

# Gestión de Alertas Automáticas en base al Monitoreo del Hardware y Actualización del Inventario de IT (G.A.M.H.I.T)

Analía Lorena Ruhl, Silvia Edith Arias, Fabián Gibellini, M. Alejandra Di Gionantonio, Nora V. Flores, Mónica M. Serna, Daniel Arch, Ezequiel Ambrogio, Milagros Zea Cárdenas, Germán Parisi, Diego Barrionuevo

*Laboratorio de Sistemas de Información*

*Departamento de Ingeniería en Sistemas de Información*

*Universidad Tecnológica Nacional, Facultad Regional Córdoba*

*Cruz Roja S/N, 5016*

*analialorenaruhl@gmail.com, s\_autn@hotmail.com, fgibellini, ing.alejandradg, ingnoraflores, sernamonicam@gmail.com, daniel.arch@pjn.gov.ar, ezequielambrogio, milyzc, germanparisi, santosdiegob@gmail.com*

## Resumen

*G.A.M.H.I.T es una herramienta de software que permite monitorear ciertos recursos de hardware del sistema y en forma automática generar alertas ante reemplazos de los mismos, informando a los agentes responsables de esta situación mediante mails, sms, la web, o alguna otra vía configurable. El sistema informa los eventos identificados como no acordes a la correcta distribución, organización y funcionamiento de los recursos monitoreados.*

*Esta herramienta se ha diseñado para interactuar con el Sistema Seguridad en Ambientes Informáticos (SAI), del Laboratorio de Ingeniería en Sistemas de Información de la Facultad Regional Córdoba de la Universidad Tecnológica Nacional (Argentina), desarrollado durante el período 2008-2010 en el marco del Proyecto acreditado y financiado por la Secretaría de Ciencia y Técnica de Código: EIPRCO757.*

## 1. Introducción

El Laboratorio de Ingeniería en Sistemas de Información (LabSis) surge poco después de haber comenzado las carreras de informática en la Universidad Tecnológica Nacional – Facultad Regional Córdoba (UTN–FRC) y es un organismo dependiente del Departamento de Ingeniería en Sistemas de Información. Inicialmente, contaba con escasos recursos que, gradualmente fueron incrementando hasta ser hoy en día una gran plataforma tecnológica, compuesta de estaciones de trabajo, servidores y

dispositivos que permiten la administración de su red interna [1].

Este incremento, se debe al crecimiento constante de alumnos ingresantes, como así también a los distintos convenios que la U.T.N. - F.R.C. logra con diferentes entidades.

La actividad primordial del LabSis es poner a disposición de docentes, alumnos y grupos de investigación, las estaciones de trabajo e infraestructura con las que cuenta, para:

- Dictado cotidiano de distintas cátedras de la carrera de grado Ingeniería en Sistemas de Información, Diplomaturas o cursos dictados en la U.T.N. - F.R.C.
- Toma de exámenes finales de diferentes cátedras, que implica un proceso de respaldo de los datos de los mismos exámenes para su posterior corrección por los docentes.
- Organización de diversas competencias de programación en la UTN-FRC – Laboratorio de Investigación de Software (LIS): Análisis de la experiencia y de las opiniones de los participantes [2] [3] [4].
- Realización de las Olimpiadas Informática Córdoba (O.I.C.) [5].
- Prácticas libres, llamadas así porque se efectúan durante el tiempo libre de los alumnos.
- Desarrollo de software, capacitación, servicio y asesoramiento a terceros [6].

Para cumplir con su objetivo, LabSis utiliza sistemas automatizados que permiten la optimización de sus procesos y protocolos llevados a cabo en cada situación, como por ejemplo, el Sistema de Seguridad

Automática e Integral denominado S.A.I. [7] que contribuye a la prevención, detección de la sustracción y ataques a los equipos de un ambiente informático. El mismo, es una suite de herramientas conformada por tres subsistemas:

- Detección de Apertura de Gabinetes DAG [8]
- Sistemas Alternativos de Video Vigilancia [9]
- Sistema Distribuido de Seguimiento Local y Remoto [10]

El S.A.I. a través de D.A.G. permite detectar cualquier apertura en los gabinetes de los equipos de cada aula; los Sistemas alternativos de Video Vigilancia permiten el monitoreo de puntos claves del laboratorio a muy bajo costo y el Sistema Distribuido de Seguimiento Local y Remoto es la interfaz necesaria para el control de D.A.G.

