

CAPITULO IV. SOFTWARE PARA MONITOREO Y SEGUIMIENTO DE TRANSACCIONES FINANCIERAS EN UNA RED DE RECAUDACIONES.

Autores:

Tamara Vera Castro, Ing.
Coordinadora De Sistemas De La Cruz Roja Del Guayas

Víctor Urdiales Ponce, Mgs.
Docente Facultad de Sistemas y Telecomunicaciones
Universidad ECOTEC
furdiales@ecotec.edu.ec

INTRODUCCIÓN

¿Cómo identificar o visualizar los errores en las transacciones financieras en tiempo real?, la idea es como ayudar a los administradores de la red de recaudaciones identificar escenarios mediante el uso de herramientas tecnológicas amigables, flexibles, escalables y personalizables, y que al mismo tiempo obtengan resultados oportunos que evidencie el escenario desde cualquier lugar en donde se dispare la novedad.

Si una transacción no se ejecuta con éxito, o simplemente no se realiza con los patrones esperados, esto genera un caso, un caso que debería ser identificado y analizado como una particularidad, la aplicación tiene la funcionalidad de reportar el acontecimiento, el tiempo de reacción para la solución del mismo depende de la identificación oportuna del agente que provoca tal situación <<servidor desconectado, archivo con algún patrón de error, datos erróneos>>, etc.

Generalmente, los administradores o controladores de las transacciones financieras, tienen un plan de monitoreo diario o una hoja de ruta que les indica como validar la funcionalidad correcta de los escenarios configurados en la aplicación <<hacer un ping a un servidor, o ejecutar un script para identificar transacciones fallidas>>, con la aplicación se pretende tener un historial de los

escenarios y del tiempo en que se presentó la situación, en una interface simple y sencilla.

Es por eso la importancia de esta investigación ya que mejora el proceso de monitoreo de servicio, logs o enlaces, propone una alternativa de control y verificación de diferentes escenarios, permite contrastar las posibles causas por las que el servicio no está disponible.

El propósito entonces es desarrollar una aplicación que permita el monitoreo de acontecimientos generados al momento de hacerse una transacción financiera, y de los componentes implicados en todo su flujo.

4.1 Monitoreo

El monitoreo trata sobre la forma de custodiar la situación o estado de un recurso, las misma que tienen como propósito de realizar el seguimiento o registro de las actividades de una red <<detección de eventos y comunicación de alertas>> para esto cada persona responsable debe tener claro la sobre la información que monitorea, tener acceso a información, las cuales se ajustan a políticas de administración, y tener conocimiento la forma del procesamiento de la información.

Drucker (2002) “Los elementos monitoreados pueden ser en su totalidad o en segmentos y se lo puede realizar en forma continua de una forma constante de los recursos o eventual, solo cuando el monitoreo lo amerita”.

En el mercado existen diferentes tipos de herramientas de monitoreo.

Tabla 1. Tipos de herramientas existentes de monitoreo.

Switch-Propietaria	NexTReTInternet	Zabbix
--------------------	-----------------	--------

Fuente: Elaboración Propia

A continuación, se presenta la interface de monitoreo de la herramienta zabbix.



Gráfico 1. Modelo de monitoreo Zabbix.
Fuente: Monitor Zabbix (Wikipedia, 2016).

4.2 Transacción.

Mientras que la transacción es una serie de operaciones que se lleva a cabo gracias a una programación aplicada a una base de datos, dicha función puede acceder al contenido de dicha base y modificarla, cada transacción debe presentar consistencia, aislamiento, exclusión y permanencia, según (Coulouris G, 2002) una transacción opera como un mecanismo para preservación de la consistencia de un conjunto de objetos de trabajo, su nacimiento ante la necesidad de los usuarios de ejecutar operaciones sobre una base de datos. La concepción de una transacción se introdujo según (Mitchell, 1982) en la teoría de sistemas distribuidos, bajo la forma de servidores de transacciones de archivos.

Monitor de Transacciones.

Es un programa <<software>> encargado de administrar peticiones simples, las mismas que son configuradas bajo ciertos requerimientos, los mismos que permiten reaccionar antes cualquier escenario, es decir, el monitor recibe una petición, luego la traduce en un lenguaje comprensible para el resto del sistema, idéntica la operación, y al final reporta un resultado al usuario.

Para el desarrollo del sistema se analiza la situación existente que permitió identificar el problema relacionado al monitoreo de transacciones, luego se identificó las tecnologías existentes <<interfaces de desarrollo>> para el desarrollo del software, y finalmente se realiza una propuesta de desarrollo de monitoreo para la resolución del problema.

Se seleccionó todos los componentes tecnológicos para la elaboración del sistema final de monitoreo de transacciones, para esto se toma los requerimientos que sirven de puente entre el levantamiento de necesidades y el desarrollo, se definieron requerimientos empresariales, funcionales y no funcionales.

