INSTITUTO DE COMPUTACIÓN - FACULTAD DE INGENIERÍA – UDELAR – HOSPITAL DE CLINICAS

Gestión de Registros y Respaldos en el Contexto Hospitalario.

Proyecto de grado

Edición 2009

Informe Final

Tribunal:	**Supervisores:**	**Responsables:**	**Estudiantes:**
Andrés Aguirre	María Eugenia Corti	Julio Carrau	Martín Calabria
Lorena Etcheverry	Ariel Sabiguero	Gustavo Perez	Gonzalo Perretti
Antonio Mauttone			

Resumen

El objetivo central del presente trabajo es contribuir con la actualización y fortalecimientos de las características de seguridad informática que posee el Hospital de Clínicas.

Un frente del proyecto se centro en actualizar el mecanismo de respaldo de información del Hospital de Clínicas debido a que en la actualidad solo se resguarda la información de los servidores con sistemas operativos Windows mediante una herramienta nativa.

Con esta finalidad se realizó el estudio de 6 herramientas de respaldo, donde se analizaron las características de cada una, identificando sus fortalezas y debilidades. En base a este análisis se escogió la que más se adecuaba a las especificaciones de los clientes y a la infraestructura de hospital, procediendo a la instalación y puesta en producción del mismo en un ambiente de prueba en el propio hospital.

El otro frente plantea la creación de un sistema que permita registrar eventos generados por los sistemas que se utilizan en el sanatorio y brinde una interfaz donde se puedan auditar los mismos remotamente. Este sistema apunta a facilitar la interoperabilidad suministrando un servicio que pueda ser consumido por el resto de las aplicaciones sin que esto suponga grandes cambios en los sistemas ya existentes.

A su vez se deben respetar los estándares definidos para los sistemas informáticos en el contexto hospitalario que establecen las características de seguridad y confidencialidad que debe proveer la solución implementada.

La solución propuesta es adaptable a distintos dominios de aplicación, estandarizada en el uso de formatos, presenta un diseño que permite la extensión a nuevas funciones y fue construido utilizando tecnologías vigentes y de gran difusión.

La implementación del perfil ATNA es la primera en Uruguay lo que convierte al Hospital de Clínicas en pionero en la utilización de un sistema que cumple con las normativas internacionales de seguridad en ambientes de salud.

Palabras Claves

Registro de eventos, ATNA, IHE, Herramientas de Respaldo, Registros.

Tabla de contenidos

Tabla de Figuras

Tabla de Figuras

1 Introducción

Este proyecto de grado se enmarca dentro de la colaboración entre el Departamento de Procesamiento Informático (DPI) del Hospital de Clínicas y el Instituto de Computación (InCo) de la Facultad de Ingeniería (FING).

Consta de dos partes, la primera es el Estudio del Estado del Arte de los programas de respaldo y la selección de una solución que se adecue a las necesidades especificadas por el DPI. La segunda tiene como objetivo la creación de un sistema informático que permita registrar eventos provenientes de las aplicaciones existentes en el contexto hospitalario, para luego realizar la auditoría correspondiente sobre los mismos.

1.1 Motivación

A continuación se presentan las principales motivaciones para cada uno de los frentes del proyecto.

Sistema de Respaldo

Los datos que se manejan en la institución consisten principalmente en información personal del paciente, como es la historia clínica. Estos datos son de importancia tanto para el paciente como para los individuos involucrados en el cuidado y tratamiento del paciente, lo cual hace que la pérdida de los mismos o la no disponibilidad por un periodo prolongado de tiempo sea algo que no se puede permitir.

Actualmente en el Hospital de Clínicas se utiliza VSS [1] (Volume Shadow Copy Service) como herramienta de respaldo de la información. Esta herramienta viene incluida en el sistema operativo Windows por lo que únicamente aquellos servidores que cuenten con el mismo podrán utilizarlo para realizar respaldos. Dada la existencia de servidores con otros sistemas operativos, surge la necesidad de utilizar otro mecanismo de respaldo para mantener resguardada la información almacenada en dichos servidores.

Por esta razón se requiere un sistema de respaldo multiplataforma, que permita resguardar la información de cualquier computadora perteneciente al entorno del Hospital de Clínicas independientemente del sistema operativo que se utilice y que a su vez brinde la posibilidad de auditar, examinar y restaurar la información respaldada.

Sistema de Registros

Tener conocimiento de las actividades que se realizan en los sistemas informáticos, realizar el seguimiento de las mismas, y verificar que la manipulación de la información se realiza correctamente son algunos de los pilares definidos en las políticas de seguridad informática del Hospital de Clínicas.

Actualmente no se dispone de ninguna herramienta que centralice el registro de eventos generados en el entorno de trabajo del Hospital.

Si bien se cuenta con el registro de eventos de Windows, este almacena mucha información que no es relevante para los administradores del DPI lo cual dificulta sensiblemente la búsqueda y seguimiento de los eventos. Por otra parte esta herramienta no fue diseñada para realizar seguimientos de las aplicaciones de usuario, en este caso desarrolladas por el DPI, cuyos eventos son de real interés para los administradores, como pueden ser eventos asociados al acceso a la información del paciente mediante algunos de las aplicaciones utilizadas en la institución. En relación a esto los clientes nos solicitaron implementar un sistema que deberá ser capaz de registrar eventos que se generen en los sistemas actuales de la institución sin que esto suponga realizar cambios de configuración del sistema y/o requiera registrar las aplicaciones que lo utilizarán.

Como última desventaja de la utilización de esta herramienta se puede mencionar la descentralización del registro de eventos en cada máquina cliente, lo cual dificulta la recolección y auditoría de los mismos y no cumple con otro de los requisitos definidos por el DPI, el cual indica que el sistema de registro de eventos debe ser centralizado en un servidor con las medidas de seguridad correspondientes. Este es un detalle importante, ya que utilizar el registro de Windows implica generalmente mantener los archivos de registro localmente en la maquina que origina los eventos, y requiere tomar una serie de medidas para proteger los mismos ante cualquier tipo de ataque. Esto puede tomar demasiado tiempo si se tiene una red de muchos nodos, como es el caso del hospital, lo cual además requiere un monitoreo periódico en cada uno de ellos para detectar la existencia de ataques que alteren el registro de eventos en el nodo.

También existen otras herramientas como la biblioteca Syslog [13], que puede utilizarse tanto en Linux como en Windows pero esta tampoco satisface los requisitos especificados por el DPI (ver sección 6.2) porque la información almacenada en cada registro puede resultar insuficiente para disponer de una auditoría correcta debido a la limitante que impone el protocolo Syslog en el tamaño del mensaje de auditoría, lo cual podría impedir almacenar en él toda la información necesaria para identificar correctamente un evento.

1.2 Objetivos

A continuación se detallan los objetivos planteados para cada uno de los frentes del proyecto.

Sistema de Respaldo

Existen múltiples aplicaciones, tanto pagas como gratuitas, que se especializan en el respaldo y recuperación de información.

El objetivo es realizar un estudio de dichos programas, de sus características, sus fortalezas y debilidades para así brindarles a los clientes una herramienta que se adecue a sus necesidades, en particular, la posibilidad de unificar las tareas de respaldos, facilitar su administración y les brinde seguridad en el manejo de la información que gestionan. También elaborar un manual donde se explicará cómo se utiliza la interfaz del programa seleccionado.

A su vez, se pretende crear un ambiente en el cual se instalará y ejecutará la herramienta seleccionada. El objetivo de este punto es facilitarle al cliente la puesta en producción de la herramienta, brindándole como referencia un ambiente correctamente configurado.

Sistema de Registros

El relevamiento de requisitos llevado a cabo en el Hospital de Clínicas (ver sección 6.2) determinó que en la actualidad no existe una herramienta que cumpla completamente con los mismos y con el perfil de seguridad ATNA[5] (Audit Trail and Node Authentication), a pesar de ello, los programas desarrollados en el Hospital fueron concebidos para ser integrados con este tipo de herramientas. El proyecto tiene como uno de sus objetivos diseñar y construir una aplicación de registro de eventos que cumpla tanto con los requerimientos del Hospital como con el perfil de seguridad ATNA.

El sistema debe, en base a los requisitos especificados por el Hospital, proveer un único punto de acceso mediante un servicio web el cual recibirá eventos que se generarán en programas que se ejecutan en las máquinas de la institución. Es importante aclarar que la generación de eventos con el formato especificado por ATNA y el envío de esta información no están implementados en los sistemas del hospital pero lo estarán a la brevedad.

A su vez, el Sistema de Registros deberá procesar la información recibida y guardarla en almacenamiento secundario para posterior auditoría. Por otro lado, la información también deberá enviarse a un servidor Syslog como mecanismo de respaldo.

Como requisitos principales del producto se definieron los siguientes:

- Único punto de acceso al sistema.
- Seguridad en el punto de acceso.
- Disponibilidad, performance y portabilidad.
- Interoperabilidad con el resto de los sistemas existentes.

Es deseable además considerar en el diseño, las tendencias y estándares existentes en las aplicaciones Web actuales, teniendo en cuenta la posible integración con otras plataformas y la evolución futura de la aplicación.

1.3 Resultados y aportes del proyecto

Sistema de Respaldo

En cuanto a los aportes logrados se destacan:

- **Elaboración de un Marco teórico del tema de respaldos**
  Se provee un documento con el estudio de 6 herramientas de respaldos y se profundiza en los detalles de la herramienta Bacula y sus componentes.

- **Unificación de respaldos**
  En base al estudio escogimos el programa Bacula ya que tiene soporte para clientes de los dos sistemas operativos utilizados en el Hospital de Clínicas (Windows y Linux) lo que permite administrar los respaldos de todas las maquinas del hospital de forma centralizada y más organizada.

- **Simplificar las tareas de administración**
  Con la utilización de un número reducido de servidores de respaldo que permitan tener protegidas todas las computadoras de la institución la tarea de control y auditoria de los administradores se facilita en buen grado.

- **Configuración y puesta en producción de Bacula**
  Para permitir a los clientes probar el sistema elegido creamos un entorno de prueba de 2 maquinas, uno Linux, en donde se instalará Bacula y además oficiará de cliente de dicho sistema y otro Windows como cliente.
  También profundizamos sobre las interfaces disponibles para administrar Bacula y escogimos Webmin, que concede la posibilidad de definir la configuración de los componentes, permite a los usuarios restaurar datos de sus respaldos y configurar tareas de respaldo y restauración.

Sistema de Registros

Brinda el primer desarrollo, dentro del Hospital de Clínicas, de un sistema orientado a cumplir con las especificaciones ATNA, ofreciendo un servicio de registro y auditoría de eventos para todos los sistemas involucrados en la institución.

Entre sus características destacamos:

- **Adaptabilidad**
  La plataforma no está restringida a un dominio particular. El sistema tiene el objetivo de operar no solo en el Hospital de Clínicas, sino que se diseñó para que funcione en todos los contextos que respeten el perfil ATNA y que cuente con la capacidad de interpretar correctamente los mensajes provenientes de los mismos.
  Se puede adaptar sin la necesidad de realizar cambios a la infraestructura, basta con cargar los roles, grupos de eventos y eventos en la base de datos utilizando la sección de administración de la interfaz web para cargar los XML con dicha información.

- **Extensibilidad**
  La solución fue desarrollada con diseño e implementación modular, lo que permite que pueda ser fácilmente extensible a nuevas funciones.
  Módulos que lo componen:

  - Sistema de Registro de Eventos mediante mensajes de auditoría ATNA, con seguridad a nivel de servicio web.
  - Sistema de Auditoría de Eventos con seguridad integrada para el ingreso de usuarios, y manejo de roles para permitir/denegar acceso a las funcionalidades ofrecidas.
  - Módulo generador y validador de mensajes de auditoría en formato XML [21] respetando la especificación introducida por ATNA. El mismo podrá ser utilizado por los sistemas existentes para generar los mensajes de auditoría a enviar al Sistema de Registro de Eventos.
  - Módulo integrable a sistemas desarrollados con la herramienta Genexus sobre el lenguaje java para brindarles a estos la posibilidad de comunicarse con el servicio web para registro de mensajes ATNA utilizando los mecanismos definidos por WS-Security.

  En el Capítulo 12 se mencionan algunas de las posibles extensiones.

- **Utilización de tecnologías y estándares actuales**

  El proyecto integra diversas tecnologías y estándares (SOA [19], WS-Security [10], ATNA, Syslog [13], etc.) y encara un tema fundamental para los entornos de salud como es el control de las acciones en los programas que se utilizan en el Hospital, así como la manipulación de la información dentro de los mismos, adecuándose a los estándares que son establecidos para este sector.

- **Usabilidad**

  Cualquier sistema que cumpla con las características de seguridad que impone el perfil ATNA puede registrar eventos mediante la utilización del web service que provee la aplicación.

- **Extensa documentación**

  Se entiende que la documentación elaborada, las guías de implementación e implantación son detalladas y de gran utilidad para la comprensión y puesta en producción del producto.

Software de Registros

- Implantación y testeo en el Hospital de Clínicas, de todas las funcionalidades de cada uno de los sistemas desarrollados en el marco de este proyecto (ver sección 6.3), para poder asegurar su puesta en producción en el corto plazo.
- Estudios de estándares y perfiles a nivel de la salud, especialmente el perfil ATNA de IHE.

1.4 Público Objetivo

Este proyecto tiene como publico objetivo a:

- Personal del DPI Hospital de Clínicas.
- Otros profesionales en el área de la salud con conocimientos similares.
- Administradores de sistemas en general.

1.5 Conocimientos Previos

Se recomienda tener conocimientos básicos sobre:

- Redes de computadores y seguridad en las mismas.
- Tecnología de servicios web.
- Conceptos de respaldo de la información.
- Arquitecturas distribuidas, principalmente Java EE 5.

1.6 Organización del documento

El documento se encuentra organizado en capítulos como se especifica a continuación:

- **Capítulo 2, Marco Conceptual**
  Presenta los conceptos y estándares investigados a lo largo del proyecto.

- **Capítulo 3, Estado del Arte - Sistemas de Respaldos**
  Presenta un estado del arte de algunas herramientas de respaldo.

- **Capítulo 4, Estado del Arte - Sistemas de Registro de Eventos**
  Presenta un estado del arte de las principales herramientas de registro de eventos y de envío de mensajes de auditoría según el perfil de seguridad ATNA.

- **Capítulo 5, Planteo de la solución propuesta para el Sistema de Respaldos**
  Describe la arquitectura del programa de respaldos escogido.

- **Capítulo 6, Planteo de la solución propuesta para el Sistema de Registros**
  Describe las características de la solución propuesta.

- **Capítulo 7, Diseño e Implementación**
  Detalla los aspectos del diseño y arquitectura de las aplicaciones desarrolladas, así como las tecnologías y decisiones tomadas para la implementación.

- **Capítulo 8, Caso de estudio**
  Presenta un caso de estudio construido con el objetivo de validar la solución planteada

- **Capítulo 9, Casos de prueba Sistema de Respaldos**
  Presenta un ejemplo con el cual probamos al sistema Bacula para comprobar que el respaldo y recuperación de información funcione correctamente.

- **Capítulo 10, Casos de prueba Sistema de Registros**

Presenta las pruebas de rendimiento de las aplicaciones desarrolladas con el fin de validar los requerimientos de performance de las mismas.

- **Capítulo 11, Conclusiones**

Detalla las conclusiones obtenidas luego del estudio e implementación de la solución.

- **Capítulo 12, Trabajos a futuro**

Presenta los posibles trabajos a futuro que pueden realizarse como extensión y mejora de la solución.

- **Capítulo 13, Gestión del Proyecto**

Describe las actividades realizadas y el esfuerzo insumido en cada una. A su vez se compara el cronograma inicial y el cronograma real del proyecto, identificando los motivos por el cual difieren.

- **Capítulo 14, Referencias**

Contiene las referencias consultadas a lo largo del proyecto.

- **Capítulo 15 al 18, Anexos**

Incluye referencias a documentación generada durante el proyecto, como manuales, configuraciones, etc.

- **Capítulo 19, Glosario**

Contiene los términos, conceptos, siglas y abreviaturas mencionadas en este documento.

- **Capítulo 20, Herramientas Utilizadas**

Contiene las herramientas utilizadas en el proyecto y sus correspondientes versiones.

2 Marco Conceptual

En este capítulo se describen los conceptos, tecnologías y estándares abordados a lo largo del proyecto con el objetivo de brindar las nociones claves para comprender la solución planteada. En primer lugar se presenta HL7 e IHE, organizaciones que cooperan en la mejora de la interoperabilidad entre sistemas así como la seguridad de los mismos. Dentro de IHE se pone el foco en el perfil de seguridad ATNA, el cual será tomado como guía para implementar el registro de eventos. Por último se presentan los conceptos de arquitectura SOA y servicios web los cuales serán tenidos en cuenta a la hora de diseñar el sistema de registro de eventos.

2.1 HL7

HL7 es una organización internacional que tiene como principal objetivo el desarrollo de estándares y frameworks destinados a mejorar la interoperabilidad, el intercambio y la compartición de información hospitalaria en formato electrónico.

HL7 provee pautas que están enfocadas a facilitar y optimizar el intercambio de datos entre aplicaciones, mediante los cuales se intenta mejorar la atención en salud, perfeccionar el flujo de trabajo, reducir la ambigüedad y mejorar la interoperabilidad de los sistemas de información en el entorno de la salud.

Algunos de los estándares más reconocidos son:

- Clinical Document Architecture (CDA)
- Messaging Standard Version 2.3.1
- HL7 Version 3 Standard

La sociedad de estándares en esta área para Uruguay, SUEIIDISS (Sociedad Uruguaya de Estandarización, Intercambio e Integración de Datos e Información de Servicios de Salud), recomienda la utilización de la versión 3 del estándar.

Se destaca la participación de HL7 en diversos proyectos vinculados con la mejora de la comunicación entre sistemas en el contexto hospitalario, entre ellos el proyecto que junto con otras organizaciones como IHE [4] y DICOM [3] e IETF (Internet Engineering Task Force) permitió definir el formato XML para el mensaje de auditoría a utilizar en el registro de eventos en redes de computadores (Ver Sección 2.4).

2.2 DICOM

DICOM (Digital Imaging and Communications in Medicine) es el estándar que define los métodos para la transferencia de imágenes médicas y la información asociada a ellas, entre equipos de imagenología y sistemas de fabricantes distintos.

El objetivo principal del Comité de Estándares DICOM es crear y mantener estándares internacionales para la comunicación de información biomédica diagnóstica y terapéutica para las disciplinas que utilizan imágenes digitales y datos relacionados. Como resultado, DICOM es utilizado por prácticamente cualquier disciplina médica que utilice imágenes dentro de la industria del cuidado de la salud.

El estándar DICOM 3.0 fue el resultado de varios estándares desarrollados por ACR (American College of Radiology) [82] y NEMA (National Electrical Manufacturers Association) que en 1983 formaron un comité para desarrollar un estándar que cumpliera los siguientes propósitos:

- Promover la comunicación de imágenes digitales, independientemente del fabricante del equipo.
- Facilitar el desarrollo y expansión de los sistemas de almacenamiento y comunicación de imágenes (PACS) capaces de comunicarse también con otros sistemas de información hospitalaria.
- Permitir la creación de bases de datos de información diagnóstica que pudiesen ser consultadas por una amplia variedad de dispositivos remotos.

El estándar DICOM se complementa con un suplemento, denominado Supplement 95: Audit Trail Messages [34] que describe la forma en que las entidades DICOM deben enviar la información de los eventos generados en sus sistemas a una aplicación de registro con el objetivo de simplificar la recopilación de información de sucesos en lugar de extraer información de auditoría de cada nodo.

Vinculado al marco de este proyecto, la importancia de DICOM radica en la definición del vocabulario utilizado en el mensaje de auditoría para el registro de eventos publicado en la RFC 3881 (ver sección 2.4).

2.3 IHE

La iniciativa IHE (Integrating the Healthcare Enterprise) nace como esfuerzo común de la RSNA (Radiological Society of North America) y la HIMSS (Healthcare Information and Management Systems Society), a la que se unió posteriormente el ACC (American Collage of Cardiology) y tiene como finalidad dar respuesta a las dificultades que tienen los diferentes sistemas de información del ámbito médico para comunicarse entre sí, aún cuando estos sean conformes a estándares de la categoría de DICOM y HL7.

IHE trata de establecer qué estándares son recomendables utilizar en circunstancias específicas, construyendo lo que se denominan “Technical Frameworks”. Estos marcos de trabajo se representan mediante un grupo detallado de documentos que se publican de forman anual que guían a los desarrolladores e integradores de sistemas de información e imagenología para:

- alcanzar una efectiva integración entre estos sistemas.
- facilitar mecanismos apropiados para intercambiar y compartir información del área.
- brindar soporte para un óptimo seguimiento del paciente.

Por otro lado, IHE elabora diversos perfiles y los agrupa en categorías según su tipo, siendo muchos de ellos aún versiones no completamente estables, sujetas a posibles variaciones. Cada una de estas categorías tiene asociado un “Technical Framework” que define las implementaciones existentes de los estándares establecidos, para lograr los objetivos propuestos. Algunas de estas categorías son:

- Perfiles de Cardiología.
- Perfiles de Laboratorio.
- Perfiles de Infraestructura IT.
- Perfiles de Radiología.

Los perfiles de infraestructura IT, más precisamente el perfil ATNA, es de interés para este proyecto ya que proporciona una infraestructura básica sobre la cual compartir información e implementar las políticas de seguridad de cada organización. Además, describe una forma de autenticación de los actores (sistemas) usando certificados, y de transmisión de eventos relacionados con la información personal a un repositorio para auditoría.

El objetivo principal es mejorar la forma en que los sistemas se comunican y comparten información. Esto se logra si se siguen los lineamientos detallados en los estándares recomendados por el perfil.

De la terminología utilizada en los Perfiles de Integración se destacan:

- **Actor:** Son sistemas de información o componentes de un sistema de información que producen, gestionan y actúan sobre la información asociada a actividades operativas de la empresa de salud. Cada uno de los actores reciben un nombre específico que los identifica de forma no ambigua.
- **Transacción:** Son interacciones entre actores que transfieren la información requerida a través de la utilización de mensajes basados en estándares.

A modo de ejemplo, para el perfil ATNA (ver Figura 1) se destacan los siguientes actores:

- Servidor de tiempo.
- Repositorio de Auditoría.

A su vez, las principales transacciones son:

- Autenticación de nodos.
- Registrar evento de auditoría.

2.4 Audit Trail and Node Authentication (ATNA)

ATNA [5] es un perfil de integración definido por IHE, en el cual se detallan medidas de seguridad básicas que deben implementar los nodos de una red perteneciente a una institución hospitalaria, así como las garantías que deben brindar para ser utilizados como parte de un ambiente médico seguro y privado. El objetivo de ATNA es asistir a los administradores de cada sistema a implementar las políticas de seguridad y confidencialidad necesarias. Los sistemas involucrados son aquellos sistemas de información en el ámbito de la salud, que manejan o procesan información protegida PHI (Protected Health Information).

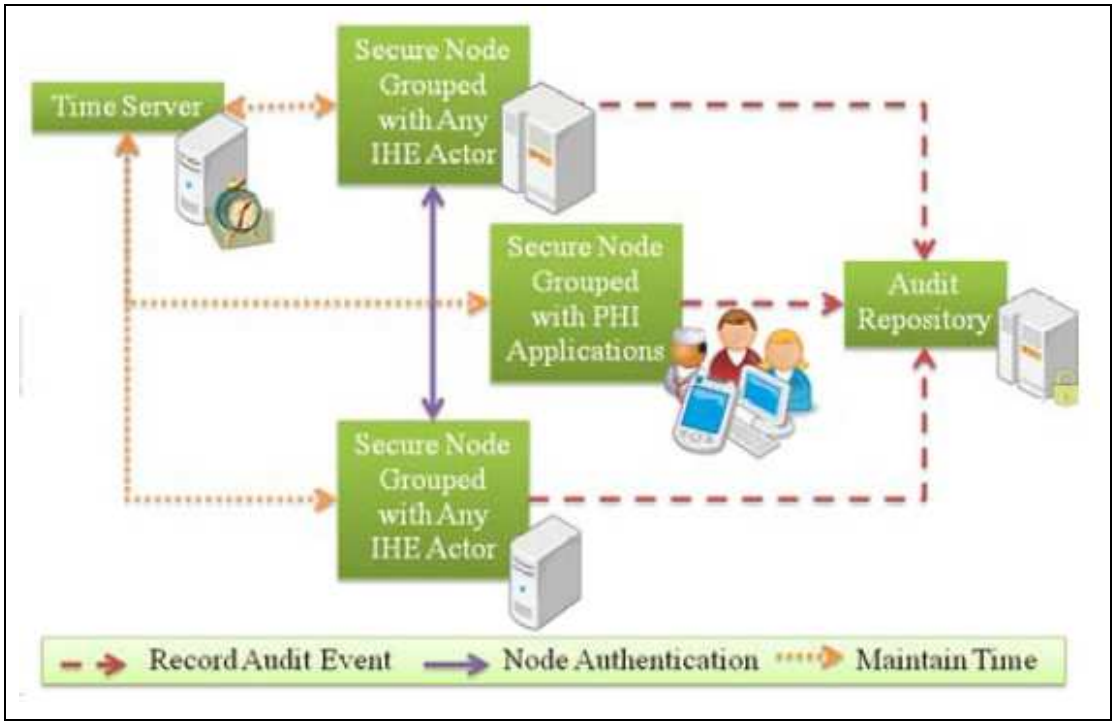


Figura 1 - Actores y transacciones del perfil ATNA [5]

La Figura 1 presenta los actores involucrados en la implementación del perfil ATNA. “Time Server” es el servidor de tiempo, que se utiliza para sincronizar todos los nodos de la red. ATNA requiere la implementación del perfil “Consistent Time”, el cual asegura una diferencia de a lo sumo un segundo en la hora de los nodos. Luego se encuentran una serie de grupos de nodos, todos ellos asegurados según el perfil, los cuales pueden comunicarse mutuamente. Por último se dispone de un repositorio de eventos, en el cual los nodos registrarán aquellos que consideren de importancia.

El perfil ATNA se construye en base a los siguientes estándares:

- WS-I Basic Security Profile 1.0 [6].

- RFC 5424 (Protocolo Syslog) [14].
- RFC 5425 (Transmisión de mensajes Syslog sobre TLS) [81].
- RFC 5426 (Transmisión de mensajes Syslog sobre UDP) [16].
- RFC 3164 (Protocolo BSD Syslog) [13].
- RFC 3881 (Definiciones para mensaje de auditoría) [2].
- RFC 5246 (Transport Layer Security 1.0) [35].

Los sistemas que implementan el perfil tendrán las siguientes características:

- Autenticación de usuario. ATNA solo requiere autenticación de usuario local, permitiendo utilizar la tecnología de control de acceso que el nodo defina, por ejemplo EUA [23] (Enterprise User Authentication) y XUA [24] (Cross-Enterprise User Assertion) son candidatos.
- Control de acceso entre nodos de una red limitando el acceso a nodos únicamente a usuarios autorizados.
- Autenticación bidireccional de nodos en la comunicación entre ambos, mediante la utilización de certificados digitales.
- Integridad y confidencialidad de los datos intercambiados entre los nodos.
- Seguimiento de usuarios para determinar acciones mal intencionadas sobre la información protegida (PHI, ver Glosario).
- TCP/IP Transport Layer Security Protocol (TLS) para la autenticación del nodo y cifrado (opcional). Si se dispone de comunicaciones entre servicios web, ATNA permite la implementación de mecanismos de seguridad punto a punto definidos en el estándar WS-Security que cumplan con el perfil básico WS-I.
- El actor debe ser capaz de configurar la lista de certificados de nodos autorizados.
- El sistema deberá registrar eventos de aplicación en un repositorio centralizado.
- La comunicación con el repositorio de eventos podrá ser mediante el protocolo Syslog sobre UDP. Dado que el avance en la implementación de servidores Syslog confiables es lenta, se debe utilizar BSD Syslog.

En el marco de este proyecto se siguieron estos lineamientos para lograr la autenticación entre los nodos del Hospital de Clínicas y los sistemas de Registro de Eventos y de Auditoría (ver capítulo 7).

A continuación se presentan los principales formatos de registro y mensaje de auditoría, de los cuales se seleccionará uno para realizar la implementación del repositorio de eventos:

Formato de Registro de Auditoría [5]

IHE define varios tipos de formato para el mensaje de auditoría, todos ellos utilizando XML Schema. Los registros de auditoría son aquellos que contendrán toda la información necesaria y serán almacenados en un repositorio de auditoría. En la actualidad se utilizan dos esquemas para el registro de auditoría:

1. **Formato provisional de IHE**
   Previamente definido como parte del framework técnico de radiología IHE. Su uso es obsoleto en el sentido que no existirán más extensiones disponibles, por lo tanto las aplicaciones deberán comenzar a utilizar el formato IHE Audit Trail.

2. **Formato IHE Audit Trail**
   Está basado en los estándares desarrollados por IETF, HL7 y DICOM para cumplir con las necesidades de auditoría medica.

Formato de Mensaje de Auditoría [5]

Se especifican dos formatos de mensajes de auditoría:

1. **Formato RFC 3881**

   Formato definido conjuntamente por IHE, HL7, DICOM, ASTM, E31, y el “Joint NEMA/COCIR/JIRA Security and Privacy Committee”.

   El framework de infraestructura técnica de IHE recomienda la utilización de este esquema para los mensajes de auditoría generado por todos los actores IHE.

   Por otro lado el estándar DICOM Suplemento 95 [34] provee vocabulario y especificación de uso de los elementos de este esquema, para eventos que pueden ocurrir en sistemas DICOM. IHE evaluó y determinó que dicho suplemento es más aplicable, extendiéndolo para uso general.

2. **IHE Audit Trail**

   El estándar DICOM no contempla todos los tipos de evento que pueden ocurrir en el contexto de la salud por lo que IHE definió eventos adicionales que podrán ser

utilizados en caso que los de DICOM no sean suficientes. Los mensajes deberán respetar el formato RFC 3881.

Ambos formatos son mensajes codificados en XML, lo que permite utilizar las extensiones de los mecanismos de extensión XML estándar.
Para el desarrollo de este proyecto se siguieron los lineamientos recomendados por IETF en la RFC 3881.

Mensaje de Auditoría RFC 3881

A continuación se describen los tipos de datos que se utilizan y luego la estructura y contenido del mensaje de auditoría que será utilizado.
Básicamente un mensaje de auditoría ATNA se compone de un elemento raíz denominado AuditMessage el cual contiene los elementos representados en la Figura 2:

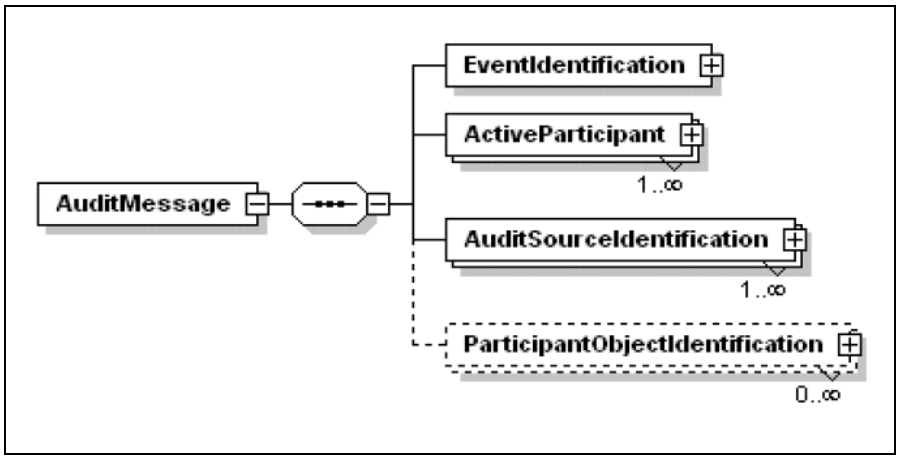


Figura 2 - Elementos del mensaje de auditoría definido en la RFC 3881

Como indica la Figura 2, el mensaje de auditoría contiene un único elemento con la identificación del evento (EventIdentification), un único elemento con la información del usuario (ActiveParticipant), un conjunto de orígenes (AuditSourceIdentification) y un conjunto opcional de objetos (ParticipantObjectIdentification).

A continuación se describen los cuatro elementos fundamentales del mensaje ATNA:

Event Identification

Contiene los campos necesarios para identificar un evento, entre otros: identificador, tipo y fecha de generación.

Active Participant

Define la información necesaria para auditar de manera eficaz quienes son los usuarios involucrados en la ejecución de los eventos.

Audit Source Identification

Es el conjunto de datos que describe las fuentes involucradas en la generación de los eventos a auditar.

Participant Object Identification

Asiste a los sistemas de auditoría indicando aquellas instancias y objetos que han sido accedidos durante el evento.

Dado que pueden existir varios objetos accedidos dentro de un mismo evento, este elemento es multivaluado. Para la posterior auditoría, este elemento es importante ya que por ejemplo permite el seguimiento de todos los datos del paciente ya sea por número de registro médico, por documento de identidad, etc.

Para concluir la descripción del mensaje de auditoría la Figura 3 resume gráficamente los principales componentes del mismo, así como los campos dentro de cada uno de ellos:

```
<?xml version="1.0" encoding="UTF-8"?>
<AuditMessage>
  <EventIdentification EventActionCode="C" EventDateTime="2010-05-18T00:19:01" EventOutcomeIndicator="0">
    <EventID code="03" codeSystem="OID HC" codeSystemName="Grupo Eventos del HC"
             displayName="Historia Clínica" originalText="" />
    <EventTypeCode code="10" codeSystem="OID HC" codeSystemName="Grupo Eventos del HC"
                   displayName="Creación de Estudio" originalText="" />
  </EventIdentification>
  <ActiveParticipant UserID="Y.YYY.YYY-Y" AlternativeUserID="Jose Perez" UserName="jose"
                     UserIsRequestor="true" NetworkAccessPointID="192.168.1.2" NetworkAccessPointTypeCode="2">
    <RoleIDCode code="3" codeSystem="OID HC" codeSystemName="Roles del HC" displayName="Doctor" />
  </ActiveParticipant>
  <AuditSourceIdentification AuditSourceID="Pacientes para Atender" AuditEnterpriseSiteID="OID HC">
    <AuditSourceTypeCode code="8" />
  </AuditSourceIdentification>
  <ParticipantObjectIdentification ParticipantObjectID="4.584.507-7" ParticipantObjectTypeCode="1"
                                   ParticipantObjectTypeCodeRole="1" ParticipantObjectDataLifeCycle="1">
    <ParticipantObjectIDTypeCode code="11" codeSystem="DNIC" codeSystemName="CI" />
    <ParticipantObjectName>Martin Calabria</ParticipantObjectName>
  </ParticipantObjectIdentification>
</AuditMessage>
```

Figura 3 – Ejemplo de mensaje de auditoría RFC 3881

2.5 *Service Oriented Architecture (SOA)*

Service Oriented Architecture [19] (SOA), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.
Permite la creación de sistemas altamente escalables y brinda una forma bien definida para desplegar e invocar servicios (normalmente servicios web, pero no excluyente), lo cual facilita la interacción entre diferentes sistemas.
A diferencia con las arquitecturas orientadas a objetos, las arquitecturas SOA están formadas por servicios de aplicación débilmente acoplados y altamente operables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma en que son desplegados y del lenguaje de programación.
La definición de la interfaz encapsula las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo.
Con esta arquitectura, se pretende que los componentes de software desarrollados sean reutilizables y fácilmente extensibles.

Si bien SOA no es una tecnología, sino un paradigma de diseño, existen una serie de estándares (XML, SOAP, WSDL, HTTP, UDDI) y productos enfocados a la implantación de esta arquitectura, como por ejemplo ESB (Enterprise Service Bus), Registro de Servicios y plataformas orientadas a arquitecturas distribuidas (caso Java EE [36])

En el marco de este proyecto se diseño una arquitectura SOA para el sistema de registro de eventos, siendo la capa de servicios del mismo quien expone las funcionalidades mediante servicios web. La plataforma sobre la cual se construyó el sistema es Java EE ya que como se mencionaba anteriormente, permite diseñar aplicaciones distribuidas en tantas capas como sea necesario.

Como consecuencia de la utilización de esta arquitectura, el Sistema de Registro de Eventos obtendrá los siguientes beneficios:

- Flexibilidad a la hora de incorporar nuevas tecnologías
- Estándarización la implementación de cada capa y la comunicación entre las mismas
- Reutilización de capas gracias al desacoplamiento de las mismas
- Mantenibilidad y desarrollo de capas en paralelo
- Simplifica la adaptación de sistemas existentes

2.6 Servicios Web

Un servicio web es un conjunto de protocolos y estándares que permiten comunicar distintas aplicaciones de software para el intercambio de datos, independientemente de los aspectos tecnológicos de las mismas y de las plataformas sobre la cual se encuentran instaladas. Esta interoperabilidad se logra mediante la adopción de estándares abiertos, de amplia utilización, como es el caso de XML, SOAP y WSDL. Por otro lado existe un organismo denominado WS-I [6] (Web Services Interoperability Organization) especialmente creado para fomentar y mejorar continuamente la interoperabilidad entre distintas implementaciones de servicios web, plataformas y lenguajes de programación, integrando estándares y definiendo perfiles.

Las organizaciones OASIS (Organization for the Advancement of Structured Information Standards) y W3C (Word Wide Web Consortium) son los comités responsables de la arquitectura y reglamentación de los servicios web.

W3C posee en la actualidad un grupo de investigación especializado en servicios web, lo que demuestra la importancia que tiene dicho concepto dentro de estos últimos años. Este grupo de trabajo ha entregado una definición formal acerca de los Servicios web, la que se presenta a continuación:

Los servicios web son muy utilizados para implementar sistemas distribuidos y orientados a servicios, por lo tanto es una buena elección para implementar una arquitectura SOA.
A continuación se presentan los estándares y protocolos recomendados para la exposición y consumición de servicios web:

- Un formato que describa la interfaz del componente (sus métodos y atributos) basado en XML. Por lo general este formato es el WSDL (Web Service Description Language).
- Un protocolo de aplicación basado en mensajes y que permite que una aplicación interactúe con el Web Service. Por lo general este protocolo es SOAP (Simple Object Access Protocol).
- Un protocolo de transporte que se encargue de transportar los mensajes por Internet. Por lo general este protocolo es el HTTP (Hiper-Text Transport Protocol).

Las principales ventajas de los servicios web son:

- Interoperabilidad entre aplicaciones de software independiente de la plataforma.

- Fomentan estándares y protocolos basados en texto, los cuales brindan facilidades para acceder a su contenido y entender su funcionamiento.
- El apoyo en HTTP permite a los servicios web aprovecharse de los sistemas de seguridad (firewall) sin necesidad de cambiar las reglas de filtrado.

En el marco de este proyecto, los servicios web jugarán un papel importante ya que serán los componentes utilizados para implementar la arquitectura SOA del sistema de registros, siendo estos la puerta de acceso por el cual los sistemas existentes enviarán sus eventos de auditoría.

Dado los requerimientos de seguridad establecidos por el perfil ATNA, es necesario incorporar mecanismos que permitan disponer de autenticación del nodo emisor, así como mantener la confiabilidad e integridad de los datos enviados al servicio web. Está problemática la resuelve el estándar WS-Security, el cual se describe a continuación.

WS-Security

WS-Security [10] es un add-on para SOAP que describe la forma en que la cabecera de un mensaje SOAP puede ser usado para incluir información sobre la seguridad y proporcionar confidencialidad, integridad y no repudio entre otros.

La seguridad de los servicios Web puede aplicarse en tres niveles distintos:

- Seguridad (de punto a punto) para plataformas/transporte.
- Seguridad (personalizada) para aplicaciones.
- Seguridad (de extremo a extremo) para mensajería.

Cada enfoque ofrece una serie de beneficios y desventajas, que se detallan a continuación. La elección del enfoque depende en gran medida de las características de la arquitectura y las plataformas que vayan a utilizarse para el intercambio de los mensajes.

Seguridad (punto a punto) para plataformas/transporte

Puede utilizarse el canal de transporte entre dos puntos finales (cliente de servicios Web y servicios Web) para garantizar la seguridad de punto a punto. La Figura 4 refleja este escenario.

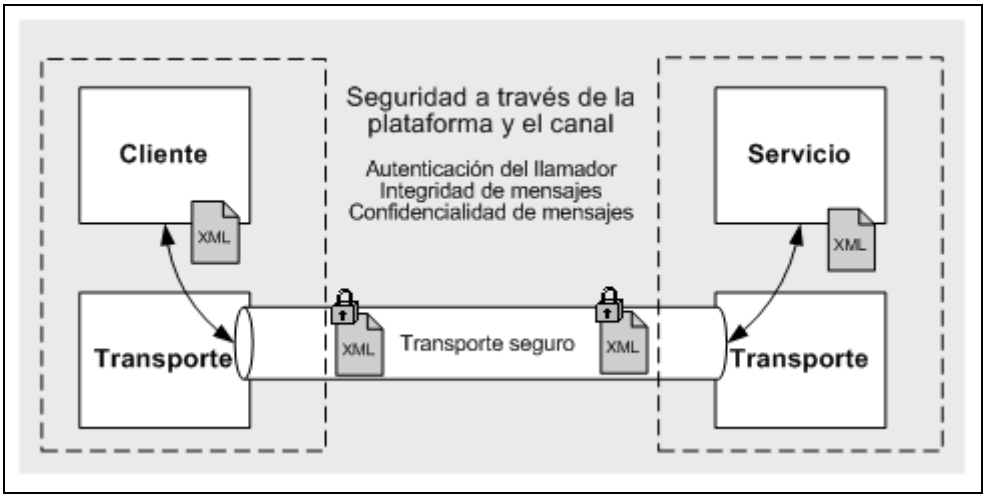


Figura 4 - Seguridad para plataformas/transporte [10-2]

El modelo de seguridad para transporte es sencillo, bien diseñado y adecuado para una gran variedad de escenarios (básicamente para intranets), en los que los mecanismos de transporte y la configuración del punto final pueden controlarse de forma exhaustiva.

Los principales aspectos que deben tenerse en cuenta con relación a la seguridad del transporte son los siguientes:

- La seguridad se integra estrechamente con la plataforma subyacente, el mecanismo de transporte y el proveedor de servicios de seguridad (NTLM, Kerberos, etc.), de los que también depende.
- La seguridad se aplica de punto a punto, sin crear provisiones para saltos múltiples ni enrutamiento a través de nodos de aplicación intermedios.

