

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE INGENIERÍA  
DIVISIÓN DE INGENIERÍA ELÉCTRICA

## SISTEMA DE CONTROL DOCUMENTAL DISTRIBUÍDO

### TESIS

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN

PRESENTA:

**MIGUEL ENRIQUE COBÁ MARTÍNEZ**

DIRECTORA DE TESIS:

**M. EN C. MARÍA CRISTINA CASIMIRO GARCÍA**



CO DIRECTORA:

**DRA. ANA MARÍA VÁZQUEZ VARGAS**

MÉXICO, D. F.

2005



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# El Sistema de Control Documental Distribuído

**Miguel Enrique Cobá Martínez**

[<miguel.coba@gmail.com>](mailto:miguel.coba@gmail.com)

Copyright © 2005 Miguel Enrique Cobá Martínez

---

## **Dedicatoria**

Con todo mi amor a la mujer más valiente y perseverante que he conocido: Ana María.

## Prólogo

Esta tesis es el resultado de un proyecto conjunto entre la Secretaría de Salud de México y Strategic Analysis de México S.A.

El proyecto es interesante en sí mismo ya que aborda y resuelve un variado conjunto de problemas prácticos que surgen durante la puesta en operación de un proyecto de software de gran tamaño. Esta es la motivación principal al elegir este proyecto como tema de tesis. En la presentación de esta tesis se muestran varias técnicas, metodologías y soluciones prácticas a problemas que, aunque muy comunes en el mundo laboral, es difícil encontrarlos en las aulas de la Universidad. Debo decir además, que es precisamente la Universidad la que te prepara para enfrentar y resolver problemas nuevos cada día. Es gracias a ella y a sus profesores, que los alumnos podemos encarar desafíos como este y vencerlos de manera satisfactoria.

En estas páginas se encuentran ejemplos de problemas interesantes, así como las soluciones que se les dieron, muchas veces cambiando la solución óptima teórica por la que en la práctica resultó ser más conveniente. La presentación del tema de esta tesis a los alumnos que aún están en la fase de aprendizaje, les brindará mayores perspectivas relativas a lo que significa trabajar en el área de Ingeniería en Computación.

Quiero agradecer el apoyo brindado para llegar a esta etapa a varias personas. Ha sido un camino arduo y largo pero fructífero. El haber llegado a esta etapa y haber aprendido tanto, pero nunca lo suficiente, se debe a gran cantidad de personas que se encontraron a mi alrededor en distintas fases de esta travesía.

Quiero agradecer, en primer lugar, a mi madre. El esfuerzo hecho y el ejemplo de valentía y constancia que me brindó es parte fundamental en mi vida. No es fácil la tarea que te echaste en hombros, pero nunca tuviste temor y ninguna meta era inalcanzable si te empeñabas. Este trabajo es de alguna manera la culminación de tus esfuerzos y los míos. Gracias mamá, esto es para ti.

Gracias también a mi familia por los valores que nos enseñaron día a día con el ejemplo. Abuelo; abuela donde quiera que estés, no fui ingeniero civil para construir la casa, pero soy ingeniero, aunque de otro tipo; tío, gracias por tu apoyo tantas veces durante este tiempo; tía; mamá Tere, pocas personas tienen el honor de tener dos mamás, y nosotros somos afortunados de tenerte a ti, gracias por cuidarnos sin distinción; Jesús, Emmanuel no desistan; Iván, es un honor para mi llamarte hermano.

Aline, gracias por abrir un mundo nuevo completamente desconocido para mí. Gracias por caminar conmigo y por mirar conmigo hacia adelante.

Mis amigos Pedro, Salomón, Ishtar, Talina, Álvaro gracias por todo lo que me enseñaron y por todo lo que conviví con ustedes. Crecimos juntos y espero que sigamos creciendo juntos.

A Margarita Santoyo y Manuel Bastera, por todo el apoyo en los primeros años y por abrirnos su casa y su biblioteca.

Además, en el ámbito profesional y durante la carrera, conocí a muchas personas que me enseñaron valiosas lecciones y que me permitieron crecer como persona.

A Cristina Casimiro y Gerardo Caballero por la libertad y la confianza otorgada para experimentar e intentar nuevas soluciones y sobre todo por el respaldo, al inicio con reservas, y después con seguridad, para introducir Software Libre como alternativa sólida, confiable y de calidad a una amplia gama de proyectos surgidos durante mi estancia en Strategic Analysis de México. El Sistema de Control Documental Distribuido es uno de tales proyectos. No tengo manera de agradecer el apoyo y la libertad de experimentar.

Además, quiero agradecer el apoyo brindado por la Secretaría de Salud de México, principalmente al Ing. José Francisco Becerril por la disposición y amabilidad al permitir el uso del SCDD como tema de tesis.

A la UNAM y sus maestros, por haberme formado y permitirme ser parte de privilegiado grupo de mexicanos que tienen acceso a sus aulas.

Finalmente, gracias a Dios por todo lo obtenido hasta ahora.

## **Tabla de contenido**

### Prólogo

#### 1. Introducción

1. Antecedentes
2. Requerimientos estructurales
3. La solución planteada
4. Organización de esta obra

#### 2. El Sistema de Control Documental

1. Objetivo
2. Arquitectura
3. Uso del SCD
  - 3.1. Uso cotidiano del SCD
  - 3.2. Beneficios del SCD
4. Nuevos requerimientos

#### 3. El Sistema de Control Documental Distribuido

1. Requerimientos
2. Arquitectura
3. Beneficios del SCDD

#### 4. Diseño y Desarrollo del SCDD

1. Servidores
  - 1.1. Fedora Core 1
  - 1.2. PostgreSQL
  - 1.3. Apache Tomcat
2. Lenguajes de Programación
3. Estándares internacionales
  - 3.1. XML
  - 3.2. XML Schema
4. Algoritmos de replicación
  - 4.1. Envío de información entre un Nodo Local y otro Nodo del SCDD
  - 4.2. Envío de información a un Nodo del SCDD
  - 4.3. Recepción de información de un Nodo del SCDD
5. Migración del SCD al SCDD

#### 5. Operación del SCDD

1. Uso del SCDD
2. Escenarios para el SCDD
  - 2.1. Operación normal
  - 2.2. Fallo en la infraestructura de red
  - 2.3. Fallo en el Nodo Central
  - 2.4. Fallo en el Nodo Local
3. Pruebas de rendimiento
  - 3.1. Tipos de pruebas de rendimiento
    - 3.1.1. Pruebas en simulador
    - 3.1.2. Pruebas en entornos reales
    - 3.1.3. Pruebas híbridas
  - 3.2. Metodología a seguir
  - 3.3. Herramientas existentes
4. Resultados
5. Análisis de resultados

#### 6. Recomendaciones

[1. Compresión de datos](#)

[2. Cifrado de datos](#)

[3. Autorización y autenticación de partes](#)

[7. Conclusiones](#)

[A. Xen y paravirtualización](#)

[B. Código fuente](#)

[C. Generación de documentos de prueba](#)

[glosario](#)

[Bibliografía](#)



# Prólogo

Esta tesis es el resultado de un proyecto conjunto entre la Secretaría de Salud de México y Strategic Analysis de México S.A.

El proyecto es interesante en sí mismo ya que aborda y resuelve un variado conjunto de problemas prácticos que surgen durante la puesta en operación de un proyecto de software de gran tamaño. Esta es la motivación principal al elegir este proyecto como tema de tesis. En la presentación de esta tesis se muestran varias técnicas, metodologías y soluciones prácticas a problemas que, aunque muy comunes en el mundo laboral, es difícil encontrarlos en las aulas de la Universidad. Debo decir además, que es precisamente la Universidad la que te prepara para enfrentar y resolver problemas nuevos cada día. Es gracias a ella y a sus profesores, que los alumnos podemos encarar desafíos como este y vencerlos de manera satisfactoria.

En estas páginas se encuentran ejemplos de problemas interesantes, así como las soluciones que se les dieron, muchas veces cambiando la solución óptima teórica por la que en la práctica resultó ser más conveniente. La presentación del tema de esta tesis a los alumnos que aún están en la fase de aprendizaje, les brindará mayores perspectivas relativas a lo que significa trabajar en el área de Ingeniería en Computación.

Quiero agradecer el apoyo brindado para llegar a esta etapa a varias personas. Ha sido un camino arduo y largo pero fructífero. El haber llegado a esta etapa y haber aprendido tanto, pero nunca lo suficiente, se debe a gran cantidad de personas que se encontraron a mi alrededor en distintas fases de esta travesía.

Quiero agradecer, en primer lugar, a mi madre. El esfuerzo hecho y el ejemplo de valentía y constancia que me brindó es parte fundamental en mi vida. No es fácil la tarea que te echaste en hombros, pero nunca tuviste temor y ninguna meta era inalcanzable si te empeñabas. Este trabajo es de alguna manera la culminación de tus esfuerzos y los míos. Gracias mamá, esto es para ti.

Gracias también a mi familia por los valores que nos enseñaron día a día con el ejemplo. Abuelo; abuela donde quiera que estés, no fui ingeniero civil para construir la casa, pero soy ingeniero, aunque de otro tipo; tío, gracias por tu apoyo tantas veces durante este tiempo; tía; mamá Tere, pocas personas tienen el honor de tener dos mamás, y nosotros somos afortunados de tenerte a ti, gracias por cuidarnos sin distinción; Jesús, Emmanuel no desistan; Iván, es un honor para mi llamarte hermano.

Aline, gracias por abrir un mundo nuevo completamente desconocido para mí. Gracias por caminar conmigo y por mirar conmigo hacia adelante.

Mis amigos Pedro, Salomón, Ishtar, Talina, Álvaro gracias por todo lo que me enseñaron y por todo lo que conviví con ustedes. Crecimos juntos y espero que sigamos creciendo juntos.

A Margarita Santoyo y Manuel Basterra, por todo el apoyo en los primeros años y por abrirnos su casa y su biblioteca.

Además, en el ámbito profesional y durante la carrera, conocí a muchas personas que me enseñaron valiosas lecciones y que me permitieron crecer como persona.

A Cristina Casimiro y Gerardo Caballero por la libertad y la confianza otorgada para experimentar e intentar nuevas soluciones y sobre todo por el respaldo, al inicio con reservas, y después con seguridad, para introducir Software Libre como alternativa sólida, confiable y de calidad a una amplia gama de proyectos surgidos durante mi estancia en Strategic Analysis de México. El Sistema de Control Documental Distribuido es uno de tales proyectos. No tengo manera de agradecer el apoyo y la libertad de experimentar.

Además, quiero agradecer el apoyo brindado por la Secretaría de Salud de México, principalmente al Ing. José Francisco Becerril por la disposición y amabilidad al permitir el uso del SCDD como tema de tesis.

A la UNAM y sus maestros, por haberme formado y permitirme ser parte de privilegiado grupo de mexicanos que tienen acceso a sus aulas.

Finalmente, gracias a Dios por todo lo obtenido hasta ahora.

# Capítulo 1. Introducción

## 1. Antecedentes

En la Secretaría de Salud del Gobierno de México se encuentra en uso el Sistema de Control Documental (SCD) desde hace más de 2 años. Dicho sistema tiene como objetivo administrar el gran número de documentos generados y recibidos diariamente por las diferentes dependencias de la Secretaría de Salud (SSA).

El SCD es un sistema que tiene contacto directo con los usuarios y que es usado como herramienta clave para el desarrollo de las actividades diarias de la SSA. Por tanto, y teniendo siempre en mente que una herramienta debe ayudar y eficientar el proceso para el que fue creada, la retroalimentación que se obtiene de las personas que hacen uso de él es de vital importancia para asegurar que éste siga cumpliendo con el objetivo inicial planteado.

Con la puesta en operación y el uso cotidiano del SCD surgieron requerimientos de los usuarios solicitando que el sistema realizara funciones que no estaban contempladas originalmente en el proyecto. Esta funcionalidad los ayudaría a realizar mejor sus actividades. Por tanto, mientras que el SCD se planeó de cierta manera, el uso continuo y los comentarios de los usuarios contribuyeron a modelarlo y adaptarlo mucho más acorde a las actividades diarias de la SSA. Como ejemplo del tipo de cambios agregados al SCD cuando ya se encontraba en pleno uso se encuentran la realización de varios reportes destinados a los directores y subdirectores de las distintas áreas de la Secretaría. Éstos les permitirían obtener información estadística del flujo de documentos generados, recibidos y atendidos dentro de su respectiva área de mando. Con este tipo de reportes, cada director de área podría saber el estado y funcionamiento de su área y tomar decisiones de acuerdo con esta información.

## 2. Requerimientos estructurales

Existen otro tipo de requerimientos que surgieron a raíz de la puesta en operación del SCD. Estos son de carácter estructural y su cumplimiento y realización implicaban cambiar la arquitectura original del SCD.

El diseño original del Sistema de Control Documental se basa en la idea de un servidor central al cual acceden todos sus usuarios y donde se almacena toda la información de los documentos y respuestas a documentos existentes en una dependencia. Este modelo es exitoso en la medida en que se satisfagan una serie de requisitos necesarios para el correcto funcionamiento del mismo. Estos requisitos son:

1. Conexión permanente, a través de la infraestructura de red, al servidor del SCD.
2. Ancho de banda suficiente para soportar la tasa de transferencia de datos entre el servidor del SCD y la dependencia que hace uso de sus servicios, de manera que no afecte la capacidad de acceso a Internet para otros usos como navegación web y correo electrónico, entre otros.

La importancia de estos requisitos para el uso eficaz del SCD se hicieron patentes conforme el sistema se extendió a varias dependencias separadas entre sí

geográficamente y conectadas a través de Internet. Se vio entonces que el SCD, tal como estaba diseñado originalmente con un servidor único al que acceden todos los usuarios desde sus respectivas dependencias, dependía directa y peligrosamente de la disponibilidad y confiabilidad de la infraestructura de red. Además, debido a que cada dependencia confiaría el manejo de su documentación al SCD, la indisponibilidad del servicio debido a fallas en ella no era una opción, ya que esto significaría un grave problema al no permitir a los usuarios realizar sus actividades de manera satisfactoria.

Debía encontrarse una solución que permitiera a los usuarios continuar con sus labores cotidianas, preferentemente de manera transparente, aún en el caso en que el enlace de comunicaciones entre la dependencia y el servidor del SCD no estuviera disponible durante un lapso de tiempo considerable. Además de esto, estaba el requisito implícito de que la información no se viera alterada por la frecuencia o duración de las interrupciones en el servicio de red y que además permitiera autonomía en la operación de una dependencia.

### **3. La solución planteada**

El Sistema de Control Documental Distribuido (SCDD) es la solución planteada a estos problemas. Éste es un sistema distribuido formado por varios nodos independientes que utiliza técnicas de replicación además de estándares internacionales para la transmisión de información y que puede operar aún en el caso de que alguno o varios de sus nodos no estén disponibles en un momento dado.

En el SCDD no existe un servidor único al que todas las dependencias se conecten y sobre la cual tengan que realizar su trabajo, tal como crear documentos o responder documentos, como sucedía en el SCD.

Por el contrario, al ser una arquitectura distribuida, el SCDD puede operar de manera más eficiente pues no concentra el trabajo en un servidor único. En lugar de eso, el SCDD está formado por varios servidores autónomos e independientes entre sí, uno por cada dependencia, los cuales brindan a sus usuarios los servicios necesarios para realizar sus actividades cotidianas.

Cada uno de estos servidores, al estar dentro de las instalaciones de la dependencia, no está sujeto a los inconvenientes derivados del SCD original, tales como inaccesibilidad en caso de fallo en el enlace de red o altos tiempos de respuesta del sistema debidos a la velocidad de acceso a Internet.

### **4. Organización de esta obra**

A lo largo de esta tesis se describirá la arquitectura, diseño, desarrollo y puesta en producción del Sistema de Control Documental Distribuido.

Para ello se estudiará el SCD usado actualmente en la Secretaría de Salud. Este es el tema del capítulo 2. Esto servirá como antecedente y como base de partida para el SCDD. Además, permitirá comprender completamente la razón por la que fue necesario crear el Sistema de Control Documental Distribuido.

En el capítulo 3 se describirá detalladamente al SCDD. Se discutirán los objetivos que debe cumplir, las restricciones que debe tomar en cuenta, la funcionalidad que debe brindar, así como otros factores que influenciaron su diseño, desde la arquitectura hasta la puesta en marcha en el entorno de producción.

En el capítulo 4 se detalla el diseño y desarrollo del Sistema de Control Documental Distribuido, mostrando cómo se materializaron las ideas planteadas y discutidas en el capítulo 3.

El capítulo 5 tiene por objeto exponer al SCDD a pruebas que permitan medir con precisión el rendimiento que se puede obtener de él. Por tanto, este capítulo muestra los escenarios de prueba planteados para él, así como los resultados de estas pruebas. Además se da un análisis de estos resultados a la luz del entorno donde operará finalmente el SCDD.

El capítulo 6 detalla recomendaciones que harán del SCDD un sistema más robusto, pero que no necesariamente agregan nueva funcionalidad al mismo. Se plantean aquí también posibles maneras de llevar a cabo estas recomendaciones.

Finalmente, el capítulo 7 expresa las conclusiones derivadas de este proyecto.

Mucha de la información usada para la realización de esta obra se colocó en los distintos apéndices, de manera que no rompieran el flujo principal de ideas que forman el centro de esta tesis. Esta información es en sí misma importante y, por tanto, se adjuntan como referencia para complementar el contenido de la tesis.

El primer apéndice contiene una pequeña introducción a Xen, la tecnología de paravirtualización utilizada en el capítulo 5 para realizar las pruebas del SCDD.

El apéndice B muestra el código fuente del programa desarrollado con objeto de generar los documentos utilizados durante las pruebas del SCDD del capítulo 5.

Finalmente, el apéndice C contiene algunas mediciones realizadas durante la generación de los datos de prueba y que pueden ser interesantes para estimar la capacidad del SCD de controlar y administrar información documental.

## Capítulo 2. El Sistema de Control Documental

En este capítulo se describe el Sistema de Control Documental existente actualmente en la Secretaría de Salud de México.

### 1. Objetivo

El Sistema de Control Documental es un producto desarrollado como parte del programa de actualización y modernización en el área de tecnologías de la información llevado a cabo por la Secretaría de Salud del Gobierno Federal (SSA) en todas sus dependencias a nivel nacional. Parte fundamental de este programa es la utilización óptima de los recursos materiales existentes, así como la reducción de los costos de operación de las dependencias que conforman la SSA.

En dicha Secretaría se generan y reciben diariamente gran cantidad de documentos en la forma de circulares, memorándums, solicitudes, oficios, etc. Cada uno de estos documentos tiene asignado un período de tiempo dentro del cual se debe dar respuesta al mismo, si aplica. Éste puede ir desde horas hasta meses. Sin embargo, no todos los documentos deben responderse. Las circulares por ejemplo, solamente se reciben y, de acuerdo con la información contenida en ellas, se toman medidas, pero no se genera ninguna respuesta. Para los documentos que requieren de ella, ésta debe generarse dentro del lapso de tiempo especificado y debe hacerse llegar oportunamente al remitente del documento, de manera que se considere cumplido el objetivo inicial del documento.

*Dadas las condiciones actuales del proceso por medio del cual se distribuyen tanto documentos como respuestas, ya sea dentro de una misma dependencia o entre varias de ellas, es difícil cumplir con las restricciones de tiempo establecidas para un documento.*

Por otra parte, a partir de la entrada en vigor de la Ley de Transparencia y Acceso a la Información Pública Gubernamental, las dependencias y entidades del gobierno federal tienen la obligación de atender todas las solicitudes de información hechas por los ciudadanos. Esta respuesta debe realizarse de manera expedita, proporcionando información confiable, precisa y completa.

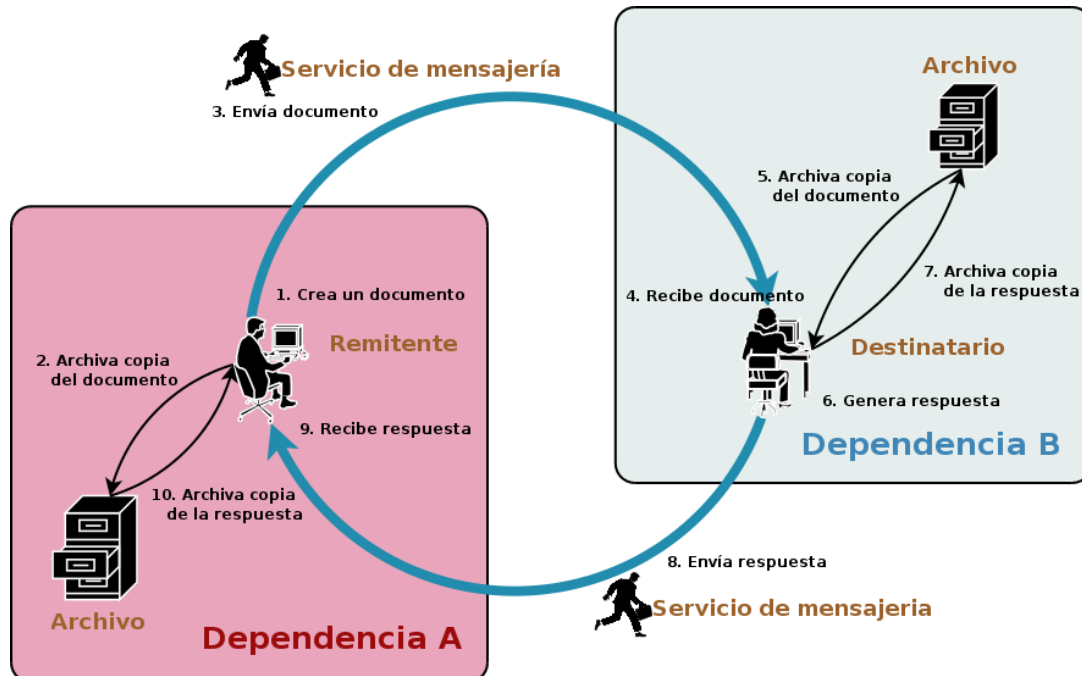
*Debido al enorme volumen de información que maneja la Secretaría, aunado al hecho de que se encuentra asentada en papel impreso, la búsqueda de la información solicitada es ardua y lenta.*

El objetivo del Sistema de Control Documental es administrar la enorme cantidad de documentos que se generan y se reciben diariamente por la SSA de manera eficiente, rápida y sencilla, proporcionando además acceso completo a la información histórica almacenada en el sistema, de manera que sea posible realizar búsquedas de información de manera detallada.

El objetivo del SCD puede entenderse mejor si se analiza el proceso tradicional por medio del cual se manejaba la información documental de la dependencia.

Un caso de uso típico del proceso de generación de documentos y respuestas en una dependencia que no cuenta con el Sistema de Control Documental se muestra en la siguiente figura:

**Figura 2.1. Dependencia sin servicios del SCD**



El proceso es como sigue:

1. Un empleado de la dependencia A, en el papel de remitente, crea un documento dirigido a un empleado de la dependencia B.
2. Una copia del documento es almacenada en el archivo de la dependencia A para futura referencia.
3. El documento se hará llegar por medio de mensajería a su destinatario en la dependencia B.

*Dentro de este punto se incluyen las actividades necesarias para que el documento pueda salir de la dependencia A con el destino correcto especificado y llegar a la oficina de recepción de documentos de la dependencia B, donde, a su vez, lo harán llegar al destinatario del documento. Este proceso puede durar desde un par de horas hasta varios días, dependiendo de la ubicación física de las dependencias, entre otros factores.*

4. El empleado de la dependencia B, en el papel de destinatario, recibe el documento.
5. Una copia del documento se almacena en el archivo de la dependencia B para futura referencia.
6. El destinatario genera una respuesta al documento enviado por el remitente.
7. Una copia de la respuesta generada se almacena en el archivo de la dependencia B para futura referencia.
8. La respuesta se hará llegar por medio de mensajería al remitente en la dependencia A.

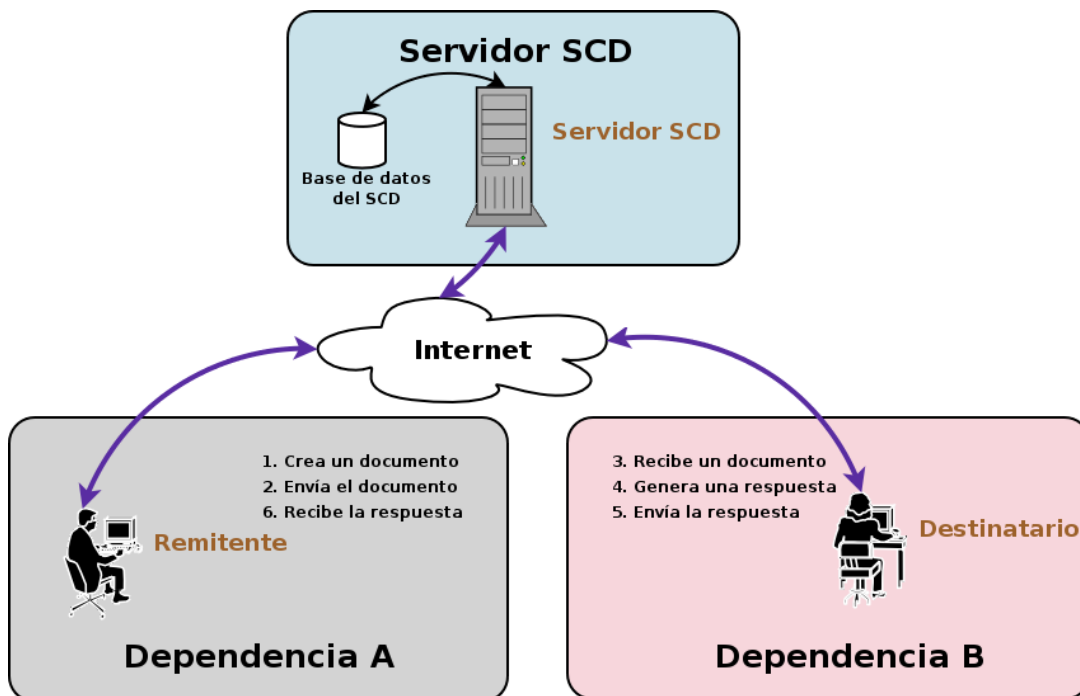
*En este punto aplican las mismas actividades mencionadas en el punto 3.*

9. El remitente recibe la respuesta generada por el destinatario.
10. Una copia de la respuesta se almacena en el archivo de la dependencia A para futura referencia.

En comparación con el proceso tradicional, el SCD proporciona a la dependencia que lo utiliza un gran número de ventajas, tanto en términos de eficiencia, como en ahorro de recursos.

Un caso de uso típico del proceso de generación de documentos y respuestas en una dependencia que sí cuenta con el Sistema de Control Documental se muestra en la siguiente figura:

**Figura 2.2. Dependencia con servicios del SCD**



El proceso es como sigue:

1. Un empleado de la dependencia A, en el papel de remitente, crea un documento dirigido a un empleado de la dependencia B.
2. El documento se hace llegar, utilizando los servicios del SCD, a su destinatario en la dependencia B.

*El SCD se encarga de enviar el documento a su destino, de manera que al momento en que el remitente envía el documento, éste aparece al destinatario. El tiempo de entrega del documento se reduce así del rango de horas en una dependencia sin SCD, a sólo unos cuantos segundos en el caso de una dependencia con SCD. Toda la información del documento es almacenada en la base de datos del SCD de manera que esté disponible para futura referencia.*



*Todo esto es hecho de manera automática y sin intervención del usuario como parte de los servicios proporcionados por el SCD.*

3. El empleado en la dependencia B, en el papel de destinatario, recibe el documento.
4. El destinatario genera una respuesta al documento enviado por el remitente.
5. La respuesta se hará llegar, utilizando los servicios del SCD, al remitente en la dependencia A.

*En este punto aplican las mismas actividades mencionadas en el punto 2.*

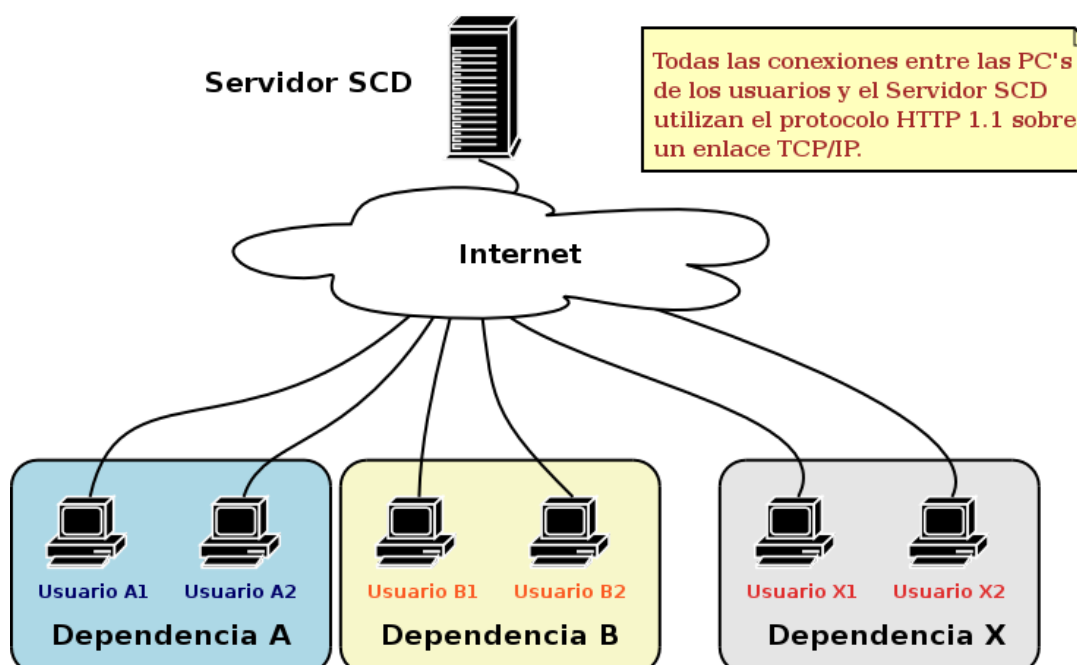
6. El remitente recibe la respuesta generada por el destinatario.

Es evidente que la introducción del SCD a una dependencia reduce considerablemente la complejidad del proceso de envío y generación de documentos y respuestas a documentos. Esto se traduce en una mejor utilización de los recursos públicos, así como en mayor eficiencia para realizar las tareas que forman el quehacer diario de la institución.

## 2. Arquitectura

La arquitectura del SCD es sencilla, pero efectiva, para cumplir con el objetivo del sistema. En la figura siguiente se muestra un diagrama de la arquitectura del sistema:

**Figura 2.3. Arquitectura del SCD**



El SCD está formado por un único servidor que brinda servicios a todos los usuarios del sistema. Este servidor único es accedido a través de la infraestructura de red y es el lugar donde se almacena toda la información del sistema. Los usuarios de él están

localizados potencialmente en varias dependencias, cada una en una ubicación geográfica distinta.

La conexión entre los usuarios y el Servidor SCD es utilizando enlaces de red, con protocolo HTTP 1.1 y transportado todo a través de enlaces TCP/IP, de manera que el Servidor SCD puede ser accedido literalmente desde cualquier región del mundo con acceso a Internet.

## **3. Uso del SCD**

### **3.1. Uso cotidiano del SCD**

A continuación se muestra un ejemplo del uso del SCD para una dependencia. El escenario planteado es el siguiente:

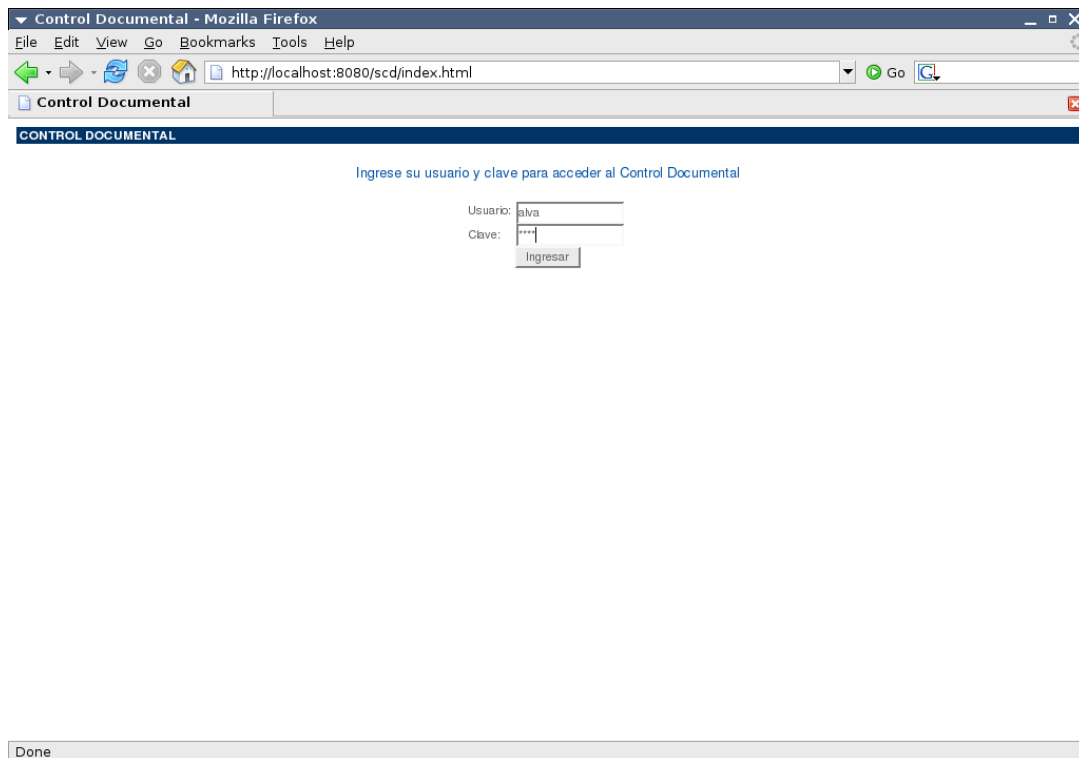
La Secretaría de Salud planea iniciar una campaña de vacunación contra la polio a nivel nacional. Para lograr esto se debe anunciar, al menos con 2 meses de anticipación, tanto en radio y televisión como en anuncios fijos, el inicio y duración de la campaña, así como las diversas maneras en que se llevará a cabo y las instrucciones para que los ciudadanos puedan vacunar a sus hijos.

