos no se saturen de mensajes y los servidores no caigan por lo que el equipo de Performance creó diferentes dashboards con los que monitoreamos todos los ambientes simultáneamente para verificar que las pruebas de desempeño se desarrollen correctamente.

Los diferentes dashboards que utilizamos durante el tiempo de prueba para monitorear los creamos en las herramientas de Datadog y Grafana y cada uno cumple una función específica para así evitar que algo externo afecte nuestras pruebas de desempeño, o en caso de que algún evento interno no esperado ocurra como puede ser una gran acumulación de mensajes o un servidor caído.

- Performance Apps: Cumple el objetivo de poder ver en tiempo real como van todos los servicios relacionados con las pruebas que emulamos en nuestras máquinas remotas y en Performance Center, para tener la visualización de este

dashboard dividimos todo en graficas por servicio, en cada uno de estos gráficos utilizamos conceptos básicos de Álgebra para tener una formalización real de los datos y poderlos acomodar mediante sus identificaciones y tags relacionadas separándolas gráficamente en 2 vectores dependiendo de cada servicio para obtener datos importantes como son las solicitudes enviadas por minuto, errores sucedidos durante la prueba, solicitudes totales enviadas, entre otros datos importantes para la empresa.

Para obtener datos importantes más específicos para nuestras pruebas de desempeño en dispositivos móviles aplicamos en este dashboard comandos de programación para realizar conceptos de Cálculo Diferencial y Cálculo Integral como lo es la combinatoria de la suma de una integral y de una derivada de los datos obtenidos de los mensajes enviados y de esta manera al hacer la sumatoria de estos cálculos obtenemos una gráfica de acumulación que nos

