UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DETECCIÓN TEMPRANA DE ANOMALÍAS EN UN PROVISIONADOR DE PLATAFORMAS DE TELECOMUNICACIONES

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN TECNOLOGÍA DE LA INFORMACIÓN

JUAN ANDRÉS CALVO RODRÍGUEZ

PROFESOR GUÍA:
NELSON BALOIAN TATARYAN

MIEMBROS DE LA COMISIÓN:
LUIS MATEU BRULE
ANDRÉS ABELIUK KIMELMAN
PATRICIO GALDAMES SEPULVEDA

SANTIAGO DE CHILE
2022

Resumen

Para vigilar la salud del sistema en una Compañía que brinda servicios de Telecomunicaciones se diseñó una Herramienta de Monitoreo, de manera de alertar a un especialista para que revise en cuanto ocurra la anomalía y realice las acciones correctivas o paliativas con anticipación, se espera que la herramienta alerte anomalías 12 horas antes que lo reporte una persona. Cuando se producen reclamos por parte de clientes finales por algún producto o servicio generalmente el problema que los provoca viene afectando la operación comercial por un tiempo considerable, lo que se traduce en una mala evaluación de la Compañía; ésto deriva en pérdida de tiempo y dinero para corregir la anomalía, arreglar información distorsionada por el problema, y por último también realizar acciones para recuperar la buena percepción de la Compañía. Para evitar esto la solución de monitoreo contempló el diseño de una aplicación web, ésta levantará alarmas cuando alguno de los indicadores pase el umbral de valores válidos, que estarán configurados, y también cuando lo informe el Algoritmo de detección temprana. El diseño fue validado por los futuros usuarios, quienes corroboran que cumple con los objetivos de facilidad de uso y además esta herramienta permite incrementar la independencia de los especialistas de la Compañía en la resolución de incidencias.

Se recolectaron datos y casos de incidencias, se seleccionaron 4 casos con anomalías, en dos de los casos los usuarios reportaron incidencias por problemas, y en los otros dos casos hubo percepción de lentitud en el sistema pero no se reportaron como incidencia, siendo el equipo de Soporte quien encontró anomalías durante la preparación de informes cuando se buscó la causa raíz del problema. En el análisis del sistema se detectó concentración de datos en la dispersión de Duración por Cantidad de transacciones, lo que permitió verificar que es posible utilizar la distribución normal. Modelando los datos de estas dos variables se observó concentración de valores no anómalos y también se advierte que los datos anómalos se sitúan en el extremo de la gráfica. Se calcularon manualmente los indicadores para validar que es posible encontrar anomalías a partir de dichos indicadores. Luego, se implementó el Algoritmo de detección temprana basado en distribución normal utilizando estas dos variables: Cantidad de transacciones y duración promedio de dichas transacciones, y se simuló el comportamiento del sistema utilizando los respectivos grupos de datos para cada uno de los cuatro casos analizados manualmente. Comparando resultados hallados manualmente y resultados generados por el algoritmo, se observa que el algoritmo es capaz de detectar la primera anomalía del día para alertar lo antes posible; en uno de los casos probados el algoritmo detectó la primera anomalía 14 horas antes que una persona reporte el problema (en el otro caso fue de 13 horas), con lo cual se valida la hipótesis, y es exitosa la detección temprana mediante el algoritmo basado en distribución normal.

Tabla de Contenido

Índice de Tablas

Capítulo 1

Introducción

1.1. Motivación

En el trabajo operativo en las empresas de telecomunicaciones el equipo de producción es el encargado de mantener la continuidad de las operación y comercialización, y diariamente se enfrenta a resolver los problemas que reportan los equipos comerciales (venta, pos-venta) y backend (cobranza, facturación, etc). El equipo de producción registra tickets de incidencias al equipo de soporte cuando los problemas reportados involucran temas técnicos de funcionamiento de algún producto del sistema comercial, uno de cuyos productos es la provisión, que consiste en la activación de algún servicio para el cliente. El reporte de la incidencia es a través de un protocolo establecido, dependiendo de la severidad de la incidencia involucra contacto telefónico (para incidencias críticas) y en todos los casos la incidencia se registra en una herramienta que gestiona el contacto y la interacción con el equipo de soporte. Este equipo apoya al de producción en la resolución de la incidencia, para lo cual se analizan los antecedentes y logs que se adjuntan en el reporte de la incidencia, pudiendo ser necesario hacer alguna corrección al software o cambiar alguna configuración.

Dentro de las incidencias del sistema comercial, las de provisión son particularmente críticas porque detienen la atención al cliente. El cliente que está en la tienda adquiriendo un nuevo equipo telefónico móvil, espera salir con el celular funcionando, pero en lugar de esto, si hay una interrupción de los procesos que habilitan los nuevos servicios de la compañía proveedora, el cliente debe quedarse esperando frente al ejecutivo de la tienda. Si la corrección de la incidencia demora entonces, el cliente deberá esperar a que en el trascurso del día se active, o la venta debe finalizarse con procedimientos manuales. En algunos sistemas comerciales simplemente no se puede concretar la venta porque el sistema tiene un error que proviene de la provisión.

Los problemas de provisión en horario de oficina son de alta presión, con lo cual poder identificarlos con 6 ó 12 horas previas a lo que hoy se hace de modo de contar con este tiempo adicional para poder resolver las incidencias que es un tiempo muy valorado por el equipo de Operaciones/ Soporte ya que ayuda a la Compañía de telecomunicaciones a lograr ventas, mantener la imagen de estabilidad en el servicio brindado (cuidar la marca), mejorar

la primera impresión del cliente recibe acerca de la compañía durante la venta, contribuyen en acentuar una buena percepción sobre la Compañía.

Las incidencias en provisión si bien son alrededor de 4 ó 5 por año, tienen tiempo altos de corrección, algunas de éstas han demorado hasta 3 días, con lo cual se afecta los ingresos y reputación del servicio de la Compañía de telecomunicaciones. Un sistema de detección temprana y monitoreo permitirá contar con más tiempo para análisis, más acciones que realizar antes del horario de oficina, anticipar la corrección del problema y por lo tanto contribuye en mejorar el servicio que ofrece Compañía.

1.2. Contexto

Amdocs, empresa en el marco de la cual se realiza esta tesis, es un proveedor de servicios y productos informáticos para compañías de telecomunicaciones (TelCo), con más de 30 años en el mercado global, 26.000 empleados y compañías clientes en los cinco continentes. Las sucursales de Brasil, México y Chile atienden el mercado de TelCo de Latinoamérica, y sus clientes son empresas grandes y medianas, tales como Telefónica, América Móvil, AT&T y Millicon.

Las TelCos cuentan con las plataformas de servicio, que pueden ser de activación y de servicio propiamente tal. Las plataformas de activación permiten al cliente (usuario) hacer llamadas, enviar mensajes y navegar por Internet. Por otro lado, las plataformas de servicios (o de valor agregado) son aquellas que ofrecen servicios ligados al plan contratado y al ciclo de vida del cliente, que puede o no ser facturado. Por ejemplo, puede suceder que un cliente, al contratar cierto plan, reciba una suscripción premium de Spotify. Este servicio está ligado al cliente mientras su plan esté activo y mantenga el plan promocional. De igual manera se puede manejar esto con otros servicios, como Netflix o TV cable.

Asimismo las TelCos cuentan con sistemas de negocio (BSS - Bussiness Support System), que utilizan los ejecutivos para comercializar los productos que ofrecen (planes, equipo, etc). Algunas compañías tienen más de un BSS, cada uno para un distinto tipo de negocio: Prepago, Pospago, Grandes Clientes o Empresa, utilizando diferentes aplicaciones y proveedores para cada uno de estos negocios; esto debido a que cada negocio requiere funcionalidades muy especificas. Por ejemplo, los grandes clientes suelen activar de manera masiva las líneas telefónicas, pueden tener varias facturas, diferentes direcciones de cobranza, etc. mientras que en algunas TelCo, por estrategia comercial, los celulares prepago se comercializan sólo a través de empresas de retail, con funcionalidades más simples. Adicionalmente, se debe considerar que, por estrategia para disminuir riesgos de operación, o por no contar con recursos financieros de inversión suficientes, es común que las compañías de telecomunicaciones inviertan en cambiar o actualizar sólo un sistema a la vez, teniendo una infraestructura variopinta.

Por último algunas compañías de telecomunicaciones ofrecen su red a otras compañías de teléfonos, para que utilicen su red, estas últimas, llamadas Operadores virtuales, tienen su propio BSS, que se conectan a la compañía, que ofrece la red, mediante la plataforma Mobile Virtual Net Enabler (MVNE). Por ejemplo, Virgin Mobile (Operador Virtual) utiliza los recursos de Movistar (Enabler); es decir, los clientes contratan planes de telefonía a Virgin

Mobile y utilizan la red de Movistar para comunicarse.

Amdocs ofrece diferentes productos para compañías de telecomunicaciones, entre los cuales se comercializa un provisionador. Esta solución es la capa de integración entre las aplicaciones de negocio (BSS, MVNE) y las plataformas de servicios de la empresa de telecomunicaciones (Figura 1.1). La función del provisionador es proveer de una capa de abstracción que orquesta los comandos técnicos y maneja la interacción con las plataformas físicas (Red). De esta manera, los sistemas de negocio tienen una capa más simple de servicios, y por lo tanto se delega en el provisionador las tareas de conexión, manejo de respuestas, envío de comandos a las diferentes plataformas, y manejo de reintentos de los comandos cuando fallan. La Figura 1.1 muestra el esquema general de orquestación que realiza el provisionador entre el negocio y la red (Central telefónica: HLR, plataforma de mensajería – SMS, plataforma de prepago, etc).

En la Figura 1.1 se puede observar las aplicaciones de negocio (BSS) que permiten a los ejecutivos de la empresa de telecomunicaciones realizar funciones de comercialización (negocio), particularmente: venta, postventa, facturación, cobranza, y atención del cliente. Estas aplicaciones conforman el sistema de soporte al negocio.

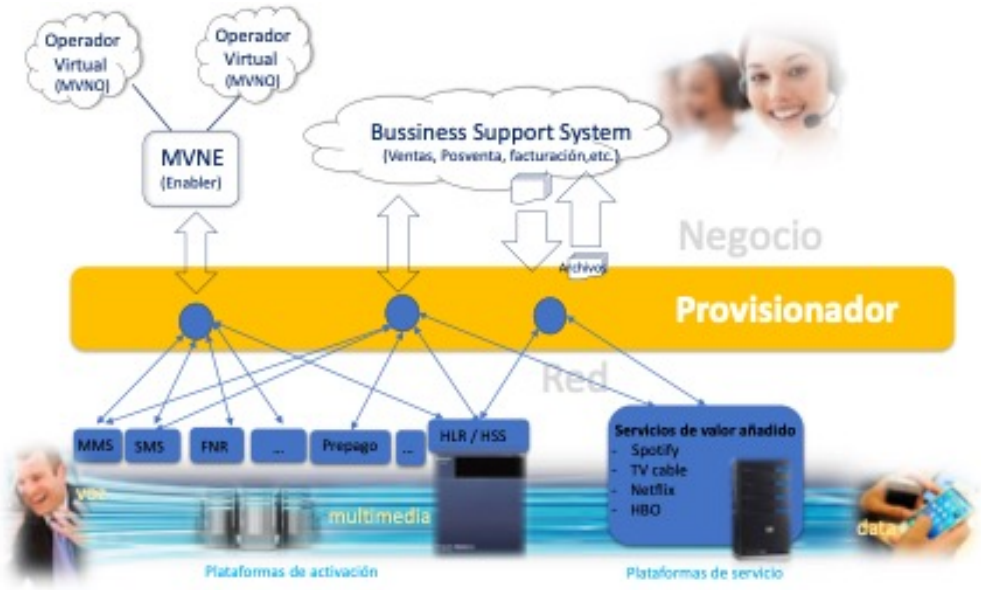


Figura 1.1: Interacción del provisionafor con aplicaciones de negocio y red

El presente trabajo de tesis abordará el desafío de detectar tempranamente anomalías en un provisionador de plataformas en el escenario antes descrito, lo cual es importante para poder corregir los problemas que ocurran y evitar el colapso del servicio de provisión, que impacta en las ventas (ya que el cliente debe irse de una tienda con el servicio habilitado y funcionando).

1.3. Arquitectura Básica del Provisionador

La Figura 1.2 muestra la arquitectura actual del provisionador de la empresa Amdocs. Allí se puede ver que el provisionador sirve para atender multi-operadores, es decir varios BSS; por ejemplo, cuando un ejecutivo de una TelCo vende un celular, se registra al cliente y al plan postpago contratado en el BSS. Esta suite envía una transacción de negocio de "alta de cliente" al Provisionador, quien tiene configurados los servicios y plataformas asociados al plan contratado en el módulo de TX Negocio (Figura 1.2). A su vez, el provisionador envía al módulo Tx Red las transacciones técnicas asociadas, de manera que este último remite los comandos de activación a cada plataforma contratada (con los valores, formatos y protocolo de comunicación que cada uno de estos entiende). El BSS genera las transacciones según las funcionalidades y casos de negocio que maneja, por ejemplo, cambia de plan (postventa) o anula a un cliente cuando el equipo de recaudación lo suspende por morosidad en sus pagos.

Por otro lado, los Operadores Virtuales utilizan su propio sistema de comercialización y se comunican con la plataforma MVNE de la compañía dueña de los recursos. La plataforma Mobile Virtual Network Enabler – MVNE es el producto donde la empresa con la red (enabler) expone servicios que puede utilizar el Operador Virtual. Ésta disponibiliza las transacciones de negocio configuradas en el provisionador.

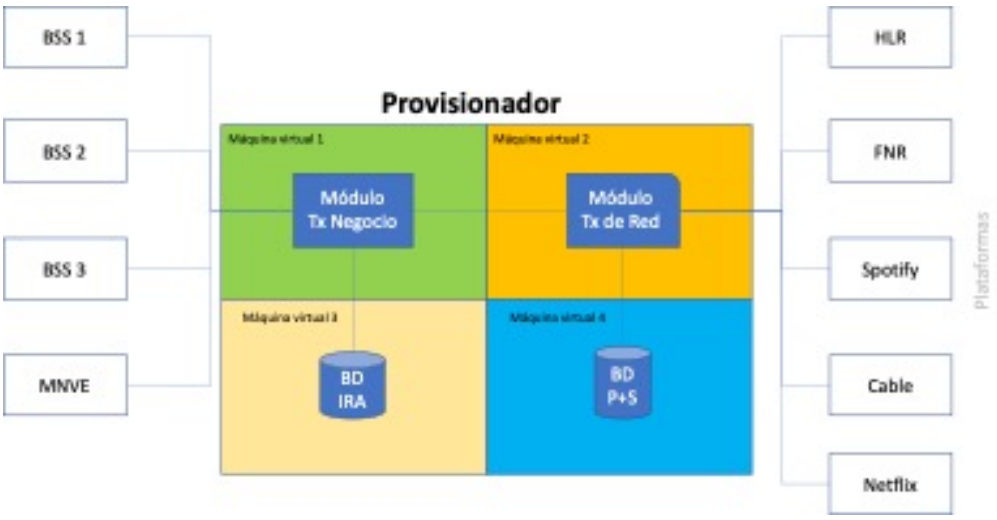


Figura 1.2: Arquitectura básica del Provisionador

Por su parte, el provisionador cuenta con los siguientes módulos:

1. **Módulo Tx de Negocio (Módulo de Transacciones de Negocio):** Este módulo está encargado de comunicarse con diferentes aplicaciones cliente (BSS, MNVE), y tiene configurada cada transacción de negocio posible de realizar (alta de una línea telefónica, modificación de un servicio, etc.), así como su relación con las transacciones de red. Además, este componente identifica a la aplicación de provisión respectiva, a la que deben impactar (es parte del módulo de transacciones de Red). Cuando las transacciones de red son completadas para una transacción de negocio, entonces se informa que la misma fue completada. En el caso de las transacciones masivas (por archivos) se disponibiliza el archivo con las respuestas cuando todas las transacciones

fueron finalizadas (con éxito o error). Este módulo cuenta con tres aplicaciones: Portal Web (comunicación web services), Servicio Online, Servicio Batch (recibe archivos de hasta 10 mil transacciones).

2. **Módulo Tx de Red (Módulo de Transacciones de Red):** Cuenta con dos tipos de aplicaciones: activadores y notificadores. Las primeras manejan instancias por cola de atención, e impactan las diferentes plataformas asociadas. Por ejemplo, en una TelCo se tiene configuradas 3 colas para: plataformas lentas, plataformas normales, y transacciones MNVO (operadores virtuales). Las notificaciones administran las respuestas, y cada módulo cuenta con varias colas e hilos de procesamiento.

1.4. Problema a abordar

El provisionador es un módulo importante en el funcionamiento operativo comercial de las empresas telefónicas, porque es el cuello de botella entre las operaciones de negocio (Venta, Postventa, Cobranza, etc.) y las plataformas. En el caso del provisionador de Amdocs, éste presenta los siguientes problemas:

1. El provisionador en ocasiones tiene una baja de performance provocada por: (1) un aumento en el procesamiento de principio a fin de las transacciones de negocio, es decir considera el flujo invocación del provisionador-respuesta de plataformas-respuesta al BSS, llegando al límite de la capacidad de Memoria RAM y/o CPU configurado, y (2) cambios en los flujos de transacciones debido a incremento en la demanda online o aumento del número de archivos que se gestionan, lo que provoca degradación del flujo de procesamiento. Estos inconvenientes impactan la venta o la postventa, incrementando tiempos de atención en el proceso de venta impactando los indicadores de comercialización diarias y creando ambiente de reclamos e insatisfacción que impactan la marca de la TelCo.

2. En casos críticos de soporte, no se ha cumplido con los niveles de servicio comprometidos (Service Level Agreement -SLA), lo cual implica penalidades y riesgo de que el cliente busque otro producto, al quedar la sensación de falta de robustez en el servicio que acompaña al producto. El equipo que brinda soporte al provisionador, puede demorar hasta dos días en realizar el análisis de una incidencia, y corregir el problema (o realizar recomendaciones), con lo cual no se cumple con los SLAs contratados.

3. El monitoreo y análisis de performance se vuelve complejo por ser ad-hoc a cada instalación de la provisión. Además, se puede tener los 2 módulos de la provisión y sus respectivas bases de datos en un único servidor, o bien puede tenerse instalado la provisión en cuatro servidores comunicados (como se muestra en la Figura 2), incluso con varias instancias de alguno de los módulos. Esto hace al análisis de performance particularmente lento en operaciones donde no se cuenta con acceso remoto (desde Chile) al ambiente de producción, en cuyo caso ha ocurrido que se ha tenido que enviar especialistas a realizar el análisis in-situ.

En el marco de este trabajo de tesis se abordarán los tres problemas antes mencionados, que afectan directamente la operación del módulo provisionador.

1.4.1. Hipótesis

Como hipótesis de la presente tesis se propone que es posible desarrollar una herramienta de monitoreo que ayude a detectar anomalías con antelación de 12 horas al reporte de una incidencia por detección humana.

Nota: Se considera 12 horas de acuerdo a lo revisado con especialistas, ya que con dicho tiempo existe la posibilidad de generar al menos una solución temporal para mitigar impacto mientras se resuelve el problema raíz.