En una de las dependencias de la SSA, llamada *Dependencia 01*, existen 2 usuarios, Alva y Enrique, los cuales pueden enviar y recibir documentos entre sus respectivas divisiones.

Alva, encargada del área de publicidad de la Secretaría, recibe la orden de coordinar la campaña publicitaria. Para poder llevar a cabo esta encomienda, Alva se apoya en Enrique, director del área de Informática, para que él sea el encargado de dirigir la realización de toda la publicidad impresa que se distribuirá a lo largo del país. Alva, en consecuencia, envía un memorándum a Enrique anunciándole el inicio de la campaña, solicitando la elaboración de la publicidad impresa y especificándole las características de la misma.

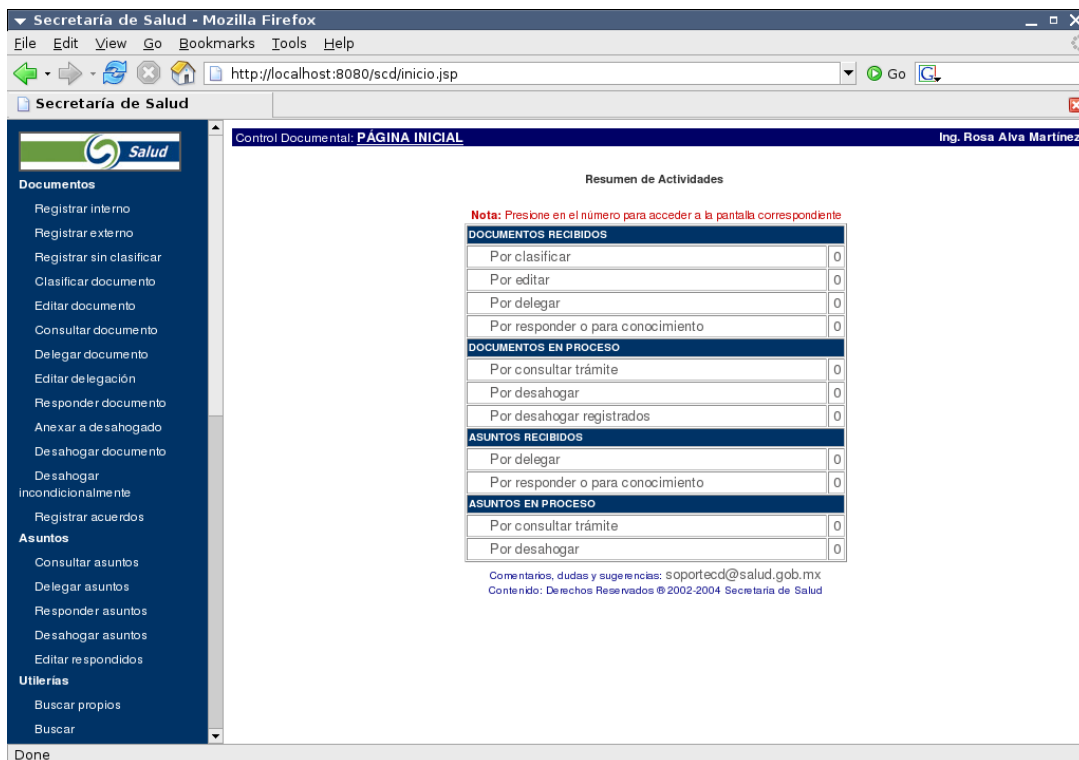
Con este fin ella inicia una sesión en el Sistema de Control Documental, ingresando con el usuario y contraseña que tiene asignados.

#### **Figura 2.4. Ingreso de Alva al SCD**



Ya dentro del Sistema de Control Documental, Alva puede notar que no tiene documentos pendientes, por lo que continúa con la creación del documento usando la opción *Registrar Interno*, dentro de la sección *Documentos* del menú.

**Figura 2.5. Pantalla inicial del SCD para Alva**



Una vez en la pantalla del registro de documentos internos, Alva captura los datos del memorándum que enviará a Enrique.

**Figura 2.6. Registro del documento**

Secretaría de Salud - Mozilla Firefox  
http://localhost:8080/scd/inicio.jsp

Secretaría de Salud

Control Documental: **REGISTRAR INTERNO** Ing. Rosa Alva Martínez

**Nota:** Los campos con asterisco(\*) son requeridos.

**Fecha de registro** (dd/mm/aaaa) 07/11/2005

**Tipo de documento\*** Memorandum

**Fecha del documento\*** (dd/mm/aaaa) 07/05/2005

**Número de oficio\*** (puede usar números, letras o símbolos, sin espacios o apóstrofes. SIN para los que no tienen un folio determinado) SSA/2005/0569

**Tiempo de vencimiento\*** (sólo números) 3 Día

**Remitente\*** Dependencia 01  
Martinez Roca Alva, Ing. -- Director

**Destinatario\*** Dependencia 01  
Torrentera Garay Jose Enrique, Ing. -- Director

**Con copia para conocimiento** Dependencia 01  
Seleccione una persona...  
Arcos Amezcua Armando, Lc. -- Director  
Martinez Roca Alva, Ing. -- Director

**Clasificación\*** Promoción de la Salud

**Subclasificación\*** Campanas de Informacion y de Promocion Para Prevencion de E

**Descripción\*** (máximo 2000 caracteres) Con motivo del inicio de la campaña de vacunación contra la polio 2006, a efectuarse del 04 de febrero de 2006 al 14 de febrero de 2006, se solicita la elaboración del material publicitario en medio impreso, el cual deberá entregarse para revisión a más tardar el día 29 de noviembre de 2005. Deberán diseñarse pósters de 1.5m por 0.8m así como folletos informativos tamaño carta.

Done

Una vez capturados los datos del documento, el SCD muestra un mensaje indicando que el documento fue registrado exitosamente. El documento será enviado a Enrique inmediatamente.

**Figura 2.7. Documento registrado y enviado al destinatario**

Secretaría de Salud - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/scd/inicio.jsp

Secretaría de Salud

Control Documental: REGISTRAR INTERNO Ing. Rosa Alva Martínez

**SU OPERACIÓN HA SIDO REGISTRADA**

Información del documento.

No. Doc.	No Oficio	Fecha del Documento (dd/mm/aaaa)	Fecha de Registro (dd/mm/aaaa hh:mm)	Fecha de Vencimiento (dd/mm/aaaa hh:mm)
47-1	SSA/2005/00394	07/11/2005	07/11/2005 02:03	09/11/2005 02:03
Registrado por		Remite		Destinatario
Ing. Rosa Alva Martínez		Ing. Rosa Alva Martínez		Ing. Jose Enrique Torrentera Garay
Status del documento		Alta		
Clasificación/Subclasificación		Promoción de la Salud /Campanas de Información y de Promocion Para Prevencion de Enfermedades		
Tipo		Memorandum		
Descripción del documento		Con motivo del inicio de la campaña contra el pólo 2006, a efectuarse del 4 al 14 de febrero de 2006, se solicita la elaboración de material publicitario en medio impreso, el cual deberá entregarse para revisión a más tardar el día 28 de noviembre de 2005. Deberán diseñarse pósters de 0.8m por 1.5m, así como folletos informativos tamaño carta.		
Palabras clave		campaña, polo, 2006		
Observaciones		Deberán seguirse los lineamientos especificados en el manual de procedimientos, en la sección Campañas de Vacunación, para la elaboración de la publicidad		
Copias para conocimiento		Ninguna		

Anexos del documento.  
No existen anexos para este documento.

Editar doc    Capturar otro Doc. interno    Regresar a Pág. Inicial

Comentarios, dudas y sugerencias: [soportecd@salud.gob.mx](mailto:soportecd@salud.gob.mx)  
Contenido: Derechos Reservados © 2002-2004 Secretaría de Salud

Enrique, por su lado, al llegar a trabajar e iniciar una sesión con el Sistema de Control Documental verá que tiene un documento pendiente de respuesta que le fue enviado por Alva.

Figura 2.8. Pantalla inicial del SCD para Enrique

Secretaría de Salud - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/scd/inicio.jsp

Secretaría de Salud

Control Documental: PAGINA INICIAL Ing. Jose Enrique Torrentera Garay

Resumen de Actividades

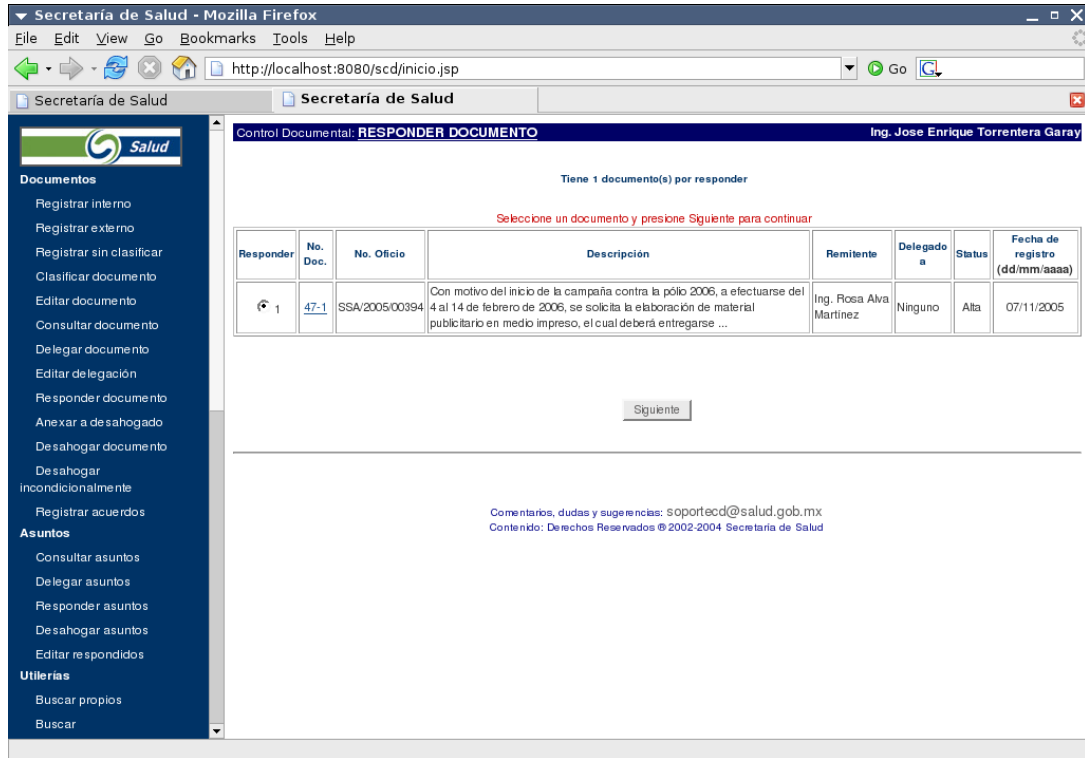
**Nota: Presione en el número para acceder a la pantalla correspondiente**

DOCUMENTOS RECIBIDOS	
Por clasificar	0
Por editar	0
Por delegar	1
<b>Por responder o para conocimiento</b>	<b>1</b>
DOCUMENTOS EN PROCESO	
Por consultar trámite	1
Por desahogar	0
Por desahogar registrados	1
ASUNTOS RECIBIDOS	
Por delegar	0
Por responder o para conocimiento	0
ASUNTOS EN PROCESO	
<b>Por consultar trámite</b>	<b>0</b>
Por desahogar	0

Comentarios, dudas y sugerencias: [soportecd@salud.gob.mx](mailto:soportecd@salud.gob.mx)  
Contenido: Derechos Reservados © 2002-2004 Secretaría de Salud

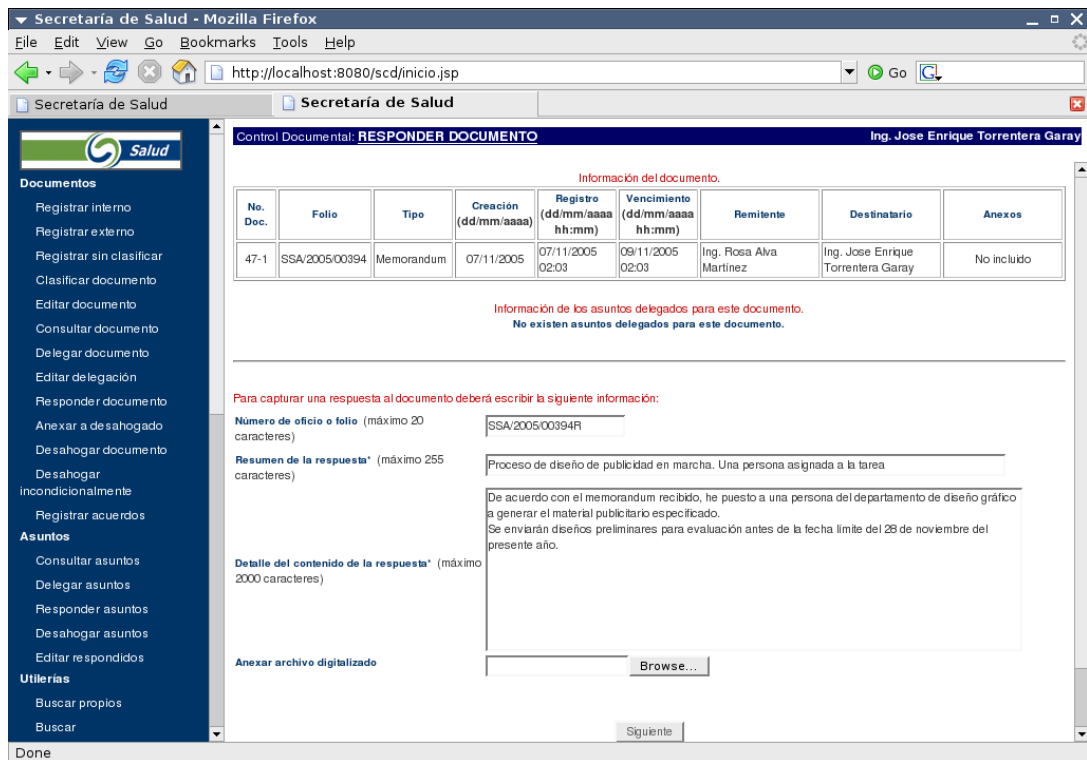
Por lo tanto ingresa a la opción *Responder Documento*, debajo del menú *Documentos*. En esta pantalla Enrique puede consultar todos los documentos que tiene pendientes de respuesta. Elige el documento que le envió Alva y prosigue a darle respuesta.

**Figura 2.9. Consulta de documentos por responder**



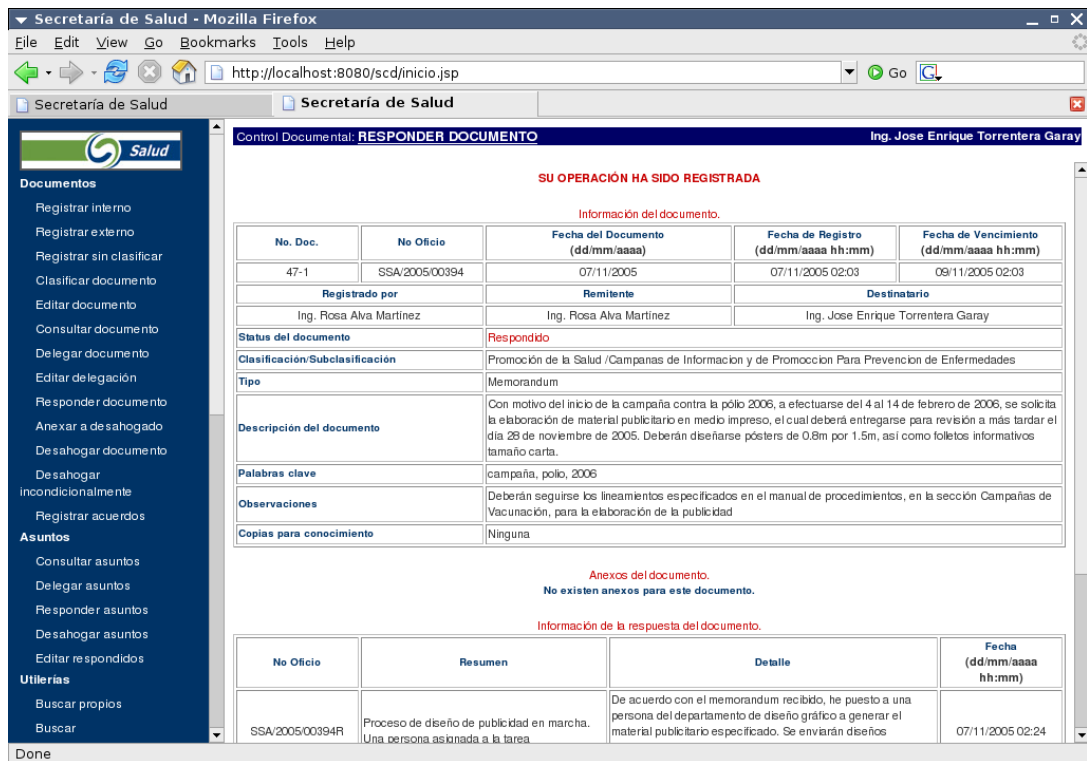
Una vez en la pantalla de respuesta de documentos, Enrique genera la respuesta apropiada de acuerdo al contenido del documento que le fue enviado. En este caso indicará que asignará recursos humanos al diseño de la publicidad necesaria para la campaña de vacunación.

**Figura 2.10. Respuesta del documento**



Una vez ingresada la respuesta para el documento, el sistema le indica a Enrique que su respuesta fue registrada correctamente. La respuesta escrita por Enrique será enviada automáticamente a Alva.

Figura 2.11. Respuesta generada y enviada al remitente del documento



Más tarde, al revisar si tiene nuevos documentos por responder, Alva se encuentra con que el documento que envió a Enrique ya tiene una respuesta asignada.

**Figura 2.12. El documento tiene una respuesta asignada**

Control Documental: PÁGINA INICIAL Ing. Rosa Alva Martínez

Resumen de Actividades

**Nota: Presione en el número para acceder a la pantalla correspondiente**

DOCUMENTOS RECIBIDOS	
Por clasificar	0
Por editar	0
Por delegar	0
Por responder o para conocimiento	0

DOCUMENTOS EN PROCESO	
Por consultar trámite	1
Por desahogar	0
Por desahogar registrados	0

ASUNTOS RECIBIDOS	
Por delegar	0
Por responder o para conocimiento	0

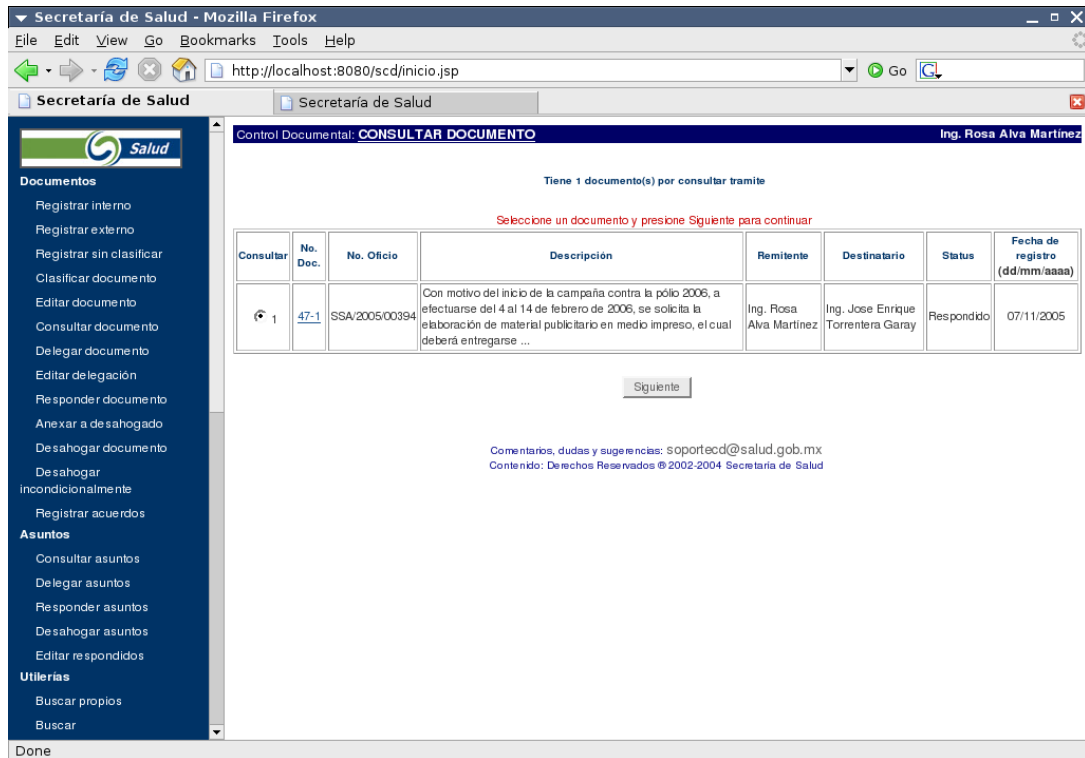
ASUNTOS EN PROCESO	
Por consultar trámite	0
Por desahogar	0

Comentarios, dudas y sugerencias: [soportecd@salud.gob.mx](mailto:soportecd@salud.gob.mx)  
Contenido: Derechos Reservados © 2002-2004 Secretaría de Salud

A continuación, Alva selecciona la opción *Consultar Documento* del menú *Documentos* para leer la respuesta que Enrique dió al memorándum enviado.

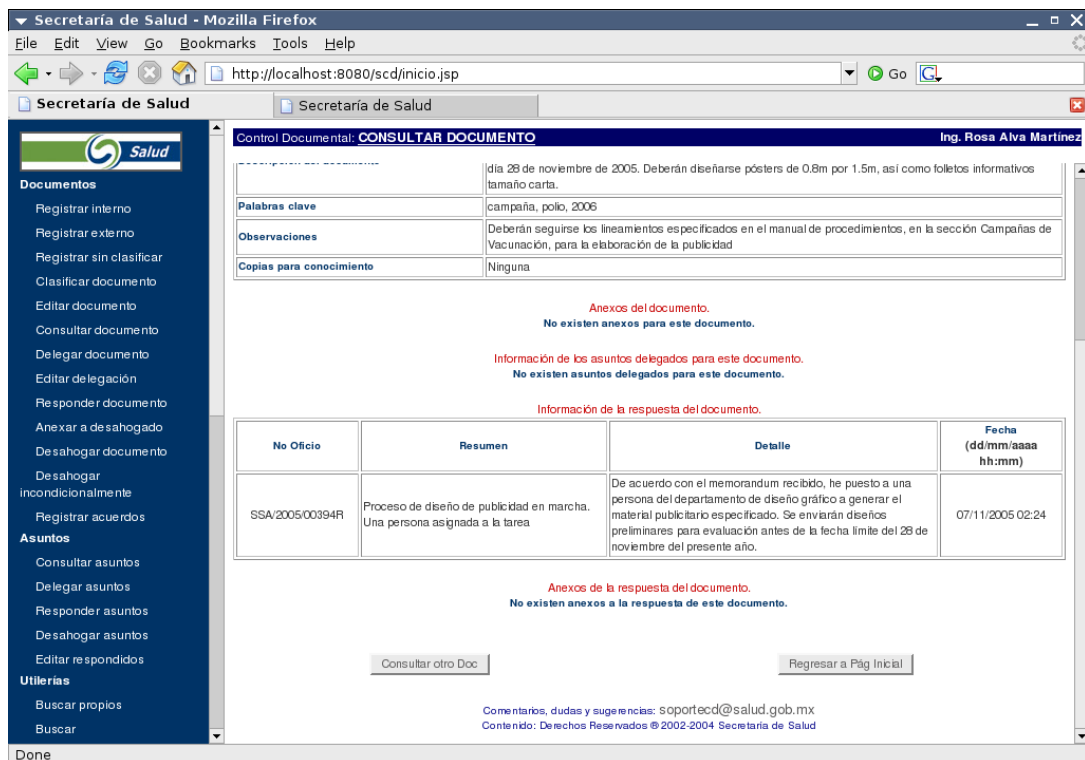
**Figura 2.13. Consulta de documentos respondidos**





Seleccionando el documento, Alva puede notar que el estado del documento ha cambiado a *Respondido* y que, además, ya tiene una respuesta asignada.

**Figura 2.14. Revisión de la respuesta al documento**



Como se puede ver, el Sistema de Control Documental es de gran ayuda para desarrollar las labores diarias dentro de una dependencia. Aunque en este ejemplo no se mostró,

Alva pudo haber anexado al memorándum original un archivo electrónico, conteniendo la publicidad del año anterior, por ejemplo, para ayudar a Enrique a desarrollar mejor la tarea encomendada. Además, Alva pudo haber enviado copias del documento a alguna persona, aparte de Enrique. Por ejemplo, si Alva hubiera recibido la orden del Subsecretario de la Secretaría de Salud, ella pudo haber enviado el memorándum a Enrique, con copia para el Subsecretario, de manera que el Subsecretario estuviera informado de que su petición ya está en marcha.

Es fácil notar el gran ahorro de tiempo y dinero que el SCD proporciona a una dependencia que hace uso de él. Ahorra tiempo al permitir a una persona estar informada del estado actual de un documento, sin necesidad de hacer visitas o llamadas telefónicas. Ese tiempo puede dedicarse a otras actividades productivas. Cuando el documento tenga algún progreso, el remitente será avisado de ello automáticamente por el SCD. Además, al prescindir de mensajeros para el envío de documentos, reduce costos innecesarios, con el beneficio adicional de que la entrega de documentos es prácticamente instantánea. En el momento en que el remitente termina de registrar el documento, éste aparece en la lista de documentos pendientes de respuesta para el destinatario.

### **3.2. Beneficios del SCD**

Además de la optimización del manejo de documentos, el SCD aporta un número de ventajas adicionales que no existen en el modelo anterior de manejo de documentos:

- Concentra en un solo lugar la información documental de las dependencias.
- Permite seguir con completo detalle el ciclo de vida de un documento y su respuesta.
- Permite realizar búsquedas sobre toda la información documental almacenada con resultados inmediatos.
- Permite generar estadísticas de uso y reportes detallados que facilitan la toma de decisiones.
- Facilita el respaldo de la información al estar centralizada en una sola ubicación.
- Permite reducir costos al prescindir de los servicios de mensajería.

El proyecto del Sistema de Control Documental ha sido instalado y utilizado exitosamente en varias dependencias internas de la Secretaría de Salud y en una dependencia externa. Durante más de dos años de operación, el SCD ha brindado a los usuarios del mismo:

- Fácil acceso a la información histórica almacenada.
- Un completo seguimiento del estado de un documento.
- La posibilidad de delegar documentos a varios usuarios para facilitar y acelerar la generación de respuestas a documentos prioritarios cuando la cantidad de documentos recibidos rebasa la capacidad de una persona para responderlos de manera satisfactoria.
- Una reducción de los costos de operación al disminuir tanto el uso de papel dentro de la dependencia, como los costos de mensajería.

## **4. Nuevos requerimientos**

El SCD ha sido ampliamente utilizado en varias dependencias de la SSA. Ha demostrado ser útil al reducir la complejidad del proceso de generación y respuesta de documentos gubernamentales. Además, ha reducido al orden de segundos la búsqueda de documentos públicos, un trámite que anteriormente duraba días o semanas.

Es indiscutible que todos estos elementos permiten calificar al SCD como un sistema exitoso en el logro de sus objetivos iniciales. A pesar de ello, todos los sistemas deben adaptarse a las necesidades de sus usuarios. El SCD no es la excepción. Durante su operación el sistema creció y mejoró de manera que pudiera adaptarse mejor al uso que los usuarios hacían de él. Muchos de estos cambios fueron mínimos, tales como modificar la interfaz de usuario o reducir el número de interacciones necesarias para realizar una tarea dentro del sistema. Algunas otras modificaciones fueron más profundas y tuvieron un impacto mayor en la funcionalidad del sistema. Todos estos cambios hicieron del SCD un sistema más apegado a las necesidades reales y cotidianas de los usuarios.

Desafortunadamente, conforme el sistema iba creciendo, tanto en funcionalidad como en popularidad, las necesidades de los usuarios afectaban más profundamente el sistema actual. El principal requerimiento fue que el SCD no debe ser afectado tan profundamente por un fallo en la infraestructura de red. El funcionamiento continuo y correcto del enlace de red estaba implícito en el diseño original del SCD, de manera que un fallo en la red de comunicaciones impedía el uso del SCD por parte de las dependencias. Esto era un riesgo inaceptable, por lo que fue necesario rediseñar el SCD de manera que se minimizara o, de ser posible, se eliminara esta dependencia en la red.

Estos nuevos requerimientos sentaron las bases para el Sistema de Control Documental Distribuido, el cual enfrenta de manera directa el problema de la vulnerabilidad del SCD a fallos o interrupciones en la red de comunicaciones.

# Capítulo 3. El Sistema de Control Documental Distribuido

En este capítulo se describen los requerimientos y la arquitectura del Sistema de Control Documental Distribuido.

## 1. Requerimientos

El Sistema de Control Documental Distribuido debe cumplir con los siguientes requerimientos:

1. El SCDD debe funcionar de manera confiable, segura y transparente para los usuarios, durante períodos razonables de tiempo, en caso de una interrupción de las comunicaciones de red.
2. El SCDD debe mantener la integridad de la información documental en todo momento.

El SCDD deberá elevar el nivel de confiabilidad y disponibilidad global del servicio proporcionado a sus usuarios. De esta manera, el objetivo principal del SCD se sigue cumpliendo, a saber: facilitar y automatizar la administración de documentos.

Al no verse afectado por interrupciones de red, el SCDD permite al usuario continuar, de manera ininterrumpida, sus labores cotidianas.

Con estos requerimientos en mente se desarrolló una nueva arquitectura para el SCDD, que permite cumplirlos pero sin descuidar el objetivo original que dió origen al SCD.

Al margen de los requerimientos funcionales del SCDD, existen también otros requerimientos no funcionales, relacionados con el costo, tanto de desarrollo como de mantenimiento del nuevo sistema. Estos requerimientos no funcionales se enlistan a continuación:

1. El SCDD debe utilizar tecnología moderna, confiable y segura para su realización.
2. El SCDD debe utilizar protocolos y estándares internacionales para la transmisión y almacenamiento de información.
3. El SCDD debe utilizar técnicas probadas y sólidas para su desarrollo.
4. El costo asociado con el desarrollo del SCDD debe ser razonable, de acuerdo con el presupuesto destinado, pero sin sacrificar la calidad del mismo.
5. El costo para una dependencia para incorporarse al SCDD deberá ser mínimo, de manera que el mayor número de dependencias puedan ser parte de él.
6. El costo por mantenimiento del SCDD deberá ser razonable, permitiendo a dependencias que tengan un presupuesto más bajo acceder a los servicios del sistema.

Estos requerimientos impactan de manera indirecta al funcionamiento del SCDD. El uso de tecnología moderna, pero confiable y probada, permite que el sistema se beneficie de los nuevos adelantos en el área de tecnologías de la información, sin que se fundamente

en tecnologías demasiado nuevas o aún en proceso de maduración. Lo mismo sucede con las técnicas de desarrollo.

El caso de los protocolos y estándares internacionales merece una explicación más detallada. Los estándares, como su nombre lo indica, permiten mantener un idioma y convenciones comunes en determinada área del conocimiento. Así como el Sistema Internacional de Unidades nos permite saber con precisión el significado de la expresión "el puente tiene 43.5 metros de largo", sin importar si somos mexicanos, españoles o franceses, ya sea ahora como dentro de 50 años; un estándar para transmisión o almacenamiento de datos nos permite reconocer, extraer e interpretar la información en cualquier lugar y cualquier momento, basándonos en lo estipulado por él.