4.3 Metodología para el desarrollo del programa de monitoreo.

Según (Ecured, 2010) La metodología de software consiste en una serie de tácticas, técnicas, herramientas y soporte registrado a la hora de desarrollar un producto (software). Se encasillan en dos tipos. Las más utilizadas son: La Programación Extrema (XP), Scrum y el Proceso Unificado de Software (RUP). Como objetivo fundamental se trata de establecer una visión sobre la importancia del oportuno monitoreo y de la existencia de algunas de las de aplicaciones existentes para este fin.

En la puesta a producción de diferentes aplicaciones los administradores suelen poner especial atención a los acontecimientos que podrían ocasionar que la operativa no se desempeñe con normalidad, pero con el pasar de los días y con la estabilidad alcanzada de la misma, podría olvidarse este procedimiento por ser tedioso y repetitivo, sin embargo, resulta importante realizar este procedimiento de forma continua y controlada. Se debe recordar que toda plataforma de negocio es un objeto binario vivo, el cual requiere de una constante supervisión de todos sus módulos, a fin de conocer a tiempo ambientes críticos como son las interrupciones de servicios, ataques a emisores-receptores de datos, tráfico anómalo o comportamientos dentro de la red que requieren de la

intervención del administrador para evitar colapsos o saturaciones que puedan poner en peligro la continuidad de la operación de la institución.

Hay que asegurarse de diseñar una solución que sea cómoda de implementar, que le ofrezca acceso a estar al tanto del estado de sus transacciones y la disponibilidad de su infraestructura de red en cualquier instante.

Es importante notar que esta aplicación debe permitir configurar los monitoreo según el giro del negocio, para esto hay que elegir una metodología que permita lograr código sin errores, con alta funcionabilidad, en plazos de tiempo satisfactorios. La Metodología de “Programación Extrema” (XP) propone la manera de alcanzar esos objetivos.

Adicional se ha escogido la programación orientada a componentes donde se logrará reutilizar código en diferentes instancias del proyecto.

4.4 Materiales y Métodos.

Monitor

Un monitor en este documento es una aplicación que almacena y alerta todos los sucesos reconocidos como riesgo, establecidos anticipadamente en la aplicación.

Prototipo

En el actual trabajo se limitó 3 categorías, 3 frentes el cual garantiza su funcionamiento: Base de Datos, Archivos (Log), Red. Inicialmente se desarrolló un modelo simple de como visualmente deberían mostrarse todos los registros identificados por el monitor en su ejecución.