1.4.2. Objetivos

El objetivo de la tesis es implementar un módulo de monitoreo con un mecanismo de detección temprana de anomalías y embeberlo en el provisionador de plataformas. Al detectar comportamientos extraños de manera temprana, se podrá alertar a los equipos de Producción de la TelCo para iniciar el análisis con más tiempo. Así se espera poder realizar acciones de manera temprana, planificada y oportuna, y de esa forma mitigar el impacto generado por los problemas antes descritos. En resumen, la implementación de este módulo de monitoreo y del mecanismo de detección temprana debería contribuir a reducir los tiempos de detección y resolución de anomalías en la infraestructura antes descrita.

A partir del objetivo general, se derivan los siguientes objetivos específicos:

1. Definir indicadores de comportamiento de datos de entrada, flujo y salida del proceso de provisionamiento. Los indicadores deben considerar las diferentes configuraciones de una instalación, por ejemplo:
   (a) Todos los módulos del provisionador en un solo servidor.
   (b) El módulo de Tx de Negocio con su base de datos en un servidor, y el módulo de TX de Red con su respectiva base de datos en otro servidor.
   (c) El módulo de Tx de Negocio, módulo Tx de Red, y cada Base de datos en un servidor diferente.
2. Implementar un proceso de extracción de datos que no degrade significativamente el proceso operativo del provisionador. Estos datos saldrán de la base de datos y de los archivos de log que produce el provisionador.
3. Implementar un algoritmo de detección temprana de anomalías que procese la información extraída, y basado en un modelo estadístico genere alarmas cuando detecte indicadores fuera del rango de normalidad. El algoritmo debe poderse adecuar manualmente, a partir de expertos y la data histórica, a las instalaciones en diferentes operaciones; es decir, el rango de normalidad debe poder graduarse a partir de datos históricos de la instalación.

1.5. Resultados esperados

Se espera desarrollar un módulo que monitoree la actividad del monitorear de forma de que se detecte ocurrencias de incidencias con una anticipación de 12 horas, antes que se reporte el evento. Esto permitirá que se pueda aplicar una solución temporal mientras se corrige el problema raíz.

1.6. Metodología

A nivel macro se realizarán las siguientes actividades:

1. Levantamiento de requisitos de equipo de producción y de casuísticas históricas:
   (a) Con el equipo de las Operadoras revisar los requisitos que debe cumplir el módulo de monitoreo para observar el comportamiento del módulo de provisión.
   (b) Revisar informes históricos de incidencias de Provisión y conversar con el equipo de Soporte sobre las causas raíz y las soluciones aplicadas. El objetivo es encontrar indicadores que reflejen anomalías del sistema.
   (c) Revisar otras soluciones de provisión y monitoreo del mercado, para extraer ideas y características.
   (d) Revisar papers, estudios de soluciones de monitoreo para ver tendencias e ideas.
   (e) Recopilar datos , logs de incidencias.
   (f) Recopilar configuraciones actuales del producto de provisión.
2. Análisis de casos e instalaciones:
   (a) Analizar los datos recopilados para determinar comportamiento de indicadores durante las incidencias verificar que los datos mantienen un comportamiento concentrado (como la campana gauss).
   (b) Analizar las diferentes instalaciones físicas del Provisionador en las actuales operaciones, (en un servidor compartido, o en otros servidores virtuales).
3. Diseño del módulo de monitoreo:
   (a) Definir Arquitectura de la solución (aplicativos, base de datos).
   (b) Determinar Indicadores, alarmas.
   (c) Diseñar el método para hallar el límite anómalo/normal.
   (d) Creación de casos de pruebas.
4. Construcción:
   (a) Fase 1: Prototipo (front-end con los indicadores).
   (b) Fase 2: Solución de Extracción de datos.

(c) Fase 3: Alarmas y módulo de detección temprana de anomalías.

5. Pruebas:

   (a) Ejecución de pruebas funcionales y de volumen para las diferentes fases.

6. Cierre:

   (a) Conclusiones y Comparación con respecto al inicio para ver si se logra detectar tempranamente y evitar anticipadamente casuísticas históricas..

7. Preparar informe:

   (a) Documentar el proceso con los resultados y las conclusiones.

Capítulo 2

Antecedentes

2.1. Marco teórico

2.1.1. ¿Qué son los provisionadores?

En el ambiente de las empresas de Telecomunicaciones, el provisionador o activador nace como la aplicación encargada de conectar las aplicaciones de negocio con plataformas físicas, como la HLR (central telefónica), la plataforma de SMS (mensajería de texto), y la FNR (central de numeración), entre otras. Con las nuevas tecnologías y servicios que han venido apareciendo en los últimos años, el provisionador se utiliza hoy para comunicar el negocio con plataformas digitales, tales como Spotify o Netflix; y también con nuevos servicios como sistemas IoT, y eSim (SIM digital).

2.1.2. Otros provisionadores en el mercado

El provisionador como componente abstracto tiene competencia hoy en día, ya que han aparecido nuevos conceptos, paradigmas y arquitecturas que abordan la comunicación entre las aplicaciones de negocio (BSS) y las plataformas físicas y digitales. Por ejemplo:

Enterprise Integrator de WS02

La plataforma de microservicios Enterprise Integrator de la compañía WS02 es de licencia gratuita y cuenta con un módulo de observability para monitoreo end-to-end de las transacciones que implementa. Puede comportarse como un Bus de servicios (ESB) o un provisionador; pudiendo interactuar con bases de datos y transformar datos utilizando lenguaje de scripting (XSLT, Xpath, etc); asimismo puede interactuar con herramientas de monitoreo externos como Prometeus, Grafana, FluentBit, ELK stack.

Simphonica de Intraway

Otro ejemplo es el provisionador Simphonica de la compañía Intraway, es un producto que funciona en cloud, comercializado como SaS(Software as service) y por licencia, pudiendo ser instalado en cloud privadas. La estructura interior que utiliza es orquestación de servicios / plataformas a partir de una transacción de negocio, brindando herramientas para la autogestión por parte del usuario(especialista) para crear flujo nuevos y ponerlos en producción. Cuenta con módulo de configuración web para ingresar nuevos flujos de orquestación, módulo de testing para probar el flujo creado o modificado y un módulo de monitoreo para analizar el comportamiento del flujo seleccionado. Tiene como objetivo que sea fácil configurar y poner en producción un nuevo flujo , y retroceder si es necesario, es decir incluye opciones de rollback. (ver artículo [12] "Codeless Provisioning Automation')

2.1.3. Otras soluciones de monitoreo

Prometheus

Prometheus es un sistema de monitoreo de código abierto basado en extracción y en métricas. Recopila datos de servicios y hosts mediante el envío de solicitudes HTTP en puntos finales de métricas. Luego, almacena los resultados en una base de datos de series de tiempo y los pone a disposición para análisis y alertas. Entre sus características recomienda medir : estado Latente (cuanto tiempo lleva atender una solicitud) , Tráfico(demanda del sistema) , errores(Tasa de saturación de solicitudes que fallan) y Saturación (ver si se degrada el servicio, si no se completan).

1. Habilita alertas cuando algo sale mal, preferiblemente antes de que salga mal. Para que alguien pueda echarle un vistazo.
2. Proporciona información para permitir el análisis, la depuración y la resolución del problema.
3. Le permite ver tendencias / cambios a lo largo del tiempo. Por ejemplo, cuántas sesiones activas en un momento dado. Esto ayuda en las decisiones de diseño y la planificación de la capacidad.

Permite las siguientes métricas: Contador (siempre aumenta), Calibre (indicador en un instante del tiempo), Histograma (por ejemplo duración de la solicitud), resumen(por ejemplo promedio de duración de respuestas), exportador, módulo de alertas , módulo de visualización y servidor central. Puede utilizarse Grafana del proyecto Apache 2.0 ara mostrar los datos recolectados por Prometheus. (ver artículo [5] "Introducción a Prometheus y Grafana")

FluentBI

Fluent Bit es un software procesador de registros, de código abierto y multiplataforma, que recopilar datos / registros de diferentes fuentes, los unifica y envía la información a múltiples

destinos. Permite procesar datos mediante querys internos en formato JSON, y configurar diferentes plugins como interfaces con plataformas. Maneja alertas y muestra información a través de otro producto (Grafana).

2.1.4. Detección temprana de anomalías utilizando distribución Gaussiana

La mayoría de las compañías telefónicas tienen software de monitoreo para las aplicaciones de la operación, las que muestran indicadores y generan reportes para la operación. Estas aplicaciones de monitoreo dependen de la información que brindan las plataformas, y usualmente no usan software para detección temprana de anomalías.

La detección de anomalías se ha utilizado en detección de fraudes , y en temas estadísticos de salud. En Telecomunicaciones la inversión en algoritmos de detección temprana de anomalías está más enfocada a ámbitos más cercanos al negocio. Estas empresas usan herramientas para detectar casuísticas de fraude, supervisar el aseguramiento de ingresos, y encontrar casuística de comercialización.

En general, en un sistema de software se espera que se comporte de manera constante cuando se procesa dentro de los límites de su arquitectura, se mantiene constante el uso que se hace de él (volumen de datos) y además son constantes las características del entorno: ancho de banda, cpu asignado, etc. En las telecomunicaciones se observan ventas diarias parecidas durante cada periodo estacional, cambios de planes más o menos constantes, y en general situaciones ordinarias frecuentes dentro de los límites del sistema. Los comportamientos especiales del negocio (de mayor exigencia del sistema) aparecen como consecuencia de campañas o eventos extraordinarios, los cuales incrementan la carga del sistema. Por lo tanto, a nivel macro el cliente (la Operadora) espera que el sistema tenga un comportamiento persistente, lo que se refleja en los indicadores de comportamiento. Se espera que los valores de estos indicadores no oscilen demasiado, y que haya una concentración de valores (datos) cuando se revisa el comportamiento general del sistema en situación normal (no anómalo); para esto utilizamos distribución gaussiana.

Distribución gaussiana

La distribución gaussiana (Normal), es una distribución de probabilidades de variable continua (regresión) que se utiliza para detectar tempranamente un valor(comportamiento), responde a la fórmula: $p_x(\mu, \theta^2) = \frac{1}{\theta\sqrt{2\pi}} \varrho^{\frac{(x-\mu)^2}{2\theta^2}}$, donde μ = media, θ = desviación estándar. La media es el valor medio de los resultados (el promedio) y la desviación estándar representa la dispersión numérica; es decir, qué tan concentrado o dispersos están los datos. Si el valor es muy alto, entonces están muy dispersos, y si es bajo entonces están muy concentrados alrededor de la media.

Si definimos el indicador de rendimiento como la cantidad de transacciones que procesa el provisionador por minuto durante un día, observamos el comportamiento de esta variable

en un día normal (graficado linealmente en la Figura 2.1) donde se ve una concentración de valores en su media.

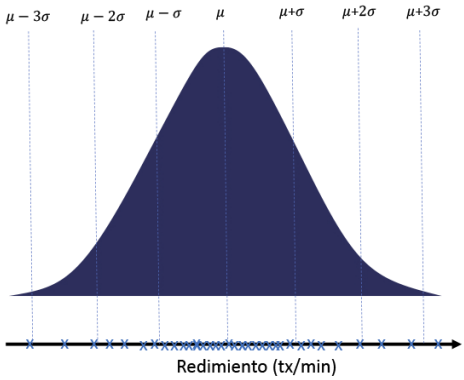


Figura 2.1: Distribución Gauss en muestra lineal

Linealmente se verá algo similar en la lectura de disco (IO). Estos dos indicadores están ligados entre sí: cuanto mayor cantidad de transacciones hay, mayor será el acceso a disco. Por esta razón, cuando las graficamos juntas. se observa una concentración de valores como se muestra en la Figura 2.2. En la tesis se revisará el uso de dos o más indicadores considerando que tienen concentración hacia la media, y esto lo podemos estudiar con la distribución gaussiana multivariable.

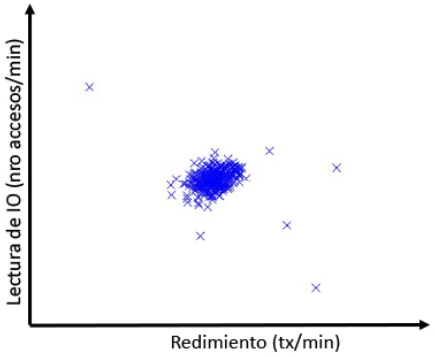


Figura 2.2: Rendimiento por Lectura de IO

Límite entre comportamiento normal y anómalo

Adicionalmente se debe calcular un límite de manera de que todo lo que se encuentre dentro del límite (ver línea roja en Figura 2.3) será normal y el resto anómalo. Este límite será definido a partir métodos que deberán definirse en la tesis. No se ve imprescindible tener un límite sobre ajustado a los datos de los indicadores calculados como en la Figura 2.4 (calculado con algoritmos de aprendizaje supervisado) ya que:

1. La detección de anomalías es el primer frente de notificación que gatillará alarmas

para que un analista revise y solucione los posibles problemas. Por lo tanto, interesa reconocer casuísticas positivas (anomalías) cuando son realmente verdaderas

2. Hay tolerancia por parte del equipo de operaciones y de soporte a revisar alarmas por positivos (anomalías) que resultan ser falsos.

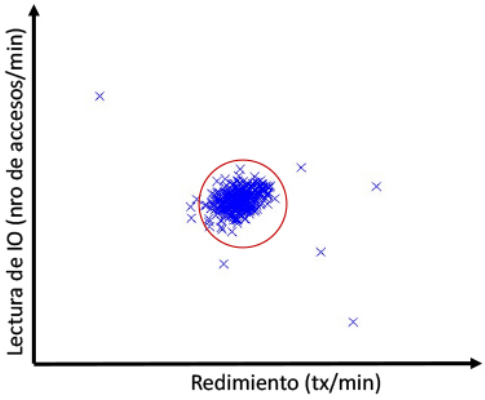


Figura 2.3: Límite calculado con distribución gaussiana

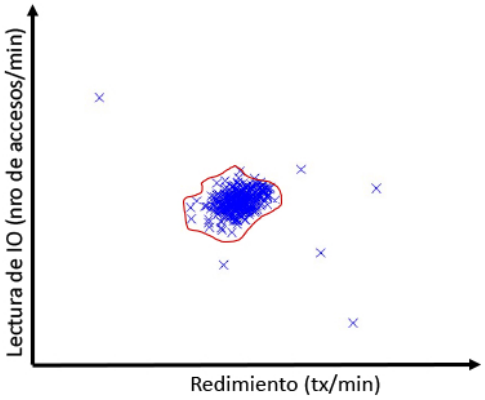


Figura 2.4: Límite ajustado, calculado con aprendizaje supervisado

Accurancy, Precision, Recall y F1-Score en modelos de detección temprana

En los modelos de Machine Learning se combina dos conceptos para poder tener el grado de confiabilidad: precisión y exactitud. A continuación, se explicita cada uno de ellos.

$$\text{ACCURANCY} = \frac{PositivosVerdaderos + NegativosVerdaderos}{Total}$$

La métrica Accurancy(Acertado) indica la proporción de valores positivos y negativos que acertó del total. Es decir la proporción de cuantas anomalías y NO anomalías acertó el algoritmo sobre el total de datos.

$$\text{Precision} = \frac{PositivosVerdaderos}{TotalPositivosPredichos}$$

La métrica Precision (Precisión)indica la proporción de valores positivos que acertó el algoritmo del total predicho. Es decir cuanto de los valores predichos como positivos (anómalos) fueron realmente positivos. (anómalos)

$$\text{Recall} = \frac{PositivosVerdaderos}{TotalVerdaderosReales}$$

La métrica Recall(Exactitud) indica proporción de valores positivos acertados del total real. Es decir cuantos de los valores positivos reales (anómalos reales) fue capaz de predecir el algoritmo.

Si tenemos 100 predicciones que da el algoritmo: 20 positivos (anómalos) y 80 negativos (no anómalos), y resulta que los 20 son verdaderamente positivos (anómalos) entonces la precisión es de 100 %. Sin embargo, si el total es de 30 positivos verdaderos (anómalos), significa que hay 10 positivos verdaderos no encontrados y además declarados como negativos (No anómalos); es decir, son falsos negativos, entonces la $Recall = \frac{20}{30} = 66\,\%$. Para poder relacionar ambos conceptos se utiliza:

$$\text{F1-Score} = 2\frac{Precision * Recall}{Precision + Recall} \text{ ,que varía entre 0 y 1}$$

(Ver artículo [10] "Material de curso-online de Coursera Machine Learning –Stanford University")

2.1.5. Detección temprana de anomalías utilizando Deep Learning

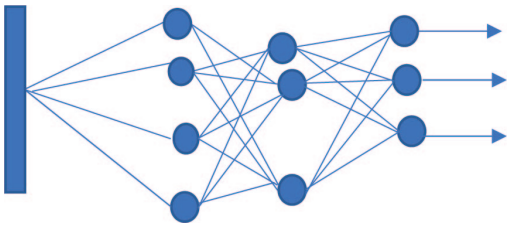


Figura 2.5: Red neuronal(Deep learning)

Utilizando algoritmos de aprendizaje automático supervisado, es posible entrenar una red neuronal (con el esquema de la Figura 2.5) de manera de reconocer anomalías, incluso determinar cual es el problema raíz, para esta se necesitan grandes cantidades de datos clasificados según cada problema raíz de manera de poder entrenar la red neuronal (alrededor de 15mil datos de diferentes casuísticas encontradas).

En esta tesis se descarta utilizar deep learning porque no se cuenta con una alta cantidad de datos de errores que además se encuentren clasificados, en especial porque no ocurren

tantos errores al año, pero se evaluará diseñar un repositorio para contar con dicha data clasificada en el futuro y en una siguiente etapa poder implementar una red neuronal para detectar tipos de errores/anomalías y poder realizar acciones de mitigación de manera automática.

2.2. Estudios previos

Dentro del marco teórico se revisaron publicaciones, cursos y otros productos para extraer propuestas que puedan enriquecer la solución que se propone

En el material del curso-online de Andrew Ng, en publicaciones de Sachim Shelar y Navonnel Chakrabarty se propone la detección de anomalías utilizando distribución gaussiana(Normal) y la multivariable gaussiana. (ver artículo [10] "Material de curso-online de Coursera Machine Learning –Stanford University", ver artículo [13] "Anomaly Detection using Gaussian Distribution" y ver artículo [1] "A Gaussian Approach to the Detection of Anomalous Behavior in Server Computers")

En el caso de la distribución normal, al utilizar dos variables, se puede utilizar una combinación de variables extraídas, definir por ejemplo $Performance = \frac{\text{Duración de Procesamiento}}{\text{Cantidad de Transacciones}}$.

