

**UNIVERSIDAD GABRIELA MISTRAL
FACULTAD DE INGENIERIA**

**Monitoreo sintético para medir disponibilidad y
tiempo de respuesta de una página web
mediante un proceso de negocio**

Memoria para optar al título de Ingeniero de Ejecución en Informática

Autor : Diego Olave Salgado.
Profesor Guía : Roberto Caru Cisternas.
Profesor Integrante : Jorge Tapia Castillo.

Santiago, Chile

Noviembre 2017

INDICE

1	INTRODUCCION	4
1.1	El problema actual	6
1.2	Objetivos de la investigación.....	9
1.3	Justificación.....	10
1.4	Hipótesis	11
2	MARCO TEORICO	12
2.1	Área de Operaciones T.I.	13
2.2	Tecnologías de Información.....	13
2.3	Infraestructura T.I.....	13
2.4	Página Web	14
2.5	Sistema Informático	15
2.6	Tiempo de respuesta	15
2.7	Plataforma Tecnológica.....	16
2.8	Ingeniería de sistemas.....	16
2.9	Software de monitoreo	17
2.10	Componente.....	18
2.11	Métricas	19
2.12	Portal web	19
2.13	Splunk Enterprise.....	20
2.14	Biblioteca Selenium	23
2.15	Ingeniería de software	25
2.16	Panel de control	25
2.17	Monitoreo Sintético.....	26
2.18	Experiencia Usuario.....	28
3	DESARROLLO	29
3.1	Metodología	29
3.1	Solución Propuesta.....	29
3.2	Recolección de requerimientos	30
3.3	Requerimientos de Software.....	31
3.4	Requerimientos de Hardware.....	34
3.5	Creación de flujo de navegación	35
3.6	Definición de ponderadores	38
3.7	Definición de disponibilidad	40

3.8	Definición de tiempo de respuesta.....	42
3.9	Creación del monitoreo sintético	43
3.10	Ejecución del monitoreo sintético.....	63
3.11	Splunk Forwarder	66
3.12	Configuración Splunk Enterprise	68
4	HALLAZGOS	74
4.1	Alto consumo de procesamiento	74
4.2	Problemáticas en la operación actual.....	74
4.3	Dependencia del monitoreo sintético	74
5	CONCLUSIONES.....	75
5.1	Realización de la implementación	75
5.2	Determinar disponibilidad y tiempo de respuesta.....	75
5.3	Navegación no representativa para usuario final	75
5.4	Es posible desplegar información de forma simple.....	75
5.5	Es posible desplegar información en tiempo real	75
5.6	Complemento al monitoreo actual	76
6	GLOSARIO.....	77
7	BIBLIOGRAFIA.....	81

INDICE DE FIGURAS

FIGURA 1: EJEMPLO UTILIZACIÓN DE DISCO	7
FIGURA 2: COMPONENTES INFRAESTRUCTURA T.I.....	14
FIGURA 3: PLATAFORMA TECNOLÓGICA	16
FIGURA 4: SOFTWARE DE MONITOREO.....	17
FIGURA 5: COMPONENTES DE UN SERVIDOR.....	18
FIGURA 6: COMPORTAMIENTO ESTRUCTURAS DE MEMORIA DE UNA BASE DE DATOS.....	19
FIGURA 7: SPLUNK.....	21
FIGURA 8: SELENIUM NAVEGADORES.....	24
FIGURA 9: PANEL DE CONTROL	26
FIGURA 10: CONCEPTOS EXPERIENCIA USUARIO	28
FIGURA 11: ASIGNACIÓN DE TAREAS SERVIDORES	32
FIGURA 12: FLUJO DE NAVEGACIÓN MONITOREO SINTÉTICO	35
FIGURA 13: PONDERADORES POR ETAPAS	39
FIGURA 14: PONDERADORES SUB-ETAPAS	39
FIGURA 15: DISPONIBILIDAD	41
FIGURA 16: EJEMPLO RESULTADOS MONITOREO SINTÉTICO	41
FIGURA 17: UMBRALES TIEMPO DE RESPUESTA DE UN CLICLO	42
FIGURA 18: CALCULO DE TIEMPO DE RESPUESTA.....	42
FIGURA 19: UMBRALES CICLO COMPLETO	43
FIGURA 20: CALCULO TIEMPO DE RESPUESTA DE CADA ETAPA	43
FIGURA 21: FLUJO DE MONITOREO SINTÉTICO	62
FIGURA 22: LOG MONITOREO SINTÉTICO	63
FIGURA 23: CREACIÓN DE ÍNDICE EN SPLUNK.....	68
FIGURA 24: CREACIÓN DE PANEL DE CONTROL EN SPLUNK	69
FIGURA 25: DEFINICIÓN DE UMBRALES EN SPLUNK	69
FIGURA 26: TABLERO DE CONTROL FINAL	73

1 **INTRODUCCION**

En la actualidad, los sistemas informáticos son el núcleo de la operación de las empresas, áreas tales como recursos humanos, contabilidad, finanzas, departamento de ventas entre otras, se ven favorecidas por ellos para la optimización y fluidez de los distintos flujos que tiene una empresa para un determinado proceso.

Si nos centramos en el rubro del *retail* específicamente en el comercio electrónico, cobran una alta relevancia, ya que sin ellos, este rubro no se habría desarrollado hasta el punto en el que nos encontramos actualmente, según indica el Índice de Ventas por Internet realizado por el Departamento de Estudios de la Cámara Nacional de Comercio, Servicios y Turismo de Chile (CNC) para el año 2016, las ventas en línea aumentaron en un 30 % anual, respecto al periodo anterior.

La comodidad de la compra electrónica permite a las empresas de este rubro, bajar el costo de la venta de un producto, no necesariamente se debe tener un producto en stock, también existe la disponibilidad de realizar ventas de productos las 24 horas, se eliminan intermediarios, entre un sin número de otras ventajas.

Para las áreas de ingeniería de sistemas involucradas en las tecnologías de información uno de los principales objetivos es mantener los sistemas informáticos constantemente operativos llegando prácticamente al 100% de disponibilidad de servicios, mantener controlados los problemas, fallas y/o incidentes que afectan las plataformas tecnológicas son las tareas básicas a realizar por las áreas de ingeniería de sistemas por el bien de la continuidad operativa de los negocios.

Para realizar estas funciones existe una larga lista de software de monitoreo que están presentes en el mercado, donde, cada uno de ellos aporta valor en una o más áreas específicas de cara a los componentes de hardware y software que posee una solución

tecnológica, tales como sistema operativo, virtualización, base de datos, telecomunicaciones, etc.

Independiente de los software de monitoreo existentes en nuestro mercado, que nos ayudan a minimizar eventos de pérdida de servicio (total o parcial), alertarnos en caso que no se cumplan reglas de monitoreo de recursos (*Cpu, Memoria, I/O*), detectar vulnerabilidades de seguridad, entre otras. Estos softwares no cuentan con la funcionalidad de monitorear procesos de negocio de comercio electrónico.

Es por este motivo que el presente trabajo tiene como finalidad poder medir en tiempo real la disponibilidad y tiempo de respuesta de una página web a partir del comportamiento del proceso de negocio de venta en línea de productos, el cual será ponderado de acuerdo al impacto correspondiente a cada una de las etapa que componen en el flujo de navegación que ejecuta un usuario dentro de la página web.

1.1 *El problema actual*

Actualmente nos encontramos con que existe en el mercado una variada cantidad de softwares de monitoreo que prestan ayuda a las personas que trabajan en las áreas de infraestructura tecnológica, estos software les permiten mantenerse informados acerca del comportamiento de su plataforma, algunos de ellos son específicos para el monitoreo de redes, otros específicos para el monitoreo de bases de datos relacionales, también tenemos software de monitoreo más estandarizados a las distintas necesidades de las plataformas tecnológicas, donde se puede ver el comportamiento tanto de recursos de sistema como también el comportamiento de servicios de sistema operativo.

El problema que se observa en la mayoría de estos softwares es que para medir un determinado servicio o recursos de sistema, la medición se hace en base a un parámetro de una alerta determinada, la cual, está asociada a un servicio o recurso de sistema, por lo tanto su foco esta puesto en medir el límite de capacidad más que en la proactividad para la detección de posibles problemas y/o incidentes.

Generalmente los parámetros de una alerta que definen el comportamiento de lo que sea que queramos monitorear se catalogan como *alerta de advertencia* y *alerta de estado crítica*, están asociados a una medida porcentual (%). Por ejemplo, los siguientes son los parámetros definidos para el monitoreo de utilización de capacidad de un disco. (Presentado a nivel de sistema operativo como /u01).

Advertencia	85% de ocupación total de capacidad
Estado Crítico	95% de ocupación total de capacidad

Para ejemplificar, para en caso de que el estado de capacidad de un disco duro, no alcance un 85 % de utilización, las áreas encargadas de realizar el monitoreo de los recursos de Sistema Operativo nunca tendrán una notificación que les indique que se está llegando al máximo de su capacidad. Por lo tanto, no sabrán que se encuentran en una condición de borde operacional.

Como se aprecia en el siguiente cuadro, la ocupación de almacenamiento diario de un punto de montaje a nivel de Sistema Operativo, se mantiene cercana a la definición de alerta de advertencia (85 % de utilización) pero las áreas encargadas de realizar el monitoreo, nunca serán notificadas porque el valor de utilización de espacio en disco nunca llega al valor definido en la alerta.

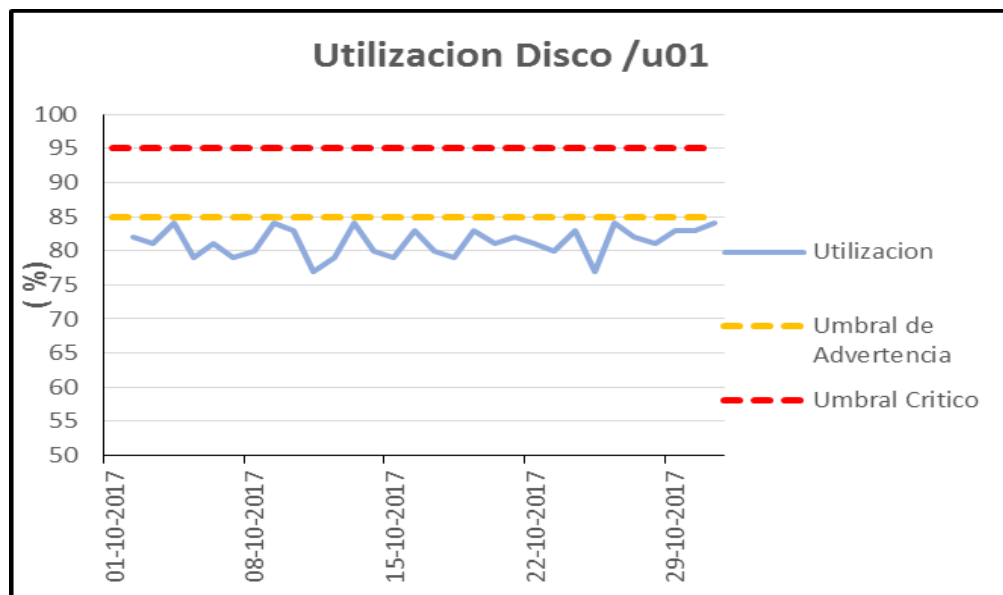


Figura 1: Ejemplo Utilización de Disco

El área de *e-commers* de una empresa del rubro *retail*, necesita implementar un monitoreo de su página web, ya que en la actualidad con las herramientas de monitoreo que tiene configurado, el negocio no puede conocer en tiempo real el comportamiento que presenta el proceso de venta en línea de productos mediante su página web en los

periodos de promociones web en los que participa (*CyberMonday*, *Black Friday*, *Cyberday*).

De acuerdo a esto, el planteamiento está enfocado a trabajar 2 problemas principales:

1. Determinar el cálculo matemático para medir la disponibilidad de la página web
2. Obtener indicador de disponibilidad y tiempo de respuesta de una página web a partir de un proceso de experiencia usuario.

1.2 Objetivos de la investigación

Como se ha expuesto anteriormente, el objetivo del trabajo es el de Diseñar un mecanismo de evaluación de disponibilidad y tiempo de respuesta de página web, esto se realizará mediante la integración de herramientas *Splunk* y el desarrollo de código de programación en lengua *Java* con la biblioteca *Selenium* que nos permita ver en tiempo real su comportamiento.

El estudio se realizará sobre los datos recolectados en un monitoreo sintético en la página web.

La investigación estará centrada en 4 objetivos principales:

- 1- Comprobar la posibilidad de diagnosticar en tiempo real la disponibilidad de una página web.
- 2- Crear un flujo de navegación de la compra de un producto perteneciente a un catálogo dentro de una página web.
- 3- Crear un monitoreo sintético mediante las librerías *selenium* para emular la acción de compra de un producto perteneciente a un catálogo.
- 4- Crear un panel de control con la herramienta *Splunk Enterprise*, donde se pueda ver en tiempo real la disponibilidad de una página web

1.3 Justificación

Actualmente, cada vez son más las empresas que optan por tener sistemas informáticos dentro de su organización, para prestar apoyo en la operación de las distintas áreas que componen una compañía.

En el caso específico de las empresas que realizan ventas de productos desde páginas web es de vital importancia mantener monitoreados y controlados los medios técnicos que componen la infraestructura que soporta el servicio de venta en línea, con estos monitoreos se puede garantizar desde los componentes de infraestructura el éxito del proceso de venta o notificar supuestas complicaciones que pueda tener dicho proceso.

A continuación se presentan las justificaciones para desarrollar la investigación:

- Monitorear en tiempo real la interfaz web que presta servicio de venta de productos. Permitiendo al negocio minimizar perdidas tanto de aspectos financieros, como en la imagen hacia los usuarios finales que requieren un servicio estable y siempre disponible.
- Disminuir incidentes y/o problemas causados por eventos inesperados, generando una herramienta proactiva más que reactiva, dado que no se basa en métricas de recursos, sino que en el comportamiento de un proceso de negocio.
- Al manejar los datos con la herramienta *Splunk* se puede procesar una mayor cantidad de información en un menor tiempo de procesamiento.

1.4 Hipótesis

Se plantea en el presente trabajo, la hipótesis de que es posible obtener un conocimiento del estado de la infraestructura web a partir de un proceso de negocio como es la venta en línea de productos de catálogo por medio de una página web.

El estudio debe enfocarse principalmente en 4 aristas:

- Determinar la posibilidad de que mediante a un cálculo matemático se puede conocer la disponibilidad de una plataforma web que interprete de forma correcta los límites de operación esperados por el negocio.
- Construir el medio técnico para extraer y procesar la información recopilada a partir de un proceso de negocio.
- Obtener un medio técnico, para desplegar la información de forma simple y en tiempo real.
- Demostrar que mediante la integración del producto *Splunk Enterprise* y el desarrollo en lenguaje *Java* la *biblioteca Selenium* se puede realizar un monitoreo de proceso de negocio de venta en línea.

2 MARCO TEORICO

El planteamiento del trabajo, es determinar ciertamente que esta empresa de *retail* mediante un monitoreo sintético, emulando los flujos de navegación que realizan sus clientes a través de su portal web para la compra de un producto de catálogo, pueda visibilizar la disponibilidad y tiempo de respuesta de la página web, como también el comportamiento de los sub-módulos web que componen el proceso de venta de productos en línea.

Al mantener monitoreados los flujos del proceso de venta de un producto, se podrá tener un indicador respecto al tiempo de respuesta con la que un cliente está navegando en los distintos módulos que componen la página web.

Esto permitirá al área encargada de la revisión del monitoreo de la plataforma web, tener un indicador adicional acerca del comportamiento de la infraestructura y los componentes que integran los sitios web, respecto al que aportan los softwares de monitoreo que se encuentran configurados. Actualmente existe un estricto control operacional de todos los componentes informáticos que componen su infraestructura.