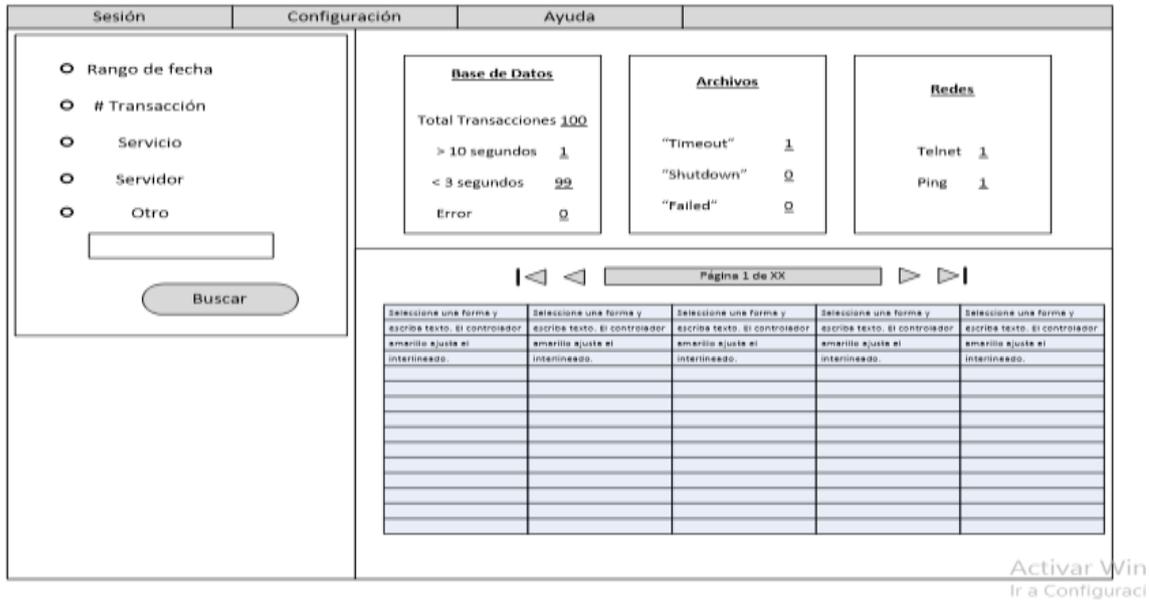


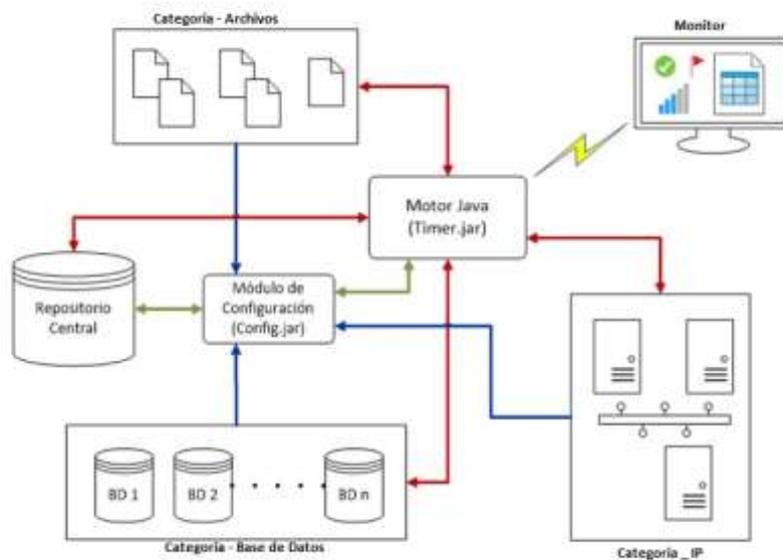
Gráfico 2. Prototipo de monitoreo.
Fuente: Elaboración propia.

Se decide que los eventos identificados para las diferentes categorías se mostraran de diferentes colores, para que así de esta forma los administradores solo con ver el color interpreten rápidamente que está comunicando el monitor.

Clases

Para lograr con el cumplimiento del monitoreo de las categorías se desarrollaron dos módulos en java. El módulo **Timer** compuesto por las clases **MonitorBean**, **MonitorTask** y **MonitorTimer**, desarrolladas para la conexión a base de datos, destinos y la ejecución de las tareas cada determinado tiempo, respectivamente. Esta clase se está ejecutando cada determinado intervalo de tiempo, verifica cuantas tareas existen configuradas y las ejecuta según la categoría, si encuentra que alguna de las condiciones programadas se cumple, arma un registro con la información de cumplimiento de la condición y la almacena en una tabla repositorio.

El módulo **Monitor** está compuesto por todas las interfaces que componen el monitor, este módulo está leyendo todos los registros que fueron almacenados en la base de datos por presentar novedades según su parametrización y las muestra en su interface principal con diferentes colores según su configuración.



Activar Winc

Gráfico 3. Diseño del manejo de información.
Fuente: Elaboración propia.

Base De Datos

Se diseñó una base de datos en Oracle 10G express para almacenar los monitoreos, las tareas y las categorías.

Adicional a las tablas de catálogos como muestra la ilustración 4 se diseñó una tabla bitacora_Transaccion. Esta tabla almacena todas las transacciones que fueron identificadas por la clase monitor.task ya sea de cualquiera de las 3 categorías. En esta tabla también está almacenado el código del color con el que se va mostrar el registro en la interface.

Name	Type	Nullable	Default	Comments
1 ID_CATEGORIA	NUMBER			Identificador de categoria
2 NOMBRE	VARCHAR2(20)			Nombre a mostrar en la seccion de categorias en la aplicacion
3 DESCRIPCION	VARCHAR2(500)	Y		Descripcion de la categoria
4 ESTADO	CHAR(1)			Estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_ALERTA	NUMBER			identificador de la alerta
2 ID_DESTINO	NUMBER	Y		identificador del destino
3 ID_PLANTILLA	NUMBER	Y		identificador de la plantilla
4 SONIDO	VARCHAR2(1000)			ruta del archivo de sonido que se utilizara
5 COLOR	VARCHAR2(100)			color del que se pintara el registro
6 ESTADO	CHAR(1)	Y		Estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_CATEGORIA	NUMBER	Y		identificador de categoria
2 ID_DESTINO	NUMBER			identificador del destino
3 SERVIDOR	VARCHAR2(30)	Y		direccion IP/Nombre del servidor destino a monitorear
4 PUERTO	NUMBER			Puerto del servidor destino a monitorear
5 USUARIO	VARCHAR2(30)			Usuario del servidor destino
6 CLAVE	VARCHAR2(30)			Clave del usuario del servidor destino
7 FRECUENCIA	NUMBER	Y		Intervalo de tiempo entre cada muestra
8 UNIDAD_FRECUENCIA	CHAR(1)	Y		Unidad de tiempo: M - Minutos, H - Horas, D - Dias
9 MOTOR_DB	VARCHAR2(30)			Motor de Base de Datos destino
10 SENTENCIA_SQL	VARCHAR2(4000)			Sentencia SQL a ejecutar para monitorear
11 SERVIDOR_SO	VARCHAR2(30)			Sistema Operativo del servidor destino
12 RUTA_ARCHIVO	VARCHAR2(2000)			Ruta del archivo a monitorear, debe llevar la extension respectiva
13 ACCION_NET	CHAR(1)			Accion de redes a ejecutar: P - Ping, T - Telnet