Seguridad (de extremo a extremo) para mensajería

Se trata del enfoque más flexible y eficaz y el que utiliza la iniciativa GXA (Global XML Web Services Architecture), en especial para la especificación de seguridad WS-Security. La seguridad de mensajes queda bosquejada en la Figura 5:

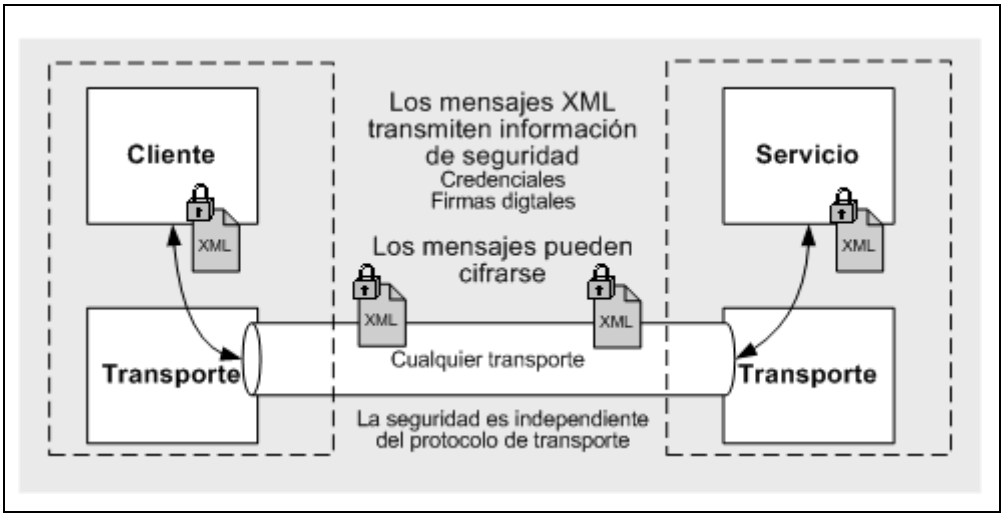


Figura 5 - Seguridad de mensajes [10-2]

A continuación se presentan dos mecanismos de seguridad definidos en el estándar WS-Security que serán tenidos en cuenta a la hora de implementar los servicios web:

UsernameToken Authentication

Permite al consumidor de servicios web suministrar un nombre de usuario y, opcionalmente, una contraseña, para poder identificar al solicitante y así autenticarse ante el proveedor del servicio web.

Para brindar mayor seguridad la contraseña puede ser cifrada mediante un algoritmo de hash, como por ejemplo MD5.

La especificación detallada de su utilización puede ser consultada en OASIS bajo el nombre “Web Services Security UsernameToken Profile” [32].

X.509 Mutual Certificate Authentication

Este protocolo involucra 2 actores: un cliente y un servidor, ambos teniendo asociado un par de claves RSA (pública y privada). Los certificados X.509 deben intercambiarse entre ambos para que cada uno pueda verificar al otro. El objetivo es intercambiar dos mensajes XML: una solicitud y una respuesta, de manera que tanto el cliente como el servidor puedan autenticarse en la sesión, manteniendo los mensajes en secreto, incluso en la presencia de un atacante activo. Para llevar a cabo esto, se utiliza la firma digital XML y el cifrado XML. El cliente envía el mensaje de petición, firmando el mismo con el algoritmo RSA-SHA1. Luego, para mantener el mensaje en secreto, lo cifra mediante el algoritmo AES utilizando una clave simétrica generada para la sesión. Esta clave también es cifrada, con el algoritmo RSA. Una vez que el servidor recibe la petición, la procesa y retorna un mensaje de respuesta al cliente. Al igual que el mensaje de petición, el mensaje de respuesta es firmado

y luego cifrado utilizando una nueva clave simétrica. Para relacionar peticiones y respuestas, el servidor firma tanto el mensaje de respuesta como el valor de la firma del mensaje de petición. Esto previene que un atacante pueda confundir al cliente intercambiando dos respuestas.

La especificación detallada de la utilización puede ser consultada en OASIS bajo el nombre “Web Services Security X.509 Certificate Token Profile” [42].

2.7 Syslog

Syslog [13] es un protocolo estándar para el envío de mensajes de log en una red de computadores IP. El protocolo es de tipo cliente-servidor por lo tanto en una maquina servidor ejecutará el denominado demonio Syslog, aceptando mensajes generalmente en el puerto 514 aunque esto puede ser modificado. A su vez, cualquier aplicación puede actuar como cliente del servidor Syslog, en cuyo caso para interactuar con éste deberá respetar el formato de mensaje que define Syslog.

Protocolo

El protocolo aplica los siguientes principios:

- No responde un ack cuando el mensaje es enviado. A pesar de que algunos medios de transporte pueden proveer la información del estado, Syslog es un protocolo de comunicación simplex puro.
- Los generadores y transmisores pueden ser configurados para enviar los mensajes a múltiples receptores u otros repetidores.
- Generadores, transmisores y receptores pueden residir en el mismo sistema.

El protocolo no especifica nada acerca de la capa de transporte, debido a que el mensaje Syslog es independiente del medio de transporte utilizado aunque en el Standard (RFC 5426 [16]) se define un medio de transporte, el cual es consistente con el tradicional formato UDP, el cual ha sido históricamente el utilizado para el transporte de mensajes Syslog.

Bajo ningún motivo, el medio de transporte debe modificar el mensaje y si por alguna razón el protocolo de transporte realiza algún cambio temporal, este debe ser revertido en el momento de la recepción.

Los servidores, demonios o servicios de Syslog generalmente ponen a la escucha un socket en UDP y puerto 514, aunque existen implementaciones que brindan transporte confiable mediante el protocolo TCP.

El perfil ATNA menciona entre otras, a Sylog como una de las posibles implementaciones para el registro de eventos a un repositorio centralizado, pero deben tenerse en cuenta los siguientes puntos sobre el mismo:

- No posee un mecanismo de autenticación del nodo emisor

- Tamaño de mensaje acotado a un máximo de 1KB.
- No posee seguridad en el transporte, que permita la confidencialidad e integridad de los datos enviados
- Está diseñado para capturar eventos de sistema no de aplicación

Analizando estos puntos se llegó a la conclusión de no utilizar Syslog para la implementación del registro de eventos. Sin embargo, será utilizado como mecanismo de respaldo de los eventos recibidos, de modo de disponer en todo momento de una copia de los mismos. Algunos aspectos del protocolo serán tenidos en cuenta para la implementación. A modo de ejemplo, el tipo de un evento será mapeado a una severidad y facilidad del protocolo Syslog. A partir de estos dos se calculará la prioridad de cada evento, la cual será importante para la toma de decisiones.

3 Estado del Arte Sistema de Respaldos

Por la naturaleza del sector en el que operan los responsables del DPI del Hospital de Clínicas, mantener la información siempre disponible y sin la más mínima interrupción es prioritario.

El respaldo de información es el mecanismo de seguridad más utilizado por los administradores del DPI para salvaguardar la información que maneja diariamente el Hospital de Clínicas, por lo que el sistema de respaldo y recuperación de la misma tiene que ser probado y eficiente, además de proveer mecanismos que les facilite la tarea de planificación y ejecución de los backups.

A continuación se brindan algunos conceptos relacionados con el respaldo de información y que se utilizarán en el estudio de las herramientas consideradas [66]:

1. **Topología:**

   Se pueden clasificar en 2 grandes grupos, centralizadas y descentralizadas.

   - Centralizadas: en ésta topología, múltiples clientes envían sus datos a un servidor de respaldo el cuál se encarga de almacenar los datos en el medio correspondiente (Figura 6).

     Entre las ventajas de esta arquitectura se destaca el menor costo en realizar la configuración y protección de los entornos de respaldos debido a que todo es controlado desde un único punto. También los costos en hardware son menores debido a que, por ejemplo, comprar un disco de 1 TB es más económico que comprar 5 de 200 GB.

   - Descentralizada: cada cliente guarda su información en los dispositivos de respaldos directamente conectados a ellos (Figura 7).

     Como ventaja se destaca que cada cliente puede restaurar sus datos independientemente sin necesitar comunicarse con el servidor.

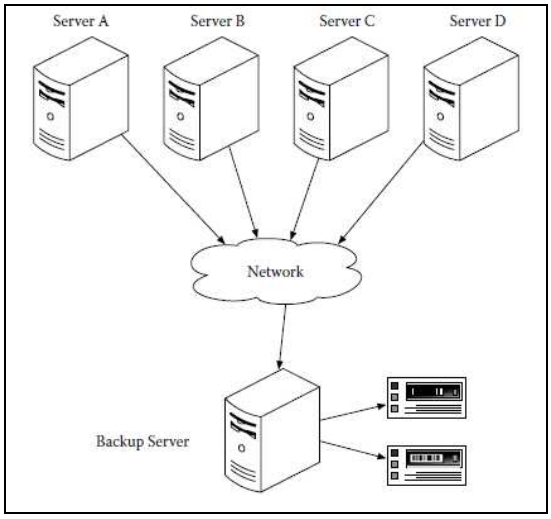


Figura 6 - Topología Centralizada

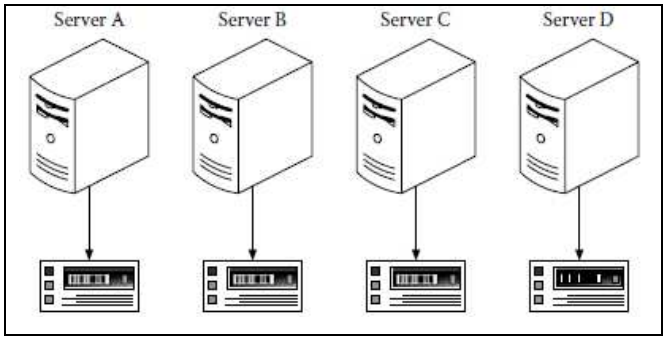


Figura 7 - Topología Descentralizada

2. Tipo:

No hay un acuerdo entre los nombres de los diferentes tipos de respaldos, ni por parte de las empresas de software, ni por parte de los administradores, pero los más frecuentes son:

- Completo (Nivel 0): Toda la información seleccionada del almacenamiento primario, tanto archivos de datos cómo meta datos (directorios, archivos, atributos, etc.) son respaldados lo que simplifica la restauración de la información para lo cual solo se necesita el dispositivo donde se realizó.

- Incremental: Solo se respaldan los archivos que fueron modificados desde el último respaldo (independientemente del tipo) lo que hace que, por lo general, el espacio necesario sea menor que el tipo anterior pero en oposición, la restauración de datos necesita tanto del respaldo actual como el del respaldo Full y todos los incrementales intermedios.

- Diferencial: Se respaldan todos los datos modificados luego del último respaldo completo por lo que no toma en cuenta si los archivos ya han sido incluidos en algún respaldo diferencial o incremental intermedio.

En base a estos tipos de respaldo surgieron otros que brindan una protección mayor de los datos como son:

- Protección continúa de datos [67]: El cual supervisa constantemente el sistema de archivos del host (en la realidad se establece una ventana de tiempo) y si se produce un cambio inmediatamente realiza el respaldo del/los archivos involucrados, por lo que puede reducir al mínimo la cantidad de información perdida en caso de que le sucediera algo al sistema.

- Sintética (synthetic backup) [68]: Una copia de seguridad sintética es igual a una copia de seguridad completa en términos de datos, pero se crea con los datos que se recopilan a partir del respaldo completo y los incrementales posteriores. El resultado final de la combinación de un reciente archivo de respaldo con la información consolidada de todos los respaldos utilizados para crearla.

  Es una alternativa muy práctica en casos donde por requisitos de tiempo o del sistema no es posible realizar una copia de seguridad completa.

Algunas estrategias de respaldos posibles:

o **Respaldo GFS (Grandfather-Father-Son)**

Es uno de los esquemas de rotación más populares en los entornos corporativos, consistente en un respaldo completo mensual al que se le llama abuelo (grandfather), un respaldo completo semanal, el padre (father), y un respaldo incremental diario, el hijo (son). La ventaja de esta metodología es que conservan respaldos recientes de la información, mientras que paralelamente se conservan también copias anteriores.

Ejemplo:

Domingo (1)	Lunes (2)	Martes (3)	Miércoles (4)	Jueves (5)	Viernes (6)	Sábado (7)
Diferencial/ Incremental o NADA	Diferencial/ Incremental	Diferencial/ Incremental	Diferencial/ Incremental	Diferencial/ Incremental	Completo	Diferencial/ Incremental o NADA
Domingo (8)	**Lunes (9)**	**Martes (10)**	**Miércoles (11)**	**Jueves (12)**	**Viernes (13)**	**Sábado (14)**
Diferencial/ Incremental o NADA	Diferencial/ Incremental	Diferencial/ Incremental	Diferencial/ Incremental	Diferencial/ Incremental	Completo	Diferencial/ Incremental o NADA

Figura 8 - Estrategia GFS (Grandfather-Father-Son).

En caso de que el sistema falle el Jueves (12):

Será necesario el Respaldo completo del Viernes (6) y

Si se utilizaron respaldos diferenciales: Sólo el Respaldo Diferencial del Miércoles (11).

Si se utilizaron respaldos incrementales: Se necesitaran todos los Respaldos Incremental desde el Sábado (7) hasta el Miércoles (11)

- **La Torre de Hanoi**

Es una la alternativa un poco más compleja al GFS, pero muy adoptada también en entornos corporativos.

El esquema se basa en el juego matemático del mismo nombre y para adaptarlo a los respaldos se consideran 5 discos etiquetados A, B, C, D, y E.

La Figura 9 muestra el patrón para el esquema de copia de seguridad. El patrón consiste en 16 días.

Nivel de copia de seguridad \ Sesión	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1 (Incremental)		A		A		A		A		A		A		A		A
2 (Diiferencial)			B				B				B				B	
3 (Diiferencial)					C								C			
4 (Diiferencial)									D							
5 (Completo)	E															

Figura 9 - Estrategia Torre de Hanoi.

Para el esquema de cinco niveles es posible recuperar los datos respaldados de hasta dos semanas atrás.

- **Duplicado de Información (RAID)**

El sistema RAID fue propuesto por primera vez en 1988 y es la sigla de Rebundant Array of Inexpensive/Independent Disk. Tiene como objetivo subsanar algunos de los problemas comunes presente en los sistemas de almacenamiento tradicionales y lograr mejoras como la tolerancia a fallos y errores, aumentar la integridad de los datos y mejorar el rendimiento.
RAID ofrece varias opciones, llamadas niveles RAID [69], cada una de las cuales proporciona un equilibrio distinto entre tolerancia a fallos, rendimiento y costo.
Todos los sistemas RAID suponen la pérdida de parte de la capacidad de almacenamiento de los discos para conseguir la redundancia o almacenar los datos de paridad.

Pero además de realizar los respaldos, es muy importante seleccionar los medios donde se realizan los mismos, los que deben cumplir con determinadas características como se detalla en el libro “Unix Backup and Recovery” Cap. 18 de W. Curtis Preston [65] donde se especifican los siguientes factores a tener en cuenta a la hora de decidir:

1. **Confiabilidad:**
   Una medida de la confiabilidad de los medios de almacenamiento es MTBF (Mean-Time-Between-Failure) que representa el tiempo aproximado entre fallas pero es obtenido en base a entornos artificiales de trabajo que intentan simular ambientes reales y por eso la mejor opción más recomendable es Internet, donde se encuentra muchas discusiones acerca del tema.

2. **Velocidad de Transferencia:**
   Es importante comparar los medios basándose en la velocidad de transferencia de datos sin comprimir que especifican los fabricantes o en el valor "head-to-tape" que representa el tiempo que se tarda en guardar la información desde que llega a la “cabeza de lectura/escritura”.
   Pero además es importante considerar también el entorno de trabajo como por ejemplo: la velocidad de transferencia de la red o la saturación de la misma y las posibilidades del sistema de respaldo para explotar las características del medio escogido.

3. **Tiempo de Acceso a los Datos:**
   Hay entornos como los de “Almacenamiento Jerárquico” o cuando se desee restaurar un archivo pequeño y no grandes volúmenes de información, en los que la “Velocidad de Transferencia” no es lo más importante, sino que juega un papel muy importante el tiempo en que se accede a la información y hay que tener muy en cuenta este factor en el momento de hacer la elección.

4. **Capacidad:**
   Suele ser uno de los factores más importantes a considerar en la decisión. No obstante depende la utilización que se le dará y de los recursos con que se cuenta debido a que si se dispone, por ejemplo, de un intercambiador de cintas, no es necesario tener un solo medio lo suficientemente grande donde pueda almacenar todo el respaldo.

5. **Costo:**
   Por lo general, los medios de almacenamiento con más capacidad y que obtiene los datos en menor tiempo son más caros pero no sólo estas características intervienen en la decisión, sino que se debe tomar en cuenta también la reusabilidad y el tiempo en que este se puede utilizar el mismo.

La realización de las copias de seguridad se debe basar en un análisis previo del sistema a respaldar y, de acuerdo a la importancia de los datos, el tamaño de los mismos, los cambios que sufren en un determinado período tiempo, disponibilidad de hardware, etc., establecer la política de respaldo adecuada.

3.1 Situación actual en el Hospital de Clínicas

La metodología actual que dispone el hospital de clínicas para realizar tareas de respaldos carece de automatización, requiriéndose la presencia de un usuario administrador, quien es el encargado de configurar y llevar a cabo los respaldos, día a día, de los archivos dentro de la red del hospital. Durante el relevamiento se detectó que en muchas ocasiones estos respaldos no se efectuaban, principalmente por olvido, y tampoco se dispone de un amplio conjunto de respaldos, aproximadamente se almacenan respaldos con una ventana de 30 días, eliminándose automáticamente aquellos que exceden dicho periodo.

Los respaldos se realizan solo sobre sistemas operativos Windows (no poseen política de respaldo para Linux), utilizando como herramienta principal las copias instantáneas. Esta

funcionalidad, también llamada Volume Shadow Copy Service (VSS), permite a los administradores realizar copias instantáneas de volúmenes de datos críticos, sin producirse interrupciones en el servicio ni incoherencias en los archivos respaldados. Esto último podría suceder si no se tuviese un adecuado control de los archivos que están siendo modificados durante el proceso de respaldo, pero VSS posee un mecanismo para controlar esta situación. Estas copias pueden ser aprovechadas para restaurar posteriormente el sistema, como simple archivado de datos o para recuperación total o parcial de los mismos.

Particularmente se utiliza la opción de instantáneas “Shadow Copies for Shared Folders”, que permite realizar respaldos de todo la información compartida en un volumen determinado y que permite, no solo a los administradores, sino también a los usuarios, recuperar sus propios archivos en caso de ser necesario de una lista de hasta 64 versiones anteriores.

En el contexto del Hospital de Clínicas, el administrador de respaldos dispone de 2 carpetas compartidas ubicadas cada una en un servidor distinto. En una de ellas, se almacenan los respaldos realizados. La necesidad de una segunda carpeta es para mantener disponibilidad en caso de que el primer servidor no estuviere disponible. La copia desde una carpeta a la otra se realiza automáticamente.

El respaldo se realiza en discos duros, aunque se tiene previsto utilizar un cargador de cintas.

No se dispone de información exacta del volumen de datos que se va a respaldar pero los clientes estimaron que se trata de alrededor de 2 GB de archivos de base de datos por cada uno de los 30 servidor que poseen y en caso de usarlo para imagenología los respaldos tendrían un volumen de entre 5TB y 10 TB.

Ventajas y Desventajas

En conjunto con los clientes evaluamos los principales pros y contras de esta herramienta donde se destacan:

Como principales ventajas se pueden encontrar las siguientes:

- Respaldos y recuperaciones rápidas.
- Respalda archivos abiertos y nos asegura su coherencia.
- Respaldos de la ACL de los archivos y directorios.
- Recuperación individual de archivos.

Las principales desventajas son:

- Requiere la presencia de un administrador que conozca la tecnología y la configure correctamente.
- No dispone de filtros para realizar búsquedas y consultas de respaldos según ciertos criterios, sino que se hace manualmente recorriendo los respaldos hasta encontrar el archivo deseado.
- El formato de las copias instantáneas no es un estándar, el único mecanismo posible para consultar o restaurar un respaldo es a través del explorador de Windows.
- No hay seguridad sobre los datos ni compresión para ahorrar espacio.
- Funcionan a nivel de volumen y no de carpeta lo que es un inconveniente importante ya que puede haber carpetas que no queremos respaldar.
- Los respaldos no son transportables y no pueden ser creados directamente en un servidor remoto ni hacerlos accesibles remotamente.
- Sólo está disponible para sistemas de archivos NTFS.

El objetivo es realizar un estudio de las herramientas de respaldos actuales, sus características, sus fortalezas y debilidades para así escoger la que se adecue a las necesidades de los clientes y les brinde seguridad en el manejo de la información que administran.

Para realizar la elección, se deben tener en cuenta los siguientes requerimientos especificados por los administradores del DPI:

- La herramienta debe ser preferiblemente de código libre.
- Ser multiplataforma, principalmente Linux y Windows.
- Permitir respaldar archivos de bases de datos en uso.
- Disponer de interfaz gráfica para facilitar la búsqueda y de archivos respaldados.

3.2 Algunas herramientas de respaldo existentes

Existe una gran variedad de programas que automatizan el respaldo de la información y con muy variadas características.
Entre los programas que encontramos escogimos los siguientes:

Código Libre

Programa	Licencia	Windows	Linux	Interfaz Gráfica
Areca Backup [71]	BSD	Si	Si	No
BackupPC [72]	GPL v2.0	Si	Si	Si
AMANDA [73]	GPL v2.0	Si	Si	Si
Bacula [74]	GPL v2.0	Si	Si	Si

Propietarios

EMC Legato NetWorker [75]	EMC Corporation	Si	Si	Si
Acronis BackUp and Recovery [76]	Acronis	Si	Si	Si

Porque pueden ser instalados tanto en Linux como en Windows y además poseen una interfaz gráfica para la planificación y auditoría de los respaldos realizados, salvo AMANDA, que se decidió incluirlo por la popularidad con la que cuenta en el rubro.
Además se dedica una sección al estudio de 2 herramientas comerciales, con motivo de tener una comparativa entre el software libre y el software comercial.
Para realizar el análisis de cada herramienta y posteriormente determinar comparativamente cual es el producto que se ajusta en mejor medida a las necesidades requeridas, seleccionamos junto a los clientes un conjunto de características deseables que deberían formar parte de la herramienta.

A continuación se detallan dichas características:

- ***Código Abierto***

Una aplicación de código abierto está licenciada de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad del código fuente.

"Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

- *La libertad de ejecutar el programa, para cualquier propósito (libertad 0).*
- *La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.*
- *La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).*
- *La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.*" [70]

➢ ***Encriptación de Datos***

Es fundamental disponer de mecanismos de encriptación de datos, ya que los respaldos almacenados podrán eventualmente ser accedidos por usuarios distintos al usuario creador del respaldo, lo cual pone en riesgo la confidencialidad de dichos datos. Se espera que las herramientas dispongan de mecanismos de clave simétrica, clave asimétrica o ambas. Dependiendo del algoritmo de encriptación seleccionado la performance a la hora de realizar el respaldo podrá incrementar o decrementar, pero debe quedar claro que aplicar encriptación de datos aumentará los tiempos de ejecución si se compara con un respaldo sin encriptación.
Los algoritmos comúnmente utilizados son los siguientes:

- AES (Advanced Encrytion Standard)
- DES (Data Encryption Standard)
- RSA (Ron Rivest, Adi Shamir, Len Adleman)

➢ ***Compresión de Datos***

La compresión de datos es una característica fundamental si no se dispone de gran cantidad de espacio en los dispositivos de almacenamiento secundario. Como contrapartida, la compresión incrementa considerablemente los tiempos de ejecución de los respaldos y en ocasiones puede perderse información de los archivos, como por ejemplo los permisos y ACL (Acces Control List) asociadas.

- ***Tipos de Respaldo***

Disponer de una amplia gama de categorías de respaldos (Completo, Incremental, Diferencial, etc) permite a los usuarios implementar distintas estrategias de respaldo de acuerdo al contexto.

- ***Planificador de Tareas***

Para la automatización del sistema de respaldos es importante disponer de un mecanismo de planificación, en el que el usuario pueda ingresar tareas de respaldo a ejecutarse en determinada fecha. En caso de no disponer de un planificador, la herramienta debería brindar alguna forma de interactuar con el planificador del sistema operativo para llevar a cabo las tareas.

- ***Utilización de estándares para el respaldo de archivos***

Es importante que la herramienta utilice estándares para respaldar archivos, como por ejemplo tar, dump, zip, etc. La razón se debe a que es fundamental poder restaurar un respaldo sin necesidad de disponer de la herramienta que lo realizó para hacerlo.

- ***Respaldo de permisos de archivos***

Para mantener la protección sobre los archivos del sistema luego de una restauración de los mismos es necesario respaldar el archivo junto con su información de control de acceso. Esto abarca permisos en ambientes Linux, y listas de control de acceso en ambientes Windows. Si no se dispone de este mecanismo el administrador del sistema deberá otorgar los permisos nuevamente, luego de la restauración del archivo.

- ***Respald o de archivos abiertos***

Dada la gran cantidad de usuarios potenciales en el sistema es muy probable que al momento de realizar respaldos existan archivos abiertos por algunos de los usuarios. Para resolver este problema es recomendable que la herramienta posee la capacidad de respaldarlos y mantener su coherencia.

- ***Facilidad y eficiencia en la restauración de archivos***

La facilidad para restauración de archivos viene en gran medida en la utilización de archivadores estándares como los mencionados anteriormente y la posibilidad de contar con una interfaz grafica que provea facilidades para buscar y restaurar los archivos deseados.

- ***Buena Documentación***

Es de suma importancia para poder comprender el diseño e implementación de los módulos que componen el sistema. Se pretende que la herramienta seleccionada tenga al menos un manual de usuario, un documento explicando el diseño y forma de extender el sistema, y un código fuente bien documentado que permita comprender el propósito y funcionalidad de cada objeto.

Para realizar la comparación entre las distintas herramientas se tendrán en cuenta las características mostradas en la Figura 10, que indica cuales de las características deseables posee la aplicación y cuáles no.
Para cada herramienta se mostrará la tabla correspondiente y luego del estudio de las mismas se confeccionará una tabla general para poder visualizar con mayor detalle la comparativa entre todas las herramientas estudiadas.

- Código abierto
- Multiplataforma
- Compresión
- Encriptación
- Filtros
- Tipos de respaldo
- Lugar de almacenamiento
- Modo de almacenamiento
- Respaldo de permisos y ACL de archivos
- Respaldos simultáneos
- Planificador de tareas
- Interfaz gráfica
- Eficiencia y facilidad para restaurar respaldos
- Utilización de estándares para respaldo (tar, dump, zip).
- Cliente-Servidor
- Buena documentación
- Extensibilidad para incorporar nuevas funcionalidades
- Soporte para respaldo de archivos abiertos
- Soporte para respaldo de bases de datos on-line (hot-backup)

Figura 10 - Tabla para especificar las características de los programas estudiados

En el documento del [Anexo15] se encuentran detalladas las características de cada uno de los programas estudiados, a continuación brindamos un resumen de las mismas:

3.2.1 **Areca BackUp [71]**

Característica	Disponible (Si, No)	Detalle	Comentarios
Código abierto	Si		
Multiplataforma	Si		Puede ser utilizado en cualquier maquina con entorno de ejecución Java.
Compresión	Si	Zip, Zip64	Partición en volúmenes de tamaño configurable por el usuario.
Encriptación	Si	AES 128 AES 256	Dispone de encriptación de nombres.
Filtros	Si	Extensión, directorio, expresiones regulares, tamaño y fecha de archivos links simbólicos.	Dispone tanto de filtros de inclusión como de exclusión.
Tipos de Respaldo	Si	Completo, Incremental y diferencial.	
Modo de Almacenamiento	Si	Standard, Delta e Imagen.	Imagen se refiere al almacenamiento de todos los archivos respaldados en un único archivo.
Lugar de Almacenamiento	Si	Disco, unidad de red, FTP, USB.	
Respaldo de permisos y ACL de archivos	Parcial		Solo respalda permisos en archivos sobre sistemas operativos Linux.
Planificador de tareas	No		Permite integrarse con planificadores propios del sistema operativo.
Interfaz gráfica	Si		
Eficiencia y facilidad para restaurar respaldos	Si		Si se dispone de programas para manipular archivos .zip es posible restaurar respaldos sin necesidad de instalar la herramienta.
Utilización de estándares para respaldo (tar, dump, zip).	Si	Zip, Zip64	Zip64 permite comprimir archivos mayores a 4G en un único archivador.
Cliente-Servidor	No		
Buena documentación	Parcial		No se detalla con claridad cómo incorporar nuevos módulos al sistema.

Extensibilidad para incorporar nuevas funcionalidades	Si		Diseñado para incorporar plugins, en la web se dispone de un ejemplo.
Soporte para respaldo de archivos abiertos	No		
Soporte para respaldo de bases de datos on-line (hot-backup)	No		

3.2.2 BackUp PC [72]

Característica	Disponible (Si, No)	Detalle	Comentarios
Código abierto	Si		
Multiplataforma	Si		Servidor: Linux, Solaris o Unix. Cliente: Windows, Linux, Solaris, Unix.
Compresión	Si	Permite ahorrar entre 30-40 % de espacio.	Requiere instalar zlib y zip.
Encriptación	Si		rsa
Filtros	No		
Tipos de Respaldo	Si	Completo, Incremental y diferencial.	
Modo de Almacenamiento	Si	Full e incremental	Realiza backup sintéticos
Lugar de Almacenamiento	Sí	Disco local	El sistema de archivos debe soportar enlaces duros.
Respaldo de permisos y ACL de archivos	Parcial	Solo Linux	
Planificador de tareas	No		
Interfaz gráfica	Si	Posee interfaz grafica web.	
Eficiencia y facilidad para restaurar respaldos	Si		
Utilización de estándares para respaldo (tar, dump, zip).	Si	Tar, zip, rsync, smb.	
Cliente-Servidor	No		Requiere smbclient y nmblookup para respaldar clientes windows.
Buena documentación	Si		
Extensibilidad para incorporar nuevas funcionalidades	No		
Soporte para respaldo de archivos abiertos	No		
Soporte para respaldo de bases de datos on-line (hot-backup)	No		

3.2.3 **AMANDA (Advanced Maryland Automatic Network Disk Archiver) [73]**

Característica	Disponible (Si, No)	Detalle	Comentarios
Código abierto	Si		
Multiplataforma	Sí		Servidor requiere Linux/Unix, puede correr en Windows con Cygwin. Soporta cualquier cliente.
Compresión	Si	Tar en Linux y Zip en Windows	
Encriptación	Si	AES128, AES192 y AES256	
Filtros	No		
Tipos de Respaldo	Si	Completo, Incremental.	Pueden ser manejados automáticamente para balancear la carga diaria.
Modo de Almacenamiento	Si	Soporta imagen.	
Lugar de Almacenamiento	Si	Disco local, cinta.	
Respaldo de permisos y ACL de archivos	Si	Requiere habilitar VSS en clientes windows.	
Planificador de tareas	Si		
Interfaz gráfica	No		
Eficiencia y facilidad para restaurar respaldos	Si		
Utilización de estándares para respaldo (tar, dump, zip).	Si	Tar, Gump, zip	
Cliente-Servidor	Si	El software cliente de AMANDA debe instalarse en los PC a respaldar.	
Buena documentación	Si		
Extensibilidad para incorporar nuevas funcionalidades	No		
Soporte para respaldo de archivos abiertos	Si	Requiere activar VSS en clientes Windows.	
Soporte para respaldo de bases de datos on-line (hot-backup)	Si	Soporte para MySQL en un módulo dedicado.	

3.2.4 **Bacula [74]**

Característica	Disponible (Si, No)	Detalle	Comentarios
Código abierto	Si		
Multiplataforma	Sí	Servidor requiere Linux/Unix. Soporta cualquier cliente.	
Compresión	Si	Gzip (nivel 0-9)	
Encriptación	Si	MD5, SHA1	
Filtros	Si	Extensión, subdirectorio,	Dispone tanto de filtros de inclusión como de exclusión.
Tipos de Respaldo	Si	Completo, Incrementa, Diferencial.	
Modo de Almacenamiento	Si	En volúmenes	Un volumen es un repositorio simple para la información respaldada
Lugar de Almacenamiento	Si	Discos, cintas, CDs, DVDs., USBs	
Respaldo de permisos y ACL de archivos	Si	Requiere habilitar VSS en clientes windows.	
Planificador de tareas	Si		
Interfaz gráfica	Si		
Eficiencia y facilidad para restaurar respaldos	Si		
Utilización de estándares para respaldo (tar, dump, zip).	Si	Gzip	
Cliente-Servidor	Si		
Buena documentación	Si		
Extensibilidad para incorporar nuevas funcionalidades	No		
Soporte para respaldo de archivos abiertos	Si	Requiere activar VSS en clientes Windows.	
Soporte para respaldo de bases de datos on-line (hot-backup)	Si	En Windows utiliza vss y en Linux usa scripts previstos por la comunidad Bacula	

3.2.5 **Legato NetWorker [75]**

Característica	Disponible (Si, No)	Detalle	Comentarios
Código abierto	No		
Multiplataforma	Si		
Compresión	Sí	Pero esta deshabilitada por defecto.	
Encriptación	Si		
Filtros	Si	Extensión, subdirectorio,	
Tipos de Respaldo	Si	Completo, Incrementa, Diferencial y Consolidado.	
Modo de Almacenamiento	Si		
Lugar de Almacenamiento	Si	Discos, cintas.	
Respaldo de permisos y ACL de archivos	Si		
Planificador de tareas	Si		
Interfaz gráfica	Si		
Eficiencia y facilidad para restaurar respaldos	Si	Para el administrador	Si el usuario quiere recuperar sus respaldos debe instalar un módulo complementario.
Utilización de estándares para respaldo (tar, dump, zip).	No		
Cliente-Servidor	Si	El software cliente se debe instalarse en los PC a respaldar.	
Buena documentación	Si		Manuales de Administración, de usuarios y de instalación.
Extensibilidad para incorporar nuevas funcionalidades	Si		Se pueden instalar módulos de la empresa para extender sus funcionalidades.
Soporte para respaldo de archivos abiertos	Si		
Soporte para respaldo de bases de datos on-line (hot-backup)	Si		Oracle, Informix, Sybase, DB2, Microsoft SQL Server, MS Exchange, Lotus Notes

3.2.6 **Acronis Backup and Recovery 10 [76]**

Característica	Disponible (Si, No)	Detalle	Comentarios
Código abierto	No		
Multiplataforma	Si		
Compresión	Sí	Pero esta deshabilitada por defecto.	
Encriptación	Si		
Filtros	Si	Extensión, subdirectorio,	
Tipos de Respaldo	Si	Completo, Incrementa, Diferencial, Consolidado.	
Modo de Almacenamiento	Si	Imagen	
Lugar de Almacenamiento	Si	DAS,NAS, SAN, raid, firewire, USB, DVDs.	
Respaldo de permisos y ACL de archivos	Si		
Planificador de tareas	Si		
Interfaz gráfica	Si		
Eficiencia y facilidad para restaurar respaldos	Si		
Utilización de estándares para respaldo (tar, dump, zip).	No		
Cliente-Servidor	Si	El software cliente se debe instalarse en los PC a respaldar.	
Buena documentación	Si		Manuales de Administración, de usuarios, de instalación y referencia de línea de comandos.
Extensibilidad para incorporar nuevas funcionalidades	Si		Se pueden instalar módulos de la empresa para extender sus funcionalidades.
Soporte para respaldo de archivos abiertos	Si		
Soporte para respaldo de bases de datos on-line (hot-backup)	Si		Además dispone de otro producto: “Acronis® Recovery™ para MS SQL Server“ que se especializa en este manejador de base de datos.

4 Estado del Arte Registro de Eventos

Este capítulo tiene como objetivo describir las principales herramientas disponibles para el registro de eventos así como para el envío de mensajes de auditoría desde un equipo cliente a un servidor, implementado las especificaciones del perfil ATNA.

4.1 Registro de Windows

En los sistemas operativos Windows existe lo que se denomina Servicio de Registro de Eventos [30], inicialmente bajo el nombre de Event Logging [31] (Windows XP, server 2003), posteriormente rediseñado y llamado Windows Event Log (Windows Vista en adelante) el cual graba eventos de aplicación, de seguridad, y de sistema. Para visualizar los mismos ofrece una interfaz denominada Visor de Sucesos, además de una API para examinar los registros de evento, la cual puede utilizarse para implementar una propia interfaz de auditoría de los mismos.

Los registros de seguridad y de sistema son para uso exclusivo del sistema operativo. Los registros de aplicación tienen como objetivo brindarle a las aplicaciones de usuario un mecanismo para que puedan registrar sus propios eventos. El desarrollador deberá determinar los eventos a registrar y codificar el registro de cada uno utilizando la API que provee el servicio, dependiendo de la versión del sistema operativo.

El primer inconveniente que se tiene a la hora de desarrollar utilizando esta API es el hecho de que es compatible únicamente con aplicaciones de usuario desarrolladas en los lenguajes de programación C y C++. En el Hospital de Clínicas se cuenta con aplicaciones desarrolladas en java y PHP por lo tanto no es posible registrar eventos desde ellas.

A su vez, los mensajes de auditoría deben definirse en un archivo de texto plano para luego ser referenciados desde el código. Esto impide generar mensajes desde el código prescindiendo de cualquier tipo de archivo, lo cual complica la codificación. La Figura 11 muestra un mensaje de auditoría definido en estos archivos, y como se puede apreciar se debe definir el identificador, categoría, facilidad y severidad del evento, así como el identificador del mensaje.

```
MessageId=0x1
Severity=Error
Facility=Runtime
SymbolicName=MSG_BAD_COMMAND
Language=English
You have chosen an incorrect command.
.

Language=Japanese
<Japanese message string goes here>
.
```

Figura 11 - Formato de Mensaje de Auditoría para definido por el Registro de Eventos de Windows

Por otro lado el contenido variable de cada mensaje se almacena en una cadena de texto sin formato alguno, lo cual puede ser difícil de auditar si se tiene en cuenta el tamaño de la información a registrar y la estructura de la misma. A modo de ejemplo, el mensaje de auditoría recomendado por el perfil ATNA indica la presencia de cuatro campos básicos: información del evento, del usuario, del sistema, y del registro del paciente, y la representación del mismo se recomienda que sea en formato XML y no en una simple cadena de texto. Dado que toda esta información debe ser colocada en el cuerpo del mensaje a enviar al Registro de Eventos de Windows, es imposible realizar búsquedas por campos de interés dentro de ese cuerpo. El Visor de Windows únicamente permite ordenar o filtrar eventos por los campos del mensaje, es decir identificador, facilidad, severidad y categoría.

Otro punto en contra radica en que las aplicaciones que deseen registrar eventos deben ser agregadas en el registro de Windows bajo la siguiente clave: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Application. Esto es poco práctico si se piensa en un gran número de nodos que potencialmente podrá ejecutar varias aplicaciones que necesiten registrar eventos. Por otra parte esto contrasta con los requerimientos definidos por el Hospital de Clínicas, de disponer de un sistema de registro de eventos que permita registrar los mismos a cualquier aplicación que tenga las credenciales suficientes, sin la necesidad de controlar el acceso en el propio sistema de registros como ocurre en el caso del Servicio de Registro de Eventos de Windows.

La vulnerabilidad del sistema es un punto crítico ya que si no es configurado correctamente, un usuario iniciando sesión como invitado, puede visualizar todos los eventos registrados en dicho sistema. Más allá de esto, si un usuario logra iniciar sesión como administrador podrá ver también todos los eventos, por lo tanto la seguridad depende de la preservación de las contraseñas en secreto, lo cual no es ideal.

Por último, existe una limitante de 300-megabytes en el tamaño del archivo de registro de eventos, lo cual puede ocasionar perdida de datos si el volumen del mismo crece considerablemente día a día.

Nota: Cabe señalar que en la nueva versión del Sistema de Registro de Eventos de Windows, que viene instalada en los sistemas operativos Windows Vista y Windows Server 2008 o posteriores, se migró a un formato de log basado en XML y se mejoró sensiblemente la API para registro de eventos, permitiéndose a aplicativos desarrollados en C# registrar los eventos de forma más sencilla. Además se eliminó la limitante de 300-megabytes en el tamaño del archivo de registro.

4.2 Bibliotecas Syslog

Si bien Syslog es el nombre del protocolo para envío de mensajes de registro en una red informática, también existen aplicaciones y bibliotecas con dicho nombre que implementan el protocolo. Dado que Syslog es una aplicación cliente-servidor, existe un computador ejecutando la parte servidor de Syslog, mientras que los computadores que deseen registrar eventos ejecutarán la parte cliente enviando mensajes a dicho servidor.

Toda implementación de Syslog debe apuntar a optimizar los siguientes tres aspectos:

- Recepción de mensajes sin pérdidas.
- Tiempo de almacenamiento del mensaje.
- Visualización en pantalla de grandes cantidades de mensajes.

Existe un punto del protocolo Syslog que ocasiona conflictos con la especificación de ATNA para el envío de mensajes en el contexto hospitalario, y el mismo es tenido en cuenta por algunas implementaciones y por otras no: el tamaño máximo del mensaje de auditoría. El protocolo Syslog establece el tamaño máximo en 1024 bytes, es decir 1KB, mientras que la especificación ATNA indica 32KB. Este tamaño de 32KB se debe a que en el contexto hospitalario es necesario almacenar suficiente información sobre ciertos eventos, que deben incluir tanto información de identificación del evento como del paciente en cuestión.

Esta limitante surge del hecho que la aplicación Syslog fue diseñada para registrar alertas puntuales de los sistemas, como por ejemplo accesos denegados y errores de hardware o software, sin incluir demasiada información acerca del evento.

Esto restringe la inclusión de información tanto del paciente en cuestión, como de los usuarios y sistemas que originaron el evento pudiéndose únicamente almacenar algunos campos que identifiquen al evento.

A continuación se presenta una lista no exhaustiva de las implementaciones existentes y luego se detallan las más destacadas:

- WinSyslog [26]
- Syslog-ng [27]
- Syslog Watcher [25]
- Kiwi Syslog [28]

4.2.1 **WinSyslog**

WinSyslog es la primera implementación de Syslog para sistemas operativos Windows y ofrece cuatro versiones: freeware, básica, profesional y enterprise, cada una de ellas introduce nuevas funcionalidades.

A continuación se presentan las principales características de WinSyslog:

- Recepción de mensajes desde firewalls y routers.
- Cumplimiento de las RFC 3164, 3195 [17] y 5424, por lo tanto ofrece envío confiable sobre TCP.
- Soporte para Windows 2000, 2003, XP, Vista, 2008 y Windows 7 tanto en sus versiones 32-bit como 64-bit.
- Acceso remoto mediante interfaz web desarrollada con tecnología ASP.
- Post procesamiento de eventos permiten tomar acciones según las reglas definidas en el servidor y el tipo de mensaje recibido, por ejemplo envío de alertas vía email y almacenamiento en base de datos.
- Monitoreo y alertas en tiempo real.
- Ejecución de múltiples instancias del servidor, cada una escuchando en un puerto distinto.

4.2.2 Syslog Watcher

Esta implementación es desarrollada y mantenida por la compañía SnmpSoft y ofrece tanto una versión comercial como una versión libre, teniendo esta última una limitación en las funcionalidades.