Esta garantía de acceso a la información independientemente del tiempo y el lugar ya es de por sí un aspecto importantísimo. Pero además hay un segundo beneficio en el uso de estándares y protocolos internacionales, en comparación con protocolos y estándares propietarios desarrollados dentro de alguna empresa y destinados al mismo fin: proporcionan libertad; libertad tanto de elegir las herramientas para acceder a la información, como de desarrollar nuevos sistemas que operen con la información o que puedan interactuar con los sistemas actuales. Si las herramientas están desarrolladas de manera que puedan operar con información basada en un estándar, entonces podemos elegir entre las distintas herramientas de acuerdo al precio, calidad, disponibilidad de la herramienta; y elegir la que mejor convenga de acuerdo a nuestros fines y presupuesto.

Si la información se almacenara utilizando un estándar propietario controlado por alguna empresa privada, entonces las herramientas para acceder a nuestra información solamente las podríamos adquirir de ellos mismos o de un tercero autorizado por ellos. Esto, en sí, ya es un inconveniente y va en detrimento de nuestra libertad de elegir el mejor producto de acuerdo a nuestras necesidades. El mayor problema es que, si la empresa privada desapareciera por algún motivo, entonces nos quedaríamos en un callejón sin salida con respecto a nuestra información. Sería nuestra información, pero no podríamos accederla porque estaría almacenada mediante herramientas que ya no se fabrican o que no tendríamos derecho a fabricar por nosotros mismos debido a patentes o derechos de autor.

Si la información estuviera almacenada utilizando estándares internacionales y abiertos, entonces, en cualquier momento podríamos decidir cambiar de proveedor de las herramientas para acceder a ella. Aún más, podríamos nosotros mismos crear las herramientas para acceder a la información. Todo esto sin importar si alguna de las empresas que proporcionaba la herramienta original desapareció.

Esto adquiere mayor importancia en el área de gobierno, donde la información debe estar disponible durante muchos años después de su creación. Si está almacenada utilizando estándares internacionales, un ciudadano dentro de 50 años puede solicitar acceso a la información creada y almacenada en este momento y no tendrá absolutamente ningún problema en conseguir una herramienta que le permita acceder a la información. Aún si no existiera una herramienta en ese entonces, debido a que ya nadie la fabrica por ser tecnología atrasada, puede él mismo o un tercero bajo su encargo, crear la herramienta basándose en el estándar publicado 50 años antes. En cualquier caso, tiene libertad completa para acceder a la información deseada. Ese es el principal beneficio de usar estándares internacionales y abiertos.

## 2. Arquitectura

El SCDD consiste de una red de nodos interconectados entre sí a través de Internet. Cada nodo esta conformado por un servidor y una base de datos. Hay dos tipos de nodos en dicho sistema, Nodo Central y Nodo Local. El primero es indispensable para que exista el SCDD; el segundo no.

El SCDD más pequeño posible tiene solamente al Nodo Central sin ningún Nodo Local, pero un sistema de este tipo no tiene uso práctico.

El SCDD mínimo con utilidad práctica está formado por el Nodo Central y un Nodo Local. Aún así, su utilidad, aunque práctica, es limitada.

El SCDD típico está formado por el Nodo Central y varios Nodos Locales.

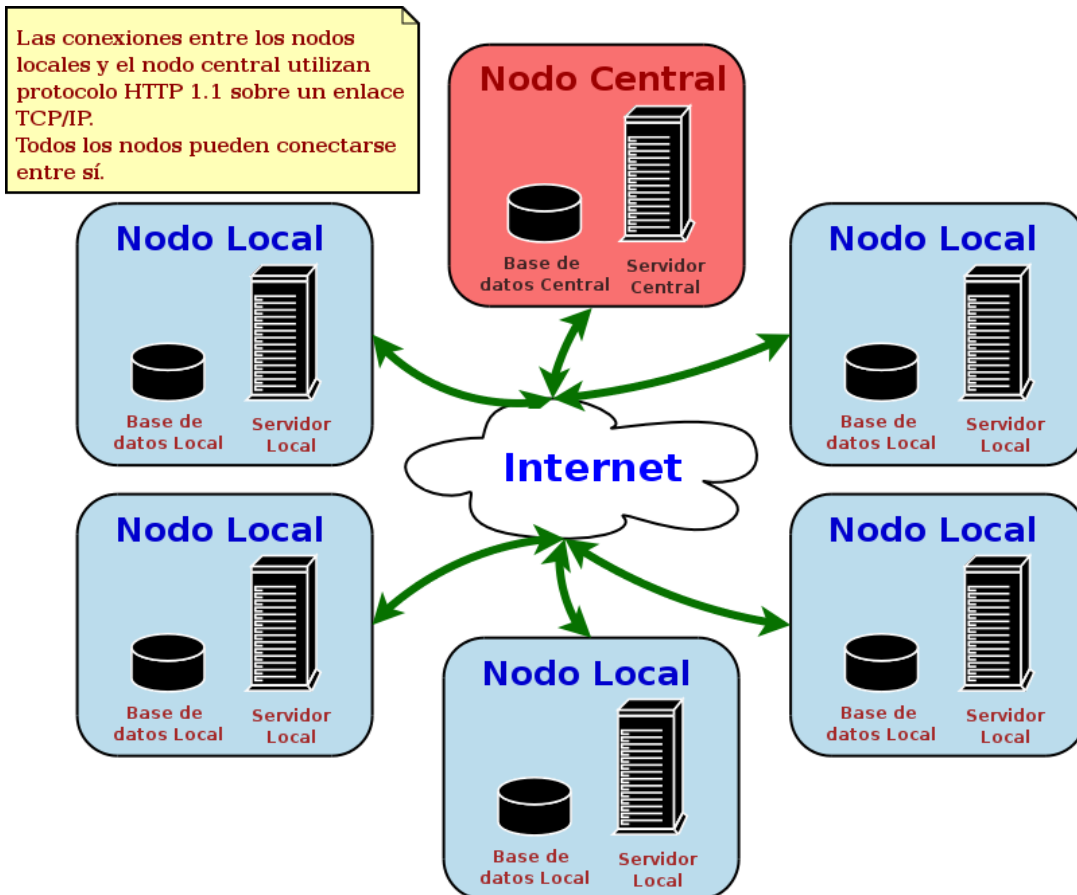
Aunque no hay límite teórico al número de Nodos Locales existentes en determinado momento en el SCDD, sí existe un límite físico, y está determinado por la capacidad del enlace de entrada del Nodo Central.

Para cada nodo, el servidor es la parte responsable de proporcionar a los usuarios los servicios de administración y control de documentos. La base de datos es donde se almacena la información documental.

Aunque no tiene utilidad práctica tener un Nodo Central por si mismo, tener un Nodo Local sí la tiene. Un Nodo Local por sí mismo es una instancia del SCD. Así pues, se puede decir que el SCDD es una red de instancias del SCD que trabajan de manera coordinada.

En la siguiente figura se muestra la arquitectura general del SCDD:

### **Figura 3.1. Arquitectura del SCDD**



El Nodo Central tiene las siguientes responsabilidades:

1. Mantener en la Base de Datos Central una copia de toda la información existente en cada una de las Bases de Datos Locales existentes en el SCDD.
2. Mantener información global necesaria para el correcto funcionamiento del SCDD.

Cada Nodo Local, a su vez, tiene las siguientes responsabilidades:

1. Proporcionar los servicios de administración y control documental a los usuarios que tiene asignados.
2. Almacenar toda la información documental en la base de datos local.
3. Proporcionar servicios de búsqueda sobre la información almacenada en la base de datos local.
4. Asegurar que la información esté siempre actualizada y sea consistente dentro del SCDD.

Todos los nodos están conectados a través de Internet. En el caso de una interrupción del enlace de red, el Nodo Local es capaz de seguir operando durante un tiempo considerable debido a que cuenta con su propio Servidor y Base de Datos Local. De esta manera puede seguir atendiendo a sus usuarios, ya que la información generada es almacenada en la Base de Datos Local. Por tanto, un fallo en la red es imperceptible para los usuarios del SCDD.

Esta arquitectura distribuída permite que un nodo continúe su operación en el caso de una interrupción en la red. Esta interrupción puede durar desde segundos hasta varios días. Toda la información generada durante este tiempo es almacenada localmente, de manera que, cuando la conexión se restablece, el Servidor Local hace contacto con el Servidor Central y transmite la información que estuvo almacenando. Además, solicita la información que se pudo haber generado durante el período que permaneció incomunicada y la almacena en la Base de Datos Local.

De esta manera, el SCDD se encuentra nuevamente con la información lo más actualizada posible. A este proceso se le llama "*Replicación de Información*". Al final del proceso se dice que la información del Nodo Local está "*sincronizada*" con la información del SCDD.

Durante el período de incomunicación del Nodo Local con el resto del SCDD, aquel está verificando constantemente si la conexión se ha restablecido. En caso de ser así, automáticamente inicia el proceso de replicación. Si no hay conexión todavía, intentará después de un período de espera.

Uno de los objetivos del SCDD es asegurar que la información almacenada en él sea siempre consistente e íntegra. Por tanto, si durante la replicación de datos se produce un error, o si la conexión se pierde nuevamente, el proceso se aborta sin que esto afecte la integridad de la información del SCDD. En el caso de un aborto en el proceso de replicación, el Servidor Local intentará nuevamente sincronizar su información después de un período de tiempo predeterminado. El proceso de replicación de información es, pues, atómico, indicando con esto que, o se copia toda la información de manera correcta, o no se copia nada de la información. Esto asegura que el SCDD siempre tiene información íntegra, completa y lo más actualizada posible, sin importar los fallos que pueda tener la red ni la frecuencia de éstos.

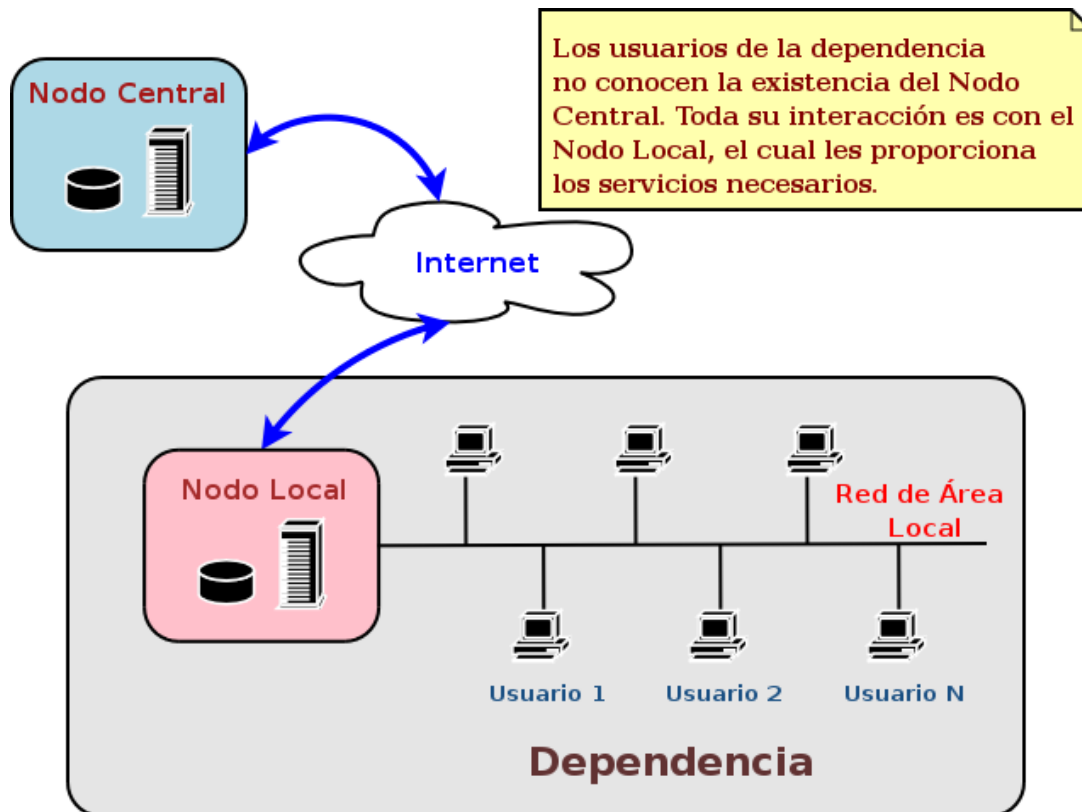
Lógicamente, el sistema está formado por varios Nodos Locales y un Nodo Central. Físicamente, un Nodo Local está generalmente localizado dentro de las instalaciones de una dependencia de la SSA. Lo mismo sucede con el Nodo Central.

De acuerdo a su disposición física y geográfica, los Nodos Locales se pueden clasificar en dos tipos:

- A. Un Nodo Local se localiza dentro del edificio de una dependencia y proporciona los servicios del SCD a sus usuarios locales. Este escenario se muestra en la siguiente figura:

### **Figura 3.2. Escenario A**

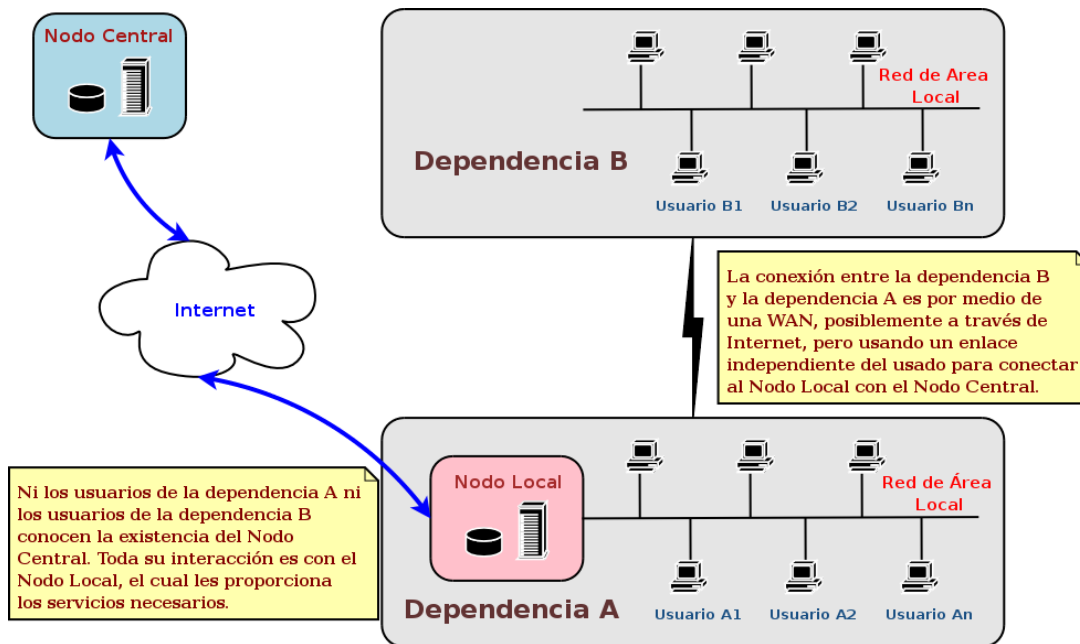




Si la conexión a Internet se pierde entre el Nodo Local y el Nodo Central, los usuarios de la dependencia pueden continuar trabajando de manera ininterrumpida durante varios días, ya que el Servidor Local les permite administrar y controlar documentos, almacenándolos en la Base de Datos Local. Al mismo tiempo, el Servidor Local intentará constantemente hacer contacto con el Servidor Central de manera automática. Cuando se restablezca la conexión, replicará la información generada y almacenada localmente y se sincronizará con el SCDD. Después de la sincronización, el Nodo Local operará como siempre, sin afectar a sus usuarios en ningún momento. La mayoría de los Nodos Locales del SCDD son de este tipo.

- B. Un Nodo Local se localiza dentro del edificio de una dependencia pero proporciona los servicios del SCD a sus usuarios locales y a los usuarios locales de otra dependencia. Estas dos dependencias están conectadas por un enlace de red que es independiente del existente entre el Nodo Local y el Nodo Central.

**Figura 3.3. Escenario B**



De manera similar a lo que ocurre en el escenario A, los usuarios pueden continuar con sus labores en el caso de un fallo en la conexión entre el Nodo Central y el Nodo Local. En este escenario los usuarios de la dependencia B pueden quedar incomunicados e incapaces de utilizar los servicios del SCD en el Nodo Local. Pero esto es un caso especial en el que el enlace WAN se pierde, siendo este enlace independiente del existente entre el Nodo Central y el Nodo Local. Idealmente, la conexión WAN no tendría relación con el enlace de acceso usado por el Nodo Local, de manera que no se pierdan ambos al mismo tiempo. Por ejemplo, el enlace WAN puede ser a través de routers en ambos extremos del enlace, los cuales transmiten toda la información entre dependencias usando fibra óptica o un enlace de microondas. En general, el enlace WAN conectaría dependencias localizadas en edificios cercanos geográficamente, por ejemplo uno enfrente de otro, o separados como máximo por unos cientos de metros. Es posible también que el enlace WAN sea a través de Internet, siempre y cuando sea un enlace independiente del usado entre el Nodo Local y el Nodo Central, por ejemplo cuando los accesos a Internet son proporcionados por diferentes proveedores, cada uno usando su propia infraestructura. Por último es necesario notar que el número de dependencias subordinadas a la dependencia que contiene al Nodo Local no tiene un límite teórico. Aunque en el diagrama muestra una sola dependencia subordinada, puede haber en todo momento varias dependencias subordinadas, siempre y cuando utilicen un enlace independiente al de los nodos.

Como se puede notar, el SCD original es un Nodo Local del tipo B, donde varias dependencias se conectan a un solo Nodo usando enlaces a través de una WAN o de Internet.

Debido a que la mayoría de los Nodos Locales del SCDD son del tipo A, hay una constante comunicación entre Nodos Locales, además del Nodo Central, durante la operación normal del sistema. Esto es así debido a que muchos de los documentos generados por un usuario de una dependencia (la cual tiene su propio Nodo Local), van dirigidos a otro usuario en otra dependencia (la cual tiene, a su vez, su propio Nodo).

Como el destinatario del documento no está asignado al mismo Nodo Local al que está asignado el remitente, el Nodo Local del remitente necesita hacer una conexión con el Nodo Local del destinatario y transmitir la información necesaria para que el destinatario tenga conocimiento del documento. El efecto de esto es que el SCDD siempre tiene una comunicación entre los distintos Nodos del sistema.

En caso de que un usuario genere un documento durante el período de tiempo en el cual no hay conexión entre su Nodo Local y el resto del SCDD, el Servidor Local genera el documento y lo almacena en la Base de Datos Local, en espera de que se restaure la conexión y pueda realmente enviar el documento al Nodo Local del destinatario. Si el destinatario del documento está asignado al mismo Nodo Local de remitente, la entrega del documentos se hace de manera inmediata y, en consecuencia, el fallo en la conexión de red no tiene influencia en este caso.

Por otro lado, el remitente no sabe que el documento está demorado. Para él, el documento fue creado y de alguna manera se hará llegar a su destino. Para todos los efectos prácticos, el usuario ya envió su documento. La persona encargada de administrar el Nodo Local, por el contrario, debe asegurarse de que el enlace se restablezca lo más pronto posible, de manera que el Nodo Local pueda replicar la información y se sincronice con el SCDD. Si no lo hace así, un documento que tenga asignado un tiempo de respuesta muy corto, puede vencerse antes de ser transmitido al Nodo Local del destinatario.

Por ejemplo, suponiendo que el documento tenía un tiempo de vencimiento de 8 horas (es decir, que el documento debía responderse antes de 8 horas) y que la falla en el enlace de red dure 24 horas, para cuando el documento logre transmitirse al Nodo Local del destinatario, llevará por lo menos 16 horas vencido.

Esto es un inconveniente de la arquitectura que no se puede evitar. Pero sí hay maneras de reducir su impacto y minimizar, en consecuencia, los efectos que esto puede tener. Algunos de ellos se mencionan a continuación:

- En toda dependencia que tenga un Nodo Local debe existir un plan de contingencia que establezca acciones a tomar y cuyo efecto sea el minimizar la duración de la desconexión del enlace de red.

Para poder saber cuál es el tiempo máximo que puede durar un fallo en el enlace de red sin que afecte los documentos generados se puede realizar un análisis estadístico cuyo objetivo sea conocer el tiempo de respuesta promedio asignado a los documentos de la dependencia en cuestión. Este tiempo de respuesta promedio proporcionará una guía para la determinación del tiempo máximo que puede durar la desconexión.

Por ejemplo, si el tiempo de respuesta promedio asignado a los documentos del Nodo Local es de 8 horas, entonces se podría elegir 4 horas como el tiempo máximo de la desconexión. El razonamiento seguido para elegir este tiempo es el siguiente: Suponiendo que un documento se genera inmediatamente antes del fallo en la red, y el restablecimiento del enlace dura cuatro horas, entonces, para cuando el enlace sea restablecido al documento le restarán solamente 4 horas antes de su vencimiento. Estas cuatro horas deben ser suficientes para que el

documento se replique al Nodo Local del destinatario y para que el destinatario lo responda adecuadamente. Si se hubiera elegido un valor muy cercano al tiempo promedio de respuesta del Nodo Local del remitente, entonces, para cuando el documento llegara al Nodo Local del destinatario, éste tendría muy poco tiempo para responderlo o ya estaría vencido, dependiendo de cuán frecuentemente revise si le han llegado nuevos documentos para responder.

- En las dependencias cuyos recursos lo permitan, debe existir un enlace confiable que permita mantener conexión con el Nodo Central en todo momento.

Un enlace confiable puede conseguirse de varias maneras:

- Contratando enlaces a Internet con varios proveedores, los cuales proporcionen medios de acceso a Internet independientes entre sí, de manera que si uno de ellos falla, pueda aún conectarse a Internet utilizando alguno de los restantes enlaces.
- Contratando con un mismo proveedor dos enlaces redundantes e independientes, cada uno con su propio medio de acceso a Internet.
- Contratar un servicio de acceso a Internet garantizado con un proveedor, de manera que, en caso de interrupción de la conexión, se pueda exigir el restablecimiento del mismo en un período de tiempo predeterminado y fijo, se puedan fijar sanciones o penalización monetarias por fallos en el servicio contratado basados en el tiempo de interrupción, o ambos.

El enlace redundante permite que la conexión esté siempre disponible, siempre y cuando no existan desperfectos físicos en el medio de acceso a Internet. El servicio de acceso garantizado permite obtener compensación en el caso de un fallo en la red. Al estar basado en el tiempo de falla, el proveedor del servicio tiene un incentivo más para corregir el problema en el menor tiempo posible.

Cada una de estas opciones tiene ventajas y desventajas. El enlace redundante, por un lado, permite tener un enlace siempre disponible, pero presupone que el medio de acceso no sufrirá fallos físicos. Esto es algo muy importante a tener en cuenta, ya que no existe equipo o medio de acceso inmune a fallos. Aún teniendo en cuenta que el medio de acceso es redundante y es proporcionado por proveedores independientes, no es aconsejable descartar que ambos fallen al mismo tiempo. Obviamente también es posible que nunca fallen los dos al mismo tiempo. El servicio de acceso garantizado, por su parte, no presupone que el medio de acceso a Internet sea inmune a fallos. Mas bien asegura que el enlace, en caso de falla, se restablecerá lo más pronto posible. A pesar de que con esta opción no se tiene disponibilidad permanente del enlace, por lo menos se asegura que el enlace se restablecerá en el menor tiempo posible. Además de esto, la dependencia puede obtener otro tipo de beneficios derivados de las penalizaciones establecidas en el contrato con el proveedor del enlace, tales como descuentos o bonificaciones, tanto monetarias como en el servicio contratado.

Esta es la manera en la que el SCDD cumple con los objetivos planteados inicialmente.  
Resumiendo:

- Debido a que un Nodo Local es una entidad independiente, autónoma y autosuficiente que puede proporcionar los servicios del SCD a los usuarios de su dependencia, es prácticamente inmune a fallos en la red que lo conecta con el resto del SCDD. Es, pues, independiente a fallos en la red de comunicaciones.
- Debido a que el Nodo Local replica la información de manera atómica, la integridad, precisión y actualidad de la información del SCDD está siempre asegurada.

Además de cumplir con los objetivos funcionales, el SCDD también cumple con los objetivos no funcionales planteados inicialmente:

- Debido a que utiliza técnicas de programación orientada a objetos, el SCDD es mucho más sencillo en su diseño, así como más claro en su programación.
- El SCDD usa XML para transmitir toda la información entre Nodos. Esto permite que la información esté disponible en cualquier momento futuro a cualquier herramienta que opere sobre información codificada en dicho formato.
- Al utilizar conexiones TCP/IP, las cuales están establecidas en un estándar internacional ampliamente utilizado y probado, el sistema puede adaptarse más fácilmente a los nuevos medios de comunicación existentes y futuros. Por ejemplo, si en algún momento se decidiera que todos los enlaces entre nodos se convirtieran a fibra óptica o a enlaces por microondas, el sistema podrá utilizarlos sin modificación alguna, siempre y cuando los ruteadores que conectan a la fibra óptica o al enlace de microondas acepten TCP/IP como protocolo para transmisión de datos.
- El SCDD utiliza el servidor Tomcat de la fundación Apache como servidor y el administrador de bases de datos PostgreSQL como base de datos en cada uno de los nodos del SCDD. Ambos utilizan estándares internacionales abiertos y publicados. Tomcat utiliza las especificaciones públicas para Servlet™ 2.3 y JavaServer Pages™ 1.2. PostgreSQL utiliza la especificación SQL 92. Por tanto, ambos proporcionan seguridad hacia el futuro de que la información depositada en el SCDD se mantendrá completamente disponible.

### 3. Beneficios del SCDD

Con el Sistema de Control Documental Distribuido:

1. Aumenta la disponibilidad de sistema.

Al existir un Nodo Local para cada Dependencia, es menor el impacto que pueda tener un fallo físico del mismo. Solamente la dependencia cuyo Nodo Local tuvo el desperfecto se verá afectada. Las demás dependencias continuarán con su operación normal sin siquiera notar que una de ellas dejó de funcionar.

En el SCD, si el Servidor Único se ve afectado por algún fallo, todas las dependencias ven interrumpidas sus operaciones, causando un desastre mayor.

2. Se reduce el impacto de fallos en la red en la operación del sistema.

Al tener cada dependencia su propio Nodo Local, durante un fallo en las comunicaciones las dependencias funcionan como entidades autónomas,

recolectando y procesando información, de manera que la interrupción de la red pasa desapercibida para el usuario y éste puede continuar con sus labores de manera ininterrumpida. Cuando el enlace de red se restablece, el Nodo Local se sincroniza automáticamente con el Nodo Central y con los otros Nodos Locales, de manera que la información siempre está actualizada y es consistente a lo largo del SCDD.

3. Se mantiene intacta en todo momento la integridad de la información documental depositada en el sistema.

La replicación de la información documental del SCDD, ya sea durante su operación normal o durante la sincronización después de una falla en el enlace de red, sucede de manera atómica, de manera que la información se transmite completa de manera correcta o no se transmite nada. De esta manera, el SCDD automáticamente mantiene la integridad y precisión de la información existente en el sistema.

# Capítulo 4. Diseño y Desarrollo del SCDD

En este capítulo se abordan los detalles del diseño y el desarrollo del SCDD. Se detallan las decisiones tomadas y se muestra cómo se realiza la arquitectura planteada en el capítulo anterior.

## 1. Servidores

Para crear el SCDD se utilizará la misma infraestructura de servidores del SCD. Esto es, cada Nodo Local estará conformado por una computadora en el papel de servidor, la cual tendrá instalado el siguiente software:

- Sistema Operativo: Fedora Core 1 o superior con actualizaciones de seguridad.
- Administrador de Base de Datos: PostgreSQL 7.3.4
- Contenedor de *Servlets*: Apache Tomcat 4.1.31

### 1.1. Fedora Core 1

Fedora Core 1 es una distribución GNU/Linux basada en el kernel versión 2.4.22. Al estar basada en Linux, el servidor proporciona confiabilidad, excelentes tiempos de operación continua y la seguridad de que se obtendrán actualizaciones de seguridad de manera oportuna. Todo esto con la ventaja del bajo costo de adquisición y de operación asociado generalmente a los sistemas GNU/Linux

El Sistema Operativo opera con la mayoría de los servicios deshabilitados, de manera que se aumente la seguridad al tener menos puntos de ataque. El único servicio que tiene activado es el necesario para que los usuarios puedan hacer uso del SCDD.

Además, debido a la frecuencia con que aparecen nuevas versiones del sistema operativo, se hace uso del proyecto Fedora Legacy para aumentar el tiempo de soporte durante el cual Fedora Core 1 recibirá actualizaciones de seguridad.

En determinado momento se deberá actualizar el sistema operativo, ya sea porque se desea contar con alguna nueva característica existente en la nueva versión del mismo, o porque el período de soporte proporcionado por Fedora Legacy se acerca a su fin. En este caso, de ser necesario, el SCDD puede migrarse sin problemas a la nueva versión del sistema operativo, o en caso de que convenga por algún motivo, migrarse de sistema operativo a algún otro basado en Linux; por ejemplo, Debian GNU/Linux.

Un sistema operativo GNU/Linux proporciona varias ventajas sobre un sistema operativo propietario como Microsoft Windows o Sun Solaris, entre ellos:

- Los requerimientos de hardware son mínimos y solamente determinados por la aplicación que se dará al servidor en última instancia.
- No tiene costos anuales por concepto de licencias de uso.
- Si no se cuenta con personal capacitado, puede contratarse un servicio de soporte técnico para resolver problemas que puedan surgir durante la operación del servidor.

- Los errores en el sistema operativo, así como en las aplicaciones que funcionan en él, son corregidas de manera oportuna y puestas a disposición de los administradores de sistemas, de manera que puedan ser instaladas de manera sencilla.
- Inmunidad prácticamente total a virus, *spyware* y otros tipos de código malicioso frecuentemente asociados con sistemas propietarios.
- Libertad de elegir el momento de actualizar el sistema operativo o el hardware sobre el que se ejecuta.

Debido a estas razones, y tomando en cuenta el requerimiento del SCDD de reutilizar al máximo posible la inversión en infraestructura del SCD, se decidió continuar con el sistema operativo Fedora Core 1 para el desarrollo e implantación del mismo.

## 1.2. PostgreSQL

Se eligió continuar utilizando el administrador de bases de datos PostgreSQL debido a los excelentes resultados obtenidos durante su uso en el SCD.

PostgreSQL es un sistema de base de datos relacional poderoso y de software libre. Tiene más de 15 años de desarrollo activo y, debido a su arquitectura probada, se ha ganado una excelente reputación en cuanto a confiabilidad, fortaleza, desempeño e integridad de la información que maneja. PostgreSQL es un administrador de bases de datos maduro, confiable, eficiente y de calidad, que soporta grandes cargas de trabajo y que proporciona excelentes tiempos de respuesta a consultas de datos, a la par de productos profesionales como Oracle o Sybase. Soporta el estándar SQL 92, con algunas extensiones, de las cuales solamente se decidió utilizar un conjunto limitado de ellas durante el desarrollo, tanto del SCD como del nuevo SCDD, de manera que si en algún momento existiera la necesidad de utilizar otro administrador de bases de datos, la migración de la información y del código fuente del sistema fuera lo más sencilla posible.

Como se mencionó, PostgreSQL proporciona muy buenos tiempos de respuesta a consultas de información. Esto es un factor necesario para el correcto y eficaz funcionamiento del SCDD. Desafortunadamente, en el momento de desarrollo de dicho sistema, PostgreSQL no contaba con la funcionalidad necesaria para resolver uno de los problemas claves para que aquél pudiera construirse. Nos referimos a la replicación de información. Esta funcionalidad es necesaria para poder transferir la información documental entre los nodos del SCDD. Al no contar con ella de manera integrada, un subproblema del sistema, surgido durante la etapa de desarrollo del mismo, fue desarrollar un sistema de replicación de información adaptado a las necesidades del sistema. Así pues, aunque no es parte de los objetivos originales del SCDD, el esquema de replicación terminó convirtiéndose en un objetivo implícito imprevisto, pero totalmente necesario, para el desarrollo exitoso del SCDD.

Este sistema de replicación de datos no intenta ser un sistema de replicación completo, en el sentido de contar con todas las características que proporciona un sistema de replicación de datos comercial como el ofrecido, por ejemplo, por Oracle. Más bien, es un sistema que debe cumplir los siguientes requerimientos:

1. Debe transferir información entre nodos de manera atómica.



2. Debe ser configurable, de manera que se puedan añadir nodos al SCDD de manera eficiente, y que éstos puedan intercambiar información con los nodos ya existentes en el sistema.
3. Debe existir la posibilidad de eliminar nodos del SCDD, sin que esto afecte el funcionamiento del sistema como un todo.
4. Debe ser resistente a fallos en las comunicaciones de red.

Si estos requerimientos se satisfacen, el SCDD podrá operar de manera correcta y cumplirá, a su vez, sus propios objetivos.