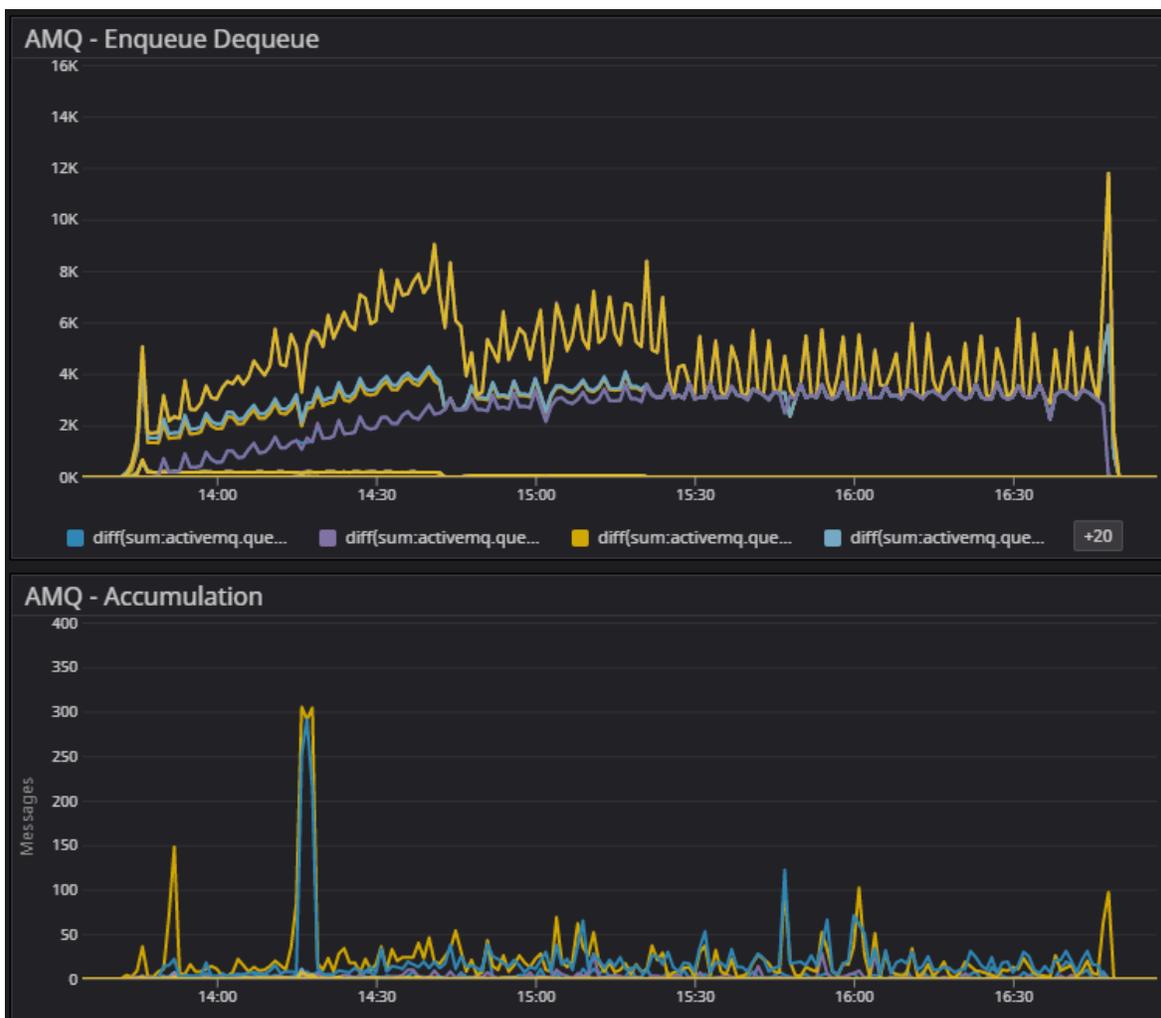


Fig. 2

ayuda a saber cuántos mensajes quedan acumulados en nuestras filas y nos dan indicios de que nuestros servidores están funcionando apropiadamente.

- Performance Grafana: Cumple el objetivo específico de monitorear las pruebas de desempeño para la emulación de dispositivos móviles en nuestras máquinas remotas, para tener la visualización de este dashboard creamos distintas gráficas a partir de los datos que el emulador manda a nuestras filas, todos los mensajes al estar en nuestras filas tienen un identificador por tipo de servicio como lo agregamos en el código C# de los emuladores; de esta manera dentro de grafana colocamos los identificadores de servicio y las tags aplicándoles una sumatoria la cual nos arroja una tabla de mensajes enviados por servicio la cual tiene la función de ver cómo se van desplazando los mensajes y verificar que se están mandando correctamente y que la cantidad de mensajes sea correcta, adicionalmente en el código C# del emulador agregamos una lista de conductores con diferentes parámetros confidenciales con los cuales los podemos graficar cuando se conectan y empiezan a manejar; de esta manera obtenemos una gráfica que nos permite visualizar que todos los conductores estén conectados y emitiendo mensajes de manera correcta.

Finalmente, para el Dashboard de Performance Grafana a la gráfica de conductores en línea anteriormente descrita aplicamos ciertas operaciones de derivación para poder visualizar de forma real el número de conductores que van entrando al sistema en tiempo real.