A continuación se presentan las principales características:

- Soporte para redes Ipv4 e Ipv6 y comunicación tanto sobre UDP como TCP.
- Compatibilidad con sistemas operativos Windows 32-bit y 64-bit (2000, XP, Vista, 7).
- Reconocimiento del estándar Syslog definido en la RFC 3164 pero a su vez de otros formatos no-estándar, por ejemplo el definido en la RFC 5424.
- Resolución de nombres y cacheo DNS.
- Visualización de mensajes en tiempo real.
- Reducción del espacio de almacenamiento mediante la configuración del tiempo de almacenamiento de los mensajes dependiendo de su severidad.
- Tecnología DynamicView permite visualizar en pantalla más de un millón de mensajes.
- Arquitectura de alta performance permite registrar en el orden de miles de mensajes por segundo.
- Exportación de mensajes a una base de datos (ODBC) o archivos XML y CSV (comma separated value).

Almacenamiento Óptimo

En un escenario donde se registran gran cantidad de mensajes provenientes de la red, el tráfico generado por estos mensajes es enorme, y si el cliente requiere mantener los mensajes por un largo periodo de tiempo, el resultado deriva en un gran crecimiento del espacio de almacenamiento disponible para almacenar los mensajes Syslog. Esto lleva a introducir limitaciones en la forma de almacenar los mensajes, manteniendo aquellos que se consideran importantes, por un periodo mayor de tiempo, mientras que los menos importantes serán mantenidos la menor cantidad de tiempo posible.

A modo de ejemplo, suponiendo que se dispone de una red que envía mensajes importantes al servidor Syslog pocas veces por hora, mensajes de información pocas veces por minuto y mensajes de debug cada 2 o 3 segundos. Si el cliente desea mantener todos los mensajes

por un largo periodo de tiempo, entonces como indica la línea roja de la Figura 12, al cabo de un año el tamaño del espacio de almacenamiento será de aproximadamente 3GB.

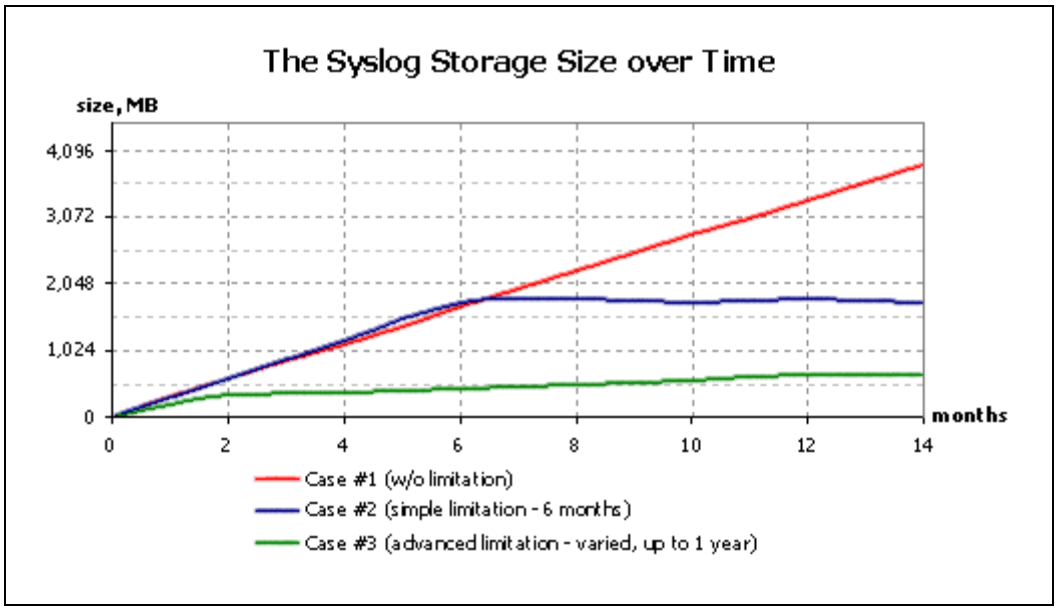


Figura 12 - Espacio de Almacenamiento sobre el tiempo para Syslog Watcher [25]

La introducción de una simple limitante que mantenga los mensajes por un periodo máximo de seis meses, permite reducir el espacio de almacenamiento a 1.5GB aproximadamente, es decir la mitad que en el primer caso. La línea azul del grafico representa este caso. Por último si se introduce una limitante avanzada, que mantenga todos los mensajes por un año, a excepción de los mensajes de debug, que serán mantenidos por un mes, el espacio de almacenamiento rondara los 500-600 MB, representado por la línea verde de la Figura 12.

Visor de Eventos

El visor de eventos es parte fundamental del servidor Syslog, ya que deberá ser capaz de desplegar gran cantidad de mensajes en pantalla con la mayor rapidez posible. Para lograr esto, se dispone de un conjunto de filtros que permita acotar la búsqueda y por ende el tiempo de respuesta del visor de eventos:

- Filtro por rango de fechas.
- Filtro por cantidad de mensajes a desplegar.
- Filtro por severidad.
- Filtro por facilidad.
- Filtro por contenido del mensaje.

A su vez, se ofrece ordenamiento de datos por cada uno de los campos mencionados anteriormente.

El espacio de almacenamiento es una base de datos especialmente optimizada, que permite leer hasta 250.000 registros por segundo.

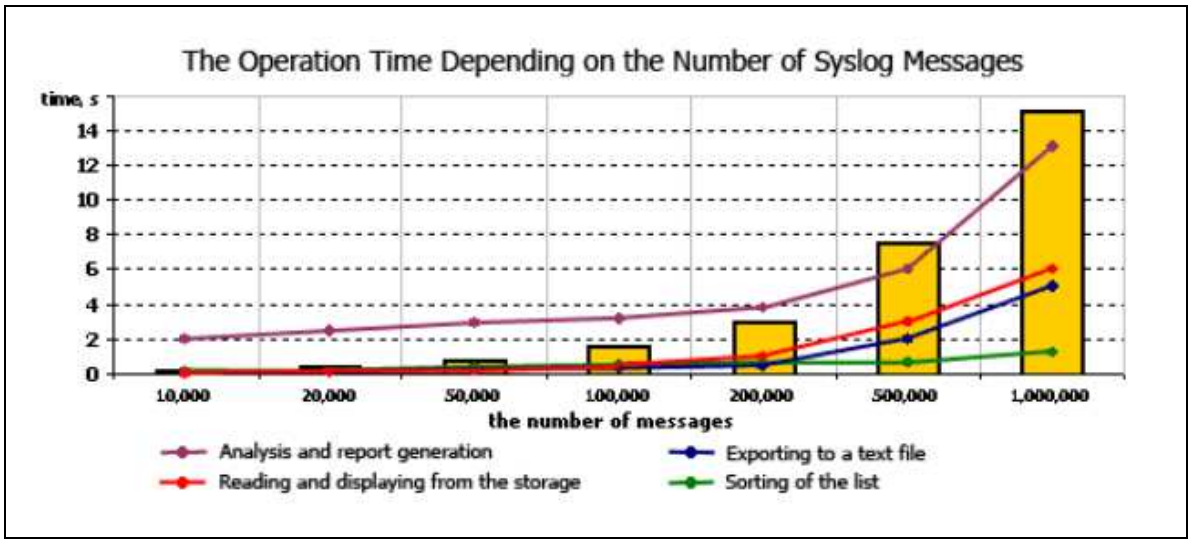


Figura 13 - Tiempos de operación para el visor de eventos Syslog Watcher [25]

La Figura 13 muestra el tiempo de respuesta según cantidad de mensajes para cuatro operaciones:

- Análisis y generación de reportes (línea violeta).
- Lectura y visualización de datos desde almacenamiento (línea roja).
- Exportación de datos a archivo de texto (línea azul).
- Ordenamiento de datos (línea verde).

4.3 Open Healthcare Framework

OHF [29] es un proyecto desarrollado en lenguaje Java por Eclipse, el cual implementa un conjunto de estándares y perfiles del área de salud, entre ellos el perfil ATNA el cual recomienda el estándar para mensaje de auditoría definido por IETF en la RFC 3881.

Si bien no se trata de una implementación de un sistema de registro de eventos, es interesante destacarla ya que una parte importante del registro de eventos, es la definición y comunicación del mensaje de auditoría entre el cliente y el servidor.

La biblioteca implementa completamente el esquema XML que se propone en la RFC 3881, pero aquí aparece un primer inconveniente. Si bien, la RFC 3881 propone un esquema XML, no tiene como objetivo que este permanezca inalterado durante el tiempo, ya que los requerimientos de seguridad de cada institución pueden cambiar y es necesario adaptarse a dichos cambios, tomando sí como base el estándar definido en RFC 3881. Esto lleva a que el esquema pueda ser modificado o extendido para agregar más información de interés, pero esto no es posible lograrlo ya que la biblioteca implementa la versión del esquema definida en la RFC 3881 y no es posible adaptarla a una nueva versión del mismo.

Por otro lado, implementa únicamente comunicación con el servidor Syslog mediante UDP sin ningún mecanismo de seguridad (SSL). En este punto, sería interesante disponer de un mecanismo de seguridad para brindar autenticación, confidencialidad e integridad en el mensaje enviado al servidor Syslog. También sería interesante que la herramienta brindara cierta flexibilidad para la implementación de otros mecanismos de comunicación como por ejemplo servicios web, el cual es un requerimiento definido por el Hospital de Clínicas.

Por último sería de utilidad una funcionalidad de exportación del mensaje de auditoría a una cadena de texto en formato XML, para únicamente concentrarse en la implementación del envío de dicho mensaje XML al Sistema de Registro de Eventos.

5 Planteo de la solución Sistema de Respaldos

En base al estudio realizado de las herramientas de respaldos podemos destacar que:

1. Las herramientas pagas ofrecen excelentes características, entre ellas, respaldos de bases de datos en uso, programación de tareas de respaldo y restauración y búsqueda de archivos, además de interfaces de administración y configuración muy amigables, intuitivas y completas pero su costo es elevado, en particular para instalar en las cientos de máquinas existentes en el Hospital de Clínicas.

2. De las herramientas gratuitas estudiadas, Bacula permite el respaldo de bases de datos en uso, programación de tareas de respaldo y restauración, búsqueda de archivos e interfaz gráfica donde realizar configuraciones y visualizar el estado de las tareas de respaldo.
Del resto de las programas, dos no respaldan bases de datos en uso (como Areca y BackupPC) y una no poseen interfaz gráfica para realizar configuraciones y facilitar su utilización (AMANDA).

Es por esto que se concluye que, de las herramientas estudiadas, `Bacula` es la que mejor se adapta a los requerimientos especificados por los clientes.

A continuación se detallan los componentes del programa de respaldo escogido, cómo se integran los mismos y sus respectivas funcionalidades, finalizando con una guía paso a paso para su instalación y configuración.

5.1. Descripción General

Es una herramienta de respaldo que no está dirigido a realizar imágenes de sistema ni orientado a restaurar sistemas completos, sino que realiza copias de seguridad a nivel de archivos.

Bacula implementa 5 módulos independientes: el Director o programa que gestiona las copias de seguridad, el Catálogo que es el encargado de almacenar en una base de datos todas las acciones realizadas y la información referida a los respaldos (archivos, localización, metadatos, etc), el componente encargado del Almacenamiento de los datos, los Clientes y por último la Consola que es donde se accede a la información y configuración del resto de los componentes.

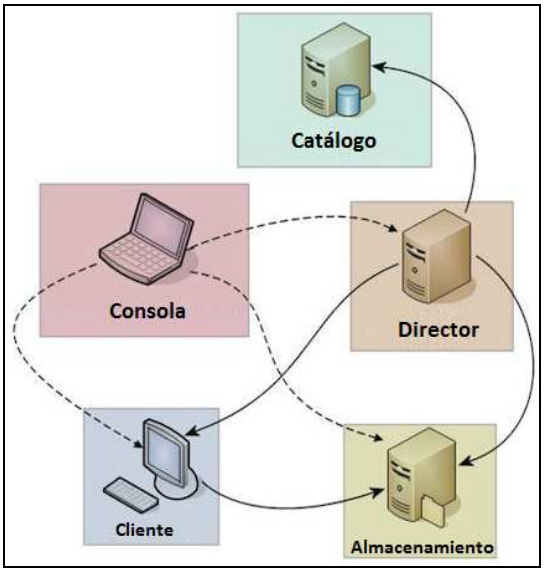


Figura 14 - Esquema simple de la arquitectura de Bacula

El Director solo puede ser instalado en sistemas Linux/Unix mientras que el resto de los componentes pueden ser instalados tanto en Windows, Linux o Mac.

En el caso de ejemplo desarrollado todos los componentes estarán instalados en la misma computadora, más precisamente un máquina virtual con sistema operativo Debian Lenny con el fin de ofrecer una visión general del programa y dando las herramientas necesarias para poder realizar la configuración en un sistema en particular, pero en realidad el sistema permite repartir sus funciones en procesos independientes, cada uno en una máquina diferente, de forma de mantener el máximo control a la hora de poner el sistema en producción.

5.2. Arquitectura de Bacula

Cómo ya mencionamos, Bacula está compuesto por 5 módulos, Director, Consola, Almacenamiento, Catalogo y Cliente, en esta sección describiremos de forma general la funcionalidad de cada uno de ellos (por detalles ver [74]):

Director: Es el eje central de la solución, controla el flujo de datos, supervisa todas las funciones necesarias para las operaciones de copia y posterior restauración, le indica a los clientes que comiencen a empaquetar los archivos para enviarlo al almacenamiento, etc. En él se declaran todos los parámetros necesarios, siendo el módulo más difícil de configurar. Realiza los backups siguiendo las pautas dadas en las tareas que se programaron, pero también se pueden realizar backups y otras operaciones a mano, mediante la herramienta de administración (bconsole) o, en nuestro caso, la interfaz Webmin.
Es importante destacar que los datos no pasan por el director, sino que transitan directamente desde los Clientes al Almacenamiento lo que hace menor el tiempo de ejecución del respaldo y no sobrecarga los recursos (la red por ejemplo) innecesariamente.

Cliente (File Daemon - FD): Es el proceso que se ejecuta en la máquina que se va a respaldar y es específico al sistema operativo donde se ejecuta. Tiene como objetivo empaquetar los datos y enviarlos al Almacenamiento.

Almacenamiento (Storage Daemon - SD): Se encarga de manejar los dispositivos físicos donde se guardarán los datos y es el responsable de escribir y leer en cada uno de los medios que se utilizarán para las copias de seguridad
Bacula es compatible con múltiples volúmenes y múltiples configuraciones de copia en cada uno de ellos

Consola: Es el programa que permite la interacción entre el administrador del sistema y el módulo Director, de manera que se accede a todas las funcionalidades de Bacula. Originalmente era un programa en modo texto, que sigue siendo válido, aunque hay interfaces gráficas y webs disponibles (nosotros utilizaremos Webmin) que hacen el manejo muy simple.

Catalogo: Es un proceso que interactúa con la base de datos y mantiene la información necesaria para la administración de Bacula. Generalmente está instalado en la misma máquina que el Director pero no es restrictivo.
Básicamente es una base de datos donde quedan grabados todos los procesos de copia y la información de todos los archivos que lo componen, con su fecha, tamaño, lugar donde se

restauran, lugar donde se almacenan físicamente etc. Es una especie de índice de todo lo que va ocurriendo. No contiene los archivos en sí mismos.
Las bases de datos soportadas son MySQL, PostgreSQL y SQlite. (nosotros utilizaremos MySQL).

Y por último, si bien no es parte de los módulos del programa, se puede considerar el **Monitor** que es una interfaz que permite examinar el estado de las copias y el de cada uno de los componentes del sistema. Al igual que para la consola utilizaremos Webmin.

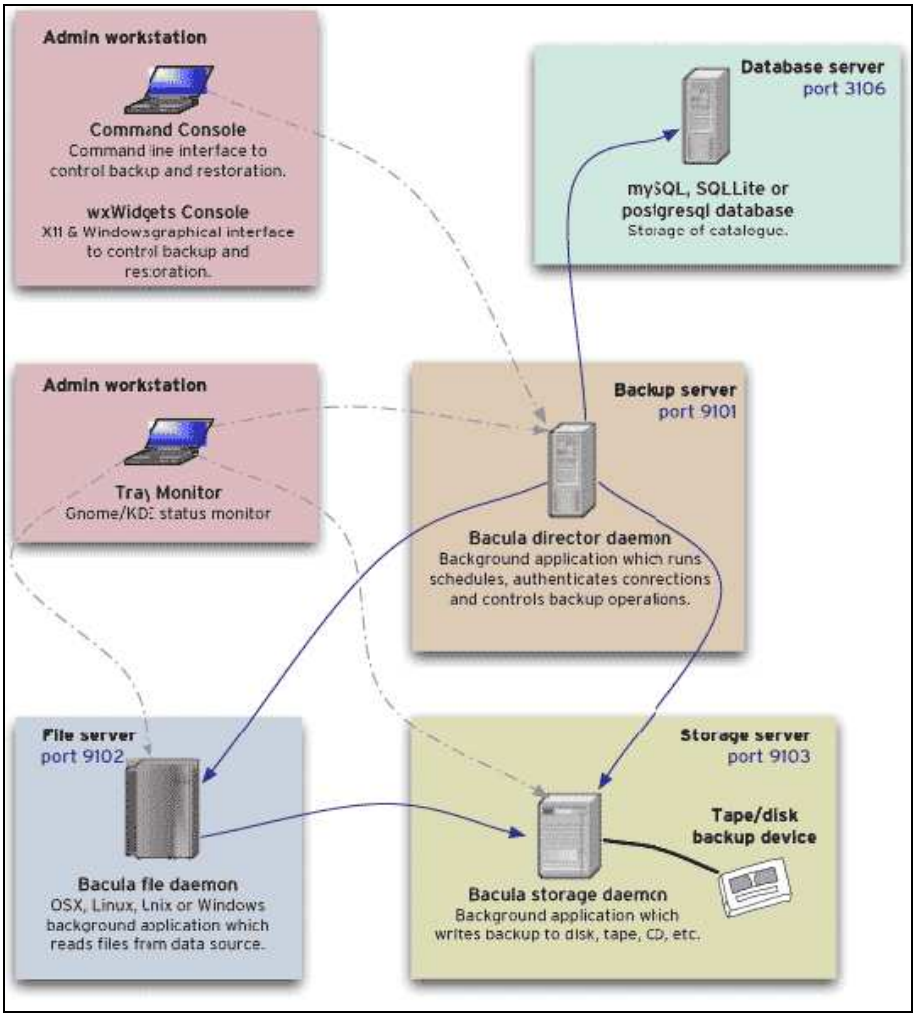


Figura 15 - Interacción de los componentes de Bacula

5.3. Configuración de Bacula

Para organizar los distintos módulos de Bacula y lograr que se reconozcan entre si hay que modificar 4 archivos, el que establece las características del Director (bacula.dir.conf), el del Almacenamiento (bacula.sd.conf), el de la Consola (bconsole.conf) y el del Cliente (bacula.fd.conf, diferente para cada uno de los clientes).

La configuración es sencilla si tenemos un ejemplo para seguir, por esta razón describiremos las secciones de cada uno de los archivos mencionados y mostraremos la configuración utilizada para el caso de prueba.

Hay que tener siempre presente que es el Director el que se autentica contra el resto de los módulos, y no al contrario, por lo que estos deben conocer la identidad del Director.

Elementos del bacula-dir.conf

Este es el archivo de configuración del Bacula Director. En este archivo se definen los detalles del Director, las tareas de respaldos, sus políticas, etc,

Los elementos que se deben definir en este archivo son:

Director: Donde se especifica la configuración del Director. Entre las opciones de configuración, se especifica la IP y el puerto donde está “escuchando” el director, la cantidad de tareas concurrentes que se pueden ejecutar como máximo, etc.

Storage: Donde se indica en qué IP y puerto “escucha” el módulo de Almacenamiento, el password que debe usar para identificarse contra él y el nombre y tipo del dispositivo de almacenamiento.

Catalog: Tiene la información para conectarse a la base de datos donde se almacena la información de los respaldos (usuario, password, nombre de la tabla)

Messages: Se define que mensajes se va a generar Bacula, por ejemplo, que envíe emails con el estado de los backups, que genere logs de todos los jobs que salieron mal, que ejecute un script después de cada job, etc.

Schedule: Sirve para implementar la política del respaldo, junto con las opciones definidas en el elemento “pool”. Define cuando se debe realizar una tarea y en donde se debe almacenar/recuperar la información.

Pool: Es un objeto lógico que se utiliza para agrupar volúmenes. Todos los volúmenes de un pool comparten las características definidas en ese pool. Los pools complementan a los schedules, y entre los dos implementan la política del respaldo.

JobDefs: Este elemento es un esqueleto para las tareas de respaldos (jobs). Se especifican parámetros por defecto. Si un job hace referencia a un JobDefs y no especifica un parámetro, toma el que está definido en el jobDefs (en caso de que ahí este definido).

Job: Acá se agrupan la información definida anteriormente (Pool, Schedule, Type, etc) para definir la tarea. Bacula ejecuta los jobs automáticamente sólo si tiene un schedule asociado, Si no lo tiene, la única forma de ejecutarlo es manualmente.
Hay tres clases de jobs: Backup, Restore y Verify (verifica que los atributos de los archivos originales sean iguales a los atributos guardados en la base de datos para esos archivos)

Client: Se indican los clientes que se van a respaldar especificando un nombre, la IP donde se encuentra, el puerto en el que “escucha” y el password con el que el director se tiene que autenticar.

FileSet: Acá decimos los directorios y archivos que se deben respaldar. También se especifica si se deben cifrar los datos (md5 o sha1), comprimirlos o si hay archivos o carpetas que se deben excluir del respaldo.

Por más detalles acerca de la configuración del Director, los elementos que lo componen y si significado y funcionalidad ver [77].

Elementos del bacula-sd.conf

En este archivo está la configuración del módulo de Almacenamiento de Bacula, es donde define la ubicación del mismo y los dispositivos de almacenamientos disponibles donde se van a almacenar físicamente los datos respaldados.

Los elementos que hay que configurar son:

Storage: Donde se especifica las características del módulo, como el nombre y la IP y el puerto donde está “escuchando” el módulo Almacenamiento. En el archivo de configuración solo puede haber una definición de este elemento.

Director: Especifica el nombre del director que tiene autorización para utilizar los servicios del demonio de Almacenamiento y el password que utiliza para autenticarse, por lo que debe coincidir con los valores correspondientes en el archivo de configuración del director. Puede haber múltiples recursos Director.

Device: Especifica los detalles de cada dispositivo de almacenamiento que puede ser usado por el modulo de Almacenamiento. Se pueden definir múltiples recursos de almacenamientos para ser usados.

Messages: Donde se definen los mensajes de error e información que se deben generar y adonde se envían.

Por más detalles acerca de la configuración del Almacenamiento en [78].

Elementos del bacula-fd.conf

En este archivo se configura el cliente y es uno de los más sencillos de configurar. Este archivo está alojado en la maquina que se va a respaldar y no en el director habiendo solo uno por cliente.

Los elementos que hay que configurar son:

Client (o FileDaemon): Define el nombre del cliente, así como la IP del equipo donde está alojado y el puerto por el cual “escucha” las conexiones de Director.
El Director identifica al cliente especificado en la tarea de respaldo mediante el nombre definido en este elemento.

Director: Especifica el nombre del director que tiene autorización para respaldarlo y el password que utiliza para autenticarse, por lo que debe coincidir con los valores correspondientes en el archivo de configuración del Director. Puede haber múltiples recursos Director.

Messages: Igual funcionalidad que en los otros módulos.

Por más detalles acerca de la configuración del Cliente en [79].

Elementos del bconsole.conf

El archivo de configuración de la Consola es el más simple de todos los archivos de configuración debido a que solo tiene el elemento Director (pueden ser varios) y solo hay que cambiar es el nombre y la contraseña según corresponda.

La Figura 16 muestra las típicas interacciones entre los servicios de Bacula para la ejecución de un respaldo El Director inicia y gestiona estas interacciones además de administrar el Catalogo.

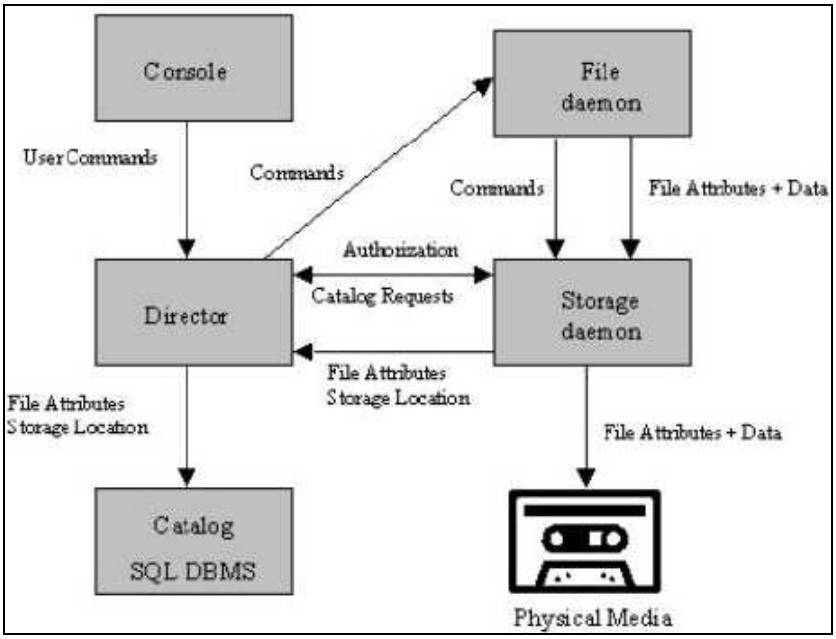


Figura 16 - Interacciones entre los servicios de Bacula.

En el documento del [Anexo15] se puede observar la configuración de ejemplo que se utilizará para el caso de prueba.

Planteo de la solución del Sistema de Registros.

6 Planteo de la solución del Sistema de Registros.

Este capítulo tiene como primer objetivo definir la problemática existente en el Hospital de Clínicas, la cual será resuelta al finalizar el presente trabajo. El segundo objetivo consiste en especificar los requisitos relevados durante las diferentes reuniones con el personal del DPI. Por último se definen y detallan las principales características de la solución propuesta, sus ventajas y desventajas, indicando las principales decisiones tomadas.

6.1 Problemática a resolver

La situación actual del Hospital de Clínicas indica que existen una serie de sistemas desarrollados en distintos lenguajes de programación (Java, PHP, Genexus) que carecen de un adecuado registro de eventos, disparados por los usuarios de dichos sistemas. Por evento se entiende aquellos definidos por el estándar de mensaje de auditoría sugerido por ATNA. Se destacan entre ellos los siguientes:

- Acceso exitoso al sistema.
- Acceso denegado al sistema.
- Creación/Consulta/Modificación de datos de uno o varios pacientes dentro del sistema.

Mantener el registro de dichos eventos es importante para el Hospital ya que permite una posterior auditoría para detectar situaciones que requieran atención, como por ejemplo accesos a los sistemas del hospital, manipulación de información de pacientes sin autorización previa y/o en forma malintencionada, etc. Esto es de gran ayuda para mantener cumplir con las políticas de seguridad y privacidad, tanto de los sistemas como de los datos de cada paciente del hospital. La pregunta que surge en este momento es por qué no implementar los mecanismos necesarios en cada sistema para poder detectar dichas anomalías en tiempo real, impidiendo que los registros de los pacientes sean modificados si no corresponde.

Actualmente en el Hospital de Clínicas los sistemas no tienen un formato bien definido para los mensajes de auditoría, por lo tanto cada sistema utiliza el formato desarrollado por los implementadores del mismo, o en el peor de los casos no se dejan pistas de auditoría.

En los casos en que se dejan, éstas carecen de mecanismos de seguridad por lo que cualquier persona que pueda obtener acceso a los mismos ilegalmente podrá destruirlos,

con las consecuencias que esto conlleva. De aquí surge la necesidad de implementar mecanismos de seguridad y de respaldo de registros de auditoría para disponer de una copia reciente en caso que los mecanismos de seguridad pudiesen ser violados. La Figura 17 describe la realidad actual en el Hospital de Clínicas:

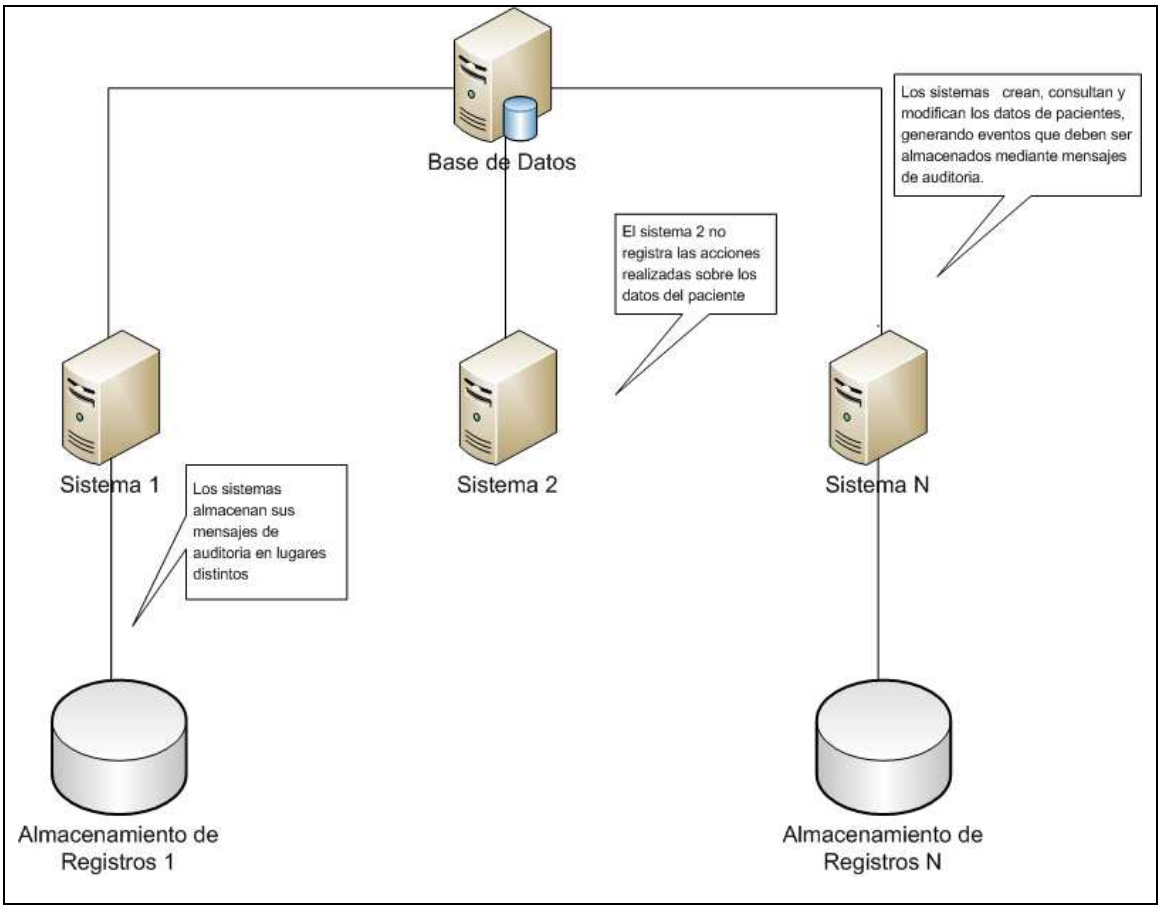


Figura 17 - Distribución actual de sistemas en el Hospital de Clínicas

Por otra parte se desea centralizar el registro de eventos en un único servidor, utilizando un único punto de entrada a utilizar por todos aquellos sistemas que requieran registro y posterior supervisión de sus eventos. La comunicación entre los sistemas y el servidor deberá contar con mecanismos de autenticación de nodos, así como mecanismos que aseguren la integridad y confidencialidad de los datos transmitidos.

Los mensajes enviados desde los sistemas cliente hacia el servidor de registros deberán respetar el formato de mensaje de auditoría definido por ATNA en la RFC 3881. En caso de no respetarse el servidor desechará el mensaje sin dejar ningún tipo de registro. Los mensajes deberán ser validados y posteriormente se almacenarán en una base de datos para ser auditados en su debido momento, así como se enviarán a un demonio Syslog para mantener un respaldo de cada mensaje recibido. Esto último es de mucha importancia ya que varias situaciones pueden alterar la información de la base de datos del servidor de

registros, como por ejemplo catástrofes que dañen físicamente los dispositivos de almacenamiento, ataques al servidor, lo cual puede resultar en modificación o eliminación de la información, o incluso administradores con suficientes permisos podrían eliminar sus propios eventos registrados previamente. Esto es de utilidad siempre y cuando el servidor que contiene al demonio Syslog tenga la debida protección y mecanismos de seguridad. A modo de ejemplo se listan algunos [61]:

- Deshabilitar todos los servicios del sistema operativo a excepción de aquellos indispensables (Syslog).
- Control de acceso root [64].
- Instalar firewall y habilitar únicamente el puerto 514/UDP que utiliza Syslog para recibir mensajes desde los clientes.
- Correr Syslog en ambiente chroot [63] para restringir la ejecución del proceso a un directorio particular, limitando el acceso unicamente a información de dicho directorio.
- Almacenar los logs de Syslog en particiones únicamente accedidas por el usuario Syslog. Para ello, se puede crear un script que mueva los logs hacia dicha ubicación y ejecutarlo periódicamente mediante un planificador, por ejemplo cron [61].

Por otro lado, aunque se tenga la posibilidad de contar con este servidor Syslog, es recomendable introducir mecanismos de seguridad en el propio servidor de registro de eventos a modo de reducir los ataques, que no solo afectaran la información almacenada sino que también podrán interrumpir la disponibilidad del servicio.

La implementación de dichas medidas no forma parte de la solución propuesta, sino que se dejan como recomendación y trabajo a futuro.

Los mensajes enviados al servidor Syslog deben respetar su formato y tamaño (1024 bytes), y deberán permitir identificar de forma precisa el evento registrado, por lo que deberá definirse cuidadosamente que información del evento se colocará en el cuerpo del mensaje Syslog para no exceder dicho límite.

Para que todo lo antes mencionado pueda ser de utilidad se requiere de un sistema que permita auditar de forma eficiente todos los registros almacenados en el servidor de registro de eventos. Actualmente el Hospital de Clínicas no dispone de tal sistema por lo que formara parte de la solución planteada junto con el sistema de registro de eventos.

Pero la solución planteada debe ir un paso más en la búsqueda de interoperabilidad entre el sistema de registro de eventos a desarrollar y los sistemas existentes en el Hospital de Clínicas, de modo que estos últimos puedan verse beneficiados a la brevedad. Es por esto que se decidió implementar los mecanismos y módulos necesarios para que un conjunto de sistemas del Hospital de Clínicas desarrollados con la herramienta Genexus y que carecen actualmente del soporte necesario para establecer una comunicación segura con el servicio web provisto por el sistema de registro de eventos puedan hacerlo.

En la Figura 18 se presenta un diagrama con la distribución de los nuevos sistemas y su interacción con los ya existentes:

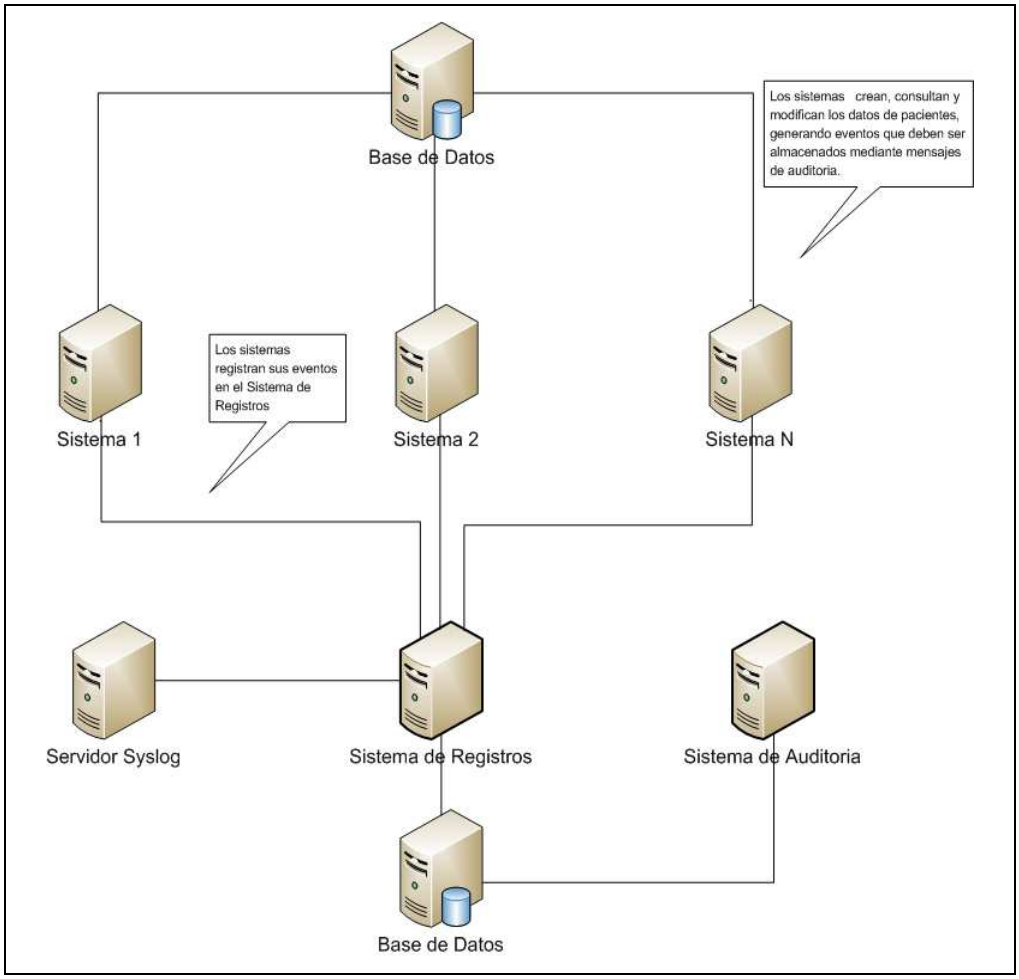


Figura 18 - Futura distribución de sistemas en el Hospital de Clínicas

En resumen los siguientes sistemas y módulos formarán parte de la solución a desarrollar:

- Sistema de Registro de Eventos.
- Sistema de Auditoría de Eventos.
- Módulo generador de mensajes de auditoría ATNA.
- Módulo de comunicación con el Sistema de Registro de Eventos.
- Módulo encapsulador de la lógica necesaria para el armado y envío del mensaje de auditoría en sistemas Genexus.

Nota: El último módulo se desarrollará en el lenguaje Genexus y utilizará el módulo generador de mensajes ATNA y el módulo de comunicación con el Sistema de Registro de Eventos. Tiene como objetivo facilitarle la creación y envío de mensajes de auditoría a los implementadores del Hospital de Clínicas.

6.2 Requerimientos del Sistema

Dada la problemática planteada y las sucesivas reuniones con los clientes, se definieron los siguientes requerimientos:

Sistema de Registro de Eventos

- Implementación de un sistema de registro de eventos.
- El punto de acceso al sistema de registros deberá ser un servicio web. Este requerimiento se debe a que el Hospital de Clínicas tiene como objetivo estandarizar la comunicación de sus sistemas mediante esta tecnología.
- Deberá implementarse mecanismos de seguridad en el servicio web para autenticar al sistema cliente, así como para garantizar la confidencialidad e integridad de los datos enviados desde los sistemas del hospital.
- Deberá respetarse el formato ATNA para el envío de datos desde el cliente hacia el sistema de registros, por lo cual deberá implementarse la generación de mensajes en dicho formato en caso de que no exista una implementación disponible. El sistema de registros deberá validar el mensaje contra el esquema XML propuesto por ATNA, desechándolo en caso de que la validación no tenga éxito.

- El sistema de registros deberá respaldar cada mensaje en un servidor Syslog, respetando el formato de mensaje que establece el propio estándar. Se deberá definir e implementar un formato para el cuerpo del mensaje Syslog, preferentemente en XML, el cual contendrá toda la información necesaria para identificar el evento, teniendo en cuenta la limitante definida por el protocolo Syslog para el tamaño del mensaje (1024 bytes).

- Debe tenerse en cuenta mecanismos que permitan que la comunicación entre el cliente y el sistema de registros sea lo más rápida posible, para no bloquear al cliente mientras se procesa el mensaje en el servidor.

- Se deberá configurar el envío automático de emails en caso que los tipos de evento sean de urgencia.

- Los parámetros del sistema deberán ser configurables por un administrador mediante una consola de administración.

- Documentación de diseño y arquitectura, y manual de Usuario en formato ISO utilizado por el Hospital de Clínicas.

- Código fuente bien documentado (Javadocs y comentarios en código).

- Almacenamiento de mensajes de auditoría en base de datos.

Sistema de Auditoría de Eventos

- Se deberá ofrecer un sistema web que permita auditar de manera eficiente los registros existentes en el sistema de registros. Las consultas podrán realizarse sobre los siguientes campos:

    - Campos de identificación del evento:

        - Fecha de registrado
        - Nivel de falla
        - Código de evento

    - Actores involucrados:

        - Nombre
        - Rol

    - Sistema que originó el mensaje:

        - Tipo de acceso: IP, nombre de máquina o teléfono.
        - IP, nombre o teléfono.
        - Código interno del sistema
        - Tipo de sistema.

    - Objetos involucrados en el evento:

        - Tipo (persona, sistema, etc…)
        - Nombre
        - Rol (paciente, doctor, etc…)
        - Ciclo de vida (creación, copia, verificación, etc…)

- Deberá permitir la visualización de gráficos con estadísticas tomadas a partir de los eventos registrados:

    - Cantidad de eventos por rango de fechas.
    - Cantidad de eventos por tipo.
    - Cantidad de eventos por nivel de falla.
    - Cantidad de eventos por actor.
    - Cantidad de eventos por sistema que lo origina.
    - Cantidad de eventos por tipo de objeto.

- Impresión de reportes en formato PDF, Excel y CVS (comma separated value) para los datos visualizados en pantalla según los filtros aplicados.
- Buena performance para desplegar la información en pantalla, teniendo en cuenta el volumen creciente de datos.
- Documentación de diseño y arquitectura, y manual de usuario respetando formato ISO utilizado por el Hospital de Clínicas.
- Código fuente bien documentado (Javadocs y comentarios en el código).

Como requerimiento general se acordó la implantación y testeo de ambos sistemas en el Hospital de Clínicas así cómo lograr el funcionamiento correcto con al menos un sistema existente dentro del mismo. Para lograr esto se deberán implementar los módulos necesarios para la comunicación entre los sistemas existentes (desarrollados en Genexus) y el servicio web con seguridad habilitada perteneciente al Sistema de Registro de Eventos.