Según ITIL (Infraestructura de Tecnologías de la Información) [11], un incidente es una interrupción no planificada de algún servicio IT, o una reducción en la calidad de un servicio IT.

Existen dos tipos de mantenimiento respecto al hardware, el mantenimiento preventivo y el mantenimiento correctivo. El Mantenimiento Preventivo describe las actividades de limpieza de partes, conexión de periféricos e inventario que se recomienda se realice periódicamente a los equipos de cómputo. Por otro lado el Mantenimiento Correctivo aborda la interpretación de los pitidos que genera el computador cuando hay problemas físicos que impiden su arranque, así como un compendio de las principales soluciones a los problemas físicos más frecuentes que impiden el funcionamiento del computador [12].

En todo ambiente informático es prioritario implementar normas de seguridad para resguardar los datos que en él se manejan, sin embargo, no podemos dejar de pensar en todo el equipo instalado que da soporte físico a los datos y que además, es un activo importante para un organismo o empresa, tales como: servidores, las estaciones de trabajo, computadoras personales y sus componentes [13] [14].

## 2. Contexto

En el marco de la gestión de incidentes, en el LabSis, cuando una estación de trabajo o servidor manifiesta un comportamiento que impide su normal desempeño, se notifica a los integrantes del Área Técnica quienes realizan pruebas sobre el equipo para detectar sus componentes afectados. En caso de ser necesario, el equipo queda fuera de cualquier uso hasta su reparación. Es decir, que el personal del establecimiento deberá, dependiendo de la cantidad de máquinas afectadas, redistribuir los recursos o las aulas. De lo antes planteado, deriva la necesidad que el personal de dicho laboratorio conozca en cualquier

momento: PCs habilitadas y el funcionamiento de sus respectivos componentes, para determinar con anterioridad a sus posibles usos, si surgió o no, alguna incidencia en las mismas. Una forma de detectar incidencias, con anterioridad a sus manifestaciones es *analizando la situación de cada computadora*. Este estadio está conformado por el *estado de cada uno de sus componentes, si el mismo*, al verificar *cambia de estado*, este es un incidente. Por ejemplo, cuando la placa de red cambia de situación puede significar que dejó de funcionar o que se reemplazó dicha placa por otra. Este cambio de estado, impacta en el estadio de la computadora a la que pertenece la placa. A su vez, cada variación de estado puede representar el inicio de un proceso de mantenimiento o reparación para el equipo, el cual puede provocar más modificaciones en el equipo referenciado, por ejemplo, el reemplazo de algún componente por uno nuevo.

Dentro de este ambiente académico, siempre está la necesidad de lograr una mejora continua de los servicios brindados. En este caso, se busca optimizar la productividad de las PCs que están distribuidas dentro de distintas subredes que conforman cada una de las aulas de LabSis y alcanzar, a mediano plazo, cierta predictibilidad en su comportamiento, dado que son el recurso más usado del laboratorio.

## 3. Objetivos

En el contexto planteado, se identificaron las siguientes necesidades:

- Llevar un inventario informatizado, catalogado y actualizado de todas las computadoras, pertenecientes a la plataforma tecnológica, distribuidas en cada una de las aulas.
- Lograr una trazabilidad del estado de los activos inventariados, lo que exige un monitoreo de los mismos, para conocer cierta predictibilidad en su comportamiento a corto plazo, ya que esto repercute sustancialmente en los sistemas informáticos que se ejecutan sobre estos activos; maximizando y potenciando su rendimiento y eficiencia.
- Dar soporte a la toma de decisiones relacionadas al hardware, como por ejemplo el tiempo promedio de vida útil de cierto dispositivo lo que permitirá establecer períodos para la compra de los mismos basados en datos de la realidad.
- Permitir mejorar la distribución de los costos de inversión en esta área.