Indican entre sus recomendaciones:

1. Que al ser la cantidad de anomalías típicamente un valor pequeño respecto a la cantidad de casos normales, entonces se puede hacer uso sólo de valores para casos normales en el ajuste del límite normal / anómalo. Es decir el Trainning set debe ser normal(no-anómalo)
2. Se recomienda que el Set de datos de testing debe contener datos anómalos y no-anómalos (normales) y el set de validación recomienda que tenga un 20 % de datos anómalos.
3. Si existen probabilidades de encontrar nuevas anomalías el día de mañana que no tienen nada que ver con otras que se han visto hasta ahora, entonces es más conveniente la detección mediante Distribución gaussiana en vez de utilizar algoritmos de Aprendizaje supervisado, puesto que este último trabaja sobre casos que ya han pasado y se repiten en el futuro. Si creyésemos que los casos futuros son similares a los ocurridos y se contase con anomalías clasificadas en volúmenes altos entonces si sería conveniente utilizar aprendizaje supervisado.
4. Trabajar los valores de las características seleccionadas para que tengan un comportamiento gaussiano, con lo cual es recomendable ajustar los valores de las características utilizando transformaciones logarítmicas. También se puede combinar otras características monitoreadas o agregar otras operaciones matemáticas para amplificar variaciones del comportamiento, por ejemplo $X = \frac{(\text{uso de CPU})^2}{\text{Tráfico de red}}$ permite apreciar mejores variaciones del uso de CPU, cuando la característica observada (uso de CPU) es normalmente muy alto.

En el manual del equipo SRE de Google (ver artículo [3] "Monitoring Distributed Systems"), para el monitoreo de Sistemas de distribuidos recomienda agrupar indicadores:

1. Estado latente: El tiempo que lleva atender una solicitud: exitosa o fallida. Es importante rastrear no solo las solicitudes exitosas sino también las fallidas.
2. Tráfico : Una medida de cuánta demanda se está aplicando a su sistema. Para un servicio web, generalmente son solicitudes HTTP por segundo.
3. Errores : La tasa de solicitudes que fallan.
4. Saturación : Qué tan completo es su servicio. El aumento de la latencia es a menudo un indicador importante de saturación. Muchos sistemas degradan su rendimiento mucho antes de alcanzar el 100 % de utilización.

En el artículo [5] "Introducción a Prometheus y Grafana", Vijay Khurana comenta que los tipos de métricas que utiliza Prometheus (un módulo genérico de monitoreo) son clasificados así:

1. Contador : El valor de un contador siempre aumentará. Nunca puede disminuir, pero se puede restablecer a cero. Entonces, si falla un raspado, solo significa un punto de datos perdido. El aumento acumulativo estaría disponible en la próxima lectura. Ejemplos: Número total de solicitudes HTTP recibidas, El número de excepciones.
2. Calibre: Un indicador es una instantánea en un momento dado. Puede aumentar o disminuir. Si falla la obtención de datos, pierde una muestra; la siguiente recuperación puede mostrar un valor diferente: ejemplos de espacio en disco, uso de memoria.
3. Histograma : Un histograma toma muestras de las observaciones y las cuenta en depósitos configurables. Se utilizan para cosas como la duración de las solicitudes o el tamaño de las respuestas. Por ejemplo, puede medir la duración de la solicitud para una solicitud HTTP específica. El histograma tendrá un conjunto de cubos, digamos 1 ms, 10 ms y 25 ms. En lugar de almacenar cada duración de cada solicitud, Prometheus almacenará la frecuencia de las solicitudes que caen en un depósito en particular.
4. Resumen : Al igual que en las observaciones de muestras de histograma, normalmente se solicitan duraciones o tamaños de respuesta. Proporcionará un recuento total de observaciones y una suma de todos los valores observados, lo que le permitirá calcular el promedio de los valores observados. Por ejemplo, en un minuto, tuvo tres solicitudes que tomaron 2,3,4 segundos. La suma sería 9 y el recuento sería 3. La latencia sería de 3 segundos.

También comenta que muchas aplicaciones exponen métricas en formato que no es de Prometheus. Para estos y para las aplicaciones que no son de su propiedad o para las que no tiene acceso al código, utiliza una herramienta de exportación para conversión de formato de datos.

El diseño contemplará el manejo de alarmas y dejará abierta la posibilidad de introducir acciones, como se plantea en el artículo de multiagente en servidores virtuales, donde el

monitoreo del sistema permite realizar escalamiento de ambiente productivo aumentando recursos virtuales. Para la presente tesis no se plantea realizar acciones pero si es bueno dejar una puerta abierta a la implementación de las mismas en un futuro. Se plantea el monitoreo con multiagentes , al igual que en el articulo Multiagentes (ver artículo [10] "Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure")

Capítulo 3

Análisis previo de datos

El objetivo de este capítulo es mostrar si es posible una detección temprana de anomalías, para lo cual se va a revisar de manera manual cuatro casos, correspondientes a distintos días. En tres de los casos hubo comportamiento anómalo que terminan en el reporte de una incidencia(evento anómalo) por parte del usuario, lo que implicó detener la operación para corregir problemas. En un cuarto caso hubo comportamiento anómalo pero este no se detectó en ese momento sino en un análisis. Recordemos que no se cuenta con mucha data histórica debido a que las anomalías detectadas por usuarios son espaciados en el tiempo, y por política de la Compañía se elimina data operativa por performance, ya que ésta no es parte de la información a conservar por requerimientos legales.

El objetivo de este análisis previo es verificar que es posible reconocer un evento anómalo antes que un humano lo detecte; por lo que se revisará el comportamiento del sistema y se comparará con la hora en que se reportó el problema, es decir, la hora en la cual un usuario levanta una queja respecto del funcionamiento del sistema. La idea es ver si un monitoreo automático de los datos producidos por el sistema podría haber levantado alarmas antes de que el problema se viera reflejado en el servicio.

La hora de registro de la incidencia se toma de la plataforma donde se registran los eventos(incidencias): Project & Portafolio management center(PPM) de Hewlett-Packard, donde el usuario de operaciones reporta la incidencia, y el equipo de Soporte de Amdocs responde, esta herramienta permite tener trazabilidad de las acciones realizadas durante la resolución de cada incidencia.

Este estudio se realizó revisando dos indicadores:

1. Duración: Tiempo promedio en segundos que demora el "Módulo Tx de Negocio" en procesar transacciones que se iniciaron en un intervalo de tiempo.
2. Cantidad de transacciones: Cantidad de transacciones que se recepcionan en un intervalo de tiempo.

En el análisis se calcularon los indicadores considerando el mismo intervalo de tiempo para cada muestra de manera de mantener consistencia. Para escoger el intervalo se tomó en

cuenta por un lado, que no sea muy grande, para poder detectar oportunamente la anomalía, y por otro lado, que no sea pequeño pues demasiadas consultas a la base de datos podrían degradar el sistema. En este proyecto se partirá probando con 5 minutos y se irá aumentando hasta lograr degradar el sistema lo menos posible y que siga siendo oportuno activar la alarma. A priori se considera que el intervalo sea menor a 20 minuto y mayor a 5 minutos, ya que el intervalo de 5 minutos ó 300 segundos permite detectar comportamientos anómalos considerando que el límite de la duración de una transacción es 30 segundos.

Cada indicador llega hasta su propio límite (que se muestran en línea punteada), pasado este límite es considerado anómalo, los valores de los límites se determinaron basado en juicio experto, luego de conversar con los especialistas de soporte y de la operación:

1. Límite Duración (que se muestra con una línea punteada en color naranja en los gráficos): 30 segundos, para el promedio de la duración de las transacciones que inician en el intervalo de 5 minutos.
2. Límite Cantidad (que se muestra con una línea punteada en color azul en los gráficos): 3.000 transacciones en 5 minutos.

El juicio experto seguido para encontrar el valor de los límites se revisan en el título 3.5 (Límite de comportamiento normal y anómalo)

Se recolectaron datos de 4 días completos (24 horas) y se analizaron utilizando los límites revisados con los especialistas, distinguiendo los criterios según si es horario comercial (10:00 a 19:00) o no comercial (00:00 a 10:00 y 19:00 a 23:59):

1. Para el horario comercial se utiliza : 30 segundos para la Duración y 3.000 transacciones para la Cantidad de transacciones; ambas variables calculadas para intervalos de 5 minutos, es decir si el promedio de transacciones procesadas en dicho intervalo excede alguno de los límites, si el promedio de procesamiento(Duración) excede los 30 segundos o el número de transacciones procesada en dicho intervalo excede las 3.000 transacciones, entonces se califica como una anomalía.
2. Para el horario No comercial se utilizó: 10 segundos para la Duración y 1.000 transacciones para la Cantidad de transacciones; ambas variables calculadas para intervalos de 5 minutos, es decir si el promedio de transacciones procesadas en dicho intervalo excede alguno de los límites, si el promedio de procesamiento(Duración) excede los 10 segundos o el número de transacciones procesada en dicho intervalo excede las 1.000 transacciones, entonces se califica como una anomalía.

En la tabla 3.1 se puede observar la cantidad de anomalías reales encontradas en el análisis previo utilizando los criterios antes descritos para cada horario, asimismo se muestra el total de transacciones por cada día analizado:

Tabla 3.1: Anomalías y transacciones analizadas

Horario	Métrica	caso 1 15-Julio	caso 2 16-Julio	caso 3 17-Julio	caso 4 03-Agosto
Comercial	Anomalía reales	13	19	61	14
	Transacciones	137.508	200.916	90.739	65.212
No Comercial	Anomalía reales	24	35	78	43
	Transacciones	374.385	300.035	47.534	279.829

3.1. Caso 1: 15-07-2020 sin incidencia reportada

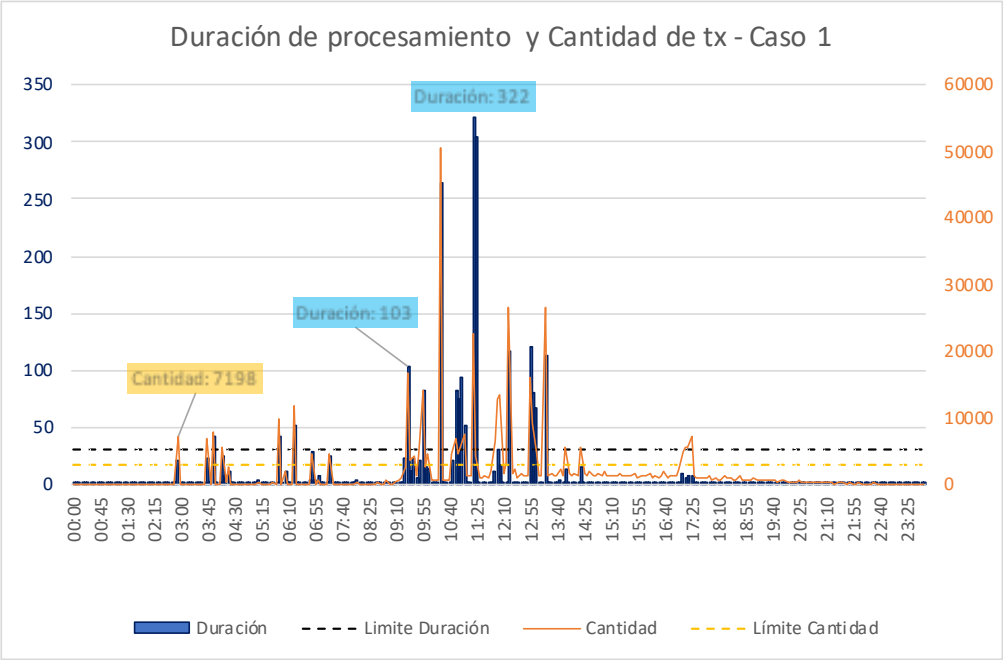


Figura 3.1: Gráfico de Duración promedio y Cantidad de transacciones del Caso 1

Se analizaron 511.893 transacciones procesadas en 24 horas, en la figura 3.1 se muestra la duración y la cantidad de transacciones en el mismo gráfico, así como los límites (en línea punteada); se observa que hubo procesamientos con altos valores de duración promedio, por encima del límite (línea puenteada azul). Si bien la mayor parte del día las transacciones se procesaron en menos de un segundo hubo un evento de 333 segundos (más de 5 minutos) a las 10:27 y otro de 399 segundos a las 11:24 (más de 6 minutos). Este comportamiento es explicable porque entre las 8:51 y 13:05 hubo una alta recepción de transacciones para ser procesados (línea naranja de la Figura 3.1), por lo que el sistema reportó varios peaks en la recepción de transacciones durante el horario comercial: 10:26 (26.534 transacciones), 11:24 (15.473 transacciones), 12:23 (21.077 transacciones), 13:23 (19.815 transacciones). Este día el sistema presentó anomalías en horario comercial (10:00 a 19:00), pero el usuario que corresponde al ejecutivo que atiende al cliente, no reportó problemas. Se procesaron correctamente las transacciones recepcionadas durante el día, pero el alto volumen impactó en la duración promedio lo que se reflejó en tiempos de hasta 6 minutos para cerrar transacciones de negocio, es decir si bien no se reportaron incidencias si hubo un malestar percibido por el

usuario (en horario comercial), traducido como lentitud del sistema.

Las anomalías aparecen desde las 02:55; al no haber monitoreo ni reclamos por parte del usuario, el equipo de operaciones no se enteró de este evento. Sin embargo, esta lentitud de procesamiento debió haber producido una advertencia del sistema para que el equipo de operaciones monitoree el funcionamiento del sistema por si aumentaba la degradación y anticipara acciones.

3.2. Caso 2 : 16-07-2020 con anomalía y sin incidencia reportada

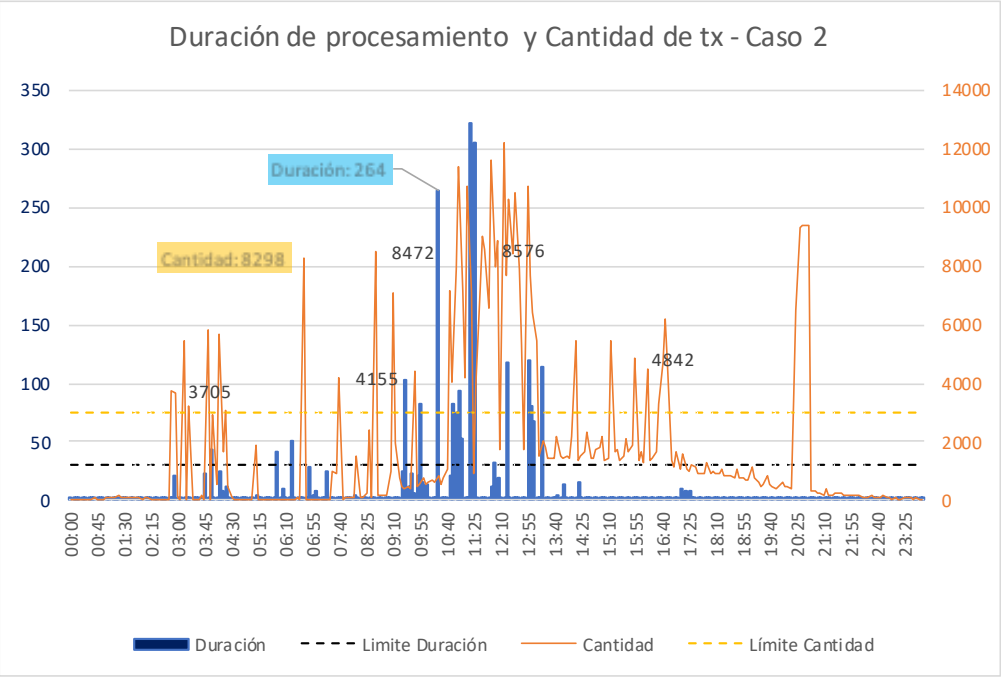


Figura 3.2: Gráfico de Duración promedio y Cantidad de transacciones del Caso 2

Se analizaron 500.951 transacciones procesadas durante el día, donde se observa en la Figura 3.2 que hubo varias transacciones con duración más altos que el límite de la duración promedio entre 30 y 65 segundos desde las 02:54; se verifica que la cantidad de transacciones recepcionadas tuvo un peak de 8.232 transacciones en 5 minutos, los cuales también son valores mayores al límite de cantidad de transacciones establecido, siendo en general, estas cantidades menores a los peaks del Caso 1 (Figura 3.2).

El peak de las 02:50 en la cantidad de transacciones recepcionadas debió levantar una alarma para que un analista monitoree y revise si había algún riesgo de degradación del sistema. Durante el día la cantidad de transacciones aumentó y pasó el límite (línea punteada naranja) en varias ocasiones y también aumentó la duración de las transacciones, llegando a peaks de 300 segundos (5 minutos). Es decir, hubo una lentitud del sistema, que el equipo de Operaciones debió monitorear, pero no se enteraron de esto.

3.3. Caso 3: 17-07-2020 con incidencia reportada

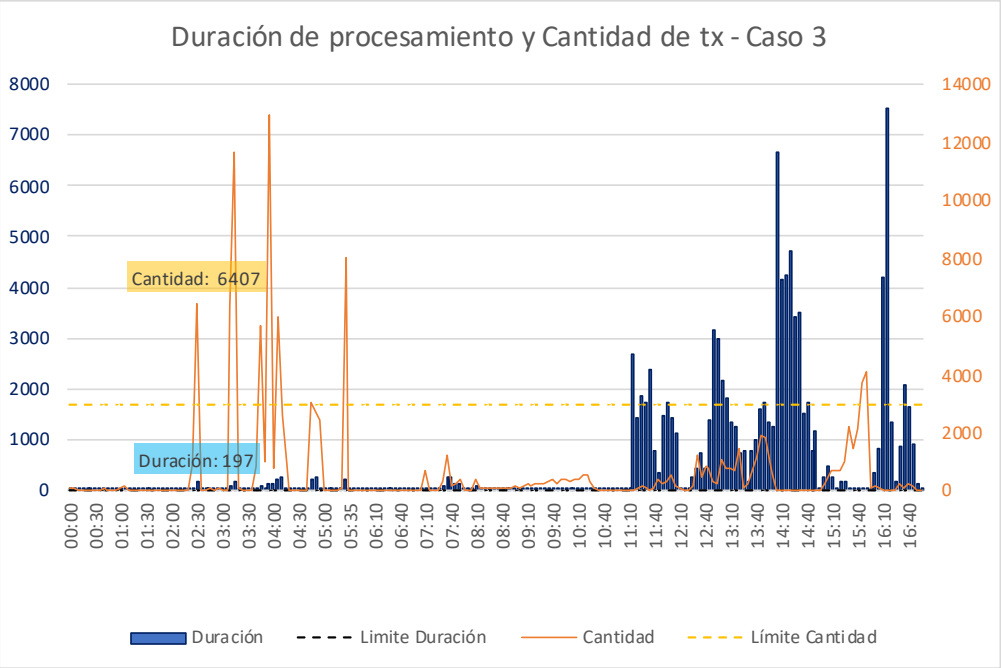


Figura 3.3: Gráfico de Duración promedio y Cantidad de transacciones del Caso 3

En este caso se analizaron 138.273 transacciones procesadas en el día, siendo un número menor de procesamiento respecto a otros días porque el sistema estuvo detenido, razón por la cual se reportó una incidencia.

A las 12:00 los usuarios llaman al equipo de Operaciones (Help Desk) para reclamar por lentitud en los módulos de ventas y posventas. El equipo de Help Desk observó encolamiento en el procesamiento de transacciones, es decir hubo muchos procesos pendientes de ser procesados. Como medida inicial el equipo de Operaciones reinició el servidor de aplicaciones donde se encuentra el Modulo de Tx Negocio, observándose que se corrige el comportamiento del funcionamiento del sistema, al menos temporalmente. Sin embargo las llamadas continúan por lo que se registra la incidencia en PPM para recibir soporte.