6.3 Descripción de la Solución Planteada

La solución planteada consiste en el diseño e implementación de dos sistemas que funcionan conjuntamente para registrar y auditar mensajes de auditoría provenientes de los sistemas existentes:

- Sistema de Registro de Eventos.
- Sistema de Auditoría de Eventos.

A su vez se implementarán los siguientes módulos:

- Módulo Generador/Interprete de Mensajes ATNA
- Módulo de Comunicación
- Módulo Integrador Genexus

El módulo generador/interprete de mensajes ATNA se implementará como una biblioteca, la cual será utilizada por el sistema de registro de eventos, como por todo aquel cliente que desee comunicarse con dicho sistema. Esta biblioteca tiene como objetivo la generación de mensajes en formato XML según la RFC 3881 de manera fácil e intuitiva, evitando tener que conocer el esquema XML. Lo único que deberán conocer es la forma correcta de utilizar la biblioteca, lo cual garantiza la generación de un mensaje de auditoría válido.
El módulo de comunicación tiene como objetivo brindar soporte para la comunicación con el servicio web del sistema de registro de eventos utilizando los mecanismos de seguridad definidos por WS-Security.
Por último el módulo integrador será implementado con la herramienta Genexus y su principal objetivo es brindar un conjunto de primitivas para que los desarrolladores Genexus del Hospital de Clínicas puedan generar y enviar mensajes de auditoría de manera rápida y sencilla. Cabe señalar que este módulo hará uso de los dos anteriores.

6.3.1 Sistema de Registro de Eventos

La Figura 19 muestra el diagrama de componentes para el sistema de registro de eventos:

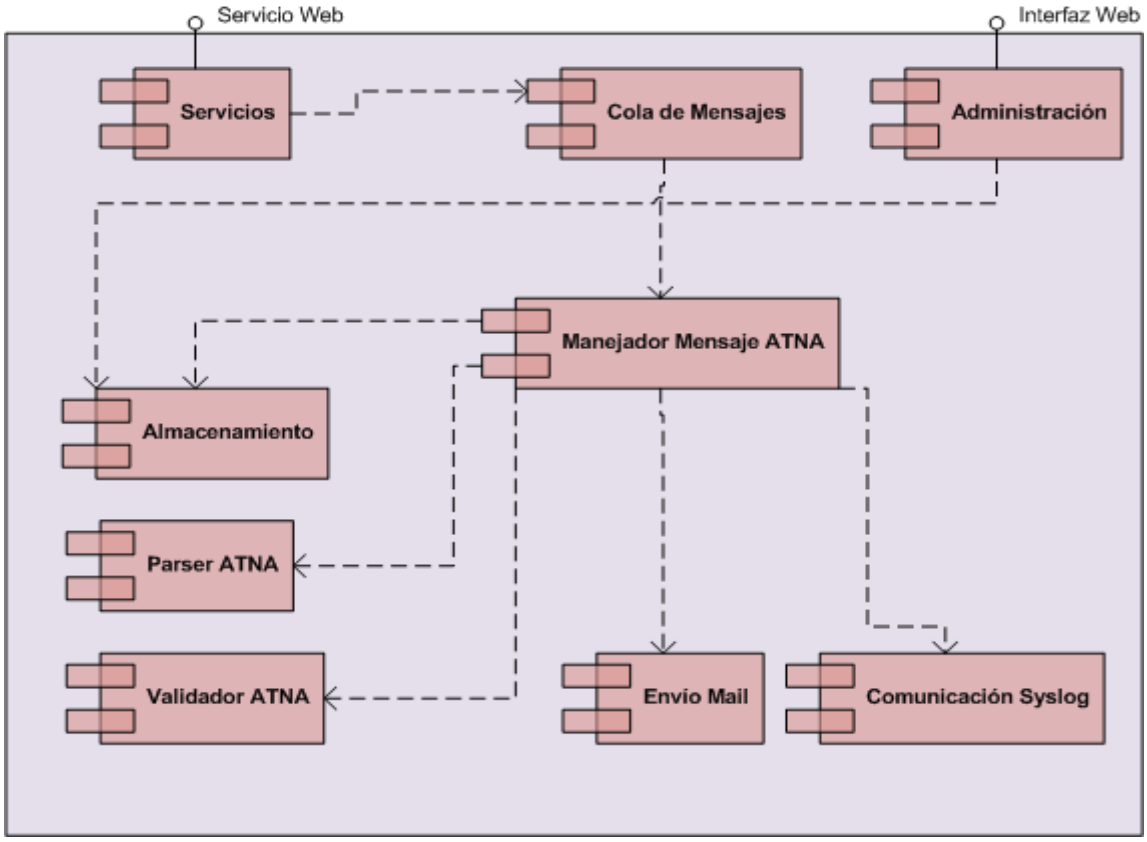


Figura 19 - Diagrama de componentes del Sistema de Registro de Eventos

Módulos del sistema

1. **Servicios**

   Presenta la interfaz de entrada al sistema de registro de eventos, mediante la definición de un servicio web. Se encarga también de los aspectos de seguridad del servicio web, y su función es comunicar el mensaje recibido al módulo de Cola de Mensajes.

2. **Cola de Mensajes**

   Se encarga de encolar los mensajes para luego ser procesados por el manejador de mensajes. Esto permite separar los componentes de servicio del manejador de mensajes, pudiendo ambos ubicarse físicamente en servidores distintos, favoreciendo la distribución del sistema. Por otro lado esto permite procesar el mensaje asincrónicamente, liberando al servicio tan pronto como el mensaje sea colocado en la cola, para poder atender nuevos pedidos, mejorando la disponibilidad del sistema.

3. Manejador Mensaje ATNA

Es el encargado de coordinar todas las acciones para completar el procesamiento del mensaje recibido, entre ellas validación, procesamiento y almacenamiento del mensaje de auditoría. Para ello delegará tareas a cada uno de los componentes, tal como indica la Figura 19.

4. Parser ATNA

Se encarga de parsear el mensaje de auditoría, el cual se encuentra en formato XML. Esto permite extraer todos los campos del mismo para su posterior análisis, validación y almacenamiento.

5. Validador ATNA

Consiste en la validación del mensaje contra el esquema XML definido por ATNA. Los campos que no se validan en el esquema mediante el uso de restricciones (ver sección **¡Error! No se encuentra el origen de la referencia.**) se validaran manualmente, por ejemplo direcciones IP, etc.

6. Envío Mail

Permite crear mails programáticamente para luego ser enviados en forma automática a los destinatarios seleccionados. Para ello el componente deberá permitir la configuración de los parámetros necesarios, tales como IP y puerto del servidor de email, nombre de usuario y contraseña, etc.

7. Comunicación Syslog

Se encarga de implementar el formato de mensaje Syslog y colocar dentro del mismo la información que se desea. A su vez establece la conexión con el servidor Syslog utilizando tanto el protocolo TCP como UDP. La decisión de cual protocolo utilizar quedará a cargo del administrador del sistema, el cual podrá configurarla mediante la consola de administración.

8. Almacenamiento

Es el componente encargado de establecer la conexión con el servidor de base de datos para el almacenamiento de la información asociada al mensaje de auditoría.

9. Administración

Provee una interfaz para administrar los usuarios y roles para el ingreso al sistema. A su vez, permite configurar todos los parámetros del sistema, como por ejemplo la cuenta de email, servidor, tipos y códigos de los eventos a registrar.

La Figura 20 presenta el diagrama de secuencia para el registro de evento desde un sistema externo:

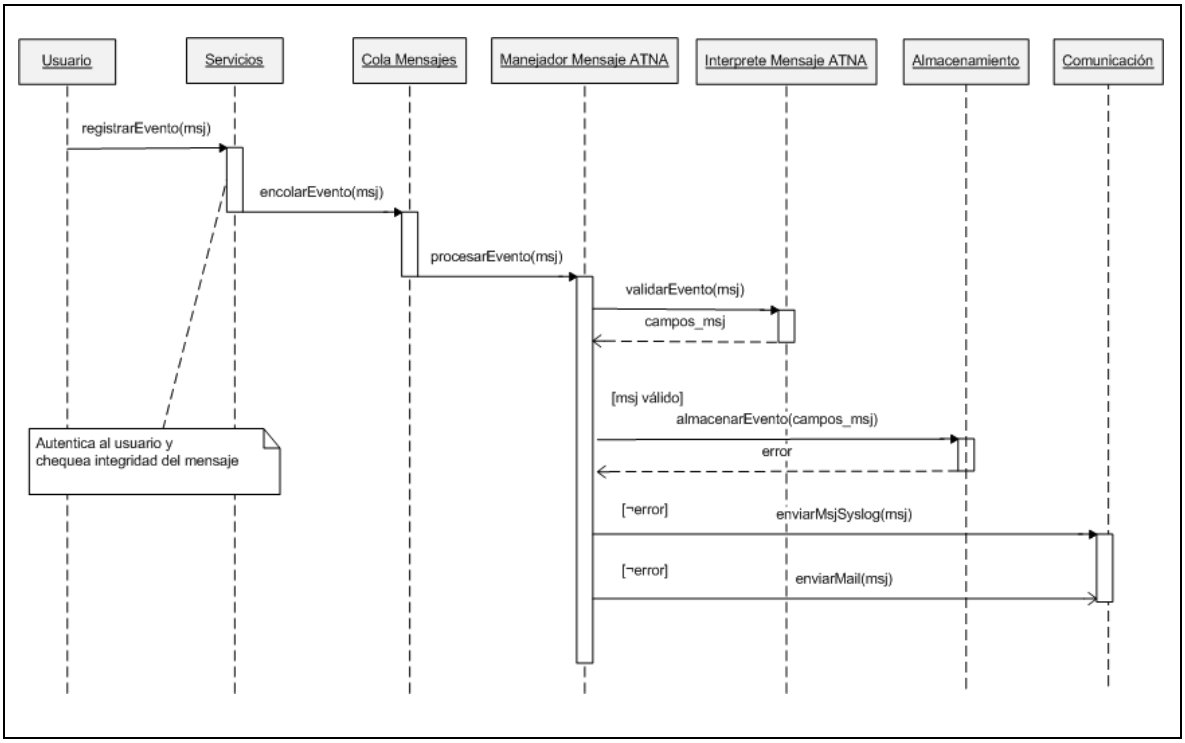


Figura 20 - Diagrama de Secuencia para el registro de evento

Como se aprecia en la Figura 20, el usuario consume el servicio para registro de eventos, enviando el mensaje de auditoría correspondiente. El componente de servicios se encarga de validar al usuario y verificar la integridad del mensaje por si existieron alteraciones del mismo durante el trayecto entre el cliente y servidor.

La cola de mensajes recibe y direcciona el mensaje al manejador de mensajes quien lo procesará una vez que este libre, ya que podría encontrarse procesando un mensaje anterior, en cuyo caso la cola de mensajes se encargará de mantener el mensaje para evitar la pérdida del mismo.

6.3.2 Sistema de Auditoría de Eventos

La Figura 21 detalla el diagrama de componentes para el sistema de Auditoría de Eventos:

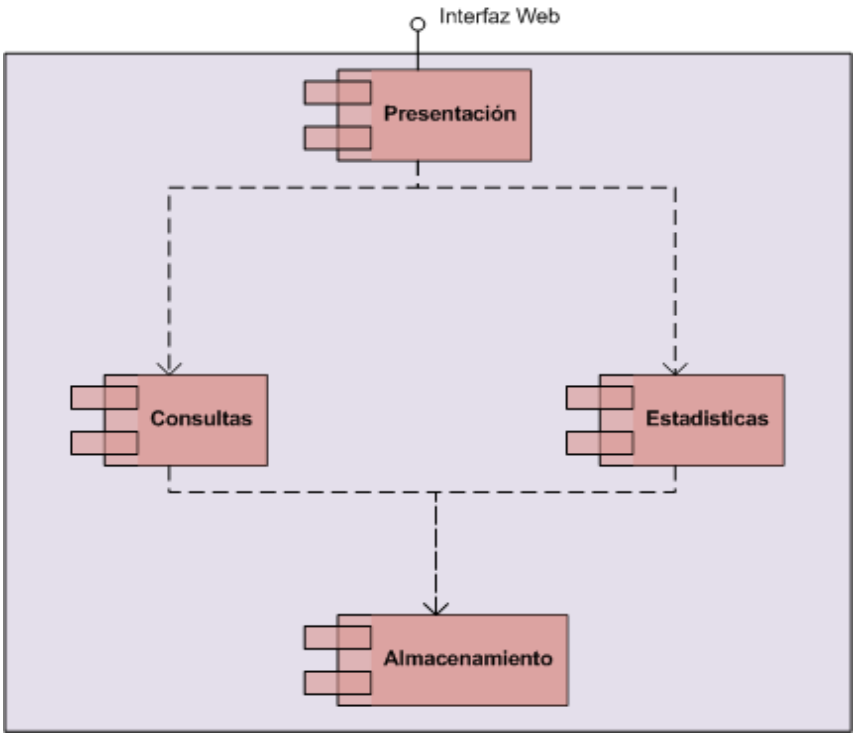


Figura 21 - Diagrama de componentes del Sistema de Auditoría de Eventos

Módulos del sistema

1. Presentación

Contiene el conjunto de páginas web que podrán ser visualizadas por los usuarios del sistema, el flujo de acceso entre las páginas.

2. Consultas

Implementa las consultas de información de los eventos, teniendo en cuenta los criterios de búsqueda establecidos por el usuario. Además se encarga de la impresión de reportes de auditoría en los formatos establecidos en los requerimientos del DPI (ver sección 4.2). La consulta de eventos puede realizarse por cualquiera de los cuatro campos fundamentales que define el la RFC 3881, a saber:

- Identificación del Evento.
- Identificación de los Participantes que formaron parte del evento a auditar.
- Identificación de las Fuentes donde se originó el evento.
- Identificación de los Objetos involucrados.

Cada sección contiene filtros de búsqueda que permiten acotar las búsquedas. Por otra parte una vez localizado el evento, es posible visualizarlo en detalle mediante una pantalla que despliega toda la información existente en la base de datos para dicho evento.

3. Estadísticas

Se encarga de obtener la información a partir de los parámetros establecidos por el usuario, para luego desplegarla gráficamente. La información obtenida es transformada y agregada al gráfico, de manera que el usuario podrá visualizar diversos conjuntos de datos (correspondientes a los parámetros seleccionados) en un mismo gráfico. Los parámetros principales son los siguientes:

- Rango de Fechas.
- Modalidad de Consulta (diaria, mensual y anual).
- Identificador de Participante.
- Identificador de Fuente.
- Identificador de Objeto.
- Punto de Acceso de Red.

4. Almacenamiento

Tiene como objetivo proveer aquellas consultas a la base de datos que sean necesarias para implementar la auditoría de eventos.

6.3.3 Módulo Generador de Mensajes de Auditoría ATNA

La Figura 22 muestra el diagrama de componentes para la biblioteca de mensajes ATNA:

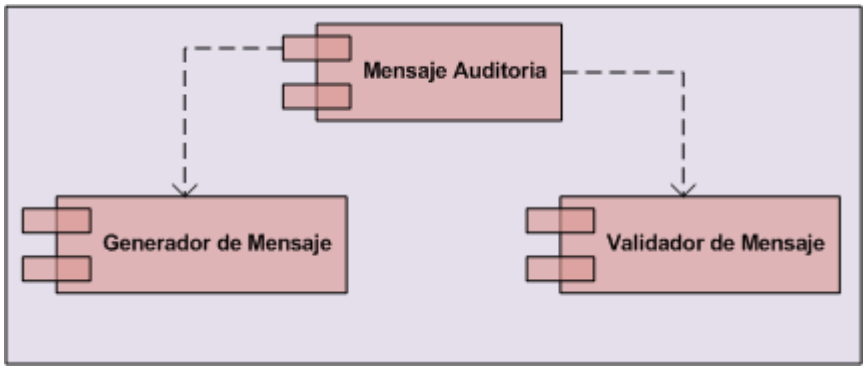


Figura 22 - Diagrama de componentes del Generador de mensajes ATNA

Módulos del sistema

1. Mensaje Auditoría

Representa la estructura de un mensaje de auditoría ATNA. Es el componente accesible para el usuario final, y dispone de dos grandes operaciones: generar el propio mensaje, así como validarlo. Esto último es de utilidad si no se tiene la seguridad de haber construido correctamente el mensaje de auditoría.

2. Generador de Mensaje

Obtiene todos los campos y valores del mensaje de auditoría y genera el archivo XML correspondiente.

3. Validador de Mensaje

Su objetivo es validar la estructura del mensaje de auditoría construido por el usuario

6.3.4 Módulo de Comunicación

El módulo de comunicación se encarga de establecer contacto con el web-service del sistema de registro de eventos, ya sea con o sin seguridad habilitada. El módulo contiene un archivo de configuración en el cual deberá indicarse la ruta local al almacén de claves (keystore) y de certificados confiables (truststore). En el almacén de claves deben encontrarse la clave y certificado que utilizará el sistema cliente para validarse contra el sistema de registro de eventos, y en el almacén de certificados confiables debe ubicarse el certificado correspondiente al sistema de registro de eventos. El módulo utiliza comunicación asíncrona con el servicio web, de manera que una vez finalizada la comunicación el sistema cliente seguirá con su normal funcionamiento, mientras que en paralelo el sistema de registro de eventos se encargará de procesar y almacenar el mensaje de auditoría recibido. Esto agiliza de gran manera la aplicación cliente, ya que en sistemas de uso diario (manejo de pacientes, etc) se requiere óptima performance. El módulo de comunicación utiliza el proyecto Metro (stack de servicios web de Glassfish) para lograr la comunicación con el servicio web.

6.3.5 Módulo Integrador Genexus

Este módulo tiene como objetivo eliminar las dificultades a la hora de integrar los dos módulos anteriores a los sistemas desarrollados con la herramienta Genexus. Genexus provee mecanismos para incluir código Java dentro del propio código Genexus, pero los mismos no son triviales y la mayoría de los desarrolladores no tienen el conocimiento necesario, ya que la documentación existente es escasa.

Por otro lado este tipo de mecanismo hace al código menos legible y mantenible ya que se produce una mezcla de código Genexus (que posteriormente será transformado en código Java) con el código Java necesario para invocar a los módulos descritos anteriormente.

Por esta razón se decide implementar dicho mecanismo brindándole al desarrollador Genexus una interfaz sencilla para construir el mensaje de auditoría, validarlo y posteriormente enviarlo.

7 Diseño e Implementación

En este capítulo se presenta el detalle de cómo se diseñó e implementó la solución, especificando las tecnologías y lenguajes utilizados. También se describen las decisiones de diseño adoptadas.

7.1 *Sistema de Registro de Eventos*

A continuación se especifican los casos de uso relevantes para la arquitectura del Sistema de Registro de Eventos.

7.1.1 Caso de Uso “Registrar Evento”

Diagrama de Caso de Uso

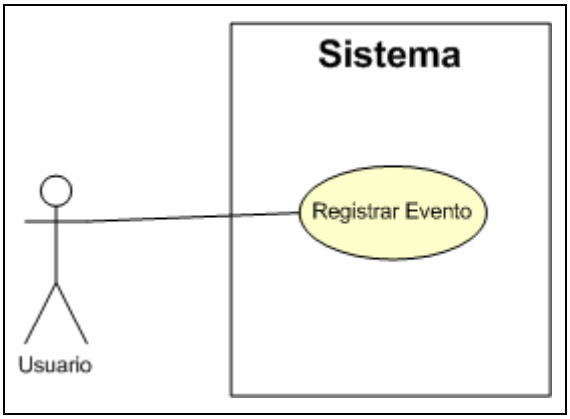


Figura 23 - Diagrama de Caso de Uso

Análisis de Caso de Uso

Nombre	Registrar Evento
Objetivo: Captar los eventos que se han definido como relevantes para su posterior análisis.	
Actores: Sistema externo	
Precondiciones: No tiene	
Descripción: El caso de uso comienza cuando el usuario envía un mensaje de auditoría al sistema. Este	

usuario generalmente será un sistema externo al Sistema de Registro de Evento. El sistema recibirá el mensaje y lo primero que hará será comprobar que el emisor del mensaje sea un sistema de los existentes en el hospital. Luego verificará que el mensaje no haya sido alterado. Ambos chequeos se llevaran a cabo utilizando los certificados digitales disponibles en la infraestructura de clave pública del hospital, asumiendo que un sistema externo es conocido si su certificado forma parte del almacén de certificados confiables del Sistema de Registro de Eventos.

Si estos dos chequeos se cumplen, entonces se procede a validar el mensaje y su contenido según el esquema de la RFC 3881. Si es válido, se interpreta para extraer los datos necesarios desde el mismo. Todos los datos obtenidos se almacenaran en la base de datos para su posterior auditoría. En caso de existir comunicación con un servidor Syslog se deberá crear un mensaje en ese formato y enviarlo. El mensaje debe contener en el cuerpo la información básica que permita recuperar el mensaje de auditoría almacenado en base de datos, en caso que el mismo haya sido eliminado por algún motivo. El formato elegido para el cuerpo del mensaje Syslog es XML y deberá respetar el límite en el tamaño del mensaje Syslog. El sistema deberá permitir configurar la ruta y el puerto en donde ubicar al servidor Syslog, así como el protocolo de transporte (UDP o TCP) por el cual viajaran los datos. Por último si el tipo de evento indica que es de gravedad o urgencia se deberá enviar un correo electrónico a los administradores del sistema indicando el identificador del evento para que los auditores puedan localizarlo fácilmente desde el Sistema de Auditoría de Eventos.

Cabe aclarar que el control es retornado una vez se recibe el mensaje de auditoría. Esto permite liberar al sistema cliente para que pueda seguir su normal funcionamiento evitando las posibles demoras que podrían ocurrir durante el procesamiento del evento en el servidor.

Flujo normal:

1. Usuario: Envía un mensaje de auditoría con la información de un evento al sistema.
2. Sistema: Verifica la identidad del usuario y la integridad del mensaje de auditoría.
3. Sistema: Extrae, valida y procesa la información del evento contenida en el mensaje de auditoría.
4. Sistema: Almacena la información en base de datos para posterior auditoría.
5. Sistema: Genera un mensaje en formato Syslog conteniendo la información necesaria para identificar el evento, y lo envía al servidor Syslog.
6. Sistema: Verifica la prioridad del evento, y en caso que corresponda envía un correo electrónico de alarma al administrador o auditor/es indicando el identificador del evento.

Flujos alternativos:

2. El sistema no valida las credenciales del usuario o la integridad del mensaje y cancela el procesamiento del mensaje.
3. El sistema encuentra errores de formato o contenido en el mensaje y cancela la operación.
4. El sistema detecta errores o inconsistencias en el almacenamiento del evento y cancela la operación deshaciendo todas las modificaciones realizadas.
5. El sistema envía mensaje al servidor Syslog utilizando el protocolo TCP pero este no responde, por lo que el sistema cancela el envío del mismo, quedando éste únicamente en la base de datos del servidor de registros.
6. El sistema envía alerta vía mail pero la comunicación falla o el usuario no es autenticado por lo que se cancela el envío.

Poscondiciones:

7.1.2 **Arquitectura**

Se propone una arquitectura en capas, de forma de tener encapsuladas y bien definidas las responsabilidades de cada una. Para implementar correctamente el caso de uso descrito anteriormente se definió la siguiente arquitectura distribuida, representada en la Figura 24:

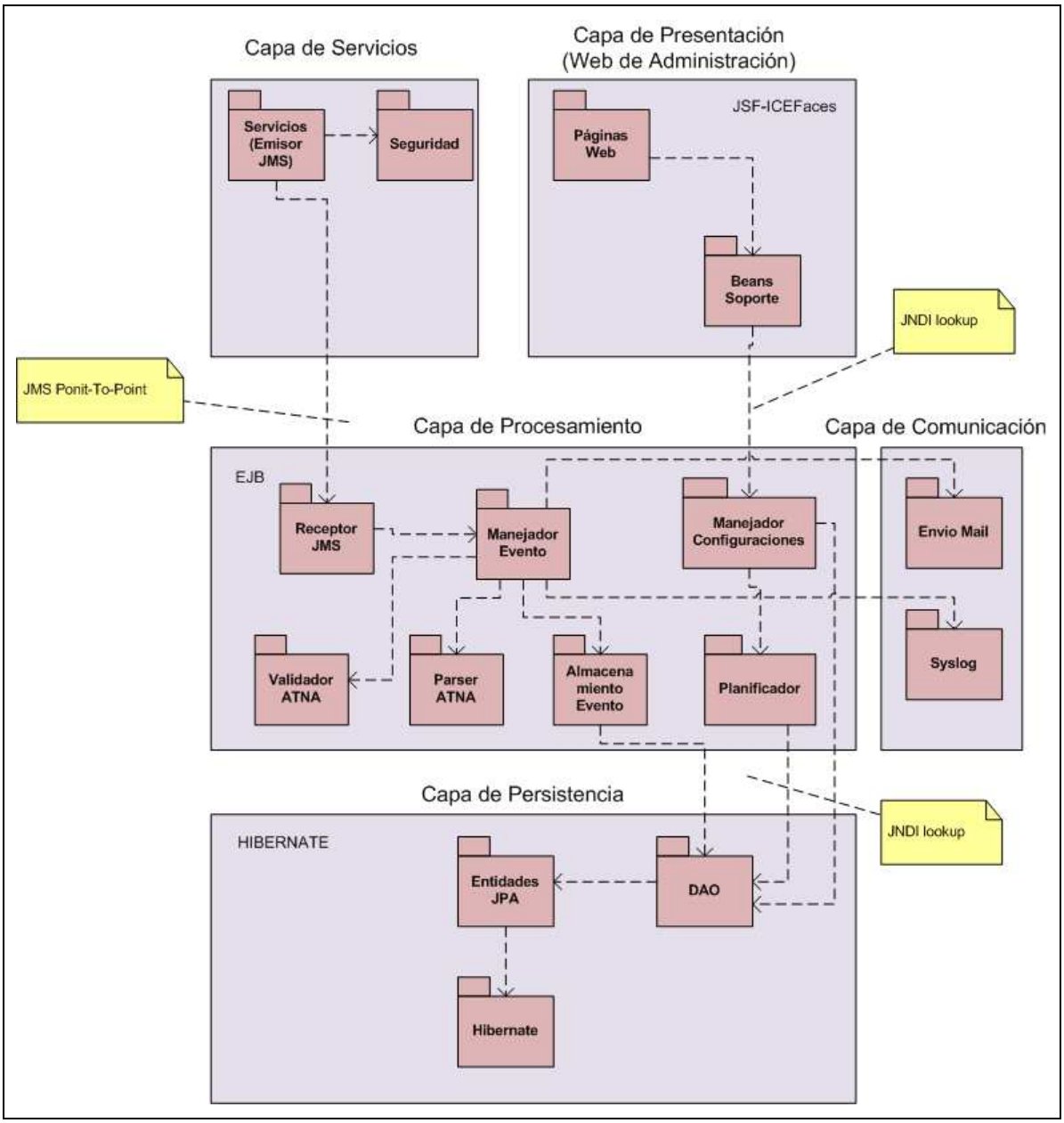


Figura 24 - Diagrama de Arquitectura del Sistema de Registro de Eventos

A continuación se describen cada una de las capas:

Capa de Servicios

Contiene el servicio web que servirá de entrada al sistema. A través de ésta capa se accederá al sistema. Deberá tener en cuenta los requerimientos de seguridad a la hora de recibir mensajes.

Capa de Presentación (Administración)

La capa de presentación corresponde a la consola de administración del sistema de registro de eventos. Contiene las páginas web necesarias para permitir visualizar y modificar las configuraciones, identificadores y tipos de eventos recibidos, así como los roles de los usuarios que los envían. Se permite tanto una carga manual, como automática a través de archivos XML que contengan toda la información necesaria. El fin de estas funcionalidades radica en que el sistema de registro de eventos disponga de todos los códigos definidos para cada evento y para cada tipo de evento, en cada uno de los sistemas que comenzarán a registrar eventos, para realizar las validaciones de los mismos. Aquellos códigos de evento y tipo de evento que se envíen en un mensaje de auditoría desde un sistema cliente y no coincida con alguno de los códigos existentes en el sistema de registro de eventos, no validará y por lo tanto quedarán indefinidos. Esto se podrá visualizar desde el sistema de auditoría de eventos, para el evento en cuestión. Lo mismo sucederá para los códigos de rol de los usuarios de cada uno de los sistemas cliente.

Capa de Procesamiento

En esta capa se procede a validar y extraer los campos del mensaje XML recibido. Interactúa con la capa de comunicación para enviar el mensaje al servidor Syslog y para enviar notificaciones vía correo electrónico. A su vez se comunica con la capa de almacenamiento para persistir los datos del mensaje de auditoría.

Capa de Comunicación

Es la encargada de crear las conexiones http para la comunicación y envío de mensajes al servidor Syslog. Por otra parte crea la sesión de correo para el envío de los mismos.

Capa de Almacenamiento

Es la encargada de mapear los objetos existentes (entidades) en la aplicación con las tablas del modelo relacional definido en la base de datos (mapeo entidad-relación). También deberá establecer la conexión con el manejador de base de datos para realizar las operaciones de persistencia y consulta de datos.

7.1.3 Aspectos de diseño y tecnológicos de cada componente

Servicios

El componente de servicios implementa la interfaz del sistema de registro de eventos con el resto de los sistemas existentes. Como se aprecia en la Figura 24 , el componente de servicios actúa como emisor de mensajes en el modelo de mensajería punto a punto. Por ende la comunicación entre el emisor y el receptor se realiza mediante una cola destinada a almacenar temporalmente los mensajes. Las tecnologías involucradas en este componente son JAXWS y JMS.

El servicio web fue implementado como EJBs Stateless (sin estado), quedando a cargo del servidor de aplicaciones la creación y destrucción de instancias del mismo. Una instancia se creará para cada petición al servicio web y el número máximo de instancias que podrán ejecutar concurrentemente dependerá de la configuración del servidor de aplicaciones. Esta configuración podrá ser modificada, teniendo en cuenta las prestaciones del servidor, para disponer de un pool de EJB más amplio, y por ende una mayor disponibilidad.

Por otro lado la introducción de la cola de mensajes tiene como objetivo minimizar el tiempo de ejecución de las instancias del servicio web. Esto se traduce en una mayor disponibilidad del servicio, ya que todo el trabajo de procesamiento del mensaje queda a cargo de la cola de mensajes, la cual puede ser instalada en un servidor dedicado.

Seguridad

Implementa los mecanismos necesarios para disponer de comunicación segura y confiable desde los sistemas externos hacia el Sistema de Registro de Eventos.

El mecanismo de seguridad utilizado es el de autenticación mutua con certificados, el cual brinda autenticación tanto del cliente como del servidor, confidencialidad e integridad en los datos enviados. La utilización de este mecanismo implica obtener los certificados de cada uno de los clientes y del servidor de registro de eventos. Los primeros deberán cargarse en el almacén de certificados confiables del servidor, mientras que el último deberá cargarse en el almacén de certificados confiables de cada cliente. La clave privada del servidor deberá almacenarse dentro del almacén de claves del mismo, y la clave de cada cliente deberá almacenarse en el almacén de claves correspondiente a cada cliente. La tecnología utilizada para lograr la implementación de este mecanismo se denomina WSIT [60], desarrollada por Sun [61], la cual implementa los mecanismos de seguridad definidos por el estándar WS-Security (ver sección 0), y tiene como objetivo lograr interoperabilidad con tecnologías Microsoft.NET 3.0. Por mayor detalle de la implementación referirse a [anexo 8].

Páginas Web (Administración)

Este componente contiene el conjunto de páginas que conforman la consola de administración del Sistema de Registro de Eventos. Por otra parte debe definir mecanismos de autenticación para el ingreso al sistema, así como la navegación entre páginas. Se utilizó JSF e ICEFaces para la implementación de las páginas y la definición de las navegaciones mediante un archivo XML.

Las funcionalidades que pueden accederse desde la interfaz web son las siguientes:

- Alta, Baja y Modificación de los códigos de evento que manejará el sistema.
- Alta, Baja y Modificación de los códigos de rol de participante que manejará el sistema.
- Alta, Baja y Modificación de los códigos de los tipos de evento que serán utilizados por el sistema.
- Carga masiva de todos los datos detallados en los puntos anteriores mediante archivos XML.
- Exportación de todos los datos del sistema a archivos XML, los cuales podrán ser descargados por el administrador del sistema.
- Alta y baja de esquemas XML para validar el formato de los mensajes de auditoría. Por defecto el sistema tiene cargado el esquema definido por la RFC 3881.
- Configuración de los tiempos de almacenamiento de los mensajes de auditoría en la base de datos del sistema, según la severidad de los mismos.
- Configuración de la sesión para envío de email, así como las condiciones que deberán cumplir los mismos para que el envío se lleve a cabo (ver Figura 25).
- Configuración del respaldo de mensajes en el servidor Syslog.

Beans de Soporte

Este componente contiene la implementación de cada EJB (Managed Beans) necesario para mantener el estado de sesión del usuario. La estrategia que se definió consiste en disponer de un managed bean por cada una de las páginas del sistema, las cuales se corresponden con una funcionalidad del mismo. Cada instancia se creará al inicio de la sesión del usuario y estará disponible hasta que se cierra la misma, momento en el que es destruido, liberando todos los recursos reservados por el mismo. Los recursos utilizados durante la sesión serán cacheados en un managed bean específico, el cual contendrá las referencias a todos los recursos utilizados, lo cual en la práctica se comprobó que reduce el tiempo de respuesta del sistema considerablemente.

Receptor

Es quien recibe los mensajes existentes en la cola de mensajes, según el modelo punto a punto de JMS. Su función es extraer el mensaje de auditoría (representado en un XML) desde el mensaje manejado por JMS y enviárselo al manejador de mensajes ATNA. Para la implementación se utilizó los denominados Message Driven Beans.

Manejador de mensajes ATNA

Es el componente central de la arquitectura ya que se encarga de coordinar al resto de los componentes para lograr el procesamiento correcto de cada mensaje de auditoría recibido. Dada la alta disponibilidad que podría requerir este componente se decidió implementarlo como un Enterprise Bean de sesión sin estado (Stateless Session Bean). Escogiendo un tamaño adecuado para el pool de beans en el servidor, logramos performance y disponibilidad al mismo tiempo. Se comunica con los restantes componentes mediante la inyección de dependencias (tecnología que suplanta las búsquedas JNDI utilizando lookups).

Manejador de Configuraciones

Tiene como objetivo resolver las operaciones de modificación y consulta de las configuraciones existentes.

También es el encargado de comunicarse con la capa de persistencia para realizar las operaciones de alta, baja, consulta y modificación de tipos de eventos en el sistema. Los tipos de evento definidos en el sistema disponen de dos campos (facilidad y severidad) los cuales son utilizados al momento de armar el mensaje Syslog. Esto requiere que el tipo de evento que viene en el mensaje exista en el sistema de registro de eventos. Caso contrario podrá ser dado de alta mediante la interfaz de administrador. En caso que no se encuentre se le asignará una severidad de valor 8, la cual representa un evento desconocido.

Por otra parte, tanto los códigos de evento, como los roles de los participantes involucrados en el mismo, podrán ser dados de alta, modificados o eliminados. Esto es de utilidad para que el sistema valide ambos campos del evento determinando si son eventos originados por sistemas del hospital de clínicas y por usuarios correctamente identificados. Entre otras configuraciones que podrán modificarse se encuentran las de sesión de email, y comunicación con el servidor Syslog.

Planificador

Este componente se encarga de planificar la ejecución de aquellas tareas que el administrador del sistema configure, entre ellas se destaca la eliminación de aquellos eventos que se consideren prescindibles, ya sea por su antigüedad, por su prioridad, o por una combinación de ambas. Como se aprecia en la Figura 24 el manejador de configuraciones se comunica con el planificador, indicándole qué tareas ejecutar y en que intervalo de tiempo, así como las condiciones de ejecución. Para la implementación de este componente se utilizó la biblioteca Quartz [80], la cual brinda las funcionalidades necesarias para poder planificar tareas de manera flexible y confiable, permitiendo definir cualquier tipo de acciones a tomar dentro de cada tarea, y respaldando las mismas en base de datos para poder retomar la ejecución en casos de fallas de hardware o desastres que impidan la normal ejecución. El módulo Planificador se comunica con el componente DAO para realizar las modificaciones en la base de datos, esto es, por ejemplo, eliminar los eventos correspondientes.

Manejador de Almacenamiento

Este componente implementa un planificador, el cual se inicializará al momento de desplegarse la aplicación en el servidor, y ejecutará una tarea cada cierto tiempo, configurable por el administrador del sistema. El objetivo de la tarea es eliminar aquellos mensajes de auditoría que son considerados prescindibles, con el fin de acotar el tamaño de la base de datos, lo cual repercute en una mejora en la performance del sistema de auditoría de eventos. Para determinar qué mensajes serán eliminados el sistema tomará en cuenta el límite de tiempo de almacenamiento de cada mensaje, según su severidad. Dicho límite puede variar en el rango de horas, días, meses o años, y quedará a decisión del administrador asignar cada uno de ellos. Para la implementación de este componente se utilizó el planificador de tareas Quartz [80], el cual permite definir tareas y ejecutarlas periódicamente, cada cierto tiempo configurable por el desarrollador o dinámicamente por el usuario del sistema.

Validador ATNA

Se encarga de validar el mensaje de auditoría en formato ATNA contra el esquema que se esté utilizando actualmente. Para la implementación se utilizó la biblioteca JDom [56] ya que la misma implementa dicha validación tomando como entrada el XML y el esquema correspondiente. Los campos que no se validan a partir del esquema (porque no existen restricciones en él), como por ejemplo la IP de la máquina que originó el mensaje son validados manualmente.

Parser ATNA

Su principal función es traducir el mensaje de auditoría en un conjunto de objetos extrayendo todos los campos existentes. Cada porción del mensaje se mapea con su respectivo objeto en el sistema para más adelante ser procesado. Se utiliza la biblioteca mensajeATNA.

Envío de Mail

Obtiene las configuraciones correspondientes a la sesión de email y realiza el envío de los mismos si este se encuentra habilitado. Para la implementación se utilizó javaMail, tecnología que permite crear mensajes de correo indicando emisor, destinatarios, asunto y cuerpo del mensaje. Se permite seleccionar si se requiere autenticación o no (esto dependerá de la configuración del servidor de correo). Los emails serán enviados si el envío está habilitado y si el tipo de evento asociado al mensaje de auditoría indica que se requiere el envío de email como mecanismo de advertencia de situación anómala. A su vez se verificarán las condiciones generales de envío, las cuales permiten especificar severidades, facilidades y fuentes de los mensajes que requieren alerta. A modo de ejemplo la Figura 25 presenta algunas condiciones de envío:

Severidad (Código)	Facilidad (Código)	Fuente (Código)
Emergencia (0)	Sistema (3)	Índice de Pacientes (0111)
Advertencia (3)	Log auditoría (13)	Todos
Todas	Todas	Todos

Figura 25 - Ejemplos de condiciones de alerta vía email

Como se aprecia en la Figura 25 una condición puede incluir un código específico para la severidad, facilidad y fuente donde se originó el evento. Cabe notar que con una única condición (la última en la tabla de la Figura 25) es posible abarcar todo el espectro de eventos. Los códigos de severidad y facilidad corresponden a los definidos por el protocolo Syslog, por lo tanto todo evento deberá tener asignados ambos, en caso de no existir una

facilidad y severidad definida para un evento, el sistema de registros asignará aquellas que corresponden a eventos desconocidos.

Syslog

Este componente se encarga de los aspectos relativos a la generación y envío de mensajes al servidor Syslog. El mensaje respeta el formato Syslog y en el cuerpo incluye un archivo XML que contiene los campos obligatorios del mensaje ATNA. Esto asegura un tamaño no superior a la cota definida por Syslog y a su vez permite identificar el evento si se realiza una búsqueda en dicho servidor. A su vez, la prioridad de cada mensaje viene dada por la facilidad y severidad de cada uno, información que deberá ingresar el administrador del sistema mediante la consola de administración.

Notar que la facilidad y severidad de un evento puede diferir de un sistema a otro, por lo tanto se deberá ingresar además de esos dos campos, el código del sistema. Para la comunicación se utilizaron sockets tanto sobre UDP como TCP. La implementación utilizada es la provista por la plataforma java (Socket para TCP y DatagramSocket para UDP). La decisión configuración queda a cargo del usuario administrador.

DAO

Implementa una fachada sobre las entidades del sistema, ofreciendo las operaciones de alta, baja, consulta y modificación para cada una de ellas. En los casos necesarios se añadieron nuevas funcionalidades como por ejemplo búsquedas según ciertos parámetros, etc. Para la implementación se utilizo JPA.

Entidades JPA

Son aquellas que se mapean directamente con las tablas de la base de datos relacional. Se utilizo JPA para su implementación y se configuro la aplicación para crear el esquema de base de datos al momento de desplegar la aplicación en el servidor. En la próxima sección se detallan las entidades en el modelo de datos utilizado.