Como se verá más adelante, varios de los algoritmos del SCDD fueron creados específicamente para cumplir o contribuir a cumplir con estos requerimientos.

Además, se utilizaron características proporcionadas por PostgreSQL para asegurar la atomicidad de las transferencias de información, por lo que, aunque no es realizada directamente por PostgreSQL, la replicación de información es ayudada y respaldada directamente por las características proporcionadas por PostgreSQL. Esto nos permite tener un elevado nivel de certeza en la integridad de los datos almacenados en el SCDD. Además, nos permite concentrarnos en cumplir los objetivos del SCDD. Así pues, PostgreSQL es una parte importantísima y vital del SCDD.

Por otro lado, el sistema de replicación se desarrolló de manera que fuera muy sencillo utilizar un sistema que incluya soporte para replicación de manera natural. Esto puede darse de dos maneras:

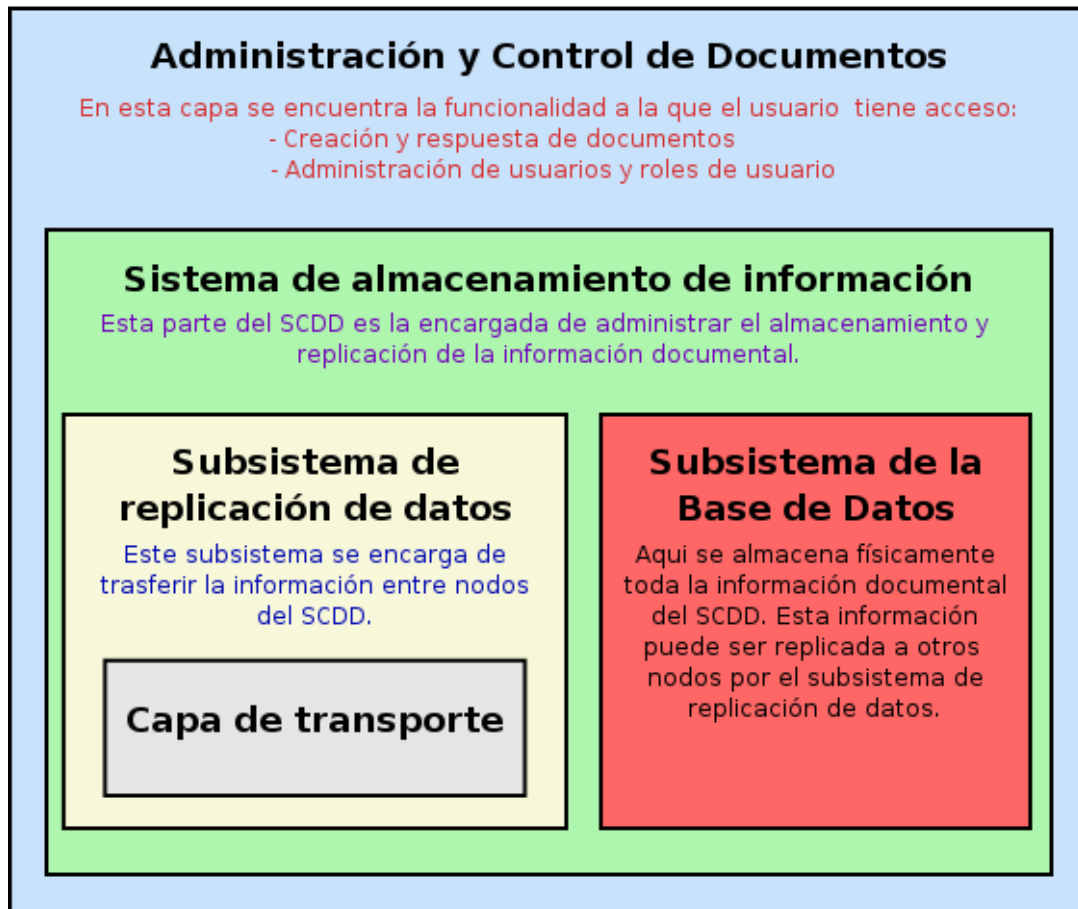
- Cambiando PostgreSQL por un administrador de bases de datos que incluya soporte para crear un esquema de replicación de manera directa, utilizando las características propias del nuevo administrador de bases de datos.
- Cambiando PostgreSQL por una versión futura que incluya ya soporte para la creación de esquemas de replicación.

En cualquiera de estos casos, el cambio no debe representar gran cantidad de trabajo, ya que el SCDD es completamente modular y, además, está estructurado en capas, de manera que la capa que hace de intermediario entre el SCDD y el subsistema de replicación está completamente definida y delimitada, haciendo al SCDD relativamente inmune al cambio de administrador de base de datos o del subsistema de replicación de datos.

Esto se puede entender mejor con el siguiente diagrama:

#### **Figura 4.1. Estructura en capas del módulo de replicación del SCDD**

## Estructura en capas del SCDD



Como se ve en la figura, el SCDD contiene capas y subsistemas completamente definidos de manera que es posible cambiar la parte que realiza determinadas funciones por otras que proporcionen la misma funcionalidad; todo esto de forma sencilla y con el mínimo esfuerzo implicado.

Esta figura muestra lo sencillo que sería cambiar de PostgreSQL a Oracle, por ejemplo, si fuera necesario. Ya que la base de datos se encuentra aislada y contenida dentro del sistema de almacenamiento de información, los usuarios, los cuales solamente acceden a la parte de administración y control de documentos, no notarían que hubo un cambio en el administrador de bases de datos.

De manera similar, si se cambiara a Oracle y se utilizara el esquema de replicación incorporado que proporciona, entonces lo único que cambiaría sería el subsistema de replicación de datos y el subsistema de la base de datos. Nuevamente, al estar ambos contenidos dentro del sistema de almacenamiento de información, esto pasaría desapercibido para los usuarios del SCDD.

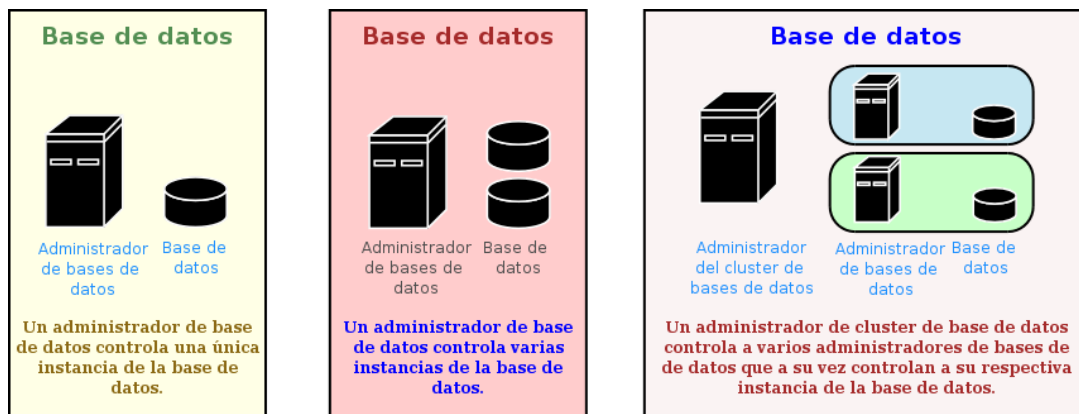
Esto muestra la solidez de la arquitectura interna del SCDD. Lleva en sí misma facilidades para crecer y para mantenerse actualizada de acuerdo a las nuevas tecnologías que puedan surgir.

Otro punto, quizá más importante a mediano plazo, es la facilidad que esta arquitectura proporciona para responder al crecimiento en el número de usuarios y en la cantidad de información almacenada y consultada en el sistema.

Aunque lógicamente el subsistema de la base de datos parece indicar que una sola instancia maneja toda la información, nada impide que un cluster de administradores de bases de datos cooperando de manera coordinada sea el que esté administrando la información documental.

En el siguiente ejemplo se muestran las diferentes configuraciones que puede tener el subsistema de la base de datos:

**Figura 4.2. Configuraciones del Subsistema de la Base de Datos del SCDD**



Como se puede ver, inicialmente la base de datos puede ser PostgreSQL manejando una sola instancia de la base de datos. Pero conforme el sistema crezca en número de usuarios y en cantidad de información almacenada, será necesario modificar este esquema original, de manera que siga siendo eficiente para realizar las operaciones del SCDD. En una dependencia mediana, puede ser necesario que se tenga un administrador de base de datos que sea capaz de utilizar varias instancias de la base de datos, de manera que la eficiencia en el almacenamiento, acceso y búsqueda de información se distribuya entre las instancias y el rendimiento y eficiencia del sistema permanezcan óptimos.

En el caso de Nodo Central, probablemente sea necesario a corto plazo crear una configuración en cluster, como la mostrada en la figura, de forma que pueda aceptar y almacenar toda la información del SCDD. Al tener el Nodo Central una copia de toda la información existente en el SCDD, conforme éste crezca, deberá hacerlo también su base de datos. Más aún, dada la importancia del papel que juega el Nodo Central en el SCDD, deberá asegurarse de que la base de datos sea capaz de operar con grandes volúmenes de información de manera eficiente en todo momento.

En cualquier caso, esto es una posibilidad remota, ya que PostgreSQL es capaz de manejar grandes cantidades de información de manera eficiente y confiable. Pero aún así es tranquilizante saber que, en caso de ser necesario cambiar de administrador de bases de datos por alguna razón, el diseño del SCDD lo permite completamente.

### 1.3. Apache Tomcat

Apache Tomcat es el contenedor de Servlets usado en la Implementación de referencia de las tecnologías Java Servlet y JavaServer Pages.

Apache Tomcat es un contenedor de servlets maduro, confiable y eficiente respaldado por la Fundación Apache que permite desarrollar aplicaciones web basadas en el modelo MVC (Model View Controller) usando el lenguaje de programación Java, así como la amplia biblioteca de clases que éste proporciona. Esto permite desarrollar aplicaciones mucho más rápidamente. Además, al estar el código fuente de la aplicación basado en Java, puede ser ejecutado en cualquier otra plataforma que tenga soporte para Java. Entre ellas se encuentran Unix, Windows y GNU/Linux. Por lo tanto, al igual que la base de datos PostgreSQL, el código fuente del SCDD puede migrarse a otra plataforma en caso de ser necesario y conveniente, sin tener que cambiar nada en su programación. Esto es un beneficio del lenguaje Java utilizado. Pero Apache Tomcat está desarrollado también en Java, por lo que, si en algún momento se deseara convertir el servidor GNU/Linux a un servidor Windows, por ejemplo, el proceso sería tan sencillo como instalar Java para Windows, la versión de Tomcat para Windows y, una vez que estuviera funcionando correctamente Tomcat en su nuevo entorno, instalar el código fuente del SCDD dentro de Tomcat.

Nuevamente se ve que el SCDD ofrece libertad de elegir la plataforma y las herramientas que mejor se adapten al objetivo y al presupuesto de las dependencias.

El SCDD está desarrollado de manera que las suposiciones sobre el sistema operativo sobre el que está ejecutándose sean mínimas. Además están completamente aisladas, de manera que si se cambia de plataforma, es fácil modificarlas sin afectar las otras partes del SCDD.

Apache Tomcat, además, al estar basado en especificaciones publicadas, permite también cambiarlo por otro contenedor de servlets que esté basado en las mismas especificaciones. Por ejemplo, si se decidiera cambiar Tomcat™ por un Servidor J2EE™ tal como JBoss™, BEA™ Weblogic Application Server™, IBM™ Websphere Application Server™, Oracle9i™ Application Server™ o Sun™ iPlanet Application Server™, esto no sería mucho problema.

Como se puede ver, el SCDD proporciona flexibilidad y libertad de elegir tanto el Sistema operativo donde se ejecuta, como el contenedor web que lo contiene.

## 2. Lenguajes de Programación

Para el desarrollo del SCDD, así como para el anterior SCD se utilizó el lenguaje de programación Java.

La decisión estuvo basada en las siguientes razones:

- Es un lenguaje ampliamente conocido y maduro utilizado en una amplia gama de aplicaciones, la cuales se ejecutan en plataformas que van desde teléfonos celulares y dispositivos móviles, hasta servidores que dan servicio a cientos de usuarios concurrentes.
- Cuenta con una extensa y probada biblioteca de clases, las cuales facilitan el desarrollo de sistemas.

- El modelo de seguridad que el lenguaje proporciona es apto para su aprovechamiento en el desarrollo de aplicaciones cliente/servidor.
- Existe una estandarización de diferentes tecnologías basadas en Java, entre ellas Java Servlet y JavaServer Pages.
- Contamos con un creciente número de herramientas de desarrollo, tanto propietarias como de software libre, que permiten acortar tiempos de desarrollo y facilitar la creación de código fuente profesional y eficiente.

Además, para el desarrollo del SCDD se utilizó también JavaServer Pages y Java Servlets. Los beneficios de usar estas tecnologías se muestran a continuación:

- Existe una especificación de ambas tecnologías desarrollada a través de un proceso abierto en el que participa la comunidad. Por lo tanto, estas tecnologías resuelven problemas que surgen continuamente durante el desarrollo de aplicaciones web.
- Existe un gran número de productos que pueden ejecutar código fuente basado en estas tecnologías, por lo que al elegir las se evita el problema de depender de un solo proveedor, proporcionando libertad de elección a la SSA.
- Existe evidencia sólida de que el uso de estas tecnologías reduce el tiempo de desarrollo de un sistema web, por lo que disminuyen costos y permiten tener sistemas productivos en menor tiempo.

Al elegir las tecnologías que se utilizarían para desarrollar el SCDD, un objetivo que se tuvo siempre en mente fue la libertad de la SSA para cambiar los componentes que considere necesarios de acuerdo a sus necesidades.

### **3. Estándares internacionales**

Los lenguajes de programación, así como los servidores de bases de datos y contenedor web, son componentes estructurales del SCDD; son los que permiten que el sistema proporcione a los usuarios los servicios de administración y control de documentos. Pero existe otra característica que hace que el SCDD sea un sistema moderno, eficiente y confiable en la realización de estos objetivos: el uso de estándares internacionales para el almacenamiento y transmisión de información entre los nodos que conforman al SCDD.

Algunos de estos estándares ya se vieron en la sección anterior, donde se explicó la importancia del estándar SQL92 para el almacenamiento, acceso y búsqueda de la información documental existente en el SCDD. El seguimiento de este estándar por parte del sistema permite que el acceso a la información sea independiente de su almacenamiento. Es decir, el SCDD utiliza SQL para solicitarle a PostgreSQL que almacene o recupere cierta información específica. Esto es así en cualquier momento durante el funcionamiento del SCDD; pero, mientras que él de alguna manera conoce la información que desea almacenar o sabe las características de la información que consulta, no tiene ningún conocimiento, ni debe tenerlo, de la manera en que PostgreSQL almacena la información físicamente. Es decir, los papeles del sistema están completamente definidos y cada parte de él realiza su trabajo de la mejor manera y sin necesidad de conocer los detalles del trabajo sus demás partes. Pero para que esta suposición se mantenga, debe existir un lenguaje común con el que se comuniquen las partes del sistema, de manera que el SCDD cumpla su función. En este caso particular,

el estándar SQL92 realiza la función de lenguaje común entre PostgreSQL y el SCDD; o, dicho de otra manera, entre la lógica del sistema y el almacenamiento de la información que maneja el sistema.

Pero éste no es el único caso en el que el SCDD necesita de estándares para poder efectuar la comunicación entre sus partes conformantes. Entre los nodos del sistema debe existir también un lenguaje común que permita transmitir información entre ellos de manera precisa y sin ambigüedades. A diferencia de la interacción entre el SCDD y el medio de almacenamiento, donde existe ya un lenguaje común y una semántica definida para las posibles interacciones entre las partes, la interacción entre los nodos del SCDD es demasiado singular y orientada a la información de la SSA como para encontrar un estándar que nos permita expresar los distintos escenarios que pueden darse. En lugar de eso, se eligió un metalenguaje que nos permitiera crear otro más específico y adaptado a las necesidades del SCDD. De esta manera, cada uno de sus nodos puede comunicarse con los demás nodos de manera tal que no hay ambigüedad alguna en el significado de la información intercambiada.

El metalenguaje que se utilizó, además, es un estándar internacional en sí mismo, por lo que de esta manera se sigue cumpliendo el objetivo del SCDD respecto a la utilización de estándares. El metalenguaje del que hablamos es XML, el cual es definido y estandarizado por el World Wide Web Consortium, el cual, además, mantiene y regula estándares claves para el funcionamiento del Internet; entre ellos los estándares HTTP, HTML, SOAP y XSL.

XML es, pues, junto con algunas otras herramientas adicionales, usado para definir el lenguaje que utilizarán los nodos del SCDD para transmitir información entre sí.

### 3.1. XML

XML son la siglas para *EXtensible Markup Language*. XML es un formato de texto simple y sencillo derivado de SGML. XML juega un papel cada vez mayor en el intercambio de una amplia variedad de datos en la Web y en otros ambientes. Es un formato de texto muy similar al HTML, donde se utilizan etiquetas para delimitar los elementos de un documento. Por ejemplo, dentro del SCDD, un documento creado por un usuario tiene varios elementos que conforman la información del documento. Esta información, por ejemplo, se muestra en el siguiente fragmento:

```
<Documento cveUnidadRegistro="1234" cveDocumento="9999">
  <cveTipoDocumento>1</cveTipoDocumento>
  <fechaDocumento>2005-09-25T12:00:00.00</fechaDocumento>
  <tiempoVencimiento>5</fechaVencimiento>
  <cveUnidadTiempo>2</cveUnidadTiempo>
  ...
  <resumen>Solicitud de presupuesto</resumen>
  <palabrasClave>solicitud presupuesto</palabrasClave>
  <observaciones></observaciones>
  ...
  <Remitente cveUnidadRegistro="1234" cveRemitente="2222">
    <cveUnidadRegistroDocumento>1111</cveUnidadRegistroDocumento>
    <cveDocumento>9999</cveDocumento>
    ...
  </Remitente>
  ...
</Documento>
```

Este fragmento muestra muchos de los elementos por los que se eligió XML como lenguaje de definición del formato utilizado para transmitir la información entre nodos. Estas características se numeran a continuación:

1. La información está codificada en texto plano.
2. La información tiene estructura.
3. Cada elemento está delimitado por un par de etiquetas descriptivas de la información que contiene.
4. Hay elementos simples que contienen información y elementos compuestos que contienen a otros elementos simples.

XML permite que un emisor y un receptor puedan enviar y recibir información, siempre y cuando acuerden previamente la disposición de los elementos que conforman al fragmento XML

Una vez acordada la disposición de los elementos, cualquiera de las partes que se comunican está capacitada para:

- Enviar información:

Para el emisor es sencillo ordenar la información que desea enviar al receptor, de manera que se apegue a lo especificado previamente entre ellos. De esta manera, el conocer la disposición de los elementos permite al emisor construir un fragmento de datos XML tal que el receptor no tendrá ningún problema en entender.

- Recibir información:

Teniendo en cuenta la convención para la disposición de los datos en un fragmento XML, el receptor puede recuperar la información enviada por un emisor y utilizarla adecuadamente. De esta manera, el conocer la disposición del fragmento XML permite al receptor entender la información construida y enviada por el emisor.

- Valida un fragmento XML:

Comparar la disposición de los elementos de un fragmento XML recibido con la disposición de los mismos acordada previamente, permite al receptor verificar la validez del contenido de aquél. De esta manera, si la disposición de los elementos especifica que un elemento A debe contener a un elemento B seguido de un elemento C, y el fragmento XML no está estructurado de manera acorde, entonces el receptor puede desechar el fragmento completo, ya que no cumple con lo acordado previamente.

Debido a estas tres características de XML, los nodos del SCDD pueden enviar y recibir información continuamente y verificar que ésta cumpla con la estructura acordada previamente.

Como se puede ver, la convención sobre la disposición de los elementos en un fragmento XML es vital para el buen funcionamiento de este esquema. Si dos nodos tratan de entender cierto fragmento de XML utilizando dos convenciones distintas sobre la disposición de los elementos, la comunicación entre ellos será imposible. En cambio, si todos acuerdan utilizar la misma convención, la comunicación será fluida y precisa.

Por tanto, más que el XML en sí, lo verdaderamente vital para la correcta transmisión de información entre nodos es la convención sobre la disposición de los elementos de un fragmento XML. Esto, por supuesto, es parte importantísima al definir un formato XML para transmitir información. Por lo tanto, el World Wide Web Consortium creó un lenguaje para definir la disposición de los elementos en un fragmento XML de manera totalmente precisa y que pudiera ser utilizada por los participantes de una comunicación para crear, validar y entender el fragmento XML. Este metalenguaje para describir lenguajes basados en XML es llamado esquema. Hay varios tipos de esquemas para definir un documento XML, pero los tres más utilizados son:

- DTD (Document Type Definition)
- XML Schema
- Relax NG

Cada uno de estos esquemas permite definir estructuralmente un documento XML. Indica, además, restricciones tanto en la estructura como en el contenido de los documentos que se apegan a él. Un documento XML es considerado válido solamente si satisface los requerimientos del esquema con el que está asociado.

Cada tipo de esquemas tiene ventajas y desventajas, las cuales determinan su uso. Cronológicamente, los DTD son los primeros en aparecer. Debido a su gran complejidad, así como a las dificultades para expresar estructuras complejas, comenzaron a aparecer gradualmente nuevas maneras de representar un esquema. Actualmente, XML Schema es maduro, estable, y corrige muchas de las deficiencias existentes en el DTD. A diferencia de DTD, XML Schema está, a su vez, basado en XML. Relax NG es una adición más reciente a la lista de esquemas para XML y, al igual que XML Schema, está basado en XML. Aunque se encuentra en una fase temprana de su desarrollo, debido a su gran expresividad está sentando las bases para colocarse como un favorito en la lista de esquemas para XML.

Considerando el estado actual de madurez, así como la disponibilidad de herramientas para validación de documentos XML, se eligió XML Schema como el mecanismo de descripción para las transmisiones XML entre los nodos que conforman al SCDD.

## **3.2. XML Schema**

El propósito de un XML Schema es definir los elementos legales de un documento XML. Un XML Schema define:

- Los elementos que pueden aparecer en un documento.
- Los atributos que puede tener cada elemento.
- El orden en que pueden aparecer los elementos de un documento.
- Qué elementos pueden ser hijos de otros elementos.
- El número de hijos que puede tener un elemento.



- El tipo de datos que puede contener cada elemento.

Debido a estas propiedades del XML Schema, un documento XML puede ser entendido, validado y procesado correctamente por el emisor o el receptor de un documento XML.

En el caso específico del SCDD se creó un XML Schema para definir las transmisiones entre los distintos nodos del sistema. Este XML Schema define los elementos que puede contener un documento XML usado para representar la información documental del SCDD.

En el siguiente fragmento de código se muestra un detalle del esquema creado para el SCDD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.ssa.gob.mx/sync"
  xmlns="http://www.ssa.gob.mx/sync"
  elementFormDefault="qualified">
  ...
  <!-- Definición de documento -->
  <xsd:complexType name="documento">
    <xsd:sequence>
      <xsd:element name="cveTipoDocumento" type="xsd:long"/>
      <xsd:element name="fechaDocumento" type="xsd:dateTime"/>
      <xsd:element name="tiempoVencimiento" type="xsd:long"/>
      <xsd:element name="cveUnidadTiempo" type="xsd:long"/>
      ...
      <xsd:element name="resumen" type="xsd:string"/>
      <xsd:element name="palabrasClave" type="xsd:string"/>
      <xsd:element name="observaciones" type="xsd:string"
minOccurs="0"/>
      ...
      <xsd:element name="Remitente" type="remitente"/>
      ...
    </xsd:sequence>
    <xsd:attribute name="cveUnidadRegistro" type="xsd:long"
use="required"/>
    <xsd:attribute name="cveDocumento" type="xsd:long"
use="required"/>
  </xsd:complexType>
  ...
</xsd:schema>
```

Hay varios puntos a notar de este fragmento.

- El XML Schema define un *namespace*.

La propiedad `targetNamespace` del elemento `xsd:schema` define el *namespace* al cual pertenece el esquema del SCDD. En este caso pertenece a `http://www.ssa.gob.mx/sync`. Dentro de este *namespace* se pueden crear todos los XML Schema que sean necesarios sin temor a causar conflictos con otros *namespaces* que definan elementos con el mismo nombre. Actualmente solamente hay un XML Schema y es el utilizado por el SCDD para definir los

elementos válidos en un documento XML utilizado para transmitir información documental entre los nodos del sistema.

- El XML Schema define los elementos permitidos en el documento XML.

Como se ejemplifica con el elemento 'documento', el XML Schema define los elementos permitidos con total precisión. En este caso, por ejemplo, un elemento 'documento' puede tener como elementos hijos a 'cveTipoDocumento', 'fechaDocumento', 'resumen', por nombrar algunos. Define, además, el tipo de datos que cada uno de ellos puede contener. Por ejemplo, un elemento 'cveTipoDocumento' será válido solamente si contienen una cadena que represente a un entero. De la misma manera, el elemento 'resumen' puede contener una cadena de texto de cualquier longitud.

- El XML Schema define el orden en que los elementos pueden aparecer.

Usando el mismo elemento 'documento' como ejemplo, se puede notar que los elementos hijos, esto es, 'cveTipoDocumento', 'fechaDocumento', etc.; están contenidos dentro del elemento 'sequence' del XML Schema. De esta manera, si un documento XML indica que el XML Schema que lo describe es el mostrado en el ejemplo, y este documento XML contiene un elemento 'documento' en él, entonces el elemento 'documento' debe contener un elemento 'cveTipoDocumento', seguido de un elemento 'fechaDocumento', seguido de un elemento 'tiempoVencimiento', etc. y así hasta completar todos los elementos especificados por el XML Schema.

De esta manera, si en un futuro fuera necesario comunicar un nuevo componente con los nodos existentes en el SCDD, o si fuera necesario acceder a la información que se transmite entre los nodos, el esquema que describe los elementos permitidos en el documento XML usado durante las transmisiones entre nodos es todo lo que se necesitaría para poder validar la información y accederla.

## **4. Algoritmos de replicación**

Ya se vió en las secciones anteriores la importancia de los distintos componentes para el correcto funcionamiento del SCDD. En esta sección hablaremos de los algoritmos utilizados dentro del SCDD para realizar las tareas de replicación y sincronización de la información documental.

Los algoritmos cumplen con los objetivos planteados para el sistema de replicación utilizado por el SCDD para transferir información entre nodos.

Los algoritmos se muestran en las siguientes secciones.

### **4.1. Envío de información entre un Nodo Local y otro Nodo del SCDD**

Este algoritmo define el comportamiento del SCDD durante el envío de información entre un Nodo Local de una dependencia y otro nodo del SCDD, sea éste el Nodo Central o algún otro Nodo Local.

```
1. Obtiene la localización del Servidor Central y de los
   Servidores Locales de las otras dependencias que conforman
   al SCDD (dependencias)
2. Calcula número de documentos a enviarse al Servidor
   Central (numDocsCentral)
3. Calcula número de respuestas a documentos a enviarse al
   Servidor Central (numRespuestasCentral)
4. if ( numDocsCentral + numRespuestasCentral > 0 ) then
   enviaDatos(central)
   endif
5. for ( i in dependencias ) do
   Calcula número de documentos a enviarse a Servidor Local de
   la dependencia i, (numDocsDestino)
   Calcula número de respuestas a documentos a enviarse al
   Servidor Local de la dependencia i, (numRespuestasDestino)
   if ( numDocsDestino + numRespuestasDestino > 0 ) then
   enviaDatos(i)
   endif
done
```

En este pseudocódigo, **dependencias** es el conjunto de Servidores Locales de las dependencias que conforman el SCDD. **numDocsCentral** y **numDocsDestino** son el número de documentos candidatos a enviarse al Servidor Central y a un Servidor Local de una dependencia, respectivamente. Similarmente, **numRespuestasCentral** y **numRespuestasDestino** son el conjunto de respuestas a documentos candidatas a enviarse al Servidor Central y a un Servidor Local, respectivamente.

## 4.2. Envío de información a un Nodo del SCDD

El siguiente algoritmo detalla el funcionamiento del método `enviarDatos()` utilizado en el algoritmo anterior. Este método es el encargado de transportar la información entre un nodos del SCDD.