A saber, a nivel de servidores y sistema operativos se realiza un monitoreo de la carga de Cpu, Memoria, Swap, tráfico en las tarjetas de red y *uptime* de los servidores web.

A nivel de telecomunicaciones se monitorea la disponibilidad de los equipos del tipo *switch* y *router*, velocidad de proveedores de internet y otros componentes dentro de los sitios que albergan los servidores web, pero esto no es suficiente para determinar si el proceso de negocio de venta de productos de catálogo en línea está presentando lentitud en alguna de sus fases.

El problema del monitoreo de componentes a partir de softwares de monitoreo, se encuentra en que, mientras los valores asociados a las métricas de recursos, no sobrepasen los umbrales definidos, las áreas encargadas de mantener controlada la disponibilidad de la plataforma, no tendrá visibilidad con respecto al tiempo de navegación de los usuarios dentro del portal web al momento de realizar una venta.

2.1 Área de Operaciones T.I.

Se le llama área de Operaciones T.I. al conjunto de personas que trabajan dentro de una gerencia de tecnología, en las que sus responsabilidades están plenamente relacionadas con el monitoreo y supervisión de la infraestructura tecnológica, siendo responsable de la continuidad operativa de los servicios informáticos de la organización (redes, bases de datos, Sistemas Operativos, Aplicaciones) y que esta funcione sin problemas.

2.2 Tecnologías de Información

Según la *Information Technology Association of America* (Asociación Americana de las Tecnologías de la Información: ITAA) TIC (ITC en inglés) las tecnologías de información es el estudio, el diseño, el desarrollo, el fomento, el mantenimiento y la administración de la información por medio de sistemas informáticos, esto incluye todos los sistemas informáticos no solamente la computadora, este es solo un medio más, el más versátil.

2.3 Infraestructura T.I.

La infraestructura de T.I. consiste en un conjunto de dispositivos físicos y/o lógicos, un conjunto de servicios, además de las aplicaciones de software que se necesitan para que una organización pueda operar. Abarcan capacidades tanto humanas como técnicas.

Los componentes de la infraestructura T.I. son las plataformas de hardware, sistemas Operativos, softwares empresariales, administración y almacenamiento de datos, redes y telecomunicaciones, internet.



Figura 2: Componentes Infraestructura T.I.

2.4 *Página Web*

Una página web, también conocida como página electrónica o página digital, es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes, y muchos otros elementos, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web.

Esta información se encuentra generalmente en formato como HTML o XHTML (no siendo los únicos), y puede proporcionar acceso a otras páginas web mediante enlaces de hipertexto. Frecuentemente también incluyen otros recursos como pueden ser hojas de estilo en cascada, guiones (scripts), imágenes digitales, entre otros.

Las páginas web pueden estar almacenadas en un equipo local o en un servidor web remoto. El servidor web puede restringir el acceso únicamente a redes privadas, por ejemplo, en una intranet corporativa, o puede publicar las páginas en la World Wide Web. El acceso a las páginas web es realizado mediante una transferencia desde servidores, utilizando el protocolo de transferencia de hipertexto (HTTP).

2.5 Sistema Informático

Un sistema informático (SI) es un sistema que permite almacenar y procesar información, es el conjunto de partes interrelacionadas tales como hardware, software y personal informático.

El hardware incluye computadoras o cualquier tipo de dispositivo electrónico, que consisten en procesadores, memoria, sistemas de almacenamiento externo, etc.

El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos.

Por último, el soporte humano incluye al personal técnico que crean y mantienen el sistema (analistas, programadores, operarios, etc.) y a los usuarios que lo utilizan.

2.6 Tiempo de respuesta

Tiempo de Respuesta en informática es el lapso de tiempo que transcurre entre que un usuario hace una petición a la red y la información pedida es recibida por éste.

En Internet depende de múltiples factores, tales como ancho de banda, calidad de la línea, velocidad de módem, congestión de la red. Por definición un usuario nunca está satisfecho con el tiempo de respuesta de la red y se acostumbra rápidamente a las mejoras de éste.

2.7 Plataforma Tecnológica

Se conoce por Plataformas Tecnológicas al conjunto de hardware y software que permiten recolectar, procesar, analizar y distribuir información para un propósito específico, son creadas por las grandes compañías de tecnología.

Permiten realizar operaciones en nanosegundos, almacenar gran cantidad de información y que sea fácilmente accesible.

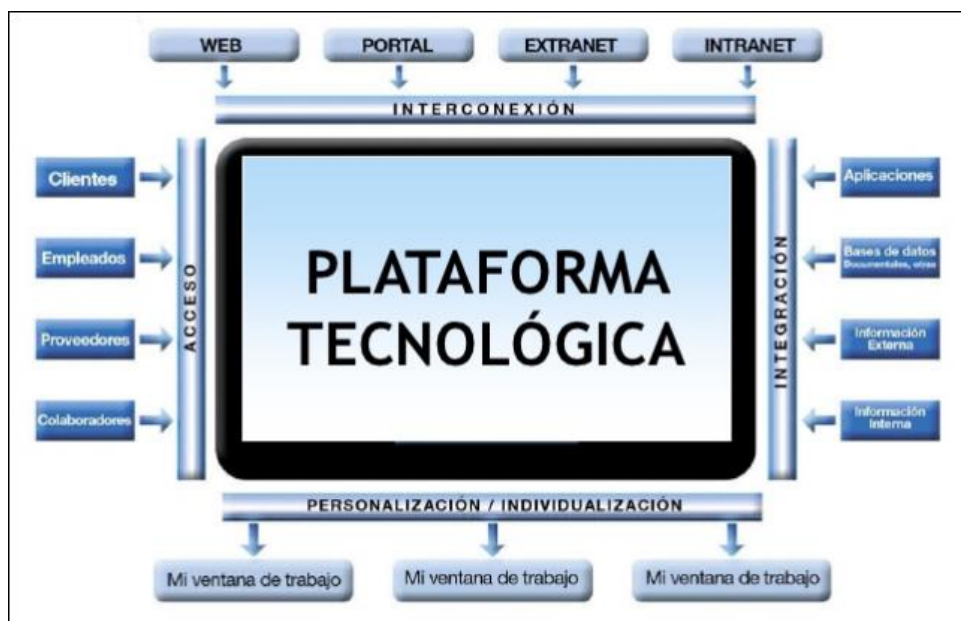


Figura 3: Plataforma Tecnológica

2.8 Ingeniería de sistemas

La Ingeniería de Sistemas es un campo de la ingeniería que se encarga del diseño, la programación, la implantación y el mantenimiento de sistemas. Utiliza un enfoque interdisciplinario que permite estudiar y comprender la realidad, con el propósito de implementar u optimizar sistemas complejos. La Ingeniería de Sistemas no construye

productos tangibles, sino sistemas abstractos mediante el uso de metodologías de la Ciencia de Sistemas.

2.9 Software de monitoreo

El software de monitoreo es principalmente un tipo de software de seguridad y vigilancia instalado en un sistema individual, bases de datos, sistemas operativos, cluster's o de redes corporativas.

Se puede tratar tanto de una aplicación independiente, o una función como parte del software o hardware de firewall, software antivirus o un paquete de software de seguridad de la información.

También notifica si identifica cualquier infracción o incumplimiento que dé como resultado un comportamiento anormal del sistema, actividad del usuario o flujo de red. Además de auditar a los empleados o las actividades que estos realizan dentro de una red corporativa.

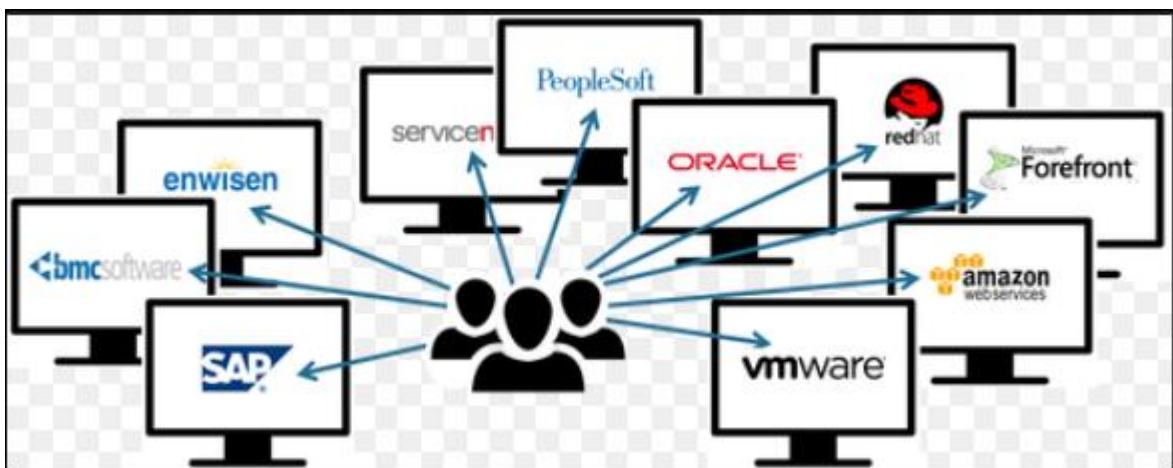


Figura 4: Software de Monitoreo

2.10 Componente

Se llama componente a cualquier artefacto tecnológico que desempeñe una tarea dentro de un servicio informático. Por ejemplo, si tomamos como referencia un sistema de ventas en línea, uno de sus servicios informáticos sería los servidores web. Estos funcionan alojados en servidores de aplicación, equipos de comunicaciones y balanceadores de carga, cada uno de estos se considera un componente dentro de una infraestructura que posee recursos o tópicos interesantes de ser medidos.

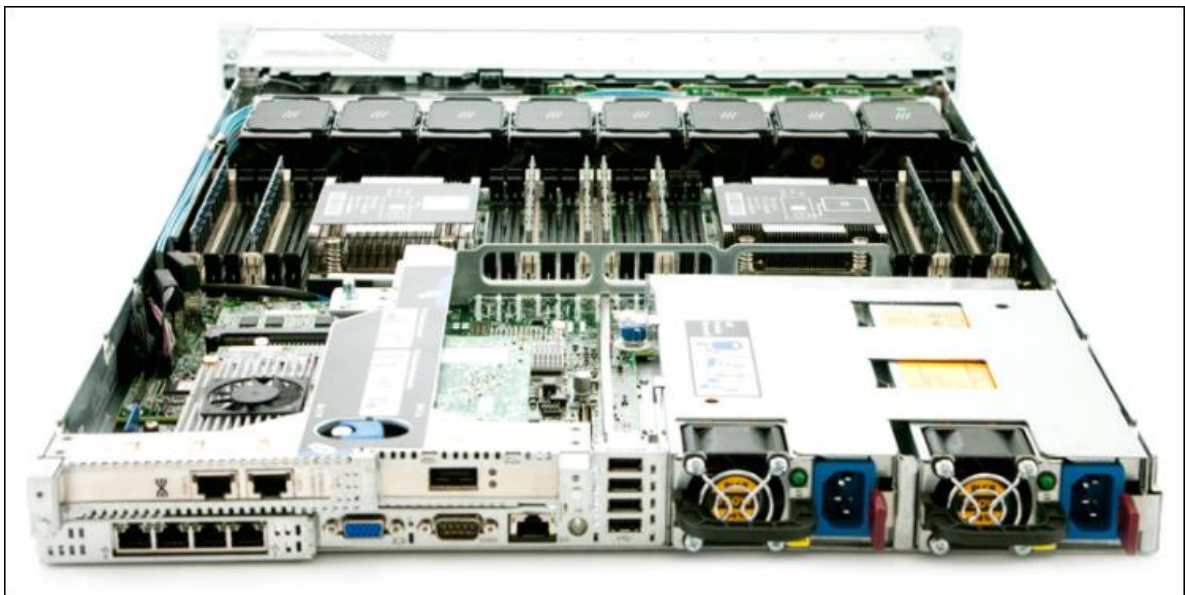


Figura 5: Componentes de un servidor

Los servidores presentan variados tópicos que resulta interesante de medir con la finalidad de tener la mayor cantidad de antecedentes posibles en caso de algún problema, los recursos que generalmente se monitorean son el consumo de Memoria, consumo de CPU, consumo de Disco, Trafico de Red.

2.11 Métricas

Las métricas son valores numéricos que son representados generalmente en una medida de porcentaje (%) en la utilización de un componente en un momento determinado. A continuación un ejemplo de lo que encontramos usualmente en las mediciones de estructuras de memoria de base de datos Oracle a través de su monitoreo estándar.

Porcentaje de utilización de las estructuras de memoria en una base de datos relacional en función del tiempo, donde los umbrales están definidos en 95% para una advertencia (warning) y 90% para eventos críticos.

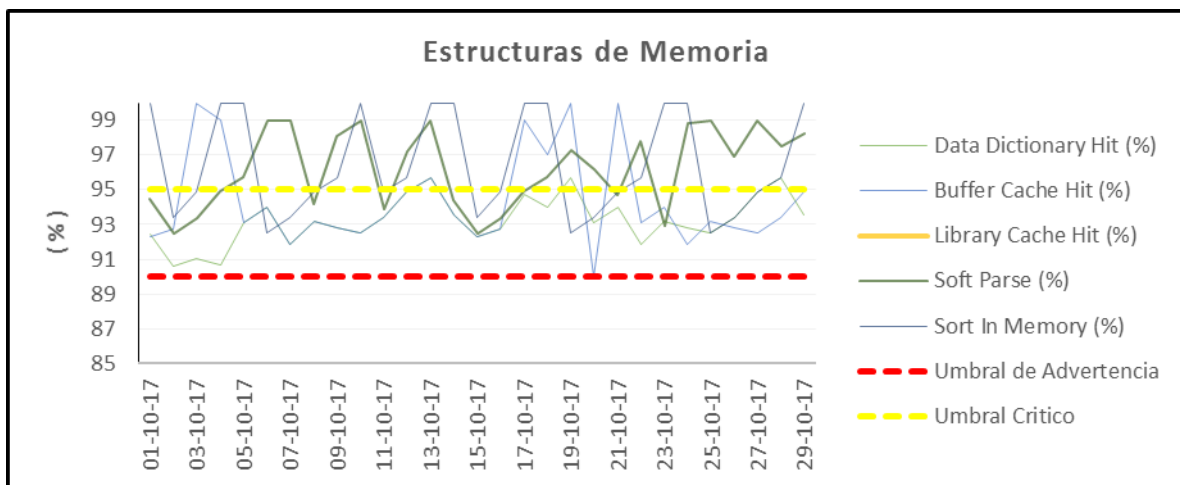


Figura 6: Comportamiento estructuras de memoria de una Base de datos

2.12 Portal web

Un portal web es un sitio web que ofrece al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Incluye: enlaces webs, buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente un portal en Internet está dirigido a resolver necesidades de información específica de un tema en particular.

2.13 Splunk Enterprise

Splunk Enterprise es un *software* perteneciente a la compañía de desarrollo *Splunk*, enfocado en la inteligencia operacional, para ser ejecutado en sistemas operativos tanto *Microsoft* como *Linux*, es utilizado para controlar, informar y analizar datos que generan tanto los dispositivos físicos de una organización, como también las aplicaciones.

Puede procesar información proveniente de sitios web, dispositivos móviles, cámaras de seguridad sensores, servidores, redes, entre otros, todo a partir de archivos *log's* que contengan el tipo de dato *tymestamp*.

Su objetivo, es hacer que los datos de estas máquinas puedan ser conocidos por cualquier área dentro de la compañía, y a partir de esta información poder generar patrones de comportamiento y/o tendencias, establecer medidas de prevención, diagnosticando problemas en cualquier área de negocio en la que se tenga configurado este tipo de extracción de información.