7.1.4 Modelo de Datos

El modelo de datos se construyo programáticamente utilizando los objetos entidad que provee JPA. Una vez creado se configura la aplicación para impactar la estructura en la base de datos relacional, lo cual genera todas las tablas, índices, secuencias y relaciones existentes entre cada una de ellas. La Figura 26 muestra el diagrama de modelo de datos:

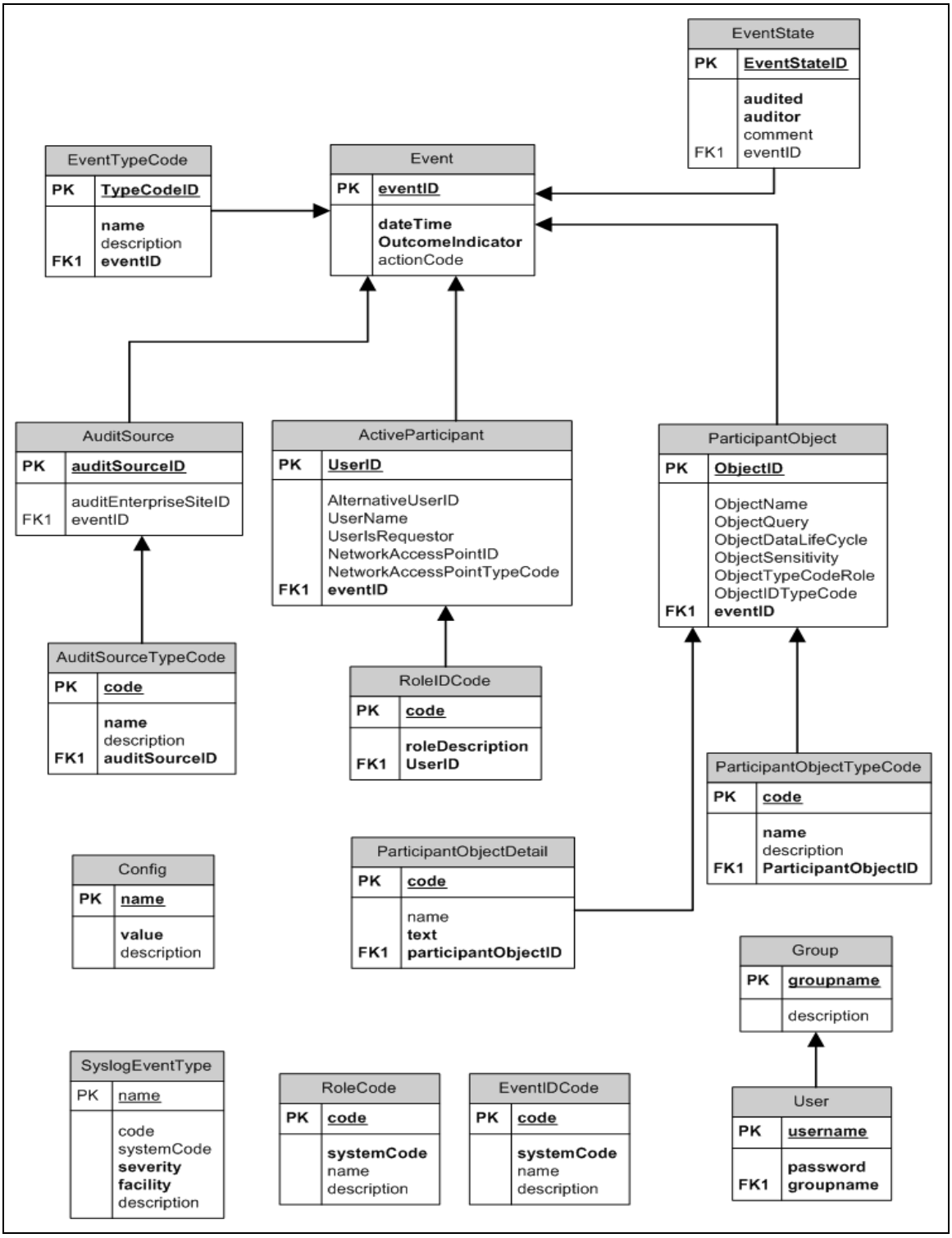


Figura 26- Diagrama de Modelo de Datos del Sistema de Registro de Eventos

Como se puede apreciar, el mismo fue construido tomando en cuenta el esquema XML provisto por ATNA para la definición del mensaje de auditoría. En negrita figuran los atributos obligatorios de las entidades. La tabla *Config* contiene todas las configuraciones del sistema, mientras que la tabla *User* y *Group* es utilizada por el servidor de aplicaciones para controlar el acceso a la interfaz de administración. El acceso se definirá mediante roles, donde cada rol tendrá permisos de acceso sobre ciertos recursos. *SyslogEventType* define

los tipos de evento de los que el sistema tendrá conocimiento. Cada uno de ellos tiene una severidad y facilidad asignada, de manera que al momento de enviar un mensaje al servidor Syslog, se pueda calcular su prioridad con la conocida formula: *prioridad = 8 * facilidad + severidad*. La tabla EventState mantiene el estado de cada evento (auditado/ no auditado) y será de utilidad para el sistema de auditoría. Cada nuevo evento deberá marcarse como no auditado. Por último las tablas RoleCode y EventIDCode serán cargadas con los códigos de rol de participantes y códigos de eventos respectivamente, de cada uno de los sistemas que requieran registrar eventos en el sistema de registros.
La Figura 27 indica el mapeo entre cada porción del esquema con el correspondiente objeto en el modelo de datos:

Elemento	Elemento Padre	Entidad	Entidad Padre	Multip. respecto al padre
AuditMessage	No tiene	No mapeado	No mapeado	No aplica
EventIdentification	AuditMessage	Event	No tiene	1 a 1
EventTypeCode	EventIdentification	EventTypeCode	Event	0 a N
AuditSourceIdentification	AuditMessage	AuditSource	Event	1 a N
AuditSourceTypeCode	AuditSourceIdentification	AuditSourceTypeCode	AuditSource	0 a N
ActiveParticipant	AuditMessage	ActiveParticipant	Event	1 a N
RoleIDCode	ActiveParticipantType	RoleIDCode	ActiveParticipant	0 a N
ParticipantObjectIdentification	AuditMessage	ParticipantObject	Event	0 a N
ParticipantObjectDetail	ParticipantObjectIdentification	ParticipantObjectDetail	ParticipantObject	0 a N
ParticipantObjectIDTypeCode	ParticipantObjectIdentification	ParticipantObjectIDTypeCode	ParticipantObject	1 a 1

Figura 27 - Mapeo entre Entidades y Elementos del mensaje ATNA

Se omitió el objeto AuditMessage que correspondería a la raíz del esquema, ya que este elemento no contiene información propia como pueden ser atributos, que sean de interés almacenarlos, por lo tanto al momento de realizar las búsquedas y consultas se puede prescindir de él y realizarlas directamente sobre los elementos hijos.

7.2 Sistema de Auditoría de Eventos

Aquí se presentan los casos de uso que definirán la arquitectura del sistema.

7.2.1 Caso de Uso “Auditoría de Evento"

Diagrama de Casos de Uso

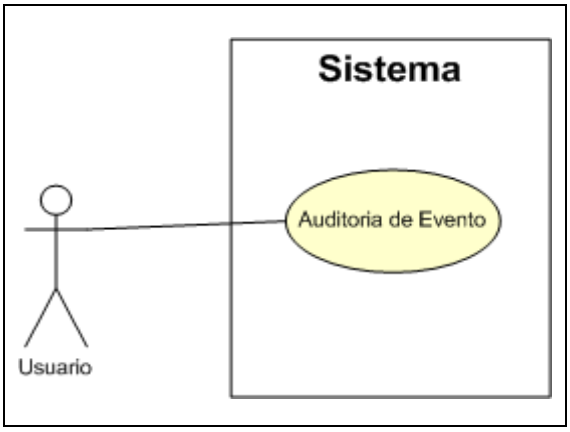


Figura 28 - Diagrama de Casos de Uso

Análisis del Caso de Uso

Nombre	Auditoría de Evento
Objetivo: Consultar la información de uno o varios eventos registrados en el sistema.	
Actores: Administrador, Auditor.	
Precondiciones: No tiene	
Descripción: Este caso de uso comienza cuando el usuario le indica al sistema que desea auditar los eventos existentes. El sistema despliega en pantalla los siguientes filtros que dispondrá el usuario para acotar la búsqueda: • Identificador del evento.	

- Rango de fechas en que se registró el evento.
- Usuario que registró el evento.
- Identificación de la máquina que originó el evento. Esto puede ser dirección IP, DNS, o teléfono.
- Código del tipo de evento.
- Nivel de falla.
- Código de acción del evento.
- Sistema desde el cual se envío el evento.

Una vez ingresados los filtros el sistema mostrará en pantalla la información correspondiente a cada registro que cumple las condiciones.

Flujo normal:

1. Usuario: Inicia sesión en el sistema.
2. Sistema: Autentica al usuario.
3. Usuario: Indica al sistema que desea consultar los eventos registrados hasta el momento.
4. Sistema: Autoriza al usuario a acceder al recurso solicitado.
5. Sistema: Despliega la pantalla de filtros.
6. Usuario: Ingresa los filtros que desea y luego confirma.
7. Sistema: Despliega en pantalla la grilla de eventos que cumplen los filtros de búsqueda.
8. Usuario: Selecciona un evento desde la grilla y consulta el detalle del mismo.
9. Sistema: Despliega en pantalla toda la información asociada al evento seleccionado.

Flujos alternativos:

2. El sistema no autentica al usuario redirigiéndolo a la pantalla de login nuevamente.
3. El usuario no tiene permisos para acceder al recurso solicitado por lo tanto el sistema despliega un mensaje de error denegando el acceso a dicho recurso.

7. El sistema no encuentra eventos que cumplan con los filtros y despliega un mensaje de advertencia al usuario.

Poscondiciones:

7.2.2 **Arquitectura**

Se definió una arquitectura en capas y distribuida tal cual propone la especificación java EE. Dicha arquitectura propone separar los aspectos de presentación de los de comportamiento. Una aplicación Java EE web básicamente se divide en tres capas. La capa de presentación la cual se ubica en el cliente y es accedida mediante los exploradores web, la capa de negocios situada en el servidor Java EE que encapsula toda la lógica de la aplicación y la capa de persistencia que se encarga de la comunicación con la base de datos. Las tecnologías utilizadas también deben ajustarse al estilo arquitectónico elegido. En este caso se utilizó el framework Java Server Faces, diseñado para este tipo de arquitecturas ya que sigue el modelo vista-controlador (MVC) el cual separa la capa de presentación en un conjunto de páginas web, mientras que el comportamiento y el manejo de estado de las acciones del usuario viene dado por los denominados *managed beans,* componentes que actúan como nexo entre la capa de presentación y la de negocios, transformando datos y realizando solicitudes hacia la capa de negocios según las acciones del usuario.

La Figura 29 ilustra la definición de las capas y los componentes de cada una:

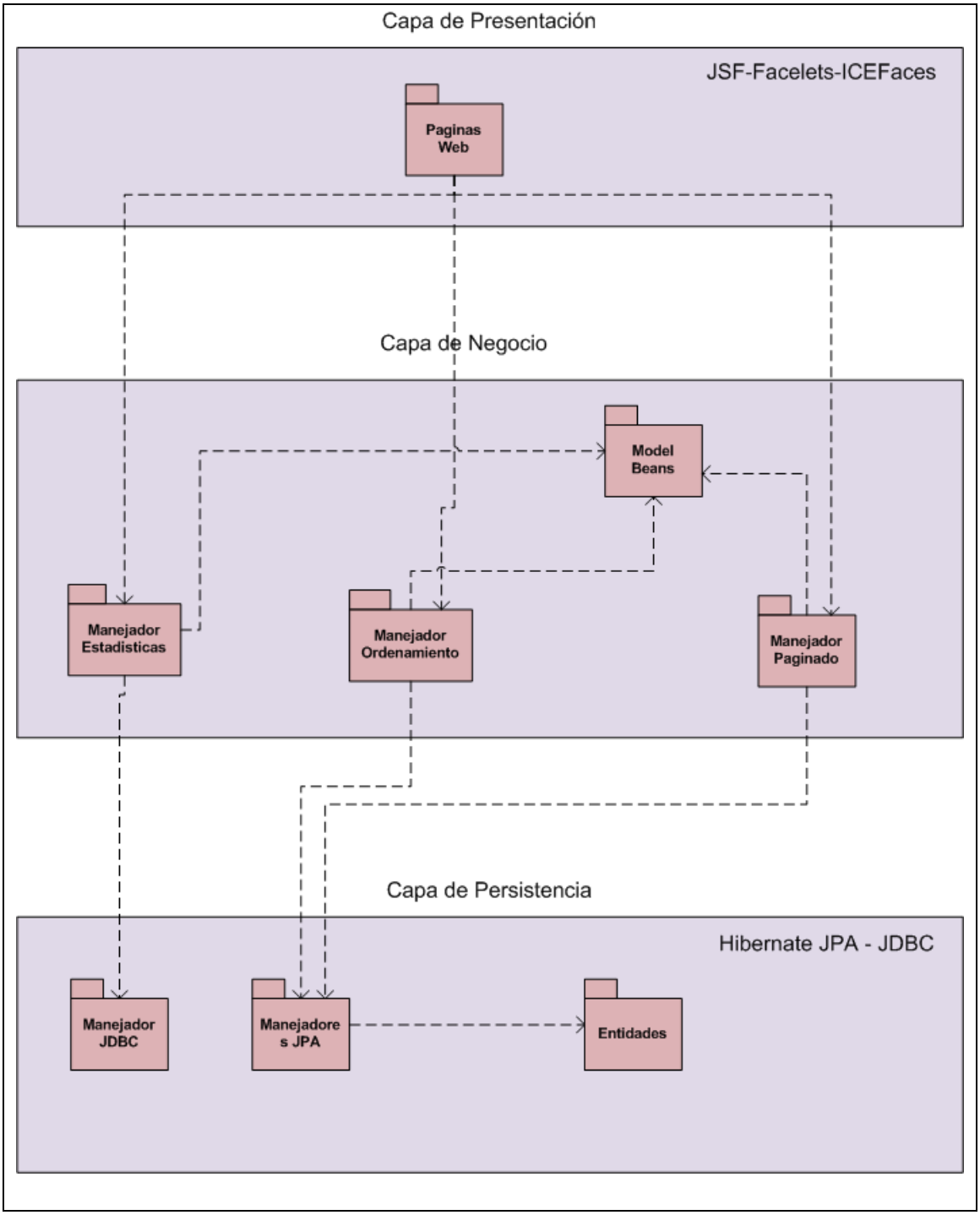


Figura 29 - Diagrama de Arquitectura del Sistema de Auditoría de Eventos

A continuación se describen los aspectos interesantes de cada capa y luego se detalla la funcionalidad de cada componente:

Capa de Presentación

Fue implementada como un componente web, el cual contiene todas aquellas páginas web, templates, hojas de estilo (css), imágenes y recursos utilizados por la aplicación para lograr la correcta visualización en el explorador web. A su vez define archivos de configuración para el control de acceso a la aplicación, definiendo los posibles roles de aquellos usuarios habilitados a acceder a la misma. Para la implementación se utilizó el framework Java Server Faces pero a su vez se integraron otras tecnologías como ICEFaces y Facelets, las cuales aportan características fundamentales como comunicación ajax con el servidor, mejoras en el aspecto visual de los componentes, nuevas técnicas de templating para facilitar la construcción y reutilización de páginas o porciones de páginas. Las páginas web fueron codificadas en HTML4 [48] y XHTML [49] según las necesidades. Este último está basado en el formato XML y por lo tanto puede ser editado, leído, visualizado o validado utilizando las herramientas estándar para el manejo de XML. Por otro lado brinda ciertos beneficios respecto a HTML4, como por ejemplo mayor velocidad de carga.

Capa de Negocio

Se puede dividir en dos subcapas, según las dos siguientes funcionalidades:

Manejo de Estado

El propósito de dicha subcapa es mantener el estado de cada sesión de usuario. Para ello, no se utiliza la sesión HTTP sino que se siguen los lineamientos sugeridos por la especificación Java EE, la cual recomienda el uso de los denominados Enterprise Java Beans en su versión Statefull (componentes con estado). Particularmente se utilizaron los denominados Managed Beans, los cuales conectan cada campo, tabla y datos de una página web con atributos de dichos componentes. Por otro lado permite asociar eventos a cada botón o componente web, permitiendo separar fácilmente la presentación del comportamiento. Los Managed Beans pueden ser configurados para mantenerse activos durante un pedido, una sesión o durante todo el transcurso de la aplicación. Configurándolo correctamente (en sesión por ejemplo) se logra mantener las acciones realizadas por los usuarios. Esto por ejemplo es muy útil al momento de realizar pasaje de parámetros desde una página a otra, ya que el/los parámetros se mantienen en el bean de sesión.

Otro ejemplo de uso de dichos componentes es el de visualización de contenido dinámico. Esta técnica se debe gracias a la tecnología Facelets ya que ésta permite definir un template xhtml dejando variable el cuerpo del mismo para que cambie según las navegaciones del

usuario. Entre otras tecnologías se utiliza ajax, ya que solo se recarga el cuerpo de la página principal con el contenido correspondiente. Para determinar qué contenido cargar se utiliza nuevamente un managed bean el cual está conectado a cada hipervínculo del menú principal y dependiendo de cual se seleccione se dispara el evento correspondiente con la consecuente carga de contenido. Cabe señalar que el contenido a cargar en el cuerpo de la página principal no es más que una página html y algunos tags de Facelets que definen qué porción de dicho html se va a cargar. Esto último es otro de los fuertes de Facelets, ya que el contenido se diseñó transparentemente como páginas html sin necesidad de utilizar lenguajes y tags nuevos.

Fragmento de código xhtml de la página principal:

```
<html>
<body>
    <ui:define name="content">
            <ice:form>
               <ui:include src="#{contentManagedBean.paginaActual}"/>
            </ice:form>
     </ui:define>
</body>
</html>
```

Fragmento de código del Managed Bean:

```
public class contentManagedBean implements Serializable {
   private String paginaActual = “paginaInicio.xhtml”;
   public void setPaginaActual(String pagina){
     this.paginaActual = pagina;
   }
   public String getPaginaActual(){
     return this.paginaActual;
   }
}
```

Negocio:

Dado que la aplicación se basa en consultas sobre un conjunto de datos, las funcionalidades básicas que se ofrecen sirven como soporte para la capa de presentación. Entre ellas se destacan:

- Ordenamiento de listas según campos, para realizar el ordenamiento por columnas en las tablas.
- Transformación de entidades en objetos utilizados por la capa de presentación para desplegar la información necesaria. Esta transformación es necesaria ya que las entidades provenientes de la capa de persistencia no siempre son fácilmente representables en pantalla, ya que pueden contener campos estructurados que deben ser adaptados, y a su vez deben contener otra información volátil para manejo de estado, como por ejemplo si está seleccionado en la tabla o no.
- Manejo del paginado para mejorar la performance, brindando a la capa de presentación, diferentes posibilidades, como por ejemplo acotar la cantidad de registros a cargar desde la base de datos en una consulta SQL, modificar la cantidad de registros que se visualizan por página, etc.
- Generación de la información y gráficos para cada una de las estadísticas requeridas, a partir de la información obtenida desde la base de datos.

Los manejadores que se observan en la Figura 29 se comunican con la capa de persistencia para obtener las entidades involucradas en la consulta, y luego las transforma a beans de modelo, los cuales contienen todos los datos de las entidad, agregándose las descripciones de cada campo a partir de los códigos que se encuentran en la entidad, según los mapeos definidos por ATNA. A su vez se cargan los campos del bean que definirán el estado del mismo, como por ejemplo si se encuentra seleccionado o no desde la grilla, o mediante un checkbox.

Para la implementación de las consultas a la base de datos, necesarias para obtener la información a representar en los gráficos, se utilizó la tecnología JDBC en lugar de Hibernate. Las pruebas realizadas sobre un conjunto de datos indicaron una mejor performance de JDBC sobre Hibernate. Esta superioridad se debe principalmente al overhead introducido por Hibernate en la transformación del lenguaje HQL (Hibernate Query Language) a SQL, y el mapeo de datos a objetos java. Por otra parte dado que la estructura para almacenar los resultados de los gráficos es sencilla (solo se requiere almacenar fecha

y cantidad de ocurrencias de cierto evento), es innecesario introducir mapeos de datos a objetos.

El proceso para generar un gráfico a partir de los filtros ingresados por el usuario consiste básicamente en cuatro pasos:

1. Obtener y validar los filtros introducidos por el usuario, por ejemplo Rango de Fechas.
2. Generar dinámicamente las condiciones de la consulta (cláusulas where) utilizando los filtros introducidos. Esto puede requerir, según los filtros, la generación de uniones (joins) entre diferentes tablas para obtener los datos del evento.
3. Ejecutar la consulta y recorrer el resultado de la misma para generar una estructura con los datos (fecha y cantidad de eventos que cumplen la condición).
4. Cargar el componente gráfico de ICEFaces con la estructura previamente cargada.
5. Visualización en la página web.

Los pasos 1 y 4 son llevados a cabo por el manejador de estadísticas (ver Figura 29 - Diagrama de Arquitectura del Sistema de Auditoría de Eventos), los pasos 2 y 3 son responsabilidad del manejador JDBC mientras que el paso 5 es efectuado por la capa de presentación.

Capa de Persistencia

Dada la cantidad de tablas y relaciones entre estas, se opto por implementar el modelo de datos utilizando la técnica de programación ORM (Object Relational Mapping) la cual convierte la información de la base de datos en objetos de sistema, proveyendo una abstracción para el desarrollador. La especificación de Java EE que unifica la manera en que funcionan las utilidades ORM es JPA (Java Persistence API).

Como implementación de JPA se utilizó Hibernate ya que brinda los siguientes beneficios sobre JDBC:

- Abstracción del modelo de datos sin necesidad que el programador codifique los mapeos entre tablas y objetos (Transparent Persistence).

- Cacheo de tuplas en la aplicación mejora la performance en los casos que se consulta muchas veces la misma información. Con JDBC dicho cacheo lo debe codificar el programador.

- Permite definir campos de versión para verificar que cada usuario tiene los datos actualizados. Por ejemplo si dos usuarios obtienen un mismo dato, y lo modifican, entonces la versión para ese dato cambiara con la actualización del primer usuario, mientras que al segundo no se le permite realizar la actualización ya que él contiene datos desactualizados.

- Lenguaje de consultas (HQL) permite realizar consultas polimórficas así como consultas SQL nativas, optimizando la performance de las mismas automáticamente.

De esta manera, la capa de persistencia implementa el mapeo entre las tablas de la base de datos en objetos, así como la relación entre estas. A su vez define un conjunto de manejadores en los cuales se encuentra la lógica necesaria para armar todos los tipos de consulta SQL que se requieren. Cabe señalar que se utilizó el lenguaje HQL orientado a objetos para realizar las consultas a la base de datos. Los manejadores realizan la consulta y devuelven una o varias entidades cargadas con la información correspondiente.

7.2.3 Modelo de Datos

El modelo de datos se comparte con el sistema de registro de eventos en lo que se refiere a la estructura de tablas que representan el mensaje ATNA:

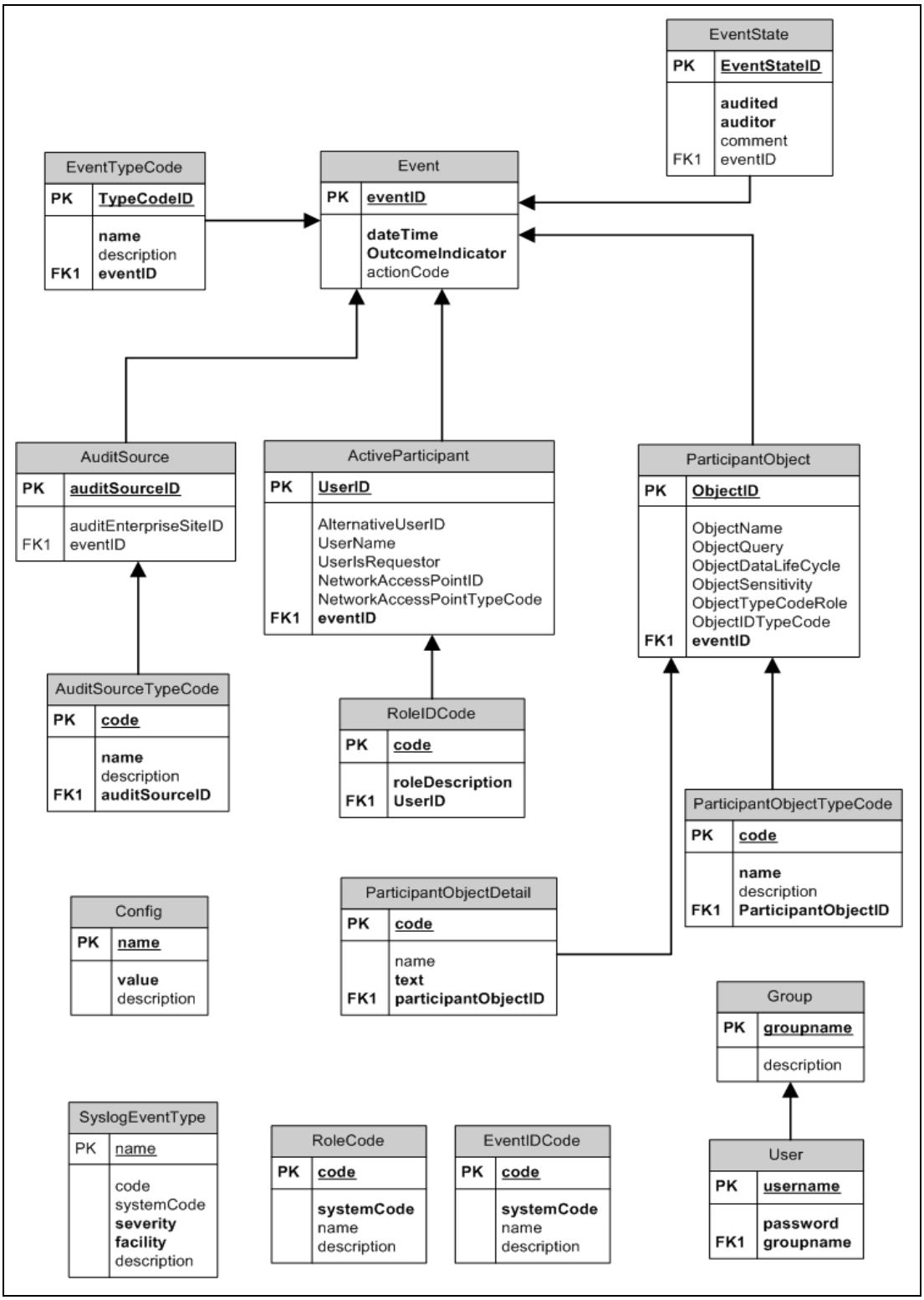


Figura 30 - Diagrama de Modelo de Datos del Sistema de Auditoría de Eventos

7.2.4 Aspectos de Seguridad

A continuación se presentan los principales aspectos de seguridad implementados en el Sistema de Auditoría, para mayor detalle ver [anexo 2]:

Autenticación y Autorización

Para la autenticación de usuarios en el sistema se utiliza el mecanismo de usuario/contraseña con un dominio de tipo JDBC para almacenar los usuarios y grupos en el servidor. La implementación fue pensada para intercambiar mecanismos de autenticación sin tener que modificar código. Para ello se definieron las directivas de seguridad en archivos XML como indica el modelo de seguridad declarativo de Java EE.

Para identificar a los usuarios y permitirle o denegarle el acceso a los recursos de la aplicación se deben seguir los siguientes pasos:

1. Codificar el prompt donde el usuario ingresa sus credenciales. Esto dependerá del mecanismo de autenticación utilizado.
2. Crear el deployment descriptor según el tipo de componente (ejb, ws, web) y colocar en él la información de roles y reglas.
3. Agregar los usuarios y grupos necesarios en el servidor.
4. Mapear los usuarios y grupos definidos en el servidor con los roles de la aplicación.

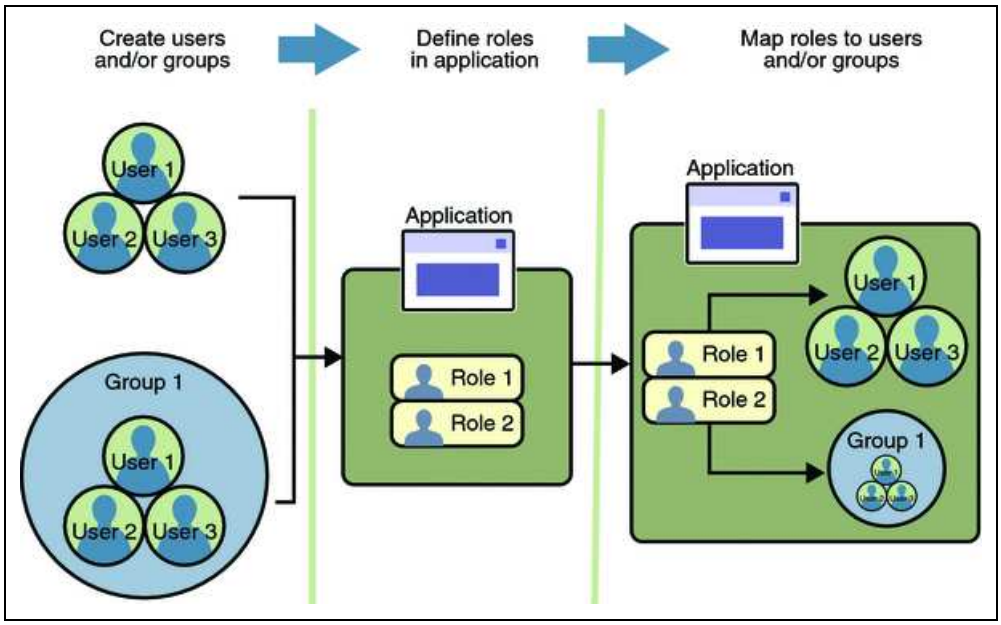


Figura 31 - Mapeo entre grupos, roles y usuarios en aplicaciones web [36]

La Figura 32 muestra el procedimiento para asegurar el acceso a una aplicación web.

El mapeo entre grupos y roles se utiliza para mantener independencia entre los roles de la aplicación y los grupos definidos en el servidor. Por otra parte existe una sutil diferencia entre ambos conceptos ya que un rol define permisos para un conjunto de recursos dentro de una aplicación, mientras que un grupo define el conjunto de usuarios autenticados en todo el servidor.

Para almacenar los grupos y usuarios se definen en el servidor los dominios, los cuales pueden verse como una base de datos de usuarios y grupos admitidos. Existen básicamente cuatro tipos de dominios:

- **File**: En el cual los usuarios y grupos se definen en un xml almacenado en el servidor. No es utilizado para clientes que se comunican mediante HTTPS sobre SSL.
- **Admin**: Define el usuario para el acceso a la consola de administración.
- **Certificate**: En este dominio el servidor almacenará las credenciales de los usuarios en una base de datos para certificados. Si se selecciona este mecanismo el servidor utilizará certificados junto con el protocolo HTTPS para validar a los clientes.
- **JDBC**: Permite almacenar la información en una base de datos relacional y accederla mediante JDBC.

El dominio utilizado puede ser cambiado por cualquiera de los restantes, según las necesidades y requerimientos de seguridad. Por ejemplo si se desea autenticación de los clientes mediante certificados simplemente bastará con seleccionar el nuevo dominio y cargarlo con los certificados utilizados.

Comunicación

Se configuró el sistema de manera que soporte la comunicación con clientes mediante SSL sobre HTTP con autenticación mutua. En la autenticación mutua una vez que se realiza la petición el servidor se autentica contra el cliente y luego el cliente lo hace contra el servidor enviando su certificado. Los certificados utilizados tanto para el cliente como para el servidor se crearon con la Autoridad Certificadora del Hospital de Clínicas, pudiendo ser sustituidos por otros. Para ello se deben agregar el nuevo certificado del servidor en el keystore y los certificados para los clientes en el truststore del servidor de aplicaciones y configurarlo para que utilice el nuevo certificado. El certificado del servidor debe ser instalado en el truststore del cliente, generalmente en el explorador web.

La Figura 32 muestra el manejo de las peticiones en una aplicación web Java EE:

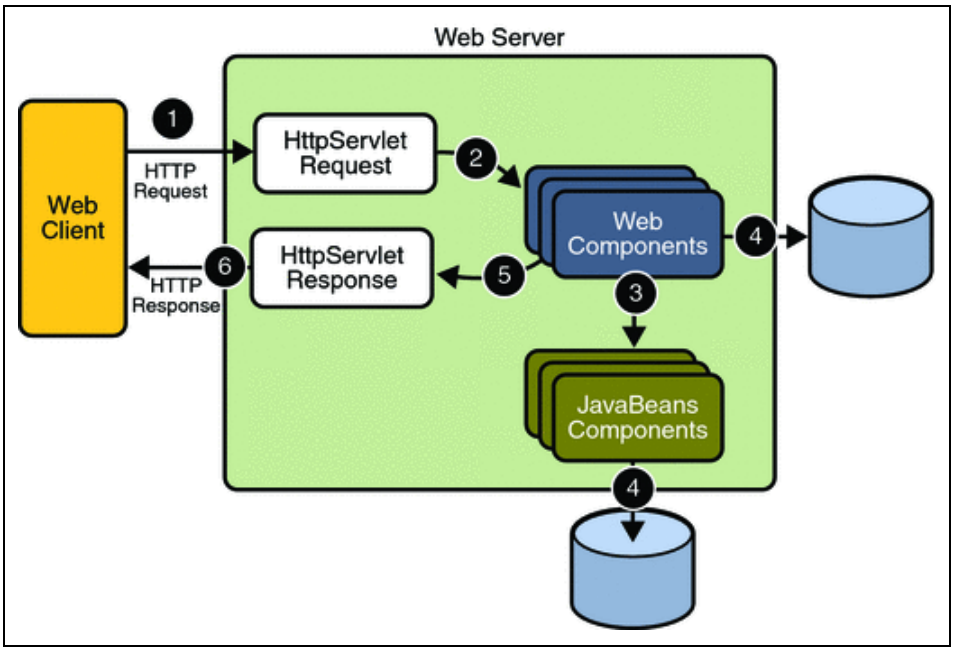


Figura 32 - manejo de peticiones HTTP en aplicaciones Java EE [36]

Se utilizó seguridad declarativa mediante los deployment descriptors correspondientes.

Archivo web.xml:

```
<web-app>

    <login-config>
        <auth-method>BASIC</auth-method>
        <realm-name>JDBCRealm</realm-name>
    </login-config>
    <security-role>
        <description/>
            <role-name>administrador</role-name>
    </security-role>
    <security-constraint>
        <web-resource-collection>
          <web-resource-name>view dept data</web-resource-name>
          <url-pattern>/sistemaAuditoria/*</url-pattern>
          <http-method>GET</http-method>
          <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
          <role-name>administrador</role-name>
        </auth-constraint>
        <user-data-constraint>
         <transport-guarantee>CONFIDENTIAL</transport-guarantee>
        </user-data-constraint>
    </security-constraint>

</web-app>
```

Figura 33 - Deployment descriptor web.xml para aplicación web [36]

En este archivo se define el mecanismo de login de usuario y el dominio que utilizará para validar a los mismos, en este caso el dominio es de tipo JDBC. Luego se definen los roles para la aplicación, en este caso existe un único rol denominado “administrador”. Por último se define una regla de seguridad la cual permite al rol administrador realizar GET y POST sobre todos los recursos web de la aplicación. El elemento <user-data-constraint> se utiliza para definir el mecanismo de seguridad cuando se tiene habilitado SSL sobre HTTPS.
El mapeo de roles con grupos del servidor se realiza en el archivo sun-web.xml:

```
<sun-web-app>
  <context-root>/AuditoriaSistemaRegistros</context-root>
  <security-role-mapping>
    <role-name>administradores</role-name>
    <group-name>administradores</group-name>
  </security-role-mapping>
    <security-role-mapping>
    <role-name>auditores</role-name>
    <group-name>auditores</group-name>
  </security-role-mapping>
</sun-web-app>
```

Figura 34 - Deployment descriptor sun-web.xml para aplicaciones web [36]

Como se puede apreciar se definen dos mapeos, y tanto los nombres de los roles como los nombres de los grupos del servidor coinciden, pero perfectamente pueden ser distintos.
Para habilitar HTTPS sobre SSL se utilizaron certificados emitidos por la Autoridad Certificadora del Hospital de Clínicas. El certificado para el servidor debe colocarse en el keystore del mismo y los certificados de los clientes en el truststore. Esto último solo es necesario si se desea autenticación mutua. Por último se debe configurar el descriptor de la aplicación para definir el método a utilizar. En el archivo web.xml de la Figura 34 se definió el elemento <user-data-constraint> en el cual se indica la potencia de la protección requerida:

- Confidential: Aplica cuando el requerimiento indica que los datos a transmitir entre el cliente y servidor no deben ser observados por terceros.
- Integral: Aplica cuando el requerimiento indica que los datos a enviar no deben ser alterados.
- None: El servidor acepta peticiones tanto con seguridad como sin seguridad incorporada.

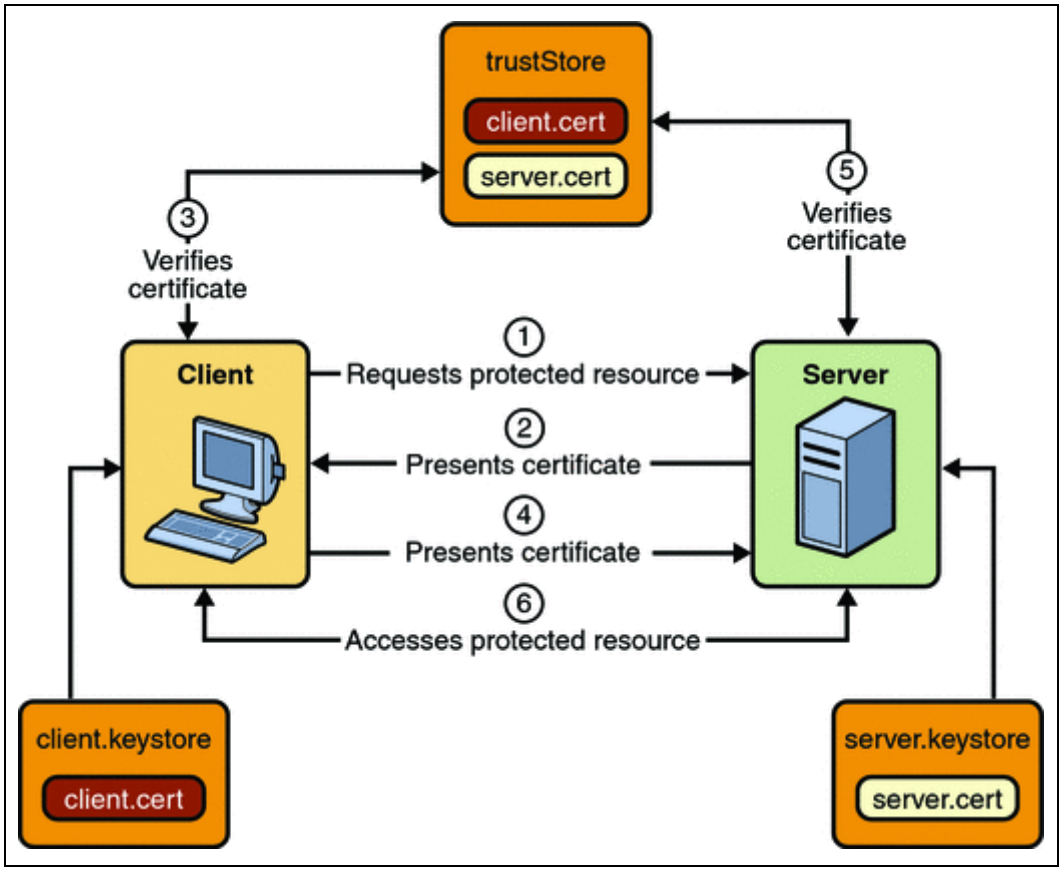


Figura 35 - Autenticación mutua con HTTPS/SSL [36]

En la autenticación mutua, tanto el cliente como el servidor se autentican uno contra el otro. El cliente realiza la petición y el servidor envía su certificado. En ese momento el cliente lo válida, chequeando que exista en su almacén de certificados confiables y en caso positivo envía su propio certificado para que el servidor lo valide. El servidor lo valida de la misma manera y en caso satisfactorio se le otorga al cliente el acceso al recurso protegido.

Control de Acceso a Recursos

Para permitir/denegar acceso de usuarios autenticados a los recursos del sistema según los roles del mismo se utilizan los atributos renderedOnUserRole y enableOnUserRol de cada componente ICEFaces. El framework ICEFaces se encarga de manera transparente de utilizar los mecanismos de seguridad de Java EE.

Por otra parte se permite dar de alta, modificar y eliminar usuarios (funcionalidad solo habilitada para el rol administrador). Cabe señalar que para mejorar la seguridad de la aplicación, en la base de datos se almacena el hash MD5 de la clave del usuario en lugar de la propia clave. Dado que la función de hash MD5 es unidireccional, es computacionalmente imposible obtener la clave del usuario a partir del hash de la misma, lo cual brinda mayor seguridad para el login de usuario.

Registro de Actividad

Se implementó el registro de actividad para aquellos usuarios que realizan operaciones de modificación o baja de eventos o algún dato del sistema.

Prevención de Ataques

Los principales ataques que previenen las tecnologías utilizadas son los siguientes:

- No se introducen agujeros de seguridad en la arquitectura Java EE al momento de utilizar XHR ya que este realiza un submit estándar disparando el ciclo de vida de JSF el cual cumple los requisitos de seguridad establecidos por Java EE.
- Prevención de inyección de código JavaScript malicioso es llevado a cabo por el framework JSF y heredado por ICEFaces.
- Compatibilidad con tecnologías de persistencia como Hibernate, las cuales previenen los ataques de inyección SQL.

7.3 Generador de Mensajes ATNA

Este módulo no forma parte de ninguno de los sistemas anteriores, sino que es utilizado por los mismos, y por aquellos clientes que quieran registrar mensajes de auditoría respetando el formato ATNA. El generador es empaquetado en un archivo .jar que deberá ser incluido dentro de las aplicaciones que deseen utilizarlo. Para la implementación se utilizó la biblioteca JDom [56].

Las funcionalidades que ofrece el módulo son las siguientes:

- Generación de mensajes en formato XML utilizando los elementos definidos en el Esquema XML definido por ATNA. Esto no significa que todo XML generado sea válido. La generación consiste en una serie de pasos, en los cuales deben agregarse los elementos obligatorios y sus atributos, en el formato y rango establecido por el esquema.
- Carga de la estructura que representa el mensaje ATNA a partir de un archivo de entrada XML, el cual será validado y posteriormente parseado. Esta funcionalidad es de importancia para aquellas aplicaciones que deseen extraer los campos del mensaje de auditoría.
- Validación de los mensajes ATNA recibidos y generados. Esto último es de utilidad para corroborar los mensajes generados antes de enviarlos al Sistema de Registro de Eventos.