Es fundamental, que la solución que cubra las necesidades planteadas sea un sistema libre y sin restricciones de funcionalidad alguna, de forma que

quede disponible para toda institución que desee implementarlo.

Para cubrir las necesidades planteadas, el sistema debe ser capaz de:

- Monitorear ciertos componentes de los activos de TI, para obtener una trazabilidad de sus estados y alimentar una base de datos que permitirá tomar futuras decisiones relacionadas al hardware,
- Generar una base de conocimiento de inventarios y cambios de estado de los activos monitoreados.
- Generar alertas enviadas automáticamente a los responsables, cada vez que se registre algún cambio y/o modificación en los equipos monitoreados.
- Brindar informes estadísticos sobre cambios de hardware ocurridos en los equipos, y reportes con información relevante a partir de los datos históricos que almacenará el mismo, para dar soporte a la toma de decisiones relacionada al hardware en el LabSis.
- Reportar diariamente el contenido de cada computadora y sus respectivos cambios, enviando un mail a la persona encargada, con el detalle de los elementos modificados en el equipo.
- Otorgar información actualizada de cada componente de hardware de las computadoras presentes en la red.
- Detectar e inventariar los componentes de las computadoras después de eventos inesperados, como cortes de energía.
- Poseer una licencia de software libre [15].

#### 4. Metodología

El presente proyecto, está inserto dentro del proyecto homologado por la Secretaría de Ciencia y Técnica de la Universidad Tecnológica Nacional "Inventario Seguro en Ambiente Informáticos con Alertas Automáticas y Registro de historial para activos de TI" - Código: EIUTNCO0002226, radicado en el Laboratorio de Ingeniería en Sistemas de Información (LabSis) de la Facultad Regional Córdoba (F.R.C.). La institución que acredita el proyecto es la Universidad Tecnológica Nacional (U.T.N. Argentina), quien es a su vez, el organismo que contribuye a su financiamiento.

El proceso de control empleado para el proyecto, es un proceso empírico o empirismo [16], el cual asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce.

Existen tres pilares que sostienen cada implementación de un proceso de control empírico: la transparencia, la inspección, y la adaptación.

- Transparencia: Los aspectos significativos del proceso deben ser visibles a los responsables de los resultados.
- Inspección: El equipo debe inspeccionar con frecuencia lo que está produciendo, y cotejar con los objetivos para detectar variaciones indeseables.
- Adaptación: Si se determina que uno, o más aspectos de un proceso se desvían fuera de los límites aceptables, y que el producto resultante será inaceptable, se debe ajustar el proceso o el material que está siendo procesado. El ajuste debe hacerse tan pronto como sea posible, para minimizar aún más la desviación.

Particularmente, se aplicó una adaptación del framework SCRUM [17] según las necesidades del equipo para aumentar la eficiencia de las personas involucradas en el proyecto a través del compromiso con el mismo y el trabajo en equipo. SCRUM tiene sus bases en los principios Ágiles [18], listados a continuación:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

Según Highsmith & Cockburn [19], lo que es nuevo en los procesos ágiles no son las prácticas que usan, sino que reconozcan a las personas como primeros implicados en el éxito de un proyecto, además de un intenso foco en la efectividad y la manejabilidad. Esto, genera una nueva combinación de valores y principios que definen una visión ágil del mundo.

SCRUM emplea un enfoque iterativo e incremental para optimizar la previsibilidad del riesgo y control. Utilizar esta metodología, consiste en:

1. Definir Equipos Scrum con funciones y roles bien definidos:
  - a. Product Owner (PO): conoce el dominio de negocio.
  - b. Scrum Master (SM): su principal responsabilidad es facilitar las gestiones que se le puedan presentar al equipo. También es parte del equipo.
  - c. Scrum Team (ST), el resto del equipo.