A partir de la supervisión y el análisis de todos los aspectos que componen una infraestructura ya sean físicos o lógicos, atreves del seguimiento de clics con la que un usuario interactúa con una página web, las transacciones que realiza un usuario en una aplicación, hasta los eventos de seguridad y la tráfico en la red, además de un sin número de otros aspectos, pueden ayudar a una compañía a obtener información que siendo trabajada mediante técnicas para indexar y correlacionar una gran cantidad de información en tiempo real, almacenándola en un repositorio para la generación de gráficos, alertas y paneles de fácil definición para el usuario la convierten en inteligencia operacional.

Los componentes de *Splunk Enterprise* están enfocados en una o más funciones específicas, como la indexación o la búsqueda de información, o el componente de envío de información desde la fuente de datos hasta la base de datos central, que es

una versión liviana de *Splunk Enterprise* con un ejecutable independiente el cual debe ser instalado en el servidor donde se realiza el monitoreo sintético.

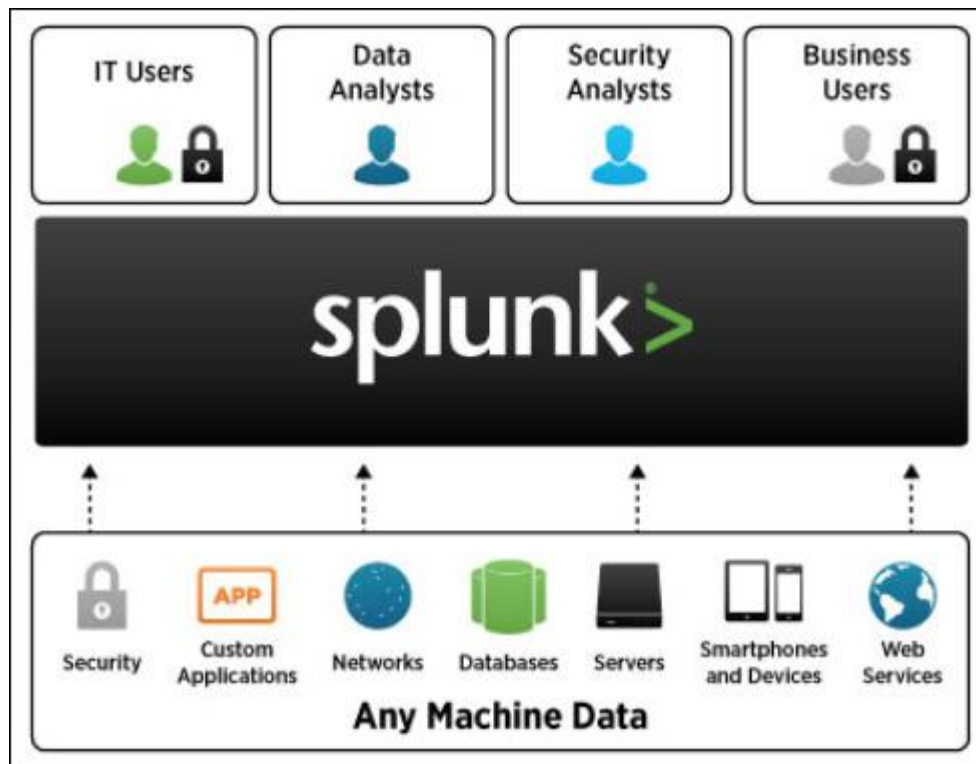


Figura 7: Splunk

Dentro de la arquitectura de *Splunk Enterprise* existen tres tipos de componentes principales, y son los siguientes:

Indexador: es el componente que proporciona el almacenamiento de los datos tanto locales como remotos, es quien contiene los datos que provienen desde las distintas fuentes de información (*log's*), para indexarlos y disponibilizarlos dentro de *Splunk Enterprise*, las funciones del indexador son las siguientes:

- Almacena datos por diversos protocolos y puertos de entrada.
- Conecta *Splunk Enterprise* de forma remota mediante script's a las aplicaciones, bases de datos y *Api's* para consultar e indexar la información

- Realiza el filtro de la información seleccionada para su almacenamiento.
- Es el encargado de gestionar las bases de datos donde se inserta la información
- Comprime los datos para reducir los recursos de almacenamiento
- Tiene la capacidad de habilitar búsquedas remotas y en paralelo para que sean más óptimas.

Search Header: Es el componente que tiene la tarea de realizar el procesamiento de la información durante el proceso de búsqueda, A nivel de configuración, mientras mayor cantidad de servidores de *Search Header* se tengan, más eficiente es la búsqueda de información, algunas de las funciones que realiza este componente son las siguientes :

- Es el proceso encargado de la Programación de alertas.
- Es Tiene la tarea de la generación de configuraciones, aplicaciones, consultas, informes, entradas de datos, búsquedas distribuidas, dashboards, licencia, usuarios y roles.
- Es el Administrador de los campos y los datos.
- Controla las funcionalidades del indexador y sus tareas programadas.
- Realiza tareas de balanceo de carga de las consultas en cada Indexador.
- Gestiona las consultas simultáneas entre usuarios, optimizando la entrega de la información de los informes y los dashboards.
- Es el encargado de concentrar configuraciones, realizar backups y rollback de configuraciones en minutos.
- Al tener un concentrado de configuraciones se evitar replicar configuraciones en cada Indexador.

Splunk Forwarder. Más que un componentes, es un agente que se instala en la fuente de datos, dependiendo del Sistema Operativo es la instalación y configuración que se debe realizar, se encarga de enviar los datos desde las fuentes de datos (páginas web, servidores, aplicaciones, etc.) hasta la base de datos *Splunk Enterprise* para que sean indexados de acuerdo a las reglas definidas por el negocio en el componente indexado.

2.14 Biblioteca Selenium

La biblioteca *Selenium* es un *framework* que es utilizado para realizar pruebas automáticas de páginas y aplicaciones con interfaz web, compatible con la mayoría de navegadores existentes en el mercado como *Internet Explorer*, *Mozilla Firefox*, *Google Chrome*, *Safari* y sistemas operativos como *Windows* y *Linux* ambos los de mayor uso en las áreas de plataformas TI de producción. Está basado en *JavaScript* y ejecuta las clases y métodos de *Java* en un navegador real, inyectando instrucciones o datos en las páginas *HTML*. Fue creada por Jason Huggins en el año 2004 como parte de un proyecto interno de la empresa *ThoughtWorks* y mejorada en el año 2006 por el ingeniero de *Google* Simon Stewart.

A partir de esta librería, se puede automatizar el uso del navegador, y con esta funcionalidad las posibilidades de uso están abiertas a realizar distintas acciones dentro de una página o aplicación web. Si bien el objetivo principal de esta herramienta es la automatización de pruebas sobre aplicaciones web, su uso no es exclusivo para esta actividad.

Entre sus principales características podemos destacar que tiene una Facilidad de registro y ejecución de las pruebas, referencia a objetos *DOM* en base al *ID* del objeto, nombre o a través de *XPath*, las acciones programadas pueden ser ejecutadas paso a paso y el resultado de estas pueden ser almacenados en distintos formatos.

Incluye un lenguaje específico de dominio de nombre *Selenese* para escribir las acciones programadas en una variada cantidad de lenguajes de programación populares incluyendo *Java*, *C#*, *Ruby*, *Groovy*, *Perl*, *Php* y *Python*.



Figura 8: Selenium Navegadores

Selenium nos provee de 3 herramientas Selenium Ide, Selenium Rc y Selenium Driver.

Selenimun IDE: es la herramienta que permite cambiar el código de programación al que uno desee, pueden ser *Java*, *C#*, *Ruby*, *Groovy*, *Perl*, *Php* y *Python*. Los scripts son generados en *Selenese*, lenguaje de scripting nativo de *Selenium*, que es el encargado de proveer de comandos que realizan las acciones sobre objetos dentro del navegador (botones, textbox, checkbox entre otros), seleccionar una opción dentro de una lista, verificar un texto dentro del navegador, realizar un *click* en un enlace.

Selenium Remote Control (RC): Es un servidor escrito en Java el cual acepta instrucciones al navegador de forma remota mediante el protocolo *HTTP*. Es el encargado de escribir las pruebas automatizadas para las aplicaciones web.

Selenium Driver: Es la evolución de *Selenium Remote Control*, ya que acepta instrucciones generadas tanto de *Selenese* como de *Api* de cliente para ser enviadas al navegador.

Es implementado mediante un controlador nativo de cada navegador como *mozilla*, *Chrome*, *Internet Explorer*. Debido a que *Selenium* tiene propias restricciones de *Javascript* (lenguaje con el que fue construido), *Selenium Driver* aporta valiosas funcionalidades ya que dependiendo del navegador que se desea probar utiliza el mecanismo más apropiado.

Por ejemplo en *Firefox* se implementa como una extensión, para *Internet Explorer* hace uso de los objetos propios de automatización. Además puede hacer uso de las características de los navegadores desde el punto de vista del sistema operativo.

2.15 Ingeniería de software

La ingeniería de software se refiere a la aplicación de un enfoque sistemático, disciplinado y cuantificable en el ámbito del desarrollo, operación y mantenimiento de software, además del estudio de estos enfoques, es decir, el estudio de las aplicaciones de la ingeniería al software. Integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.

2.16 Panel de control

Un panel de datos (en inglés *dashboard*) es en donde las empresas pueden visualizar su información más relevante, es decir, es una representación gráfica de los principales indicadores que intervienen en la obtención de los objetivos claves de negocio, está orientada a la toma de decisiones permitiendo la optimización de la estrategia de la empresa.

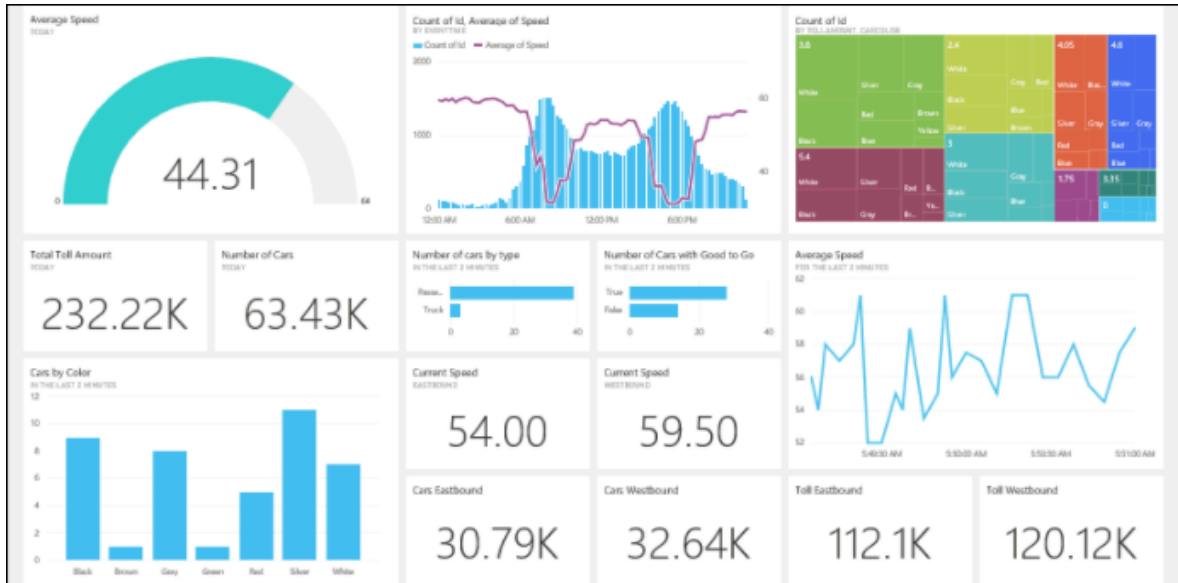


Figura 9: Panel de Control

Un panel de control lo que hace es transformar los datos en información y así mejorar la toma de decisiones. Es algo así como un “resumen ejecutivo” visual y gráfico sobre el estado actual de aquello que se desea monitorear.

2.17 Monitoreo Sintético

Un monitoreo sintético es la supervisión del sitio web que se realiza mediante una emulación de navegador web o grabaciones de ejecución de transacciones web.

Los scripts de rutas son creados para simular una acción o ruta que un cliente o usuario final debe seguir dentro de un sitio. Esas rutas son supervisadas continuamente en intervalos específicos para el rendimiento, tales como: funcionalidad, disponibilidad y medidas de tiempo de respuesta.

Esta técnica es valiosa porque permite a las áreas de infraestructura identificar problemas y/o determinar si un sitio web o aplicación web presenta lentitud o experimenta inactividad antes que el problema afecte al cliente/usuario final.

Una de las tantas ventajas que aporta este tipo de monitoreo es que permite a las organizaciones realizar pruebas de aplicaciones web en una modalidad 24x7, o realizar pruebas de nuevas funcionalidades antes que sean liberadas en el ambiente de producción.

El monitoreo de las plataformas tecnológicas es muy importante para mantener controlada la continuidad operativa de los negocios, la detección oportuna de incidentes, fallas en su comportamiento o caída de servicios críticos para cada organización, el monitoreo de los aplicativos resulta fundamental dado que es ahí donde se encuentra lo que aporta valor al negocio de cara a un usuario final, más aun si existe la necesidad del negocio de mantenerlos operativos en modalidad 24x7.

Para los encargados de los sistemas que soportan la infraestructura web resulta fundamental conocer las métricas que les permitan determinar las cargas de trabajo óptimas de los recursos con que cuenta su infraestructura web.

El monitoreo sintético permite simular el comportamiento de un usuarios ejecutando un subconjunto de pruebas automatizadas de una aplicación web, monitoreando continuamente y en intervalos específicos de tiempo para obtener datos de funcionalidad, disponibilidad y mediciones de tiempo de respuesta.

Al evaluar la calidad de las aplicaciones mediante pruebas de carga permite al negocio optimar sus estrategias en periodos de promociones web para así ofrecer una mejor experiencia a los usuarios y con ellos mejorar las ventas.

2.18 Experiencia Usuario

El concepto de experiencia usuario tiene su origen dentro de las áreas de Marketing digital, y se relaciona a partir de la interacción de un usuario mediante un dispositivo electrónico el que puede ser un computador, un tablet's o un teléfono inteligente, con una marca, producto o servicio.

Es el proceso que se encarga de la satisfacción del usuario mejorando la usabilidad, la accesibilidad y la interacción entre el usuario y el producto por intermedio de los dispositivos físicos, con la finalidad de cumplir con los objetivos planteados por el negocio, enfocado desde la perspectiva del usuario en cada parte del proceso de realización de un producto para obtener datos tanto cuantitativos como cualitativos para posterior análisis de estos.

La experiencia de usuario ha evolucionado desde el marketing digital hacia los sistemas informáticos presentándose particularmente en los entornos que tienen relación con las páginas web y dispositivos móviles.