La incidencia se reportó a las 15:59, el equipo de Soporte de Amdocs empieza a las 16:00 a recolectar información, y al revisar los logs, detectó que hubo una falla de conexión del Módulo de Negocio con la aplicación de Origen (Oracle Service Bus) durante el proceso de notificación, que es el último paso del flujo.

El problema de comunicación continúa ocurriendo de manera intermitente, por lo que se decide instalar el Notificador en el mismo servidor virtual donde se encuentra la base de datos del módulo de Negocio. Esta acción se realiza después de verificar que la comunicación por red entre la aplicación de Notificación y la aplicación de Origen es estable. La acción mitigadora se realizó a las 22:00. En paralelo se prepara una versión del Notificador con un log con mayor nivel de detalle para poder realizar un mejor análisis y encontrar la causa raíz del problema.

A las 6:00 del día siguiente, luego de modificar el Notificador, se reinstala esta aplicación en

el servidor virtual que le corresponde y se encuentra que existían problemas de comunicación intermitente con base de datos del Modulo de Negocio, pero con la medida tomada ya no se pierde comunicación con la aplicación Origen.

Recordemos que la solución de Provisión utiliza cuatro servidores virtuales que están en una granja de servidores; cada módulo y cada base de datos se encuentran en un servidor virtual distinto. Estas son: Modulo de Tx Negocio y el Notificador, Módulo de Tx Red, base de datos de Módulo de Negocio, y Base de datos de Módulo de Red. Las bases de datos estaban en segmentos virtuales de red distintos a los módulos de las aplicaciones; por políticas de seguridad del Cliente los datos deben estar en un segmento de red especial para datos.

Sin embargo, se detecta un nuevo problema: en la Figura 3.3 se observa que hay transacciones con peaks de 192 segundos (3 minutos) desde las 02:30 de la mañana, llegando a 425 segundos (7 minutos) a las 7:28; también se observa que a las 11:00, existen demoras de 2.600 segundos (43 minutos), llegando hasta 15.869 segundos (más de 4 horas) que coinciden con las quejas de los usuarios. Los datos posteriores a las 15:00 están distorsionados porque se manipuló la base de datos para procesar un grupo de transacciones al día siguiente, de manera que al reiniciar la Provisión no se tomen en cuenta y así priorizar la normalización de otras transacciones.

Como solución final se decide mover los servidores virtuales de los Módulos de Tx Red , Notificador y de Tx Negocio al mismo clúster físico de las base de datos, manteniéndose cada uno en su servidor virtual y conservando segmentación virtual distinta; con esta modificación se corrigió el problema.

De acuerdo a la Figura 3.3 existen indicadores para haber podido levantar alarmas y empezar a revisar desde las 02:00, incluso desde el día anterior (Caso 2 - Figura 3.2) cuando se detectan anomalías.

3.4. Caso 4: 03-08-2020 con incidencia

En este caso se analizaron 346.041 transacciones, habiendo habido una lentitud del sistema que fue reportado por el usuario en una incidencia.

Esta incidencia se reportó a las 17:59, y se indica: "Afectación en el procesamiento de la aplicación Origen, observando lentitud en el proceso de notificació". La aplicación Origen es Oracle Service Bus (OSB) que llama al Provisionador. Este último impacta plataformas internas de la compañía y externas como Spotify, para finalmente notificar la respuesta a la aplicación Origen.

En el análisis de los logs se encontró que hubo aumento en el tiempo de notificación debido a una malformación de respuesta NAK del Origen y se detuvo totalmente la notificación por cierre de puerto comunicación (de un firewall) utilizado por el notificador, al parecer por error humano. Adicionalmente hubo tiempos altos en respuesta de plataformas externas, que impacta el Provisionador.

El problema se solucionó a nivel de red cuando se reabrió el puerto de notificación y

se reenviaron los movimientos desde la aplicación Origen, de esta manera finalizaron las transacciones que se encontraban encoladas, se procesó y pasó a tablas de registro como parte del tratamiento de información histórica.

Se observa que desde 03:15 hay transacciones que demoran hasta 1.119 segundos (más de 18 minutos) en procesarse debido a lentitud en plataformas externas (Spotify, etc.) que impacta el provisionador.

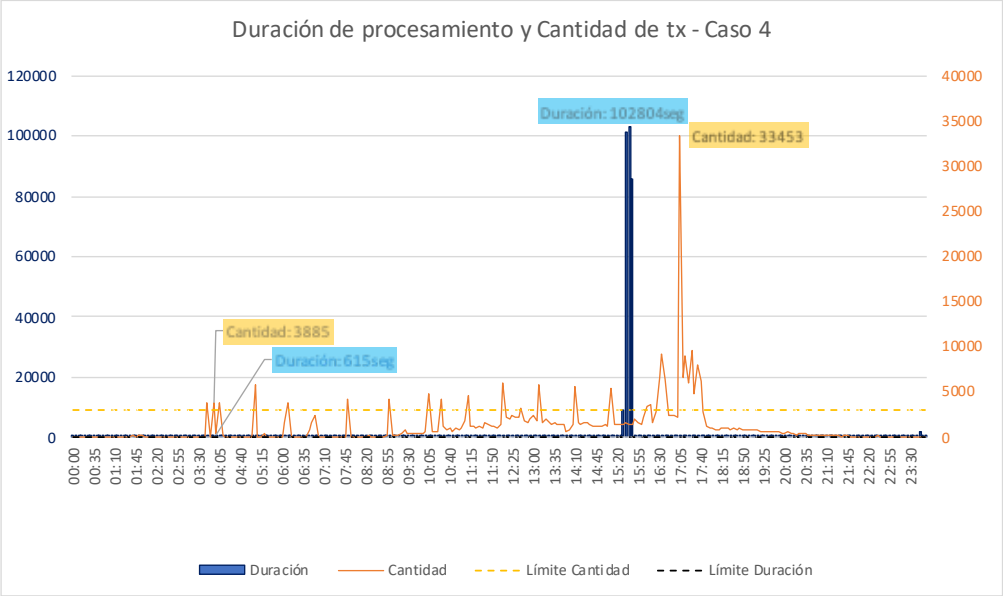


Figura 3.4: Gráfico de Duración promedio y Cantidad de transacciones del Caso 4

Se puede ver que a las 15:17 existen transacciones que llegaron a procesarse en 27 horas, esto se debe interpretar como una detención total de la notificación y su posterior regulación, ocasionado por el cierre del puerto. Las diferencias en el comportamiento que muestra antes de las 14:00 no se aprecian bien en la Figura 3.4 debido al cambio de escala donde se pasa de miles a cientos de miles de segundos de duración.

Como se muestra, hay un comportamiento anómalo en el sistema desde las 03:15 relacionada a la lentitud que se reportó , asimismo hubo una detención de la notificación a las 15:17 que se reporta a las 17:59, existiendo varias horas que se pudieron haber empleado en corregir los diferentes problemas que se venían registrando.

3.5. Límite de comportamiento normal y anómalo

Para el valor del límite de duración de una transacción normal se tomará 30 segundos, que es el 10 % del intervalo de medición (5 minutos = 300 segundos). Los especialistas indicaron que si bien el tiempo promedio normal que demora cada transacción en ser procesada es menor a 1 segundo, se debe considerar un valor mayor a este último para que el módulo de monitoreo no levante alarmas innecesarias cuando el sistema aún puede recuperarse en su desempeño; por ejemplo el tiempo de procesamiento aumenta cuando llegan muchas transacciones, pero

al no ser constante la cantidad de transacciones recepcionadas el sistema procesa lo pendiente en intervalos de menor actividad, poniéndose al día.

El límite para la cantidad se declara en 3.000 transacciones cada 5 minutos, este valor se validó con especialistas y se encontró recopilando antecedentes del sistema: la última actualización de la Provisión fue certificada para procesar 1 millón de transacciones por día, es decir fue certificada para procesar 3.472 transacciones cada 5 minutos, redondeando el límite a 3.000. Actualmente se procesan entre 350 mil y 500 mil transacciones por día.

Los límites propuestos son un buen inicio, pero cuando el proceso de detección de alarmas opere, se debe probar con varios valores para optimizar el encendido de alarmas. Por lo tanto se realizó el correspondiente análisis buscando:

- Ratificar o corregir valores de los límites de Duración y Cantidad como frontera del comportamiento normal y el anómalo en Horario comercial
- Revisar si vale la pena implementar límites distintos para horario comercial y NO comercial
- Revisar si existe concentración de datos no anómalos al cruzar información de ambos indicadores, es decir si es aplicable el modelo estadístico de la normal.

En el análisis de los datos y en conversaciones con los expertos se observó que el sistema es exigido de manera distinto en Horario comercial y en horario no comercial. Se define Horario comercial, entre 10:00 y 19:00, y Horario No comercial para el resto del día; el horario comercial corresponde a la atención al público en oficinas en centros comerciales.

Se verifica que la concentración de datos para el horario no comercial ocurre para límites Cantidad:1.000 y Duración:10; que coincide con lo expresado por los expertos que en horario no comercial el sistema es exigido sólo un 30 % respecto a lo que ocurre en horario comercial.

En la Figura 3.5 se muestra "cantidad de transacciones" vs. "duración de procesamiento", comportamiento del sistema para el caso 1. Se aprecia una concentración de datos, puntos de color verde dentro de los límites y en rojo los datos fuera de los límites. Recordemos que para el horario comercial se consideran el límite de Cantidad: 3.000 y el límite de Duración: 30.

Para poder apreciar mejor la concentración se ha graficado cantidades menores a 12.000 y duraciones menores a 120; es decir hay valores anómalos más alejados (dispersos) que no se muestran en el gráfico.

En la Figura 3.6 se muestra comportamiento del sistema en horario No comercial para el mismo caso 1, utilizando los límites de Cantidad: 1.000 y Duración: 10 se observa una concentración de puntos de color azul y también se aprecia dispersión de valores en rojo fuera de estos límites. Como se observa visualmente si se utilizaran los límites de horario comercial se tomaría como valores no anómalos los valores dispersos.

En la figura 3.7 se puede observar la concentración de datos en verde (para horario comercial), datos en azul (horario No comercial), y en rojo los valores dispersos, con lo cual los valores de los límites también se cumple para el Caso 2.

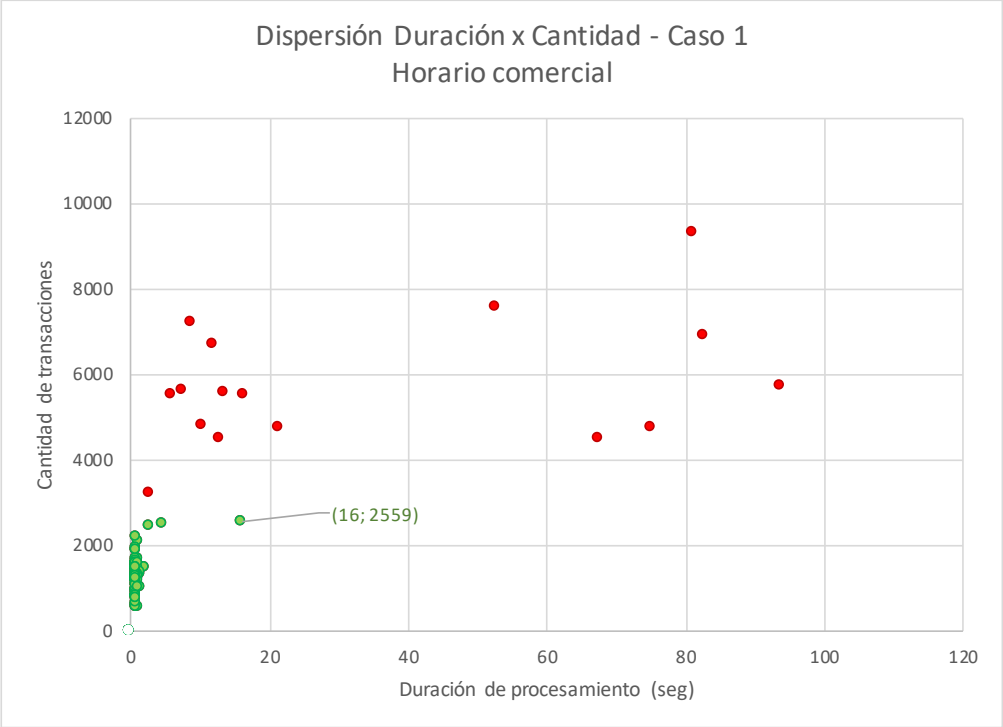


Figura 3.5: Dispersión para el Caso 1: Horario comercial

En la figura 3.8 también se observa concentración para diferentes límites, para horarios comercial y No comercial.

En la Figura 3.9 se observa concentración de datos No anómalos dentro de los límites: verde para horarios comercial, azul No comercial; y los datos en rojo muestran eventos anómalos.

3.6. Resultado del análisis previo

Al revisar todas las gráficas de dispersión se repite el patrón que existe concentración de datos para los valores no anómalos, tanto en horario comercial (en verde), como en horario no comercial (azul), considerando diferentes límites para cada horario; al existir concentración de la mayor cantidad de datos se puede utilizar la distribución normal de gauss.

Los valores de los limites distintos por horario comercial y no comercial coinciden con los limites de la concentración de datos y se observa también una mayor dispersión para valores fuera del límite.

Como se ha visto anteriormente el análisis previo permitió:

1. Demostrar que hay concentración de datos para datos normales (no anómalos).
2. Mejorar la identificación de datos anómalos utilizando límites distintos para horarios comercial y No comercial.

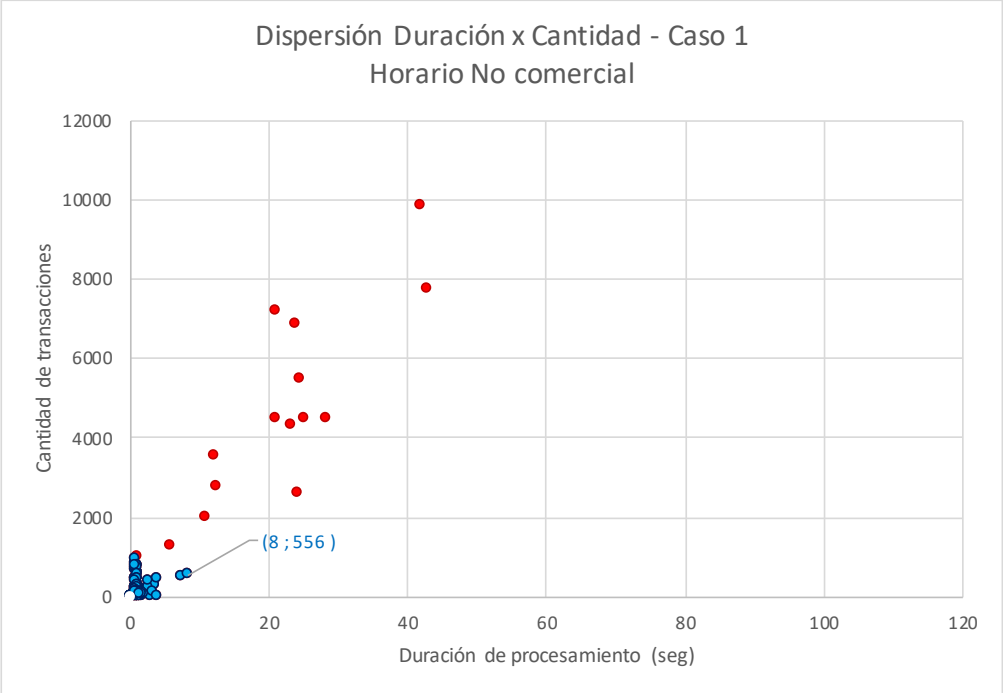
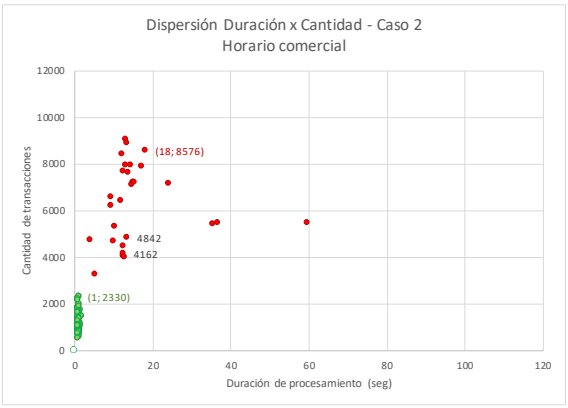
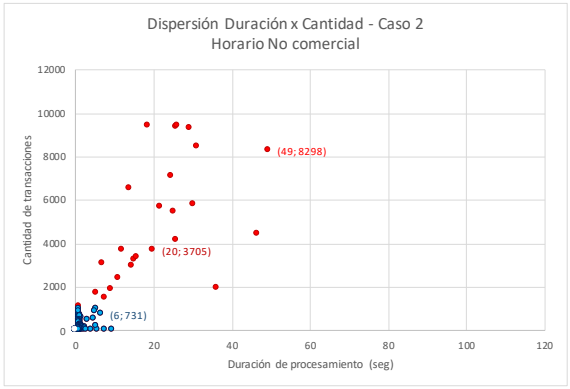


Figura 3.6: Dispersión para el Caso 1: Horario No comercial

3. Encontrar los valores de los límites.

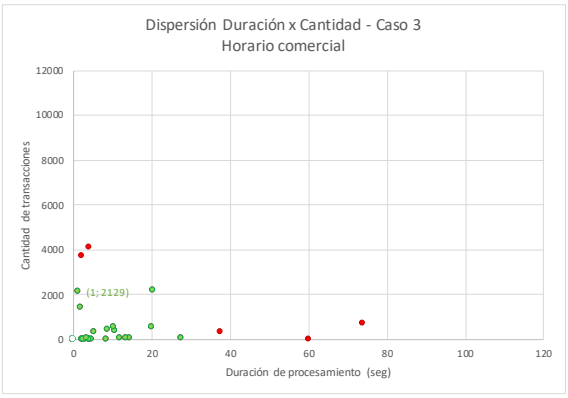


(a) Horario comercial

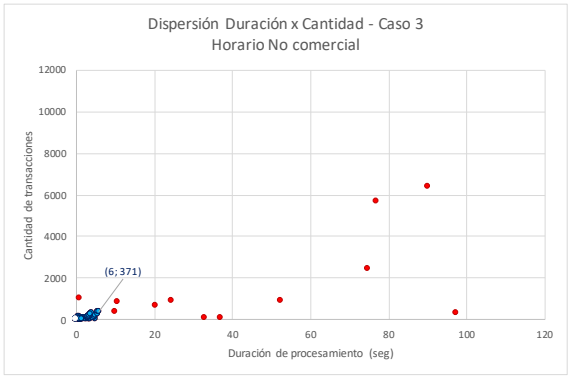


(b) Horario No comercial

Figura 3.7: Dispersión para el Caso 2

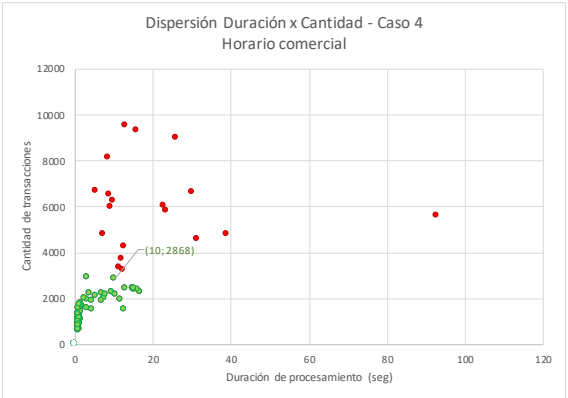


(a) Horario comercial

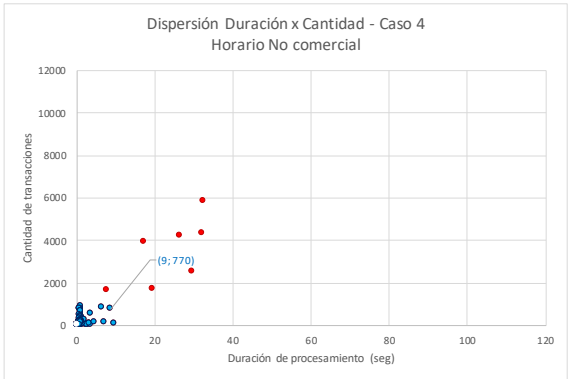


(b) Horario No comercial

Figura 3.8: Dispersión para el Caso 3



(a) Horario comercial



(b) Horario No comercial

Figura 3.9: Dispersión para el Caso 4

Capítulo 4

Solución

El objetivo buscado es contar con una herramienta para actuar de manera temprana ante la ocurrencia de anomalías y disminuir o corregir más rápido los efectos en el negocio cuando la anomalía perturbe la operación(comercialización).