2. Establecer cuatro reuniones formales para la inspección y la adaptación:
  - a. Sprint Planning (Planificación de Sprint)
  - b. Daily Scrum
  - c. Sprint Review (Revisión del Sprint): Reunión con el PO para la aceptación o no del resultado del sprint.
  - d. Sprint Retrospective (Retrospectiva del Sprint): El objetivo es mejorar el equipo a través de un feedback respecto al sprint que está terminando.

La adecuación realizada a SCRUM para que se adapte a las características de este proyecto respeta la definición de equipo propuesta, pues cada miembro de este grupo de investigación tiene bien definido su rol. Se fijó cada iteración con una duración de 25 días. Las reuniones formales establecidas se cumplen pero no con la frecuencia sugerida. Las reuniones Sprint Planning y Sprint Review, se llevan a cabo en una sola instancia cada 25 días. En ella, cada equipo expone su trabajo, que incluye los avances y dificultades que se hayan presentado. Luego, se analizan los resultados del sprint en forma conjunta. Por último, en base a éste, se planifican las actividades del próximo sprint y se distribuyen las tareas. La Daily Scrum, se realizan

cada 5 días y es informal. La Sprint Retrospective se realiza el mismo día después de la Sprint Review o en ciertas circunstancias, se fija una fecha la misma.

Después de una investigación exploratoria de las herramientas de software libres existentes en el mercado y posteriormente, un análisis exhaustivo de cada una de ellas considerando criterios de funcionalidad como: sistemas operativos soportados, mantención de un historial de cambios, componentes soportados, generación de alertas automáticas, exportación de datos. Se determinó la reutilización de un software llamado OCS Inventory [20] [21] y tomar sus servicios para que se empleen como entradas del subsistema que se encarga del monitoreo, la generación de alertas ante cambios y el historial de inventarios.

Las primeras pruebas de funcionamiento de la herramienta G.A.M.H.I.T, se realizaron en LabSis, donde seguimos trabajando en la detección de fallas y el diseño de mejoras. En las siguientes etapas, se llevará a cabo la puesta a punto del sistema, para posteriormente extender su implementación a los laboratorios de otras especialidades u organizaciones e instituciones que deseen utilizarlo.

En la Figura 1 se muestra de forma ilustrativa una salida del subsistema de monitoreo, generación de alertas ante cambios y el historial de inventarios.

The screenshot shows a web application interface titled 'Sistema de Alertas Automáticas'. On the left is a navigation menu with 'Sincronizar', 'Reportes', and 'Configuración'. The main content area is titled 'Configurar roles por tipo de cambio' and contains a table with columns for 'Tipo de cambio' and 'Roles de usuario' (Administrador, Operador, Técnico). A grid of checkboxes allows assigning roles to various change types. An 'Actualizar' button is at the bottom right of the table.

Tipo de cambio	Roles de usuario		
	Administrador	Operador	Técnico
Cualquier Cambio	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Nuevas	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Procesadores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bios	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Discos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Memorias	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Monitores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Periféricos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Placas De Red	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Placas De Sonido	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Placas De Video	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

LABSIS | 2014

**Figura 1. Presenta la configuración de qué tipos de cambios serán reportados a cada usuario según su rol. Por ejemplo a un Operador va a ser notificado ante cualquier cambio que ocurra en las máquinas monitoreadas.**

Sin embargo, a partir de las pruebas realizadas al subsistema de monitoreo se detectó cierta inconsistencia en los datos generados por el OCS

Inventory. El patrón de estas inconsistencias no pudo ser identificado.

La aplicación de esta metodología de trabajo, con sus respectivas variaciones, junto con la autonomía del