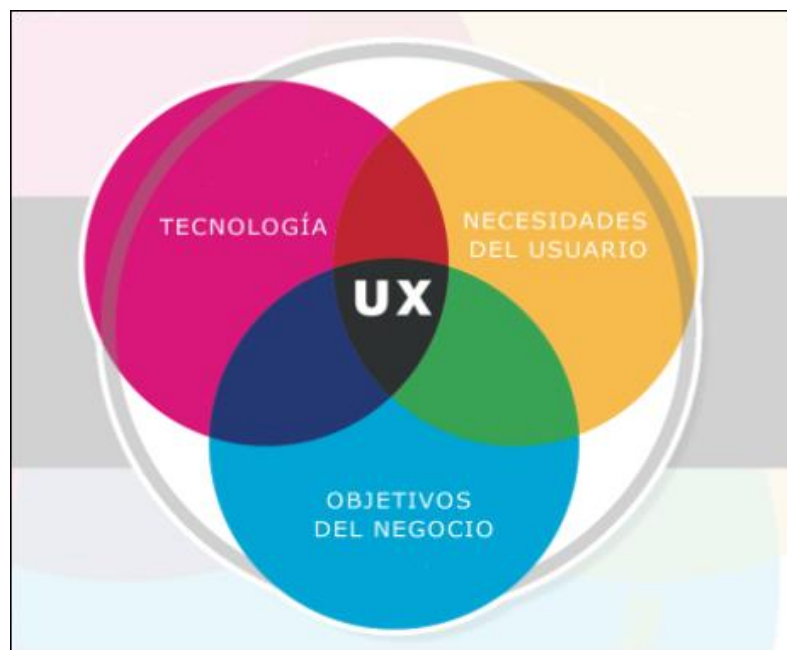


Figura 10: Conceptos experiencia usuario

3 DESARROLLO

3.1 Metodología

Para la realización del presente trabajo se utilizara una metodología empírica y cálculos matemáticos, abordando estudios al comportamiento de un flujo de navegación web definida en un plan de experiencia usuario implementada mediante un monitoreo sintético, para intentar obtener un medio de interpretación que sea de utilidad respecto al conocimiento que se tiene de ello.

Se realizaran cálculos matemáticos que permitan identificar la disponibilidad global de la página web, además de conocer el tiempo de respuesta para cada una de las etapas de la navegación definida en el monitoreo sintético.

De esta forma, se plantea el objetivo de determinar mediante un cálculo matemático la disponibilidad global de una plataforma web.

3.1 Solución Propuesta

Desde la mirada del monitoreo tradicional, los objetivos están puestos en el comportamiento de los componente de infraestructura y/o de los servicios que se ejecutan en los servidores web y/o los equipos de comunicación,

Se propone que mediante un proceso de negocio, - en este caso el proceso de venta en línea-, también se puede realizar un monitoreo de disponibilidad de una página web, no medirá el estado de un componente o de un servicio en particular dentro de los servidores que soportan la infraestructura web, pero si nos otorgara un indicador de disponibilidad, enfocado, no en la infraestructura propiamente tal, sino que en la experiencia del usuario al momento de realizar una compra dentro del sitio web.

Para determinar la disponibilidad y tiempo de respuesta de la página web, se propone el diseño de un proceso de experiencia usuario, donde se define estrictamente el flujo de navegación de un cliente al momento de adquirir un producto.

Dicho flujo de navegación será programado en métodos de lenguaje *Java* utilizando librerías *Selenium*, se configurara un robots que ejecute cada un minuto el proceso de experiencia usuario emulando el flujo de navegación que necesita realizar un cliente para comprar un producto de catálogo dentro de la página web.

El flujo de navegación correspondiente al proceso de experiencia usuario será dividido en etapas, con la finalidad de asignar ponderadores que representan la importancia que tienen para el negocio cada una de las etapas que componen la experiencia de navegación del usuario final, en este caso un cliente.

Es indispensable que para la correcta interpretación del resultado, el despliegue de la información obtenida sea de forma gráfica y simple, además que sea desplegada en tiempo real para ayudar a los responsables de la plataforma en la rápida, oportuna y correcta toma de decisiones.

3.2 Recolección de requerimientos

Parte importante para realizar cualquier diseño de una solución de software es la recolección de requerimientos, dentro de la ingeniería de software tenemos a la ingeniería de requerimientos como unas de las disciplinas fundamentales.

A través de los años se ha podido constatar que los requerimientos marcan el punto de inicio de actividades que en su evolución nos permitirán llegar a una solución específica.

Mediante la utilización de diversos métodos y técnicas podemos realizar planeamientos para definir estimaciones de tiempo y costos, también se puede realizar estimaciones para definir los recursos físicos, de software, de personas y principalmente en la elaboración de cronogramas que nos permiten generar mecanismos de control con los que se contará durante el proceso de desarrollo.

Para el desarrollo del presente trabajo la técnica utilizada será entrevistas a personas.

En estas entrevistas es importante definir todas las entradas, salidas y reglas del problema, también se debe incluir los documentos que serán parte del proceso para expresar lo deseado de la solución. Se trabaja con los usuarios involucrados tanto directa como indirectamente realizando entrevistas, con la finalidad de entender las necesidades y especificaciones de los requerimientos de sistema.

Cabe señalar, que en las entrevistas realizadas a los usuarios se definió entre otras cosas, tanto el software como el hardware necesario para desarrollar la solución, además del flujo y las etapas de navegación del monitoreo sintético, y la forma en cómo deben ser desplegados los datos.

3.3 Requerimientos de Software

Tanto en el diseño como en la Implementación del monitoreo se desarrollara con las herramientas existentes en la compañía que solicita el monitoreo, el motivo de esto es que, dada una fuerte inversión de recursos económicos se realizó una renovación tecnológica en su plataforma central, pensada en alinear a la compañía con los más altos estándares de tecnológica tanto de software como de hardware, con la finalidad de permitir una fácil administración y mantención de los sistemas existentes dentro de la compañía.

Esta compañía cuenta con una amplia gama de recursos computacionales construida bajo tecnología física como también de virtualización, para el desarrollo del proyecto se han otorgado tres servidores físicos, dos de ellos cuentan con sistema operativo Linux distribución Oracle y el servidor restante cuenta con sistema operativo Windows.

Se ha definido que la asignación de tareas en los servidores otorgados para el desarrollo del proyecto sea la siguiente:

SERVIDOR	TAREA
LnxSelenium	Propietario de métodos Java con librerías Seleniun
WinNavegacion	Robot de navegación sintética
LnxSplunk	Propietario de Splunk Enterprise

Figura 11: Asignación de tareas servidores

La siguiente es la distribución del software instalado en las maquinas que componen la solución.

LnxSelenium:

Para el servidor asignado a la tarea de ejecutar mediante una Java Virtual Machine los métodos javas programados en base a las librerías *Selenium*, el software que se necesita utilizar es el siguiente:

- Sistema Operativo : Oracle Linux 7 x86
- Tipo de Instalación : Stand Alone.
- Software : JDK 6
- Librerías : Selenium
- Splunk : Componente Splunk Forwarder

WinNavegacion:

Para el servidor que cumplirá el rol de robot para las tareas de emular el flujo de navegación de experiencia usuario el software a utilizar es el siguiente:

- Sistema Operativo : Microsoft Windows Server 2012 Release 2
- Service Pack : SP 2
- Navegador : Chrome
- Tipo de Instalación : Miembro del Dominio

LnxFusion:

En lo que respecta al servidor asignado para las tareas a realizar con *Splunk Enterprise*, el software a utilizar es el siguiente:

- Sistema Operativo : Oracle Linux 7 x86
- Herramienta : Splunk Enterprise
- Tipo de Instalación : Stand Alone

Todas las máquinas pertenecientes a organización cumplen con los requisitos necesarios para el correcto desempeño de los sistemas operativos instalados en ellos.

3.4 Requerimientos de Hardware

El nivel de hardware para realizar la implementación de la solución, está relacionado directamente con las necesidades de las herramientas de desarrollo antes mencionadas (*Splunk Enterprise* y librerías *Selenium*).

Adicionalmente a lo anterior en la compañía existe un estándar de hardware, donde las máquinas para la solución planteada deben presentar las siguientes características.

LnxSelenium:

Para el servidor asignado a la tarea de ejecutar mediante una *Java Virtual Machine* los métodos *Java* programados en base a las librerías *Selenium*, el hardware que se necesita utilizar es el siguiente:

- CPU : 4 intel xeon 2.4 Ghz; 512 Kb.
- RAM : 16 Gb.
- HD : 1 Tb.
- NIC : 1 Mbps

WinNavegacion:

Para el servidor asignado a las tareas de ejecución correspondiente al flujo de navegación el software a utilizar es el siguiente:

- CPU : 4 intel xeon 2.4 Ghz; 512 Kb.
- RAM : 16 Gb.
- HD : 1 Tb.
- NIC : 1 Mbps.

LnxSplunk:

En lo que respecta al servidor asignado para las tareas a realizar con el software *Splunk Enterprise*, el software a utilizar es el siguiente:

- CPU : 8 intel xeon 2.4 Ghz; 512 Kb.
- RAM : 255 Gb.
- HD : 5 Tb.
- NIC : 1 Mbps

Cabe destacar que los requerimientos de hardware se corresponden a las exigencias especificadas por los fabricantes tanto de *Splunk Enterprise*, como de librerías *Selenium*.

3.5 Creación de flujo de navegación

Para el desarrollo del monitoreo del proceso de negocio de venta de productos mediante un portal web, se ha creado un flujo de navegación y este corresponde a los pasos que debe realizar un cliente al momento de efectuar la compra de un producto dentro del portal web, Descripción grafica del flujo de navegación del proceso de experiencia usuario.

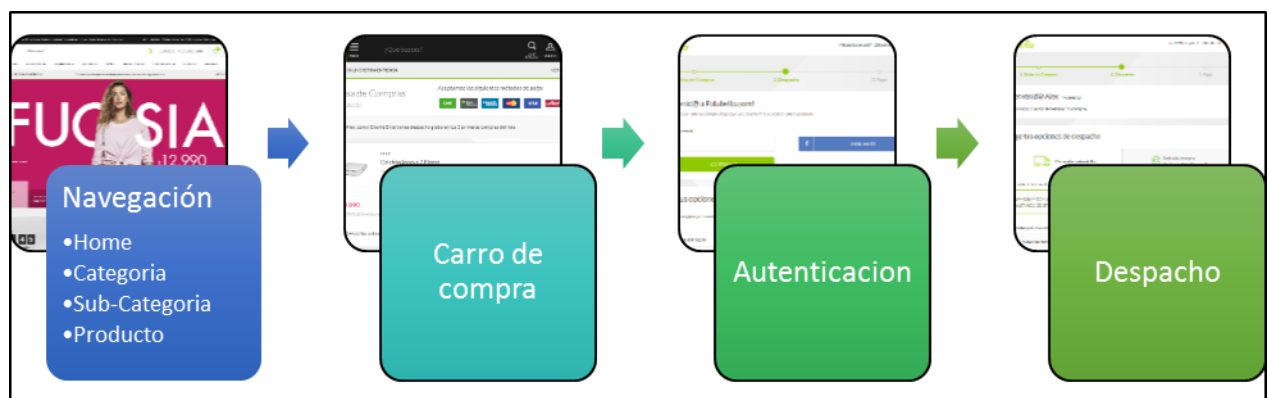


Figura 12: Flujo de navegación monitoreo sintético

De acuerdo a la información recopilada en el levantamiento de requerimientos, El flujo de compra se compone de 4 etapas, trata de que el usuario busque un producto determinado perteneciente a una categoría y su correspondiente subcategoría.

Por ejemplo, si el producto que el usuario desea adquirir es un refrigerador, este pertenece a la categoría Electrodomésticos y a la subcategoría Refrigeradores.

Como se trata de emular con precisión los pasos que debe realizar el usuario, una vez encontrado el producto este se debe agregar al carro de compra, autenticarse en el sistema y definir el tipo de despacho que desea. Con esta acción se cierra el flujo de navegación del proceso de experiencia usuario.

Como se indica anteriormente se definieron 4 etapas de navegación las que se dividen en los siguientes ítems:

- Navegación.
- Carro de Compra
- Autenticación
- Despacho

Por restricciones del negocio dentro del proceso de experiencia usuario no se realiza el pago del producto, ya que no cuentan con un proceso automático para reversar una compra.

Navegación:

La navegación, corresponde a la primera etapa del flujo, es el punto de partida del proceso de experiencia usuario, aquí las acciones que se realizaran en orden estricto y están divididas en 4 sub etapas siendo las siguientes:

- Direccionamiento al home de la página web
- Seleccionar una categoría del producto
- Seleccionar la sub-categoría
- Seleccionar el producto

Carro de compra:

Corresponde a la segunda etapa del flujo, en esta etapa es donde el proceso de experiencia usuario selecciona el producto que se desea comprar y realiza la acción de agregarlo al carro de compra.

Autenticación:

Corresponde a la tercera parte del flujo y en esta etapa el proceso de experiencia usuario debe realizar las acciones de autenticación mediante usuario y contraseña.

Despacho

Corresponde a la cuarta y última etapa del flujo de navegación y es donde el cliente debe seleccionar las opciones de despacho que desea, retiro en tienda o despacho a domicilio.

3.6 Definición de ponderadores

El cálculo de disponibilidad se realiza en base a las 4 etapas del flujo de venta y se compone a partir del uso de ponderadores.

A cada una de las etapas que corresponden al flujo de navegación se le asigna un porcentaje (%),

La primera etapa de navegación que es donde el monitoreo sintético realiza 4 acciones representa el 40 % del total de la disponibilidad.

La segunda etapa correspondiente a la acción de agregar el producto que se desea comprar al carro de compra, representa el 20 % del total de la disponibilidad.

La tercera etapa que corresponde a la autenticación del cliente representa un 20 % de total de la disponibilidad.

La etapa final correspondiente al despacho del producto representa un 20 % del total de la disponibilidad.

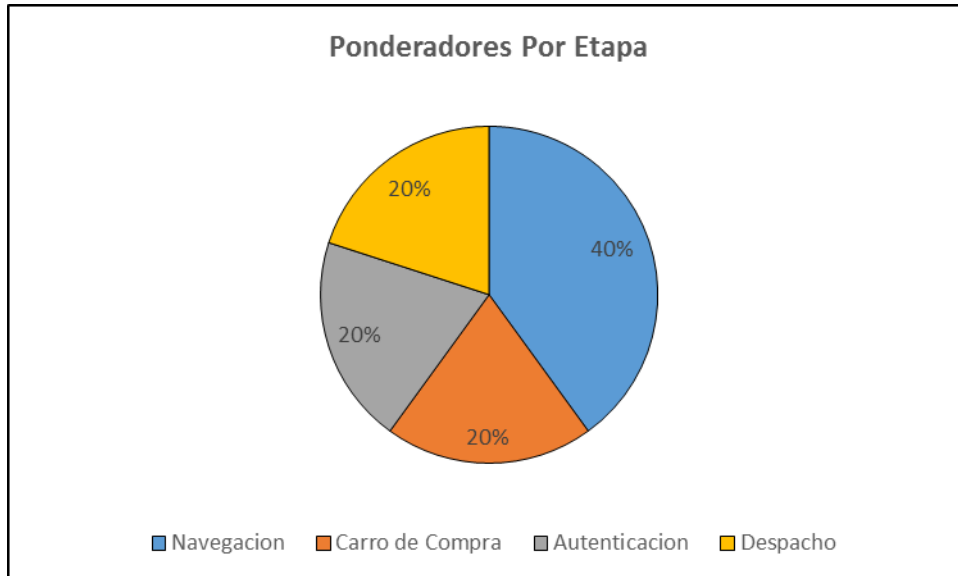


Figura 13: Ponderadores por etapas

Para el caso de la primera etapa del flujo de navegación, en la que en el monitoreo sintético realiza 4 sub etapas (ir al Home de la página, seleccionar una categoría de producto, seleccionar una subcategoría de producto y finalmente seleccionar el producto) y que corresponde un 40 % del total del flujo de navegación, se le asigna un ponderador a cada una de las sub etapas.