Para lograr esto, se construirá el módulo de monitoreo, el cual mostrará información necesaria (gráficos e indicadores) para ayudar al especialista a una idea de la salud del sistema. Además, este módulo implementará la componente de detección temprana de anomalías. Esta componente detectará comportamientos anómalos de manera automática y levantará una alarma cuando se detecte un caso no normal, adelantándose a la percepción de un humano. La información que muestra el módulo de monitoreo está pensada para permitir al analista contar con información necesaria para realizar el análisis de incidencias (anomalías) y revisar tendencias que permitan identificar una degradación del sistema sin tener que revisar los Logs, para acelerar de esta manera el análisis y por lo tanto la reacción.

4.1. Datos a monitorear

Se ha elegido un conjuntos de indicadores para medir el estado en que se encuentra el sistema, donde una de las principales dimensiones es el tiempo: cuanto demora una transacción en el sistema y cuanto demora en algunas partes del proceso. Para esto se miden la cantidad de transacciones recepcionadas, así como la cantidad de transacciones por Estado, en general Información Tx Negocio. Esto nos permite tener una imagen de como está trabajando el sistema. También se consideran indicadores de Demanda del sistema, para lo cual se miden recursos utilizados y disponibles del sistema(Información de Servidores): CPU, Disco, Memoria física, Memoria Swap.

Para facilitar el análisis, posterior a que se gatille una alarma, se muestra el total de errores y se buscan indicadores que reflejen la saturación del sistema, para poder revisar si hay una degradación del sistema: Totales de transacciones y movimientos por Estado. El objetivo es permitir el análisis sin tener que revisar los logs.

El algoritmo de detección temprana de anomalías utilizará dos indicadores : Duración

promedio de transacciones y Cantidad de transacciones por unidad de tiempo.

Los siguientes son los indicadores que se consideran en el módulo de monitoreo:

4.1.1. Información Tx Negocio

1. Duración promedio de transacciones
2. Cantidad de transacciones recepcionadas por unidad de tiempo
3. Cantidad de archivos por estado (pendientes/en proceso/procesados) por unidad de tiempo.
4. Cantidad de transacciones por tipo de proceso y por estado: ejecutados correctamente, ejecutados con error, pendientes de notificar.

4.1.2. Información de Módulo Tx de Red

1. Cantidad de movimientos por estado (pendiente de ejecución , en ejecución, pendiente de notificar, error).
2. Tiempo de respuesta promedio de cada plataforma.

Nota: Recordemos que cada transacción de negocio orquesta uno o varias transacciones de red(movimientos).

4.1.3. Información de Servidores

1. CPU utilizada. (valor actual, medido cada minuto)
2. Disco disponible. (valor actual, medido cada minuto)
3. Memoria física disponible. (valor actual, medido cada minuto)
4. Memoria Swap disponible. (valor actual, medido cada minuto)

4.2. Algoritmo de detección temprana de anomalías

El algoritmo implementa la distribución gaussiana para evaluar dos indicadores: Duración y Cantidad. Como vimos en el capítulo Antecedentes, sección 2.1.4 Distribución Gaussiana este algoritmo utiliza dos parámetros: media y desviación estándar (dispersión), donde la media es el valor medio de los resultados (el promedio) y la desviación estándar representa la dispersión numérica, es decir que tan concentrados están los datos.

En el algoritmo se muestran las siguientes variables:

1. X : es una matriz con r registros, cada registro es producto de procesar información en un periodo, en este piloto se procesó información recepcionada durante 5 minutos, para cada registro se han calculado los dos indicadores: Duración y Cantidad, almacenados en la columna 0 y 1 (X[r,0] y X[r,1])
2. epsilon : es una constante utilizado para regular la asertividad del algoritmo, se varia manualmente entre cero y uno.
3. media y desviación_estándar: cada una de estas matrices tiene dos valores, para Duración y Cantidad.

Se definió el algoritmo de manera que pueda escalar en el futuro y así incorporar nuevos indicadores que serán evaluados por el algoritmo.

Algoritmo:

media= matriz con media de valores normales por cada indicador i
desviación_estándar= matriz con dispersión de valores normales por cada indicador i
X = matriz de r registros e i indicadores, donde X[r,0] es Duración y X[r,1] es Cantidad
while Existan registros sin leer en X **do**
    Leer registro X[r]
    for cada indicador i a evaluar **do** ▷ i = 0;1 porque sólo hay dos indicadores
        P[r,i]= distribucionGaussiana (X[r], media[i], desviación_estándar[i])
        if $P[r, i] \leq epsilon$ **then**
            Es una anomalía, informar alarma
        else
            NO es una anomalía
        end if
    end for
end while

Se calculan los valores de los parámetros: media, desviación_estándar a partir de datos de entrenamiento, se calculan para cada indicador (Duración, Cantidad). Se carga la matriz a analizar con datos por cada indicador (Duración, Cantidad) en intervalos de 5 minutos , es decir se registra la "Duración" promedio y la "Cantidad" de transacciones que iniciaron en el intervalo de 5 minutos; Se lee cada registro de la matriz a analizar y se calcula la función de la distribución de gauss por cada indicador, si el resultado es menor a epsilon entonces se califica como una anomalía.

4.3. Diseño arquitectónico

4.3.1. Principios arquitectónico

El diseño del módulo de monitoreo consideró los siguientes principios que se levantaron en reuniones con usuarios del sistema y con especialistas que brindan soporte a la Provisión. Se abordan dos categorías: primero Negocio, agrupando aquellas necesidades funcionales de

la operación que requiere la comercialización de telecomunicaciones, y segundo Tecnología, apuntando a definir límites tecnológicos en los cuales implementar el nuevo módulo.

Principios de Negocio

1. La información debe poder ser consultada desde una página web con varios perfiles, incluyendo uno para generación de reportes para la Jefatura
2. Los respaldos de información deben quedar en una carpeta determinada por el área de seguridad de información.
3. En un futuro debe poderse incorporar la funcionalidad de realizar acciones en los procesos de la Provisión, como permitir al perfil del equipo de Operaciones: bajar y subir procesos.
4. El equipo de Soporte debe poder consultar la aplicación dentro de la red de la compañía
5. En una primera fase, las alarmas serán en la página web; pero se debe permitir que escale en el futuro a otras canales como enviar emails o mensajes a celular.
6. No afectar la performance del Provisionador.

Principios Tecnológicos

1. Utilizar tecnología usada actualmente en el producto de Provisión (Linux RedHat, J2EE, C++, java, php) excepto productos Oracle(base de datos y Weblogic), estos últimos debido que las empresas Cliente requieren disminuir costos en licenciamiento.
2. Utilizar tecnología objetivo definida por estrategia de evolución de la compañía (Angular, Spring, PosgreSQL, Tomcat, Python)
3. El monitoreo no debe degradar la performance de la Provisión

4.3.2. Flujo de datos

En la Figura 4.1 se muestra que el módulo de Monitoreo estará formado por un conjunto de procesos agrupados en cuatro fases: extracción, análisis de información, presentación de información y paso a Histórico. Los otros procesos y módulos de la Provisión están bajo el título de Provisión.

Durante la Extracción, el proceso capturará datos desde las tablas de registro de los módulos de Negocio y de Red de la Provisión, y también leerá información desde archivos de datos. Estos últimos serán generados por los procesos de Generación de Logs (Negocio y Red) de la Provisión, que serán modificados para que escriban información en dichos archivos en un formato ad-hoc. Estos procesos actualmente escriben información en archivos Logs, con lo cual se busca conservar la carga de procesamiento para mantener la performance operativa. Adicionalmente Se instalará un agente en cada servidor virtual para tomar lecturas de datos del servidor (ocupación de cpu, memoria swap, etc.). Estas lecturas, se ejecutarán

en intervalos de tiempo configurables, buscando afectar lo menos posible la performance de los procesos de la Provisión existentes, así se atenderá el principio arquitectónico tecnológico.

Durante el análisis de información, se ejecutarán dos procesos: el proceso Generación de indicadores que toma los datos almacenados durante la extracción y de acuerdo a la configuración (periodo de evaluación, etc.) genera los indicadores mencionados en la sección 4.1. En paralelo, el proceso de Detección temprana de Anomalías evalúa los indicadores Duración y Cantidad, de manera de gatillar Alarmas cuando corresponda.

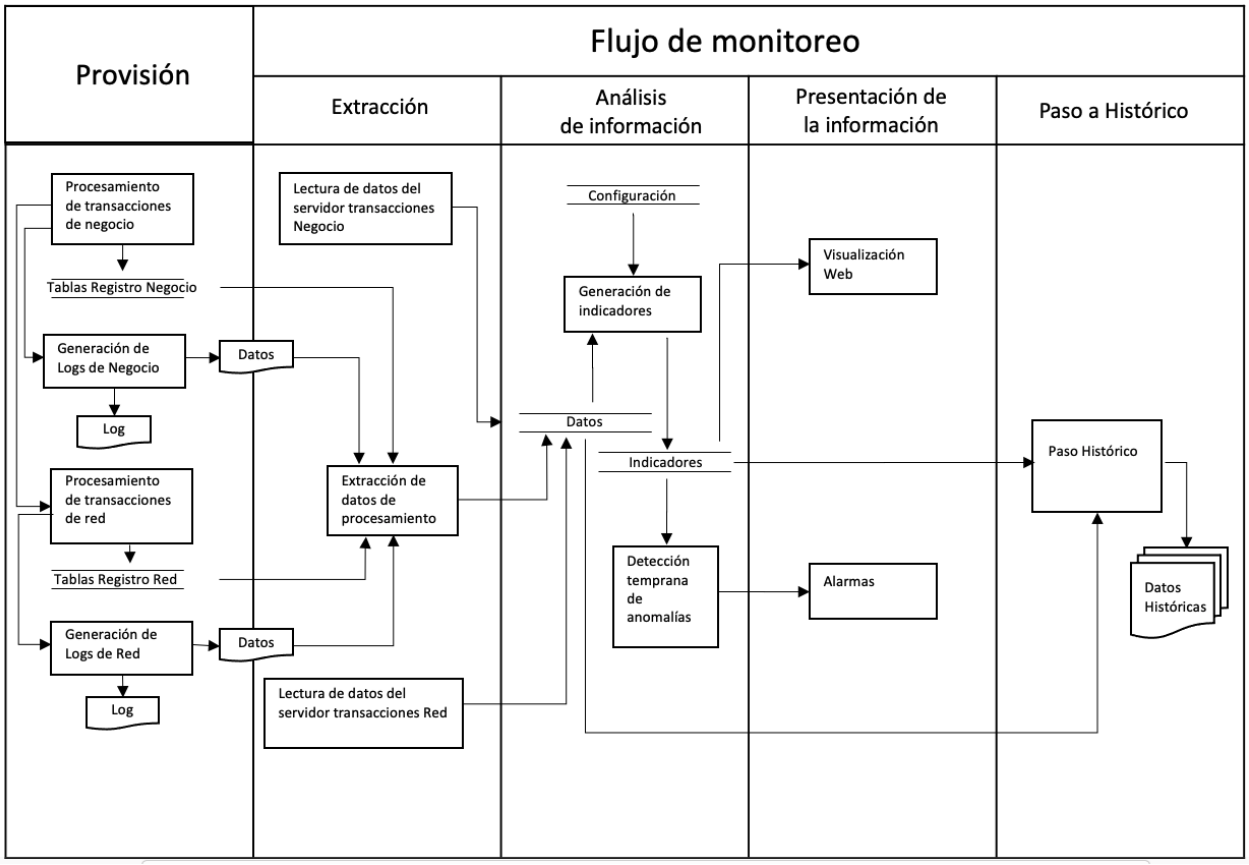


Figura 4.1: DFD del módulo de monitoreo

La Presentación de Datos mostrará los indicadores en diferentes niveles de manera de facilitar el análisis del sistema. Las Alarmas se ejecutarán de acuerdo a la configuración registrada (acción, mail, mensaje, lista de distribución, etc)

Finalmente el Paso Histórico limpiará las tablas de indicadores y datos pasados, trasladando la información a un almacenamiento histórico (archivos); de esta manera se mantendrá la performance de ejecución de los procesos de análisis de información.

4.3.3. Diagrama de Contexto

En el Diagrama de contexto se mostrará la interacción de la Solución de Provisión con otros sistemas, en los siguientes diagramas (Contenedores y Componentes) se va detallando más el módulo de la Solución de monitoreo hasta mostrar sus componentes.

En la Figura 4.2 se muestran las interacciones de la solución de Provisión en el marco de la compañía de Telecomunicaciones y las interacciones con otros sistemas de la compañía y otras compañías. La solución de monitoreo será un módulo de la Provisión. En el gráfico se muestra la Provisión dentro de la compañía de Telecomunicaciones, se puede ver la interacción con otros sistemas: el BSS y el MNVE; el primero es una solución utilizada por la misma compañía para operar el negocio(ventas, posventa, facturación) y el segundo (MNVE) es la solución para que otras compañías de Telecomunicaciones puedan utilizar la infraestructura(red) de la compañía anfitriona.

El Provisionador recepciona transacciones del BSS y MNVE para gestionar acciones con las diferentes plataformas dentro de la compañía, también con otras compañías(Cable) y con plataformas en la nube. (Spotify, eSim)

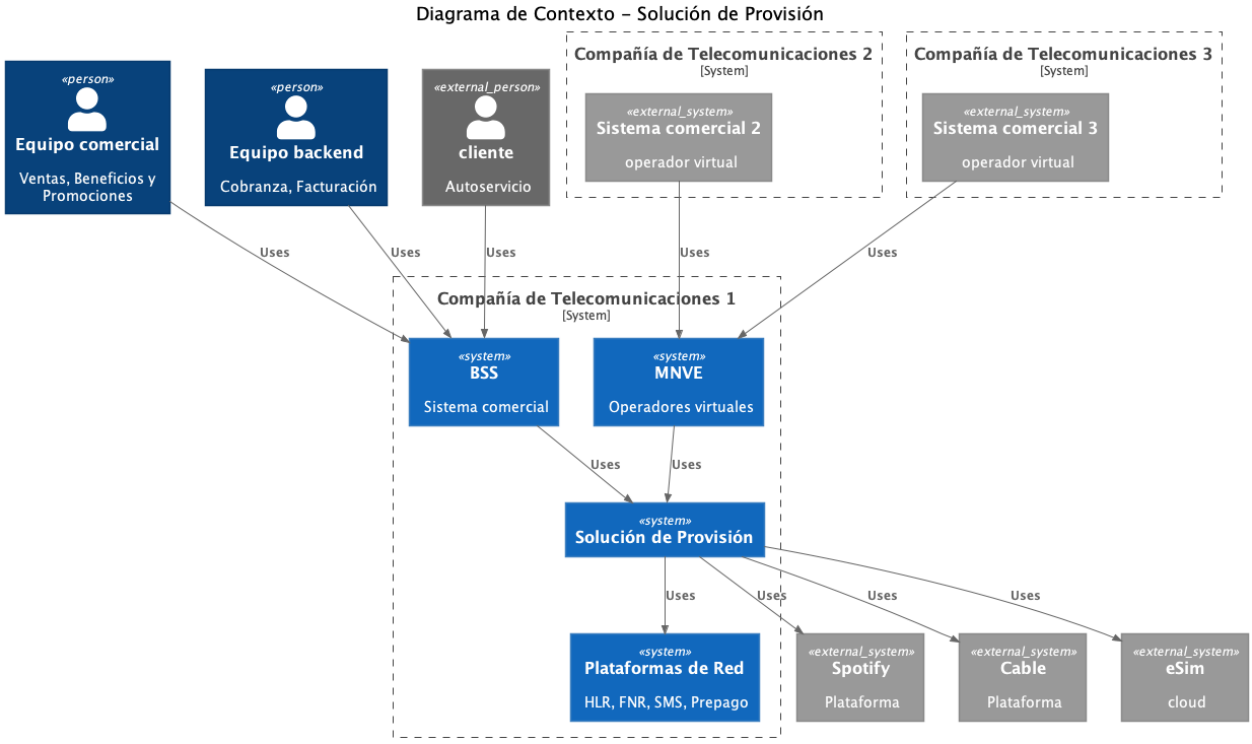


Figura 4.2: Diagrama de Contexto – Solución de Provisión

4.3.4. Diagrama de Contenedores

En la Figura 4.3 se muestra la solución de Provisión, que está compuesta por los componentes: módulo de negocio, módulo de red y sus respectivas bases de datos; se añadirá la solución de Monitoreo como un nuevo módulo.

El gráfico presenta interacciones con los diferentes componentes, se aprecia que el módulo de Negocio es el que recibe las transacciones de negocio del BSS y MNVE, y este módulo orquesta las transacciones de red (movimientos) que envía al módulo de red, este último envía movimientos a las diferentes plataformas (cable, eSim, Plataformas de red, etc). Adicionalmente ambos módulos escribirán en archivos (Repositorio de Archivos), de esta manera el

nuevo módulo, la solución de monitoreo, tomará datos desde las bases de datos de ambos módulos (Negocio y Red) y desde los archivos almacenados en el Repositorio de archivos.

Con los datos que recoge el módulo de monitoreo se generará información que será presentada al usuario mediante gráficos, y también analizará estos datos para activar alarmas en caso ocurra una anomalía.