Name	Type	Nullable	Default	Comments
1 ID_USUARIO	NUMBER			identificador de usuario
2 USERNAME	VARCHAR2(30)			nombre de usuario en el sistema
3 NOMBRE_COMPLETO	VARCHAR2(300)			nombre completo del usuario
4 ID_PERFIL	NUMBER			identificador del perfil en el sistema
5 ESTADO	CHAR(1)			estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_OPCION	NUMBER			identificador de la opcion en el sistema
2 DESCRIPCION	VARCHAR2(50)			descripcion de la opcion
3 ESTADO	CHAR(1)			estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_PLANTILLA	NUMBER			identificador de la plantilla
2 DESCRIPCION	VARCHAR2(30)			Descripcion de la plantilla
3 ESTADO	CHAR(1)			Estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_DESTINO	NUMBER			identificador de destino
2 ID_PLANTILLA	NUMBER			identificador de la plantilla utilizada para el destino
3 ESTADO	CHAR(1)			Estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_OPCION	NUMBER			identificador de la opcion del sistema
2 ID_PERFIL	NUMBER			identificador del perfil
3 ESTADO	CHAR(1)	Y		estado: A - Activo, I - Inactivo

Name	Type	Nullable	Default	Comments
1 ID_PERFIL	NUMBER			identificador del perfil
2 DESCRIPCION	VARCHAR2(100)			descripcion del perfil
3 ESTADO	CHAR(1)			estado: A - Activo, I - Inactivo

Gráfico 4. Variables a utilizarse en la base de datos.

Fuente: Elaboración propia.

Desarrollo de software

Como metodología de desarrollo de software se escogió la programación extrema XP por ser liviana y simple, permite liberar versiones pequeñas del proyecto y así ir visualizando la versión final. Esta metodología hace énfasis en las pruebas unitarias y en las historias de usuarias. Y es en función a las historias de usuarios que se identificaron las necesidades y se desarrollaron luego los casos de uso y las pruebas funcionales.

RETOS Y PERSPECTIVAS DE LAS TECNOLOGÍAS DE INFORMACIÓN

Esta metodología también fue escogida por promover el principio de simplicidad en todo su desarrollo, a fin de evitar desarrollar de forma complicada y poner en riesgo el cronograma del proyecto, dado el proyecto es considerado como un desarrollo a corto plazo.