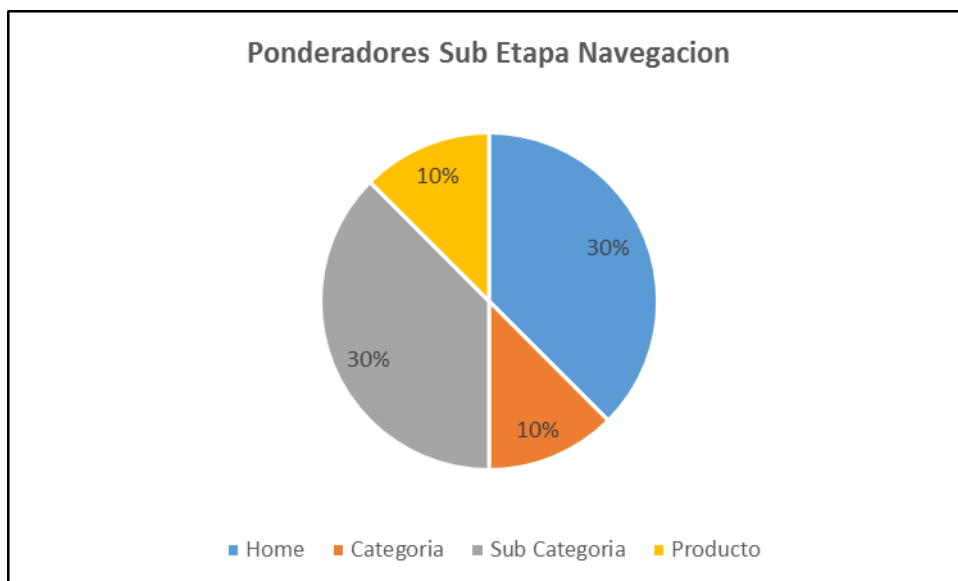


Figura 14: Ponderadores sub-etapas

La primera sub etapa de navegación que corresponde al direccionamiento del home de la página equivale al 30%.

La segunda sub etapa que corresponde a seleccionar una categoría equivale al 10%.

La tercera sub etapa que corresponde a una la selección de una subcategoría equivale a un 30%.

La última sub etapa que es la búsqueda del producto a comprar equivale a un 30%.

Adicionalmente a los ponderadores antes declarados, también debemos definir una variable donde se registra el estado de la página, (si fue cargada o no en un periodo de tiempo definido) en ella se indica si la navegación del usuario en cada una de las etapas y las sub etapas, que en total son 7, finalizan de forma exitosa o con error.

Teniendo definidos los valores de ponderación de cada una de las etapas y sub etapas que componen el flujo total del monitoreo sintético, más el estado de término de cada una de las etapas que componen el flujo de la navegación, nos encontramos en condiciones de poder calcular la disponibilidad de cada una de las etapas y sub etapas de navegación. Con ellos podemos conocer la disponibilidad global de la página web.

3.7 Definición de disponibilidad

Dado que uno de los enfoques declarados es determinar que los límites de operación sean los esperados por el negocio, y que la disponibilidad sea medida a partir de procesos de negocio, es que las áreas *e-commers* han declarado que esperan los siguientes umbrales

- Estado Crítico : Disponibilidad de 99.6 % o menor
- Advertencia : Disponibilidad entre 99.6% y 99.7%
- Operación Normal : Disponibilidad mayor a 99.8% de disponibilidad

Para la correcta interpretación de disponibilidad esperada por el negocio se ha determinado que el siguiente cálculo matemático es el más indicado

Disponibilidad

Cálculo

$$\Sigma$$

$$(Exitosos/Total*Ponderación)*100$$

Figura 15: Disponibilidad

Por ejemplo, a partir del proceso de experiencia usuario en el siguiente cuadro se presentan los valores que desde el monitoreo sintético se obtienen para un flujo de navegación de un tiempo determinado, el detalle de los valores para cada etapa del flujo de navegación son:

ETAPA	TOTAL	EXITOSOS	PONDERACION	% DISPONIBILIDAD
1.1 - Home	19	19	12 %	12,00
1.2 - Categoría	19	19	4 %	4,00
1.3 - Sub Categoría	19	16	12 %	10,10
1.4 - Producto	19	16	12 %	10,10
2 - Carro de Compra	19	16	20 %	16,84
3 - Autenticación	18	12	20 %	13,33
4 - Despacho	18	8	20 %	8,88
TOTAL				75,25

Figura 16: Ejemplo resultados monitoreo sintético

Podemos ver que tras varias ejecuciones del monitoreo sintético para un periodo de tiempo determinado, las etapas 1.3 Sub Categoría, 1.4 Producto, 2 Carro de Compra, 3 Autenticación y 4 Despacho no finalizaron de forma exitosa, con esto se obtiene una disponibilidad de 75,25% para ese periodo de tiempo, muy por debajo de lo esperado por el negocio.

A partir del resultado obtenido las áreas encargadas de la revisión de los monitoreos pueden determinar que la disponibilidad global del sitio no es la esperada por el negocio, además de conocer en cuál de las fases del proceso de negocio está generando problemas, con esta información pueden contrastar información con los monitoreos tradicionales que se tienen configurados.

3.8 Definición de tiempo de respuesta

Para el desarrollo del trabajo se consideran dos tiempos de respuestas uno global y otro por etapas

El indicador de respuesta global, muestra el tiempo total que demoró en ejecutarse en promedio cada ciclo completo del Flujo de Compra, de acuerdo a umbrales de tiempo definidos.

Los umbrales fueron definidos para cada un ciclo completo de navegación son los siguientes:



Figura 17: Umbrales tiempo de respuesta de un ciclo

El cálculo para obtener el tiempo de respuesta global de página web es el siguiente

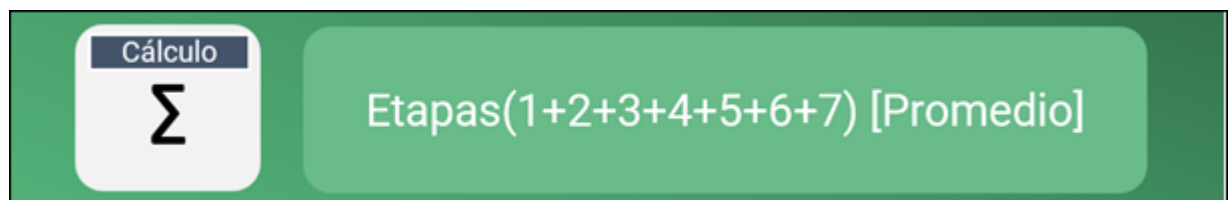


Figura 18: Calculo de tiempo de respuesta

El resto de los indicadores de tiempo de respuesta, muestran el tiempo total que demoró en promedio cada etapa del flujo de navegación realizada por el proceso de experiencia usuario, desde la etapa 1 que es posicionarse en el home de la página, hasta la etapa en que se define el despacho del producto.

Los umbrales definidos para cada etapa del ciclo completo de navegación son los siguientes:

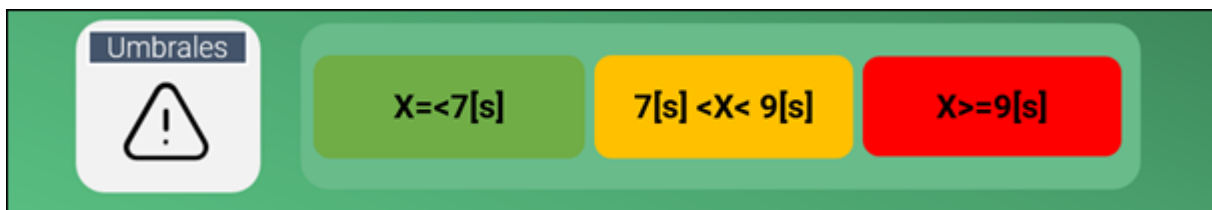


Figura 19: Umbrales ciclo completo

El cálculo para obtener el tiempo de respuesta de cada una de las etapas que componen el flujo

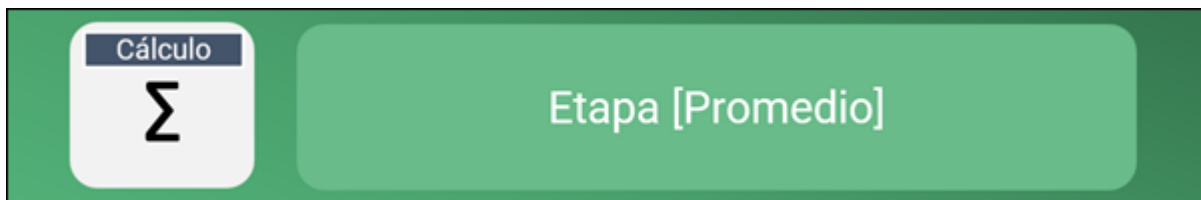


Figura 20: Calculo tiempo de respuesta de cada etapa.

3.9 Creación del monitoreo sintético

Para el desarrollo del presente trabajo se decide que la mejor alternativa para realizar el proceso de monitoreo sintético es la utilización del componente *Selenium Driver*, ya que cumple con la condición de realizar el desarrollo específico de clases y métodos *Java*.

Algunas de las razones para el uso de *Java* como lenguaje de programación se debe al hecho de que es un lenguaje multiplataforma, además no se necesita pagar licenciamiento por el producto, es un lenguaje en el que la mayoría de sus librerías son de código abierto y existe una amplia documentación del producto tanto en el sitio de Oracle como en comunidades de internet.

El siguientes es el código de programación que crea el monitoreo sintético:

```
package cl.bs2.webmonitor;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.Test;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Parameters;
import org.testng.Assert;
import org.testng.ITestResult;
import java.util.regex.Pattern;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.time.Instant;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collections;
import java.util.Enumeration;
import java.util.List;
import java.util.Properties;
import java.util.concurrent.TimeUnit;
import org.testng.annotations.*;
import static org.testng.Assert.*;
import org.apache.commons.io.FileUtils;
import org.apache.commons.lang3.time.StopWatch;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.remote.RemoteWebDriver;

public class TESTFALA {
    public static String siteRef;
    public static String data;
```

```
public static String browser;
public static String webDriverLocation;
public static String logFileCSV;
public static String logFileNative;
public static String screenshotsLocation;
public static String emailAddress;
public static String total;
public static String selenium_hub;
public static String pselenium_hub = "";
public static int timeout_home;
public static int timeout_cat;
public static int timeout_plp;
public static int timeout_pdp;
public static int timeout_cart;
public static int timeout_login;
public static int timeout_oms;
public static int timeout_clean;
public static String estado_home = "ERROR";
public static String estado_cat = "ERROR";
public static String estado_plp = "ERROR";
public static String estado_pdp = "ERROR";
public static String estado_cart = "ERROR";
public static String estado_login = "ERROR";
public static String estado_oms = "ERROR";
public static Long ts_home = 0L;
public static Long ts_cat = 0L;
public static Long ts_plp = 0L;
public static Long ts_pdp = 0L;
public static Long ts_cart = 0L;
public static Long ts_login = 0L;
public static Long ts_oms = 0L;
public static Long tiempo_home = 0L ;
public static Long tiempo_cat = 0L ;
public static Long tiempo_plp = 0L ;
public static Long tiempo_pdp = 0L ;
public static Long tiempo_cart = 0L ;
public static Long tiempo_login= 0L ;
public static Long tiempo_oms = 0L ;
public static int tiemponderado = 0;
public static boolean logready = false;
```

```
public static boolean continua = true;
public static boolean encuesta = false;
public static Long   inicia_sec = Instant.now().getEpochSecond();
public static String correo = "";

private static WebDriver driver;
private String baseUrl;
private boolean acceptNextAlert = true;
private StringBuffer verificationErrors = new StringBuffer();
StopWatch pageLoad = new StopWatch();
//MyLog registro = new MyLog(siteRef, logFileCSV, logFileNative);----

@BeforeMethod

@BeforeClass(alwaysRun = true)
public void setUp() throws Exception {
    if (driver == null) {
        loadRuntimeParams();
        System.out.println("Browser Seleccionado:" + browser);
        DesiredCapabilities capabilities = DesiredCapabilities.chrome();
        pselenium_hub="http://10.75.17.14:4444/wd/hub";
        System.out.println("Selenium-HUB: " + pselenium_hub );
        driver = new RemoteWebDriver(
            new URL(pselenium_hub),capabilities);
        driver.manage().window().setSize(new Dimension(1920,1080));
    }
    baseUrl = "http://www.falabella.com/";
}

@Test
@Parameters("emailAddress")
public void testFALACL(String emailAddress) throws Exception {
    // Inicio [1.-HOME]
    correo=emailAddress;
    pageLoad.start();
    driver.get(baseUrl + "/falabella-cl/");
    JavascriptExecutor jsx = (JavascriptExecutor)driver;

    checkpoll();
}
```

```
for (int second = 0;; second++) {
    if (second >= timeout_home) {continua=false; tearDown(); break; }
    checkpoll();
    try { if (isElementPresent(By.id("fb-overlay")))
{ts_home=Instant.now().toEpochMilli();estado_home="OK"; break; }} catch (Exception e)
{}
    Thread.sleep(1000);
    System.out.println("Esperando HOME-- "+second+ "/" + timeout_home);
}
pageLoad.stop();
tiempo_home=pageLoad.getTime();
System.out.println("HOME-"+estado_home+ " / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_home));
if (estado_home.equals("ERROR")) {continua=false;} else {tiemponderado = 12 ;}
// Fin [1.-HOME]]

// INICIO [1.1.-CATEGORIA]]
if (continua) {
pageLoad.reset();
pageLoad.start();
driver.get("http://www.falabella.com/falabella-cl/category/cat2020/Colchones");
for (int second = 0;; second++) {
    if (second >= timeout_cat) {continua=false; tearDown(); break; }
    checkpoll();
    try { if (isElementPresent(By.xpath("//a[contains(text(),'Colchón 2
Plazas')]"))) {ts_cat=Instant.now().toEpochMilli();estado_cat = "OK"; break;} } catch
(Exception e) {}
    Thread.sleep(1000);
    System.out.println("Esperando CAT-- "+second+ "/" + timeout_cat);
}
pageLoad.stop();
tiempo_cat=pageLoad.getTime();
System.out.println("CAT-"+estado_cat+ " / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_cat));
if (estado_cat.equals("ERROR")) {continua=false;} else {tiemponderado = 16 ;}
}
// FIN [1.1.-CATEGORIA]]

// INICIO [1.2./PLP]]
```



```
if (continua){
    pageLoad.reset();
    pageLoad.start();
    checkpoll();
    jsx.executeScript("window.scrollTo(0,900)", "");
    driver.get("http://www.falabella.com/falabella-cl/category/cat2020/falabella-
cl/category/cat90010/Colchon-2-Plazas?isPLP=1");
    for (int second = 0;; second++) {
        if (second >= timeout_plp) {continua=false; tearDown(); break; }
        checkpoll();
        try { if (isElementPresent(By.linkText("Colchón Ergo T 2 Plazas"))) {
ts_plp=Instant.now().toEpochMilli();estado_plp = "OK"; break; } } catch (Exception e) {}
        Thread.sleep(1000);
        System.out.println("Esperando PLP-- "+second + "/" + timeout_plp);
    }
    pageLoad.stop();
    tiempo_plp=pageLoad.getTime();
    System.out.println("PLP-"+estado_plp+" / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_plp));
    if (estado_plp.equals("ERROR")) {continua=false;} else {tiemponderado = 28 ;}
}
// FIN [1.2./PLP]]

// INICIO [1.3.-PDP]]
if (continua){
    pageLoad.reset();
    pageLoad.start();
    checkpoll();

    driver.get("http://www.falabella.com/falabella-cl/product/5035100/Colchon-2-plazas-
Art-6");

    for (int second = 0;; second++) {
        if (second >= timeout_pdp) {continua=false; tearDown(); break; }
        checkpoll();
        try { if (isElementPresent(By.cssSelector("button.fb-btn:nth-child(2)")))
{driver.findElement(By.cssSelector("button.fb-btn:nth-child(2))).click(); break;} }
catch (Exception e) {}
        Thread.sleep(1000);
        System.out.println("Esperando PDP 1/2 -- "+second+ "/" + timeout_pdp);
```

```
}

//System.out.println("paso el boton!");
for (int second = 0;; second++) {
    if (second >= timeout_pdp) {continua=false; tearDown(); break; }
    checkpoll();
    try { if (isElementPresent(By.cssSelector("#fb-modal-add > div.fb-added-to-
basket_footer > div.fb-added-to-basket_ctas > a")))
{ts_pdp=Instant.now().toEpochMilli(); estado_pdp = "OK"; break; } } catch (Exception e)
{}
    Thread.sleep(1000);
    System.out.println("Esperando PDP 2/2-- "+second+ "/" + timeout_pdp);
}

pageLoad.stop();
tiempo_pdp=pageLoad.getTime();
System.out.println("PDP-"+estado_pdp+ " / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_pdp));
if (estado_pdp.equals("ERROR")) {continua=false;} else {tiemponderado = 40 ;}
}
// FIN [1.3.-PDP]]

// INICIO [2.-CART]]
if (continua){
pageLoad.reset();
pageLoad.start();
checkpoll();
driver.findElement(By.cssSelector(".fb-overlay_inject > div:nth-child(1) >
div:nth-child(1) > div:nth-child(3) > div:nth-child(2) > a:nth-child(1)")).click();
for (int second = 0;; second++) {
    if (second >= timeout_cart) {continua=false; tearDown(); break; }
    checkpoll();
    try { if (isElementPresent(By.cssSelector(".fb-btn"))) {
ts_cart=Instant.now().toEpochMilli();estado_cart = "OK"; break; } } catch (Exception e)
{}
    Thread.sleep(1000);
    System.out.println("Esperando CART 1/2 -- "+second+ "/" + timeout_cart);
}

pageLoad.stop();
```

```
tiempo_cart=pageLoad.getTime();
System.out.println("CART-"+estado_cart+" / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_cart));
if (estado_cart.equals("ERROR")) {continua=false;} else {tiemponderado = 60 ;}
}
// FIN [2.-CART]]

// INICIO [3.-LOGIN]]
if (continua){
pageLoad.reset();
pageLoad.start();
System.out.println("LOGIN!!!");

driver.findElement(By.xpath("/html/body/div[2]/main/div/div[2]/div[3]/div/div[1]/form/div[2]/div[2]/div[2]/button")).click();
//driver.findElement(By.cssSelector(".fb-btn")).click();

for (int second = 0;; second++) {
if (second >= timeout_login) {continua=false; tearDown(); break; }
checkpoll();
try { if (isElementPresent(By.id("emailAddress"))) { break; } } catch
(Exception e) {}
Thread.sleep(1000);
System.out.println("Esperando LOGIN 1/3 -- "+second+ "/" + timeout_login);
}

for (int second = 0;; second++) {
if (second >= timeout_login) {continua=false; tearDown(); break; }
checkpoll();
try { if (isElementPresent(By.xpath("//div[@id='fbra_checkoutFasterCheckout']/div/div/div/div/form/div[2]/button"))) { break; } } catch (Exception e) {}
Thread.sleep(1000);
System.out.println("Esperando LOGIN 2/3 -- "+second+ "/" + timeout_login);
}

checkpoll();
driver.findElement(By.id("emailAddress")).sendKeys(emailAddress);
checkpoll();
```

```

driver.findElement(By.xpath("//div[@id='fbra_checkoutFasterCheckout']/div/div/div/div/f
orm/div[2]/button")).click();
    for (int second = 0;; second++) {
        if (second >= timeout_login) {continua=false; tearDown(); break; }
        checkpoll();
        //try { if ("Ir a pagarEn el siguiente paso podrás seleccionar tu medio de
pago\n\nContinuar".equals(driver.findElement(By.xpath("//div[@id='fbra_checkoutDelivery
Actions']/div/div/span/div")).getText())
        {
ts_login=Instant.now().toEpochMilli();estado_login = "OK";break;} } catch (Exception e)
{}
        try
        {
        if
(isElementPresent(By.xpath("//div[@id='fbra_checkoutDeliveryActions']/div/div/span/div/
div/div/div[2]/button"))) { ts_login=Instant.now().toEpochMilli();estado_login =
"OK";break;} } catch (Exception e) {}

        Thread.sleep(1000);
        System.out.println("Esperando LOGIN 3/3 -- "+second+ "/" + timeout_login);
    }
    pageLoad.stop();
    tiempo_login=pageLoad.getTime();
    System.out.println("LOGIN-"+estado_login" / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_login));
    if (estado_login.equals("ERROR")) {continua=false;} else {tiemponderado = 80 ;}
    }
    // FIN [3.-LOGIN]]

    // INICIO [4.-OMS]]
    if (continua){
    pageLoad.reset();
    pageLoad.start();
    checkpoll();
    System.out.println("ANTES DE SELECTOR");
    // JavascriptExecutor jsx = (JavascriptExecutor)driver;
    jsx.executeScript("window.scrollTo(0,900)", "");
    Thread.sleep(1000);

    driver.findElement(By.cssSelector(".fbra_test_checkoutComponentDeliveryActions__continu
eToSecurePaymentButton")).click();

    //driver.findElement(By.xpath("//div[@id='fbra_checkoutDeliveryActions']/div/div/span/d