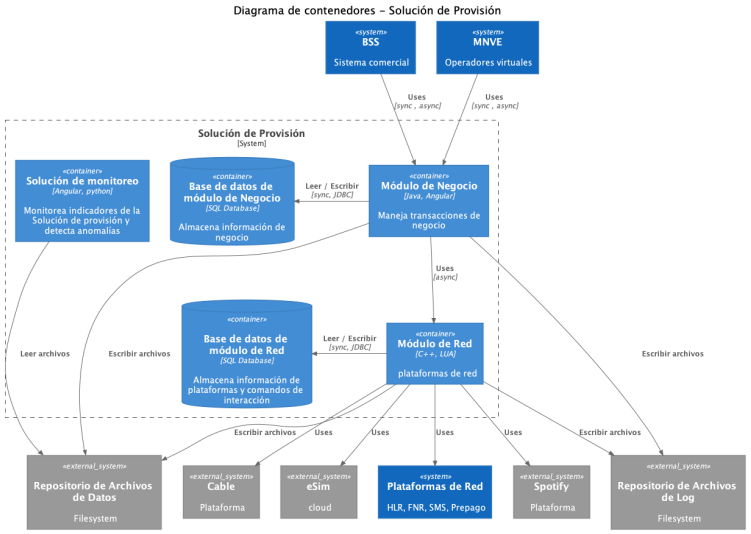


Figura 4.3: Diagrama de Contenedores – Solución de Provisión

4.3.5. Diagrama de componentes (Solución de Monitoreo)

La solución de Monitoreo realizará tareas de extracción de datos, generación de indicadores, presentación de información, y paso a histórico

Adicionalmente, existirá un componente para la detección temprana de anomalías. Dos de los indicadores que se generan son Duración promedio y Cantidad de transacciones de negocio, serán utilizadas por el algoritmo de detección temprana. El componente de alarmas será activada si se detecta una anomalía. Entonces la solución de Monitoreo estará compuesto por las siguientes componentes:

1. Extracción de datos
2. Generación de indicadores
3. Detección temprana de anomalías
4. Notificación (Generación de alarmas)
5. Presentación de información (página web)
6. Registro de información
7. Paso Histórico

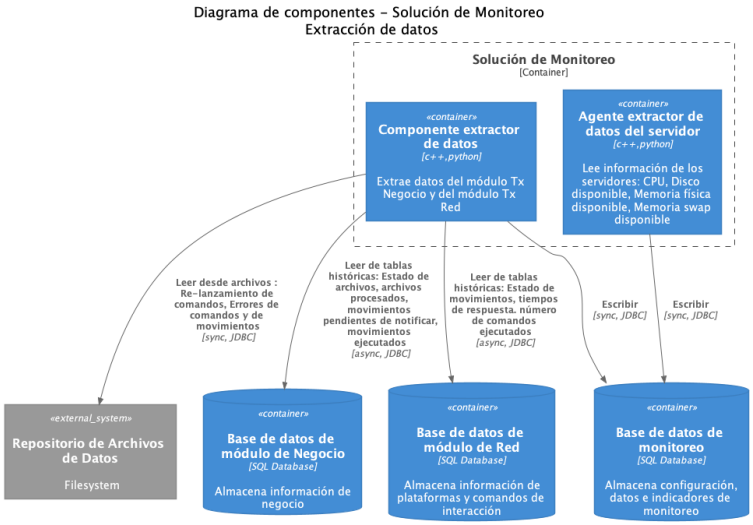


Figura 4.4: Diagrama de Componentes – Extracción

En la Figura 4.4 se muestra las dos formas de extracción de datos:

1. Componente extractor de datos: Recoge datos consultando las tablas de registro de la base de datos de los módulos Tx Negocio y módulo Tx Red. Se consulta tablas de registro para minimizar impacto de performance de la Provisión que trabaja con las tablas operativas.
2. Agente extractor: Recoge información de cada servidor donde se encuentra instalado

Ambos componentes registran datos extraídos en las tablas de Datos de monitoreo.

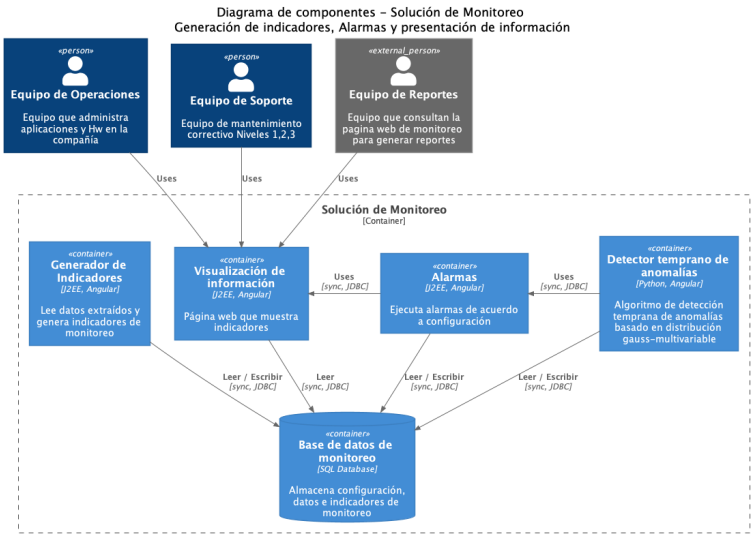


Figura 4.5: Diagrama de Componentes – Análisis y presentación de información

En la Figura 4.5 se muestran los componentes de análisis y presentación de información:

1. Generación de indicadores: Consulta información de las tablas de datos de monitoreo,

procesa los mismos de acuerdo a parametrización (duración de intervalo, etc) y registra información en las tablas de indicadores.

2. Visualización de información: Consulta información de los indicadores y muestra en la solución web, también indica si algún valor esta fuera de rango tolerable (previamente configurado) y en este caso informa al componente de alarmas.

3. Alarmas: Este componente recibe información de anomalías y gestiona las alarmas a la lista de distribución y de acuerdo a los canales definidos (indicador visual en solución web, email, etc.)

4. Detección temprana de anomalías: Consulta la información de los indicadores y procesa con el algoritmo de detección temprana de anomalías, de manera anticipar la detección de un problema

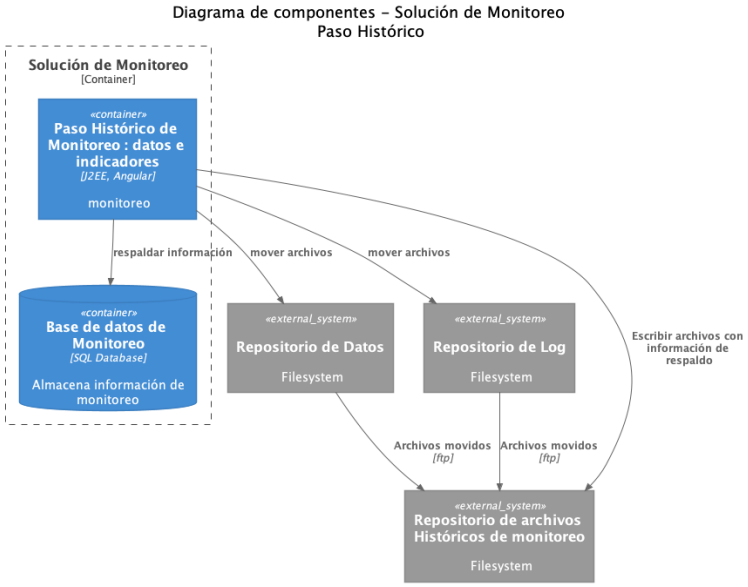


Figura 4.6: Diagrama de Componentes – Paso Histórico

En la Figura 4.6 se muestran los componentes de paso histórico, este proceso recopila información que ya no se utiliza en la operación y se migra hacia la base de datos histórica.

4.3.6. Diagrama de despliegue

En la Figura 4.7 se observa la arquitectura actual del Provisionador y el despliegue del módulo de Monitoreo; se considera que los servidores virtuales de Negocio pueden ser clonados si se necesita mayor capacidad de procesamiento, mientras que la base de datos de Negocio es única para todos los servidores virtuales que existan del Negocio, la base de datos trabaja en clúster. De similar manera para el bloque de servidores virtuales del módulo de Red y su respectiva base de datos. Para el caso del Módulo de Monitoreo y su base de datos, ambos serán únicos, trabajando en servidores virtuales independientes para afectar lo menos posible la operación de la provisión.

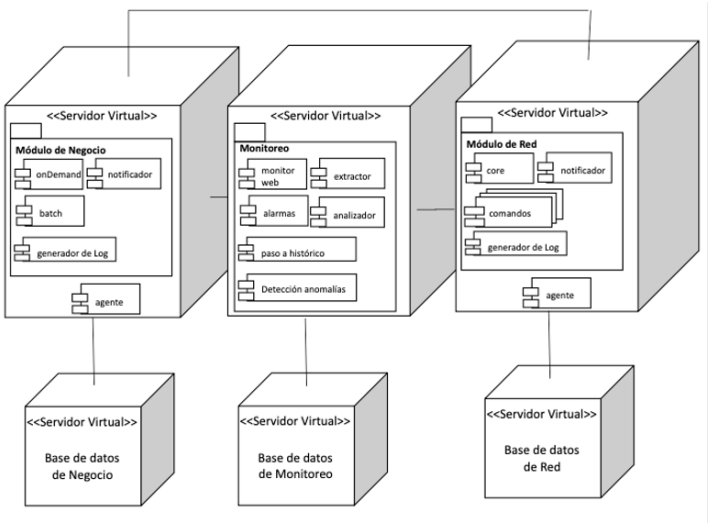


Figura 4.7: Diagrama de despliegue del monitoreo y la provisión

4.3.7. Diseño de base de datos

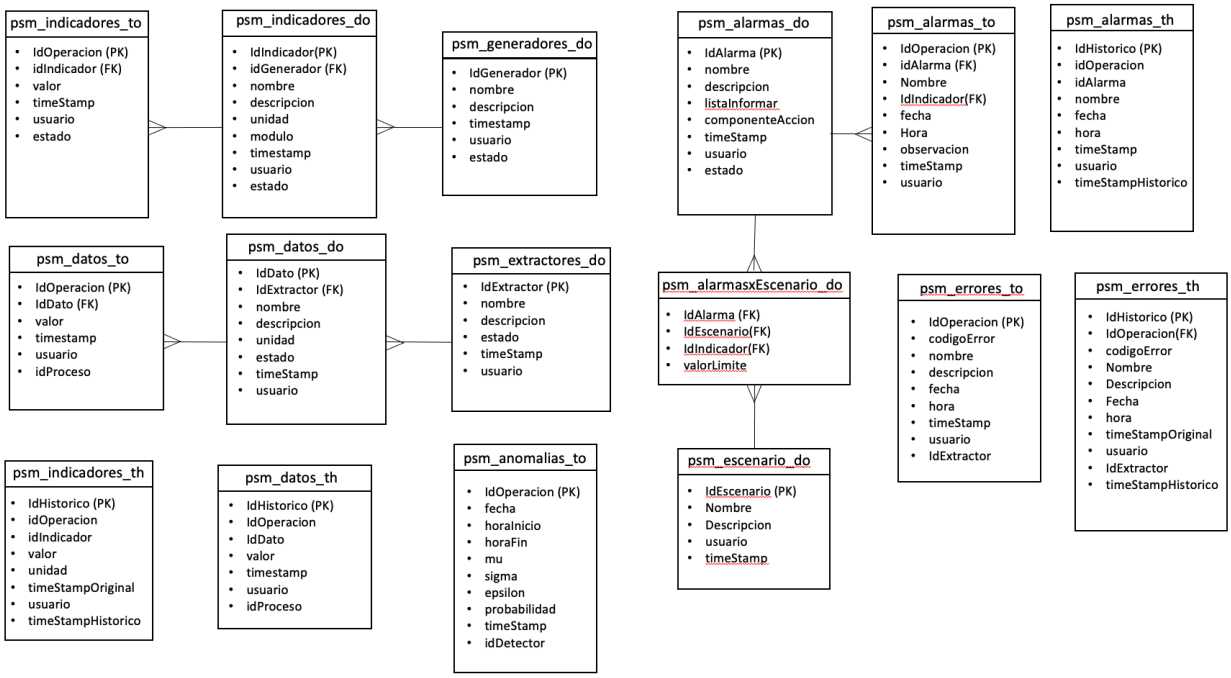


Figura 4.8: Diseño de base de datos de monitoreo

Observar que las tablas finalizadas en "_do" son de dominio o configuración, las tablas que terminan en " _to"· son operativas y la tablas que finalizan en "_th" son históricas.

El módulo de monitoreo contará con 16 tablas:

1. Tabla de datos, (psm_datos_do), contiene la descripciones de los datos que se extraerán: porcentaje de ocupación de cpu, duración de transacción, etc; se indica la unidad del dato.

2. Tabla de datos, (psm_datos_to), contienen los datos extraídos, tanto de la provisión como de los servidores donde están instalados estos procesos.

3. Tabla de grupos de indicadores (psm_grupos_do), contiene la configuración de los grupos de indicadores, está pensado en facilitar la administración, para esto maneja estados (activo, desactivado), para activar o desactivar la ejecución del proceso de generación de los indicadores que contiene el grupo. El componente de Generación de indicadores está conformado por varios procesos/ejecutables, y cada ejecutable generará un grupo de indicadores. De esta manera será más fácil añadir indicadores, así como las aplicaciones que generan dichos indicadores.

4. Tabla de configuración de indicadores (psm_indicadores_do), contiene la descripción del indicador, el grupo al que pertenece, descripción de la unidad en que se calculó la información, estado (activo, desactivado), módulo (General, Negocio, Red),que es otra agrupación, para tener la referencia a la función que se está midiendo, y no solo al proceso que lo genera, este campo se agregó para conservar el origen de la función que se mide, pensando en complementar el grupo, que está más referenciado al proceso que genera el indicador, que crecerá por temas prácticos, es decir se considera que aparecerán en el futuro muchos procesos generadores de indicadores, por servidor e incluso por nuevos tipos de indicadores como las tendencias.

5. Tabla de indicadores (psm_indicadores_to) donde los procesos del componente Generador de indicadores almacenarán los valores de los indicadores que generan. Contiene ID del valor(secuencial),id del indicador(de la tabla de configuración de indicadores), id del proceso/aplicación que lo genera, se contempla que existan varios procesos de generación de indicadores por especialización de tarea:

   (a) Generación de indicadores de servidor

   (b) Generación de indicadores de operación negocio batch

   (c) Generación de indicadores de operación negocio ondemand

   (d) Generación de indicadores de operación negocio de red

6. Tabla de datos históricos de los indicadores,(psm_indicadores_th) donde se almacenan los indicadores del mes. Considerar que la data operativa es la del día, la data histórica es la del mes, y finalmente la data con antigüedad mayor a un mes se almacena en archivos para ser mantenidas de acuerdo a la política del Compañía.

7. Tabla de registro de anomalías (psm_anomalias_to), cada vez que el algoritmo de detección temprana de anomalías detecta un evento anómalo, lo inscribe en esta tabla. Se registra el proceso de Detección que encuentra el evento en el campo idDetector, porque se considera que habrán varios procesos que detectan anomalías.

8. Tabla de configuración de escenarios (psm_escenario_do), donde se configuran los escenarios de evaluación de los indicadores, originalmente se consideran los siguientes:

   (a) Horario comercial, de Lunes a viernes de 10:00 a 19:00

   (b) Horario no comercial, de Lunes a viernes de 00:00 a 10:00 y 19:00 a 00:00

   (c) Fin de semana, sábado y domingo

9. Tabla de relación escenario-indicador-alarma (psm_alarmasXEscenario_do) relaciona las alarmas que deben gatillarse cuando el valor del indicador excede el límite (valorLimite) para una escenario.

10. Tabla de configuración de las alarmas (psm_alarmas_do), donde se configuran acciones posibles de seleccionar como alarma. Inicialmente se considera mostrar alarma en aplicación de monitoreo. Se activa una campana; el modelamiento de tablas está diseñado para permitir escalar el número de procesos de alarmas que son posibles de gatillar.

11. Tabla de registro de las alarmas gatilladas (psm_alarmas_to), se almacenan sólo las alarmas levantadas del día. Esto por performance.

12. Tabla histórico de alarmas (psm_alarmas_th), aquí se trasladan las alarmas gatilladas en el mes, luego se pasa a archivos.

13. En la tabla de errores (psm_errores_to) se registran los errores que se registran en el log, la razón es poder realizar un análisis de correlación de errores con anomalías detectadas de manera de en el futuro construir otro algoritmo para detectar anomalías de manera temprana basado en errores que el sistema detecta. Aquí se almacenan solo información de un día.

14. En la tabla de errores histórica (psm_errores_th) se registran los errores históricos del mes, luego de este tiempo se trasladará esta información a archivos.

4.4. Prototipo del Monitoreo

A continuación se muestra y explica el diseño preliminar de la interfaz web del módulo de monitoreo, se excluye la configuración y se muestran las pantallas que se utilizarán operativamente. si bien, para esta esta etapa se considera solamente una implementación en plataforma web, se evalúa seriamente que a futuro esta aplicación se implemente como aplicación para celulares plataformas de teléfonos inteligentes dada la criticidad en la recepción de la información que genera, en especial lo que se refiere a las alarmas que gatilla. El diseño para mostrar los indicadores va de lo general a lo particular, mostrando en la primera página página un resumen de los indicadores del sistema, de manera que en una mirada se pueda tener una idea general de la salud de éste (Solución de la Provisión); el detalle se puede revisar en las páginas que se alcanzan a partir de esta.

El portal de monitoreo será utilizado por usuarios técnicos y por jefaturas, se piensa consultar como herramienta web. En la figura 4.9 se muestran los datos solicitados para ingresar al sistema.

En la figura 4.10 se muestra la pantalla principal en primer plano, y otras pestañas a las cuales se puede acceder para tener mayor detalle. El detalle que se puede revisar son los dos submódulos de Provisión: Negocio y Red. Se ha dividido la parte de negocio en Batch y On-demand, debido a que la entrada de datos a la solución de Provisión es a través de dos componentes distintos del Submódulo de Negocio, cada uno de estos componentes atienden comportamientos distintos: Negocio On-demand que interactúa de manera síncrona mediante un webservices y el componente Negocio Batch que recibe archivos, trabaja de manera masiva

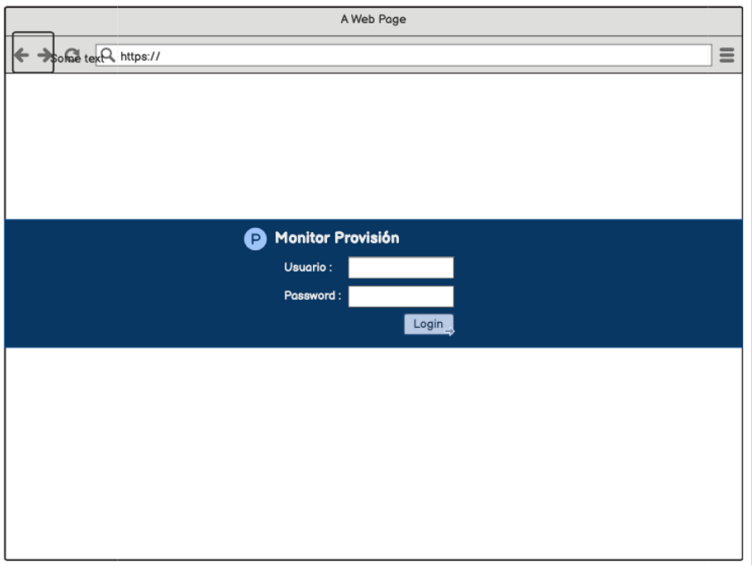


Figura 4.9: Pantalla de autenticación del monitoreo

y asíncrona; éstos son componentes independientes. Adicionalmente se cuenta con la salida en el componente de Red y por último una pestaña de Errores que recoge el sistema, para facilitar el análisis.