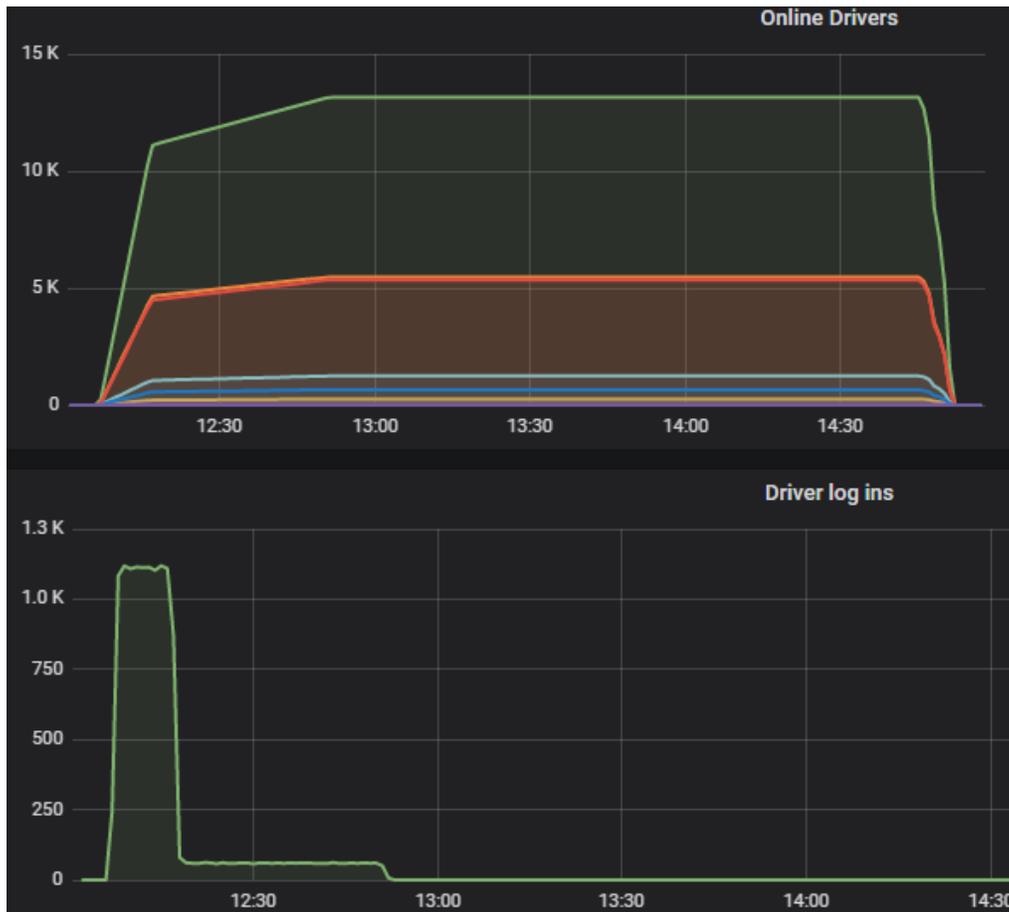


Fig. 3

- Performance Resources: Cumple el objetivo específico de monitorear el uso de CPU, memoria y red de todos los servicios durante las pruebas de desempeño para la emulación de dispositivos móviles en nuestras máquinas remotas y para los Scripts de Performance Center, para poder tener la visualización de este dashboard creamos distintas gráficas a partir de poner las API's de los servidores junto con los identificadores de servicio, aplicando una sumatoria por gráfica obtenemos los distintos valores para CPU, memoria y red que ocupamos para verificar que los servidores no estén muy saturados de datos al momento de realizar la prueba y en caso de que lleguen a estar muy saturados ver los errores que se generan a partir de los logs que encontramos por cada servicio; para saber el número de archivos abiertos por los identificadores acumulados

en la gráfica de igual manera aplicamos una combinatoria de una derivada y de una integral definida de cada servicio con identificadores únicos en un lapso de tiempo definido.

Durante el tiempo de prueba monitoreamos todas nuestras pruebas de desempeño utilizando los múltiples dashboards de Datadog y Grafana anteriormente descritos y estos nos ayudan a tener una mejor visualización de todo lo que está pasando respecto a las pruebas de rendimiento mientras sin perder ningún detalle de las mismas durante su tiempo de ejecución siendo así una manera más eficaz debido a que podemos ir interpretando los datos y encontrar posibles bugs simultáneamente optimizando el tiempo para el equipo de Performance para la parte de análisis de datos.

Finalmente, cuando las pruebas de desempeño terminan, verificamos que todo el ambiente esté funcionando de manera óptima utilizando el Dashboard de Datadog Manage Monitors y empezamos la etapa de recopilar todos los datos generados durante el tiempo de prueba para encontrar bugs y posibles errores en algún servicio.

Bugs

Los bugs principalmente los vamos detectando simultáneamente mientras realizamos las pruebas de desempeño, principalmente su detección surge a partir del monitoreo de las gráficas de los dashboards de Performance Resources, Performance Apps y Performance Grafana.

Inicialmente nuestro enfoque va hacia la parte de acumulación de mensajes donde por diversos motivos se pueden saturar, recientemente detectamos un bug debido a la falta de actualización de un software que generaba una alta acumulación de mensajes en nuestras filas; confirmamos que el bug estaba siendo generado por este software debido al monitoreo de su uso de CPU, ahí nos dimos cuenta que al momento de utilizarse el CPU al 100% se acumulaban los mensajes alentando nuestros servicios.