```

```
iv/div/div/div[2]/button")).click();
    jsx.executeScript("window.scrollTo(0,900)", "");
    for (int second = 0;; second++) {
        if (second >= timeout_oms) {continua=false; tearDown(); break; }
        checkpoll();
        try {
            if (isElementPresent(By.cssSelector("body >
div.fbra_container.fbra_mainContainer > div.fbra_mainContent >
div.fbra_paymentOptionsTitle > h2"))) {
                ts_oms=Instant.now().toEpochMilli();estado_oms
= "OK";break; }
            } catch (Exception e) {}
        Thread.sleep(1000);
        System.out.println("Esperando OMS-- "+second+ "/" + timeout_oms);
    }
    pageLoad.stop();
    tiempo_oms=pageLoad.getTime();
    System.out.println("OMS-"+estado_oms+ " / Tiempo:"
+TimeUnit.MILLISECONDS.toSeconds(tiempo_oms));
    if (estado_oms.equals("ERROR")) {continua=false;} else {tiemponderado = 100 ;}
}
// FIN [4.-OMS]

// INICIO [5.-CLEAN]
if (estado_login.equals("OK")){
driver.get("http://www.falabella.com/falabella-cl/cart/cart.jsp");
checkpoll();
driver.get("http://www.falabella.com/falabella-cl/cart/cart.jsp");
checkpoll();
for (int second = 0;; second++) {
    if (second >= timeout_clean) {continua=false; tearDown(); break; }
    checkpoll();
    try {
        if (isElementPresent(By.cssSelector("a.fb-product__action-list__item:nth-
child(1)"))) break;
    } catch (Exception e) {}
    Thread.sleep(1000);
    System.out.println("Esperando CLEAN 1/4-- "+second+ "/" + timeout_clean);
}
driver.findElement(By.cssSelector("a.fb-product__action-list__item:nth-
child(1)")).click();
for (int second = 0;; second++) {
    if (second >= timeout_clean) {continua=false; tearDown(); break; }
    checkpoll();
    try {
        if (isElementPresent(By.cssSelector(".fb-order-status__empty-basket"))) {
```

```
break; } } catch (Exception e) {}
    Thread.sleep(1000);
    System.out.println("Esperando CLEAN 2/4 -- "+second+ "/" + timeout_clean);
}
driver.get("https://secure.falabella.com/falabella-cl/myaccount/homeZP.jsp");

for (int second = 0;; second++) {
    if (second >= timeout_clean) {continua=false; tearDown(); break; }
    checkpoll();
    try { if (isElementPresent(By.xpath("//div[@id='freshLogin']/a[2]/span"))) {
break; }} catch (Exception e) {}
    Thread.sleep(1000);
    System.out.println("Esperando CLEAN 3/4-- "+second+ "/" + timeout_clean);
}
checkpoll();
driver.findElement(By.xpath("//div[@id='freshLogin']/a[2]/span")).click();
}
// FIN [5.-CLEAN]]

if (!logready) {genera_log();}
}

@AfterMethod

@AfterClass(alwaysRun = true)
public void tearDown() throws Exception {
    //driver.close();
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        Assert.fail(verificationErrorString);
    }
    if (!logready) {genera_log();}
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
```

```
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}

private static void genera_log(){
    System.out.println("Generando LOG");
    System.out.println("Estado HOME:" + estado_home + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_home) + "TimeStamp:" + ts_home);
    System.out.println("Estado CAT:" + estado_cat + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_cat) + "TimeStamp:" + ts_cat);
    System.out.println("Estado PLP:" + estado_plp + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_plp) + "TimeStamp:" + ts_plp);
    System.out.println("Estado PDP:" + estado_pdp + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_pdp) + "TimeStamp:" + ts_pdp);
    System.out.println("Estado CART:" + estado_cart + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_cart) + "TimeStamp:" + ts_cart);
}
```

```
System.out.println("Estado LOGIN:"+ estado_login + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_login) + "TimeStamp:" + ts_login);
System.out.println("Estado OMS:"+ estado_oms + " / Tiempo de Carga:" +
TimeUnit.MILLISECONDS.toSeconds(tiempo_oms) + "TimeStamp:" + ts_oms);
if (encuesta){
    System.out.println("Estado Encuesta: Aparecio y fue denegada!");
} else {
    System.out.println("Estado Encuesta: No Aparecio");
}
Long tiempototal = tiempo_home + tiempo_cat + tiempo_plp + tiempo_pdp +
tiempo_cart + tiempo_login + tiempo_oms ;
System.out.println("Tiempo Total de Carga del Sitio: " +
TimeUnit.MILLISECONDS.toSeconds(tiempototal));
if (ts_home == 0L) {ts_home=Instant.now().toEpochMilli();}
if (ts_cat == 0L) {ts_cat=Instant.now().toEpochMilli();}
if (ts_plp == 0L) {ts_plp=Instant.now().toEpochMilli();}
if (ts_pdp == 0L) {ts_pdp=Instant.now().toEpochMilli();}
if (ts_cart == 0L) {ts_cart=Instant.now().toEpochMilli();}
if (ts_login == 0L) {ts_login=Instant.now().toEpochMilli();}
if (ts_oms == 0L) {ts_oms=Instant.now().toEpochMilli();}
data = inicia_sec + ";" + inicia_sec + ";" + siteRef + ";" + correo + ";" +
"1.1" + ";" + estado_home + ";" + tiempo_home + ";" + ts_home + ";" ;
data = data + "1.2" + ";" + estado_cat + ";" + tiempo_cat + ";" + ts_cat +
";" ;
data = data + "1.3" + ";" + estado_plp + ";" + tiempo_plp + ";" + ts_plp +
";" ;
data = data + "1.4" + ";" + estado_pdp + ";" + tiempo_pdp + ";" + ts_pdp +
";" ;
data = data + "2" + ";" + estado_cart + ";" + tiempo_cart + ";" + ts_cart +
";" ;
data = data + "3" + ";" + estado_login + ";" + tiempo_login + ";" + ts_login
+ ";" ;
data = data + "4" + ";" + estado_oms + ";" + tiempo_oms + ";" + ts_oms +
"\n" ;

MyLog registro = new MyLog(siteRef, logFileCSV, logFileNative, inicia_sec,
correo, data);
registro.finalize();
if (!continua) {
    try {
```



```
        shottake("ERROR");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
};
logready = true;
}
public void checkpoll () {
try {
        if (isElementPresent(By.linkText("No, gracias."))) {
            shottake("POLL");
            driver.findElement(By.linkText("No, gracias.")).click();
            encuesta = true;
        }
    } catch (Exception e) {}
}
public static void shottake(String pTipo) throws IOException {

        File scrFile = ((TakesScreenshot)driver).getScreenshotAs (OutputType.FILE);
        FileUtils.copyFile(scrFile, new
File(screenshotsLocation+inicia_sec.toString().trim()+"-"+pTipo+".jpg"));
        System.out.println("Grabando Screenshot de : " + pTipo);
    }

private static void loadRuntimeParams() {
try {
        File file = new File("bs2webmonitor.properties");
        FileInputStream fileInput = new FileInputStream(file);
        Properties properties = new Properties();
        properties.load(fileInput);
        fileInput.close();

        siteRef = properties.getProperty("siteRef");
        logFileCSV = properties.getProperty("logFileCSV");
        logFileNative = properties.getProperty("logFileNative");
        screenshotsLocation = properties.getProperty("screenshotsLocation");
        webDriverLocation = properties.getProperty("webDriverLocation");
        browser = properties.getProperty("browser");
        total = properties.getProperty("total");
    }
}
```

```
timeout_home      = Integer.parseInt(properties.getProperty("timeout_home"));
timeout_cat       = Integer.parseInt(properties.getProperty("timeout_cat"));
timeout_plp       = Integer.parseInt(properties.getProperty("timeout_plp"));
timeout_pdp       = Integer.parseInt(properties.getProperty("timeout_pdp"));
timeout_cart      = Integer.parseInt(properties.getProperty("timeout_cart"));
timeout_login     = Integer.parseInt(properties.getProperty("timeout_login"));
timeout_oms       = Integer.parseInt(properties.getProperty("timeout_oms"));
timeout_clean     = Integer.parseInt(properties.getProperty("timeout_clean"));
selenium_hub      = properties.getProperty("selenium-hub");

// Presenta los parametros de ejecucion del ROBOT
System.out.println("Parametros de ejecucion:");
Enumeration enuKeys = properties.keys();
while (enuKeys.hasMoreElements()) {
    String key = (String) enuKeys.nextElement();
    String value = properties.getProperty(key);
    System.out.println(key + ": " + value);
}

} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

} //fin loadrunners

@AfterMethod
public void takeScreenshotOnFailure(ITestResult testResult) throws IOException {
    if (testResult.getStatus() == ITestResult.FAILURE) {
        //Calendar calendar = Calendar.getInstance();
        //SimpleDateFormat formater = new SimpleDateFormat("dd-MM-yyyy_hh-mm-ss");

        //File scrFile =
        ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        //FileUtils.copyFile(scrFile, new File(screenshotsLocation+inicia_sec+"-ERROR.jpg"));
    }
}
```

```
        shottake("ERROR");
    }
}
}
```