7.4 Distribución de los Sistemas

Esta sección presenta alguna de las posibles distribuciones de los sistemas desarrollados:

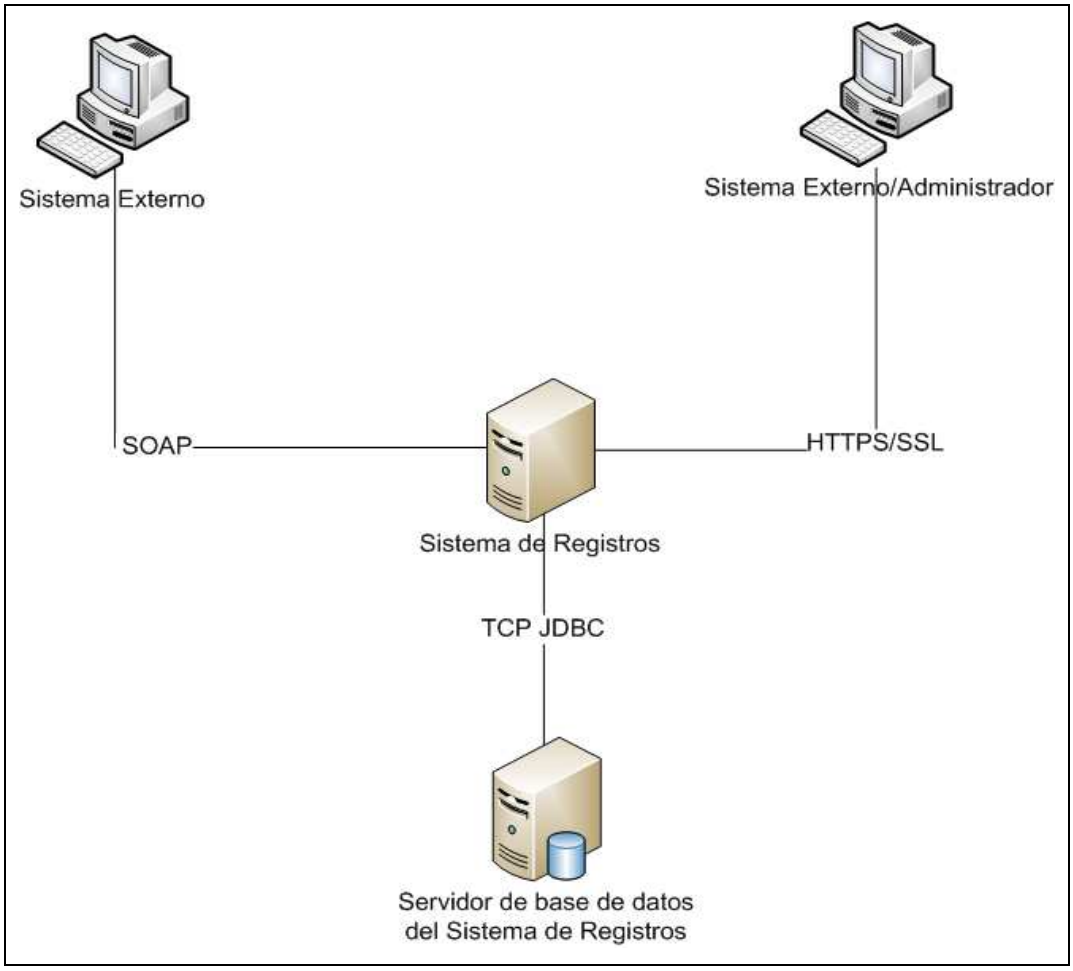


Figura 36 - Escenario I. Sistema instalados en un único servidor

La Figura 36 muestra un escenario en el cual la distribución del sistema es nula, ya que todos los componentes del Sistema de Registro de Eventos se encuentran en el mismo servidor, y a su vez el Sistema de Auditoría de Eventos también. Esta configuración no es la más recomendable ya que puede ocasionar problemas de performance si se tiene una tasa elevada de registro de eventos y a la vez se están auditando los mismos.

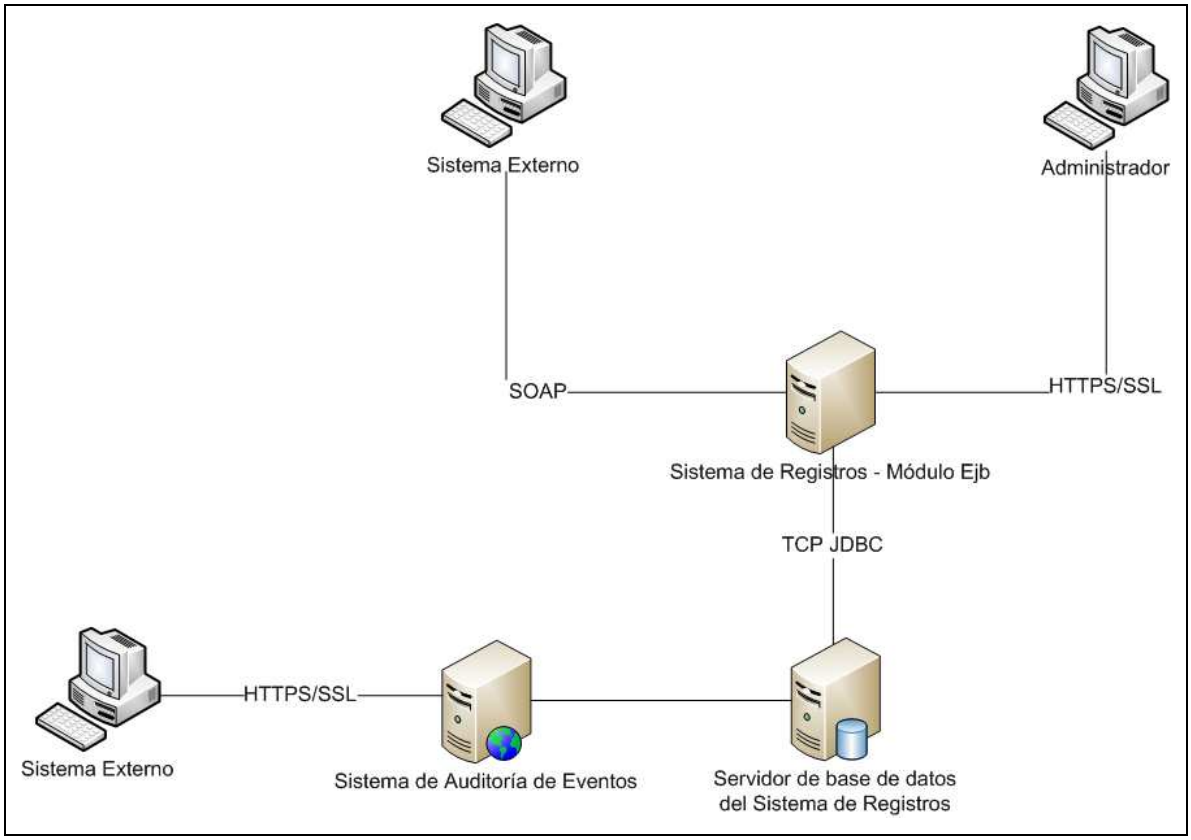


Figura 37 - Escenario II. Sistema parcialmente distribuido

La Figura 37 presenta un escenario parcialmente distribuido ya que si bien los componentes del Sistema de Registro de Eventos se encuentran instalados en un mismo servidor, el Sistema de Auditoría de Eventos se encuentra en otro. Esta podría ser la configuración ideal, ya que la performance del Sistema de Auditoría no se ve afectada por el registro de eventos, y además, el tiempo de registro de eventos es menor que en el caso distribuido ya que no se tienen demoras por comunicación entre servidores.

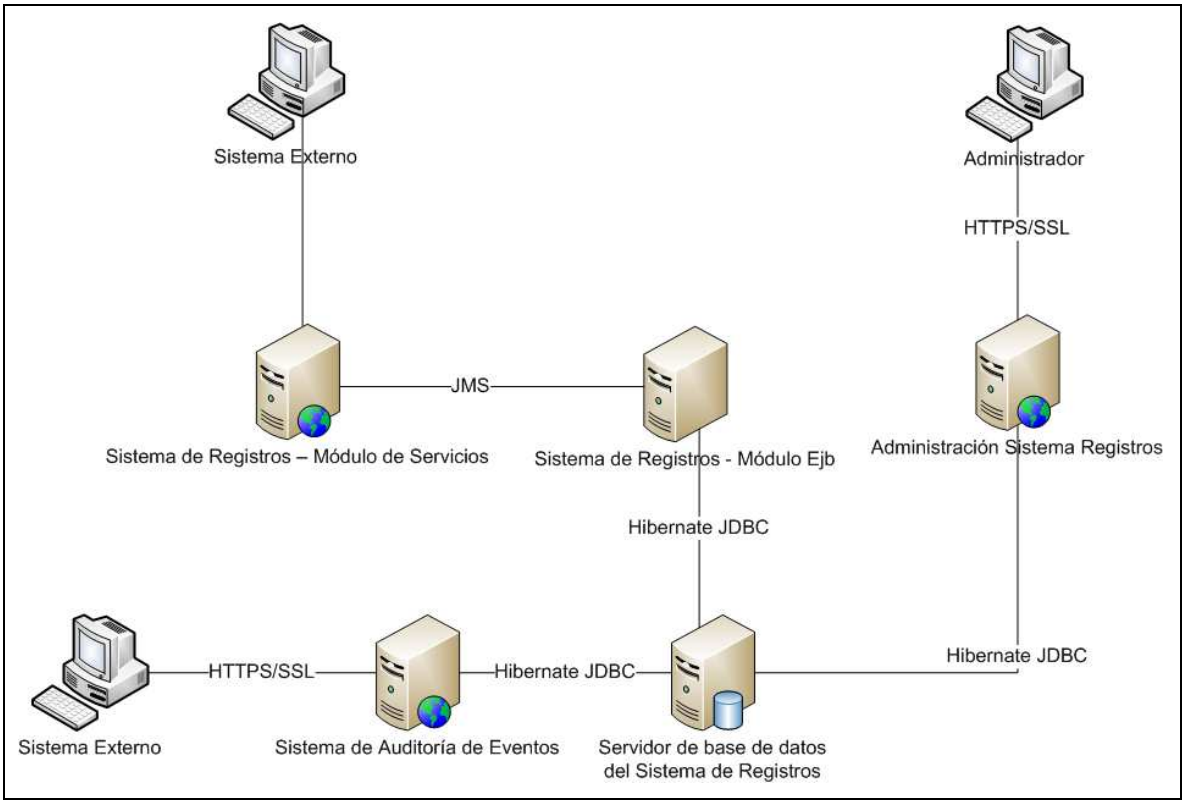


Figura 38 - Escenario III. Sistema altamente distribuido

La Figura 38 muestra una distribución altamente distribuida, en la cual cada módulo del sistema de registro de eventos se encuentra ubicado en un servidor dedicado, y en el que el sistema de auditoría también se encuentra en su propio servidor, favoreciendo la performance de este último, pero agregando un overead en el registro de eventos debido a la comunicación entre los servidores.

7.5 Aplicación de las Tecnologías Seleccionadas

Esta sección describe las tecnologías utilizadas en la implementación de los sistemas y su aplicación para resolver las problemáticas encontradas.

- **AJAX**

  Asinchronus JavaScript And XML es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Interface Application). Estas aplicaciones se ejecutan en el cliente, es decir en el navegador del usuario mientras se mantiene una comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

  Fue ampliamente utilizada para lograr los efectos visuales y mejorar la performance de la interfaz web del sistema de auditoría de eventos.

- **EJB 3.0**

  La tecnología EJB [36] es el componente en la arquitectura java EE. Permite el desarrollo de sistemas distribuidos, transaccionales, escalables, seguros y portables a partir de una infraestructura que brinda un conjunto inicial de servicios “middleware”, lo cual libera al usuario del diseño e implementación de los mismos. Entre ellos se destacan:

  - Conexión a base de datos.
  - Manejo de transacciones.
  - Seguridad.
  - Mensajería.

  Existen dos modelos de EJB, los que mantienen el estado (Statefull) y los que no (Stateless). Estos últimos tienen mejor performance y son recomendados para peticiones atómicas, mientras que los Statefull son de importancia para mantener sesiones con los usuarios.

  El servicio web implementado en la solución fue expuesto como EJB ya que el mismo obtendrá todos los beneficios del contenedor EJB del servidor de aplicaciones. Por otro lado, dada la naturaleza “sin estado” de los servicios web, el tipo de EJB para servicios web es Stateless. El contenedor se encarga de manejar el conjunto de instancias de

dicho EJB, generadas con cada petición al servicio web, para optimizar la performance y disponibilidad del mismo.

- **JAX-WS**

Java API for Web Services [36] es una API para simplificar el desarrollo y despliegue de servicios web y clientes que se comunican mediante XML. Con JAX-WS una invocación a una operación de un servicio web es representada por un protocolo basado en XML, como SOAP. Las especificaciones SOAP definen la estructura, reglas y convenciones para las invocaciones y respuestas, las cuales son transmitidas como mensajes SOAP (archivos XML) sobre HTTP. La gran ventaja de los clientes y servicios web implementados con JAX-WS es la independencia de la plataforma, lo cual permite por ejemplo a un cliente JAX-WS acceder a un servicio web ejecutando en un ambiente distinto a la plataforma Java y viceversa. Esto se debe a la utilización de estándares: HTTP, SOAP y WSDL.

- **Metro 1.4**

Metro [55] es un stack de servicios web de alta performance, extensible y fácil de utilizar, desarrollado y mantenido por la comunidad Glassfish. Las principales funcionalidades que ofrece son las siguientes:

  - Seguridad. Implementa la especificación WS-Security para proveer mecanismos de autenticación, confidencialidad y mantenimiento de la integridad de los datos en comunicaciones punto a punto en las que pueden existir intermediarios.
  - Disponibilidad. Asegura que los sistemas puedan recuperarse de fallas causadas por mensajes perdidos o desordenados durante el transporte.
  - Transporte. Dispone de diferentes tecnologías de transporte de datos, como pueden ser HTTP, MTOM, XOP, SOAP sobre TCP, etc.
  - Transaccionalidad. Mantiene la consistencia de los datos demarcando transacciones en las que si ocurre una falla se cancelan todas las acciones realizadas hasta el momento.

Por otra parte implementa la API JAX-WS y los mecanismos básicos de interoperabilidad como WS-I Basic Profile, WS-I Attachments Profile, and WS-Addressing.

Esta tecnología fue utilizada para construir el servicio web del sistema de registro de eventos, así como los clientes del mismo. También fue de utilidad para dotar a las aplicaciones desarrolladas en Genexus a que dispongan de una conexión segura al servicio web del sistema de registro de eventos.

- **Quartz Scheduler**

Quartz [80] es una tecnología de código abierto que ofrece un servicio para la planificación de tareas, el cual se puede integrar con aplicaciones Java EE o Java SE. Por tarea se entiende la ejecución de un proceso, programado por el usuario mediante componentes estándar de Java, el cual podrá realizar acciones sobre el sistema.
En el marco de este proyecto fue utilizado para ejecutar tareas de mantenimiento del sistema de registro de eventos, como es el caso de la eliminación de eventos según su tiempo límite de almacenamiento. La tarea ejecutará periódicamente consultas a la base de datos verificando la fecha y severidad de cada evento y en base a las prioridades definidas por el administrador eliminará los eventos correspondientes (ver sección 7.1.3).

- **JMS**

Java Message System provee una API estándar para el acceso a MOM (Message Oriented Middleware). MOM es aquel software que reside tanto en el cliente como en el servidor arquitectónicamente hablando, y es capaz de recibir llamadas asíncronas entre la aplicación cliente y el servidor.

JMS dispone de dos modelos de mensajería: Point-to-Point (Queue) y Publish-Subscriber (Topics). En el primero cada emisor coloca su mensaje en la cola correspondiente y los clientes lo obtienen de las mismas:

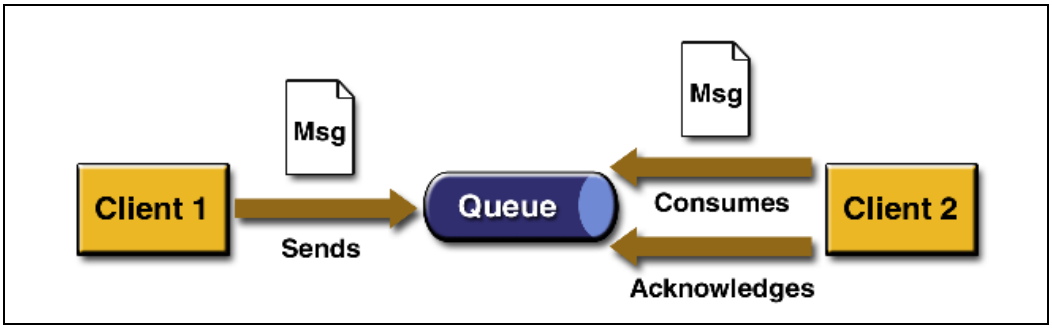


Figura 39 - Modelo Point-To-Point (extraído del manual Java EE) [36]

En la Figura 39 se presenta el modelo punto a punto, en el cual “Client1” actúa como emisor, enviando el mensaje “Msg” a la cola de mensajes “Queue”. “Client2” consume el

mensaje y envía una señal de Acknowledge a la cola indicando que ya obtuvo el mensaje. JMS se encarga de recibir dicha señal y procesar el siguiente mensaje.

En el segundo caso los receptores se suscriben a los tópicos y los emisores envían los mensajes a estos últimos. JMS se encarga de distribuir cada mensaje a los correspondientes suscriptores según el tópico del mismo:

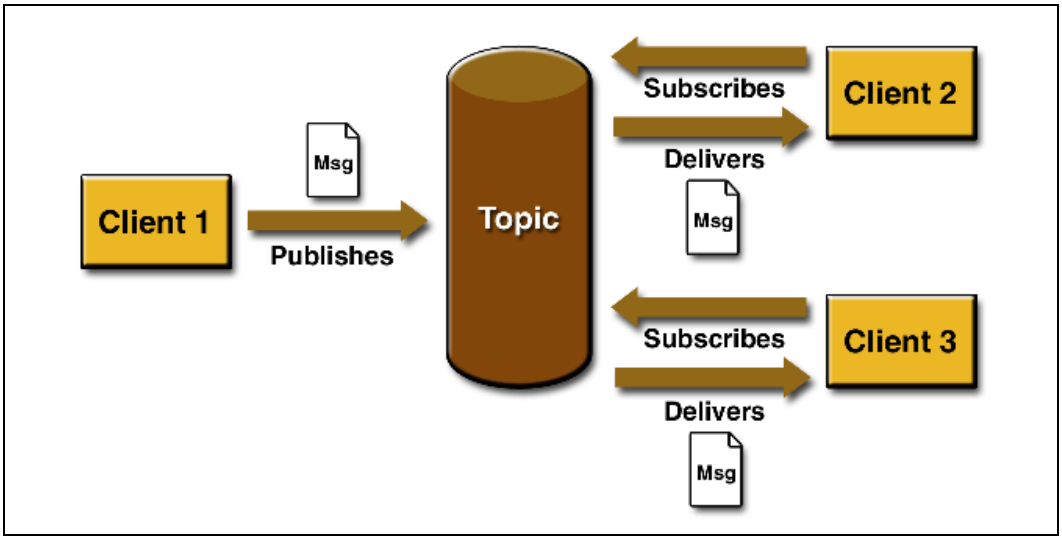


Figura 40 - Modelo Publisher-Subscriber (extraído del manual Java EE) [36]

En la Figura 40 "Client1" publica el mensaje en la cola correspondiente al tópico de dicho mensaje, mientras que los clientes "Client2" y "Client3", suscritos al tópico, son avisados de la existencia de dicho mensaje. JMS resuelve de forma transparente la publicación del mensaje en el tópico correspondiente, así como la entrega del nuevo mensaje a todos aquellos clientes suscritos a dicho tópico.

Para la implementación del Sistema de Registro de Eventos se siguió la estrategia Point-To-Point ya que es el más adecuado para las siguientes condiciones:

- Cada mensaje tiene un único consumidor.
- El emisor y el receptor no tienen dependencias de tiempo, por lo tanto el mensaje será procesado una vez que el receptor este libre, sin bloquear al emisor a la espera del resultado.

Con esta estrategia se garantiza una mejor performance a la hora de registrar los eventos, y una mayor disponibilidad por parte del Sistema de Registro de Eventos ya que el servicio web (actúa como emisor) queda libre rápidamente para estar a la espera de nuevos mensajes provenientes de otros sistemas.

- **Hibernate 3**

Mecanismo de persistencia que implementa y extiende las funcionalidades definidas por JPA (Java Persistence API). Ocupa la categoría de ORM (Object Relational Mapping) que define la forma de transformar un modelo de datos orientado a objetos en un modelo de datos relacional. Para poder hacer uso eficiente de dicha característica, Hibernate provee funcionalidades para crear la estructura de una base de datos a partir de los objetos definidos en la aplicación. La estructura se crea o actualiza al momento de desplegar la aplicación en el servidor de aplicaciones.

Otras características:

- Definición de relaciones: Uno-Uno, Uno-Muchos, Muchos-Muchos.
- Generación de claves incrementales
- Generación de secuencias
- Generación de índices.
- Lenguaje de consultas SQL (denominado HQL) orientado a objetos.

- **JavaMail 1.4.3**

JavaMail [59] es una API independiente del protocolo que permite el envío y la recepción de emails. Soporta IMAP4, SMTP y POP3.

Se utilizó para implementar el envío de alertas al correo electrónico del administrador del sistema.

- **JDom**

JDom [56] es una biblioteca para manejo de archivos XML. Fue de gran utilidad ya que permite la construcción de archivos XML de una forma orientada a objetos. Por ejemplo se dispone de objetos Document, Element y Attribute que representan la raíz del documento, los elementos hijos y sus atributos respectivamente. A su vez permite generar a partir de un archivo XML la estructura programática que lo representa, la cual puede ser manipulada para obtener el valor de cada atributo para cada elemento.

Fue ampliamente utilizado en la biblioteca de generación de mensajes ATNA. Se utilizó tanto para generar archivos XML programáticamente, como para parsearlos y extraer sus campos.

- **JXL**

Java Excel API [57] es una biblioteca desarrollada utilizando el JDK 1.5 y brinda una abstracción a la hora de crear planillas Excel. Su orientación a objetos permite ser una herramienta intuitiva para el desarrollador. Brinda soporte para:

- Creación de planillas
- Creación de hojas dentro de una planilla
- Creación/Modificación/Eliminación de filas y columnas dentro de una hoja.
- Creación/Modificación/Eliminación de celdas, aplicando formatos, fuentes, formulas, etc.
- Exportación de la planilla a un archivo en disco o a un flujo de bytes lo cual hace flexible el manejo de planillas en memoria, y la descarga de las mismas como adjuntos en un response HTTP.

Fue utilizada conjuntamente con los componentes de reporte ICEFaces para generar los mismos.

- **iText 2.1**

iText [58] es una biblioteca que permite la generación automatizada de archivos PDF, cuyo contenido se basa en los datos de entrada del usuario. Dispone de versiones en lenguaje java y C#. Las principales funcionalidades de iText son las siguientes:

- Generar documentos dinámicamente tomando la información desde archivos XML o base de datos.
- Mostrar documentos PDF en el explorador Web.
- Firmar digitalmente el documento PDF generado.

Fue utilizada conjuntamente con los componentes de reporte ICEFaces para generar los mismos en formato PDF.

- **JSF 1.2**

Es un framework para la implementación de interfaces de usuario en aplicaciones java web cuya arquitectura se encuentra en la categoría MVC (Model Controller View). Este

estilo arquitectónico separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. La Figura 41 muestra la estructura MVC:

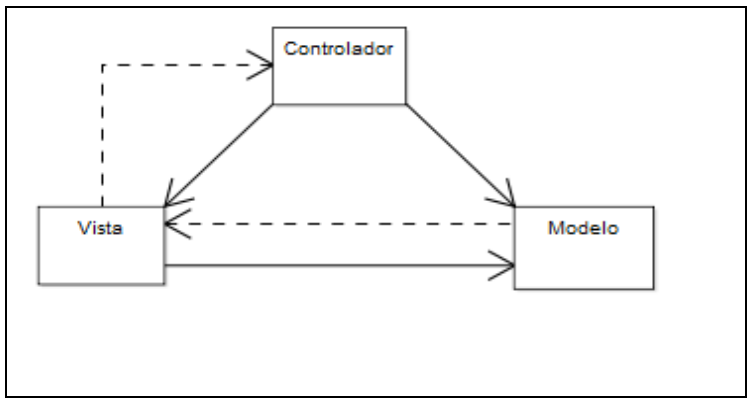


Figura 41 - Modelo MVC

- **Modelo**: Esta es la representación específica de la información con la cual el sistema opera.
- **Vista**: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador**: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Por otra parte JSF presenta una API para representar componentes gráficos y manejar su estado, manejar eventos, realizar validaciones en el servidor, definir la navegación entre las páginas. Los principales beneficios de esta tecnología son:

- Separación entre comportamiento y presentación, a un nivel fino, similar al que ofrecen las aplicaciones cliente-servidor. Esto se logra manteniendo en el servidor el estado de los componentes gráficos.
- Dicha separación permite a los desarrolladores concentrarse en cada pieza de la aplicación, según sus conocimientos, y sin la necesidad de conocer el resto ya que JSF provee un modelo de programación que facilita la vinculación de componentes.
- No limita al uso de los componentes existentes en su especificación, sino que permite al desarrollador crear sus propios componentes, o utilizar otros existentes, como por ejemplo componentes Facelets o de ICEFaces.

- **JNDI**

Java Naming Directory Interface permite el acceso a servicios de nombres o directorios, definiendo una interfaz de acceso estándar a ambos. En otras palabras permite acceder de manera uniforme a los recursos existentes, como por ejemplo colas de mensajes, sesiones de email, contexto de persistencia, etc.

- **Facelets 1.1.14**

La capa de presentación fue implementada mediante facelets ya que provee un mecanismo de composición de páginas, el cual nos permite:

- Definir marcos que puedan ser reutilizables o comunes a varias páginas, por ejemplo menú de usuario, foto de perfil, etc. La referencia a estos marcos solo requiere una línea de código evitando tener que copiar todo el código del marco en cada página en la que se desee mostrar.

- Definir partes de una página para que sean re implementadas por otras páginas de la aplicación. Por ejemplo en el template principal se define un área llamada “body” que ocupa cierta porción de la página. Dicha área deberá implementarse por cada página variando el contenido si es necesario, caso contrario no se visualizara información alguna en ese espacio.

Esta tecnología fue de gran utilidad ya que no se poseía experiencia en el desarrollo de interfaces de usuario, y la experiencia con JSF reflejó demasiadas dificultades.

- **ICEFaces 1.8.2**

Es un framework [44] para desarrollo de aplicaciones Java utilizando la tecnología Ajax. ICEFaces [46] permite al desarrollador crear y desplegar aplicaciones RIA (Rich Interface Application) en java puro para clientes livianos (thin-client), es decir no es necesario ningún tipo de plugin o applet en el cliente. Una aplicación ICEFaces básicamente es una aplicación JSF con ciertos beneficios:

- Experiencia del usuario: Dispone de una suite de más de 50 componentes JSF que permite mejorar considerablemente la efectividad y la experiencia del usuario, a través de la incorporación de ajax a los mismos.

- Código Abierto: Es el framework Ajax para Java más exitoso y con una comunidad que excede los 65000 desarrolladores.

- Basado en estándares: ICEFaces está desarrollado enteramente en Java por lo tanto los desarrolladores pueden incluirla en sus aplicaciones y continuar trabajando de la misma forma que lo venían haciendo.

- Facilidades Ajax: Brinda facilidad a la hora de realizar comunicaciones vía ajax entre el cliente y servidor evitando tener que codificar las mismas en JavaScript sino en Java/JSF. Cada componente podrá realizar un pedido al servidor mediante ajax, definiendo una asociación entre el evento ejecutado sobre el componente y una operación en el lado del servidor. Cuando se lleva a cabo el evento (por ejemplo clic en un botón) se dispara la operación asociada en el servidor, retornando los resultados y actualizando únicamente los componentes en cuestión.

- Seguridad: Es compatible con SSL, y está diseñado para prevenir las siguientes situaciones:

  - Cross-site scripting.

  - Inyección de código malicioso.

  - Prevención de Data Mining no autorizado.

  - Ataques de inyección SQL (SQL Injection Attack)

- **PosgreSQL 8.4**

Como manejador de base de datos se utilizó PostgreSQL 8.4 y como conector la biblioteca postgresql-8.4-701.jdbc4.jar.

- **JQuery**

Biblioteca JavaSript que permite un manejo más sencillo de los componentes que forman parte de una página web. A su vez se utilizó para realizar llamadas mediante ajax.

- **Glassfish V2.1**

Servidor de aplicaciones utilizado para el despliegue de los sistemas implementados. Implementa completamente la especificación Java EE 5 y dispone de soporte para seguridad a nivel de servicios web a través del stack de servicios web denominado Metro.

- **NetBeans 6.7**

Entorno de desarrollo utilizado para la implementación de los sistemas. Provee soporte para el servidor de aplicaciones Glassfish V2 y posteriores así como integración con las tecnologías ICEFaces y Facelets, lo cual facilita el desarrollo con las mismas.

- **Keytool**

Utilidad java para el manejo de claves públicas, privadas y certificados. Fue utilizado para importar claves primarias y certificados al almacén de claves y certificados confiados del servidor de aplicaciones respectivamente.

- **JDK 1.6 update 18**

La Plataforma Java se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real. Por ejemplo, los usuarios finales suelen interactuar con la maquina virtual de java y el conjunto estándar de bibliotecas. Además, las aplicaciones Java pueden usarse de forma variada, como por ejemplo ser incrustadas en una página Web. Para el desarrollo de aplicaciones, se utiliza un conjunto de herramientas conocidas como JDK (Java Development Kit, o herramientas de desarrollo para Java).

- **GeneXus 9.0 Update 6**

GeneXus es una herramienta de desarrollo de software basada en conocimiento, orientada principalmente a aplicaciones de clase empresarial para la web y plataformas Windows. El desarrollador especifica sus aplicaciones en alto nivel (de manera mayormente declarativa), a partir de lo cual se genera código para múltiples entornos.

GeneXus incluye un módulo de normalización, que crea y mantiene una estructura de base de datos óptima basada en el modelo de datos no normalizado definido por los usuarios, un lenguaje declarativo (basado en reglas) y un lenguaje procedural simple pero poderoso.

Los lenguajes para los que se puede generar código incluyen:

- Cobol
- Visual Basic
- Visual FoxPro
- Ruby
- C#
- Java

Actualmente con énfasis en los últimos tres. Los DBMSs más populares son soportados, como Microsoft SQL Server, Oracle, IBM DB2, Informix, PostgreSQL y MySQL.

La herramienta fue utilizada para implementar un módulo que encapsule la lógica para la generación y posterior envío de un mensaje de auditoría al servicio web del sistema de registro de eventos (Ver sección 6.3.5). La construcción del mensaje respeta el formato definido por RFC 3881.

7.6 Decisiones Tomadas

A continuación se comentan las principales elecciones tecnológicas tomadas durante el transcurso del proyecto.

7.6.1 MySQL vs PostgreSQL

Para tomar la decisión se realizo una comparativa entre ambos manejadores de base de datos. Los resultados son los siguientes (para más detalle ver el documento de Decisiones Tomadas):

MySQL:

Su principal objetivo de diseño fue la velocidad. Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria. Tiene licencia GPL a partir de la versión 3.23.19.

- **Ventajas:**
    - Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir consultas simples.
    - Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
    - No hay límites en el tamaño de los registros.
    - Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- **Inconvenientes:**
    - No soporta transacciones, "roll-backs" ni subselects (No se aplica al motor InnoDB).
    - No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación (No se aplica al motor InnoDB).

PostgreSQL:

Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase. Tiene licencia BSD.

- **Ventajas:**
    - Por su arquitectura de diseño, Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
    - Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
    - Tiene mejor soporte para triggers y procedimientos en el servidor.
    - Soporta un subconjunto mayor de instrucciones SQL del que soporta MySQL. Además, tiene ciertas características orientadas a objetos.
    - Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle
- **Inconvenientes:**
    - Consume más recursos y carga más el sistema que MySQL.
    - Límite del tamaño de cada fila de las tablas a 8k. (se puede ampliar a 32k recompilando, pero con un costo añadido en el rendimiento).
    - Es de 2 a 3 veces más lenta que MySQL en queries simples.

En general, sistemas en los que la velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante (pueda bastar con hacer backups periódicos que se restaurarán tras una caída del servidor). En cambio, para sistemas más serios en las que la consistencia de la BD sea fundamental (BD con información realmente importante, bancos, etc.) PostgreSQL es una mejor opción pese a su mayor lentitud.

En base al estudio realizado la decisión fue utilizar como SGBD a PostgreSQL debido a que es prioridad la integridad de los datos debido a la importancia de los mismos.

También nos basamos en una experiencia que tuvimos anteriormente al desarrollar una aplicación utilizando MySQL como SGBD.

En esta, cuando hacíamos una consulta que devolvía muchos registros, aprox. 20.000, el sistema nos deba un error por falta de memoria.

En ese momento, los clientes (que son los mismos que para el proyecto actual), nos sugirieron usar PostgreSQL como SGBD.

7.6.2 Glassfish V2.1 vs JBoss 5.1.0 AS [33]

Destacar las diferencias entre GlassFish y JBoss de ningún modo significa impugnar la calidad de JBoss. Sin embargo, es importante destacar las razones que nos llevaron a escoger GlassFish, en particular la versión 2.1 como servidor de aplicaciones java para el proyecto.

A continuación se presentan algunos ejemplos de funciones y características específicas en las que GlassFish tiene ventaja sobre JBoss.

Certificación JAVA

GlassFish fue el primer servidor de aplicaciones en obtener el certificado Java EE 5, y por lo tanto lleva una ventaja de 3 años sobre JBoss comercializando productos con esta categoría.

Sun ha apoyado ya dos arquitecturas completas (v1 y v2) de GlassFish y se compromete a apoyar la v3, proporcionando un nivel de experiencia y apoyo que JBoss no posee. GlassFish tiene el objetivo de ser el primer servidor certificado de aplicaciones Java EE 6, dando a los desarrolladores una ventaja en el aprovechamiento de los últimos avances en la tecnología Java. Por ejemplo, GlassFish v3 Preview ofrece acceso temprano a las tecnologías Java EE 6 y el nuevo Java EE 6 Web Profile SDK Preview.

Performance

- GlassFish es el único servidor de aplicaciones de código abierto probado que supera a servidores propietarios, como lo demuestran los resultados de las pruebas SPECjAppServer. Por ejemplo, en idénticas configuraciones de Sun Fire T2000, GlassFish V2 superó a WebSphere (servidor de aplicaciones pago) en un 43%.

  JBoss nunca se ha presentado a las pruebas SPECjAppServer, dejando una interrogante en cuanto a su rendimiento general.

- El tiempo de inicio del servidor medido por parte de Sun GlassFish v3 Preview es 7,56 segundos, el de GlassFish v2.1 es de 22,1 segundos y el de JBoss fue de 73,6 segundos.

Facilidad de uso y Administración

GlassFish está bien calificado por los desarrolladores para su facilidad de uso y las características de administración. Algunos ejemplos:

- GlassFish v2 y GlassFish v3 Preview proporcionan una interfaz guiada para establecer las configuraciones, lo que simplifica la tarea de administración.
  Con JBoss tareas administrativas se realizan a través de JMX beans. Incluso adicionando una consola de administración, como es JOPR, las funcionalidad no aproximan a GlassFish.

- Mientras que GlassFish y JBoss proporcionar un interfaz de control, sólo GlassFish ofrece un seguimiento no intrusivo y excelente integración con DTrace (herramienta que permiten medir, controlar, registrar, etc. variables del sistema) de los sistemas operativos Solaris y Macintosh que permite tener un control en todas las plataformas.

Servicios web

GlassFish v2 proporciona el “Metro Stack” que consiste en un conjunto de APIS como son JAX-WS, JAXB y WSIT, lo que permite a los desarrolladores crear y desplegar de forma segura, confiable y transaccional.

Personal de Sun y Microsoft se reúnen regularmente para ofrecer las más seguras y de alto rendimiento APIs de servicios web para inter operar entre .NET y J2EE con GlassFish, haciendo a GlassFish la mejor opción para inter operar con Microsoft .NET.

Soporte de IDEs/Herramientas

A diferencia de JBoss, GlassFish proporciona preconfiguraciones para los entornos de desarrollo NetBeans y Eclipse. Por ejemplo, NetBeans 6.7 admite GlassFish v2.1 y GlassFish v3, con soporte completo del API Java EE 5 y asistentes para facilitar la experiencia con NetBeans.

Un plug-in para el IDE de Eclipse está disponible a través del centro de actualizaciones e IntelliJ IDEA 7 en adelante también incluye un plug-in de GlassFish.

Conclusión

GlassFish combina capacidades de clase empresarial con la rentabilidad y flexibilidad del modelo de código abierto. Es el único servidor de aplicaciones de código abierto que combina la certificación de Java EE 5, documentación exhaustiva y accesible, configuración y administración intuitiva, integración con NetBeans.

La elección de un servidor de aplicaciones es altamente estratégica para los proyectos, y GlassFish ha demostrado para ser una opción estratégica superior para los desarrolladores de aplicaciones de Java.

7.7 Dificultades encontradas

Comprensión del esquema XML ATNA para mensajes de auditoría y el significado de todos sus campos dada la escasa información existente.

Comprender los conceptos introducidos por la especificación java EE, su arquitectura distribuida y como cooperan y se comunican los componentes unos con otros. Algunos ejemplos:

- Manejo de la transaccionalidad y de la persistencia (objetos attached, deattached). Relaciones entre las entidades y como se mapean con las relaciones propias de una base de datos relacional (1 a 1, 1 a N, N a N).
- Comunicación entre componentes, saber desde que contexto se pueden inyectar dependencias y en cuales se debe hacer lookup manualmente.
- Configuración de la aplicación (bibliotecas de base de datos dieron problemas a la hora de realizar el despliegue de la aplicación).
- Configuración del servidor de aplicaciones para soportar los requerimientos de la aplicación (JMS, JavaMail, PostgreSQL, etc).
- Diseño de la interfaz de usuario utilizando las tecnologías antes mencionadas, ya que la curva de aprendizaje es elevada.

No fue posible lograr interoperabilidad a nivel de servicio web utilizando el mecanismo de seguridad UserNameToken definido por la especificación WS-Security, entre clientes implementados en PHP utilizando el framework WSO2 y el sistema de registro de eventos implementado en java. La dificultad radica en la interpretación de los mensajes SOAP generados con la herramienta WSO2.

8 Caso de estudio

En esta sección se presenta el caso de estudio definido para analizar y validar la solución planteada. Por más detalles sobre el caso de uso referirse a [Anexo 17] y [Anexo 18].

8.1 Descripción

El objetivo de esta etapa es aplicar, analizar y mostrar la viabilidad de la solución.

Puntualmente se pretende validar que el sistema implementado permita registrar eventos sucedidos en distintas aplicaciones del entorno clínico y la posterior auditoría de los mismos mediante la cual se pueden observar flujos de trabajo, alta, baja y modificación de información, errores, etc.

Para esto se implementó un sistema en el cual, cuando un usuario ingresa al mismo, dependiendo de su rol, se habilita las características correspondientes. El objetivo de este sistema es disponer de un software que permita simular el manejo de la información de los pacientes en el Hospital de Clínicas. De esta manera se puede distinguir 4 funcionalidades principales, estas son:

a. Índice de Pacientes: Permite el ingreso de la información de los pacientes que serán atendidos en el Hospital de Clínicas.

b. Pacientes para Atender: Otorga la posibilidad a los médico de ver los pacientes que debe atender y su información asociada.

c. Pacientes para Analizar: Permite a los usuarios del Laboratorio ver los análisis que se le deben practicar a un paciente.

d. Medicamentos para Entregar: Permite a los farmacéuticos entregar los medicamentos recetados a los pacientes del Hospital de Clínicas.

El flujo de actividades que permitirán simular la atención de un paciente en la institución médica tiene básicamente 5 etapas:

1. Una persona llega por primera vez al Hospital de Clínicas en donde un administrativo lo registrado como paciente. Luego de registrado se le asigna una fecha y hora para ser atendido por un Médico.

2. En la fecha especificada el Médico atiende al paciente y le indica que se debe realizar unos análisis para completar los estudios.

3. El paciente concurre al laboratorio, se realiza los análisis y el laboratorista especifica los resultados en la Historia Clínica del paciente.

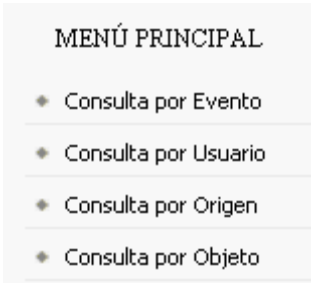
4. El paciente vuelve al médico luego de realizado los análisis, este observa los resultados y le prescribe algunos medicamentos.

5. El farmacéutico entrega los medicamentos recetados al paciente.

Cada uno de estos sistemas generan y registran logs en el Sistema de Registros de los sucesos que se producen en los mismos, por ejemplo, inicio y cierre de la aplicación, login y logout de usuarios, modificación de información relativa al paciente, etc.

En total se registraron 30 eventos, cuyos detalles se pueden observar en [anexo 16], donde figura la estructura del mensaje, los datos del mismo y el código java que lo produce.

Para observar los eventos se debe ingresar a la web de auditoría donde se podrán ver los eventos desagregados en sus 4 secciones, evento, participante, origen u objetos involucrados.



8.2 Ejecución del caso de estudio

Aquí se enumeran los pasos, en el orden en que se deben seguir, para llevar a cabo la ejecución del caso de estudio.

1. UN ADMINISTRATIVO

- Inicio de Programa.

El usuario de Windows “Gonzalo” ejecuta el programa “Índice de Pacientes” desde la PC con IP: 192.168.1.1 del Hospital de Clínicas en forma exitosa.

- Login Incorrecto.

El usuario de Windows “Gonzalo” intenta loguearse el programa “Índice de Pacientes” desde la PC con IP: 192.168.1.1 del Hospital de Clínicas en forma fallida.

- Login Correcto.

El usuario de Windows “Gonzalo” intenta loguearse el programa “Índice de Pacientes” desde la PC con IP: 192.168.1.1 del Hospital de Clínicas en forma exitosa.

- Alta Nuevo Paciente (Creación Instancia).

El usuario “gonzalo” del sistema “Índice de Pacientes” ingreso un nuevo paciente de nombre “Martín Calabria” al mismo.

- Modificación de Paciente (Modificación de Instancia).

El usuario “gonzalo” del sistema “Índice de Pacientes” modifica el paciente ingresado anteriormente.

- Crea una orden para un médico.

El usuario “gonzalo” del sistema “Índice de Pacientes” creo una orden de atención para que el paciente de nombre “Martín Calabria” pueda ser atendido por el doctor “Jose Pérez”.

- Logout.

El usuario “gonzalo” del sistema “Índice de Pacientes” se desloguea del programa.

- Fin de Programa.

El usuario de Windows “Gonzalo” cierra el programa “Índice de Pacientes”.

2. UN DOCTOR

- Inicio de Programa.

El usuario de Windows “Jose” ejecuta el programa “Pacientes para Atender” desde la PC con IP: 192.168.1.2 del Hospital de Clínicas en forma exitosa.

- Login Correcto.

El usuario de Windows “Jose” intenta loguearse el programa “Pacientes para Atender” desde la PC con IP: 192.168.1.2 del Hospital de Clínicas en forma exitosa.

- Visualización de Paciente.

El usuario “jose” del sistema “Pacientes para Atender” visualiza la información del paciente “Martín Calabria”.

- Ingreso de Información a la Historia Clínica.

El usuario “jose” del sistema “Pacientes para Atender” especifica características del paciente “Martín Calabria”.

- Estudio Creado.

El usuario “jose” del sistema “Pacientes para Atender” indica un análisis para el paciente “Martín Calabria”.

3. UN LABORITORISTA

- Inicio de Programa.

El usuario de Windows “Laboratorio” ejecuta el programa “Pacientes para Analizar” desde la PC con IP: 192.168.1.3 del Hospital de Clínicas en forma exitosa.

- Login Correcto.

El usuario de Windows “Laboratorio” intenta loguearse el programa “Pacientes para Analizar” desde la PC con IP: 192.168.1.1 del Hospital de Clínicas en forma exitosa.