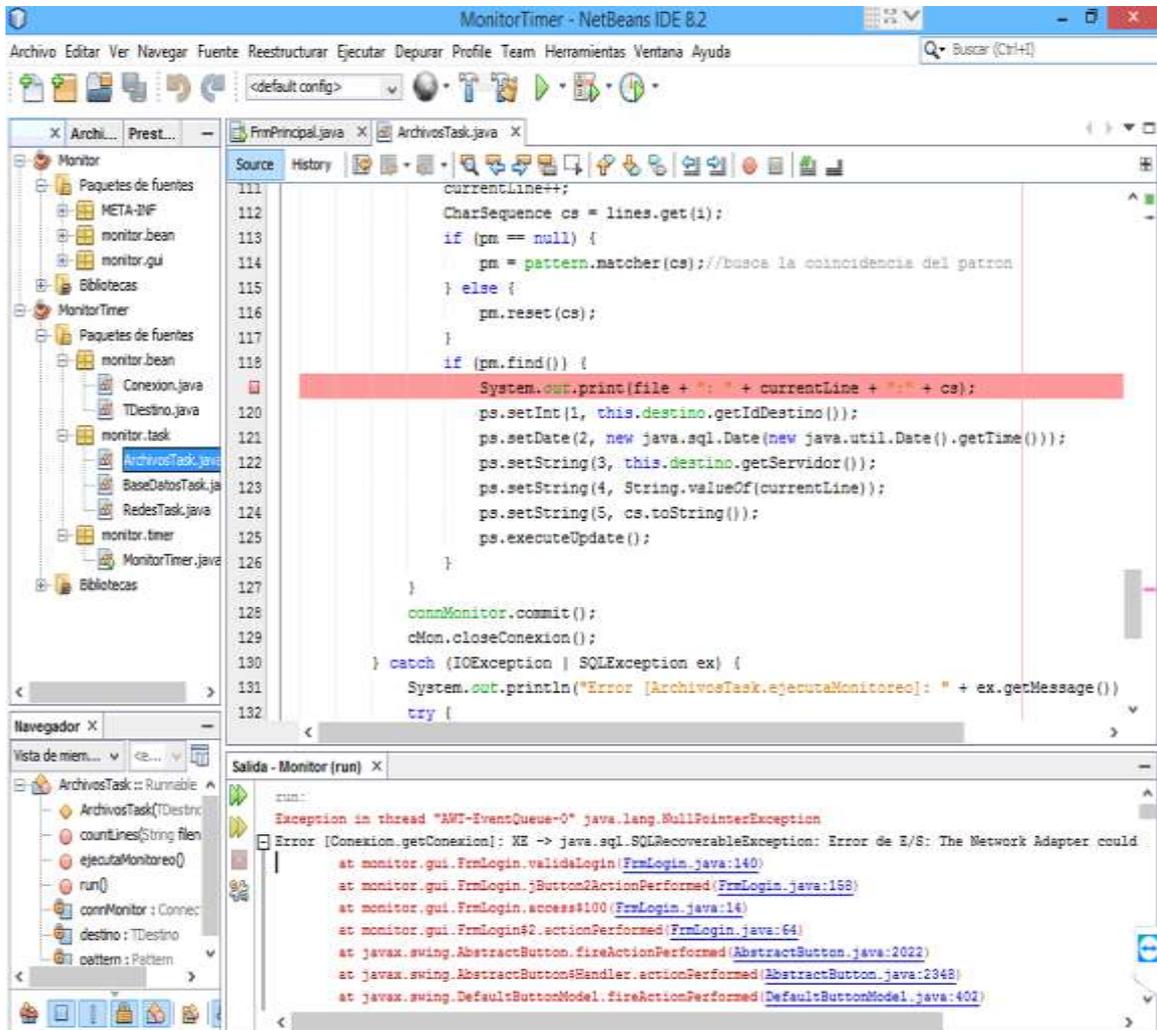


Gráfico 5. Interface de desarrollo de software.

Fuente: Elaboración propia.

5.5 Resultados

El monitor propuesto desarrollado en java y base de datos Oracle, genera una solución simple para la automatización de los monitoreos manuales.

La tabla 1 detalla en la acción 1 el monitoreo para identificar todas las transacciones que tienen como código de resultado '99', identificando el código

99 como un código de error. De esta misma forma se pueden configura diferentes escenarios que se necesite controlar en los datos. Como ya indicamos estos escenarios esta configurados en los destinos en este caso en el destino Base de Datos. Pueden existir diferentes destinos en diferentes servidores.

Tabla 2 Validación de acción 1.

Acción 1	Data	Esperado	Resultado
Se configura un destino a una base de datos Oracle, se parametriza el script de validación y se configura el tipo de alerta.	Select To_Number(Lg_Referencia), Lg_Fecha_Insert, 'Servidor' Servidor, 'CNEL' Producto, Lg_Valor_Total, Decode(Lg_Type_Trn, '10004', 'Recaudacion', 'Consulta') Tipo, Lg_Hora_Trn, Lg_Cod_Resultado From Monitor.Sb_Log Where Lg_Cod_Resultado = '99'	El monitor debe mostrar transacciones de ese destino con resultado 99	OK

Fuente: Elaboración propia.

En la acción 2 también muestra una configuración de transacciones donde el tiempo de respuesta es mayor a un valor de 5, se debe notar que en este escenario la transacción es tomada desde un servidor SQL.

En la acción 3 se parametriza la búsqueda de un texto en un archivo en una ruta especifica.

RETOS Y PERSPECTIVAS DE LAS TECNOLOGÍAS DE INFORMACIÓN

Tabla 3. Validación de acción 2.

Acción 2	Data	Esperado	Resultado
Se configura un destino adicional de base de datos SQL, se parametriza el script de validación y se configura el tipo de alerta.	SELECT lg_REFERENCIA ,lg_fecha_insert ,lg_TERMINAL ,lg_TYPE_TRN ,lg_VALOR_PAGADO ,lg_PRODUCTO FROM dbo.SB_LOG WHERE datediff(SECOND,lg_fecha_insert ,lg_fecha_update) > 5	El monitor debe mostrar transacciones de ese destino con tiempo de ejecución mayor a 5. Adicional se debe visualizar las transacciones del otro monitoreo	OK
Acción 3	Data	Esperado	Resultado
Se configura un monitoreo adicional de un destino archivos y se configura el tipo de alerta.	C:/Log.txt	El monitor debe alertar que el log está generando la palabra reservada. Debe continuar mostrando el monitoreo de los otros destinos.	OK

Fuente: Elaboración propia.

La acción 4 detalla el monitoreo de la IP que pertenece a un servidor.

Tabla 4. Validación de acción 3.