Código de programación para la configuración del log que debe generar el monitoreo sintético, el cual será leído por la herramienta Splunk Forwarder.

```
package cl.bs2.webmonitor;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.time.Instant;

public class MyLog {

    private long timeStamp;
    private long elapsed = 0;
    private String label = "http://www.falabella.com/falabella-cl/";
    private long responseCode =200;
    private String responseMessage = "OK";
    private String threadName = "ThreadGroup";
    private String dataType = "text";
    private String success = "true";
    private String failureMessage = "";
    private long bytes = 0;
    private int grpThreads = 1;
    private int allThreads = 1;
    private long latency = 0;
    private long IdleTime = 0;
    private String estadoPagina;
    private String etapa;
    private long tiempoTotal = 0;
    private String site;
    private long secuencia;
```

```
private Writer logFileCSV;
private Writer logFileNative;

public MyLog(String pSite, String pFileCSV, String pFileNative, Long psecuencia,
String pCorreo, String Data) {

    secuencia = psecuencia;
    site = pSite;
    failureMessage = pCorreo;

    // Abre el archivo en formato nativo en modo de agregación
    try {
        File logFile = new File(pFileNative);
        FileOutputStream is = new FileOutputStream(logFile, true);
        OutputStreamWriter osw = new OutputStreamWriter(is);
        logFileNative = new BufferedWriter(osw);
        logFileNative.write(Data);

    } catch (Exception e) {
        System.err.println("Problemas con el archivo "+pFileNative);
    }

    // Abre el archivo en formato CSV en modo agregación
    try {
        File logFile = new File(pFileCSV);
        FileOutputStream is = new FileOutputStream(logFile, true);
        OutputStreamWriter osw = new OutputStreamWriter(is);
        logFileCSV = new BufferedWriter(osw);

        // Graba la cabecera del archivo

//logFileCSV.write("timeStamp;elapsed;label;responseCode;responseMessage;threadName;dat
aType;success;failureMessage;bytes;grpThreads;allThreads;Latency;IdleTime;EstadoPagina;
Etapa;TiempoTotal;SITE;Secuencia\n");

    } catch (Exception e) {
        System.err.println("Problemas con el archivo "+pFileCSV);
    }
}
```

```
public boolean flushInfoToLog() {

    // Genera el registro en formato CSV
    logCSVFormat();

    // Genera el registro en formato Nativo de Splunk
    logNaviteFormat();

    return(true);
}

public boolean flushInfoToLog(String pEtapa, String pEstadoPagina, long
pTiempoTotal, long ptimestamp, String pCorreo) {
    etapa = pEtapa;
    estadoPagina = pEstadoPagina;
    tiempoTotal = pTiempoTotal;
    timeStamp = ptimestamp;
    failureMessage = pCorreo;

    return this.flushInfoToLog();
}

private void logCSVFormat () {
    try {
        // Formato nativo para Splunk-----
        String logLine = this.timeStamp + ";" +
            this.elapsed + ";" +
            this.label + ";" +
            this.responseCode + ";" +
            this.responseMessage + ";" +
            this.threadName + ";" +
            this.dataType + ";" +
            this.success + ";" +
            this.failureMessage + ";" +
            this.bytes + ";" +
            this.grpThreads + ";" +
            this.allThreads + ";" +
            this.latency + ";" +
            this.IdleTime + ";" +
            this.estadoPagina + ";" +
```

```
        this.etapa + ";" +
        this.tiempoTotal + ";" +
        this.site + ";" +
        this.secuencia +
        "\n";

// Graba el evento
logFileCSV.write(logLine);

// Depuracion para validar el resultado
//System.out.println(logLine);

} catch (IOException e) {
    System.err.println("Problemas al grabar el evento en formato CSV.");
}

}

private void logNaviteFormat() {

    try {
        // Formato nativo para Splunk
        String logLine = "_time=" + this.timeStamp +
            " secuencia=" + this.secuencia +
            " etapa=" + this.etapa +
            " estadoPagina=" + this.estadoPagina +
            " msec=" + this.tiempoTotal +
            " site=" + this.site +
            "\n";

        // Graba el evento
        logFileNative.write(logLine);

        // Depuracion para validar el resultado
        //System.out.println(logLine);

    } catch (IOException e) {
        System.err.println("Problemas al grabar el evento en formato nativo.");
    }

}
```

```
@Override
public void finalize() {
    try {
        // Cierra el archivo de log en formato Splunk
        logFileNative.close();
        System.out.println("Cierra el archivo de log en formato nativo.");

        // Cierra el archivo de log en formato CSV
        logFileCSV.close();
        System.out.println("Cierra el archivo de log en formato CSV.");

    } catch (IOException e) {
        System.err.println("Problemas con el registro de eventos.");
    }
}
}
```

Representación gráfica del proceso del monitoreo sintético a partir del flujo de navegación de experiencia usuario:

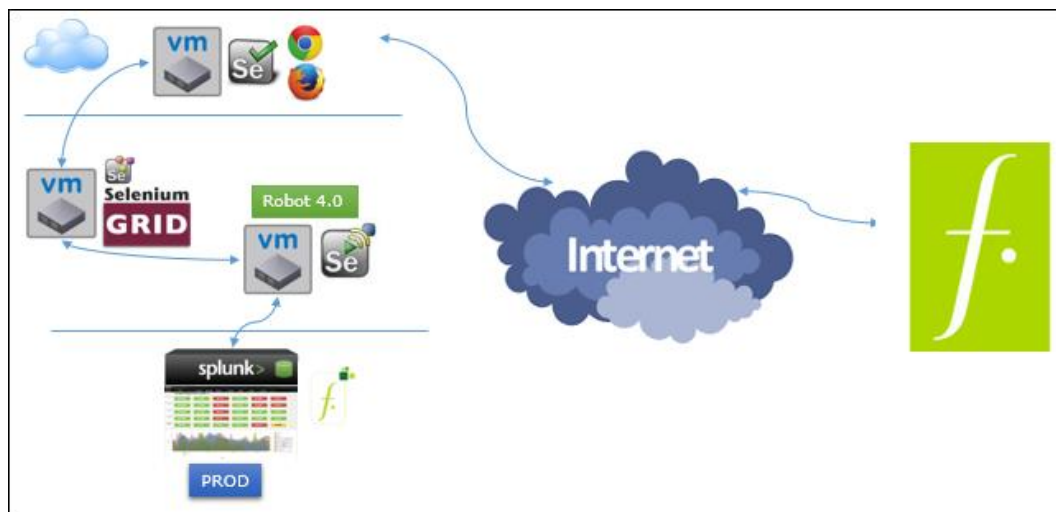


Figura 21: Flujo de monitoreo sintético

A continuación se presenta una muestra del archivo log que crea el monitoreo sintético, para el ciclo de navegación de todas las etapas que componen el flujo.

2017110141103	http://www.falabella.cl	monsint@gmail.com	OK	1.1	3975	1495963621
2017110141113	http://www.falabella.cl	monsint@gmail.com	OK	1.2	4980	1495963621
2017110141155	http://www.falabella.cl	monsint@gmail.com	OK	1.3	4072	1495963621
2017110141223	http://www.falabella.cl	monsint@gmail.com	OK	1.4	5625	1495963621
2017110141244	http://www.falabella.cl	monsint@gmail.com	OK	2	6525	1495963621
2017110141259	http://www.falabella.cl	monsint@gmail.com	OK	3	38019	1495963621
2017110141313	http://www.falabella.cl	monsint@gmail.com	OK	4	12922	1495963621

Figura 22: Log monitoreo sintético

Detalle de campos generados en el archivo log:

- Primer campo : Registra la fecha en formar YYYYMMDDHHMMSS
- Segundo campo : Registra la url que se monitorea
- Tercer campo : Registra la cuenta con la que se realiza el monitoreo sintético
- Cuarto campo : Registra el estado de la navegación en la etapa del flujo.
- Quinto campo : Registra la etapa correspondiente al flujo de navegación.
- Sexto campo : Registra la secuencia asociada al flujo completo de navegación

3.10 Ejecución del monitoreo sintético

Un vez que tenemos programados las clases y los métodos *Java* con las librerías *Selenium*, solo nos queda tener que calendarizar la periodicidad con la que se debe realizar el monitoreo sintético, para ello se ha definido que la tarea de ejecución controlada sea mediante la herramienta de Sistema Operativo *crontab*.

Para su ejecución, se ha creado una *Shell* la cual es la encargada de realizar la llamada para iniciar la ejecución de una *Java Virtual Machine (JVM)* que es el proceso nativo de *Java*, que se ejecuta a nivel de sistema operativo, para interpretar y ejecutar las instrucciones expresadas en un código binario especial (*Java bytecode*), el que a su vez es generado por el compilador del lenguaje *Java* para emular el monitoreo sintético.

De acuerdo a la información recopilada en el proceso de requerimientos se definió que cada ciclo de navegación del monitoreo sintético se debe ejecutar cada un minuto, con la finalidad de tener información real del comportamiento de la página web durante las 24 horas, poniendo especial énfasis en los periodos de promociones web, ya que para el negocio es fundamental conocer el comportamiento de la infraestructura web durante los periodos de alta carga en la plataforma web.

A continuación el código en *ShellScript* que controla la ejecución del proceso de navegación:

```
#!/bin/sh
export DBUS_SESSION_BUS_ADDRESS=/dev/null
START=$(date +%s.%N)

cd /falabella/script/bin/
#ps -aux | grep defunc >> /falabella/script/log/procesos.log 2>&1
#echo "Procesos chromedriver a eliminar: " >>
/falabella/script/log/procesos.log 2>&1
killall --quiet --older-than 10m chromedriver
killall --quiet --older-than 10m chrome
#/usr/sbin/pidof chromedriver | awk '{print substr($0, index($0, $5))}' >>
/falabella/script/log/procesos.log 2>&1
#/usr/sbin/pidof chromedriver | awk '{print substr($0, index($0, $5))}' |
xargs kill
#echo "Procesos chrome a eliminar: " >> /falabella/script/log/procesos.log
2>&1
#/usr/sbin/pidof chrome | awk '{print substr($0, index($0, $5))}' >>
/falabella/script/log/procesos.log 2>&1
#/usr/sbin/pidof chrome | awk '{print substr($0, index($0, $5))}' | xargs kill

USER1=$(ps -fea | grep USER1 | grep java | wc -l)
USER2=$(ps -fea | grep USER2 | grep java | wc -l)
if [ "$USER1" == "1" ]
```

```
then
  echo "El USER1 esta en ejecucion...."
  if [ "$USER2" == "1" ]
    then
      EJECUTA=NADA
      echo "Ambos Usuarios en ejecucion, nada que hacer! $(date)" >>
/falabella/script/log/bs2webmonitor.log
      exit
    else
      ## Ejecuta el USER2
      EJECUTA=USER2
      echo "Ejecutando USER2 $(date)" >>
/falabella/script/log/bs2webmonitor.log
      java -cp
"/falabella/script/lib/testng.jar:/falabella/script/bin/falac1.jar:/falabella/
script/lib/jcommander.jar:/falabella/script/lib/client-combined-3.3.1-
nodeps.jar:/falabella/script/lib/selenium-server-standalone-3.3.1.jar"
org.testng.TestNG USER2.xml >> /falabella/script/log/bs2webmonitor-USER2.log
2>&1
      fi
fi
if [ "$USER2" == "1" ]
then
  echo "El USER2 esta en ejecucion...."
  if [ "$USER1" == "1" ]
    then
      EJECUTA=NADA
      echo "Ambos Usuarios en ejecucion, nada que hacer! $(date)" >>
/falabella/script/log/bs2webmonitor.log
      exit
    else
      ## Ejecuta el USER1
      EJECUTA=USER1
      echo "Ejecutando USER1 -- $(date)" >>
/falabella/script/log/bs2webmonitor.log
      java -cp
```

```
"/falabella/script/lib/testng.jar:/falabella/script/bin/falac1.jar:/falabella/
script/lib/jcommander.jar:/falabella/script/lib/client-combined-3.3.1-
nodeps.jar:/falabella/script/lib/selenium-server-standalone-3.3.1.jar"
org.testng.TestNG USER1.xml >> /falabella/script/log/bs2webmonitor-USER1.log
2>&1
    fi
fi
if [ "$USER1" != "1" ] && [ "$USER2" != "1" ]
then
    # Ejecuta User1
    EJECUTA=USER2
    echo "Ejecutando USER1 $(date)" >> /falabella/script/log/bs2webmonitor.log
java -cp
"/falabella/script/lib/testng.jar:/falabella/script/bin/falac1.jar:/falabella/
script/lib/jcommander.jar:/falabella/script/lib/client-combined-3.3.1-
nodeps.jar:/falabella/script/lib/selenium-server-standalone-3.3.1.jar"
org.testng.TestNG USER1.xml >> /falabella/script/log/bs2webmonitor-USER1.log
2>&1
fi
END=$(date +%s.%N)
TIEMPO=$(echo "($END - $START)/1" | bc)
echo "Proceso de Usuario $EJECUTA Realizado en $TIEMPO - Segs " >>
/falabella/script/log/tiempos.log
```

3.11 Splunk Forwarder

Como se definió anteriormente dentro de *Splunk Enterprise*, existe el componente de nombre *forwarder*, que es el encargado de proporcionar la información entre la fuente de datos y la base de datos de *Splunk*.

Para desarrollar el trabajo se debe instalar este componente en el servidor donde escribe el *log* del monitoreo sintético (de nombre monitoreo.csv), en este caso en la maquina *LnxSelenium*.

A continuación se detalla paso a paso las instrucciones que se debe realizar a nivel de Sistema Operativo para instalar y configurar el envío automático de información mediante el componente *Forwarder*:

- Descargar desde la página de *Splunk* el instalador para Linux del producto *Splunk Forwarder* (splunkforwarder-5.0.1-143156-linux-2.6-x86_64.rpm)

- Instalar a nivel de sistema operativo los binarios del agente, mediante la instrucción :

```
rpm -i splunkforwarder-5.0.1-143156-linux-2.6-x86_64.rpm
```

- En caso de reinicio del servidor se debe habilitar el inicio automático del agente mediante la instrucción:

```
/opt/splunkforwarder/bin/splunk enable boot-start
```

- Configurar el agente con los datos del servidor que recibirá la información, en este caso *LnxDsplunk* con el puerto 9997, mediante la instrucción:

```
/opt/splunkforwarder/bin/splunk add forward-server  
LnxDsplunk:9997
```

- Sincronizar el envío automático de datos a partir del log que genera el monitoreo sintético con el índice (de nombre *monitoreo*) de la base de datos, la instrucción a ejecutar es:

```
/opt/splunkforwarder/bin/splunk add monitor  
/path/to/app/logs/monitoreo.csv -index monitoreo
```

Teniendo instalado y configurado el agente, cada vez que se ejecute el monitoreo sintético y se genere una nueva línea dentro del archivo log, esta será enviada automáticamente a la base de datos, específicamente en el índice de *Splunk Enterprise* de nombre *monitoreo*.

3.12 Configuración Splunk Enterprise

Una vez instalado y configurado el componente *forwarder* en nuestro servidor fuente de datos, se debe crear el índice dentro de la base de datos de *Splunk*, como se explica anteriormente es el índice el componente de la base de datos que contiene el *log* del monitoreo sintético, la creación del índice se realiza desde la herramienta de administración de *Splunk*, solo se debe proporcionar nombre, mismo nombre definido en la configuración del agente *forwarder* y tamaño del objeto:

The screenshot shows the 'New Index' configuration window in Splunk. The window has a title bar with 'New Index' and a close button. Below the title bar is the 'General Settings' section. The settings are as follows:

- Index Name:** A text input field containing 'monitor'. Below it is a small text description: 'Set index name (e.g., INDEX_NAME). Search using index=INDEX_NAME.'
- Home Path:** A text input field containing 'optional'. Below it is a small text description: 'Hot/warm db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/db).'
- Cold Path:** A text input field containing 'optional'. Below it is a small text description: 'Cold db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/colddb).'
- Thawed Path:** A text input field containing 'optional'. Below it is a small text description: 'Thawed/resurrected db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/thaweddb).'
- Data Integrity Check:** A toggle switch with 'Enable' selected and 'Disable' unselected. Below it is a small text description: 'Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.'
- Max Size of Entire Index:** A text input field containing '500' and a dropdown menu set to 'GB'. Below it is a small text description: 'Maximum target size of entire index.'
- Max Size of Hot/Warm/Cold Bucket:** A text input field containing 'auto' and a dropdown menu set to 'GB'. Below it is a small text description: 'Maximum target size of buckets. Enter 'auto_high_volume' for high-volume indexes.'

At the bottom of the window, there are two buttons: 'Cancel' on the left and 'Save' on the right.

Figura 23: Creación de índice en Splunk

El siguiente paso es crear el tablero de control en donde se desplegara la información procesada y correlacionada en forma gráfica, este tablero se diseña desde el diseñador de *dashboard* que otorga la consola de administración de *Splunk*, en él se deben definir los objetos que corresponderán a cada uno de los indicadores que se quiere mostrar.