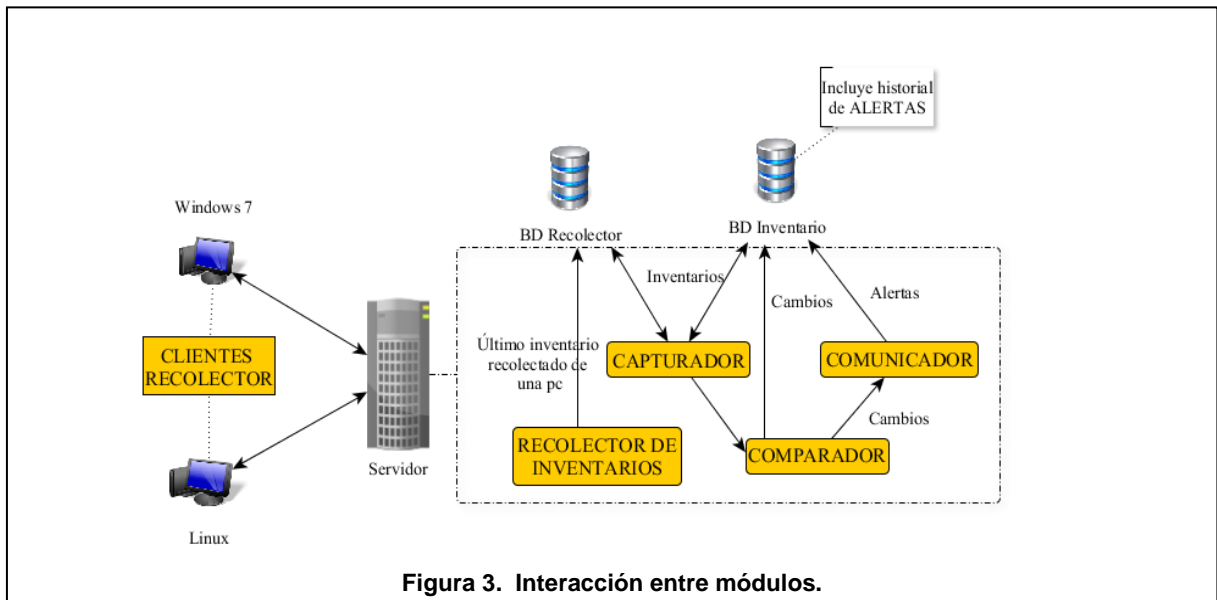
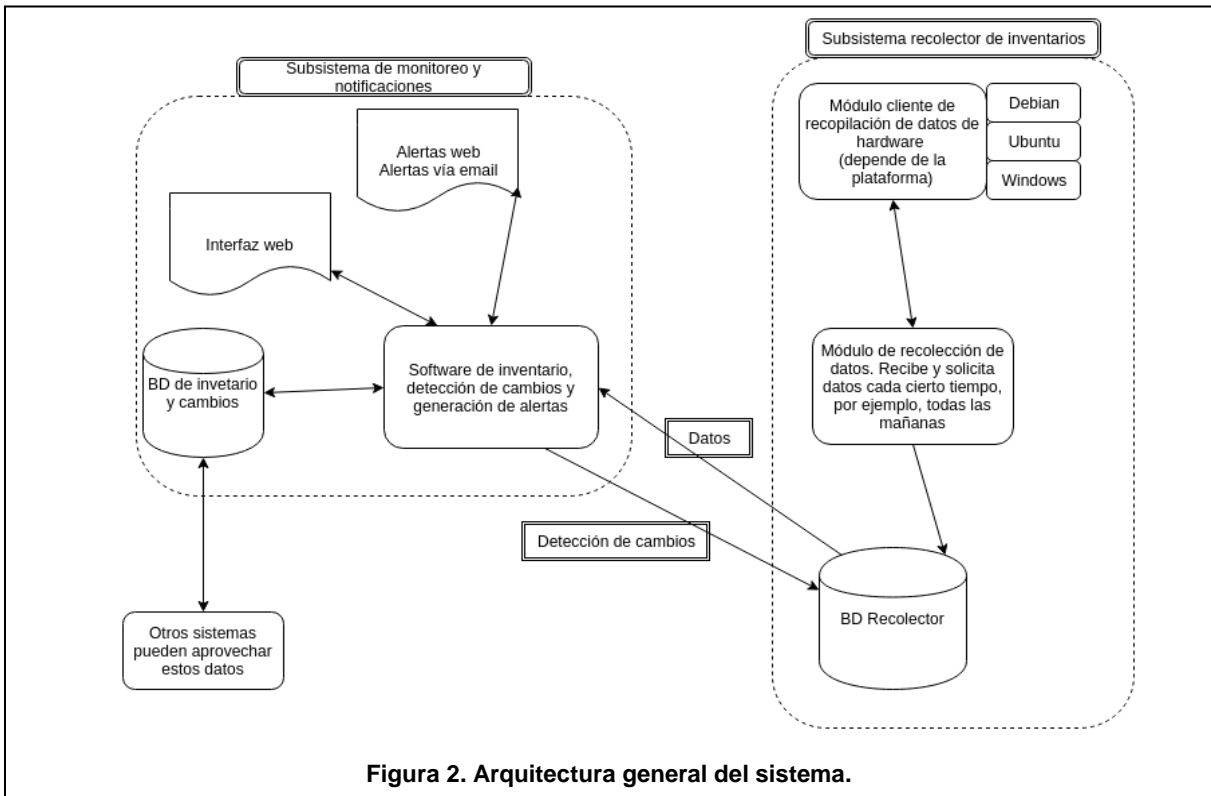
subsistema ya desarrollado, permitió analizar el problema y sugerir otras soluciones. De este análisis, surgieron nuevos aspectos a considerar en la investigación, el replanteo de los alcances del sistema en desarrollo y una readequación de los tiempos de calendario inicialmente propuestos.