La página principal muestra en la parte superior 3 grupos de indicadores de sistema, correspondiente a los servidores con los cuales está trabajando la solución de Provisión, pueden trabajar varios servidores del componente on-Demand, varios servidores del componente Negocio Batch y varios servidores del componentes Red. Cada grupo de indicadores (CPU, ocupación de Memoria, ocupación de Disco) corresponde a un componente, en la figura 4.10 se aprecia que hay 2 servidores Negocio Batch, 2 servidores on-Demand y 3 servidores de Red.

En la parte inferior/media se observan 3 grupos de pestañas, cada grupo corresponde a cada componente. En el caso de negocio (on-Demand y Batch) cada pestaña muestra indicadores y gráficos por cada aplicación cliente que interactúa con la provisión; recordemos que varias aplicaciones como BSS, MNVE pueden enviar transacciones de negocio de manera paralela (ver Figura 1.2), en la figura se aprecia que hay dos clientes para cada componente de Negocio (on-Demand y Batch). En el caso de Red, las pestañas muestran los servidores que están trabajando con el componente de Red. A nivel operativo se suelen utilizar varios servidores agrupando las plataformas que se impactan. Es común agrupar plataformas con similar comportamiento, se suelen agrupar plataformas lentas(como plataformas de valor agregado como spotify), plataformas prioritarias (como el HLR que activa la línea telefónica) y plataformas que responden con velocidad media.

Como se dijo al comienzo de esta sección, primero se esta pensando en que exista una versión web, pero se considerará también una opción web-celular en especial de esta página.

Puede observarse una campana amarilla en lado superior derecho, ésta es la advertencia que hace el sistema si hay algún tema que se debe revisar.

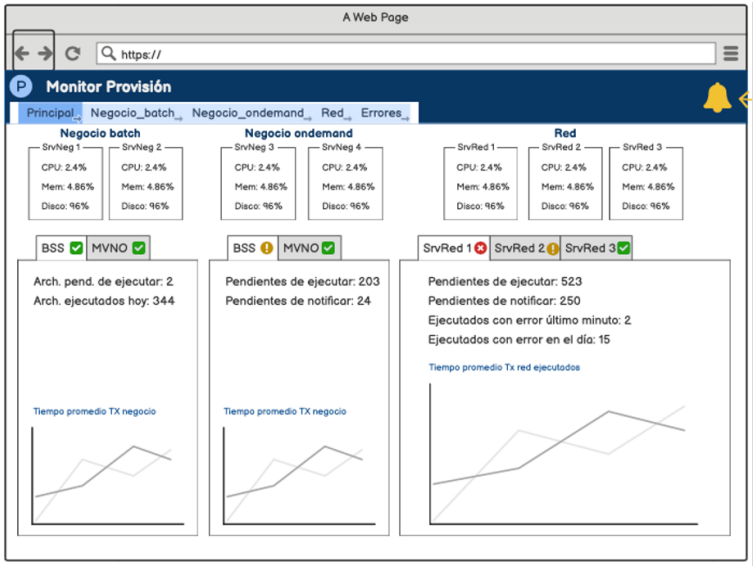


Figura 4.10: Pantalla Principal

En la figura 4.11, se observa la pantalla del componente de Negocio batch, este componente es asíncrono y masivo; a través del mismo el Provisionador puede recibir archivos con las transacciones que debe ejecutar. Por ejemplo las acciones de suspensión de servicio por no pago se envían de manera masiva.

En la parte superior se muestran indicadores del sistema: los servidores que brindan la funcionalidad de procesamiento de negocio batch; mostrando CPU, memoria y ocupación de disco. En la parte inferior se muestran pestañas, una por cada aplicación cliente configurado que envía archivos a la Provisión. En cada pestaña se muestran tres gráficos:

1. Tiempo promedio Transacciones de negocio.
2. Cantidad de archivos recepcionados.
3. Cantidad de archivos procesados.

Estos gráficos (indicadores) toman el período de evaluación configurado (que partirá en 5 minutos),así por ejemplo el tiempo promedio de transacciones de negocio se calcula tomando muestras en el intervalo (5 minutos) y calculando el promedio de duración de las transacciones.

En caso ocurriera un problema, se muestra si el problema está antes del procesamiento(recepción de archivos), durante el procesamiento, haciendo diferencia entre el procesamiento de las transacciones y la completitud del archivo. Esto último se debe a que el proceso batch lee el archivo por bloques y envía a procesar las transacciones leídas, pudiendo haber competencia con otras aplicaciones cliente que están utilizando el negocio batch, con lo cual es información relevante ver si el procesamiento individual de las transacciones esta teniendo alguna anomalía, o si hay demora para completar todas las transacciones del archivo.

En la figura 4.12 se puede ver la pantalla del componente de Negocio on-Demand, este componente es síncrono y puntual. Un ejemplo de transacciones que se gestionan por este

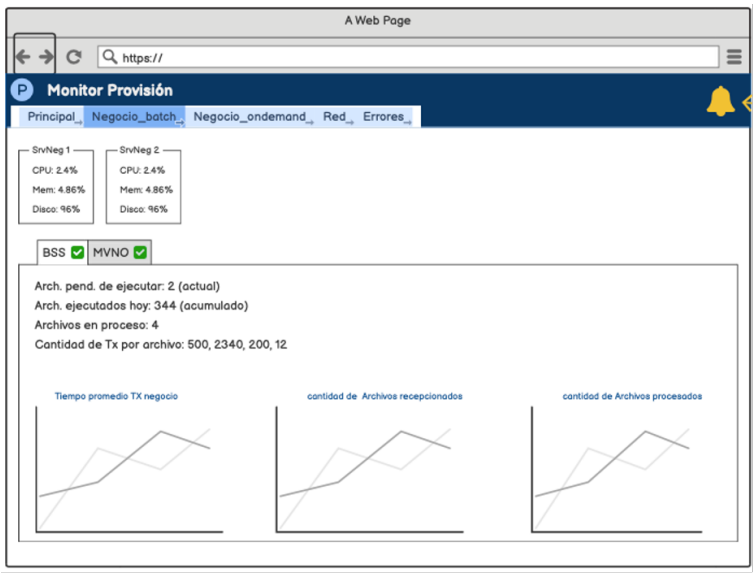


Figura 4.11: Pantalla procesamiento de Negocio batch

módulo son por ejemplo las ventas en oficina, que son transacciones puntuales, el ejecutivo usa una aplicación cliente que espera respuesta de que se ha realizado la activación del celular de manera correcta, para indicarle al cliente que puede salir de la tienda con el dispositivo activo.

Se muestra en la parte superior indicadores del sistema: los servidores que brindan la funcionalidad de procesamiento de negocio batch; mostrando CPU, memoria y ocupación de disco. En la parte inferior se muestran pestañas, una por cada aplicación cliente configurado que se comunica con la Provisión de manera online, en cada pestaña se muestran tres gráficos:

1. Tiempo promedio Tx Negocio
2. Cantidad de archivos recepcionados
3. Cantidad de archivos procesados

Los datos son procesados sobre muestras en el periodo de tiempo configurado (5 minutos).

En la figura 4.13 se presenta la pantalla del módulo de red, donde se muestra la actividad de la Provisión con las plataformas. Muchos de los atrasos en procesar las transacciones que llegan a la Provisión obedecen a la lentitud de alguna de los interacciones con alguna plataforma, por lo que esta vista está diseñada para ser utilizada por el personal técnico cuando analizan plataforma que empiezan a degradarse y se quiere revisar tendencias de indicadores, de esta manera se puede empezar a revisar comportamiento en la plataforma que empieza a degradarse.

La aplicaciones cliente que alimentan al módulo de red son las aplicaciones del módulos de negocio(batch y online). El módulo de red estará instalado en varios servidores, en el ejemplo se muestran tres servidores, esta configuración es muy común, ya que se dividen las plataformas, en las de alta demanda, plataformas lentas y plataformas de respuesta "normal".

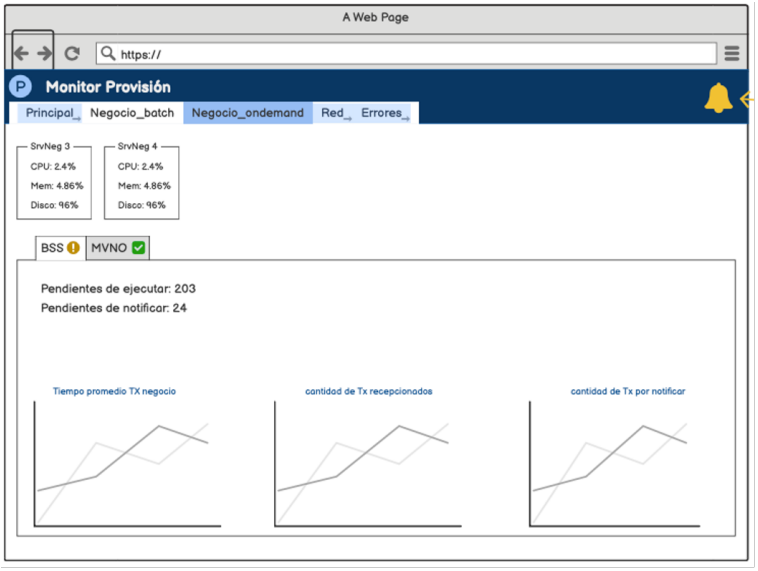


Figura 4.12: Pantalla procesamiento de Negocio on-Demand

Se muestra en la parte superior indicadores del sistema: los servidores que brindan la funcionalidad de procesamiento de negocio batch; mostrando CPU, memoria y ocupación de disco. En la parte inferior se muestran pestañas por cada servidor donde se instaló la aplicación; se muestran el gráfico de Tiempo promedio de transacciones ejecutadas, para mostrar tendencias. Además se muestran los siguientes valores:

1. Pendientes de ejecutar.
2. Pendientes de notificar.
3. Ejecutados con error último minuto (recientemente)
4. Ejecutados con error en el día

Las transacciones pendientes de ejecutar son las transacciones de red encolados; respecto a las transacciones pendientes de notificar, existe un proceso auxiliar, luego que la plataforma responde, se ocupa de enviar respuesta (notificar) al servidor de negocio(batch u online) que inició la transacción.

Hay ocasiones en que la plataforma indica algún error, este es informado y registrado para que el Analista realice la corrección, los errores pueden ocurrir por valores enviados, conexión, etc.

Adicionalmente en la misma pestaña del servidor se muestra un conjunto de pestañas ordenados verticalmente con las plataformas con las que interactúa el servidor, en estas nuevas pestañas se muestra la misma lista de indicadores y el gráfico de "Tiempo promedio de transacciones ejecutadas"pero estos están individualizados para la plataforma que se selecciona.

En la figura 4.14 se presenta la pantalla que muestra los errores que actualmente detecta el provisionador y los escribe en el log, uno de los objetivos del monitoreo es almacenar estos

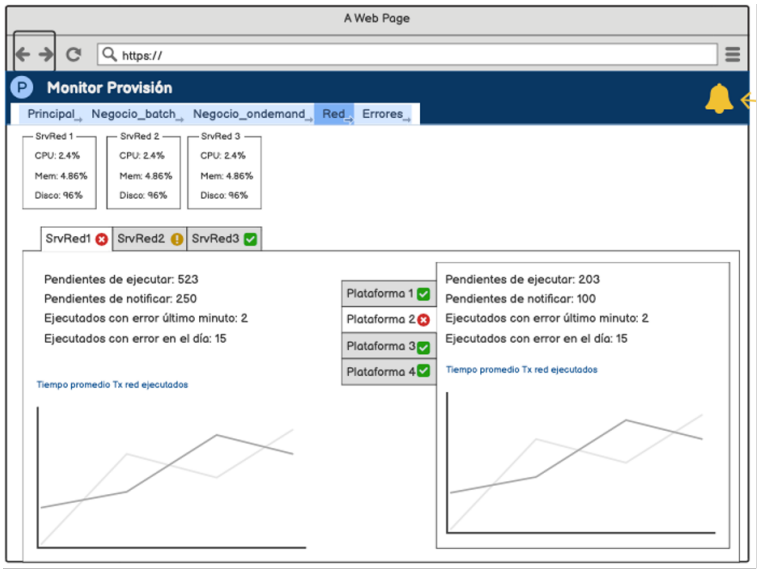


Figura 4.13: Pantalla procesamiento del módulo de Red

errores, e incrementar la lista de errores que se pueden detectar para en el futuro poder contar con datos clasificados por error para poder utilizarlos en reconocer anticipadamente algún error.

En la parte superior se agrupan los errores por clientes que invocan la Provisión y en la parte inferior se presentan los errores por plataforma con la que la solución de Provisión interactúa.

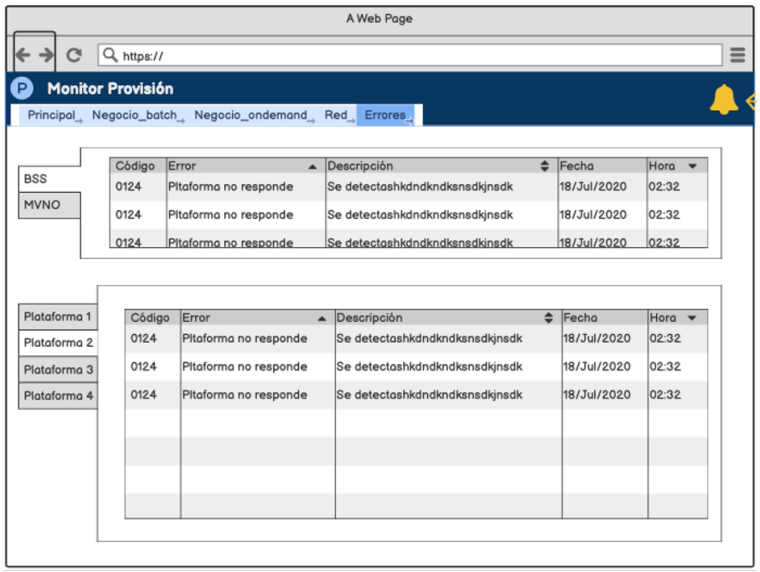


Figura 4.14: Pantalla procesamiento del módulo de Errores

Capítulo 5

Pruebas de efectividad del algoritmo de detección temprana de anomalías

En este capítulo se quiere probar la efectividad del algoritmo propuesto para detectar anomalías tempranamente. Para ello se implementó una aplicación prototipo con el algoritmo de detección temprana de anomalías. Este programa se probó con el "set de valores reales" usados en los casos del análisis previo y y se revisó en que medida el algoritmo de distribución normal es capaz de detectar las anomalías ocurridas. El algoritmo analiza los valores de dos indicadores:

1. Duración: duración promedio de procesar transacciones desde que ingresan a la Provisión hasta que responde
2. Cantidad: cantidad de transacciones promedio, que se realizan en un lapso de tiempo de 5 minutos.

Se cuenta con cuatro sets de datos reales del Análisis Previo(uno por cada caso estudiado), cada set ha sido calificado agrupando las transacciones en intervalos de 5 minutos, para cada intervalo se calcula la "Duración" promedio y la "Cantidad" de transacciones. Luego se clasifican como anomalías aquellos intervalos que tienen Duración mayor a 30 segundos o Cantidad mayor a 3.000 transacciones cuando es Horario Comercial y se clasifican como anomalías aquellos intervalos que tienen Duración mayor a 10 segundos o Cantidad mayor a 1.000 transacciones cuando es Horario No comercial. Entonces se cuenta con cuatro "set de valores reales" con intervalos calificados como "Anómalo" o "No Anómalo".

El algoritmo utiliza 3 parámetros: media, desviación estándar y epsilon que son encontrados ha partir del "set de valores de entrenamiento" .

Con los parámetros hallados para Horario Comercial y No comercial, se inician las pruebas del prototipo para cada caso y se calcularon las siguientes métricas para revisar que tan bien o mal clasifica los valores:

1. Accurancy (Acertado) = indica la proporción de cuantas anomalías y NO anomalías acertó el algoritmo sobre el total de datos.

2. Precision (Precisión) = indica la proporción de cuantos de los valores predichos como anómalos fueron realmente anómalos.
3. Recall (Exactitud) = indica la proporción de cuantos de los anómalos reales fue capaz de predecir el algoritmo
4. F1-score es una métrica que permite evaluar el algoritmo considerando simultáneamente Precision y Recall, cuanto más cercano a 1(uno) mejor es el algoritmo.

Las métricas Accurancy, Precision, Recall, F1 score son calculados a partir de los indicadores:

1. Positivo verdadero: Realmente es una anomalía y el algoritmo lo clasificó correctamente como una anomalía.
2. Positivo falso: Realmente es NO anómalo, pero el algoritmo lo clasificó incorrectamente como una anomalía.
3. Negativo verdadero: Realmente es NO anómalo y el algoritmo lo clasificó correctamente como NO anómalo.
4. Negativo falso: Realmente es una anomalía, pero el algoritmo lo clasificó incorrectamente como NO anómalo.

5.1. Datos de entrenamiento

El algoritmo utiliza 3 parámetros: media, desviación estándar y epsilon (explicados en item 4.2 Algoritmo de detección temprana de anomalías)

Para encontrar los valores de los parámetros se tomó un set de datos de entrenamiento, tomando valores del caso 1 debido a que fue un día ese observó un bajo número de anomalías, se tomó el 60 % de los datos como lo sugiere el artículo [1] "A Gaussian Approach to the Detection of Anomalous Behavior in Server Computers" y el artículo [10] "Material de curso-online de Coursera Machine Learning –Stanford University".Con este proceso se buscando mejorar la asertividad del algoritmo.

Se calcularon los valores de los parámetros de la distribución normal (media, desviación estándar) a partir de un set de datos de entrenamiento con datos No anómalos considerando horarios comercial y No comercial. En este proceso se consideró lo siguiente:

1. Sólo se utilizan datos de casos no anómalo
2. Se tomaron muestras cada 5 minutos
3. Se hizo diferenciación por horarios No comercial (00:00 a 10:00 y 19:00 a 23:59) y Comercial (10:00 a 19:00)

4. Se consideran datos no anómalos en horario no comercial aquellos con duración máxima de 10 segundos y cantidad máxima de 1.000 transacciones
5. Se consideran datos no anómalos en horario comercial aquellos con duración máxima de 30 segundos y cantidad máxima de 3.000 transacciones

Epsilon se fijó en 0,0000001 como valor inicial y se pensó en ir cambiando dependiendo de si los resultados salían muy alejados de la realidad, pero finalmente quedó este valor como definitivo para la prueba.

Los parámetros encontrados fueron:

Tabla 5.1: Parámetros calculados de distribución normal

Horario	**Parámetro**	**Indicador Duración (seg)**	**Indicador Cantidad (und)**
No comercial	media	0,52	162,23
	desviación estándar	0,78	203,33
Comercial	media	0,61	1.325,13
	desviación estándar	2,05	447,73

En la tabla 5.1 se verifica que la media es 0,52 segundos para el indicador Duración (duración promedio de transacciones) en horario No comercial y su desviación estándar es 0,78; mientras que la media de 162,23 transacciones es del indicador Cantidad (cantidad de transacciones) en el mismo horario No comercial, y su desviación estándar es de 203,33.