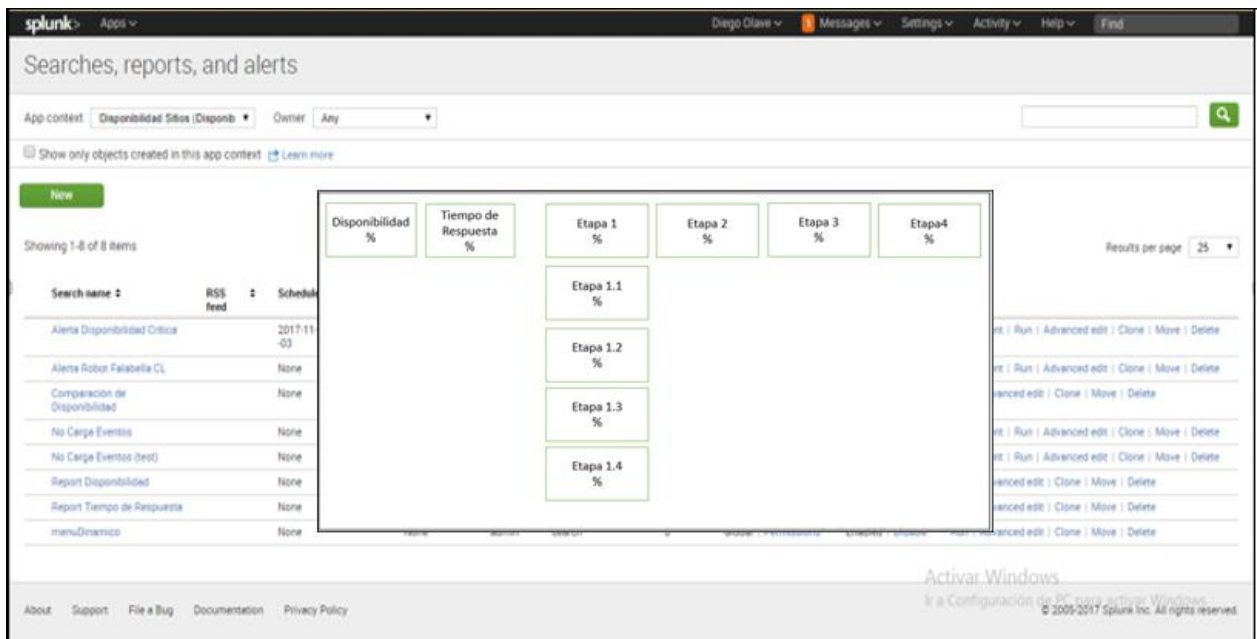


Figura 24: Creación de panel de control en Splunk

Ya que la información que será desplegada en los objetos del tablero será del tipo numérico y expresada en forma de porcentual, se pueden establecer umbrales para pintar el objeto a partir de la definición de disponibilidad, esto se define desde la propiedad el objeto y se correlaciona con el resultado de la query.

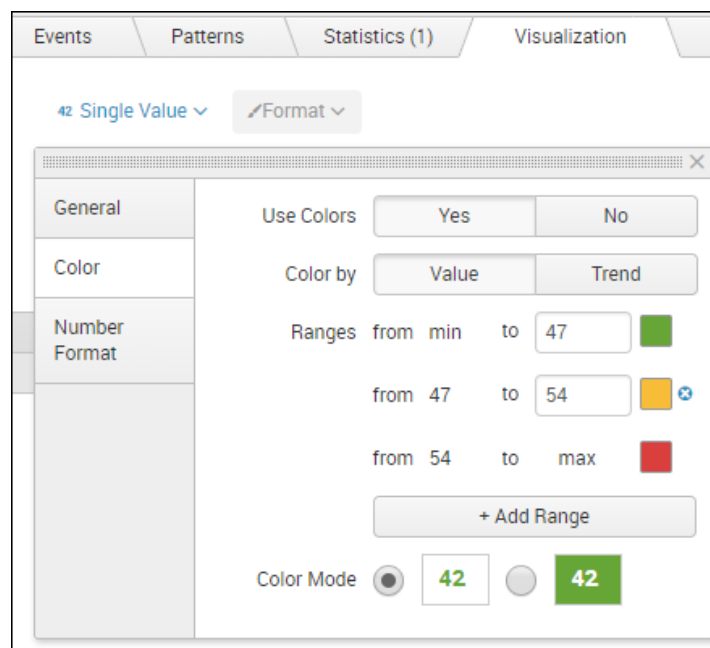


Figura 25: Definición de umbrales en Splunk

Posterior a la creación del índice, se debe crear el objeto datamodel, el cual corresponde al entorno definido dentro de la base de datos para almacenar la información que será requerida en el tablero de control, en a partir de un proceso interno de *Splunk*

Una vez que ya se encuentra configurado el ambiente se debe programar los objetos correspondientes a para presentar la información a partir de las consultas a la base de datos, esta programación se escribe mediante el lenguaje nativo de *Splunk* conocido como *SPL*.

A continuación se muestra el código con el que se construyen los indicadores para cada una de las etapas de navegación, para el tiempo de respuesta y para obtener la disponibilidad de la página web.

Disponibilidad global de la página:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion SPLITROW eEtapa AS eEtapa SORT 100000 date_month ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0 | eval Disponibilidad=round(Exitosos/Total*Ponderacion,6)*100 | stats sum(Disponibilidad) as Disponibilidad | eval Disponibilidad=round(Disponibilidad,2) | append [| makeresults | eval Disponibilidad=0] | fields Disponibilidad range | head 1
```

Tiempo de respuesta global de la página:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad sum(TiempoTotal) AS AvgTime count(is_Exitosos) AS Exitosos SPLITROW Secuencia AS Secuencia SPLITROW _time AS _time PERIOD hour SORT 0 Secuencia ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 1 | where Exitosos=7 | stats avg(AvgTime) as TResp | append [| makeresults | eval TResp=0] | rangemap field=TResp low=0.1-47 elevated=47.01-53.999 severe=54-9999 default=severe | fields TResp range | head 1
```

Para obtener el porcentaje del tiempo de disponibilidad de cada una de las etapas de navegación, las consultas son las siguientes:

Disponibilidad Primera etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa in (1.1,1.2,1.3,1.4) SORT 1000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0
SHOWOTHER 0| eval disp= round(Exitosos/Total*Ponderacion/0.4,4)|stats sum(disp) as
Disp,dc(eEtapa) as conta by _time| where conta=4 | eval Dispond=Disp*100.00 | stats
avg(Dispond) as Disp | append [| makeresults | eval Disp=0] | rangemap field=Disp
severe=0-99.599 elevated=99.6-99.799 low=99.8-100 default=0| fields Disp range | head
1
```

Disponibilidad Primera Sub etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa is "1.1" SORT 10000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval
disp= round(Exitosos/Total*100,2)| stats avg(disp) as Disp | append [| makeresults |
eval Disp=0] | rangemap field=Disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields Disp range | head 1
```

Disponibilidad Segunda Sub etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa is "1.2" SORT 10000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval
disp= round(Exitosos/Total*100,4)| stats avg(disp) as Disp | append [| makeresults |
eval Disp=0] | rangemap field=Disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields Disp range | head 1
```


Disponibilidad Tercera Sub etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa is "1.3" SORT 10000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval
disp= round(Exitosos/Total*100,4)| stats avg(disp) as Disp | append [| makeresults |
eval Disp=0] | rangemap field=Disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields Disp range | head 1
```

Disponibilidad Cuarta Sub etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa is "1.3" SORT 10000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval
disp= round(Exitosos/Total*100,4)| stats avg(disp) as Disp | append [| makeresults |
eval Disp=0] | rangemap field=Disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields Disp range | head 1
```

Disponibilidad Segunda etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa is "2" SORT 1000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval
disp= round(Exitosos/Total*100,4)| stats avg(disp) as disp | append [| makeresults |
eval disp=0] | rangemap field=disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields disp range | head 1
```

Disponibilidad Tercera etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion SPLITROW _time AS
_time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER Etapa is "3" SORT 1000 _time
ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval disp=
round(Exitosos/Total*100,4)| stats avg(disp) as Disp | append [| makeresults | eval
Disp=0] | rangemap field=Disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields Disp range | head 1
```

Disponibilidad Cuarta etapa de navegación:

```
| pivot DisponibilidadModel3 DetalleDisponibilidad count(DetalleDisponibilidad) AS
Total count(is_Exitosos) AS Exitosos latest(xPondera) AS Ponderacion avg(TiempoTotal)
AS TiempoTotal SPLITROW _time AS _time PERIOD minute SPLITROW eEtapa AS eEtapa FILTER
Etapa is "4" SORT 10000 _time ROWSUMMARY 0 COLSUMMARY 0 NUMCOLS 0 SHOWOTHER 0| eval
disp= round(Exitosos/Total*100,4)| stats avg(disp) as Disp | append [| makeresults |
eval Disp=0] | rangemap field=Disp severe=0-99.599 elevated=99.6-99.799 low=99.8-100
default=0| fields Disp range | head 1
```

Teniendo programados cada uno de los objetos que componen el tablero, la información será desplegada de forma automática ya que la compilación y parseo de la consulta se realiza en tiempo de ejecución.

A continuación se muestra el resultado de la ejecución de las consultas a la base de datos mediante lengua SPL antes mencionadas en los objetos dentro del tablero de control con la información obtenida a partir del proceso de navegación de experiencia usuario.



Figura 26: Tablero de control Final

4 HALLAZGOS

4.1 Alto consumo de procesamiento

Dentro del análisis de comportamiento de las máquinas que desarrollan el monitoreo sintético, se aprecia desde el monitoreo tradicional mediante agentes de Sistema Operático, un alto consumo de CPU y Memoria, el que llega prácticamente al 100% de utilización, lo que avizora un problema, ya que produce encolamiento en la tarea de ejecución de monitoreo sintético.

4.2 Problemáticas en la operación actual

Las principales problemáticas en el actual diseño de monitoreo están dadas principalmente por la gran cantidad de recursos que se monitorean en los servidores que componen la infraestructura web, los que no son repetitivos para medir procesos de negocio

4.3 Dependencia del monitoreo sintético

Dado que la solución de monitoreo sintético se realiza solo en una máquina (no en un *cluster* de máquinas), existe la posibilidad por diversas causas que pierda disponibilidad (humanas, hardware, software, etc.), con esto no será posible la ejecución del proceso de negocio en el monitoreo sintético y con ello no poder obtener datos en tiempo real.

5 CONCLUSIONES

A partir de la investigación desarrollada, se puede obtener las siguientes conclusiones.

5.1 Realización de la implementación

Es posible realizar la implementación de monitoreo sintético mediante librerías *Selenium* y *Splunk Enterprise*. Ya que en el *log* del monitoreo sintético se crea un registro del tipo *TymeStamp*, tipo de dato necesario en *Splunk* para procesar información.

5.2 Determinar disponibilidad y tiempo de respuesta

A partir de la definición de ponderadores para las fases del proceso de navegación de un usuario, se puede aplicar un cálculo matemático para determinar la disponibilidad global y tiempo de respuesta de la página web.

5.3 Navegación no representativa para usuario final

El tiempo de respuesta total de la página web, no es representativo a la realidad de los usuarios finales, ya que la navegación no mide tiempos de respuesta asociados a los distintos proveedores de internet.

5.4 Es posible desplegar información de forma simple

Es posible graficar de una forma simple la información en una medida porcentual, que permita a los encargados una rápida lectura de la información, para determinar si requiere una revisión, descartar o/confirma un problema según sea el caso.

5.5 Es posible desplegar información en tiempo real

Es posible graficar la información en tiempo real, gracias al agente *splunk forwarder* que envía información en cuanto se genera en la fuente de datos y la interpretación de esta información mediante la consulta a la base de datos programada en cada uno de los objetos del panel.

5.6 Complemento al monitoreo actual

Si bien esta solución no está pensada en reemplazar los monitoreos que se tienen configurados, es un buen complemento para las áreas de operaciones ya que permite visibilizar si existen problemas en alguna de las etapas que componen el flujo del proceso de venta en línea.

6 GLOSARIO

C

Cpu

CPU Sigla de la expresión inglesa central processing unit, 'unidad central de proceso', que es la parte de una computadora en la que se encuentran los elementos que sirven para procesar datos. "los elementos que conforman la CPU de un ordenador son la memoria principal, la unidad aritmética y los registros de control"

D

Dashboard

Un dashboard es una representación gráfica de las principales métricas o indicadores que intervienen en la consecución de los objetivos de una estrategia.

E

E-Commers

Es la definición en inglés para comercio electrónico el cual consiste en un método de compraventa de bienes, productos o servicios mediante la plataforma de internet como medio, es decir, comercializar un producto en línea.

H

Hardware

El hardware es un término inglés que permite definir no sólo a los componentes físicos internos (disco duro, placa madre, microprocesador, circuitos, cables, etc.), sino también a los periféricos (escáner, monitores, impresoras), que trabaja o interactúa de algún modo con el computador.

I

I/O

El término I/O (acrónimo en inglés para input / output) se utiliza para describir cualquier programa o dispositivo que transfiere datos desde un ordenador y hacia un dispositivo periférico o viceversa.

J

Java

Lenguaje de programación orientado a objetos, fue diseñado específicamente para poder ser ejecutado en diferentes plataformas, usando un motor específico elaborado para tal efecto.

Su principal potencia es la portabilidad, lo que permite escribir el código una vez y ejecutar donde sea.

R

Retail

Es un término en inglés que se utiliza para nombrar a la venta minorista. La comercialización de productos al por menor, por lo tanto, constituye el **retail**. Es el concepto de asociar la venta de grandes cantidades de distintos productos, pero a muchos compradores diferentes.

Router

Un router es un dispositivo de hardware que permite la interconexión de ordenadores en red. El router o enrutador es un dispositivo que opera en capa tres de nivel de 3. Así,

permite que varias redes u ordenadores se conecten entre sí y, por ejemplo, compartan una misma conexión de Internet.

S

Script

En informática, el script es un programa simple, que habitualmente tiene sólo una tarea determinada y se usa como parte de un programa o procesamiento más complejo.

Shell

Una Shell es una secuencia de ordenes o comandos que puede interpretar el sistema operativo Unix/Linux. Su utilidad radica en lograr una secuencia de órdenes simples.

Software

Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Switch

Nombre en inglés que significa conmutador, es un dispositivo de interconexión de redes informáticas. En computación y en informática de redes, un switch es el dispositivo analógico que permite interconectar redes operando en la capa 2 o de nivel de enlace de datos del modelo OSI

Swap

Swap (en inglés, intercambio) se refiere a que en cualquier ámbito, está muy extendida su utilización para hacer referencia al trueque o intercambio de cualquier material. En informática, al espacio de intercambio de memoria a un disco.

T

TimeStamp

El *timestamp* es una secuencia de caracteres que representa una fecha y/u hora específicas. El *timestamp* es el tiempo en que un evento es guardado por un computador.

7 BIBLIOGRAFIA

<http://docs.splunk.com/Documentation/Splunk>

The complete documentation collection for Splunk (manuals, tutorials, references, etc.)

<http://www.splunk.com/view/education-videos/SP-CAAAGB6>

Educational videos produced by Splunk on a variety of topics Exploring Splunk Book

<http://www.splunk.com/goto/book>

The official Splunk book, available for free download or purchase a hardcopy

www.seleniumhq.org

Web Browser Automation

<https://www.ibm.com/developerworks/library/wa-selenium2/index.html>

End-to-end functional testing of your web applications in multiple browsers

https://www.researchgate.net/profile/Arnold_Vermeeren/publication/221518375_Understanding_scoping_and_defining_user_experience_A_survey_approach/links/00463514184e5eb983000000.pdf

Understanding, Scoping and Defining User eXperience

https://s3.amazonaws.com/academia.edu.documents/28563226/hassenzahl-ihm08.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1512508713&Signature=SvgMe9KDvzzgIMazN%2BaQcaGsnc4%3D&response-content-disposition=inline%3B%20filename%3DUser_experience_UX_towards_an_experiential.pdf

User Experience (UX): Towards an experiential perspective on product quality