### 5. Resultados

La arquitectura general del sistema, como lo muestra la Figura 2, está conformada por dos subsistemas: el subsistema de monitoreo y notificaciones encargado de la secuencia de chequeos permanente, historial de cambios, notificaciones a los usuarios ante cualquier modificación y de alimentar

una base de datos potencialmente usable por otros sistemas; y el subsistema recolector de inventarios encargado de solicitar, recibir y almacenar los registros de las máquinas monitoreadas.

La flexibilidad y autonomía lograda en el subsistema de monitoreo, permitió reemplazar el software externo OCS Inventory por un subsistema recolector de inventarios, que actualmente está en desarrollo, con un impacto mínimo ya que se logró una capa de abstracción para el procesamiento de los datos con cierta flexibilidad e independencia de la fuente que lo genera.



El subsistema “Recolector de Inventarios”, conformado por un módulo homónimo y módulos clientes, para lo que se ha determinado qué componentes de hardware se registrará en cada máquina. Los clientes recolectores tendrán soporte para sistemas operativos Linux y Windows. Una vez instalado un cliente, se registrará la IP del servidor para que el equipo sepa a quién reportar, de esta manera quedará vinculada la relación entre cliente-servidor. En la primera conexión cliente-servidor, el servidor enviará un archivo de configuración que contiene un número de identificación, tiempo programado para que el cliente se ejecute y los datos de los componentes solicitados (disco, memoria ram, procesador, etc.). Cada vez que un cliente reporta su inventario, éste se almacena en la BD Recolector sobrescribiendo el último inventario registrado para ese cliente.

El subsistema de monitoreo y notificaciones, está integrado por el módulo Capturador, Comparador y Comunicador. El Capturador, es el encargado de tomar los datos de la BD Recolector de los equipos reportados, por un lado y por otro los datos con su correspondiente equipo de la BD Inventario, armando así tres listas de equipos, los recolectados, los ya inventariados y los nuevos equipos. Estos datos son enviados al módulo Comparador, quien de las tres listas ordenadas utiliza las dos de las máquinas existente comparando registro por registro si hubo algún cambio, y la tercer lista solo contiene los equipos nuevos, por lo cual no se contrasta con ninguna anterior. Una vez conocidos los cambios ocurridos, el módulo Comunicador será encargado de registrar los cambios ya sea de actualización de un equipo existente o registración de un equipo nuevo en la BD Inventario y genera automáticamente las notificaciones que se envían a los usuarios correspondientes mediante e-mail y alertas web.