Acción 4	Data	Esperado	Resultado
Se configura que haga test a un servidor que no existe.	192.168.0.100	El monitor debe alertar que no hace ping.	OK

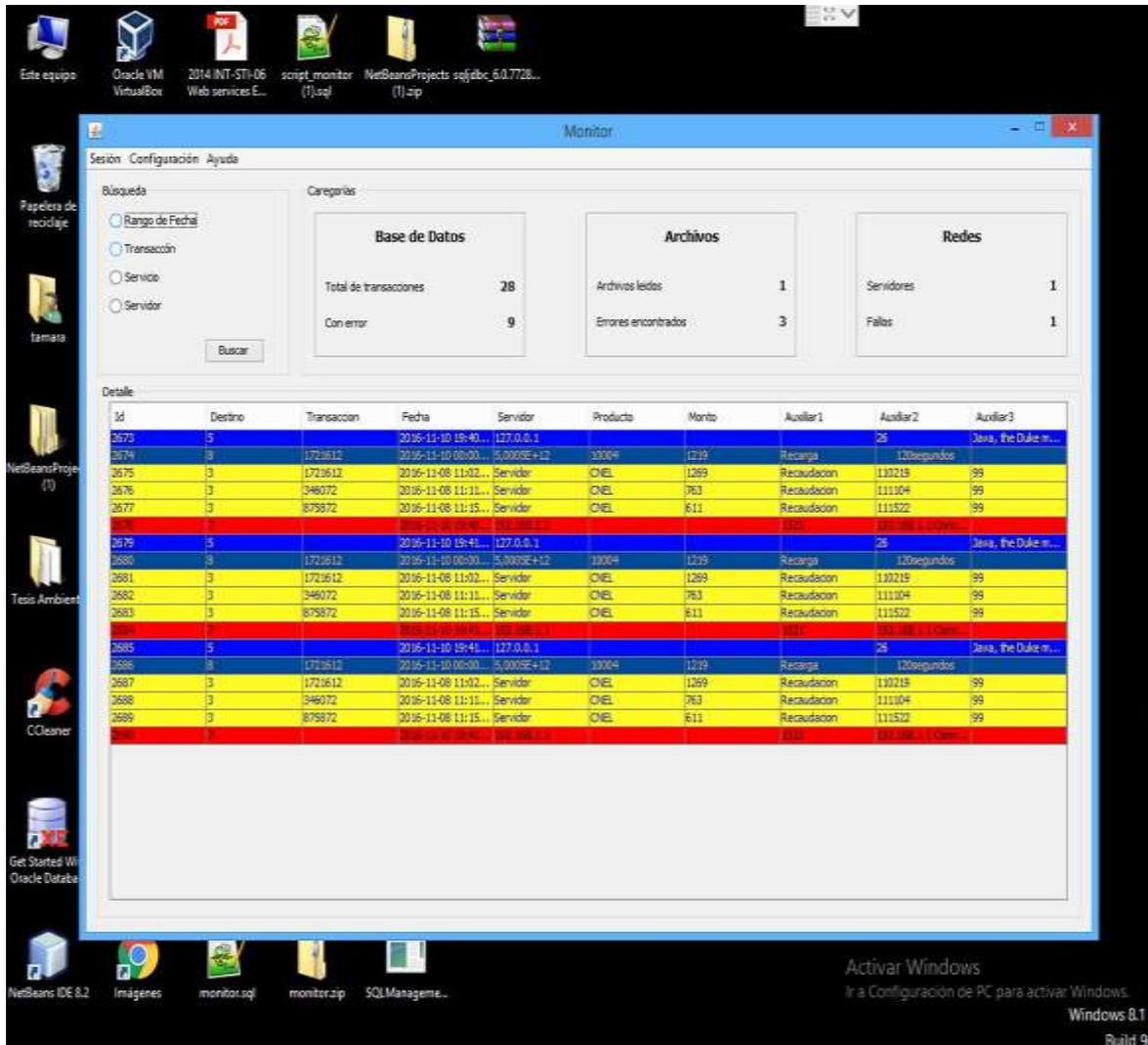
Fuente: Elaboración Propia

Finalmente se puede observar en el gráfico 6 como todas las transacciones que han sido identificadas en las diferentes bases de datos, los escenarios

RETOS Y PERSPECTIVAS DE LAS TECNOLOGÍAS DE INFORMACIÓN

identificados en los log o servidores son mostrados en diferentes colores en la interface principal. Esta interface se va refrescando cada determinado tiempo, el mismo que también fue parametrizado en la aplicación.

Gráfico 6. Interface de monitoreo.



Fuente: Elaboración propia.

Discusión

Uno de los mecanismos archivos Log para identificar eventos reportados, lo puede hacer antes que los usuarios lo identifiquen.

CONCLUSIONES

Se pensó el Software de monitoreo utilizando interfaces sencillas y amigables al operador intentando que la interpretación de la información mostrada por el mismo sea fácilmente analizada.

En el desarrollo de esta aplicación se manipuló la metodología de programación ágil XP a fin de alcanzar cumplir todas las funcionalidades de la aplicación. Todas las historias de usuario fueron desarrolladas en función a un ejemplo inicial logrando cubrir todos los desafíos planteados.