Respecto al Horario comercial se encontró que la media es 0,61 segundos para el indicador Duración (duración promedio de transacciones) y su desviación estándar es 2,05; mientras que la media del indicador Cantidad (cantidad de transacciones) es 1.325,13 transacciones en el mismo horario Comercial, y su desviación estándar es de 447,73.

5.2. Métricas de Horario No comercial (00:00 a 10:00 / 19:00 a 23:59)

En la tabla 5.2 se pueden comparar las métricas para los diferentes casos en horario No comercial, se observa en general valores cercanos a 1.

Recall es la métrica que se prefiere cerca a 1(uno) para este análisis del algoritmo; esto debido a que el algoritmo de detección debe encontrar la mayor cantidad de anomalías reales,

Tabla 5.2: Métricas con datos de horario No comercial

Métrica	**caso 1 15-Julio**	**caso 2 16-Julio**	**caso 3 17-Julio**	**caso 4 03-Agosto**
Accurancy	0,98	0,97	0,95	0,98
Precision	0,76	0,82	0,90	0,78
Recall	1,00	0,95	1,00	1,00
F1 score	0,86	0,88	0,95	0,88

siendo los falsos negativos (anomalías que no lo son realmente) despejados por un Analista, este último utiliza otros indicadores de la herramienta de monitoreo para dicho análisis. Es decir es preferible detectar y que el Analista descarte a que se escape alguna anomalía.

En la tabla 5.3 se puede revisar el detalle al comparar las anomalías reales de la muestra y las anomalías detectadas por el algoritmo, considerando que el "Total de transacciones" son las transacciones analizadas dentro del horario No comercial, estas son procesadas por el algoritmo en 178 intervalos de 5 minutos, excepto en el caso 3 cuando ocurrió una detención del sistema.

Tabla 5.3: Detalle de cálculos en horario No comercial

Métrica	**caso 1 15-Julio**	**caso 2 16-Julio**	**caso 3 17-Julio**	**caso 4 03-Agosto**
Anomalía reales	13	19	61	14
Anomalías detectadas	17	22	68	18
Intervalos de 5min.	178	178	154	178
Transacciones	137.508	200.916	90.739	65.212
Positivo verdadero	13	18	61	14
Positivo falso	4	4	7	4
Negativo verdadero	161	155	86	160
Negativo falso	0	1	0	0

5.3. Horario comercial (10:00 a 19:00)

Tabla 5.4: Métricas con datos de horario comercial

Métrica	**caso 1 15-Julio**	**caso 2 16-Julio**	**caso 3 17-Julio**	**caso 4 03-Agosto**
Accurancy	0,97	0,99	0,85	0,83
Precision	0,92	1,00	0,85	0,70
Recall	0,96	0,97	0,97	1,00
F1 score	0,94	0,99	0,91	0,83

En la tabla 5.4 se observa que los valores también tienden a 1(uno) y que se logran mejores valores para la métrica Recall en horario comercial (entre 0,96 y 1) comparándolo con el horario No comercial (entre 0,86 y 1).

En la tabla 5.5 se puede revisar el detalle al comparar las anomalías reales de la muestra (24 horas) y las anomalías detectadas por el algoritmo, considerando que el "Total de transacciones" son las transacciones analizadas dentro del horario No comercial, éstas son agrupadas y procesadas por el algoritmo en 108 "intervalos" de 5 minutos, excepto en el caso 3 cuando ocurrió una detención del sistema.

Tabla 5.5: Detalle de indicadores en horario No comercial

Indicador	**caso 1 15-Julio**	**caso 2 16-Julio**	**caso 3 17-Julio**	**caso 4 03-Agosto**
Anomalía reales	24	35	78	43
Anomalías detectadas	25	34	89	61
Intervalos de 5min.	108	108	101	108
Transacciones	374.385	300.035	47.534	279.829
Positivo verdadero	23	34	76	43
Positivo falso	2	0	13	18
Negativo verdadero	82	73	10	47
Negativo falso	1	1	2	0

5.4. Revisión de primera anomalía del día

Ocurren dos a cuatro reportes de anomalías por año, pero como se muestra en el caso 3 pueden detener el sistema lo cual genera problemas operativos y comerciales grandes. La primera anomalía del día es importante, porque que si el algoritmo lo detecta podrá dar más tiempo al equipo de Operaciones para realizar correcciones al comportamiento o realizar acciones de mitigación que permitan disminuir el impacto; de acuerdo a lo revisado con los especialistas las anomalías suelen ser reflejo de un problema raíz; de no corregirse el sistema continua mostrando anomalías, llegando en extremo a colapsar el sistema.

Tabla 5.6: Comparación de detección de primeras anomalías

Eventos	**caso 1 15-Julio**	**caso 2 16-Julio**	**caso 3 17-Julio**	**caso 4 03-Agosto**
Reporte de incidencia	-	-	15:59:00	17:59:00
Primera anomalía real	02:55:00	02:50:00	02:25:00	03:45:00
Primera anomalía detectada	02:55:00	02:50:00	02:25:00	03:45:00

En la tabla 5.6 se observa valores para los 4 casos del análisis previo, se muestra:

1. Reporte de incidencia, es la hora en que el usuario reporta un problema con el sistema (incidencia) pudiendo ser muy variado: lentitud con el sistema, que la aplicación del Ejecutivo no recibe respuesta, etc.
2. Primera anomalía real, es la hora en que se detectó la primera anomalía cuando se realizó el análisis previo.
3. Primera anomalía detectada, es la primera anomalía que detectó la implementación prototipo del algoritmo de detección temprana.

Se observa que el algoritmo fue capaz de detectar la primera anomalía del día en cada caso, recordemos que esta anomalía fue encontrada en el análisis previo y es anterior a la hora en que se reporta la incidencia. En los casos 3 y 4 habría permitido empezar el análisis con 13 y 14 horas de anticipación, comparándolo con la hora en que se reportó la incidencia por parte del usuario, pudiendo realizar acciones paliativas para que el sistema continúe operando correctamente y no se interrumpa; esos días si hubo interrupción del servicio.

5.5. Conclusiones de las pruebas de efectividad del algoritmo

Se han realizado pruebas simulando datos del sistema, utilizando el prototipo del algoritmo de detección temprana para procesar, leyendo las transacciones en intervalos de 5 minutos, el prototipo detectó anomalías procesando con la distribución normal para dos variables: Duración y cantidad de transacciones.

Al analizar sus resultados se llega a las siguientes conclusiones:

1. El algoritmo con distribución normal detecta anomalías correctamente para los cuatro casos del análisis previo, con métricas de Recall entre 0,96 y 1,00; F1-Score varia entre 0,83 y 0,99; analizando dos indicadores: Duración y Cantidad. Es decir en ciertos casos se detectaron todas las anomalías y para los otros casos se detectaron casi la totalidad de las anomalías; adicionalmente los valores que el prototipo predijo como anómalos lo fueron, es decir no predijo falsos anómalos en la mayoría de casos y predijo muy pocos en otros casos; con lo cual tuvo alta efectividad.
2. Se detecta correctamente las primeras anomalías reales del día; con una anticipación de hasta 14 horas, es decir el algoritmo automático detecta anomalías antes que lo reporte una persona.
3. Funcionó la implementación del algoritmo para permitir evaluar dos indicadores de manera relacionada, adicionalmente es escalable, es decir permitirá relacionar nuevos indicadores. La implementación se realizó tomando individualmente los parámetros de cada indicador y relacionándolos en la decisión, el resultado fue que ha detectado correctamente la anomalía.

4. En las pruebas se verifica que será necesario agregar un indicador de cantidad de transacciones que se procesan en paralelo, y aún están pendiente de finalizar; esto debido a que las transacciones con duración mayor al intervalo de muestra (5 minutos) no son considerados en la evaluación.

Respecto al punto 4, los dos indicadores elegidos (Cantidad y Duración) fueron suficiente para detectar anomalías en los 4 casos, pero este tercer indicador será necesario si todas las transacciones duran repentinamente más de 5 minutos, como por ejemplo en un desconexión de la base de datos. Para entender mejor a continuación explico con más detalle: las transacciones tienen duración aproximada de un segundo, siendo considerado anómalo después de los 30 segundos, con lo cual una transacción que dura minutos cae en la categoría de anomalía, la degradación del sistema en los cuatro casos han sido paulatina, es decir las transacciones duraban cada vez más hasta que el sistema colapsó, con lo cual el algoritmo ha sido capaz de detectar la anomalía. Pero si hubiese un corte abrupto de actividad, por ejemplo un corte de conexión a la base de datos, entonces las transacciones estarán no finalizadas y no habrá nuevas transacciones, con lo cual no habrían transacciones para calcular el indicador de Duración ni el de Cantidad.

Cuando se implemente este nuevo indicador dentro del algoritmo: 'cantidad de transacciones procesadas en paralelo', entonces en caso de desconexión de una plataforma o detención de actividad de una aplicación cliente, las transacciones procesadas en paralelo se comenzarán a acumular en este indicador y permitirán al algoritmo detectar la anomalía.

Capítulo 6

Pruebas de usabilidad y utilidad del prototipo

Se realizó un testeo de usabilidad y utilidad por medio de una encuesta, evaluando el prototipo mostrado en el item 4.4 Prototipo del Monitoreo; se presentó a la solución a 2 equipos de Operaciones de dos compañías de Telecomunicaciones y al equipo de Soporte de Amdocs, con el fin de recibir retroalimentación del diseño del monitoreo.

Recordemos que actualmente las incidencias son levantadas cuando se reciben quejas de usuarios finales, ejecutivos vendiendo celulares y servicios de telefonía a clientes, actualmente el análisis de las incidencias es muy dependiente del equipo de Soporte de Amdocs, que debe conectarse desde Chile a la operación de Latinoamérica con problemas para revisar logs y recomendar acciones.

Se aprovecharon las reuniones para revisar el protocolo de atención de servicio de Mantenimiento Correctivo (Soporte) y además la distribución de responsabilidades entre el equipo de la Operadora y de Amdocs, adicionalmente se recibió feedback de temas relacionados al servicio, esto último no se incluyen en este documento porque la evaluación es solo de la solución de monitoreo.

6.1. Factores a evaluar

Se elige un enfoque de evaluación formativo, buscando verificar si se están alcanzando las metas (factores) definidas para la herramienta de monitoreo y así verificar la hipótesis que se evalúa.

El gran objetivo de la solución de monitoreo es que ayude a detectar incidencias tempranamente, el diseño de la información que se muestra debe estar en sintonía con dicho objetivo, más allá que se implementará la funcionalidad del algoritmo de detección temprano de anomalías, cuyo prototipo es evaluado de manera distinta.

Los factores que ayudan a que el diseño del monitoreo empuje hacia este gran objetivo

son:

1. Facilidad de uso
2. Incremento en independencia de equipos de la Operadora
3. Completitud de indicadores
4. Detección y análisis anticipado de anomalías

Respecto a la facilidad de uso, se busca que el diseño sea percibido como fácil de utilizar para que se utilice realmente, de lo contrario puede ocurrir que los equipos sigan realizando las labores como lo realizan actualmente, con lo cual no se conseguirían los beneficios que se buscan con esta herramienta.

El incremento en la independencia de equipos de la Operadora, permite ganar tiempo ya que se aumenta el análisis en tiempos más tempranos, incluso si fuese el caso que el equipo de la Operadora pueda corregir algún desvío o entregar mayores antecedentes cuando se levante la incidencia al equipo de Amdocs, con lo cual también se gana tiempo.

Respecto a la Completitud de indicadores, es importante tener feedback de que se incluyen los indicadores que utilicen en su labor o que generen manualmente.

Detección y análisis anticipado de anomalías, la herramienta incorpora gráficos para analizar tendencias de comportamiento del sistema, que se espera que el analista revise diariamente ,sin esperar la alarma de una anomalía; también se muestran indicadores que podrán revisarse como parte del protocolo diario, esto es adicional a la alarma(campana) que muestra el monitoreo cuando detecta que algún indicador sobrepasó el umbral permitido. Por último la herramienta de monitoreo tendrá la página de errores para que se pueda consultar rápidamente errores que retornan las plataformas y el sistema, de manera que el análisis de logs sea un paso posterior.

6.2. Formato de la encuesta

En la tabla 6.1 se muestra el formato utilizado en la encuesta realizada, se consideró una escala de 5 valores, siendo 5 el más satisfactorio y 1 el menos satisfactorio.

En la tabla 6.2 se muestra la agrupación de las preguntas por los factores a evaluar, los factores están alineados con el objetivo general de contar con una herramienta de monitoreo que permita detectar antes las anomalías, y adicionalmente que el proceso de análisis sea más rápido.

Tabla 6.1: Formato de encuesta

	1	**2**	**3**	**4**	**5**
¿El monitoreo es fácil de utilizar? ¿La página principal es suficiente para una primera vista? ¿Distribución de pestañas es adecuado?					
¿El monitoreo permitirá acelerar el análisis de incidencias? ¿El equipo de la Operadora podrá realizar un análisis más profundo?					
¿Están todos los indicadores? ¿Son útiles los gráficos para mostrar tendencias?					
¿Se podrá detectar problemas con mayor anticipación? ¿La página de errores se puede utilizar antes que la revisión del log?					

Tabla 6.2: Preguntas de la encuesta por cada factor

Factores	**Preguntas**
Facilidad de uso	¿El monitoreo es fácil de utilizar? ¿La página principal es suficiente para una vista rápida? ¿Distribución de pestañas es adecuado?
Incremento en la independencia de equipos de la Operadora	¿El monitoreo permitirá acelerar el análisis de incidencias? ¿El equipo de la Operadora podrá realizar un análisis más profundo?
Completitud de indicadores	¿Están todos los indicadores? ¿Son útiles los gráficos para mostrar tendencias?
Detección y análisis anticipado de anomalías	¿Se podrá detectar problemas con mayor anticipación? ¿La página de errores se puede utilizar antes que la revisión del log?

6.3. Resultado y Conclusiones

Luego de realizar la encuesta al equipo de Operaciones de dos operadores y al equipo de de Soporte Amdocs, se lograron los promedios agrupados por factor que se aprecian en la tabla 6.3 Preguntas de la encuesta por cada Factor.

De acuerdo a los valores encontrados se verifica que el diseño de la solución de monitoreo es calificado positivamente con los factores que se evaluaron, acercándose al valor más satisfactorio.

Tabla 6.3: Preguntas de la encuesta por cada Factor

Factores	**Promedio**
Facilidad de uso	4,70
Incremento en la independencia de equipos de la Operadora	4,50
Completitud de indicadores	4,67
Detección y análisis anticipado de anomalías	4,72

Capítulo 7

Conclusiones y próximos pasos

Los resultados de las pruebas del algoritmo de detección temprana (dentro del monitoreo) validan la hipótesis de que posible construir una herramienta de monitoreo que detecte las anomalías antes que una persona, en al menos 12 horas, con lo cual el especialista de Provisión contará con más tiempo para realizar actividades de análisis, corrección y mitigación del problema ocasionado. Estas pruebas se presentan en la sección 5 Pruebas de efectividad del algoritmo de detección temprana de anomalías.

Asimismo el diseño presentado para la herramienta de monitoreo, fue validado por usuarios y especialistas en el item 6 Pruebas de usabilidad y utilidad del prototipo, quienes ratificaron que es una herramienta que contiene los indicadores necesarios para acelerar el análisis y para navegar de lo general a lo particular, es decir revisar la salud del Provisionador a partir de la opción global de los indicadores, para luego poder ampliar la información o revisar indicadores focalizados en cada módulo que compone la Provisión.

El diseño arquitectónico consideró no impactar la operación con la captura de datos y el procesamiento de los indicadores, este problema no fue simulado dado que se debe ajustar in-situ con la solución implementada durante la instalación de la herramienta de monitoreo de acuerdo a las características de los servidores y configuración del Provisionador en ambiente Productivo. Para esto se utilizará parámetros que brindan flexibilidad a la herramienta, se deberá probar diferentes configuraciones para lograr el equilibrio entre información y performance entre la herramienta de monitoreo y el sistema de Provisión. Para implementar dicha característica de flexibilidad la herramienta de monitoreo tomará datos desde tablas existentes de registro para no impactar la operación, contará con el apoyo de aplicaciones livianas e independientes en cada servidor que capturarán la información necesaria para generar los indicadores, la frecuencia de captura será configurable, asimismo el intervalo de lectura de datos para el cálculo de indicadores será también parametrizable.

Con esta herramienta de monitoreo el equipo de soporte podrá abordar de mejor manera el reto que significa el cumplimiento de nivel del servicio de soporte(mantenimiento correctivo), en especial cuando se atienden incidencias críticas, ya que podrán ser analizados y mitigados en etapas tempranas, antes que colapse la solución de Provisión, y cuando ocurra una anomalía que provoque una incidencia crítica, esta será detectada tempranamente para que el especialista mitigue la situación y no se afecte la operación.

7.1. Próximos pasos

Se implementará la solución de monitoreo a partir del diseño presentado, incluyendo el algoritmo de distribución normal introduciendo un tercer indicador: cantidad de transacciones procesadas en paralelo. Adicionalmente se utilizarán los parámetros encontrados como valores iniciales de pruebas, y se irá revisando en pruebas de certificación para llegar a valores óptimos de detección cuando se instale la solución en ambiente productivo.

La herramienta de monitoreo está diseñada para registrar anomalías y errores que presentan: la solución de Provisión, las plataformas y también las aplicaciones con las que interactúa; este almacenamiento es con el objetivo de implementar la siguiente versión del algoritmo de detección temprana utilizando deep learning, buscando aumentar el tiempo de 12 horas en la detección de anomalías.

Bibliografía

[1] Navoneel Chakrabarty. *A Gaussian Approach to the Detection of Anomalous Behavior in Server Computers*. Artículo en Medium, Germany, 2019.

[2] Michael Paul DeHaan. *Monitoring software provisioning*. Red Hat Inc, United States, 2016.

[3] Rob Ewaschuk. *Monitoring Distributed Systems*. SRE Google, United States, 2019.

[4] Shahrooz S. Kasrai. *Method and system for automatically verifying provisioning of telecommunications services*. Alcatel USA Sourcing LP, United States, 2001.

[5] Vijay Khurana. *Introducción a Prometheus y Grafana*. Geekflare, 2020.

[6] Michael Roy King. *Data processing system having monitoring of software activity*. Intel Deutschland GmbH, United States, 2019.

[7] Richard Kirchhofer. *Automatic monitoring and just-in-time resource provisioning system*. EMC Corp, United States, 2012.

[8] Dilip KhandekarDragutin PetkovicPratap SubrahmanyamBich Cau Le. *Provisioning of computer systems using virtual machines*. VMware Inc, United States, 2009.

[9] Mustafa Daraghmeh & Qutaibah Althebyan Mahmoud Al-Ayyoub, Yaser Jararweh. *Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure*. Springer Link, United States, 2015.

[10] Andrew Ng. *Material de curso-online de Coursera Machine Learning –Stanford University*. Stanford University, United States, 2019.

[11] Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly, 2017.

[12] Prometheus. *Codeless Provisioning Automation*. Prometheus, 2020.

[13] Sachim Shelar. *Anomaly Detection using Gaussian Distribution*. Kaggle, 2018.

[14] FluentBit v 1.6. *Monitoring*. FluentBit, 2020.