## 6. Discusión

Los resultados obtenidos con el subsistema de monitoreo permitieron extender el horizonte del proyecto ya que gracias a la capa de abstracción en el subsistema ya desarrollado, se decidió sustituir el sistema OCS Inventory por uno propio, el *subsistema recolector de inventario*. Por otro lado, los datos generados por el subsistema de monitoreo son datos potencialmente utilizables por otros sistemas que apliquen diversos algoritmos para alcanzar cierta predictibilidad en el comportamiento de los activos monitoreados, lo que dio pie a iniciar una colaboración con el proyecto “Generación de Modelo Descriptivo para la prevención de incidentes en equipos informáticos en el contexto del laboratorio de sistemas (Fase II)” [22], el cual también se lleva a cabo en la UTN - FRC.

## 7. Conclusion

La flexibilidad que ha adquirido el sistema y dado que las necesidades que este cubre son actualmente las mismas necesidades de muchas organizaciones, entidades e instituciones tanto públicas como privadas que manejan grandes plataformas de activos IT que deben ser monitoreados y resguardados, es lo que le da al software la capacidad de ser aplicable en cualquier ámbito a un bajo costo de implementación

Si bien el desarrollo de este Sistema fue pensado para cubrir las necesidades del LabSis, puede ser aplicable a cualquier institución pública o privada, con una adaptación adecuada de su código, solo en caso de ser necesaria, gracias a que el software poseerá una licencia libre.

De esta forma, se le otorga un control total de adaptación, modificación y distribución del software para quien desee implementarlo, siempre respetando el mantener el nombre original de la entidad que desarrollo la versión original, es decir LabSis.

## 8. Referencias

- [1] Laboratorio de Sistemas (LabSis). <http://labsys.frc.utn.edu.ar/noticias.php?id=9>. Última Visita: 18-08-2016.
- [2] Laboratorio de Investigación de Software (LIS). Página oficial: <http://www.investigacion.frc.utn.edu.ar/mslabs/>. Última Visita: 18-08-2016.
- [3] Castillo, J., Serrano, D., Cárdenas, M. “Organización de competencias de programación en la UTN-FRC: Análisis de la experiencia y de las opiniones de los participantes” Dpto. Ingeniería en Sistemas de Información - Universidad Tecnológica Nacional – Facultad Regional Córdoba. 2013. <http://www.conaiisi.unsl.edu.ar/2013/221-562-1-DR.pdf>. Última Visita: 18-08-2016.
- [4] Castillo, J., Cárdenas, M., Serrano, D. “Experiencias en el Desarrollo de Competencias de Programación en UTN-FRC” Dpto. Ingeniería en Sistemas de Información - Universidad Tecnológica Nacional – Facultad Regional Córdoba. TE&ET 2011. <http://sedici.unlp.edu.ar/handle/10915/18414>. Última Visita: 18-08-2016.
- [5] Marciszack, M., Muñoz, R., Castillo, J., Delgado A., Serrano, D., Gatto, S. "Colaboración entre el Gobierno de la Provincia de Córdoba y la UTN - FRC para el desarrollo de Olimpíadas Informáticas", Departamento de Ingeniería en Sistemas Universidad Tecnológica Nacional, Facultad Regional Córdoba y Dirección de Divulgación y Enseñanza de las Ciencias Ministerio de Ciencia y Tecnología de la Provincia de Córdoba. 2013. <http://conaiisi.frc.utn.edu.ar/PDFsParaPublicar/1/schedConfs/4/234-650-1-DR.pdf> Última Visita: 18-08-2016.
- [6] <http://agenciadeempleoyfp.cba.gov.ar/gobierno-amplia-su-oferta-de-cursos-de-oficios-con-la-utn-y-la-fuo/>
- [7] Gibellini, F., Morales, C., Quispe, M., “Seguridad en Ambientes Informáticos (SAI)”. Laboratorio de Sistemas.

Departamento de Sistemas Universidad Tecnológica Nacional Facultad Córdoba. JIDIS.  
<http://www.jidis.frc.utn.edu.ar/papers/e7c362c8b5427c807ee23beab34d.pdf> Última Visita: 18-08-2016.

[8] Morales, C. “Detección de Apertura de Gabinetes”. Laboratorio de Sistemas Departamento de Sistemas. Universidad Tecnológica Nacional Facultad Córdoba. CNEISI.  
<http://www.cneisi.frc.utn.edu.ar/papers/b998c93b46bb857450dfc6a89a03.pdf>. Última Visita: 18-08-2016.