En la fase de pruebas se demostró que el monitor de aplicaciones se vinculaba a diferente base de datos, emparejaba las transacciones con error y las mostraba con los colores parametrizados en las alertas preliminarmente configuradas.

Adicional en la fase de pruebas se pudo exponer otras 2 funcionalidades del monitor, la disponibilidad de servidores o equipos en la red y la identificación de palabras claves en los log generados por distintas aplicaciones.

REFERENCIAS

- Aguilar, L. J. (2001). Fundamentos de Programación. Algoritmos y estructura de datos. Segunda edición. Madrid: Mc Graw Hill.
- ARAPE, N. (NOVIEMBRE de 2003). Desarrollo de Software Basado en Componentes . 8.
- Armijos, I. W. (2005). ANALISIS Y DISEÑO DE LA RED DE DATOS . Sangolqui.
- Atrium, O. (2014). Open Atrium. Recuperado el 10 de Octubre de 2014, de Open Atrium: <http://openatrium.com/#/>
- CCM Benchmark. (2012). CCM Enciclopedia virtual. Obtenido de <http://es.ccm.net/contents/295-redes-de-area-local>
- CISCO. (2012). Cisco. Obtenido de http://www.ie.itcr.ac.cr/acotoc/CISCO/R&S%20CCNA1/R&S_CCNA1_IT_N_Chapter9_Divisi%C3%B3n%20de%20redes%20IP%20en%20subredes.pdf

- Connolly, T. M., & E.Begg, C. (2005). Sistema de base de datos, Un enfoque práctico para diseño, implementación y gestión. Cuarta Edición. Madrid: Person Educación.
- Coulouris G, D. J. (2002). DISTRIBUTED SYSTEMS. ADDISON WESLEY.
- Daoqui, Y. (2010). Java Persistence with JPA. Outskirts Press.
- Darder Mesquida, A., & Pérez Garcías, A. (2011). La Tutoría para la Dirección de Proyectos de Investigación. Universitat de les Illes Balears. Palma: Universitat de les Illes Balears.
- De Siqueira Rocha, J. M. (2011). Desarrollo y Validación de soluciones tecnológicas para el Aprendizaje a través de Plataformas de e-learning Ingenio. Valencia: Universitat Politècnica de València.
- DLINK. (2015). DLINK. Obtenido de <http://www.dlink.com/uk/en/business-solutions/switching/managed-switches/layer-2/dgs-3420-series-xstack-l2plus-managed-stackable-gigabit-switches>
- Dominio Publico. (2012). DominioPublico. Obtenido de http://dominiopublico.com/intranets/lan_wan.php
- Drucker, P. F. (2002). La Gerencia en la Sociedad Futura. (J. Cárdenas, Trad.) Bogotá: Norma.
- EcuRed. (2015). EcuRed. Obtenido de http://www.ecured.cu/index.php/Red_de_%C3%81rea_Loca
- ecured. (mayo de 2010). <https://www.ecured.cu>. Obtenido de <https://www.ecured.cu>: https://www.ecured.cu/Metodolog%C3%ADas_de_desarrollo_de_softwar
e
- Feltre Oreja, R. (2008). Software Libre y La Construcción Ética de la Sociedad del Conocimiento. Barcelona, España: Icaria Editorial.
- Fernandez Alvarez, C. (06 de 07 de 2007). Obtenido de <http://hdl.handle.net/2099.1/5293>
- Fernando Pech-May, M. A.-R.-D.-J. (2010). Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces.
- Ferrer Martínez, J. (2012). Implantación de Aplicaciones Web. RA-MA.
- Fortinet. (2015). Fortinet Firewall. Obtenido de <http://www.fortinet.com/solutions/data-center-firewalls.html>
- Hernández, A. Los sistemas de informacìon. despertar y desarrollo.