- Estudio Visualizado.

El usuario “Laboratorio” del sistema “Pacientes para Analizar” visualiza los estudios indicados al paciente “Martín Calabria”.

- Estudio Modificado.

El usuario “Laboratorio” del sistema “Pacientes para Analizar” indica los resultados del análisis para el paciente “Martín Calabria”.

4. UN DOCTOR

- Estudio Visualizado.
  El usuario “jose” del sistema “Pacientes para Atender” visualiza los resultados de los estudios indicados al paciente “Martín Calabria”.
- Prescribir Medicamentos.
  El usuario “jose” del sistema “Pacientes para Atender” receta medicamentos al paciente “Martín Calabria”.
- Información de paciente exportada de la aplicación.
  El usuario “jose” del sistema “Pacientes para Atender” imprime la orden de medicamentos recetados.
- Logout.
  El usuario “jose” del sistema “Pacientes para Atender” se desloguea del programa.
- Fin de Programa.
  El usuario de Windows “Jose” cierra el programa “Pacientes para Atender”.

5. UN FARMACÉUTICO

- Inicio de Programa.
  El usuario de Windows “Julio” ejecuta el programa “Medicamentos para Entregar” desde la PC con IP: 192.168.1.4 del Hospital de Clínicas en forma exitosa.
- Login Correcto.
  El usuario de Windows “Julio” intenta loguearse el programa “Medicamentos para Entregar” desde la PC con IP: 192.168.1.4 del Hospital de Clínicas en forma exitosa.
- Entregar Medicamentos.
  El usuario “julio” del programa “Medicamentos para Entregar” entrega los medicamentos al paciente “Martín Calabria”.
- Logout.
  El usuario “julio” del sistema “Medicamentos para Entregar” se desloguea del programa.
- Fin de Programa.
  El usuario de Windows “Julio” cierra el programa “Medicamentos para Entregar”.

Los detalles de los eventos generados, así como los códigos que los generan y las imágenes de los mismos se pueden observar en [anexo 16].

9 Casos de prueba Sistema de Respaldos

A continuación se describirán dos pruebas que se le hizo el sistema de respaldos.

Para las mismas el Director, Almacenamiento, Consola y Catalogo corren en la misma máquina al que llamaremos servidor y esta montado sobre un sistema operativo Debian 5.0 (Lenny), también disponemos de 2 clientes, uno en la misma máquina y el otro en un sistema Windows XP. La configuración detallada de los mismos se encuentra en el documento [Anexo 15] pero es importante destacar que todos los módulos tienen un nombre y una contraseña, los que deben coincidir para lograr que estos se comuniquen. Para simplificar el ejemplo, el password se utilizó siempre fue "debian-dir" pero esto no es necesario.

La Figura 42 muestra un esquema con la relación que debe haber entre los distintos módulos y los usuarios y contraseñas definidos en ellos.

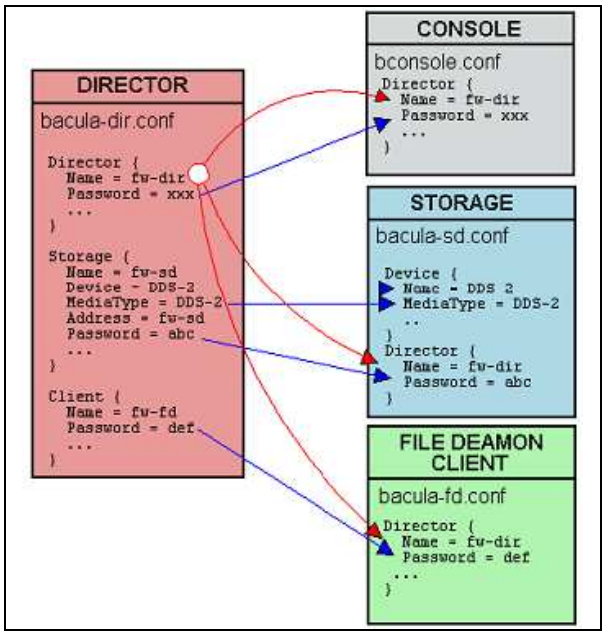


Figura 42 - Esquema con la relación entre los distintos módulos de Bacula.

El objetivo es mostrar una configuración de ejemplo tanto para Windows cómo para Linux y realizar un par de pruebas sencilla para comprobar el funcionamiento de Bacula tanto en el respaldo como en la restauración de archivos.

Es así que en el cliente Linux se respaldó una carpeta situada en el escritorio, excluyendo algunos archivos, luego se eliminó y por último se restauró nuevamente con el fin de mostrar

principalmente la capacidad de Bacula para mantener los permisos de las carpetas en este sistema operativo.

En el cliente Windows se respaldó la carpeta donde el manejador de base de datos MySQL tiene almacenados los archivos de bases de datos, al igual que en el caso anterior se eliminó y se recuperó para testear la eficacia del programa en el respaldo de bases de datos en uso. Esto se debe a que Bacula utiliza el servicio VSS de Windows para realizar las tareas de respaldo y por eso los mismos pueden ser copiados sin que sufran problemas de inconsistencia.

9.1 Prueba en un cliente Linux

Para esta prueba fueron configuradas dos tareas, una de respaldo y otra de restauración. Los archivos a respaldar son los que se muestran en la Figura 43:

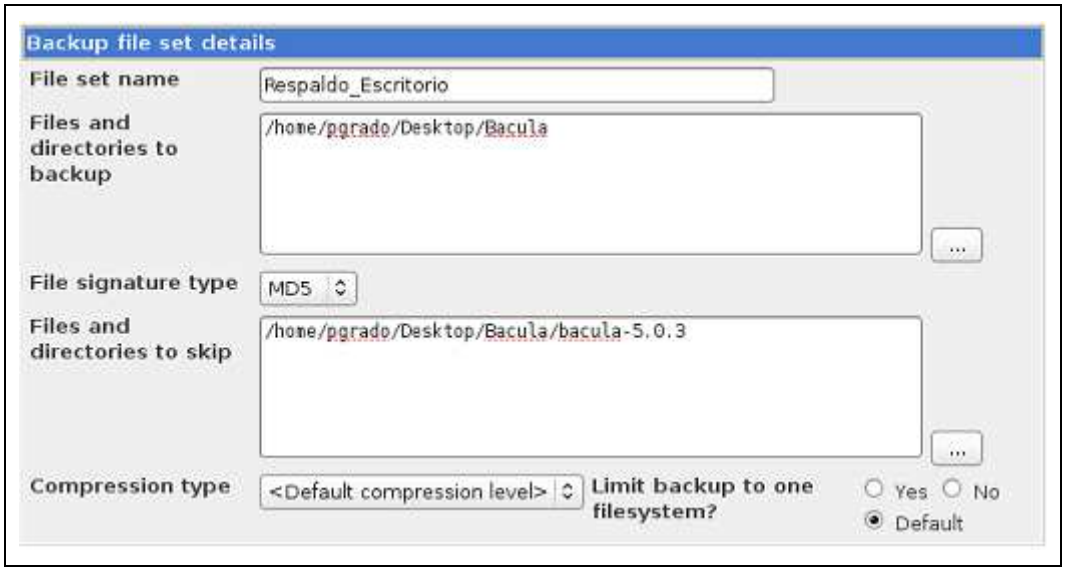


Figura 43 - Archivos para respaldar en Linux

Para facilitar la diferenciación de los permisos de la carpeta lo que se hizo fue asignarle permisos de lectura/escritura/ejecución al propietario, de lectura y escritura al grupo y solo de lectura al resto (CHMOD 764) como se muestra en la Figura 44.

```
debian:/home/pgrado/Desktop# ls -luR Bacula
Bacula:
total 12
drwxrw-r-- 2 pgrado pgrado 4096 ago 31 10:45 Conf_Ejemplo
drwxrw-r-- 2 pgrado pgrado 4096 ago 31 10:45 Conf_Original
drwxrw-r-- 2 pgrado pgrado 4096 ago 31 10:45 Otros

Bacula/Conf_Ejemplo:
total 28
-rwxrw-r-- 1 pgrado pgrado 5985 ago 31 10:45 bacula-dir.conf
-rwxrw-r-- 1 pgrado pgrado  945 ago 31 10:45 bacula-fd.conf
-rwxrw-r-- 1 pgrado pgrado 6406 ago 31 10:45 bacula-sd.conf
-rwxrw-r-- 1 pgrado pgrado  102 ago 31 10:45 bat.conf
-rwxrw-r-- 1 pgrado pgrado  159 ago 31 10:45 bconsole.conf

Bacula/Conf_Original:
total 28
-rwxrw-r-- 1 pgrado pgrado 6002 ago 31 10:45 bacula-dir.conf
-rwxrw-r-- 1 pgrado pgrado  945 ago 31 10:45 bacula-fd.conf
-rwxrw-r-- 1 pgrado pgrado 6406 ago 31 10:45 bacula-sd.conf
-rwxrw-r-- 1 pgrado pgrado  102 ago 31 10:45 bat.conf
-rwxrw-r-- 1 pgrado pgrado  159 ago 31 10:45 bconsole.conf

Bacula/Otros:
total 12
-rwxrw-r-- 1 pgrado pgrado 1743 ago 31 09:20 Instalar_Bacula.sh
-rwxrw-r-- 1 pgrado pgrado  168 ago 31 09:20 Instalar_VMWare_Tools.sh
-rwxrw-r-- 1 pgrado pgrado  446 ago 31 10:45 Red.txt
```

Figura 44 - Permisos de las carpetas entes del respaldo.

Luego de eso se ejecutó la tarea llamada “Respaldo_Escritorio” desde la interfaz de administración de Bacula obteniendo así un resultado exitoso como se indica en la Figura 45.

```
Starting backup job Respaldo_Escritorio ..

Automatically selected Catalog: BaculaDB
Using Catalog "BaculaDB"
A job name must be specified.
The defined Job resources are:
     1: Respaldo_Escritorio
     2: Restaurar_Escritorio
     3: Respaldar_mySQL_WinXP
     4: Restaurar_mySQL_WinXP
Select Job resource (1-4): 1
Run Backup job
JobName:  Respaldo_Escritorio
Level:    Full
Client:   debian-fd
FileSet:  Respaldo_Escritorio
Pool:     Respaldo (From Job resource)
Storage:  File (From Job resource)
When:     2010-08-31 11:08:17
Priority: 10
OK to run? (yes/mod/no):

.. the backup job is now running. When complete, the results will be shown below ..

  Build OS:               i486-pc-linux-gnu debian 5.0.1
  JobId:                  4
  Job:                    Respaldo_Escritorio.2010-08-31_11.08.17_24
  Backup Level:           Full
  Client:                 "debian-fd" 3.0.1 (30Apr09) i486-pc-linux-gnu,debian,5.0.1
  FileSet:                "Respaldo_Escritorio" 2010-08-31 11:06:54
  Pool:                   "Respaldo" (From Job resource)
  Catalog:                "BaculaDB" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         31-Aug-2010 11:08:17
  Start time:             31-Aug-2010 11:08:19
  End time:               31-Aug-2010 11:08:19
  Elapsed time:           0 secs
  Priority:               10
  FD Files Written:       17
  SD Files Written:       17
  FD Bytes Written:       29,568 (29.56 KB)
  SD Bytes Written:       31,736 (31.73 KB)
  Rate:                   0.0 KB/s
  Software Compression:   None
  VSS:                    no
  Encryption:             no
  Accurate:               no
  Volume name(s):         VolumenRespaldos
  Volume Session Id:      4
  Volume Session Time:    1283260732
  Last Volume Bytes:      65,266,444 (65.26 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK
```

Figura 45 - Tarea de respaldo exitosa.

Como siguiente paso se borró la carpeta recién respaldada del escritorio y se restauró el trabajo anterior mediante la ejecución de la tarea “Restaurar_Escritorio” obteniendo un también un resultado correcto (Figura 46)

```
Starting backup job Restaurar_Escritorio ..

Automatically selected Catalog: BaculaDB
Using Catalog "BaculaDB"
A job name must be specified.
The defined Job resources are:
     1: Respaldo_Escritorio
     2: Restaurar_Escritorio
     3: Respaldar_mySQL_WinXP
     4: Restaurar_mySQL_WinXP
Select Job resource (1-4): 2
Run Restore job
JobName:         Restaurar_Escritorio
Bootstrap:       /var/lib/bacula/client1.bsr
Where:           /
Replace:         always
FileSet:         Respaldo_Escritorio
Backup Client:
Restore Client:  debian-fd
Storage:         File
When:            2010-08-31 11:10:03
Catalog:         BaculaDB
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no):

.. the backup job is now running. When complete, the results will be shown below ..


31-Aug 11:10 debian-dir JobId 5: Start Restore Job Restaurar_Escritorio.2010-08-31_11.10.03_28
31-Aug 11:10 debian-dir JobId 5: Using Device "CarpetaRespaldos"
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         102 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         159 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         945 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado        6406 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado        5985 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: drwxrw-r--   2 pgrado   pgrado        4096 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         102 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         159 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         945 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado        6406 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado        6002 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: drwxrw-r--   2 pgrado   pgrado        4096 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         168 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado         446 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: -rwxrw-r--   1 pgrado   pgrado        1743 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: drwxrw-r--   2 pgrado   pgrado        4096 2010-08-31 10:44:24  /home/pgrado/Desktop/I
debian-fd JobId 5: drwxrw-r--   6 pgrado   pgrado        4096 2010-08-31 10:59:28  /home/pgrado/Desktop/I
31-Aug 11:10 debian-sd JobId 5: Ready to read from volume "VolumenRespaldos" on device "CarpetaRespaldos
31-Aug 11:10 debian-sd JobId 5: Forward spacing Volume "VolumenRespaldos" to file:block 0:65233738.
31-Aug 11:10 debian-sd JobId 5: End of Volume at file 0 on device "CarpetaRespaldos" (/backups), Volume
31-Aug 11:10 debian-sd JobId 5: End of all volumes.
31-Aug 11:10 debian-dir JobId 5: Bacula debian-dir 3.0.1 (30Apr09): 31-Aug-2010 11:10:05
  Build OS:               i486-pc-linux-gnu debian 5.0.1
  JobId:                  5
  Job:                    Restaurar_Escritorio.2010-08-31_11.10.03_28
  Restore Client:         debian-fd
  Start time:             31-Aug-2010 11:10:05
  End time:               31-Aug-2010 11:10:05
  Files Expected:         0
  Files Restored:         17
  Bytes Restored:         29,568
  Rate:                   0.0 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

Figura 46 - de restauración exitosa.

Por último se corroboró que los permisos de las carpetas restauradas fueran los mismos que antes del respaldo como lo representa la Figura 47:

```
debian:/home/pgrado/Desktop# ls -luR Bacula
Bacula:
total 12
drwxrw-r-- 2 pgrado pgrado 4096 ago 31 11:12 Conf_Ejemplo
drwxrw-r-- 2 pgrado pgrado 4096 ago 31 11:12 Conf_Original
drwxrw-r-- 2 pgrado pgrado 4096 ago 31 11:12 Otros

Bacula/Conf_Ejemplo:
total 28
-rwxrw-r-- 1 pgrado pgrado 5985 ago 31 11:06 bacula-dir.conf
-rwxrw-r-- 1 pgrado pgrado  945 ago 31 11:06 bacula-fd.conf
-rwxrw-r-- 1 pgrado pgrado 6406 ago 31 11:06 bacula-sd.conf
-rwxrw-r-- 1 pgrado pgrado  102 ago 31 11:06 bat.conf
-rwxrw-r-- 1 pgrado pgrado  159 ago 31 11:06 bconsole.conf

Bacula/Conf_Original:
total 28
-rwxrw-r-- 1 pgrado pgrado 6002 ago 31 11:11 bacula-dir.conf
-rwxrw-r-- 1 pgrado pgrado  945 ago 31 11:11 bacula-fd.conf
-rwxrw-r-- 1 pgrado pgrado 6406 ago 31 11:11 bacula-sd.conf
-rwxrw-r-- 1 pgrado pgrado  102 ago 31 11:11 bat.conf
-rwxrw-r-- 1 pgrado pgrado  159 ago 31 11:11 bconsole.conf

Bacula/Otros:
total 12
-rwxrw-r-- 1 pgrado pgrado 1743 ago 31 11:06 Instalar_Bacula.sh
-rwxrw-r-- 1 pgrado pgrado  168 ago 31 11:06 Instalar_VMWare_Tools.sh
-rwxrw-r-- 1 pgrado pgrado  446 ago 31 11:11 Red.txt
```

Figura 47 - Permisos de las carpetas después de la restauración

9.2 Prueba en un cliente Windows

Al igual que para el caso anterior esta prueba también tiene dos tareas configuradas, un respaldo y una restauración. El respaldo tendrá como objetivo almacenar todo el contenido de la carpeta con la información de las bases de datos que está manejando MySQL (Figura 48).

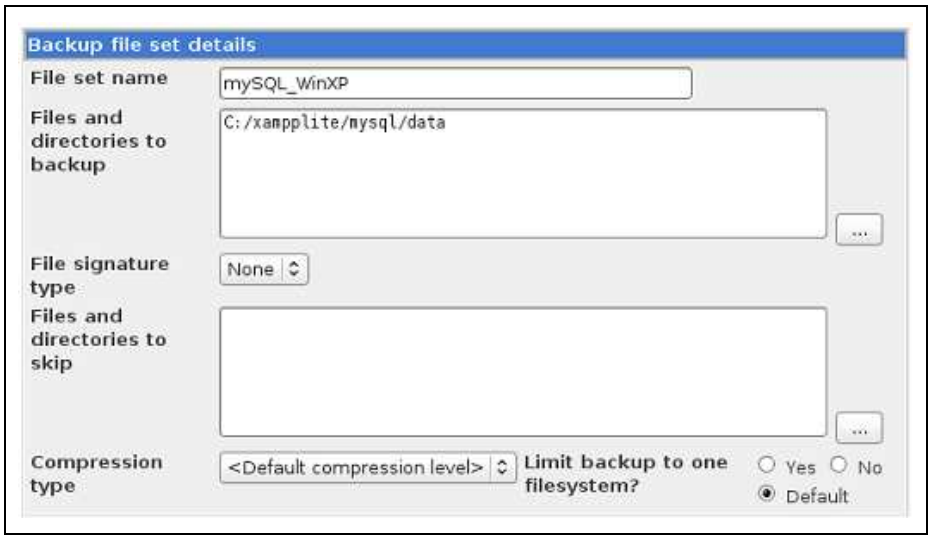


Figura 48 - Archivos para respaldar en Windows

Para probar la capacidad de respaldar bases de datos en uso (Hot backup) ejecutamos phpMyAdmin y abrimos una base cualquiera como en la Figura 49 con el fin de simular que tiene actividad.

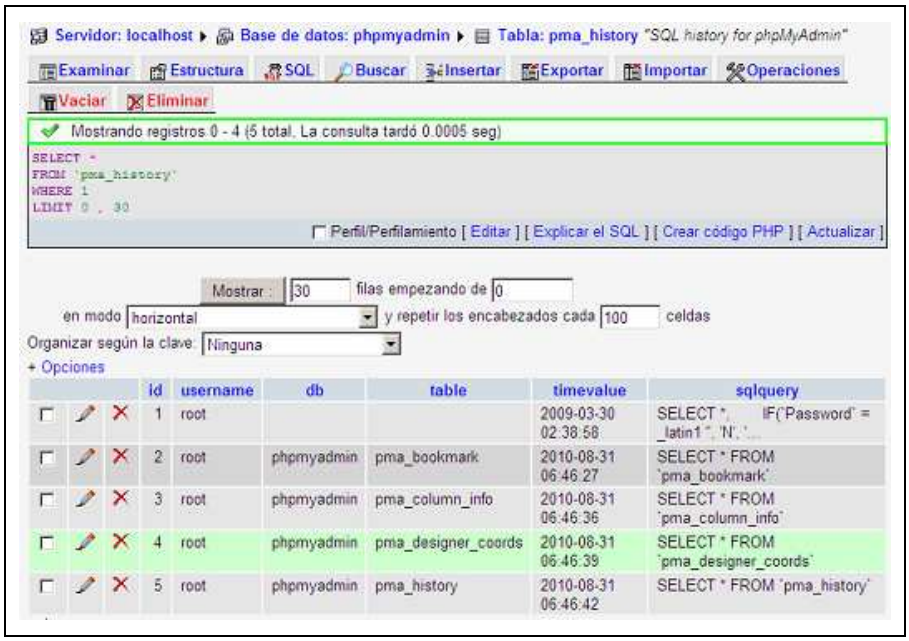


Figura 49 - Base de datos en uso

Luego de eso se ejecutó la tarea llamada “respaldar_MySQL_WinXP” que realiza un respaldo completo de los archivos desde la interfaz de administración de Bacula obteniendo así un resultado exitoso. Es importante observar también que durante la ejecución del respaldo, el sistema nos indica que se está utilizando VSS de Windows (Figura 50).

```
Starting backup job Respaldar_mySQL_WinXP ..

Automatically selected Catalog: BaculaDB
Using Catalog "BaculaDB"
A job name must be specified.
The defined Job resources are:
     1: Respaldo_Escritorio
     2: Restaurar_Escritorio
     3: Respaldar_mySQL_WinXP
     4: Restaurar_mySQL_WinXP
Select Job resource (1-4): 3
Run Backup job
JobName:  Respaldar_mySQL_WinXP
Level:    Full
Client:   test-XP-fd
FileSet:  mySQL_WinXP
Pool:     Respaldo (From Job resource)
Storage:  File (From Job resource)
When:     2010-08-31 11:13:57
Priority: 10
OK to run? (yes/mod/no):

.. the backup job is now running. When complete, the results will be shown below ..
  Build OS:               i486-pc-linux-gnu debian 5.0.1
  JobId:                  6
  Job:                    Respaldar_mySQL_WinXP.2010-08-31_11.13.57_32
  Backup Level:           Full
  Client:                 "test-XP-fd" 5.0.3 (04Aug10) Linux,Cross-compile,Win32
  FileSet:                "mySQL_WinXP" 2010-08-31 11:13:57
  Pool:                   "Respaldo" (From Job resource)
  Catalog:                "BaculaDB" (From Client resource)
  Storage:                "File" (From Job resource)
  Scheduled time:         31-Aug-2010 11:13:57
  Start time:             31-Aug-2010 11:14:00
  End time:               31-Aug-2010 11:14:22
  Elapsed time:           22 secs
  Priority:               10
  FD Files Written:       108
  SD Files Written:       108
  FD Bytes Written:       23,591,805 (23.59 MB)
  SD Bytes Written:       23,607,216 (23.60 MB)
  Rate:                   1072.4 KB/s
  Software Compression:   None
  VSS:                    yes
  Encryption:             no
  Accurate:               no
  Volume name(s):         VolumenRespaldos
  Volume Session Id:      6
  Volume Session Time:    1283260732
  Last Volume Bytes:      88,894,076 (88.89 MB)
  Non-fatal FD errors:    0
  SD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Backup OK
```

Figura 50 - Tarea de respaldo exitosa utilizando VSS

Luego de tener los datos respaldados ingresamos a phpMyAdmin y eliminamos 3 bases de datos propias del programa (Figura 51) y luego actualizamos la pantalla del explorador.

Al ser necesarias las bases que borramos, el programa dejó de funcionar correctamente obteniéndose así el mensaje de error de la Figura 52.

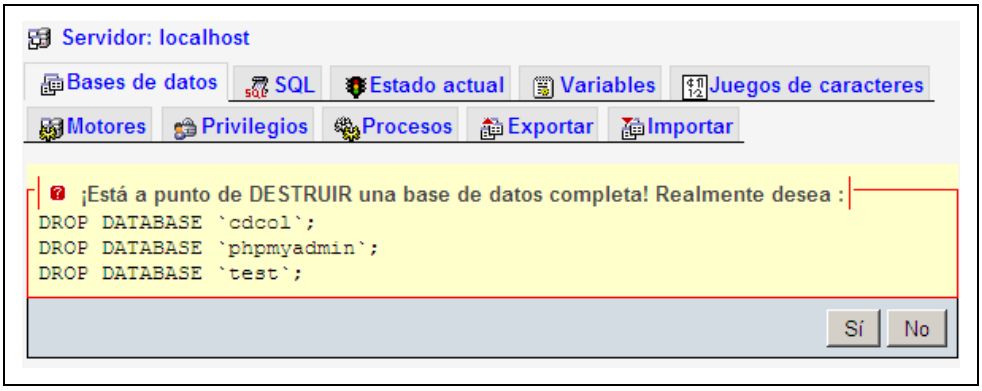


Figura 51 - Bases borradas en phpMyAdmin

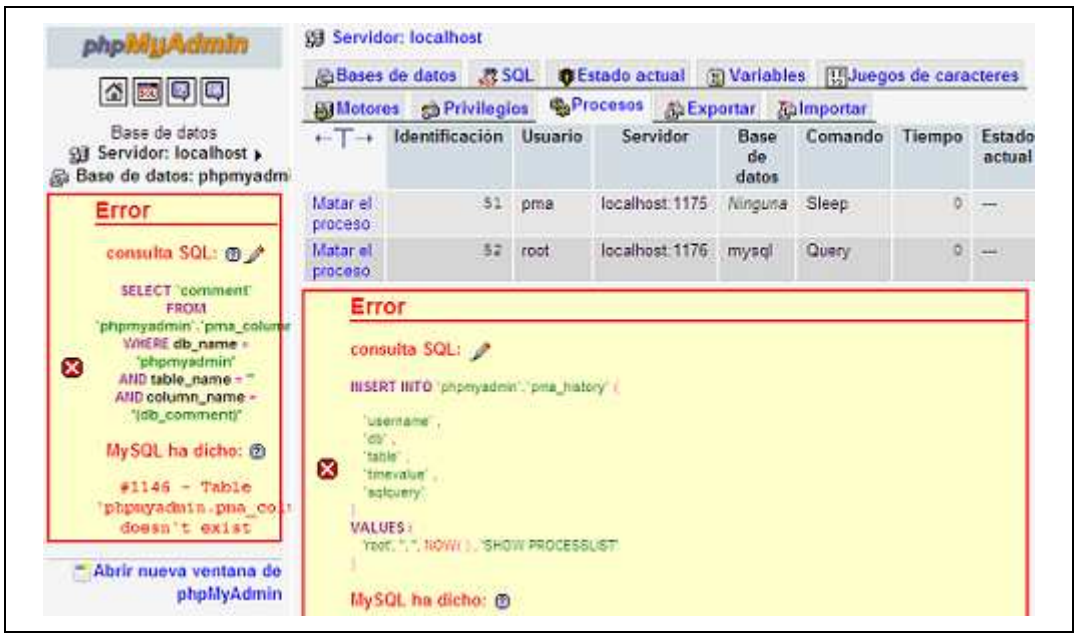


Figura 52 - Error de phpMyAdmin luego de borrar las bases

Finalmente ejecutamos la tarea de restauración de los archivos de la base de datos a la cual se le llamó “Restaurar_mySQL_WinXP” de forma exitosa (Figura 53) y volvimos a refrescar el explorador para comprobar que el problema que se había generado en phpMyAdmin al borrar las bases de datos había quedado solucionado cómo se observa en la Figura 54.

```
Starting backup job Restaurar_mySQL_WinXP ..

Automatically selected Catalog: BaculaDB
Using Catalog "BaculaDB"
A job name must be specified.
The defined Job resources are:
     1: Respaldo_Escritorio
     2: Restaurar_Escritorio
     3: Respaldar_mySQL_WinXP
     4: Restaurar_mySQL_WinXP
Select Job resource (1-4): 4
Run Restore job
JobName:         Restaurar_mySQL_WinXP
Bootstrap:       /var/lib/bacula/winXP.bsr
Where:           *None*
Replace:         always
FileSet:         mySQL_WinXP
Backup Client:
Restore Client:  test-XP-fd
Storage:         File
When:            2010-08-31 11:18:36
Catalog:         BaculaDB
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no):

.. the backup job is now running. When complete, the results will be shown below ..


31-Aug 11:18 test-XP-fd JobId 8: -rwxrwxrwx   1 0        0               0 2010-08-31 07:39:15  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: -rwxrwxrwx   1 0        0            2048 2010-08-31 07:39:15  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: -rwxrwxrwx   1 0        0            8656 2010-08-31 07:39:14  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: -rwxrwxrwx   1 0        0               0 2010-08-31 07:39:15  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: -rwxrwxrwx   1 0        0            2048 2010-08-31 07:39:15  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: drwxrwxrwx   1 0        0               0 2010-08-31 07:39:14  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: drwxrwxrwx   1 0        0               0 2010-08-30 17:10:33  C:/xam
31-Aug 11:18 test-XP-fd JobId 8: drwxrwxrwx   1 0        0               0 2010-08-30 17:10:22  C:/xam
31-Aug 11:18 debian-dir JobId 8: Bacula debian-dir 3.0.1 (30Apr09): 31-Aug-2010 11:18:40
  Build OS:               i486-pc-linux-gnu debian 5.0.1
  JobId:                  8
  Job:                    Restaurar_mySQL_WinXP.2010-08-31_11.18.36_40
  Restore Client:         test-XP-fd
  Start time:             31-Aug-2010 11:18:38
  End time:               31-Aug-2010 11:18:40
  Files Expected:         0
  Files Restored:         108
  Bytes Restored:         23,591,805
  Rate:                   11795.9 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

Figura 53 - Tarea de restauración exitosa

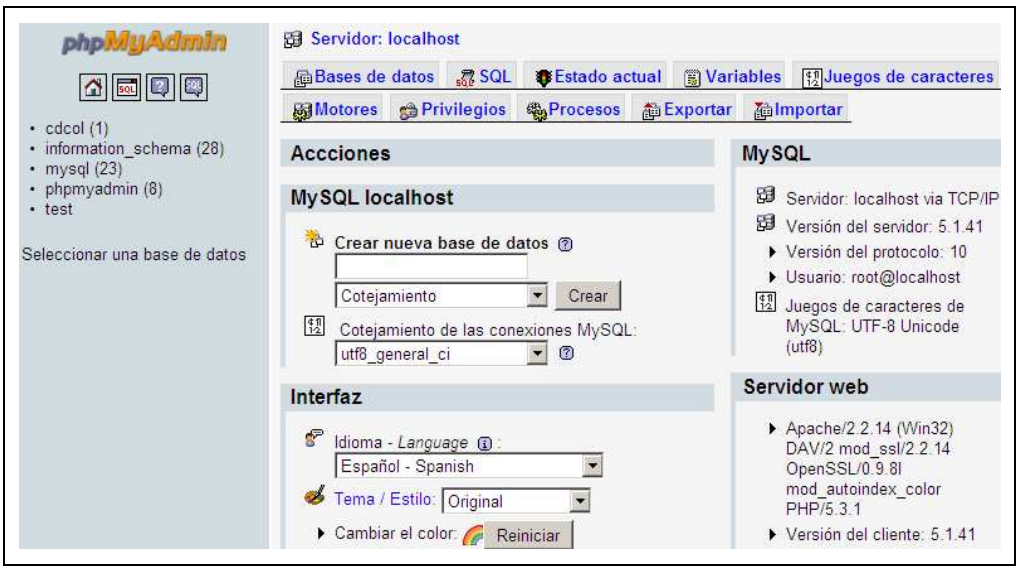


Figura 54 - phpMyAdmin luego de la restauración

10 Casos de prueba Sistema de Registros

En esta sección se detallan los casos de prueba más relevantes realizados a cada uno de los aplicativos desarrollados.

10.1 Generador de mensajes ATNA

Los casos de prueba para el generador se encuentran en el documento [anexo 9], junto con la explicación de cómo se crearon las pruebas y qué mensaje ATNA se genera en cada caso. Esto está pensado para que sea de utilidad para los desarrolladores del Hospital de Clínicas quienes contarán con ejemplos de utilización de la herramienta.

10.2 Sistema de Registro de Eventos

Los casos de prueba para las funcionalidades del sistema de registro de eventos se tomaron del caso de estudio, ya que en él se utiliza el mismo para procesar y almacenar todos los mensajes generados por el cliente.

Los casos de prueba para medir la performance del sistema se hizo en base a un cliente que genera exactamente el mismo mensaje de auditoría en todas las instancias de prueba. Se definió un formato y contenido de mensaje de auditoría (ver Figura 55), de manera que sea similar a los que podrían ser utilizados en una comunicación real, ya que si se utilizaran mensajes de otras características estaríamos evaluando erróneamente el comportamiento y la performance del sistema.

El objetivo de este tipo de pruebas es entonces medir el tiempo de respuesta del servicio web, bajo cierto ambiente y para distintas configuraciones del mismo, como por ejemplo con seguridad no habilitada, seguridad habilitada, etc.

El tiempo de respuesta del servicio web se tomó del monitor de performance que incluye el propio servidor Glassfish. Este tiempo de respuesta es el tiempo en que le toma al servicio web ejecutar los algoritmos de seguridad para verificar la validez del mensaje y colocar el mismo en la cola para su posterior procesamiento.

El tiempo de transmisión del mensaje, es decir el tiempo en que demora el envío desde una maquina cliente al servidor no es tenido en cuenta ya que en primer lugar los servicios web funcionan asíncronamente y no devuelven ningún resultado, por lo que el cliente no necesita esperar, y segundo que lo que realmente debe medirse para determinar futuros problemas de disponibilidad es el tiempo de respuesta.

La Figura 55 presenta el mensaje de auditoría utilizado para las pruebas:

```
<?xml version="1.0" encoding="UTF-8"?>
<AuditMessage>
  <EventIdentification EventActionCode="C" EventDateTime="2010-05-18T00:07:23" EventOutcomeIndicator="0">
    <EventID code="03" codeSystem="OID HC" codeSystemName="Grupo Eventos del HC"
             displayName="Historia Clínica" originalText="" />
    <EventTypeCode code="20" codeSystem="OID HC" codeSystemName="Grupo Eventos del HC"
                   displayName="Modificacion de Historia Clinica" originalText="" />
  </EventIdentification>
  <ActiveParticipant UserID="Y.YYY.YYY-Y" AlternativeUserID="Jose Perez" UserName="jose"
                     UserIsRequestor="true" NetworkAccessPointID="192.168.1.2" NetworkAccessPointTypeCode="2">
    <RoleIDCode code="3" codeSystem="OID HC" codeSystemName="Roles del HC" displayName="Medico" />
  </ActiveParticipant>
  <AuditSourceIdentification AuditSourceID="Pacientes para Atender" AuditEnterpriseSiteID="OID HC">
    <AuditSourceTypeCode code="8" />
  </AuditSourceIdentification>
  <ParticipantObjectIdentification ParticipantObjectID="4.584.507-7" ParticipantObjectTypeCode="1"
                                   ParticipantObjectTypeCodeRole="1" ParticipantObjectDataLifeCycle="1">
    <ParticipantObjectIDTypeCode code="11" codeSystem="DNIC" codeSystemName="CI" />
    <ParticipantObjectName>Martin Calabria</ParticipantObjectName>
  </ParticipantObjectIdentification>
</AuditMessage>
```

Figura 55- Mensaje de auditoría reducido para el caso de prueba.

Como se aprecia, el mensaje contiene cada uno de los cuatro campos disponibles, repetido una única vez, es decir un campo *EventIdentification* con la identificación del evento, un elemento *ActiveParticipant* con la identificación del usuario, un elemento *AuditSourceIdentification* con la identificación de la fuente del mensaje, y un elemento opcional, del tipo *ParticipantObjectIdentification* con los datos del paciente en cuestión. Este podría ser el tipo de mensaje que más se ajuste a la realidad del hospital de clínicas por lo que será el primero en ser evaluado.

A continuación se presentan los casos de prueba:

Caso 1 - Servicio Web sin implementación de mecanismos de seguridad.

Para tener una estimación del tiempo necesario para ejecutar los algoritmos de seguridad, primero se realizó una medición del tiempo de respuesta del servicio web sin aplicar ningún mecanismo de seguridad.

Caso 2 - Servicio Web con autenticación mediante UsernameToken.

En la teoría, la autenticación mediante usuario y clave es más performante que la autenticación mediante certificados, este caso de prueba tiene como objetivo medir el tiempo de respuesta del primero para posteriormente compararlo con el segundo.

Caso 3 - Servicio Web con autenticación bidireccional mediante certificados mutuos.

10.2.1 Entorno de Prueba

El entorno de pruebas se elaboró teniendo en cuenta los siguientes puntos:

- Todos los módulos del sistema de registro de eventos se desplegarán en un mismo servidor, esto es: módulo de servicios, módulo de cola de mensajes, módulo de negocio ejb y módulo de persistencia. El servidor será dedicado a ejecutar únicamente el sistema de registro de eventos, a modo de maximizar la performance del servicio web.

- La base de datos se instalará en el mismo servidor en el que ejecuta el sistema de registro de eventos.

- El componente cliente de prueba se instalará en un servidor distinto. Con esto se logra generar el contenido de los mensajes a transmitir en el cliente, sin desperdiciar recursos del servidor de registros. Por otro lado, esto es posible hacerlo ya que el objetivo es medir el tiempo de respuesta del servicio web y no el tiempo de transferencia de mensajes por la red.

El entorno donde se ejecutaron las pruebas se compone de:

- Sistema operativo: Debian 5.0.3 kernel linux 2.6-26-2-686.
- CPU: Intel Pentium DualCore E5200 2.5GHz.
- Motherboard ASRock ALiveNF6P-VSTA
- Memoria del sistema 2000MB (DDR2-800 SDRAM)
- Tarjeta de red: 100 Mb/s.
- Servidor de aplicaciones: Glassfish V2.1.
- Servidor de base de datos: PostgreSQL 8.4
- Java JDK 1.6 update 20.

El entorno de pruebas tuvo lugar en las instalaciones del DPI dentro del Hospital de Clínicas, de manera de utilizar tanto servidores como clientes propios del hospital y tener una mejor aproximación a la performance del sistema de registros en el ambiente de producción.

Se realizaron pruebas de estress para cada uno de los casos de prueba obteniéndose los siguientes resultados:

10.2.2 Caso de Prueba 1 – Servicio Web sin seguridad

La siguiente tabla presenta los resultados obtenidos para el el servicio web sin aplicar ningún mecanismo de seguridad:

Cantidad de Mensajes	Tiempo medio de Respuesta (ms)	Tiempo mínimo de Respuesta (ms)	Tiempo máximo de Respuesta (ms)	Rendimiento (solicitudes por segundo)
10	4	3	8	250
20	18	3	62	55,56
50	33	4	161	30,3
100	39	3	200	25,64

Como se aprecia en la tabla, conforme aumenta la cantidad de mensajes enviados al servidor, los tiempos de respuesta aumentan, causando un rendimiento menor. Esto posiblemente se debe a la sobrecarga en el servidor, el cual maneja las peticiones concurrentemente. A mayor número de peticiones mayor será la concurrencia (asumiendo que la frecuencia de arribo de solicitudes es mayor a la del procesamiento de las mismas), y por lo tanto el tiempo promedio de procesamiento se verá incrementado. El servidor mantiene un tope configurable, en la ejecución de peticiones concurrentes, de manera de brindar mayor o menor disponibilidad, pero como contrapartida se tendrán menores y mayores tiempos de respuesta respectivamente en caso que dicho tope no se ajuste a la frecuencia de mensajes recibidos.

Más allá de la caída del rendimiento, el mismo es muy aceptable para las etapas iniciales de la puesta en producción, ya que según los datos obtenidos en el DPI, se espera un pequeño conjunto de clientes enviando mensajes con una frecuencia relativamente baja. Se estima recibir 1 o 2 peticiones por segundo.

10.2.3 Caso de Prueba 2 – Servicio Web con mecanismo Usertoken Name

La siguiente tabla presenta los resultados cuando se aplica el mecanismo de seguridad de usuario – contraseña (Usertoken Name):

Cantidad de Mensajes	Tiempo medio de Respuesta (ms)	Tiempo mínimo de Respuesta (ms)	Tiempo máximo de Respuesta (ms)	Rendimiento (solicitudes por segundo)
10	29	4	108	34,48
20	31	7	91	32,26
50	37	4	100	27,03
100	42	3	177	23,81

Claramente, el rendimiento es menor al caso 1, principalmente debido al tiempo requerido para validar las credenciales del usuario. Las mismas vienen incluidas en el mensaje SOAP, aplicando un algoritmo digest (SHA-1) sobre la clave, para mayor seguridad. Este mecanismo corresponde a los mecanismos de clave simétrica, y para cifrar la clave se utiliza el certificado del servidor. Dentro de este, las credenciales de los usuarios habilitados se encuentran en un archivo XML, por lo tanto se debe consultar el mismo en cada petición, aplicar el algoritmo SHA-1 y comparar dicho resultado con el digest que se encuentra en el mensaje SOAP.

El rendimiento del servicio continúa siendo aceptable.

10.2.4 Caso de Prueba 3 – Servicio Web con mecanismo de autenticación mutua

La siguiente tabla presenta los resultados cuando se aplica autenticación mutua mediante certificados:

Cantidad de Mensajes	Tiempo medio de Respuesta (ms)	Tiempo mínimo de Respuesta (ms)	Tiempo máximo de Respuesta (ms)	Rendimiento (solicitudes por segundo)
10	54	37	119	18,51
20	51	25	93	18,52
50	54	25	100	19,61
100	57	3	171	17,54

Los resultados muestran un rendimiento inferior al caso 2 ya que en general los mecanismos de seguridad con clave simétrica tienen mejor performance que los de clave publica.

Como conclusión general se puede afirmar que aún en este caso el rendimiento del sistema es optimo para la carga a la cual será sometido en el ambiente de producción, por lo tanto se recomienda la utilización de seguridad en el servicio web, en particular el mecanismo de autenticación bidireccional.

11 Gestión del Proyecto

Este capítulo tiene como objetivo describir la planificación, los cambios en la misma, y los motivos de dichos cambios. Se comienza con la descripción del alcance del proyecto y luego se presentan los cronogramas para cumplir con ellos. Por último se muestran una serie de gráficos que reflejan la dedicación y esfuerzo del equipo en cada una de las actividades del proyecto.

11.1 Alcance del Proyecto

El alcance de este proyecto abarca los siguientes puntos:

- Estudio de estándares y perfiles existentes para el área de la salud, en particular los que aplican al registro de eventos y a la seguridad y auditoría de sistemas, recomendando la utilización de los que correspondan.
- Estudio del estado del arte en sistemas de registro de eventos evaluando la funcionalidad y adecuación de las herramientas existentes al contexto del Hospital de Clínicas.
- Diseño y desarrollo de un sistema de registro de eventos que cumpla con los requisitos definidos por el Hospital de Clínicas, en caso que ninguna herramienta se adecue a los mismos.
- Diseño y desarrollo de una interfaz web para el sistema de registro de eventos, que permita auditar de forma remota todos los eventos registrados en el sistema.
- Implantación en el ambiente de testeo del Hospital de Clínicas de ambos sistemas.
- Estudio del estado del arte en sistemas de respaldo de información, evaluando las herramientas existentes.
- Elección, instalación e ejecución de la herramienta que mejor se adecue a los requisitos impuestos por el Hospital de Clínicas.