```
enviarDatos(destino)
1. Establecemos una conexión http hacia servidor destino.
2. while ( (doc in docsDestino) <> null ) do
   enviarStatusDocumentos(doc)
   enviarRemitentes(doc)
   enviarDestinatarios(doc)
   enviarAnexos(doc)
   marcarEnvioDocumento(doc)
done
3. while ( (resp in respuestasDocumento) <> null ) do
   enviarStatusRespuesta(resp)
   enviarAnexosRespuestaDocumento(resp)
   marcarEnvioRespuestaDocumento(resp)
done
4. Obtenemos código de repuesta del servidor destino
5. Si el código de respuesta del servidor fue correcto, entonces
   se hacen permanentes los cambios a la base de datos (los
   marcadores de envío de documentos y respuestas), de lo contrario,
```

los cambios efectuados a la base de datos son descartados y la base de datos queda inalterada.  
6. Cerramos la conexión a **destino**

Donde **destino** es ya sea el Servidor Central de SCDD o algún Servidor Local de una dependencia del SCDD. **docsDestino** es el conjunto de documentos disponibles para envío al servidor **destino**.

### 4.3. Recepción de información de un Nodo del SCDD

Este algoritmo es utilizado por un Nodo del SCDD para recibir la información documental de algún Nodo Local.

1. Se acepta una conexión proveniente del Servidor Local de alguna dependencia.
2. Se inicia una transacción en la base de datos.
3. while (*existan datos por transmitir*) do  
    **tipo** = obtenTipoDato()

```
case tipo in
  'documento' =>
    do
      obtenDatosDocumento()
      insertaDatosDocumento()
    done;
  'status_documento' =>
    do
      obtenStatusDocumento()
      insertaStatusDocumento()
    done;
  'remitente_documento' =>
    do
      obtenRemitenteDocumento()
      insertaRemitenteDocumento()
    done;
  'destinatario_documento' =>
    do
      obtenDestinatarioDocumento()
      insertaDestinatarioDocumento()
    done;
  'anexo_documento' =>
    do
      obtenAnexoDocumento()
      insertaAnexoDocumento()
    done;
  'respuesta_documento' =>
    do
      obtenRespuestaDocumento()
      insertaRespuestaDocumento()
    done;
  'anexo_respuesta_documento' =>
    do
      obtenAnexoRespuestaDocumento()
      insertaAnexoRespuestaDocumento()
    done;
endcase
done
```

4. Si la conexión terminó de manera exitosa, entonces hacemos