[9] Morales, C. “Sistemas alternativos de video vigilancia”. Laboratorio de Sistemas Departamento de Sistemas. Universidad Tecnológica Nacional Facultad Córdoba. CNEISI.  
<http://www.cneisi.frc.utn.edu.ar/papers/b998c93b46bb857450dfc6a89a03.pdf>. Última Visita: 18-08-2016.

[10] Arias, Silvia et al. “Sistema Distribuido de Seguimiento Local y Remoto”. Laboratorio de Sistemas Departamento de Sistemas. Universidad Tecnológica Nacional Facultad Córdoba. CNEISI.  
[http://www.frsf.utn.edu.ar/cneisi2010/archivos/10-Sistema\\_Distribuido\\_de\\_Seguimiento\\_Local\\_y\\_Remoto.pdf](http://www.frsf.utn.edu.ar/cneisi2010/archivos/10-Sistema_Distribuido_de_Seguimiento_Local_y_Remoto.pdf)  
<http://laboratorios.fi.uba.ar/lie/Revista/Articulos/020205/A2ago2005.pdf> Última Visita: 18-08-2016.

[11] Cartledge, A., et al. “The IT Infrastructure Library An Introductory Overview of ITIL® V3”. ITSMF. Ed: Xansa - Steria, itSMF UK. ISBN: 0-9551245-8-1. UK. 2007.

[12] Armero Kreisberger, S. “Mantenimiento de Computadoras”. Universidad del Cauca (Ed.). Popayán, Colombia. ISBN: 978-958-732-084-8. Mayo 2011.

[13] Sheran Lin, S., Thompson, G., Malaviya, V. “Embedded approach for device inventory collection utilizing OS programmability” SSTG, Cisco Systems 170 W Tasman Drive, San Jose, California, U.S.A. E- ISBN: 978-1-4244-9220-6. Print ISBN: 978-1-4244-9219-0. May 2011.

[14] N. D. Arnold and D. A. Dohan. “Connection-Oriented Relational Database of the APS Control System Hardware”. Argonne National Laboratory, Argonne, IL 60439.USA. 2003.

[15] Yingkui, Z., Jing, Z., Liye, W. “Justification of Free Software and its Enlightenment”. Econ. & Manage, Beijing Univ. of Chem. Technol., Beijing, China. Software Engineering (WCSE). Volume: 2. Page(s):171 – 173. ISBN: 978-1-4244-9287-9 DOI: 10.1109/WCSE.2010.47. IEEE. 19-20 Dec. 2010.

[16] Sini, C., Vattimo G., Et al. “Enciclopedia Garzanti della Filosofia”. 2004.  
<http://filosofiaypensamiento.blogspot.com/2009/03/y2a-empirismo.html>. Última Visita: 18-08-2016.

[17] Framework Scrum. <https://www.scrumalliance.org/why-scrum/scrums-guide>. Última Visita: 18-08-2016.

[18] “Manifiesto Ágil”. 2001, Acceso a la información: <http://agilemanifesto.org/>. Última Visita: 18-08-2016.

[19] Highsmith, J., Cockburn, A. “Agile Software Development: The Business of Innovation”. Computer Science. Department. University of Southern California .Editor: Barry Boehm. Los Ángeles, CA 90089. Febrero 2009

[20] OCS Inventory, <http://wiki.ocsinventoryng.org/index.php/Documentation:Administration> Última Visita: 18-08-2016.

[21] OCS Inventory, Sincronización, <http://wiki.ocsinventory-ng.org/index.php/Documentation:OCSSynchroLDAP>

[22] Corso, C., Maldonado, C., Gibellini, F., Ciceri, L., Martinez, G., Pereyra, F., Donnet, M. “Generación de Modelo Descriptivo para la prevención de incidentes en equipos informáticos en el contexto del laboratorio de sistemas (Fase II)”. Departamento de Ingeniería en Sistemas Universidad Tecnológica Nacional, Facultad Regional Córdoba.