11.2 Planificación Inicial vs Real

Inicialmente se estimó una duración de proyecto de aproximadamente 9 meses, con la participación de tres personas. La Figura 56 describe la planificación inicial para el desarrollo del proyecto en los tiempos establecidos:

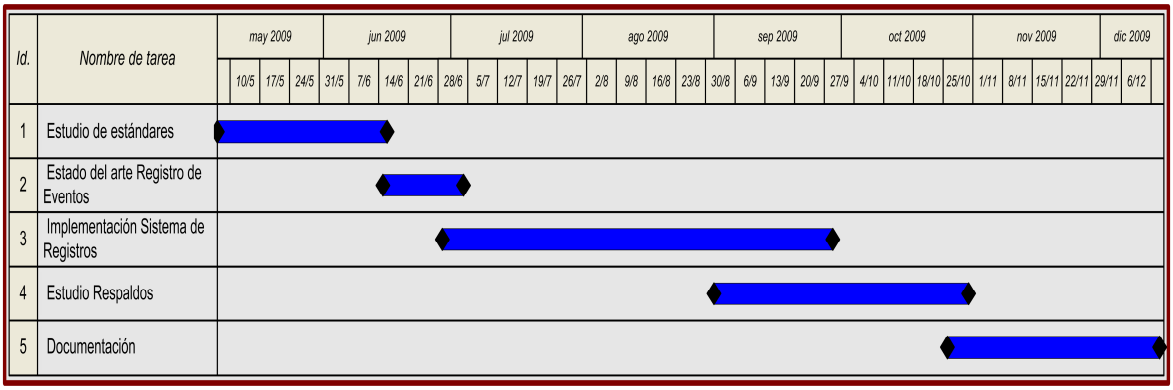


Figura 56 - Cronograma inicial

Se identificaron cinco actividades principales y se definió como estrategia que los tres integrantes del equipo trabajaran conjuntamente sobre ellas, a excepción de la implementación del sistema, tarea que sería solapada con el estudio de herramientas de respaldos para poder llegar en plazo a la fecha de fin de proyecto.

Esta planificación era ambiciosa, y se acordó una posible extensión del plazo en caso que las actividades de implementación necesitaran más tiempo del estimado, debido posiblemente a la no adecuación de ninguna herramienta existente que pudiera ser reutilizada en el desarrollo de la solución.

La planificación real del proyecto difiere respecto de la inicial ya que el tiempo estimado no fue suficiente para cumplir con todas las tareas. Esto se debe principalmente a los siguientes motivos:

- Mala estimación del esfuerzo requerido para algunas actividades, sobre todo implementación del sistema de registros e investigación de herramientas de respaldo. La solución propuesta fue construida completamente por el grupo sin posibilidades de incorporar alguna herramienta de registro de eventos existente.
- Poca participación de un integrante y posterior abandono a mediados del proyecto derivo en un incremento del tiempo en prácticamente todas las actividades.

- Poca dedicación de los integrantes en algunos pasajes del proyecto, sobre todo en los meses 1, 6 y 7 (ver Figura 61 - Horas invertidas durante el proyecto).
- Investigación y prueba de concepto sobre tecnologías que finalmente no fueron utilizadas, como es el caso de PHP (Framework Cake).

En este punto podría haberse planteado una renegociación del alcance del proyecto, recortando el mismo para poder cumplir en tiempo y forma con los requerimientos principales. Esto se descartó principalmente por el gran interés del Hospital de Clínicas en realizar la totalidad del alcance, lo cual les permitiría disponer de un prototipo para la gestión de registros de eventos y una herramienta para la gestión de respaldos, así como documentación de ambos.

En la Figura 57 se presenta el cronograma seguido durante el proyecto:

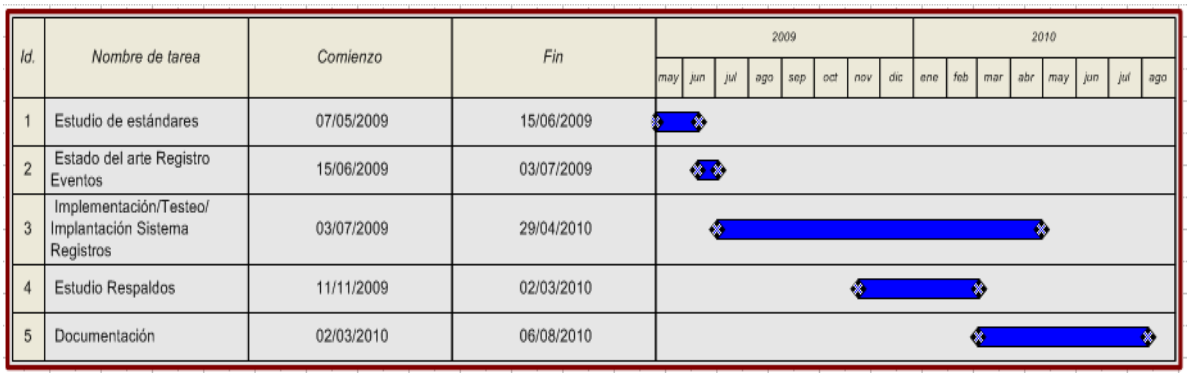


Figura 57 - Planificación del Proyecto

Como se aprecia en la Figura 57, la duración real del proyecto fue de aproximadamente 15 meses, prácticamente el doble de lo estimado inicialmente.

Se comenzó con el estudio de estándares y perfiles del área de salud y seguridad (ATNA, HL7, IHE, WS-Security) y con el estado del arte en el área de registro de eventos para determinar qué porcentaje del sistema debía ser implementado y que componentes se podían reutilizar de soluciones ya existentes.

La implementación se realizó sin utilizar componentes de terceras partes por lo que fue la actividad que demando mayor tiempo, aproximadamente 10 meses, siendo solapada con el estudio de respaldos y documentación final.

Los hitos logrados durante el proyecto son los siguientes:

- 01/11/2009 se libera primera versión Sistema de Registro de Eventos.

- 01/03/2010 se finaliza estudio y documentación de Respaldos.
- 15/03/2010 se libera primera versión Sistema de Auditoría de Eventos.
- 01/05/2010 se libera versión final Sistema de Registro de Eventos.
- 10/05/2010 se libera versión final Sistema de Auditoría de Eventos.
- 10/06/2010 se implanta y testea versiones finales de los sistemas en DPI.
- 30/08/2010 se finaliza documentación del proyecto.

11.3 Actividades

A continuación se presentan gráficas con las actividades realizadas así como el tiempo dedicado a cada una de ellas.

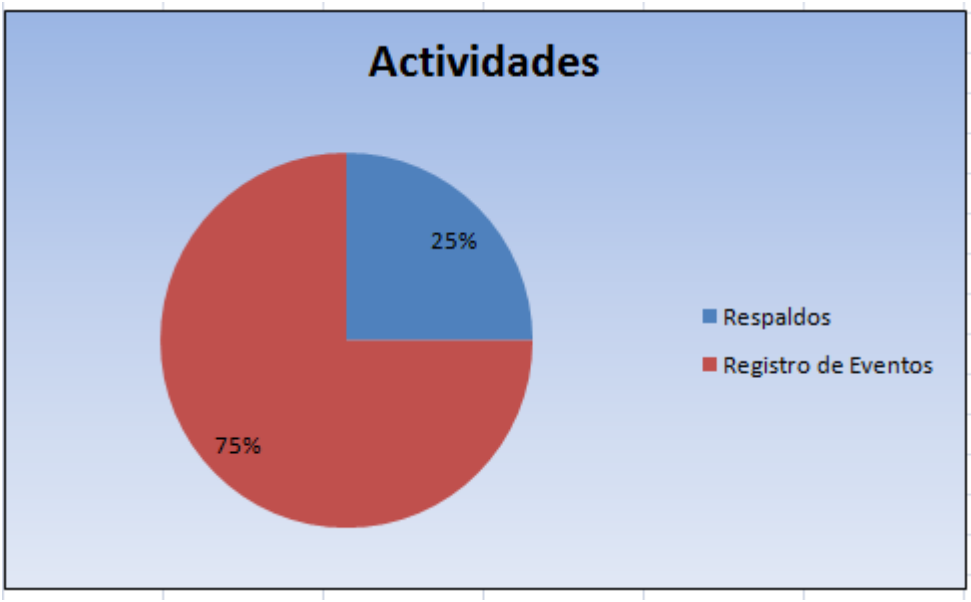


Figura 58 - Principales actividades del proyecto

La Figura 58 muestra claramente una mayor dedicación a la temática de registro de eventos que a la de respaldos de información. Esto se debe principalmente a los intereses del personal del DPI, quien priorizó la investigación de estándares y herramientas en el área de registro de eventos, así como la puesta en producción de un sistema de registro de eventos capaz de registrar y auditar eventos remotamente. Por otro lado, la necesidad de diseñar e implementar dicho sistema, adecuado a los requisitos del DPI, llevó a que la dedicación a esta área fuera aún mucho mayor.

La Figura 59 presenta las principales tareas realizadas con respecto al respaldo de información:

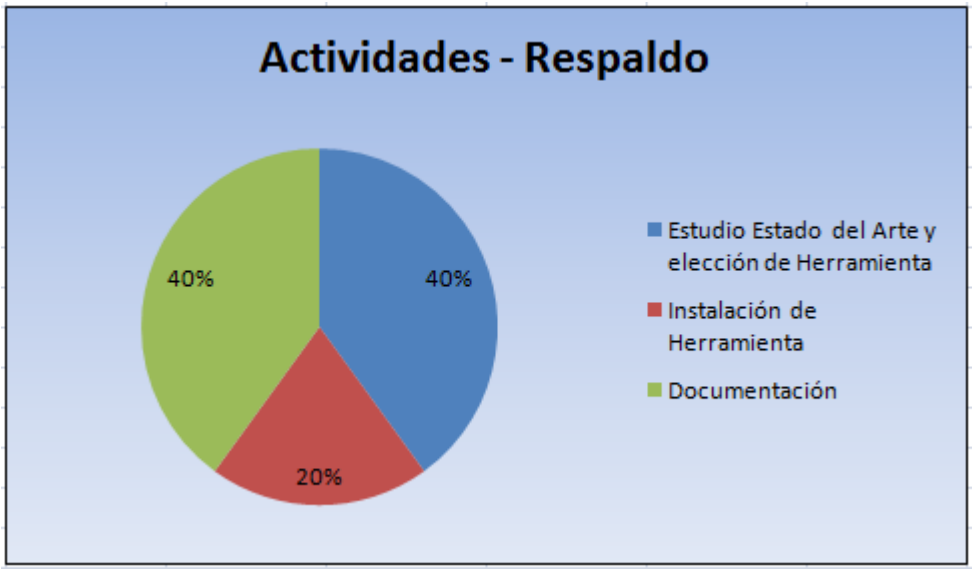


Figura 59 - Dedicación a actividades relacionadas con el respaldo de información

Las grandes actividades con respecto a la temática de respaldos corresponden al estudio del estado del arte, analizando las herramientas existentes, su adecuación al contexto del Hospital de Clínicas y la elección de la que se considere mejor.

Por otro lado se requirió un 20% del tiempo para instalar y configurar la herramienta seleccionada ya que contiene diversos módulos que deben instalarse tanto en el servidor de respaldos como en el cliente.

Figura 60 - Dedicación a actividades relacionadas con el registro de eventos

La mayor dedicación en el área de registro de eventos corresponde al diseño e implementación la solución propuesta. Las actividades de implementación que se tienen en cuenta son las siguientes:

- Reuniones de relevamiento de requisitos en el DPI.
- Análisis de la realidad actual y diseño de la solución.
- Investigación de tecnologías y pruebas de concepto.
- Implementación de los sistemas y módulos.
- Correcciones y pedidos realizados por el DPI luego del testeo.

La implantación se lleva un porcentaje mayor al esperado ya que se debió generar y configurar un ambiente en el propio DPI para luego poder implantar los sistemas.

Por último la documentación incluye todas las secciones del informe final que hacen referencia al registro de eventos y todos los anexos escritos como complemento de la solución (guías de instalación y configuración, manuales de usuario, casos de prueba, etc.).

La Figura 61 detalla el esfuerzo (en horas) mes a mes del equipo durante el transcurso del proyecto.

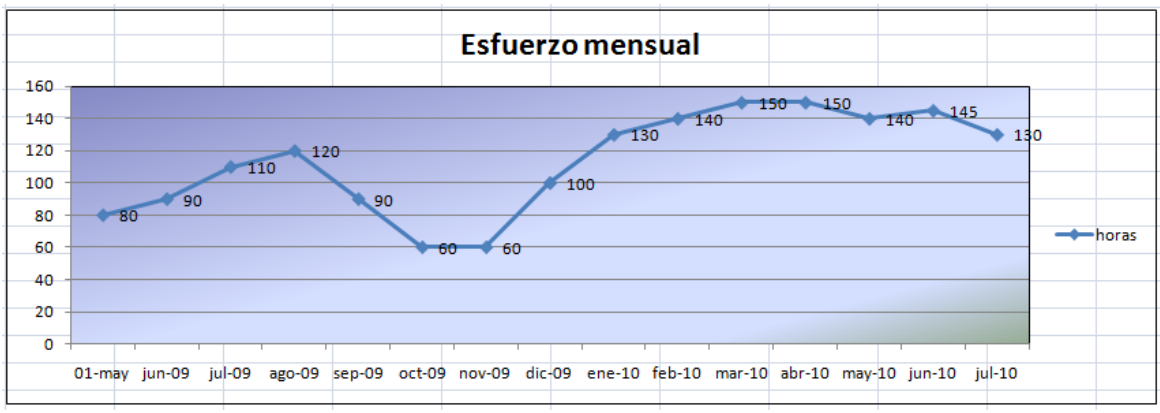


Figura 61 - Horas invertidas durante el proyecto

Como se ve claramente en la Figura 61, no fue posible realizar un trabajo sostenido y parejo durante los 15 meses, sino que hubo extremos en la dedicación al proyecto. En los primeros meses existió una falta de concientización sobre el esfuerzo requerido y volumen del proyecto, principalmente por la definición tardía del alcance del mismo. Luego se comienza a registrar un incremento en el esfuerzo hasta el fin del proyecto que permite realizar simultáneamente tareas de implementación, estudio y documentación (ver Figura 57).

12 Resumen y Conclusiones

El proyecto tuvo desde su propuesta dos frentes pero un mismo objetivo, incrementar, perfeccionar y modernizar la medidas de seguridad existentes en el Hospital de clínicas.
Para lograr este objetivo los clientes presentaron pautas generales y guías en cuanto a las áreas de investigación.

12.1 Resumen

A continuación se presentan un sumario del trabajo realizado.

Sistema de Respaldo

En base a esto se planteó por un lado, la búsqueda e investigación de las herramientas actuales de respaldo de información para así realizar un Estado del Arte con ventajas y desventajas de cada una de ellas con el fin de hacer una recomendación fundamentada de cuál es la que más funcionalidad brinda y mejor se adapta al entorno de trabajo existente en el Hospital.
El estudio abarco los siguientes programas de respaldos:

- AMANDA
- Areca Backup
- BackupPC
- Bacula
- EMC Legato NetWorker
- Acronis BackUp and Recovery

En base a dicho estudio se observo:

1. Las herramientas pagas ofrecen excelentes características, entre ellas, respaldos de bases de datos en uso, programación de tareas de respaldo y restauración y búsqueda de archivos, además de interfaces de administración y configuración muy amigables, intuitivas y completas pero su costo es elevado, en particular la versión “Acronis Backup & Recovery 10 Server” con 8 clientes, cuyo precio es aproximadamente U$S 1500, la mitad que la licencia del “Workgroup Edition” de “Legato Networker” con la misma cantidad de clientes, lo que se vuelve una solución extremadamente cara para instalar en las cientos de máquinas existentes en el Hospital de Clínicas.

2. De las herramientas gratuitas estudiadas, “Bacula” permite el respaldo de bases de datos en uso, programación de tareas de respaldo y restauración, búsqueda de archivos e interfaz gráfica donde realizar configuraciones y visualizar el estado de las tareas de respaldo.
Del resto de las programas, dos no respaldan bases de datos en uso (como Areca y BackupPC) y una no poseen interfaz gráfica para realizar configuraciones facilitar su utilización (AMANDA).
Es por esto que decidimos que, de las herramientas estudiadas, Bacula es la que mejor se adapta a los requerimientos especificados por los clientes.

Sistema de Registros

La otra propuesta era lograr la implementación de un sistema que permita a las aplicaciones registrar eventos generados en las mismas.
Las áreas de investigación en las que se enmarcó revisten gran interés en el entorno clínico, así como para la industria de las tecnologías de la información investigando técnicas emergentes que fueron el marco teórico para implementar el sistema.
Y si bien ya se contaba con conocimientos en algunas de las áreas, la compresión del resto, principalmente el perfil ATNA, los mecanismos de seguridad que debía proveer la aplicación para cumplir con el mismo y de las características de una arquitectura orientada a servicios y de los estándares involucrados demandaron un gran esfuerzo.

Decidimos diseñar e implementar en primera instancia un prototipo de la solución con el objetivo de atacar tempranamente estas dificultades, lo que fue muy beneficioso, porque saco a relucir problemas que en primera instancia no habíamos considerado, como fue la imposibilidad de utilizar WS-Security con la herramienta de desarrollo Genexus, la cual es muy utilizada por los clientes.

Por esta razón, y con la aprobación de ellos, implementamos la biblioteca en java que es utilizada por Genexus, para comunicarse con el Sistema de Registro de Eventos cumpliendo con los requisitos de seguridad necesarios.

12.2 Conclusiones

Por un lado se logró construir un sistema que permitirá la recepción de eventos y la auditoría de los mismos en una aplicación basada en estándares que se aplican al ambiente hospitalario, en particular el perfil ATNA, brindando así un producto que no está restringido a un dominio particular sino que funciona en todos los contextos que respeten el perfil antes mencionado. Este sistema es de suma importancia para el accionar del hospital ya que permitirá centralizar la recepción de eventos provenientes de los sistemas existentes utilizando un único formato, lo cual simplificará la tarea del administrador, y aplicando los mecanismos de seguridad que sean necesarios para prevenir ataques. La auditoría remota permitirá a cualquier usuario calificado acceder al sistema vía web para facilitar su tarea.

Su diseño modular y la utilización de tecnologías, plataformas y estándares actuales como son Java EE, SOA, WS-Security entre otros, permite que pueda ser fácilmente extensible a nuevas funciones y servicios.

A su vez se logró una muy buena integración con los sistemas que existen actualmente en el hospital debido a la implementación de dos módulos auxiliares que simplifican la interacción con el sistema. El primero es el módulo generador y validador de mensajes de auditoría, que se encarga de crear los mensajes a partir de la información suministrada por el sistema que lo utilice, liberando así a los desarrolladores de conocer en detalle la estructura del mensaje que se debe generar. El segundo módulo apuntó al desarrollo Java en Genexus y permitirá la comunicación con el servicio web donde se registran los eventos utilizando los mecanismos definidos por WS-Security ya que dicha herramienta no lo soporta.

Por otro lado estudiamos seis herramientas de respaldo lo que permitió elaborar un buen marco teórico acerca del tema optando por Bacula como sistema de respaldo el cual les otorgará a los clientes la posibilidad de unificar los respaldos de todos las maquinas del hospital facilitando así su administración. Si bien en el alcance del sistema no estaba contemplado, decidimos realizar la instalación del mismo y la puesta en marcha del programa para así brindarles una posible configuración a modo de ejemplo.

Se entiende que la documentación elaborada, las guías de implementación e implantación son detalladas y de gran utilidad para la comprensión y puesta en producción del producto.

Se destaca la excelente relación con los clientes y la prestancia de los mismos para brindar acceso a las instalaciones del hospital, pudiendose así instalar los prototipos desarrollados

en primera instancia para descartar posibles dificultades en el desarrollo y puesta en marcha.

Es importante mencionar también que en el correr del proyecto surgió el contratiempo del abandono de un integrante del equipo, el cual comenzó con dificultades para colaborar en el mismo en el mes de agosto y terminó abandonándolo en el mes de noviembre, lo que afecto las estimaciones que se habían realizado y derivando en primera instancia en la reducción del alcance del Sistema de Respaldos al estudio del Estado del arte del mismo y la elección de la herramienta que más se adecuaba al entorno de trabajo del Hospital de Clínicas, dejando de lado la puesta en producción del mismo, cosa que igualmente se realizó como ya fue mencionado.

Habiendo finalizado el proyecto y por lo antes mencionado se considera que tanto la solución implementada para el Sistema de Registros como el programa escogido para el Sistema de Respaldos cumplen con los objetivos planteados al inicio del mismo contribuyendo así con la actualización y fortalecimientos de las características de seguridad informática que posee el Hospital de Clínicas.

13 Trabajos a futuro

Como se mencionó en varias instancias, la solución propuesta tanto para el Sistema de Respaldos, como para el Sistema de Registros, pretende ser una herramienta flexible que permita ser utilizada en los más diversos ambientes, no obstante, todavía queda mucho camino por recorrer, en este capítulo, se detallan algunos puntos que se consideran importantes al momento de pensar en un trabajo futuro que extienda la solución.

Sistema de Respaldo

El estudio del estado del arte de las herramientas de respaldo nos permitió ahondar en las características y funcionalidades que las mismas proporcionan teóricamente pero para comprobar realmente sus posibilidades es importante testarlas.

Por esta razón, si bien realizamos una prueba de respaldo y restauración de ficheros en el servidor Debian Lenny donde está instalado Bacula y un sistema windows, consideramos importante poder poner a prueba la capacidad de respaldar bases de datos para comprobar que las características que se describen son reales.

Además es de utilidad hacer comparaciones y pruebas de rendimiento entre los sistemas investigados, porque, a pesar de que las características técnicas del sistema escogimos se ajustan mejor a los requerimientos de los clientes, hay que comprobar en la práctica que esto se verifica.

Sistema de Registros

En cuanto al sistema se consideran importantes los puntos que se describen a continuación para ser tenidos en cuenta como mejoras a futuro:

- **Disponibilidad del Web-Service**

  Debido a que la cantidad de eventos que se registren en el web-service puede ser potencialmente muy grande, se corre el riesgo de que el mismo se sature y pierda información, lo que no nos podemos dar el lujo de permitir, para evitar eso implementamos una cola que va almacenando los eventos registrados, para después procesarlos, pero esto puede no ser suficiente, por lo que es conveniente estudiar otros mecanismos para asegurar que no se pierdan registros.

- **Controlar que los sistemas están registrando eventos.**

Otra posibilidad que puede ofrecer el sistema es brindar la facultad de que se permita "inscribir" aplicaciones al sistema de registros, de manera que este sepa cuáles deben registrar eventos y así chequear que estas lo estén haciendo realmente o informar a los administradores en caso de que esto no suceda.

- **Informar cuando se registran varios eventos con cierta característica.**

En ocasiones es importante saber cuando en un sistema se registran muchos eventos que cumplen con determinadas particularidades, fallas mientras se inicia la aplicación, logueos incorrectos, información borrada, etc.
Una posibilidad para agregar esta funcionalidad es permitir registrar las condiciones que caracterizan estos eventos y chequear los registros ingresados contra esa información, computando la cantidad de veces que se produce una coincidencia.

- **Implementar medidas de seguridad en el servidor registro de eventos.**

Si bien se implementaron medidas de seguridad en el punto de acceso al Sistema de Registro de Eventos, el servidor en el cual se aloja el mismo, es propenso a ataques que pueden alterar la información almacenada en la base de datos.
Es recomendable implementar los mecanismos y políticas necesarias para evitar estos ataques.

- **Utilizar SSL/TSL en la comunicación con el servidor Syslog.**

La solución planteada define una serie de medidas a tomar en el servidor Syslog para evitar ataques de diversos tipos, pero no se implementó ningún mecanismo para garantizar la autenticación, confidencialidad e integridad en el envío de mensajes desde el Sistema de Registro de Eventos hacia el servidor Syslog.
Cabe mencionar que pocas implementaciones de Syslog implementan SSL, una de ellas es Syslog-ng y WinSyslog (ver Capitulo 4).

- **Crear bibliotecas y extensiones**

Estas extensiones podrían permitan a los desarrollados de otros lenguajes distintos a java (joomla es uno de ellos y está escrito en PHP), comunicarse con el servicio web con seguridad habilitada que provee el sistema de registro de eventos.

- **Integración con el sistema DICOM**

Esto implica relevar y posteriormente cargar en el Sistema de Registro de Eventos aquellos códigos de sistema, tipos de evento y roles definidos por DICOM para poder validar mensajes de auditoría provenientes de dicho sistema.

14 Referencias

[1] Windows VSS (Volume Shadow Copy Service) - (Octubre 2009)
http://en.wikipedia.org/wiki/Shadow_Copy

[2] RFC 3881 - Estructura del XML para mensajes de auditoría. - (Octubre 2009)
http://www.ietf.org/rfc/rfc3881.txt

[3] DICOM (Setiembre 2009)
http://www.rsna.org/Technology/DICOM/index.cfm

[4] Integrating the Healthcare (IHE) - (Setiembre 2009)
http://www.ihe.net/

[5] Audit Trial And Node Authentication (ATNA) - (Agosto 2010)
http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_Rev7-0_Vol2a_FT_2010-08-10.pdf

[6] WS-I Basic Profile 1.0 (Enero 2010)
http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[7] Unified Modeling Language (UML) (Julio 2009)
http://www.uml.org/

[8] Health Level Seven (HL7) - (Setiembre 2009)
http://www.hl7spain.org/.

[9] HL7 Reference Information Model (RIM) - (Setiembre 2009)
http://www.hl7.org/Library/data-model/RIM/modelpage_mem.htm

[10] WS-Security (Enero 2010)
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[11] Artículo describe en detalle la Arquitectura de Seguridad para Web Services propuesta por Microsoft e IBM. - (Enero 2010)
http://www.ibm.com/developerworks/library/specification/ws-secmap/

[12] Servicios Web (Web Services) - (Diciembre 2009)
www.w3.org/2002/ws/

[13] Syslog (Agosto 2009)
http://www.Syslog.org/

[14] RFC 3164 - Protocolo Syslog - Primera Versión (Agosto 2009)
http://www.ietf.org/rfc/rfc3164.txt

[15] RFC 5424 - Protocolo Syslog - Segunda Versión - Actual (Agosto 2009)
http://www.ietf.org/rfc/rfc5424.txt

[16] RFC 5426 - Transmisión de Syslog sobre UDP (Noviembre 2009)
http://www.ietf.org/rfc/rfc5426.txt

[17] RFC 3195 - Envío fiable para Syslog (Enero 2010)
http://www.ietf.org/rfc/rfc3195.txt

[18] Understanding PKI: Concepts, Standards and Deployment Considerations. Carlisle Adams, Steve Lloyd 2002. (Diciembre 2009)

[19] Service Oriented Architecture (SOA) (Noviembre 2009)
http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[20] Arquitectura de los servicios webs (Noviembre 2009)
http://www.w3.org/TR/ws-arch/

[21] XML (Noviembre 2009)
http://www.w3.org/XML/

[22] XML Schema (Noviembre 2009)
http://www.w3.org/TR/xmlschema-2/

[23] Enterprise User Authentication (EUA) (Julio 2009)
http://wiki.ihe.net/index.PHP?title=Enterprise_User_Authentication

[24] Cross-Enterprise User Assertion (XUA) (Julio 2009)
http://wiki.ihe.net/index.PHP?title=Cross-Enterprise_User_Assertion_%28XUA%29

[25] Syslog Watcher (Febrero 2010)
http://www.snmpsoft.com/Syslogwatcher/Syslog-server.html

[26] WinSyslog (Febrero 2010)
http://www.winSyslog.com/en/

[27] Syslog-ng logging system (Febrero 2010)
http://www.balabit.com/network-security/Syslog-ng/

[28] Kiwi Syslog (Febrero 2010)
http://www.kiwiSyslog.com/

[29] Open Healthcare Framework Project (OHF) - (Enero 2010)
http://www.eclipse.org/ohf/index.PHP

[30] Registro de Eventos de Windows (Marzo 2010)
http://www.microsoft.com/spain/technet/recursos/articulos/secmod53.mspx

[31] Widows Event Logging API (Marzo 2010)
http://msdn.microsoft.com/en-us/library/aa363652%28v=VS.85%29.aspx

[32] Oasis - Web Services Security UsernameToken Profile (Noviembre 2009)
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

[33] Glassfish V2.1 vs JBoss 5.1.0 AS (Setiembre 2009)
http://www.arabteam2000-forum.com/index.php?app=core&module=attach&section=attach&attach_id=96239

[34] DICOM - Supplement 95: Audit Trail Messages (Setiembre 2009)
ftp://medical.nema.org/medical/dicom/supps/sup95_fz.pdf

[35] RFC 5246 - The Transport Security Layer Protocol (Octubre 2009)
http://tools.ietf.org/html/rfc5246

[36] The Java EE 5 tutorial (Octubre 2009)
http://download.oracle.com/docs/cd/E17477_01/javaee/5/tutorial/doc/

[37] Sun Glassfish Enterprise Server V2.1 Performance Tuning Guide (Octubre 2009)
http://docs.sun.com/app/docs/doc/820-4343

[38] Sun Glassfish Enterprise Server V2.1 (Octubre 2009)
https://glassfish.dev.java.net/

[39] Sun Glassfish Enterprise Server V2.1 Administration Guide (Octubre 2009)
http://docs.sun.com/app/docs/doc/820-4335

[40] Sun Glassfish Enterprise Server V2.1 High Availability Admin. Guide (Octubre 2009)
http://docs.sun.com/app/docs/doc/820-4341

[41] The Web Service Interoperability Technologies Tutorial (Noviembre 2009)
http://java.sun.com/webservices/reference/tutorials/wsit/doc/index.html

[42] OASIS X.509 Certificate Token Profile (Diciembre 2009)
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf

[43] PostgreSQL (Setiembre 2009)
http://www.postgresql.org/

[44] Facelets
https://facelets.dev.java.net/

[45] Hibernate
http://www.hibernate.org/

[46] ICEFaces
http://www.icefaces.org/

[47] TortoiseSVN (Julio 2009)
http://toroisesvn.net/

[48] HTML
http://www.w3.org//html/

[49] XHTML
http://www.w3.org/TR/xhtml1/

[50] Apache
http://www.apache.org/

[51] Debian
http://www.debian.org/ Abril 2010

[52] Certificado X509 (Diciembre 2009)
http://www.uv.es/sto/cursos/seguridad.java/html/sjava.html#toc3

[53] HTTPS
http://tools.ietf.org/html/rfc2818

[55] Metro Proyect
https://metro.dev.java.net/

[56] JDom Proyect
http://www.jdom.org/

[57] Java Excel API
http://jexcelapi.sourceforge.net/

[58] iText PDF Project
http://www.itextpdf.com/

[59] Java Mail API
http://java.sun.com/products/javamail/

[60] Web Service Interoperability Technologies (WSIT)
https://wsit.dev.java.net/

[61] Syslog.org - Establishing a Hardened Syslog Server (Febrero 2010)
http://www.Syslog.org/logged/establishing-a-hardened-Syslog-log-server/

[62] SSH Syslog (Febrero 2010)
http://www.deer-run.com/~hal/sysadmin/SSH-SyslogNG.html

[63] Chroot (Febrero 2010)
http://www.sun.com/bigadmin/content/developer/howtos/chrooted.jsp

[64] Manual de seguridad Red Hat Enterprise Linux 4 (Febrero 2010)
http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/security-guide/s1-wstation-privileges.html

[65] Unix Backup and Recovery - W. Curtis Preston (Enero 2010)
http://books.google.com/books?id=_i1sO47qNnMC&lpg=PA373&ots=kjdHKjGo5C&dq=Unix%20Backup%20and%20Recovery%20online&pg=PP1#v=onepage&q=Unix%20Backup%20and%20Recovery%20online&f=false

[66] Enterprise Systems Backup and Recovery: A Corporate Insurance Policy - Preston de Guise - Capítulo 3 (Enero 2010)
http://books.google.com/books?id=2OtqvySBTu4C&lpg=PP1&dq=Backup%20%26%20Recovery&hl=es&pg=PA37#v=onepage&q=Full%20Level&f=false

[67] Protección continúa de datos (Enero 2010)
http://www.articulosinformativos.com.mx/Proteccion_continua_de_datos-a936314.html

[68] Respaldo Sintético (Enero 2010)
http://www.msexchange.org/articles_tutorials/exchange-server-2007/high-availability-recovery/synthetic-backup-data-protection.html

[69] Niveles de RAID (Enero 2010)
http://bytepile.com/raid_class.php

[70] Software Libre (Enero 2010)
http://www.gnu.org/philosophy/free-sw.es.html

[71] Areca: (Enero 2010)
http://www.areca-backup.org/documentation.php

[72] BackUp PC (Enero 2010)
http://backuppc.sourceforge.net

[73] AMANDA (Enero 2010)
http://amanda.zmanda.com/

[74] Bacula: (Enero 2010)
http://www.bacula.org/es/?page=documentation

[75] Legato (Enero 2010)
http://www.emc.com/products/detail/software/networker.htm http://dlc.sun.com/pdf/875-3427-10/875-3427-10.pdf

[76] Acronis (Enero 2010)
http://www.acronis.com/enterprise/

[77] Configuración del Módulo Director de Bacula. (Febrero 2010)
http://www.bacula.org/en/dev-manual/main/main/Configuring_Director.html

[78] Configuración del Módulo Almacenamiento de Bacula. (Febrero 2010)
http://www.bacula.org/en/dev- manual/main/main/Storage_Daemon_Configuratio.html

[79] Configuración del Módulo Cliente de Bacula. (Febrero 2010)
http://www.bacula.org/en/dev-manual/main/main/Client_File_daemon_Configur.html

[80] Quartz Scheduler
http://www.quartz-scheduler.org/

[81] RFC 5425 Transmisión de mensajes Syslog sobre TLS (Febrero 2010)
http://tools.ietf.org/search/rfc5425/

[82] American College of Radiology
http://www.acr.org/

Apéndice A: Investigación

15 Apéndice A: Investigación

En este apéndice, se presentan los documentos de investigación de tecnologías y estándares.

[Anexo 1] Documento donde se analizan 2 sistemas de gestión de base de datos y 2 servidores de aplicaciones Java para escoger el más adecuado para el proyecto.
Decisiones Tomadas.pdf.

[Anexo 2] Estudio de los Mecanismos de Seguridad que brinda Java EE.
Seguridad Java EE - ICEFaces.pdf

[Anexo 3] Resumen de la estructura mensaje ATNA y significado de sus atributos.
Mensaje de Auditoría ATNA.pdf

[Anexo 4] Análisis de la estructura y significado de 2 mensajes ATNA
Ejemplo de Mensajes de Auditoría ATNA.pdf

Apéndice B: Implementación

16 Apéndice B: Implementación

En este apéndice, se presentan los documentos de implementación.

[Anexo 5] Define los estándares, convenciones y nomenclatura utilizados a la hora de escribir y documentar el código de cada sistema desarrollado.
Estándar de Implementación.pdf

[Anexo 6] Describe las convenciones y nomenclaturas utilizadas a la hora de diseñar cada uno de los componentes del sistema.
Estándar de Documentación Técnica.pdf

[Anexo 7] Describe los pasos necesarios para construir servicios web utilizando la herramienta Netbeans 6.7.
Desarrollo de Servicios Web en Netbeans.pdf

[Anexo 8] Guía para los desarrolladores encargado de implementar la conexión entre un sistema del Hospital de Clínicas y el Sistema de Registro de Eventos mediante el servicio web disponible.
Generación Clientes Servicio Web.pdf

[Anexo 9] Describe la funcionalidad que ofrece la herramienta de generación de mensajes de auditoría ATNA, brindando ejemplos de cómo utilizarla y de los mensajes de auditoría generados en cada uno de ellos.
Biblioteca mensajeATNA.pdf

Apéndice C: Implantación

17 Apéndice C: Implantación

En este apéndice, se presentan los documentos que los usuarios pueden utilizar para comprender y utilizar el sistema implementado así como especificaciones de configuración.

[Anexo 10] Guía de Instalación de JAVA y GLASSFISH
Instalación de Java y Glassfish.pdf

[Anexo 11] Guía de Instalación PostrgreSQL y configuración Base de Datos
Instalación PostrgreSQL y configuración de Base de Datos.pdf

[Anexo 12] Guía de Configuración para activar la autenticación de usuario mediante el servidor.
Configuración Glassfish.pdf

[Anexo 13] Guía de Instalación de TinyCA, crear una autoridad certificadora y crear los certificados que utilizamos en la aplicación
Creación de Autoridad Certificadora.pdf

[Anexo14] Manual de Usuario del Sistema de Auditoría.
Manual Usuario Sistema de Auditoría.pdf.

[Anexo15] Estado del Arte Sistema de Respaldos
Estado del Arte Sistemas de Respaldo.pdf

[Anexo16] Manual de Usuario del Sistema de Respaldos “Bacula”.
Manual de Usuario del Sistema de Respaldos Bacula.pdf

Apéndice D: Caso de Prueba

18 Apéndice D: Caso de Prueba

En este apéndice, se presentan los documentos que los usuarios pueden utilizar para comprender y utilizar el sistema implementado.

[Anexo 17] Códigos definidos para identificar Grupos de Eventos, Eventos y Roles en el caso de prueba.
Códigos definidos.pdf

[Anexo 18] Datos de los eventos generados, el código que los genera e imágenes de la estructura de los mensajes generados.
Datos.pdf

Apéndice E: Glosario

19 Apéndice E: Glosario

A continuación se presentan las abreviaciones, siglas y términos utilizados en este proyecto

API	Application Programming Interface	Conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
ATNA	Audit Trail and Node Authentication	Establece medidas de seguridad, las cuales proveen la confidencialidad e integridad de la información de los pacientes. A su vez contribuye con el control de acceso en una red limitando el acceso entre nodos de la misma, y restringiendo el acceso a dichos nodos solo a usuarios autorizados.
CDA	Clinical Document Architecture	Estándar que pertenece al protocolo HL7 y su finalidad es especificar la estructura y la semántica de los documentos clínicos electrónicos con el propósito de intercambiarlos. Un documento CDA en Uruguay contiene un cabezal que incluye un autor u organización origen, un custodia destino, un paciente con institución asociada y un tipo de documento. También incluye un cuerpo que puede contener texto, imágenes, sonidos u otros contenidos multimedia.
DICOM	Digital Imaging and Communication in Medicine	Estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas. Incluye la definición de un formato de archivo y de un protocolo de comunicación de red. El protocolo de comunicación es un protocolo de aplicación que usa TCP/IP para la comunicación entre sistemas.

EAR	Enterprise Archive	Es un archivo JAR que contiene componentes de aplicaciones Java EE. Un archivo .EAR puede ser desplegado en un servidor de aplicaciones Java EE.
EJB	Enterprise JavaBean	Es una especificación que describe una plataforma basada en Java para construir aplicaciones empresariales. El objetivo es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.
GlassFish		Servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito y de código libre.
HL7	Health Level Seven	Organización internacional encargada de desarrollar estándares del cuidado de la salud, más específicamente se enfoca al dominio de datos clínicos y administrativos. Particularmente en Uruguay existe el grupo HL7 Uruguay que trata de adaptar estos estándares a la realidad uruguaya. HL7 también se le denomina al protocolo de capa de aplicación que contiene la definición de datos a ser intercambiados, el itinerario de intercambio y la comunicación de ciertos errores.
IHE	Integrating the Healthcare Enterprise	Iniciativa que intenta mejorar la forma en que los sistemas de computación comparten información en ámbito hospitalario. Promueve el uso de estándares reconocidos, como es el caso de DICOM y HL7 para mantener un cuidad optimo del paciente, a través del intercambio de información de forma sencilla, ágil y efectiva.

JAR		Es un formato desarrollado por Sun que permite empaquetar diferentes archivos asociados a una aplicación desarrollada en Java.
JAX-WS		API para la creación de servicios web y clientes que se comunican usando XML.
JMS	Java Message Service	API que brinda un estándar de mensajería permitiendo a los componentes de aplicaciones basados en la plataforma de Java2EE crear, enviar, recibir y leer mensajes.
Metro		Plataforma para el desarrollo y despliegue de servicios web que forma parte del proyecto Glassfish.
OID	Object Identifier	Son cadenas de números separados por comas ordenados de forma jerárquica. Son usados en diferentes protocolos como un estándar para identificar entidades o atributos. Existen varias raíces que se le agregan OID sucesivamente para identificar países, instituciones, características, etc.
PHI	Protected Health Information	Sigla que hace referencia a la información protegida en el área de la salud.
SOA	Service Oriented Architecture	Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos del usuario. Permite la creación y/o cambios de los procesos de negocio desde la perspectiva de TI de forma ágil, a través de la composición de nuevos procesos utilizando las funcionalidades de negocio que están contenidas en la infraestructura de aplicaciones actuales o futuras (expuestas bajo la forma de servicios web).
SOAP	Single Object Access	Protocolo estándar para intercambiar mensajes Protocol basados en XML a través de redes de computadoras, normalmente utilizando HTTP. Es uno de los protocolos utilizados para implementar servicios web.
SYSLOG		Syslog es un estándar de facto para el envío de mensajes de registro en una red informática IP. Por

		Syslog se conoce tanto al protocolo de red como a la aplicación o biblioteca que envía los mensajes de registro. Un mensaje de registro suele tener información sobre la seguridad del sistema, aunque puede contener cualquier información. Junto con cada mensaje se incluye la fecha y hora del envío.
UDDI	Universal Description Discovery and Integration	Es un directorio de servicios web distribuido y basado en Web que permite que se listen, busquen, y descubran este tipo de componentes de software.
WAR	Web Archive	Especificación desarrollada por Sun que permite agrupar un conjunto de clases y documentos que conforman una aplicación web en Java.
WSDL	Web Service Description Language	Lenguaje de la interfaz pública para los servicios web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de un esquema XML. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.
WEB SERVICE		Un servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. La interoperabilidad es uno de sus puntos fuertes, ya que aplicaciones construidas con distintos lenguajes de programación y corriendo en plataformas distintas pueden intercambiar datos mediante web-services. Esto se debe a la adopción de estándares abiertos (XML, SOAP, WSDL).

XML	Extensible Markup Language	Metalenguaje extensible que permite definir la gramática de lenguajes específicos. Se aplica en internet, y además se propone como un estándar para el intercambio de información estructurada en diferentes plataformas.

Apéndice F: Versiones de Herramientas

20 Apéndice F: Versiones de Herramientas

Hibernate 3.2.6
http://www.hibernate.org/

Glassfish v2.1
https://glassfish.dev.java.net/

Netbeans 6.7
http://netbeans.org/

PostgreSQL 8.4
http://www.postgresql.org/

Java JDK 6 update 18
http://java.sun.com/javaee/

Genexus 9.0 update 6
http://www.genexus.com/portal/

Apache Tomcat 6.0.25
http://tomcat.apache.org/