```
permanentes los cambios a la base de datos, de lo contrario,
revertimos los cambios hechos de manera que la base quede
inalterada.
5. Terminamos la conexión con el Servidor Local
```

Donde los métodos **obtenXXXXDocumento()** e **insertarXXXXDocumento()** son funciones auxiliares que nos permiten obtener e insertar toda la información relacionada con **XXXXX** del flujo de datos transmitidos. El método **obtenTipoDato()** obtiene del flujo de datos la información necesaria para saber que clase de datos son los que se recibirán a continuación.

Estos son los principales algoritmos del SCDD. Ellos brindan al SCDD la capacidad de replicar la información documental de manera confiable y mantienen la integridad de la información.

## 5. Migración del SCD al SCDD

La migración del SCD al SCDD debe realizarse de manera gradual, integrando la información contenida actualmente en el servidor del SCD que tiene la dependencia.

Para una dependencia que ya cuenta con un servidor SCD, éste deberá convertirse en un Nodo Local del SCDD. Por lo tanto, ahora la dependencia contará con un Servidor Local y una Base de Datos Local. Una vez instalado el Servidor Local y configurado el acceso a la Base de Datos Local, se puede proceder a la migración. La parte más importante es la transformación de la información de la base de datos del SCD al formato usado por la Base de Datos Local del SCDD. Esto se puede hacer de manera automatizada si se crea un script que realice las siguientes operaciones:

1. Conectarse a la base de datos del SCD de la dependencia y extraer toda la información documental contenida en ella.
2. Con esta misma información, llenar las tablas de la Base de Datos Local, agregando la información necesaria para convertir los datos del SCD a datos que el SCDD pueda utilizar.

El segundo paso por sí mismo no debe implicar pérdida de información. Tal y como se ha comentado a lo largo del texto, el SCDD es un superconjunto de la funcionalidad ofrecida por el SCD. Por tanto, toda la información almacenada en el SCD es utilizada y almacenada también en el SCDD. La diferencia es que se agregará información en varios puntos, de manera que la información documental tenga referencias al Nodo Local, tanto del remitente como del destinatario de los documentos y respuestas de documentos, de manera que el subsistema de replicación del SCDD tenga los elementos necesarios para poder transmitirlo al Nodo Local correspondiente.

Una vez migrada toda la información del Servidor SCD al Nodo Local de la dependencia, se procederá a hacer la replicación de información entre el Nodo Local y el resto del SCDD, de manera que el Nodo Local esté completamente sincronizado.

Ya terminada la replicación, la información que una vez estuvo solamente en el servidor SCD, ahora estará tanto en el nuevo Nodo Local de la dependencia, como en el Nodo Central del SCDD.

Además, si hay algún documento dirigido a alguna persona asignada a otro Nodo Local del SCDD, este documento habrá sido también transmitido al Nodo Local del destinatario del documento. De la misma manera, si en los otros Nodos Locales del SCDD existen documentos dirigidos a personas de la dependencia, éstos serán replicados adecuadamente en el momento en que los Nodos Locales de los remitentes sincronicen su información con el SCDD.

El resultado final es que el Nodo Local de la dependencia que contaba con el SCD estará completamente integrado al SCDD y funcionará de manera coordinada con los demás nodos del sistema.

Suponiendo que este proceso de migración se realizara durante un período en que no afecte a los usuarios del SCD, por ejemplo, durante un fin de semana, los usuarios no notarían que el sistema que usan es ahora parte de un sistema distribuido de nodos cooperando, en lugar de un solo servidor independiente y aislado de aquellos de las demás dependencias.

Esa es una de las consecuencias de la arquitectura creada para el SCDD.

# Capítulo 5. Operación del SCDD

En este capítulo se muestra en funcionamiento del SCDD en distintos escenarios posibles, así como también los resultados de las pruebas a las que se sometió al SCDD a fin de medir su rendimiento con diferentes cargas de trabajo.

## 1. Uso del SCDD

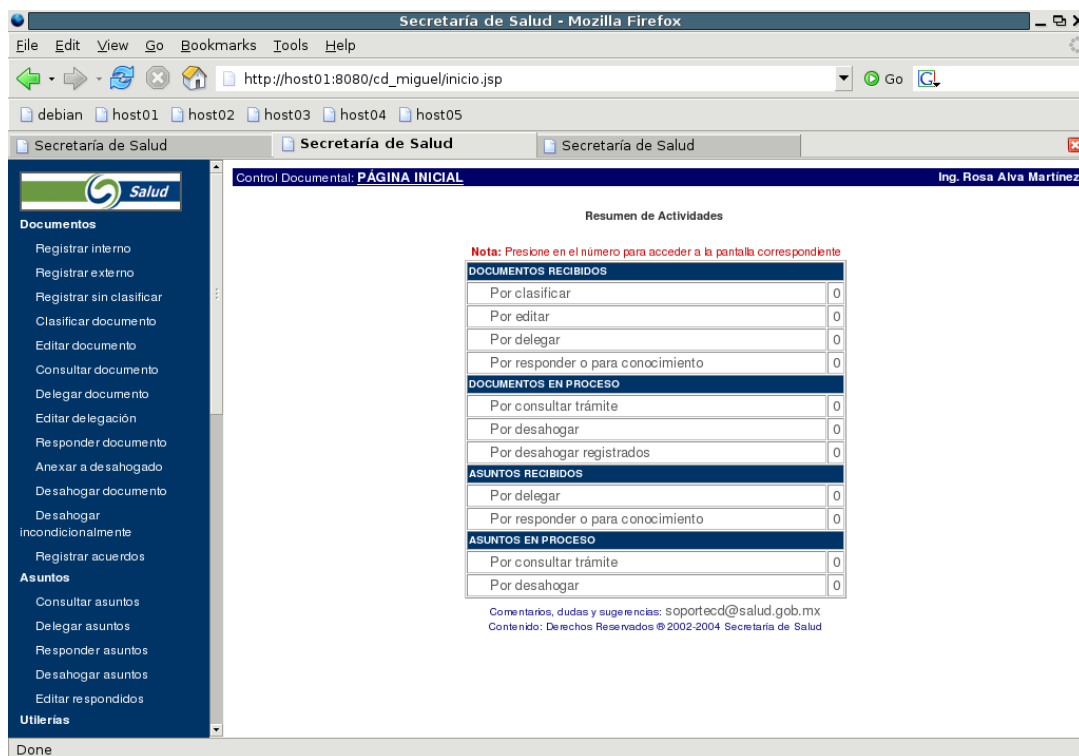
A continuación se muestra un ejemplo del uso típico que se le da al Sistema de Control Documental Distribuido.

El escenario es el siguiente. Existen dos dependencias, cada una con su propio Nodo Local. En la primera dependencia, llamada Dependencia 01, labora Alva. El nombre del servidor del Nodo Local de esta dependencia es *host01*. En la segunda dependencia, llamada Dependencia 02, trabaja Dante. El nombre del servidor del Nodo Local de la segunda dependencia es *host02*.

Alva debe avisar oficialmente a Dante de la fecha en la cual éste deberá presentar su informe anual de actividades. Para ello hace uso del SCDD.

Por tanto, Alva ingresa al servidor de su Nodo Local, llamado *host01*, con su usuario y password asignados. En ese momento, Alva no tiene ningún documento pendiente.

Figura 5.1. Página inicial de Alva en host01 (Dependencia 01)



A continuación, ella ingresa a la opción *Registrar Interno* del menú *Documentos* y captura la información necesaria para que Dante conozca la fecha y lugar donde deberá presentar su informe.

**Figura 5.2. Alva captura el documento para Dante**

Al terminar de ingresar la información del documento, Alva pasa a la sección de captura de anexos para el documento.

**Figura 5.3. Captura de anexos para el documento de Dante**

No. Doc.	No Oficio	Fecha del Documento (dd/mm/aaaa)	Fecha de Registro (dd/mm/aaaa hh:mm)	Fecha de Vencimiento (dd/mm/aaaa hh:mm)
157-1	ssa/dgti/2005-123	20/11/2005	20/11/2005 21:24	22/11/2005 21:24

**Registrado por:** Ing. Rosa Alva Martinez  
**Remitente:** Ing. Rosa Alva Martinez  
**Destinatario:** Ing. Dante Almonte Garcia

**Status del documento:** En registro

**Clasificación/Subclasificación:** Comunicación Social/Compereencias ante el Poder Ejecutivo

**Tipo:** Oficio

**Descripción del documento:** Se le informa que debera presentarse sin falta el día 25 de enero de 2006 en la oficina del Secretario para rendir su informe anual de trabajo. La cita es en Amadeo Betancourt 250 a las 1630 horas.

**Palabras clave:** informe, anual

**Observaciones:** ninguna

**Copias para conocimiento:** Ninguna

**Anexos del documento.**  
No existen anexos para este documento.

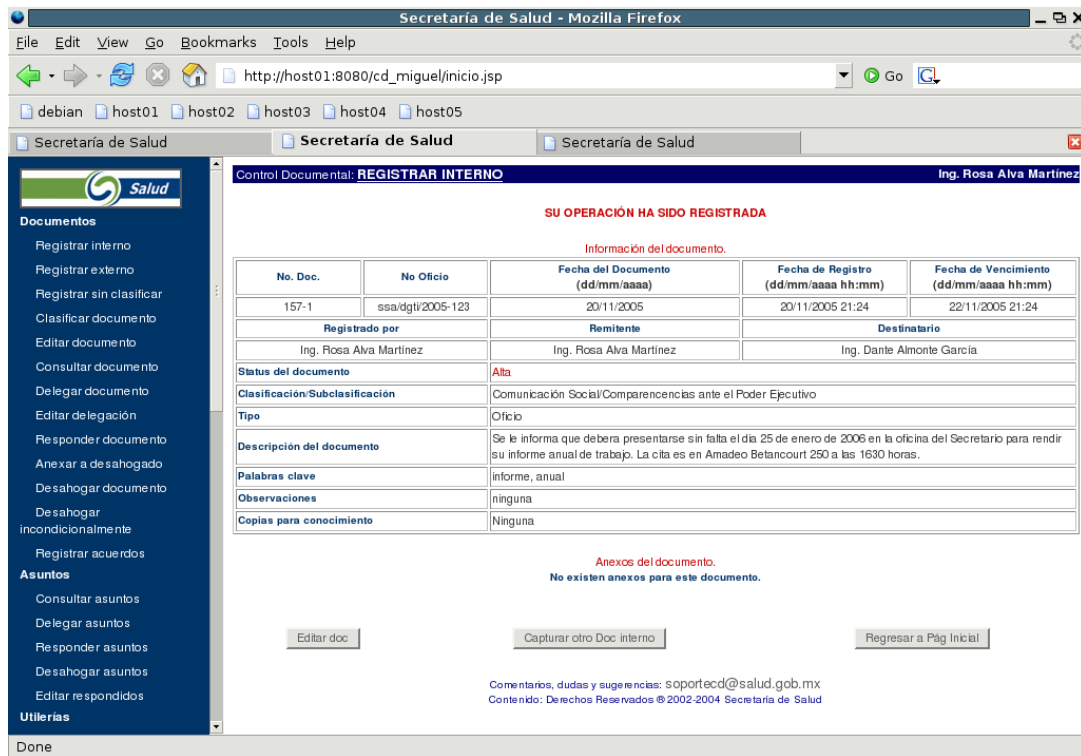
**Nota:** Los campos con asterisco(\*) son requeridos.

**Tipo de anexo\*:** Seleccione un tipo de anexo...



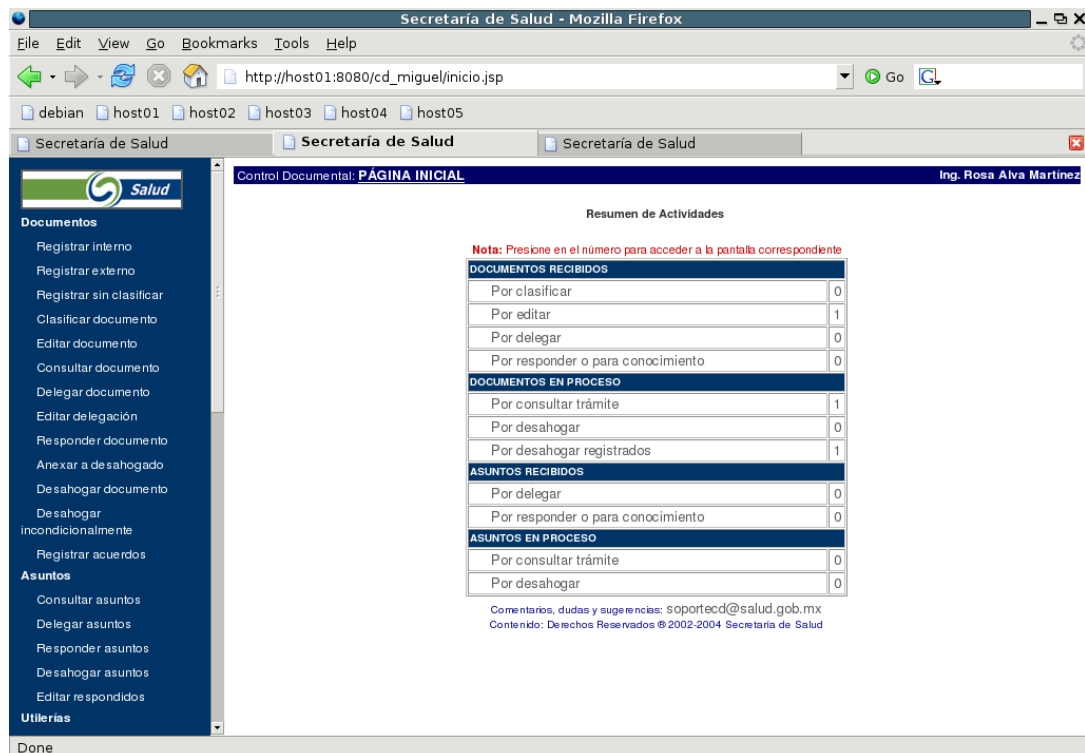
Como el documento no necesita tener anexos Alva da fin a la captura del documento. El documento, por consiguiente, es dado de alta en el SCDD.

**Figura 5.4. Fin del registro del documento para Dante**



Alva puede ver, en su pantalla inicial, que el documento ya está en la lista de documentos pendientes de consultar trámite. Cuando el documento tenga respuesta, podrá consultar ahí cuál fue ésta.

**Figura 5.5. Documentos por consultar para Alva en host01**

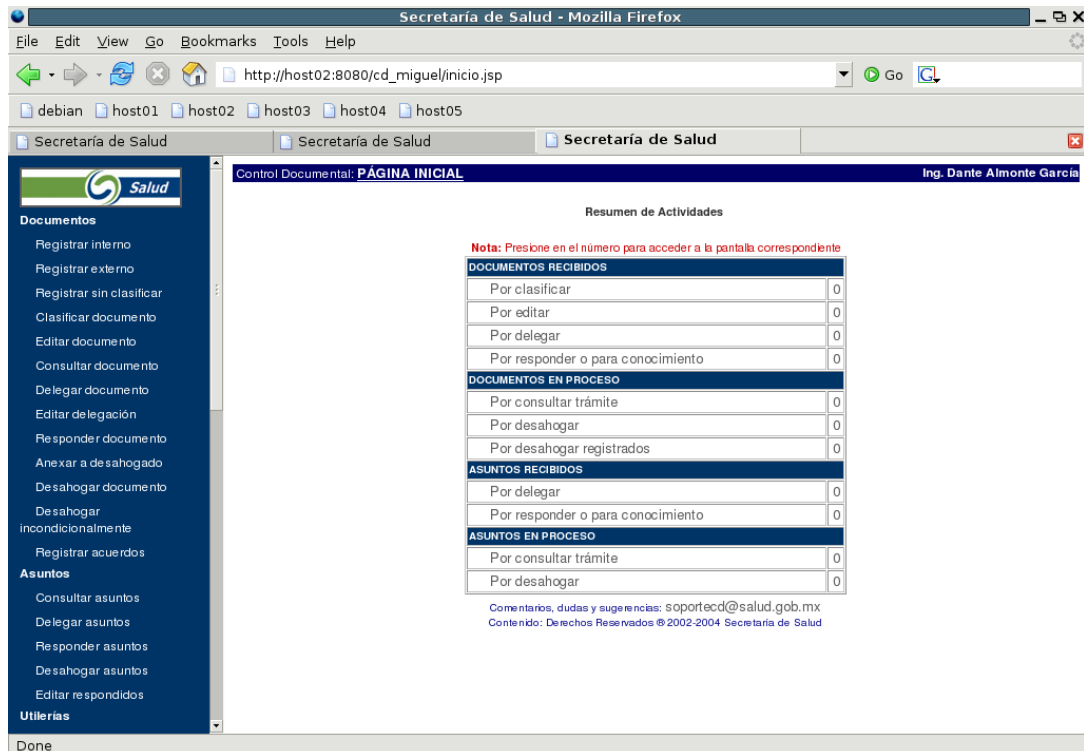


Supongamos que el periodo de replicación configurado en el Nodo Local de la dependencia donde labora Alva está configurado a 1 hora. Esto significa que, a intervalos de una hora, el Nodo Local de la Dependencia 01 se conectará al Nodo Central y a los otros Nodos Locales para enviar y recibir toda la información documental que se haya generado.

Si Alva, por ejemplo, terminó de capturar el documento a las 11:15 am y el Nodo Local de su dependencia replicará su información hasta las 12:00 pm, Dante no sabrá nada del documento que Alva le envió hasta después de las 12:00 pm.

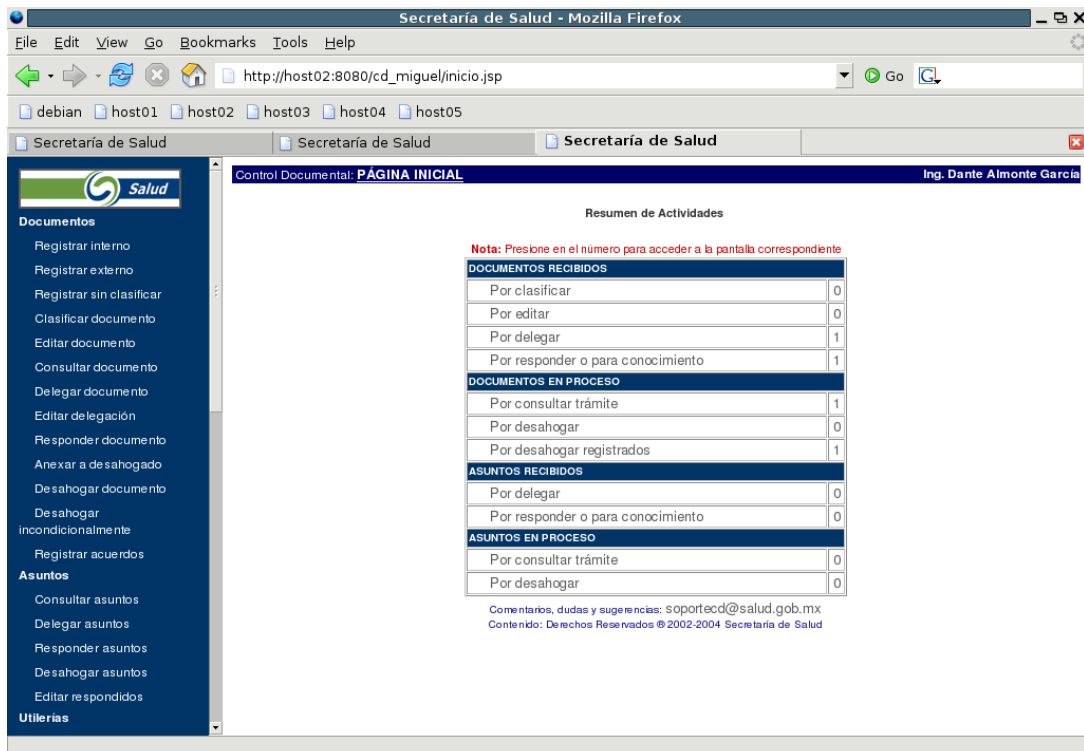
Si Dante ingresa al servidor del Nodo Local de su dependencia antes de las 12:00 pm, verá una pantalla como la siguiente

**Figura 5.6. Página inicial de Dante en host02 (Dependencia 02) antes de la replicación de datos.**



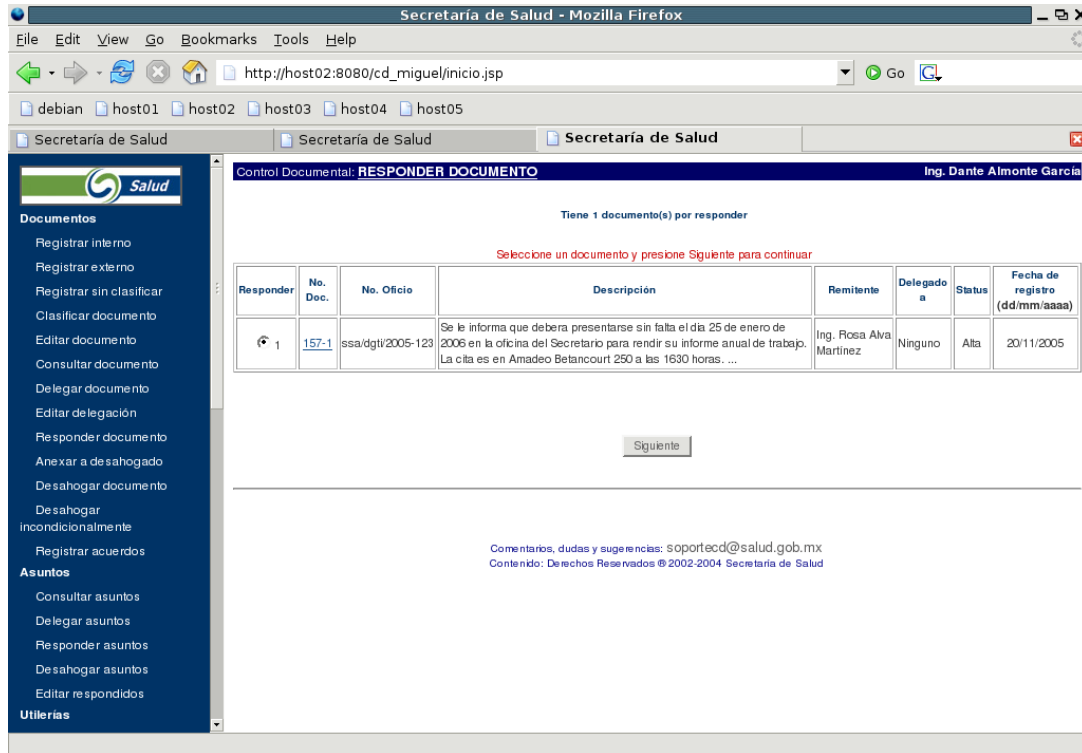
Después de las 12:00 pm, Dante revisa si tiene algún documento pendiente. Como ya se efectuó la replicación de los datos, Dante encuentra que tiene un documento pendiente por responder.

**Figura 5.7. Página inicial de Dante en host02 (Dependencia 02) después de la replicación de datos.**



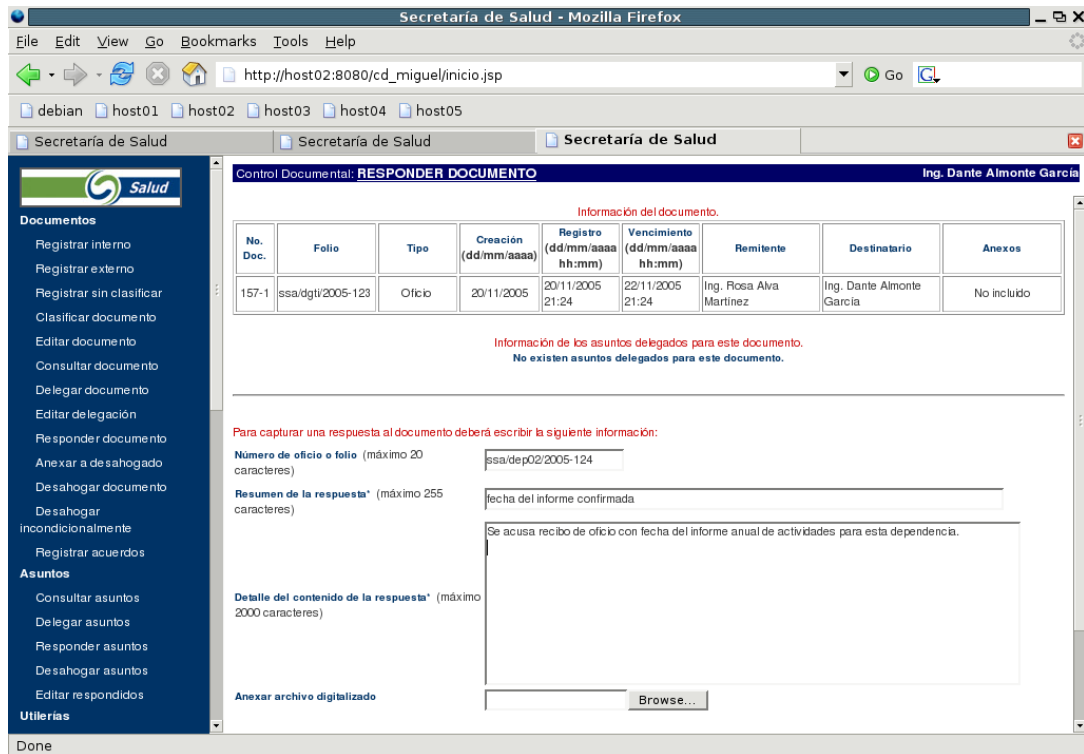
En consecuencia, elige la opción Responder Documento del menú Documentos. Una vez en esa pantalla, puede ver la lista de documentos que tiene pendientes de respuesta. En este caso solamente tiene uno, cuyo remitente es Alva.

**Figura 5.8. Consulta de documentos por responder**



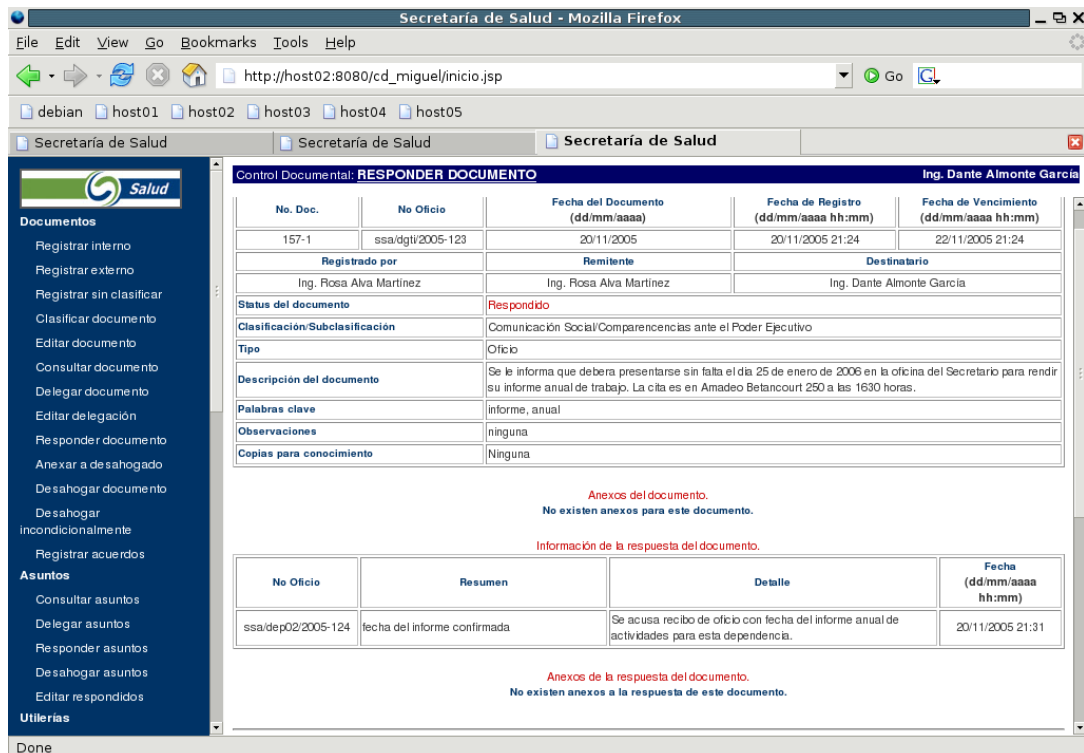
A continuación, Dante lo selecciona y comienza a redactar la respuesta que dará al documento enviado por Alva.

**Figura 5.9. Dante captura la respuesta al documento enviado por Alva**



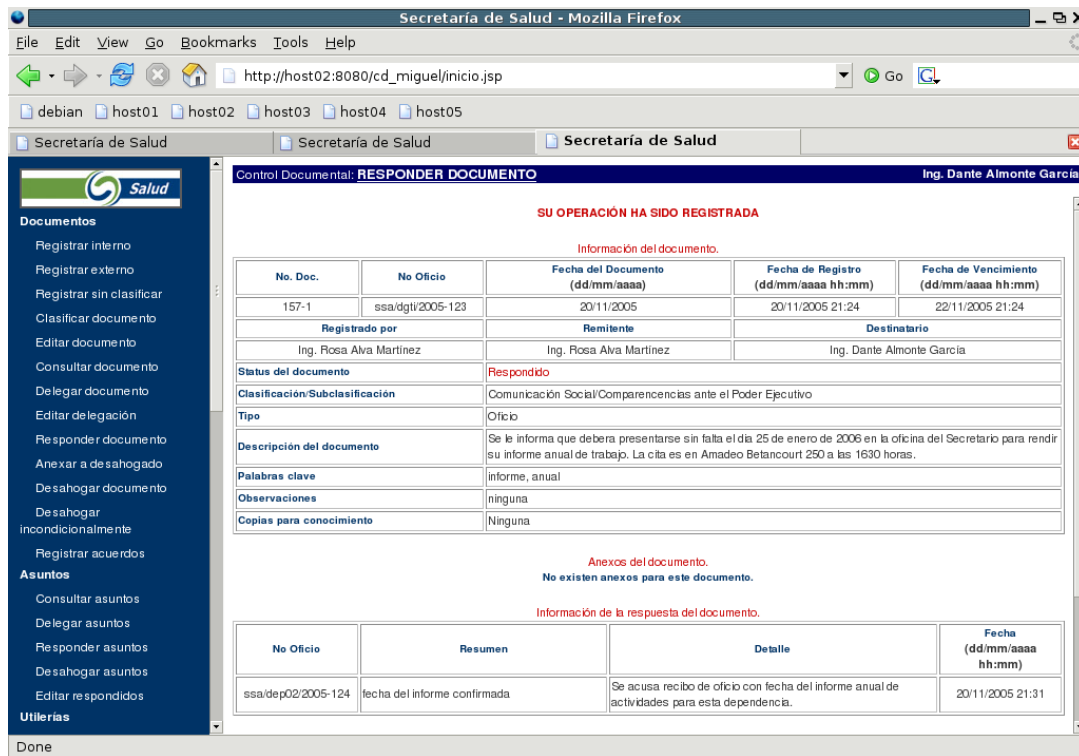
Siguiendo con el proceso de captura de respuestas a documentos, Dante llega a la sección de los anexos para la respuesta del documento. Como en este caso no es necesario agregar anexos a la respuesta, Dante termina de ingresar la respuesta del documento.

**Figura 5.10. Captura de anexos para la respuesta al documento**



Finalmente, el SCDD le indica a Dante que la respuesta que redactó para el documento recibido está dado de alta. Con esto, el documento cambia su estado a respondido.

**Figura 5.11. Fin del registro de la respuesta al documento enviado por Alva**



De nuevo en la pantalla inicial, Dante observa que ya no tiene documentos por responder. En cambio, el documento que acaba de responder, ahora está en la lista de documentos por desahogar. Una vez que el documento esté desahogado, la respuesta ya no se podrá modificar y por tanto será la repuesta final al documento que le dio origen. Cuando se ejecute la siguiente replicación de datos del Nodo Local de la dependencia donde labora Dante, la respuesta recién ingresada se hará llegar a su remitente, en este caso Alva.

**Figura 5.12. Documentos por desahogar para Dante**

Control Documental: **PÁGINA INICIAL** Ing. Dante Almonte Garcia

Resumen de Actividades

**Nota: Presione en el número para acceder a la pantalla correspondiente**

DOCUMENTOS RECIBIDOS	
Por clasificar	0
Por editar	0
Por delegar	0
<b>Por responder o para conocimiento</b>	<b>0</b>

DOCUMENTOS EN PROCESO	
Por consultar trámite	1
Por desahogar	1
Por desahogar registrados	0

ASUNTOS RECIBIDOS	
Por delegar	0
Por responder o para conocimiento	0

ASUNTOS EN PROCESO	
Por consultar trámite	0
Por desahogar	0

Comentarios, dudas y sugerencias: [soportecd@salud.gob.mx](mailto:soportecd@salud.gob.mx)  
 Contenido: De derechos Reservados © 2002-2004 Secretaría de Salud

En consecuencia, Dante desahoga el documento, para indicarle al SCDD que ya puede replicar la respuesta que ingresó.

**Figura 5.13. Dante desahoga el documento para el que registró una respuesta**

Control Documental: **DESAHOGAR DOCUMENTO** Ing. Dante Almonte Garcia

**SU OPERACIÓN HA SIDO REGISTRADA**

Información del documento.

No. Dec.	No Oficio	Fecha del Documento (dd/mm/aaaa)	Fecha de Registro (dd/mm/aaaa hh:mm)	Fecha de Vencimiento (dd/mm/aaaa hh:mm)
157-1	ssa/dgti/2005-123	20/11/2005	20/11/2005 21:24	22/11/2005 21:24

Registrado por	Remitente	Destinatario
Ing. Rosa Alva Martinez	Ing. Rosa Alva Martinez	Ing. Dante Almonte Garcia

Status del documento: **Desahogado**

Clasificación/Subclasificación: Comunicación Social/Comparecencias ante el Poder Ejecutivo

Tipo: Oficio

Descripción del documento: Se le informa que debiera presentarse sin falta el día 25 de enero de 2006 en la oficina del Secretario para rendir su informe anual de trabajo. La cita es en Amadeo Betancourt 250 a las 1630 horas.

Palabras clave: informe, anual

Observaciones: ninguna

Copias para conocimiento: Ninguna

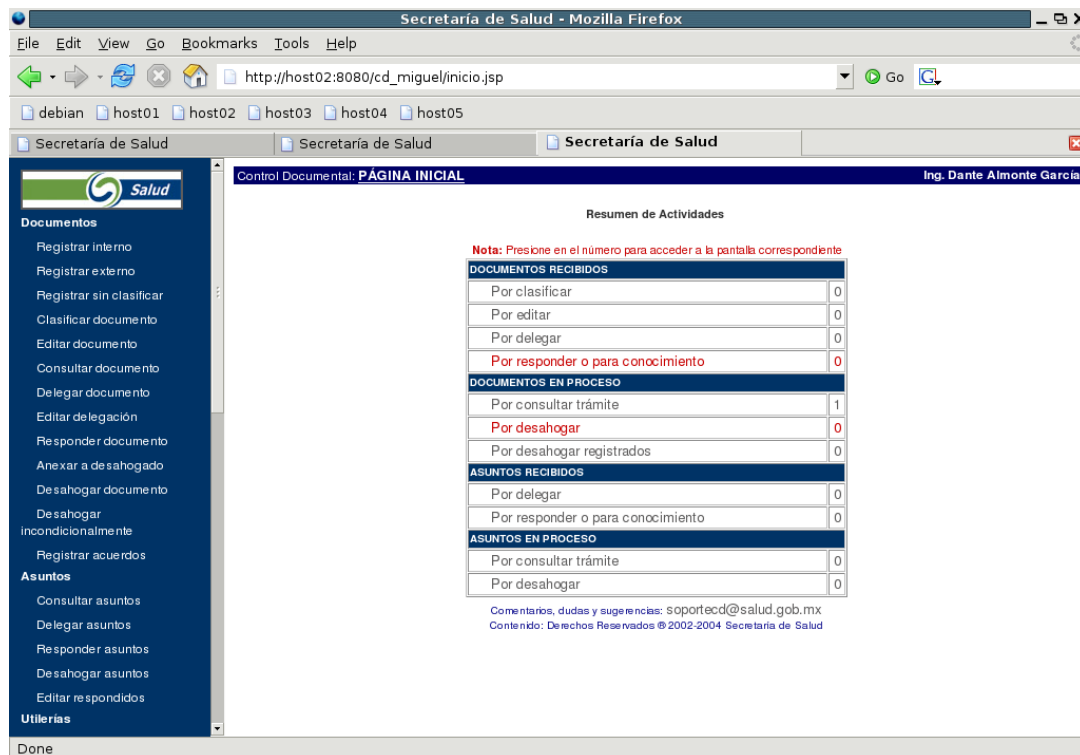
Anexos del documento.  
No existen anexos para este documento.

Información de la respuesta del documento.

No Oficio	Resumen	Detalle	Fecha (dd/mm/aaaa hh:mm)
ssa/dep02/2005-124	fecha del informe confirmada	Se acusa recibo de oficio con fecha del informe anual de actividades para esta dependencia.	20/11/2005 21:31

Finalmente, Dante verifica en su pantalla inicial que ya no queda ningún documento por responder ni por desahogar.

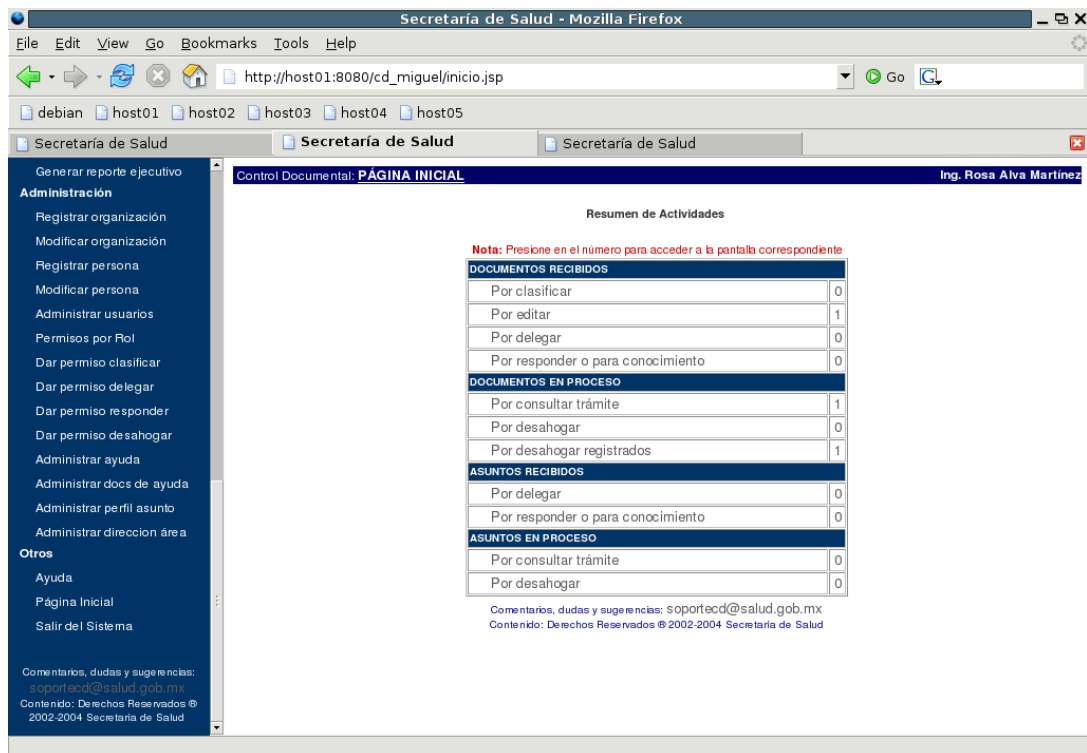
**Figura 5.14. Página inicial de Dante, sin documentos por responder o por desahogar.**



Supongamos que Dante escribió la respuesta al documento de Alva a las 12:45 pm, que el Nodo Local de su dependencia (*host02*) está configurado para replicar su información cada media hora y que la siguiente replicación será a las 1 pm. Con estas condiciones, si Alva revisa su pantalla inicial antes de la 1 pm, ella verá una pantalla como la siguiente, ya que la replicación de la información aún no ha sucedido.

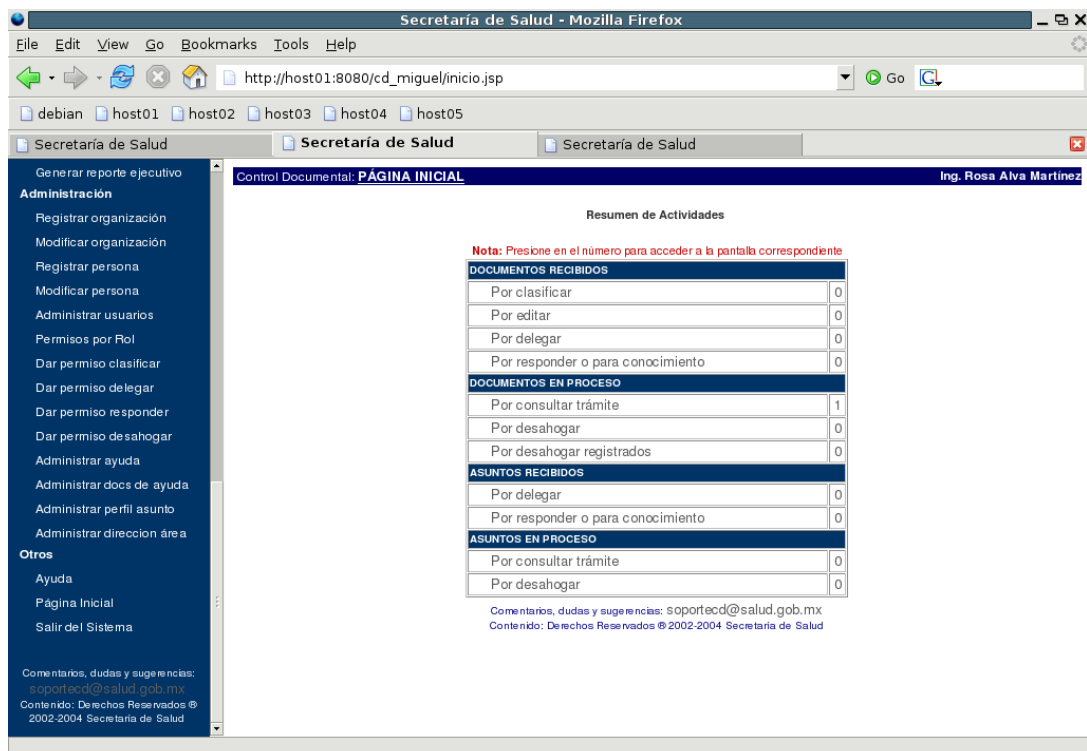
**Figura 5.15. Página inicial de Alva en host01 (Dependencia 01) antes de la replicación de datos.**





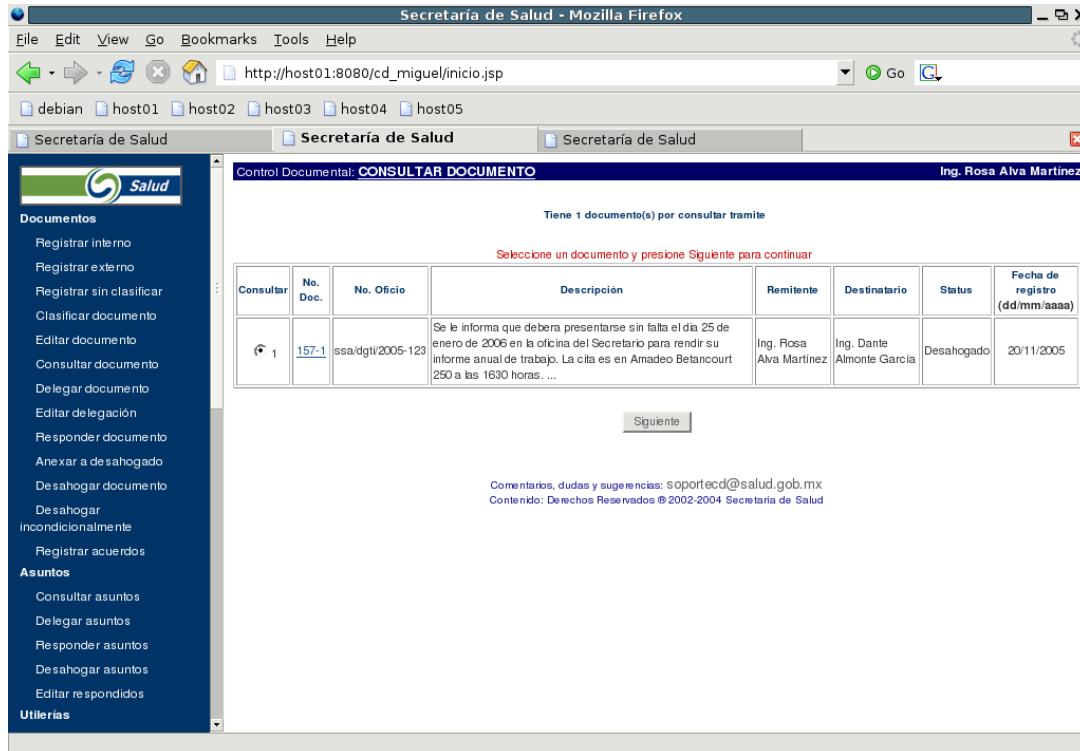
Después de la 1 pm, Alva revisa nuevamente su pantalla inicial. Como para ese momento los datos ya fueron replicados, Alva puede consultar el estado del documento ingresando a la opción *Consultar Documento* del menú *Documentos*.

**Figura 5.16. Página inicial de Alva en host01 (Dependencia 01) después de la replicación de datos.**



En la pantalla de consulta de documentos, Alva puede ver que el documento que envió a Dante, ya cambió al estado de Desahogado, indicando con esto que ya tiene una respuesta definitiva.

**Figura 5.17. Consulta de documentos para Alva**



Al seleccionar el documento, Alva puede ver los detalles de la respuesta que Dante registró.

**Figura 5.18. Detalle del documento respondido**

Control Documental: **CONSULTAR DOCUMENTO** Ing. Rosa Alva Martínez

No. Doc.	No Oficio	Fecha del Documento (dd/mm/aaaa)	Fecha de Registro (dd/mm/aaaa hh:mm)	Fecha de Vencimiento (dd/mm/aaaa hh:mm)
157-1	ssa/dgltv/2005-123	20/11/2005	20/11/2005 21:24	22/11/2005 21:24

Registrado por	Remitente	Destinatario
Ing. Rosa Alva Martínez	Ing. Rosa Alva Martínez	Ing. Dante Almonte García

**Status del documento:** Desahogado

**Clasificación/Subclasificación:** Comunicación Social/Comparencias ante el Poder Ejecutivo

**Tipo:** Oficio

**Descripción del documento:** Se le informa que deberá presentarse sin falta el día 25 de enero de 2006 en la oficina del Secretario para rendir su informe anual de trabajo. La cita es en Amadeo Betancourt 250 a las 1630 horas.

**Palabras clave:** informe, anual

**Observaciones:** ninguna

**Copias para conocimiento:** Ninguna

**Anexos del documento.**  
No existen anexos para este documento.

**Información de los asuntos delegados para este documento.**  
No existen asuntos delegados para este documento.

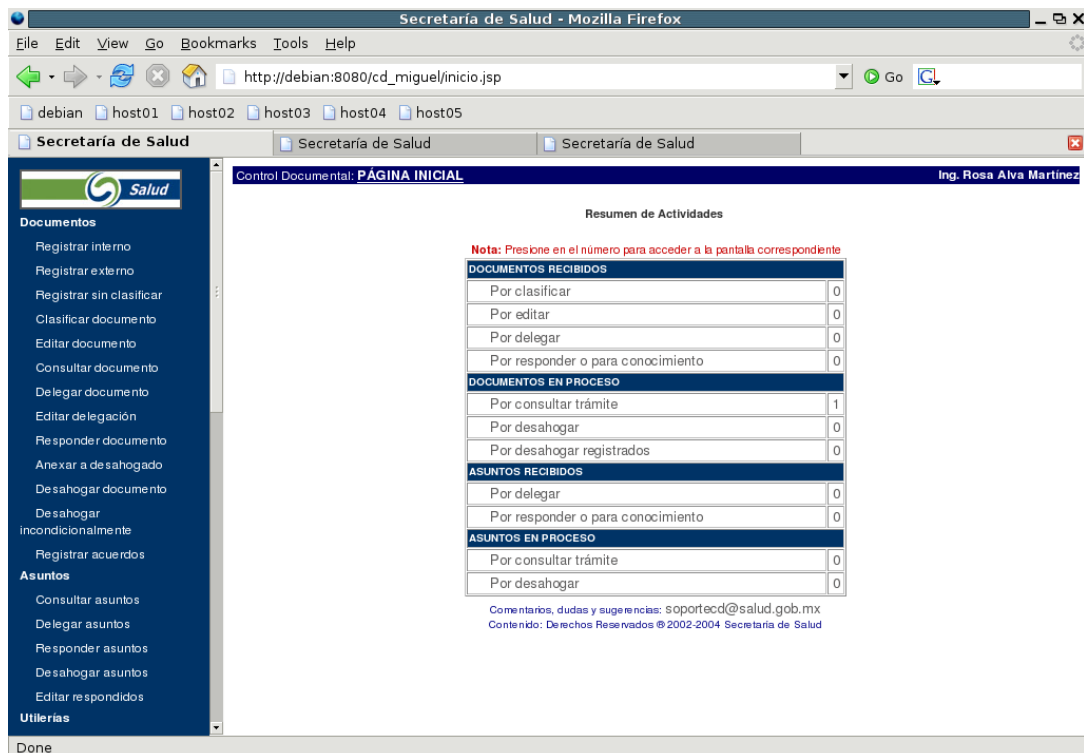
**Información de la respuesta del documento.**

No Oficio	Resumen	Detalle	Fecha (dd/mm/aaaa hh:mm)
ssa/dep02/2005-124	fecha del informe confirmada	Se acusa recibo de oficio con fecha del informe anual de actividades para esta dependencia.	20/11/2005 21:31

De esta manera, y de forma totalmente transparente para el usuario final, el SCDD se encarga de transportar la información de manera correcta y eficiente entre los Nodos Locales adecuados.

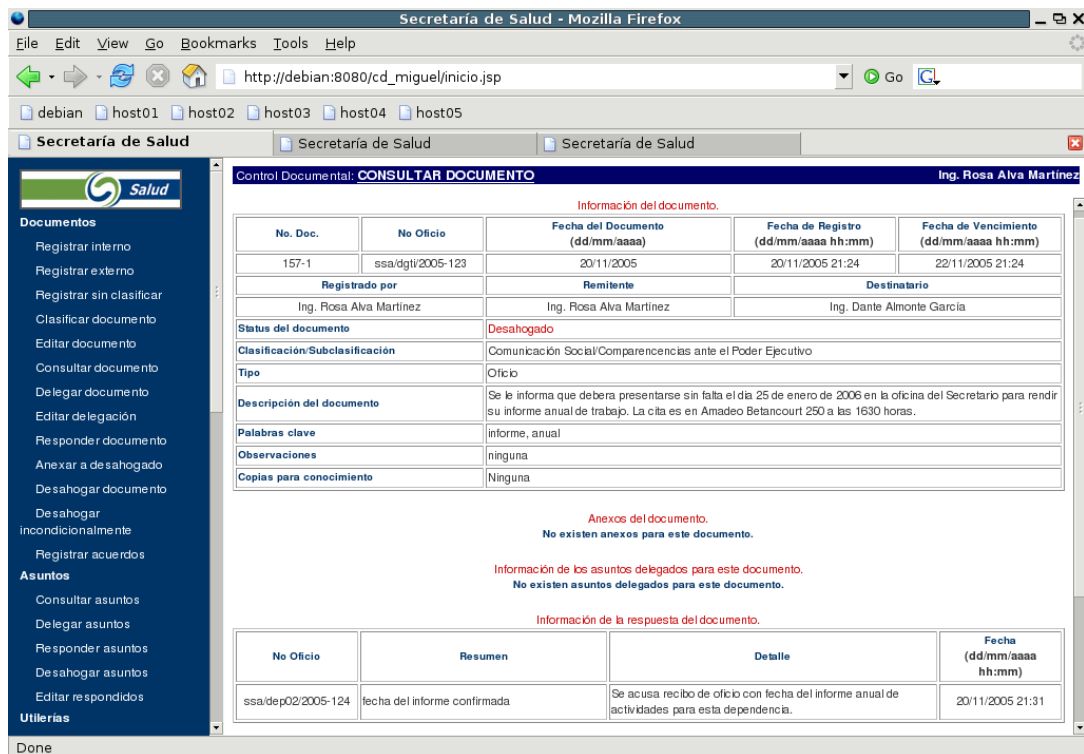
Además, debido a la naturaleza de la replicación en el SCDD, la información de los Nodos Locales es sincronizada en primer lugar con el Nodo Central. De esta manera, el Nodo Central contienen una copia de seguridad de la información de todos los Nodos Locales del SCDD. Por tanto, si Alva pudiera ingresar al Nodo Central utilizando su usuario y su password, ella vería lo siguiente:

**Figura 5.19. Pantalla inicial de Alva en el Nodo Central del SCDD**



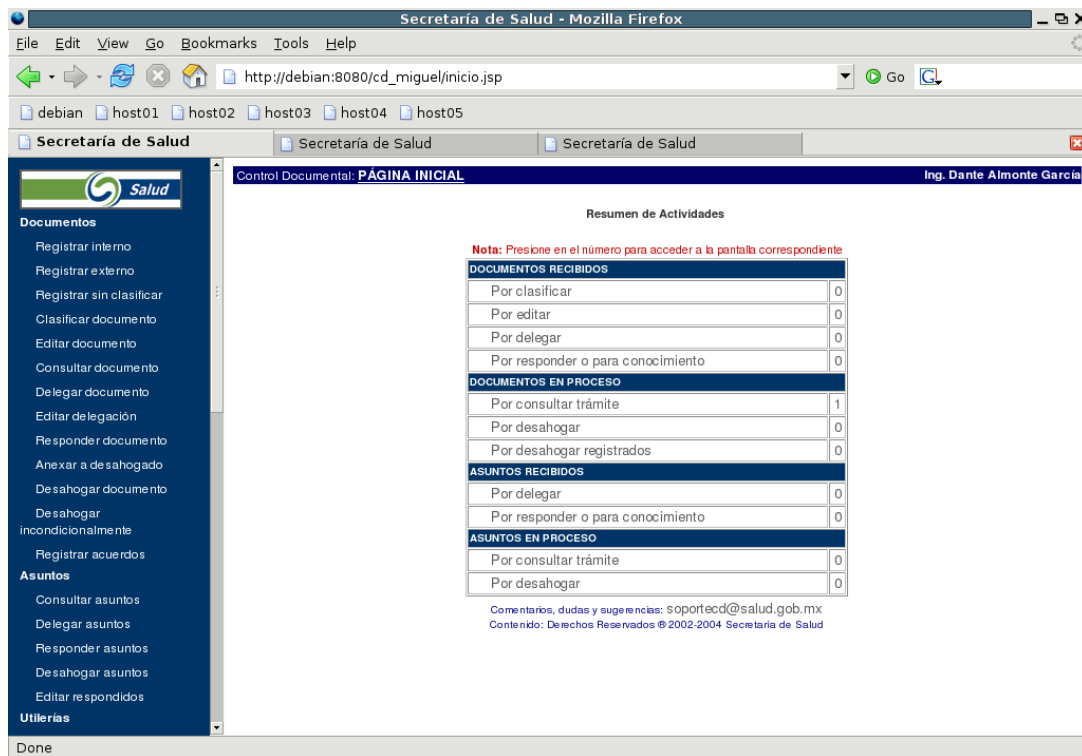
Y al ingresar a la lista de documentos por consultar, podría ver el detalle del documento y la respuesta que se asignó a ese documento.

**Figura 5.20. Consulta de documentos para Alva en el Nodo Central del SCDD**



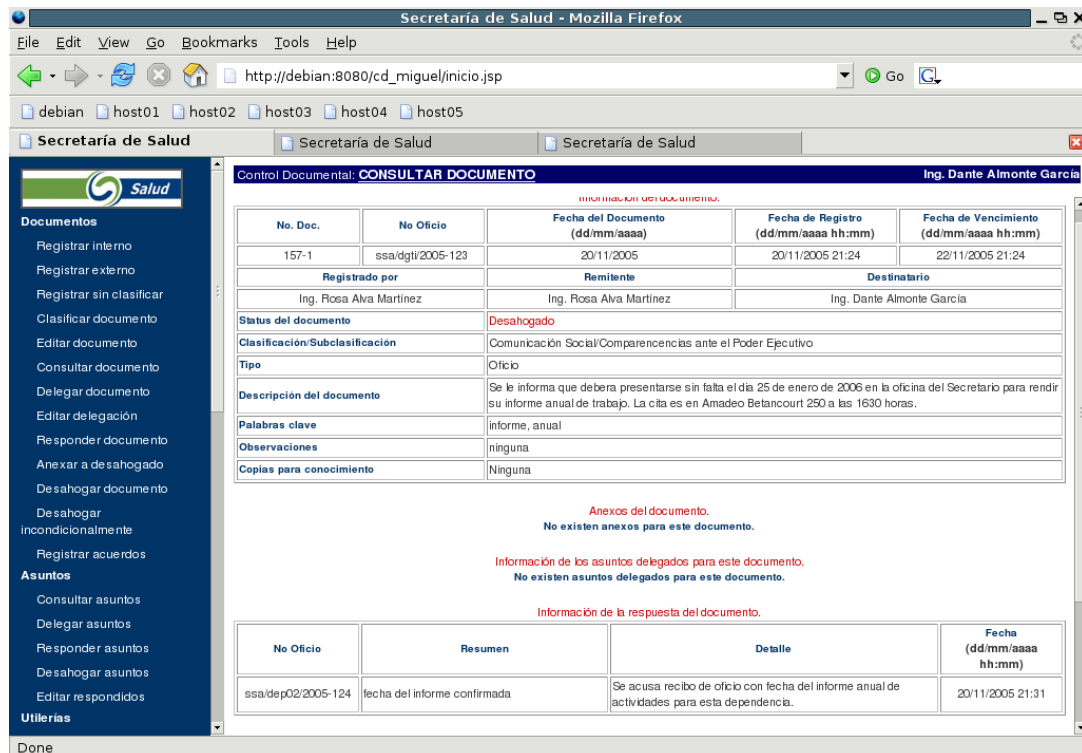
Algo similar sucedería para Dante si ingresara al Nodo Central.

**Figura 5.21. Pantalla inicial de Dante en el Nodo Central del SCDD**



También él, al ingresar a la lista de documentos por consultar, podría ver todos los detalles del documento y de la respuesta que él mismo generó.

**Figura 5.22. Consulta de documentos para Alva en el Nodo Central del SCDD**



## 2. Escenarios para el SCDD

### 2.1. Operación normal

El primer escenario posible y el que se espera ocurra el 99% del tiempo es el funcionamiento normal del SCDD. En este escenario se dan las siguientes condiciones:

- El Nodo Central funciona correctamente.
- El Nodo Central es accesible a través de la infraestructura de red.
- Al menos un Nodo Local funciona correctamente.
- Cada Nodo local es accesible a través de la infraestructura de red.

En este caso todo funciona perfectamente y el SCDD no tiene que realizar ninguna tarea adicional para asegurar la integridad de la información.

### 2.2. Fallo en la infraestructura de red

Este escenario es, desafortunadamente, muy común para el SCDD. Como hemos visto, éste es el principal motivo que originó su creación.

Debido al tipo de efectos que produce un fallo en la infraestructura de red dentro del SCDD, este escenario es equivalente a alguno de los escenarios donde hay un fallo en el Nodo Central o en alguno de los Nodos Locales.

En el caso de un fallo en el enlace de entrada del Nodo Central, los Nodos Locales no pueden acceder a él y, en consecuencia, no pueden sincronizar su información con él. Si el enlace estuviera funcionando correctamente y fuera el Nodo Central el que tuviera un fallo, los Nodos Locales se encontrarían en la misma situación y no podrían sincronizar su información con el Nodo Central. Para efectos prácticos, dentro del SCDD, ambos tipos de fallos resultan en la imposibilidad de los Nodos Locales para sincronizar la información con el Nodo Central.

Algo similar sucede en el caso de un fallo en el enlace de red de un Nodo Local. En este caso, al igual que si el fallo fuera en el Nodo Local y no en su enlace de acceso, los demás Nodos Locales que conforman al SCDD se verían, de hecho, imposibilitados para sincronizar su información con el Nodo Local.

Así pues, durante un fallo en la infraestructura de red, el SCDD se comporta de manera exactamente equivalente a cuando falla un Nodo de los que lo conforman.

Para que este escenario se dé, se deben cumplir cualquiera de dos conjuntos de condiciones.

El primero es el siguiente:

- El Nodo Central no es accesible a través de la infraestructura de red.
- El Nodo Central puede o no operar correctamente en el momento del fallo de la red.

Debido a que hay un fallo en la red, no hay manera para un Nodo Local de saber si el Nodo Central está funcionando. Por tanto, es irrelevante, desde la perspectiva de un Nodo Local, el estado del Nodo Central.

Este caso es equivalente a un fallo en el Nodo Central.

El segundo conjunto de condiciones es el siguiente:

- Un Nodo Local no es accesible a través de la infraestructura de red.
- Un Nodo Local puede o no operar correctamente en el momento del fallo de la red.

Al igual que en el caso anterior, no hay manera de saber, para los demás Nodos Locales del SCDD, si el Nodo Local está funcionando durante una interrupción del enlace de red. Por tanto, también es irrelevante el estado del Nodo Local para los otros Nodos Locales.

Este caso es equivalente a un fallo en un Nodo Local

### **2.3. Fallo en el Nodo Central**

Este es, por mucho, el escenario más peligroso de todos. Como se ha visto en los capítulos anteriores, el Nodo Central es indispensable para el funcionamiento del SCDD. Por tanto, aunque los Nodos pueden seguir operando perfectamente por varios días, si se da este escenario, debe resolverse con la mayor celeridad.

Este escenario se da cuando se cumplen las siguientes condiciones:

- El Nodo Central tiene un fallo.
- El enlace de red del Nodo Central puede o no estar disponible.

Se puede ver que el estado del enlace de red no altera en nada el hecho de que el Nodo Central es inalcanzable para los Nodos Locales. El efecto neto es que la información de los Nodos Locales no se puede sincronizar con el Nodo Central.

Este caso, por ser el más peligroso, debe ser minimizado al máximo. Ya se ha hablado en el capítulo anterior de las distintas maneras en que se puede reducir el impacto de un fallo en los Nodos del SCDD, así que no se ahondará más en ellos aquí.

### **2.4. Fallo en el Nodo Local**

Este escenario, aunque peligroso en sí mismo y, por supuesto, problemático para los usuarios de las dependencias que hacen uso de sus servicios, no es, visto desde la perspectiva del SCDD como un todo, de mayor relevancia para el funcionamiento óptimo del sistema. Esto es así debido a que el SCDD se diseñó desde el principio para afrontar este tipo de problemas y minimizarlos al máximo.

Este escenario se da cuando se cumplen las siguientes condiciones:

- Un Nodo Local falla.

- El enlace de red del Nodo Local puede o no estar disponible.

Al igual que con el fallo en el Nodo Central, durante un fallo en un Nodo Local es irrelevante el estado del enlace de red. En cualquier caso, los demás Nodos Locales no pueden sincronizar su información con el Nodo Local que sufrió el fallo.

Debe remarcarse aquí que el SCDD seguirá operando sin ningún problema, aún cuando se den varios escenarios como éste de manera simultánea durante la operación del mismo. Como se dijo antes, y sólo para recalcarlo, el SCDD se diseñó desde el inicio precisamente para reducir al máximo la influencia de este tipo de eventualidades en su funcionamiento diario.

### **3. Pruebas de rendimiento**

Para un sistema tan grande como es el SCDD y con la importancia de la información que controla, se pueden y deben hacer un gran número de pruebas para sondear y cuantificar, en la medida de lo posible, el nivel de servicio que proporcionará una vez puesto en operación. La mayoría de las pruebas miden puntos específicos del funcionamiento. Otras intentan exponer el sistema a situaciones extraordinarias para observar su respuesta, de manera que pueda afinarse o mejorarse el funcionamiento del mismo en este tipo de situaciones. Finalmente están las que miden el grado de rendimiento que se obtendrá del sistema una vez operando.

Todos sabemos que un sistema, como arquitectura o como diseño sobre el papel, es capaz de atender a un número enorme de usuarios simultáneos o de realizar cierto número de tareas por segundo de manera sostenida. Lamentablemente, una vez realizado el sistema y puesto en operación, el sistema rara vez se acerca a los números pronosticados teóricamente.

En el caso del SCDD, y para el objetivo de esta tesis, el tipo de pruebas que nos interesan son las de rendimiento. Las otras clases de pruebas, importantes en sí mismas, se han realizando durante el desarrollo del sistema.

Por tanto, en este capítulo desarrollaremos pruebas de rendimiento del SCDD y analizaremos los resultados obtenidos

#### **3.1. Tipos de pruebas de rendimiento**

Existen varios tipos de pruebas de rendimiento.

##### **3.1.1. Pruebas en simulador**

En este caso se construye un sistema independiente con el objetivo único de simular el comportamiento del sistema desarrollado en distintas situaciones y sujeta a varias restricciones, variando un conjunto de parámetros de manera que se pueda, con la información obtenida, inferir el comportamiento del sistema en el entorno real.

**Ventajas:**



Aquí la inversión más grande se da durante la etapa de construcción del simulador que llevará a cabo las pruebas. Después de esto, el costo es prácticamente nulo. Una vez completado el simulador, se pueden llevar a cabo infinidad de pruebas utilizando distintos valores de entrada para parámetros clave y medir y comparar los resultados de dichas pruebas.

#### **Desventajas:**

Dado que el sistema desarrollado no se está probando en condiciones equivalentes a las que tendrá durante su puesta en producción, los datos arrojados por el simulador pueden ser demasiado optimistas y a veces hasta irreales. Esto puede reducirse considerablemente añadiendo complejidad al simulador, de manera que pueda proporcionar un entorno lo más parecido al entorno real para el sistema en prueba. Esto, dependiendo de la complejidad del entorno real, así como del presupuesto asignado al desarrollo del sistema, puede variar desde lo trivial hasta lo imposible.

#### **3.1.2. Pruebas en entornos reales**

En este tipo de pruebas en lugar de invertir recursos, tanto humanos como económicos, en la construcción de un simulador, se invierten en la creación de un entorno similar al que será usado durante la etapa de producción del sistema. En este caso, las condiciones son lo más parecidas posibles a las reales. Por tanto, los resultados que se obtengan tienen un mayor grado de confiabilidad en la estimación del rendimiento del sistema en el entorno final.

#### **Ventajas:**

Las ventajas obvias son el mayor grado de confiabilidad de los datos obtenidos. Además, existen otras ventajas adicionales que un simulador no puede dar, pero que las pruebas en entornos reales sí proporcionan. En una prueba en un entorno real podemos estimar los fallos físicos y efectos del entorno en el funcionamiento del sistema puesto a prueba. Por ejemplo, un sistema que esté destinado a funcionar en una habitación muy pequeña, probablemente sufra fallos por sobrecalentamiento debido al calor emitido por los componentes mecánicos y electrónicos del sistema. En este caso, durante la prueba con entornos reales, estos problemas saldrán a relucir y podrán tomarse medidas para evitarlo en el sistema de producción. Esto no se hubiera podido analizar en un sistema de pruebas simulado sin invertir tanto recursos económicos, como humanos y tiempo que se podría utilizar para mejorar el sistema que se está poniendo a prueba, en lugar a destinarlos a mejorar el simulador.

#### **Desventajas:**

La desventaja más grande de este tipo de pruebas es su costo. Este costo puede ser mínimo en algunos casos. En otros puede ser completamente desorbitado o completamente ajeno al económico. Por ejemplo, puede ser imposible construir un entorno de pruebas similar al que será usado durante la producción porque el costo de dicho entorno de pruebas sería superior al costo del sistema que se quiere probar. En este caso es un problema de costo-beneficio. Por otro lado,

quizá sea imposible conseguir equipo similar al de producción. Por ejemplo, si cierto sistema va a ser utilizado en un sistema de hardware especializado del que sólo existen 3 en el mundo, quizá no sea posible conseguir tiempo para pruebas en alguno de ellos, debido a la demanda de tiempo que tienen estos sistemas. Finalmente, quizá el recurso económico no sea el factor decisivo, sino el que, a pesar de contar con todos los recursos económicos, no sea posible utilizar el entorno de producción para las pruebas del sistema. Este caso podría darse, por ejemplo, en un sistema destinado a una planta nuclear, donde no se puede detener el sistema actual para hacer las pruebas del nuevo sistema, y tampoco se puede construir otra planta con el único fin de probar el nuevo sistema.

### **3.1.3. Pruebas híbridas**

Este tipo de pruebas combinan las ventajas de cada uno de los otros dos tipos de pruebas y tratan de minimizar las desventajas de las mismas.

Un entorno de pruebas de este tipo, por ejemplo, podría utilizar simuladores en ciertas etapas del desarrollo, y entornos reales de partes específicas del entorno de producción, durante otras. Podría utilizar, por ejemplo, simuladores en las partes más caras de reproducir, y entornos reales en los puntos donde se desea saber si existen variables ocultas que puedan afectar el rendimiento del sistema y que no se hayan previsto o no puedan ser integradas al simulador.

#### **Ventajas:**

Bajo costo en comparación con los otros dos entornos de pruebas si se eligen adecuadamente los escenarios que se simularán y los que se probarán en entornos reales.

#### **Desventajas:**

Si no se eligen correctamente los entornos que serán simulados y los que serán probados en entornos reales, este tipo de pruebas pueden ser contraproducentes y aumentar el costo del sistema, además de que brindarán a los desarrolladores del mismo una falsa seguridad basada en los datos arrojados por las pruebas. Esto puede llevar a expresar una confianza desmedida en el rendimiento del mismo. Confianza que se verá desmentida drásticamente cuando el sistema esté siendo usado en producción.

## **3.2. Metodología a seguir**

Para el caso del SCDD seguiremos un entorno de pruebas híbridas, ya que usaremos tanto simuladores, como entornos similares al de producción. Estas pruebas intentarán dar un estimado del nivel de respuesta del sistema durante el procesamiento de grandes cantidades de datos documentales.

El objetivo de éstas es dar resultados basados en el rendimiento mínimo para el SCDD.

El Sistema de Control Documental Distribuído será usado por dependencias de gobierno y no gubernamentales para almacenar información de alto valor durante un largo

período de tiempo. Se supone en esta tesis que, para conseguir tal fin, cada dependencia que cuente con un Nodo Local del SCDD hará una inversión razonable y significativa en la infraestructura destinada al Nodo Local. Esto puede traducirse, por ejemplo, en la compra de equipo de cómputo moderno y de calidad, teniendo en mente el objetivo que se espera de él. Pero no solamente eso, sino también en la contratación de recursos humanos que se encarguen de que el Nodo Local opere constantemente en los niveles óptimos, y se encarguen, además, de darle mantenimiento y de monitorear su desempeño, haciendo ajustes tanto al software como al hardware en caso de ser necesario.

Esto es lo que se supone que será el entorno de producción final. Por tanto, si las pruebas son realizadas en un entorno que sea un subconjunto del entorno final de producción, entendiendo por esto equipo de cómputo de menor poder de procesamiento y no tan moderno, así como software y hardware no optimizado al máximo; es razonable suponer que los valores obtenidos serán representativos del rendimiento mínimo esperado del entorno real. Es más, al ser el sistema de producción de mayor capacidad y operado con mayor calidad que el sistema de pruebas, es razonable suponer que el rendimiento será también mayor en el sistema real que en el usado por las pruebas. Si el rendimiento en el entorno de pruebas es satisfactorias, entonces el rendimiento en el entorno de producción lo serán aún más.

Esa es la idea que se desarrolla en esta tesis para medir el rendimiento del SCDD.

El entorno de pruebas constará de los siguientes elementos:

- 1 Nodo Central

Este Nodo Central será el depositario último de toda la información documental existente en el SCDD.

- 5 Nodos Locales

Estos serán los encargados de controlar y administrar documentos para 5 dependencias.

Las características de cada uno de estos nodos son las siguientes:

**Tabla 5.1. Características de hardware de los nodos**

<b>Nodo</b>	<b>Tipo de Nodo</b>	<b>Nombre</b>	<b>CPU</b>	<b>Velocidad (GHz)</b>	<b>Memoria (MBytes)</b>	<b>Disco Duro (GBytes)</b>	<b>Sistema operativo</b>
0	Central	debian	Intel PIV	2.0	121	40	Debian GNU/Linux 3.1
1	Local	host01	Intel PIV	2.0	127	1.25	Debian GNU/Linux 3.1
2	Local	host02	Intel PIV	2.0	127	1.25	Debian GNU/Linux

Nodo	Tipo de Nodo	Nombre	CPU	Velocidad (GHz)	Memoria (MBytes)	Disco Duro (GBytes)	Sistema operativo
							3.1
3	Local	host03	Intel PIV	2.0	127	1.25	Debian GNU/Linux 3.1
4	Local	host04	Intel PIV	2.0	127	1.25	Debian GNU/Linux 3.1
5	Local	host05	Intel PIV	2.0	111	1.25	Debian GNU/Linux 3.1

Además, cada host tiene la siguiente configuración:

**Tabla 5.2. Configuración de los nodos**

Nodo	Dirección IP	/ (GBytes)	swap (MBytes)	Tomcat	PostgreSQL
debian	192.168.254.1	6.0	2024	4.1.31-3	7.4.7-6sarge1
host01	192.168.254.11	1.0	256	4.1.31-3	7.4.7-6sarge1
host02	192.168.254.12	1.0	256	4.1.31-3	7.4.7-6sarge1
host03	192.168.254.13	1.0	256	4.1.31-3	7.4.7-6sarge1
host04	192.168.254.14	1.0	256	4.1.31-3	7.4.7-6sarge1
host05	192.168.254.15	1.0	256	4.1.31-3	7.4.7-6sarge1

Parte de la simulación consiste en representar 5 dependencias, cada una de las cuales tiene un Nodo Local. Estas dependencias y los usuarios pertenecientes a cada una de ellas se muestran a continuación:

**Tabla 5.3. Información del SCDD para los nodos**

Nodo	Nombre de la dependencia	Siglas	Clave de la dependencia	Usuario de la dependencia	Clave del usuario	Nombre del usuario
debian	Dirección General de Tecnologías de la Información	DGTI	47	Todos	Todos	Todos
host01	Dependencia 01	DEP01	157	alva	3113	Rosa Alva Martínez
host02	Dependencia 02	DEP02	158	dante	3114	Dante Almonte García
host03	Dependencia 03	DEP03	159	cesar	3115	César

Nodo	Nombre de la dependencia	Siglas	Clave de la dependencia	Usuario de la dependencia	Clave del usuario	Nombre del usuario
						Mendoza Trujillo