Para ello realizamos nuestras pruebas de desempeño con diferentes cargas para encontrar un punto de quiebre, sin embargo nos dimos cuenta que este error no era por la carga que poníamos en el sistema por medio del emulador sino porque el software era muy lento para su procesamiento lo que nos llevó a solicitar el apoyo del equipo de IT para una actualización del software.

Una vez actualizado el software utilizando la misma carga con la que llegamos al 100% de uso de CPU que es una carga igual a la del ambiente de producción; su acumulación bajó drásticamente de más de 40 mil mensajes acumulados a un máximo de 350 mensajes acumulados por minuto; esta mejora debido a la eliminación de este bug

podimos notarla en el uso del CPU en la misma prueba debido a que con la misma carga en diferentes ocasiones ya no alcanzaba un uso de 100% sino que ahora se mantiene estable en un 7% de uso de CPU mejorando el desempeño en el ambiente de pruebas dándonos la seguridad de que esta misma actualización debe hacerse en producción para evitar futuros errores generados por la acumulación como caída de servidores y bases de datos.

Posteriormente otro bug que encontramos fue que teníamos muchos *Deadletters* (mensajes que fallaron al enviarse) los cuales estaban siendo generados por una falta de credenciales en algunos usuarios que utilizamos en el emulador; estos no tenían todas las credenciales en los datasets inicialmente debido a que los usuarios reales no contaban con ellas, lo anterior generaba avisos de *Deadletters* y notificaciones excesivas por lo que modificamos nuestros datasets agregando credenciales temporales a los usuarios para que estos estuvieran 100% autenticados en nuestros sistemas generando que nuestro porcentaje de error se redujera de un 5% a un 0% en nuestras pruebas de desempeño.

Finalmente hemos encontrado bugs en la parte de reportes y sitios web, estos los hemos encontrado mediante el análisis de datos y búsqueda en los logs, es decir después de ver como resultaron nuestras pruebas si notamos un bug con los datos obtenidos tratamos de replicarlo para ver y analizar porque se esta generando el mismo y de esta manera poderlo corregir para que no suceda en producción.

Un ejemplo de lo anterior fue recientemente cuando una de nuestras transacciones fallaba y al replicarla manualmente paso a paso pudimos darnos cuenta que esta no fallaba en el ambiente de producción, sin embargo en el ambiente de Performance para nuestro equipo sí debido a que se estaban generando parámetros duplicados en las llamadas lo que nos llevo a documentar este error que estaba siendo generado por el mismo servidor alertando que si no actualizan los servicios o si no se ponen correctamente los parámetros en los lanzamientos pueden ocurrir estos errores.

Gracias a la detección y corrección de los errores ya mencionados hemos podido llegar en nuestros servicios a una tasa de error de 0% en todas nuestras pruebas de desempeño actualizando todos nuestros servicios y dándole el mantenimiento correcto, de esta manera también prevenimos la generación de bugs en nuestras pruebas para así ser más eficientes con nuestros clientes otorgándoles un mejor servicio.

Otra forma preventiva que tenemos para mejorar nuestros servicios mediante pruebas de desempeño y evitando bugs es adentrándonos en el código (revisar el código línea por línea e irlo ejecutando por secciones de manera calmada) y realizando diferentes escenarios de código en los cuales mejoramos el tiempo de respuesta en nuestro

ambiente; posteriormente esto lo documentamos y lo enviamos al equipo de IT, DB y managers para que tomen las medidas respectivas hacia si es conveniente implementar nuestras mejoras en el código de producción o no.

Análisis de datos

Previo a dar inicio al análisis de datos recopilamos todos los datos en bruto generados durante nuestras pruebas de desempeño, para la parte de dispositivos móviles descargamos en el dashboard Performance Apps un archivo formato .csv que contiene todos los mensajes que fueron mandados a cada fila desde nuestro emulador y también obtenemos el archivo de los logs generado el cual se encuentra en la máquina virtual donde se inició el emulador, este archivo que contiene los logs contiene más detalles específicos de los servicios en dispositivos móviles de la empresa.