- <http://www.lab.inf.uc3m.es/~a0080802/RAI/jpa.html>. (s.f.). Obtenido de lab.inf.uc3m.es
- Informatica Hoy. (2013). Informatica Hoy. Obtenido de <http://www.informatica-hoy.com.ar/redes/LAN-WAN-MAN-WLAN-WMAN-WWMAN-SAN-PAN.php>
- INTECO. (2003). ISO 10006:2003 - Sistema de Gestión de Calidad - Directrices para la Gestión de Calidad en los Proyectos. ISO/TC 176/SC 2.
- INTEL. (2012). Herramientas colaborativas en línea. (C. Intel, Ed.) Recuperado el 28 de Septiembre de 2014, de Iniciativa Intel Educación y el Programa Intel Educar: <http://www.intel.com/education/la/es/elementos/>
- IntelWorks. (2011). IntelWorks. Obtenido de http://www.intelworks.net.ec/index.php?option=com_content&view=article&id=69&Itemid=105
- ISO. (2014). ISO 9126. Recuperado el 10 de Octubre de 2014, de International Organization for Standardization: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=22750
- Jame, M. (2006). <http://scrumreferencecard.com/>. Recuperado el 12 de 06 de 2016, de <http://scrumreferencecard.com/reference-card-de-scrum/>
- Kaspersky Labs. (2014). Kaspersky. Obtenido de <https://usa.kaspersky.com/internet-security-center/definitions/web-filter#.VfcXGxGqqko>
- Kroenke, D. M. (2003). Procesamientos de Bases de Datos, Octava Edición. Mexico: Person Educación.
- Luján Mora, S. (2002). Programación de Aplicaciones Web: Historia, Principios Básicos y Clientes Web. Alicante, España: Editorial Club Universitario.
- Mendoza, M. A. (2014). WeLiveSecurity. Obtenido de <http://www.welivesecurity.com/la-es/2014/07/29/por-que-necesario-firewall-entornos-corporativos/>
- Mesquida, A. D., & García, A. P. (Agosto de 2014). Grupo de Tecnología Educativa. Recuperado el 14 de Agosto de 2014, de Grupo de Tecnología Educativa: <http://gte.uib.es/pape/gte/sites/gte.uib.es.pape.gte/files/tutoriaproyectos.pdf>

- Microsoft Corp. (2013). Technet. Obtenido de <https://technet.microsoft.com/es-es/library/cc755282.aspx>
- Microsoft Corp. (2015). Windows Microsoft. Obtenido de <http://windows.microsoft.com/es-xl/windows/what-is-firewall#1TC=windows-7>
- Ministerio de Telecomunicaciones y de la Sociedad de la Información. (Junio de 2014). Libro Blanco de Territorios Digitales en el Ecuador. Recuperado el 25 de Septiembre de 2014, de Ministerio de Telecomunicaciones y de la Sociedad de la Información: <http://www.telecomunicaciones.gob.ec/>
- Mitchell, J. J. (1982). A Comparison of Two network-based file servers. Comms. Vol 25, No 4 . ACM.
- Mohammed Abdul, J. F., & Ramirez Velarded, R. V. (2009). Herramientas Web 2.0 para el Aprendizaje Colaborativo. CYTED.
- NexTReT. (2015). NexTReT. Obtenido de www.nextret.net/
- P. Stevens, R. P. (2002). Utilización de UML en Ingeniería del Software con objetos y componentes . En R. P. P. Stevens, Utilización de UML en Ingeniería del Software con objetos y componentes (pág. 312). España: Pearson Educación.
- Pérez, D. R. (2007). Obtenido de <http://metinvc.blogspot.com/2012/02/t5b-proyecto-de-investigacion.html>
- Pérez, S. E. (05 de 05 de 2005). MailxMail. Recuperado el 28 de 05 de 2015, de <http://www.mailxmail.com/curso-componentes-pc-s/software-1>
- Project Management Institute. (2013). Guía de los Fundamentos de la Dirección de Proyectos PMBoK (Quinta ed.). Newtown Square, PA, E.E.U.U.: Project Management Institute.
- Propia, A. (2016). Guayaquil.
- ProyectosAgiles.org. (Enero de 2010). <http://www.proyectosagiles.org>. Obtenido de <http://www.proyectosagiles.org>: <http://www.proyectosagiles.org/que-es-scrum>
- Redbooth. (2014). Redbooth. Recuperado el 8 de Octubre de 2014, de Redbooth - Online Project Management Tools & Collaboration Platform: <https://redbooth.com/>
- Redes de Datos. (2012). Redes de Datos. Obtenido de <http://redesdedatosinfo.galeon.com/enlaces2128630.html>

- Redmine. (2014). Redmine. Recuperado el 10 de Octubre de 2014, de Redmine:
<http://www.redmine.org/>
- Rodríguez Gallardo, A. (2005). Tecnologías de la información y brecha digital en México 2001-2005 (Vol. IX). México: UNAM.
- Sampieri. Baptista, C. (2006). Metodología de la Investigación. McGrawHill.
- Universidad Nacional Autónoma de México. (2013). Firewall. México.
- Universidad Nacional Río Cuarto. (2011). La investigación tecnológica. Obtenido de <https://www.unrc.edu.ar/publicar/23/dossidos.html>
- WIKIPEDIA, F. (13 de MAYO de 2016). WIKIPEDIA. Obtenido de WIKIPEDIA:
<https://es.wikipedia.org/wiki/Zabbix>
- Wingwit. (2012). Wingwit Redes. Obtenido de <http://ordenador.wingwit.com/Redes/local-networks/72132.html#.VfcXaxGqqkp>
- ZABBIX. (OCTUBRE de 2016). ZABBIX. Obtenido de <http://www.zabbix.org/>