Para la parte de reportes y sitios web recopilamos los datos en bruto directamente de Performance Center, estos lo descargamos en formato .zip y adicionalmente de tener los datos en bruto, contiene un reporte ya generado de manera automática el cual utilizamos para tener una primera visualización de cómo se comportó el ambiente de pruebas con las diferentes cargas de los múltiples servicios y tener una primera impresión de que todo está funcionando apropiadamente y que ningún servicio ha fallado.

Posteriormente hemos creado múltiples Scripts en lenguaje de programación R compatibles con otros ya existentes dentro del equipo de Performance que cumplen la función de separar todos los servicios por transacción y obtener datos relevantes para nuestros clientes y la empresa como son: valores máximos, porcentaje, valores mínimos, comparativa entre versiones, desviación estándar, 90 percentil y SLA Status (Service Level Agreement); para obtener estos datos en los Scripts que son de vital importancia para la toma de decisiones en las reuniones utilizamos diferentes conceptos:

- Cálculo Integral y Cálculo Diferencial para obtener datos importantes como el número de conductores que ingresan al sistema por minuto o número de peticiones por minuto.
- Estadística para hacer el cálculo de promedios, valores máximos, valores mínimos, percentiles y saber si algunos cambios son estadísticamente significativos.
- Programación para hacer todo el código desde su inicio hasta tenerlo de manera automatizada con diferentes lenguajes aprendidos.

Todos los datos ya procesados y comparados los ponemos en un reporte de Confluence (herramienta que utilizamos en la empresa para vaciar todos los datos importantes de la misma) con métricas del perfil de pruebas utilizado, gráficas que generamos mediante nuestros Scripts de R que nos muestran la comparativa entre diferentes versiones por lo que de esta manera al cliente le es más fácil ver las mejoras al producto.

Finalmente, todos los datos los guardamos en una nueva branch de Gitlab la cual esta enfocada únicamente a resultados; esta branch es revisada por los managers y por los equipos que requieran datos de las pruebas de rendimiento para visualizarlas y comparar resultados con otros servicios.

Conclusiones

Durante mi paso en la Facultad de Ingeniería me hubiera gustado aprender SQL y manejo de bases de datos en alguna materia adicional o como un tema complementario de las asignaturas de programación ya que para el ambiente laboral es muy importante conocer este rubro debido a que gran parte de las empresas trabajan con bases de datos y gestión de la información día con día.

Posteriormente en las mismas materias de programación avanzada se debería enseñar expresiones regulares (RegEx) ya que en el campo laboral en el área de desarrollo de software en el momento de la creación de código cuando se requiere crear un Script con Tokens que caducan continuamente complica un poco el avance óptimo para la creación de código para pruebas de rendimiento, carga, etc..

Sería deseable también incorporar otros lenguajes de programación al plan de estudios y no exclusivamente C o C# por ejemplo R, Python o Java, para darle más herramientas a los estudiantes de ingeniería para desarrollar nuevas tecnologías en el campo laboral.

Es vital tener pruebas de rendimiento y que estas sean analizadas y realizadas cuidadosamente paso a paso en todos los procedimientos ya que si alguno de estos no se lleva a cabo en el ambiente de producción algo puede fallar o tener errores generando descontento en los clientes y posibles accidentes.

Haber participado en dinámicas profesionales durante mi formación profesional en la Facultad de Ingeniería me permitió aprender metodologías de diseño como las que estudio en la materia de Diseño mecatrónico con metodologías ágiles, no obstante, de lo ya mencionado me permitió acoplarme fácilmente al campo laboral.

La preparación durante los primeros años de la carrera me dio las herramientas para tener un buen criterio y poder desempeñar correctamente mis funciones con un grado alto de confiabilidad entregando resultados óptimos a mi empresa.

Bibliografía

- [1] Omnitracs México, «Linkedin.com,» [En línea]. Available:
<https://www.linkedin.com/company/omnitracs-méxico/about/>.
- [2] R. Greer, «Omnitracs.com,» 30 Marzo 2020. [En línea]. Available:
<https://www.omnitrac.com/es/blog/carta-abierta-del-ceo-de-omnitrac-usted-es-nuestra-red-de-seguridad>.
- [3] Omnitrac, «omnitrac.com,» [En línea]. Available:
<https://www.omnitrac.com/omnitrac-de-mexico>.