host04	Dependencia 04	DEP04	160	german	3116	Germán Mejías Arellano
host05	Dependencia 05	DEP05	161	gerardo	3117	Gerardo Caballero Treviño

Antes de las pruebas ningún nodo contenía información documental. Solamente la información necesaria para permitir ingresar a sus usuarios y crear documentos.

Por lo tanto, en cada uno de los nodos se crearon 10,000 documentos, de manera que se pudieran hacer las pruebas de replicación de información entre los nodos. Los documentos tienen la siguiente información:

**Tabla 5.4. Información de los documentos**

Elemento	Valor
Número de documento	DDD-NNNNN
Número de oficio	Doc-NNNNN
Fecha del documento	10/10/2005
Tiempo de vencimiento	2 días
Clasificación	Adicciones y Salud Mental
Subclasificación	Atención y Control de Trastornos Mentales y Neurológicos
Tipo de documento	Carta
Descripción del documento	Documento auto-generado #NNNNN
Palabras clave	Documento, generado, NNNNN
Observaciones	Es muy fácil generar documentos
Copias para conocimiento	Ninguna

Donde **DDD** es la clave de la dependencia y **NNNNN** es el número del documento.

Cada documento, a su vez, tiene 10 anexos. Un anexo es otro documento que se envía como agregado al documento principal. Este documento anexo está subordinado al documento principal y puede complementar la información del mismo. Por ejemplo, si se enviara un memorándum ordenando la impresión de carteles para una campaña de vacunación a nivel nacional, se enviaría como anexo un archivo electrónico con el diseño del cartel.

Para el objetivo de esta prueba, se generaron anexos ficticios a cada documento, cada uno con la siguiente información:

**Tabla 5.5. Información de los anexos.**

Elemento	Valor
Tipo de anexo	Documento(s) en papel
Descripción	Anexo <b>M</b> del documento <b>NNNNN</b>
Documento electrónico	No incluido

Donde **M** es el número de anexo, entre 1 y 10, y **NNNNN** es el número de documento.

Además, cada documento tiene un remitente y un destinatario. El destinatario variaba de acuerdo al remitente y a la dependencia. Como hay 5 dependencias y cada dependencia tiene un usuario, para cada dependencia se crearon documentos dirigidos a cada uno de los usuarios de las otras cuatro dependencias. Aproximadamente un 25% de los documentos generados fue enviado a cada dependencia. El número real de documentos enviados a cada dependencia se muestra en la siguiente tabla.

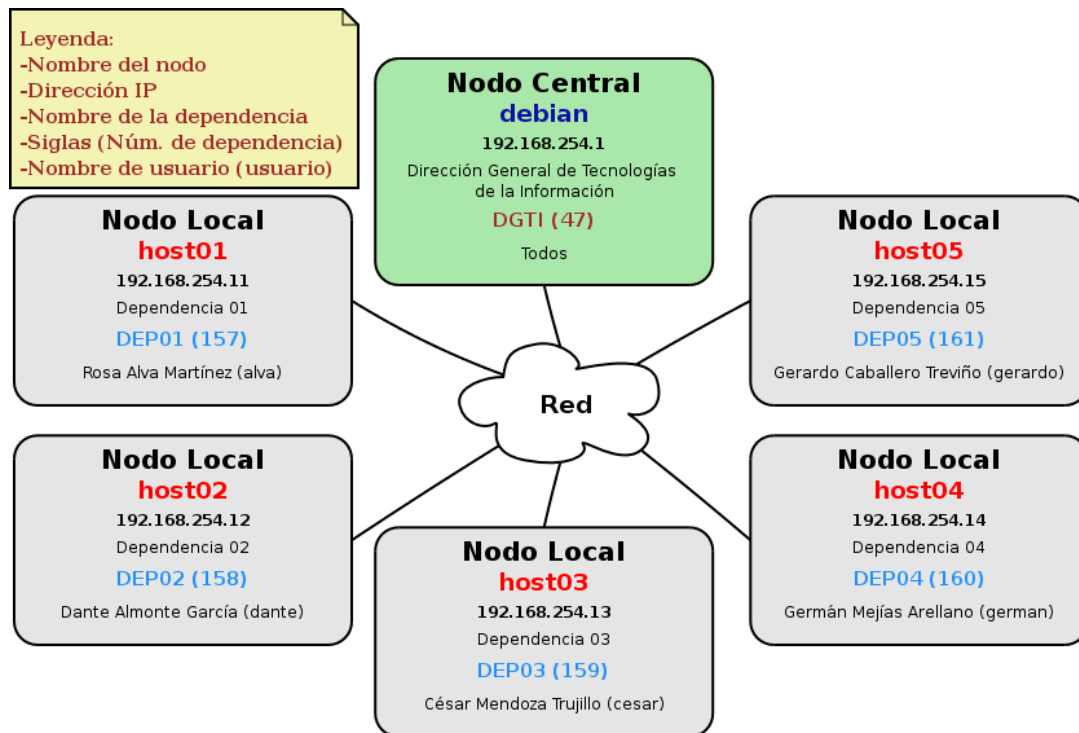
Cada renglón muestra al transmisor de la información documental y cada columna muestra al receptor de la misma. La intersección de un renglón y una columna muestra el número de documentos enviados. Por ejemplo, host03 envió 2501 documentos a host05.

**Tabla 5.6. Número de documentos transmitidos**

	debian	host01	host02	host03	host04	host05	Total de documentos enviados
host01	10,000	N/A	2,499	2,500	2500	2,501	<b>20,000</b>
host02	10,000	2,499	N/A	2,500	2500	2,501	<b>20,000</b>
host03	10,000	2,499	2,500	N/A	2500	2,501	<b>20,000</b>
host04	10,000	2,499	2,500	2,500	N/A	2,501	<b>20,000</b>
host05	10,000	2,499	2,500	2,500	2501	N/A	<b>20,000</b>
Total de documentos recibidos	<b>50,000</b>	<b>9,996</b>	<b>9,999</b>	<b>10,000</b>	<b>10,001</b>	<b>10,004</b>	

Un diagrama de los nodos usados para estas pruebas se muestra en la siguiente figura:

**Figura 5.23. Configuraciones del Subsistema de la Base de Datos del SCDD**



Una vez establecido el escenario de pruebas, se procedió a realizar la replicación de la información documental entre los nodos del sistema. Cada nodo transmitirá primero sus 10,000 documentos, cada uno con sus 10 anexos, al Nodo Central. Si no se produjo ningún error durante la sincronización con el Nodo Central, el Nodo Local procederá a sincronizarse con cada Nodo Local, si es que tiene documentos destinados a ese nodo. Cada Nodo Local transmitirá, por lo tanto, 2500 documentos aproximadamente, además de los anexos de cada documento, a cada uno de los otros cuatro Nodos Locales.

Al final de las pruebas, el Nodo Central contendrá una copia de cada documento existente en cada Nodo Local, totalizando:

$$(5 \text{ nodos}) \times (10,000 \text{ documentos/nodo}) = 50,000 \text{ documentos únicos}$$

Además, contendrá también 10 anexos por cada documento sumando:

$$(10 \text{ anexos/documento}) \times (50,000 \text{ documentos}) = 500,000 \text{ anexos.}$$

Cada Nodo Local, por otro lado, tendrá aproximadamente 10,000 documentos propios, más 10,000 documentos recibidos de los otros cuatro Nodos Locales, totalizando 20,000 documentos. Cada uno de estos documentos tendrá también 10 anexos, por lo que el número de anexos por Nodo Local será de aproximadamente:

$$(20,000 \text{ documentos}) \times (10 \text{ anexos/documento}) = 200,000 \text{ anexos.}$$

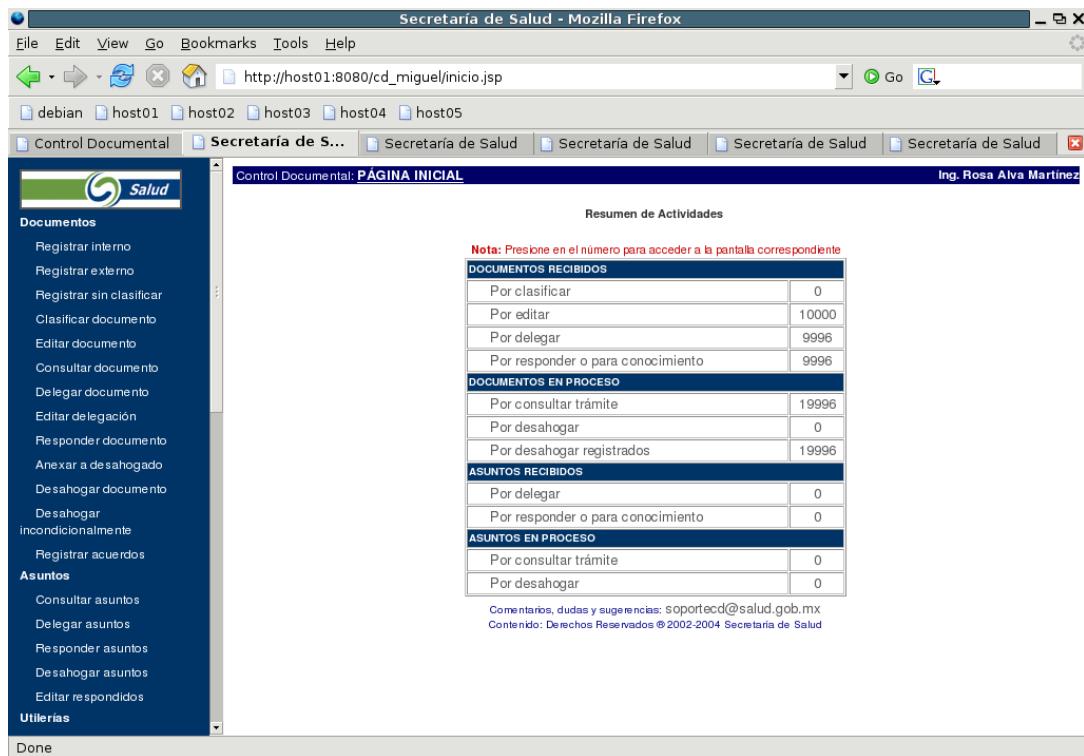
Debido a la manera en que se almacena la información en la base de datos de cada nodo del SCDD, el número de registros o *tuplas* necesarias para almacenar un documento, junto con sus 10 anexos, es de 15. Por tanto, el número de registros transmitidos por nodo es entonces 15 veces el número de documentos que transmitió. El número exacto de documentos, anexos y tuplas en cada nodo se muestra en la siguiente tabla:

**Tabla 5.7. Número de documentos y anexos por nodo**

Nodo	Documentos por Nodo	Anexos por Nodo	Tuplas por Nodo
debian	50,000	500,000	750,000
host01	19,996	199,960	299,940
host02	19,999	199,999	299,985
host03	20,000	200,000	300,000
host04	20,001	200,010	300,015
host05	20,004	200,040	300,060

En las siguientes figuras se muestran la pantalla inicial de cada uno de los usuarios en cada una de las dependencias de estas pruebas. En primer lugar, Alva que se encuentra asignada a la Dependencia 01:

**Figura 5.24. Total de documentos recibidos por Alva en host01 (Dependencia 01)**



Dante por otro lado, está asignado a la Dependencia 02:

**Figura 5.25. Total de documentos recibidos por Dante en host02 (Dependencia 02)**



Control Documental: PÁGINA INICIAL Ing. Dante Almonte Garcia

Resumen de Actividades

**Nota: Presione en el número para acceder a la pantalla correspondiente**

DOCUMENTOS RECIBIDOS	
Por clasificar	0
Por editar	10000
Por delegar	9999
Por responder o para conocimiento	9999
DOCUMENTOS EN PROCESO	
Por consultar trámite	19999
Por desahogar	0
Por desahogar registrados	19999
ASUNTOS RECIBIDOS	
Por delegar	0
Por responder o para conocimiento	0
ASUNTOS EN PROCESO	
Por consultar trámite	0
Por desahogar	0

Comentarios, dudas y sugerencias: [soportecd@salud.gob.mx](mailto:soportecd@salud.gob.mx)  
 Contenido: Derechos Reservados © 2002-2004 Secretaría de Salud

El destinatario de los documentos de la Dependencia 03 es César:

**Figura 5.26. Total de documentos recibidos por César en host03 (Dependencia 03)**

Control Documental: PÁGINA INICIAL Ing. Cesar Luis Mendoza Trujillo

Resumen de Actividades

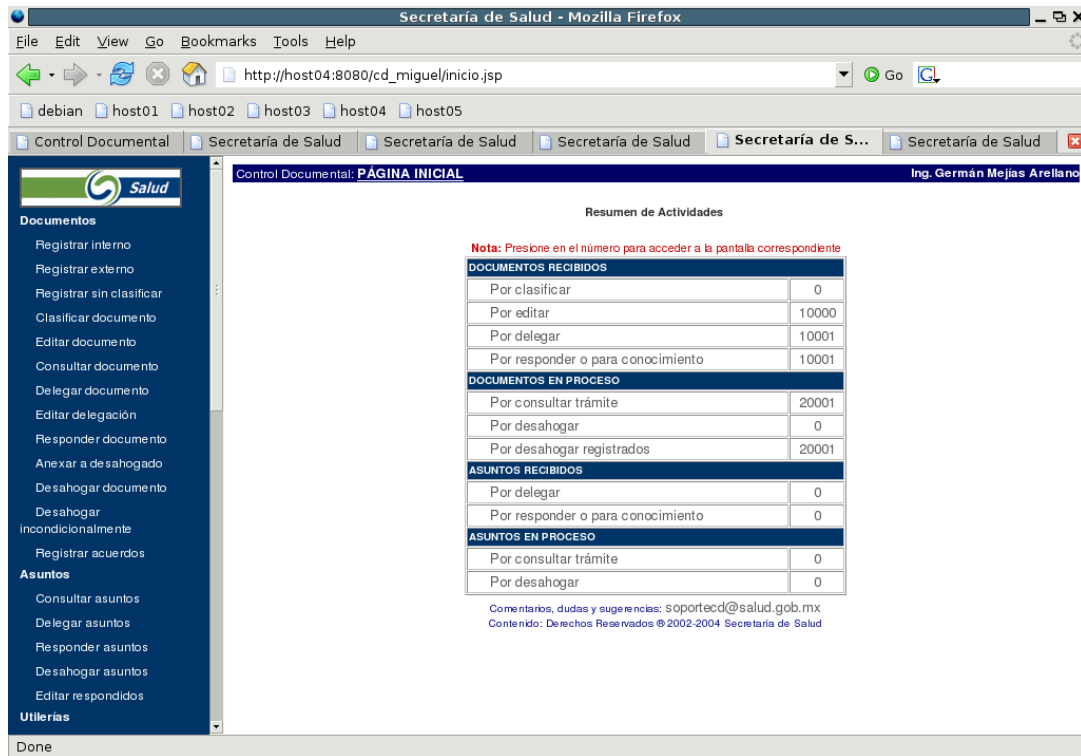
**Nota: Presione en el número para acceder a la pantalla correspondiente**

DOCUMENTOS RECIBIDOS	
Por clasificar	0
Por editar	10000
Por delegar	10000
Por responder o para conocimiento	10000
DOCUMENTOS EN PROCESO	
Por consultar trámite	20000
Por desahogar	0
Por desahogar registrados	20000
ASUNTOS RECIBIDOS	
Por delegar	0
Por responder o para conocimiento	0
ASUNTOS EN PROCESO	
Por consultar trámite	0
Por desahogar	0

Comentarios, dudas y sugerencias: [soportecd@salud.gob.mx](mailto:soportecd@salud.gob.mx)  
 Contenido: Derechos Reservados © 2002-2004 Secretaría de Salud

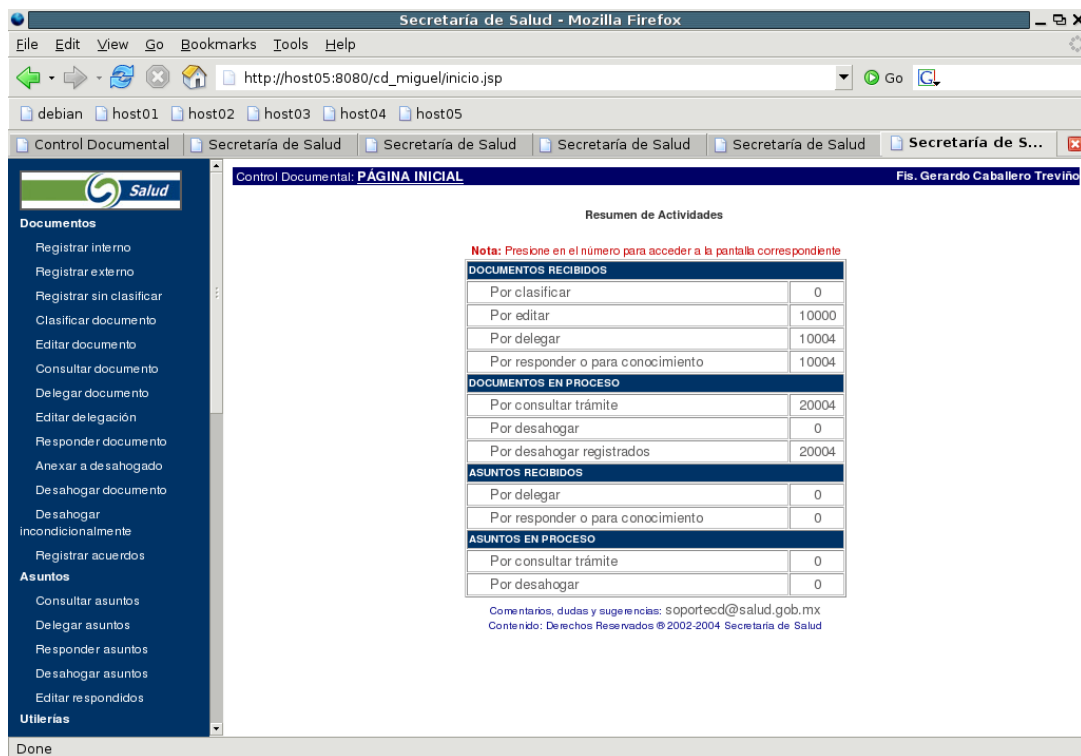
Para la Dependencia 04, el destinatario es Germán:

**Figura 5.27. Total de documentos recibidos por Germán en host04 (Dependencia 04)**



Finalmente, en la Dependencia 05, el destinatario es Gerardo:

**Figura 5.28. Total de documentos recibidos por Gerardo en host05 (Dependencia 05)**



Por otro lado, cada documento generado para estas pruebas, junto con sus 10 anexos, requiere de 6540 Bytes como mínimo para transmitirse al nodo destino. Por tanto, como cada Nodo Local transmitió sus 10,000 documentos tanto al Nodo Central como a los otros Nodos Locales, cada Nodo Local transmitió 20,000 documentos en total. Por tanto, cada Nodo Local transmitió:

$(20,000 \text{ documentos}) \times (6540 \text{ bytes/documento}) = 130800000 \text{ bytes en total.}$

Es decir, aproximadamente 124.74 MBytes.

Se harán mediciones del tiempo utilizado por cada Nodo Local para transmitir sus 10,000 documentos al Nodo Central, en segundos, así como el utilizado para transmitir los documentos a cada uno de los otros Nodos Locales.

Con estos resultados se podrá estimar la capacidad del SCDD para replicar la información con el Nodo Central, así como con los otros Nodos Locales.

### **3.3. Herramientas existentes**

Como se dijo en la sección anterior, para las pruebas de rendimiento del SCDD se utilizó un esquema híbrido. El escenario de pruebas físico está representado por el hardware donde se ejecutan los Nodos Locales y el Nodo Central. Las características de este hardware se detallan en la sección anterior. En esta sección detallaré la parte de las pruebas que utilizará un simulador.

La simulación en este caso se hizo en cuanto a que las computadoras pertenecientes a cada dependencia, ya que ninguna de ellas es real. Esto es posible debido al uso de herramientas de *virtualización*. Para estas pruebas se utilizó el software desarrollado en el *University of Cambridge Computer Laboratory* llamado Xen.

Xen es un *Virtual Machine Monitor* (monitor de máquinas virtuales) desarrollado para procesadores con arquitectura x86 que permite a múltiples sistemas operativos compartir hardware convencional de manera administrada, tanto en recursos como en seguridad, pero sin sacrificar ni rendimiento ni funcionalidad para el sistema operativo huésped.

Xen, por tanto, fue usado para crear, en una sola máquina, 6 instancias simultáneas del sistema operativo Debian GNU/Linux 3.1. De estas 6 instancias, una fue utilizada para el Nodo Central (debian) y las restantes 5 para los Nodos Locales (host01 a host05).

Debido al pequeño impacto en rendimiento que introduce Xen al sistema operativo huésped, los datos arrojados por las pruebas siguen siendo representativos del rendimiento esperado del SCDD operando en el entorno de producción. Además, debido a que estamos suponiendo el mínimo rendimiento, una vez que el sistema esté en producción, usando equipo de cómputo más eficiente y software optimizado, los resultados serán indudablemente mejores. Por tanto, se supone que la realización de las pruebas utilizando Xen para simular los seis nodos de nuestro entorno de prueba no demerita ni disminuye la calidad y confiabilidad de los resultados obtenidos.

La versión de Xen utilizada para estas pruebas fue la 2.0.7 y fue compilada junto con un kernel de Linux versión 2.6.11 para crear el sistema de pruebas para esta tesis. Se creó, además, un programa que se encargó de generar los 10,000 documentos en cada Nodo Local. El código fuente de este programa se lista en el apéndice B.

## 4. Resultados

Una vez generados los documentos en cada Nodo Local, se replicó esta información tanto al Nodo Central como a los demás Nodos Locales. Los resultados de las mediciones en segundos se muestran en la siguiente tabla:

**Tabla 5.8. Duración de la replicación de documentos (segundos)**

	<b>debian</b>	<b>host01</b>	<b>host02</b>	<b>host03</b>	<b>host04</b>	<b>host05</b>
<b>host01</b>	457	N/A	98	113	107	87
<b>host02</b>	362	98	N/A	96	86	90
<b>host03</b>	424	96	88	N/A	93	90
<b>host04</b>	415	110	86	103	N/A	108
<b>host05</b>	489	100	130	90	83	N/A

La tabla muestra el tiempo utilizado para enviar los documentos desde el Nodo Local indicado por el renglón, hacia el Nodo Local indicado por la columna. Por ejemplo, host01 tardó 113 segundos en enviar sus documentos a host03.

El número de documentos enviados por cada Nodo Local se detalla en la sección anterior, en la tabla *Número de documentos transmitidos*.

En la siguiente tabla se muestra el número de documentos transmitidos por segundo por cada Nodo Local. Este tiempo se calculó dividiendo el número de documentos transmitidos, entre el tiempo en segundos utilizado para transmitirlos. Las unidades son en documentos por segundo.

**Tabla 5.9. Documentos por segundo transmitidos por cada Nodo Local**

	<b>debian</b>	<b>host01</b>	<b>host02</b>	<b>host03</b>	<b>host04</b>	<b>host05</b>
<b>host01</b>	21.88	N/A	25.50	22.12	23.36	28.75
<b>host02</b>	27.62	25.50	N/A	26.04	29.07	27.79
<b>host03</b>	23.58	26.03	28.41	N/A	26.88	27.79
<b>host04</b>	24.10	22.72	29.07	24.27	N/A	23.16
<b>host05</b>	20.45	24.99	19.23	27.78	30.13	N/A

En la tabla siguiente se muestran algunos cálculos estadísticos realizados sobre estos datos:

**Tabla 5.10. Estadística descriptiva del número de documentos por segundo transmitidos por cada Nodo Local**

<b>Medida</b>	<b>Valor</b>
Media	25.45
Moda	25.5
Mediana	25.5
Desviación estándar	2.91
Varianza	8.46
Error estándar	0.58

Sabemos, por de la sección anterior, que cada documento enviado requiere de 6540 Bytes para su transmisión. Si multiplicamos el número de documentos promedio durante la sincronización de datos entre Nodos Locales por este valor, obtendremos el número de bytes transmitidos durante la sincronización entre los nodos. Esto se muestra en la siguiente tabla. Las unidades están en KiloBytes por segundo.

**Tabla 5.11. Bytes por segundo transmitidos por cada Nodo Local (KBytes/segundo)**

	<b>debian</b>	<b>host01</b>	<b>host02</b>	<b>host03</b>	<b>host04</b>	<b>host05</b>
<b>host01</b>	139.75	N/A	162.86	141.30	149.22	183.60
<b>host02</b>	176.43	162.86	N/A	166.32	185.66	177.48
<b>host03</b>	150.63	166.25	181.44	N/A	171.69	177.48
<b>host04</b>	153.90	145.09	185.66	155.02	N/A	147.90
<b>host05</b>	130.61	159.60	122.82	177.41	192.45	N/A

En la siguiente tabla se muestran algunos cálculos estadísticos sobre estos datos:

**Tabla 5.12. Estadística descriptiva del número de bytes por segundo transmitidos por cada Nodo Local**

<b>Medida</b>	<b>Valor</b>
Media	162.54
Moda	162.86
Mediana	162.86
Desviación estándar	18.58
Varianza	345.09
Error estándar	3.71

## **5. Análisis de resultados**

Para el caso del número de documentos transmitidos por segundo por cada Nodo Local, la Media es de 25.45 documentos/segundo, indicando que el promedio de documentos enviados por segundo por un Nodo Local cualquiera es de aproximadamente 25 documentos/segundo. El hecho de que, en este caso, la desviación estándar para las mediciones realizadas es pequeña, indica que la mayoría de las veces que se transmitan datos documentales entre los Nodos del sistema, se transmitirán 25 documentos/segundo con una variación de aproximadamente 3 documentos/segundo hacia arriba y hacia abajo.

Así, podemos decir que **cada Nodo Local es capaz de transmitir entre 22 y 28 documentos por segundo.**

En cuanto al número de bytes transmitidos por segundo por cada Nodo Local, la Media es de 162.54 KBytes / segundo. Además, debido a que la desviación estándar para las mediciones es pequeña, podemos asumir que la mayoría de las veces se transmiten 162 KBytes/segundo de datos documentales, con una variación de 18.6 Kbytes/segundo hacia arriba y hacia abajo.

Por lo tanto, podemos decir que **cada Nodo Local es capaz de transmitir entre 144 y 180 KBytes por segundo.**

Aunque las mediciones que se muestran son únicamente para el número de documentos o el número de bytes enviados por cada Nodo Local, debo señalar que aplica igualmente al número de documentos y bytes que un Nodo Local puede recibir. Esto es así debido a que la transmisión es síncrona y además la conexión es punto a punto. Esto quiere decir que para que un Nodo Local sea capaz de transmitir un documento o un byte correctamente, el Nodo Local destino debe ser accesible en el instante preciso en que está enviando esta información. De no ser así, ésta no podría ser enviada. Por consiguiente, cada documento o byte enviado correctamente por un Nodo Local implica necesariamente un documento o byte recibido por algún otro Nodo Local.

En consecuencia, es irrelevante la distinción entre documentos o bytes enviados o recibidos. Se puede decir entonces que, dadas las condiciones establecidas para el entorno de pruebas del SCDD, éste es capaz de transmitir entre 22 y 28 documentos por segundo y entre 144 y 180 KBytes por segundo durante su operación normal.

Debo ahora hacer algunos comentarios sobre el entorno de pruebas e indicar los posibles errores ocultos en el mismo, que pudieran restar credibilidad a las mediciones realizadas y a las conclusiones obtenidas.

Debido a la utilización de Xen para la creación de los Nodos Locales y del Nodo Central, todos los nodos se localizan físicamente en la misma computadora, aunque completamente aislados uno de otro. Aún más, desde el punto de vista de un observador externo que hiciera uso de los servicios del Nodo Local, no habría manera de saber si está conectándose a un servidor real o a una instancia ejecutándose dentro de una máquina virtual de Xen.

Además, Xen se encarga de proporcionar la infraestructura de red que conecta a cada una de las 6 instancias que contienen a los nodos del SCDD en el entorno de pruebas. Xen le proporciona a cada instancia una interfaz de red virtual equivalente a una interfaz

ethernet de 100 Mbits/segundo. Convertido a MBytes, sería una interfaz de 12.5 MBytes/segundo. Puede pensarse que esto es un factor que favorece el resultado de las pruebas en ventaja del SCDD. Nada más alejado de la realidad. Veamos. Cada Nodo usa solamente 160 Kbytes/s para la transmisión. Esto es más o menos 0.160 Mbytes/s o el 1.28% del ancho de banda disponible proporcionado por la interfaz de red virtual de Xen.

Por otro lado, una dependencia generalmente contratará un enlace conexión a Internet con algún proveedor de servicios, por ejemplo Telmex o Avantel. Esta conexión será usada por la dependencia para sincronizar su Nodo Local con los Nodos Locales de las otras dependencias, así como con el Nodo Central. Estos proveedores proporcionan enlaces que varían en cuanto a capacidad de conexión como se muestra en la siguiente tabla. Los valores mostrados representan el máximo teórico posible y casi nunca es una velocidad sostenida.

**Tabla 5.13. Capacidad de conexión para enlaces a Internet disponibles públicamente**

Tipo de enlace	Velocidad (KBits/s)	Velocidad (KBytes/s)	Velocidad (MBits/s)
Dial-up	56	7	0.056
Cable 64	64	8	0.064
DSL 128	128	16	0.128
DSL 256	256	32	0.256
DSL 512	512	64	0.512
DSL 1024	1024	128	1.024
DS1 (antes T1)	1544	193	1.544
E1	2048	256	2.048
DS3(antes T3)	44736	5592	44.736

De todos estos posibles tipos de enlace, los más utilizados debido al factor costo-beneficio son los que van desde DSL 128 hasta el E1, donde el E1 será utilizado por dependencias que cuenten con recursos que permitan sufragar los gastos que implica la renta de este servicio.

Además, cabe decir que en los tipos de enlace DSL, en particular los del subtipo ADSL, que son los ofrecidos por Telmex, por ejemplo, la velocidad marcada en la tabla aplica solamente para velocidades de recepción de datos, y no para velocidades de transmisión. Generalmente la relación es 4:1; por lo que, si se contrata un DSL 512, el cliente podrá recibir información a 512 Kbits/s pero solamente podrá enviar información a 128 Kbits/s.

Esto afecta grandemente la capacidad de transmisión de datos de un Nodo Local. Como vimos, un Nodo Local puede enviar información de manera sostenida aproximadamente a 160 KBytes por segundo. Por tanto, si se deseara sincronizar la información documental a esta velocidad de manera sostenida, la dependencia necesitaría contar con un enlace del tipo DS1, de manera que tuviera un ancho de banda de 193 Kbytes/s, 160 de los cuales sería usados por el Nodo Local para sincronizar su información con el

resto del SCDD y los restantes 32 KBytes serían usados por los usuarios de la dependencia para otros usos como envío y recepción de correo electrónico o navegación en la WWW. Cabe mencionar que la utilización del ancho de banda aplica solamente durante la sincronización de datos. En cualquier otro momento, el enlace estará disponible en su totalidad para los usuarios de la dependencia.

En la práctica es raro suponer que todas las dependencias contarán con enlaces del tipo DS1 como mínimo. Lo más común es que la mayoría de ellas cuenten con enlaces DSL de 256 t 512 Mbits/s, debido al costo reducido que implican.

Así pues, las dependencias que cuenten con enlaces T1, o mayores, no se ven afectados por las velocidades de transferencia del Nodo Local y, por tanto, podrán soportar las transmisiones del Nodo Local sin ningún problema. En consecuencia, los resultados obtenidos en esta prueba aplican perfectamente a ellos.

Para el caso de las dependencias que no puedan costear enlaces tan grandes como un T1, podrán seguir operando, aunque las transmisiones de información del Nodo Local se verán influenciadas por la velocidad del enlace.

En este caso, la velocidad de transmisión será solamente la máxima permitida por el enlace a Internet. El Nodo Local, a pesar de todo, tendrá capacidad de operar con un mayor rendimiento, sólo que no será utilizado a su capacidad completa. En el momento en que se aumente la capacidad del enlace, el Nodo Local del SCDD operará a su máximo nivel de transmisión y, por tanto, estará siendo usado de manera óptima.

Hay que señalar que los resultados obtenidos muestran el valor mínimo posible en transmisión y recepción de documentos por segundo y de bytes por segundo, para cada Nodo Local.

Como se recordará, en el entorno de pruebas se utilizó equipo que no es de la clase utilizada para servidores de producción. Era más bien una computadora de escritorio, usada con fines de servidor. Es de esperar que las características del Nodo Local real sean determinadas teniendo en cuenta su función como servidor. Por tanto, no es irreal suponer que probablemente tendrá dos procesadores en lugar de uno, al menos 2 GB de memoria RAM, y discos duros de tipo SCSI, probablemente configurados en RAID o algún otro tipo de disposición que permita elevar su rendimiento. Aún así sería un servidor modesto, tomando en cuenta la importancia de la información que controlará, así como el papel que juega en las actividades diarias de la dependencia.

Además, como también se mencionó anteriormente, la base de datos, así como el sistema operativo, no están optimizados. Se utilizaron tal y como vienen de origen. Es de esperar que cada dependencia tendrá un administrador encargado de monitorear el funcionamiento tanto del servidor como de la base de datos. Podrá, por tanto, optimizar el sistema operativo eliminando servicios no utilizados, a fin de usar todo el poder de procesamiento de cómputo posible. Además, puede optimizar la base de datos, ya sea creando índices para acelerar las búsquedas, o configurando la localidad donde se almacenan físicamente las tablas de la base de datos, de manera que si se encuentran en diferentes discos físicos, puedan accederse simultáneamente a mayor velocidad que si estuvieran en el mismo disco físico.



Con estas consideraciones, la capacidad de transmisión de documentos del SCDD se elevará considerablemente, por lo que no es descabellado pensar que se puedan lograr 100 documentos por segundo o más en un Nodo Local bien administrado y optimizado.

Lo mismo se puede decir del enlace de conexión a Internet. No se puede operar un Nodo Local teniendo una conexión de 256Kbits/s y esperar que la información se transmita eficientemente entre los Nodos Locales del SCDD. Aunque este factor no es determinante para el uso del SCDD, ya que se pueden programar las sincronizaciones de datos durante la noche. De esta manera, los usuarios no ven reducida la velocidad de acceso a Internet, ya sea para navegación web o para envío y recepción de correo electrónico. Además, no importará tanto cuánto tarde el proceso de sincronización, ya que se tiene toda la noche para llevarlo a cabo y con el enlace de red usado exclusivamente para este fin. Pero si un Nodo Local se administra correctamente y la base de datos está completamente optimizada, de nada sirve tener capacidad de enviar 100 documentos/segundo, si el enlace de salida a Internet solamente tiene capacidad para enviar 10 documentos/segundo.

Lo que trato de mostrar es que, en realidad, el Nodo Local siempre se podrá configurar y optimizar para lograr un rendimiento mucho mayor que los resultados aquí mostrados. A pesar de esto, el enlace de acceso a Internet será el cuello de botella que impedirá utilizar al 100% la capacidad del Nodo Local.

Por tanto, los resultados aquí presentados reflejan de manera práctica y realista el rendimiento que se puede esperar del SCDD durante la sincronización de la información documental que se genere en su operación normal.

## Capítulo 6. Recomendaciones

En este capítulo se hacen recomendaciones y posibles mejoras para una segunda etapa del Sistema de Control Documental.

### 1. Compresión de datos

Actualmente toda la información transmitida entre los Nodos Locales del SCDD se envía como texto plano. Esto es así debido a que se transmite como un flujo de datos XML. La idea detrás de XML es que la información se codifique usando dicho tipo de texto.

Es ampliamente conocido que la información codificada en texto plano es ineficiente y no hace un uso óptimo de los recursos. Por otro lado, la compresión usando el algoritmos de compresión como zip, gzip o bzip2 permite comprimir texto hasta en un 85%. Por tanto, si se integrara una fase de compresión al flujo de datos XML justo antes de la transmisión de la información, así como una fase de descompresión justo después de la recepción de la información, se podría hacer un mejor uso de la capacidad del enlace a Internet de los nodos que conforman al SCDD.

Suponiendo que se lograra un nivel de compresión del 60%, entonces los datos ocuparían el 40% del tamaño original y por tanto, utilizarían el 40% del ancho de banda usado anteriormente. Si antes se utilizaban 160Kbytes/segundo, con la compresión de los datos se utilizarían 64KBytes/s.

Por tanto, si con 64 Kbytes/segundo podemos enviar 25 documentos por segundo entonces, si continuáramos usando un ancho de banda de 160KBytes/segundo, con 160KBytes/segundo podríamos enviar 62.5 documentos comprimidos por segundo entre los Nodos del SCDD.

Esta compresión puede tener algún impacto en el uso del procesador por parte del Nodo Local, pero dada la capacidad de cómputo de los procesadores actuales, este impacto es mínimo comparado con el impacto que el ancho de banda tiene sobre la capacidad de transmisión de datos del Nodo Local. Tomando en cuenta estas consideraciones, es absolutamente conveniente agregar esta mejora al SCDD.

### 2. Cifrado de datos

La información que se transmite entre los nodos del SCDD, aunque pública, debido a la Ley de Transparencia y Acceso a la Información Pública Gubernamental, puede ser utilizada para fines ilegales o para obtener ventajas de algún tipo si es interceptada durante su travesía por Internet entre nodo y nodo.

Por tanto, es aconsejable agregar una fase de cifrado de datos entre los nodos que están sincronizando información documental, a fin de evitar este tipo de problemas.

El tipo de cifrado más conveniente sería uno que utilizara llaves públicas, de manera que cada nodo del SCDD pueda comunicarse con cualquiera de los otros y evitar el engorroso proceso de intercambio de llaves. Si se utilizara un cifrado simétrico, el

número de llaves a manejar crecería hasta un punto inmanejable conforme aumente el número de nodos del SCDD.

### **3. Autorización y autenticación de partes**

Otro punto a considerar seriamente es la adquisición de certificados electrónicos que permitan a cada nodo del SCDD comprobar la autenticidad del nodo al que trata de conectarse o para decidir si se autoriza la conexión al nodo que intenta enviarle información.

Esto es un punto importante por la misma razón que hizo recomendable el cifrado, a saber, que la información existente en el SCDD puede ser utilizada para fines no autorizados.

Además, el uso de certificados permite resolver tanto el problema de la autenticación y autorización de las partes comunicantes, como el de cifrado de la información una vez establecida la conexión.

## Capítulo 7. Conclusiones

El Sistema de Control Documental Distribuído cumple de manera completamente satisfactoria los objetivos que se plantearon originalmente y que motivaron su creación.

A saber:

- El SCDD opera correctamente y de manera continúa para sus usuarios en el caso de fallos en la red de comunicaciones.
- El SCDD garantiza la integridad de la información documental contenida en él.
- El SCDD utiliza las herramientas, tecnologías y estándares más modernas tanto para el desarrollo del sistema, como para la transmisión y almacenamiento de la información documental.
- El SCDD es capaz de controlar grandes volúmenes de datos generados por sus usuarios de manera eficiente.
- El SCDD es capaz de transmitir grandes volúmenes de información entre sus distintos nodos utilizando un mínimo de recursos.
- El SCDD está diseñado para crecer y adaptarse a las nuevas necesidades que puedan surgir durante su uso.
- El SCDD es una solución de bajo costo, pero con grandes ventajas tanto en calidad y estabilidad, como en mantenimiento y adaptabilidad.

## Apéndice A. Xen y paravirtualización

Xen es un monitor de máquinas virtuales para la arquitectura x86 basado en paravirtualización. Un monitor de máquinas virtuales es llamado también un hipervisor. Xen puede ejecutar múltiples máquinas virtuales en un solo sistema físico, de manera segura y con un rendimiento muy cercano al 100%.

Xen está disponible para distribuciones basadas en GNU/Linux con kernel versión 2.4 y 2.6, así como para sistemas operativos NetBSD.

Los posibles escenarios de uso para Xen incluyen:

- Desarrollo del kernel.

Se pueden probar y depurar modificaciones al kernel en un ambiente controlado. No hay necesidad de una máquina separada de pruebas.

- Sistemas operativos con distintas configuraciones.

Se pueden ejecutar múltiples sistemas operativos simultáneamente con fines de control de calidad del software o de compatibilidad.

- Consolidación de servidores.

Se pueden mover múltiples servidores en un solo host físico, con aislamiento de fallos y de rendimiento provisto por el hipervisor.

- Cómputo usando *clusters*.

El nivel de granularidad de Xen proporciona mayor flexibilidad que administrar separadamente cada host, pero mejor control y aislamiento que soluciones basadas en un solo sistema por host, particularmente debido a que es posible migrar instancias con fines de balanceo de cargas.

Un sistema Xen está estructurado en capas. La capa con mayores privilegios es Xen mismo. Esta es la capa más baja y es la que se encuentra en contacto directo con el hardware. Xen, por su parte, puede hospedar a varios sistemas operativos, cada uno ejecutándose dentro de una máquina virtual. Cada una de estas máquinas virtuales es llamada un dominio.

La tarea de Xen, por tanto, es programar y asignar tiempos de ejecución para cada una de estas máquinas virtuales, de manera que cada una de ellas tenga acceso y haga un uso efectivo de los CPUs físicos disponibles.

Hay dos clases de dominios en Xen. El dominio 0 y los dominios no privilegiados. El dominio 0 (Dom0) es creado automáticamente por el hipervisor durante el arranque del hardware. Tiene privilegios especiales de administración y acceso total a todo el hardware disponible, tal como discos duros, CPUs, interfaces de red, etc.

Los dominios no privilegiados (DomU), por otro lado, no tienen acceso al hardware físico, sólo a representaciones virtuales de ellos. Por ejemplo, si un dominio no privilegiado desea transmitir información a través de una interfaz ethernet, lo hará enviándola a una interfaz de red virtual que es creada por el dominio 0 para uso exclusivo de este dominio no privilegiado.

Por tanto, el dominio 0 se encarga de presentar a los dominios no privilegiados una abstracción del hardware existente. Además, se encarga de otras tareas administrativas, tales como suspender, reanudar o migrar máquinas virtuales.

El dominio 0 puede crear cualquier número de dominios no privilegiados. Para un usuario externo que intentara conectarse a través de internet a uno de estos dominios no privilegiados, le parecería que está conectándose realmente con un sistema operativo que se ejecuta de manera exclusiva sobre cierto hardware. Xen y el dominio 0 se encargan de mantener esta apariencia en todo momento.

## Apéndice B. Código fuente

En la generación de los documentos usados para las pruebas de replicación en el capítulo 5, se desarrolló un programa especializado. El programa fue desarrollado en la forma de un servlet que puede ser accedido a través de un navegador o usando alguna herramienta especializada que pudiera emular una conexión http, tal como el programa **wget** disponible en Debian GNU/Linux.

El comando utilizado para iniciar la generación de los documentos fue la siguiente:

```
$ wget -T15 -t1 -v -O/dev/null
http://localhost:8080/servlet/replicacion.CrearDocs?p_cveusuario=UUU&p
_dependencia=DDD
```

Donde UUU es la clave del usuario que enviará el documento y DDD es la clave de dependencia donde labora. Ambos valores se encuentran en la tabla *Información del SCDD para los nodos*, del capítulo 5.

El código fuente del *servlet* se muestra a continuación:

```
/*
*****
*****
* $Id: CrearDocs.java,v 1.1 2005/11/03 00:54:28 miguel Exp $
*****
*****
* Derechos Reservados © 2003-2005, Strategic Analysis de México,
S.A.
* http://www.sadm.com.mx
*****
*****
* Descripción:
*
*
*
*
*
*
* Notas:
* El manejo, cambio y actualización de este archivo está orientado a
un perfil
* con conocimientos en HTML, JSP's, CSS y JAVA.
*
*****
*****
*
* Fecha de Expiración:
* Dos años después de la fecha de la última actualización
*
* La Secretaría de Salud se reserva el derecho de modificar y
* actualizar este archivo.
*
*/
```

```

*****
*****
*/

package replicacion;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CrearDocs extends HttpServlet {

public class CrearDocs extends HttpServlet {
private ssa.catalogo.ServletBean sbc = new ssa.catalogo.ServletBean();
private ssa.documento.ServletBean sbd = new
ssa.documento.ServletBean();

// Número de documentos a generar
private static final int NUM_DOCUMENTOS = 10000;
// Número de anexos por documento a generar
private static final int NUM_ANEXOS = 10;

private static final String personas[] = { "3446", // alva
"3447", // dante
"3448", // cesar
"3445", // german
"3449" }; // gerardo

// Tipo de contenido por default para el servlet
private static final String CONTENT_TYPE = "text/html;
charset=UTF-8";

public void doPost(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException
{
response.setContentType(CONTENT_TYPE);

PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>Sistema de Control
Documental</title></head>");
out.println("<body>");
out.println("<p>Este método no está permitido para el Control
Documental</p>");
out.println("</body></html>");
out.close();

return;
}

// Obtiene la clave de la clasificación en base
// a las subclasificaciones especificadas
private String obtenerCveClasificacion(String cveClasificacion1,
String cveClasificacion2) {

// Código no mostrado

```



```

}

// Obtiene la clave de la dependencia donde se está
// registrando el documento
private String cveUnidadRegistro() {
    // Código no mostrado
}

// Obtiene el siguiente número de documento disponible para
registrar
// el documento.
private String siguienteCveDocumento() {
    // Código no mostrado
}

// Registra un documento en la base de datos local.
private boolean insertarDocumento(String cveUsuario, String
cveUnidadRegistroDocumento,
                                String cveDocumento, String
cveTipoDocumento, String tiempoVencimiento,
                                String cveUnidadTiempo, String
numOficio, String fechaDocumento,
                                String cveClasificacion, String
resumen, String palabrasClave,
                                String observaciones, String
cveRemitente, String cveDestinatario){
    // Código no mostrado
}

// Actualiza el status de un documento.
private boolean actualizarStatusDocumento(String cveUsuario,
String cveUnidadRegistroDocumento,
                                String cveDocumento,
String status){
    // Código no mostrado
}

// Registra un anexo para un documento existente en la base de
datos local.
private boolean insertarAnexoDocumento(String cveUsuario, String
cveUnidadRegistroDocumento,
                                String cveDocumento, String
cveAnexoDocumento,
                                String cveTipoAnexo, String
descripcionAnexo) {
    // Código no mostrado
}

// Genera NUM_DOCUMENTOS, cada uno con NUM_ANEXOS,
// registrados por el usuario cveUsuario y con origen en la
// dependencia cveDependencia
private void generaDocumentos(String cveUsuario, int
cveDependencia) {
    // Variables para el documento
    String cveUnidadRegistroDocumento = "";
    String cveDocumento = "";
    String cveClasificacion = "";
    String cveTipoDocumento = "";
    String fechaDocumento = "";
    String numOficio = "";
    String tiempoVencimiento = "";

```

```

String cveUnidadTiempo = "";
String cveRemitente = "";
String cveDestinatario = "";
String clasificacion1 = "";
String clasificacion2 = "";
String resumen = "";
String palabrasClave = "";
String observaciones = "";

// Variables para los anexos del documento
String cveAnexoDocumento = "";
String cveTipoAnexo = "";
String descripcionAnexo = "";

////////////////////////////////////
// Inicia generación de documentos //
////////////////////////////////////

Date tInicio = new Date();

String remitente = personas[cveDependencia-1];

String destinatario = null;

// Información que es constante para los documentos generados
cveTipoDocumento = "2"; // Carta
fechaDocumento = "10/10/2005";
tiempoVencimiento = "2";
cveUnidadTiempo = "1"; // Días
clasificacion1 = "23"; // Adicciones y Salud mental
clasificacion2 = "410"; // Atención y Control de
Transtornos Mentales y Neurológicos

// Información que es constante para los anexos generados
cveTipoAnexo = "1"; // Documento en papel

// Clave de la clasificación para las subclasificaciones
especificadas
cveClasificacion = obtenerCveClasificacion(clasificacion1,
clasificacion2);

// La unidad de registro es la clave de la unidad
administrativa
// donde se están registrando los documentos
cveUnidadRegistroDocumento = cveUnidadRegistro();

int j = 0;
destinatario = personas[j];

// Generamos los documentos
for (int i = 1; i <= NUM_DOCUMENTOS; i++) {

// Cada 25% de documentos cambiamos de destinatario
switch(i) {
case NUM_DOCUMENTOS * 25 / 100:
case NUM_DOCUMENTOS * 50 / 100:
case NUM_DOCUMENTOS * 75 / 100:
j++;
destinatario = personas[j];
break;
}
}

```

```

    }

    // Si el destinatario pertenece a la dependencia
    // para la que estamos generando documentos

    // lo saltamos
    if (j == cveDependencia - 1) {
        j++;
        destinatario = personas[j];
    }

    String tag =
        (i < 10 ? "000" + i :
         (i < 100 ? "00" + i :
          (i < 1000 ? "0" + i : "" + i)
         )
        );

    //Date inicio = new Date();

    //////////////////////////////////////
    // Registro del documento //
    //////////////////////////////////////

    // Información que no cambia para cada documento
    numOficio = "Doc-" + tag;
    cveRemitente = remitente;
    cveDestinatario = destinatario;
    resumen = "Documento auto-generado #" + tag;
    palabrasClave = "documento, generado, " + tag;
    observaciones = "es muy fácil generar documentos";

    // Obtenemos una clave para el nuevo documento
    cveDocumento = siguienteCveDocumento();

    // Tratamos de insertar el documento
    if(!insertarDocumento(cveUsuario,
cveUnidadRegistroDocumento, cveDocumento,
cveUnidadTiempo, numOficio,
resumen, palabrasClave,
cveDestinatario)){
        System.err.println("No se insertó el documento.");

        continue;
    }

    // Actualizamos el status del documento a En Registro
    if(!actualizarStatusDocumento(cveUsuario,
cveUnidadRegistroDocumento,
"EnRegistro")){
        System.err.println("No se insertó el status del
documento a EnRegistro");
        continue;
    }

    //////////////////////////////////////

```

```

// Registro de los anexos para el documento //
//
// Generamos los anexos
for (int anexo = 1; anexo <= NUM_ANEXOS; anexo++) {

    // Información que no cambia para cada anexo
    descripcionAnexo = "Anexo " + anexo + " del documento
" + tag;

    // Obtenemos una clave para el nuevo anexo
    cveAnexoDocumento =
sbd.obtenerSiguieteCveAnexoDocumento();

    // Insertamos el anexo del documento
    if(!insertarAnexoDocumento(cveUsuario,
cveUnidadRegistroDocumento, cveDocumento,
                                cveAnexoDocumento,
cveTipoAnexo, descripcionAnexo)) {
        System.err.println("No se insertó el anexo al
documento.");
        continue;
    }

    // Actualizamos el status del documento.
    if(!actualizarStatusDocumento(cveUsuario,
cveUnidadRegistroDocumento,
                                cveDocumento, "Alta")){
        System.err.println("No se cambió el status del
documento de EnRegistro a Alta");
        continue;
    }

    //Date fin = new Date();

    //System.out.println("Doc: " + tag + " tiempo: " +
(fin.getTime() - inicio.getTime()) + "ms");
}

    Date tFin = new Date();
    System.out.println("Tiempo total: " + (tFin.getTime() -
tInicio.getTime())/1000.0 + "s");

    return;
}

public void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException
{

    // Variables para almacenar los parámetros
    String p_cveusuario = "";
    String p_cvemenu = "";
    String p_dependencia = "";
    int dependencia = -1;

    // Especificamos que queremos los parámetros como UTF-8

```

```
request.setCharacterEncoding("UTF-8");

// Guardamos los parametros en las variables correspondientes
p_cveusuario = request.getParameter("p_cveusuario");
p_cvemenu = request.getParameter("p_cvemenu");
p_dependencia = request.getParameter("p_dependencia");

try {
    dependencia = Integer.parseInt(p_dependencia);
} catch (NumberFormatException nfe) {

System.err.println("NumberFormatException(ssa.replicacion.CrearDocs):"
+ nfe);
    nfe.printStackTrace();
}

// Generamos los documentos
generaDocumentos(p_cveusuario, dependencia);

return;
}
}
```

## Apéndice C. Generación de documentos de prueba

Para la realización de las pruebas detalladas en el capítulo 5 de esta obra se generaron gran cantidad de documentos de manera automatizada, de manera que se pudieran realizar mediciones en las velocidades de transmisión de documentos entre los distintos nodos del entorno de pruebas.

En el apéndice anterior se muestra el código fuente utilizado para generar los documentos en cada uno de los cinco nodos existentes. Antes de generar los 10,000 documentos en cada nodo se realizaron varias pruebas con diferentes cantidades de documentos, con el fin de estimar la capacidad del SCD para crear documentos.

En la siguiente tabla se muestran los resultados obtenidos en los distintos Nodos Locales al generar un número variable de documentos. Las unidades están en segundos

**Tabla C.1. Duración de la generación de documentos de prueba en los Nodos Locales del SCDD**

Número de documentos	Número de anexos por documento	host01	host02	host03	host04	host05	Promedio	Documentos por segundo	Promedio de Documentos por segundo
100	3	2.538	2.994	2.903	2.872	2.823	2.826	35.39	
500	3	14.038	14.324	14.142	14.139	14.217	14.172	35.28	
1,000	3	28.038	29.955	28.127	26.635	27.11	27.973	35.75	
10,000	3	283.864	291.144	304.877	283.721	288.695	290.46	34.43	<b>35.21</b>
100	10	4.972	5.293	4.622	4.647	4.652	4.8372	20.67	
500	10	23.795	26.274	23.892	23.701	23.062	24.1448	20.71	
1,000	10	47.744	46.352	46.888	56.22	49.935	49.4278	20.23	
10,000	10	476.293	486.643	533.661	519.99	507.73	504.863	19.81	<b>20.36</b>

Como se ve en la tabla, en promedio se pueden crear **35.21 documentos por segundo**, con **3 anexos** por documento. Además, se pueden crear **20.36 documentos por segundo**, si cada uno de estos documentos tiene **10 anexos**.

Estas mediciones muestran la capacidad de generación de documentos que proporciona el Sistema de Control Documental. Es difícil, aunque no imposible, que una dependencia logre mantener una tasa de creación de documentos de esta magnitud de manera sostenida a lo largo de un día de trabajo. Además, en este caso aplican las mismas consideraciones que se discutieron en el capítulo 5 respecto a las características del hardware del Nodo Local, así como de la optimización del mismo. Estos resultados muestran el peor caso posible para un Nodo Local usando hardware de características modestas. Es de esperar que una dependencia utilice equipo de cómputo de mejor

calidad y, por tanto, que los resultados presentados en esta sección mejoren sustancialmente.

# Glosario

## B

### Base de Datos Central

Base de datos relacional que existe en el *Servidor Central*. Su función es mantener una copia completa de toda la información existente en el SCDD, la cual se encuentra esparcida en los diferentes Servidores Locales de las Dependencias.

Contiene siempre la información más actualizada posible del SCDD y es el lugar idóneo para efectuar los procedimientos de respaldo de la información.

### Base de Datos Local

Base de datos relacional que existe en el *Servidor Local* de una *Dependencia*. Su función es almacenar información de manera que es posible para el *Servidor Local* continuar su operación de manera ininterrumpida y segura en el caso de un fallo en la red.

La información de esta base de datos es sincronizada a intervalos regulares con la *Base de Datos Central*.

### Base de Datos Única

Base de datos relacional que existe en el *Servidor ÚnicoD* y que contiene toda la información existente en el SCD.

### BDC

Ver Base de Datos Central.

### BDL

Ver Base de Datos Local.

## D

### Dependencia

Una dependencia es una unidad administrativa o una organización como un todo. Se refiere a una entidad lógica más que física. Describe a un conjunto de usuarios que trabajan dentro del mismo edificio o son parte de la misma empresa. p.e. SSA, Dirección General de Tecnologías de la Información, Instituto Nacional de Enfermedades Respiratorias, etc. Generalmente se reconocen porque tienen un *Servidor Local* dentro de sus instalaciones para atender a sus usuarios.



## Documento

Un documento es un nombre genérico para denotar al toda la información escrita que se crea y circula diariamente dentro de una *dependencia*. Puede tratarse de un Memorándum, una Carta, una Circular, un Aviso, una Minuta, etc. En general es todo aquél medio impreso que contiene información de utilidad y con valor para una dependencia y que es susceptible de ser almacenada en formato electrónico para su posterior consulta.

## S

### SC

Ver Servidor Central.

### SL

Ver Servidor Local.

### SU

Ver Servidor Único.

### SCD

Ver Sistema de Control Documental.

### SCDD

Ver Sistema de Control Documental Distribuído.

## Secretaría de Salud

Secretaría de Salud de México.

## Servidor Central

Servidor globalmente conocido y accedido por los servidores locales para concentrar la información del SCDD. Contiene la información completa existente en el SCDD y de la cual cada Servidor Local contiene solamente una parte. Es utilizada como centro lógico y operativo más que geográfico del SCDD.

Todos los Servidores Locales sincronizan su información al Servidor Central a intervalos regulares, de manera que la información es siempre la más actualizada posible.

## Servidor Local

Servidor localizado dentro de una dependencia y que tiene como función el brindar a sus usuarios servicios tales como crear documentos, responder documentos y delegar documentos.

Contiene una *Base de Datos Local* que sirve para continuar operando en el caso de una interrupción en las comunicaciones de red con el Servidor Central. Esta base de datos es sincronizada a intervalos regulares con la *Base de Datos Central* del una vez que se restablecen el enlace, así como a intervalos regulares durante su funcionamiento.

### Servidor Único

En el SCD solamente existe un solo servidor, llamado Servidor Único, que tiene la tarea de brindar a todos los usuarios de las diferentes dependencias los servicios de creación, respuesta y delegación de documentos. Este servidor contiene la Base de Datos Única que es donde se encuentra concentrada toda la información del SCD. Para poder acceder a este servidor desde una dependencia debe haber siempre una conexión de red disponible.

### Sistema de Control Documental

Nombre del sistema de cómputo utilizado por la SSA y cuyo objetivo es mantener una base de datos de información acerca de los documentos, tanto internos como externos, que manejan las dependencias pertenecientes a las SSA. Este sistema permite al usuario crear documentos, responder documentos o delegarlos a algún otro usuario para su atención.

Toda la información se encuentra centralizada en una base de datos y, para interactuar con el sistema, todos los usuarios deben conectarse utilizando un navegador Web.

El sistema es accedido por los usuarios desde su dependencia utilizando la red Internet como canal de comunicación.

### Sistema de Control Documental Distribuido

Nombre de la arquitectura, así como de su implementación, creada con motivo de resolver los problemas que tiene el SCD cuando hay interrupciones en las telecomunicaciones entre una dependencia y la dependencia donde se encuentra hospedado el Servidor Central.

Es una arquitectura distribuida que reduce a niveles aceptables el impacto de una falla en las telecomunicaciones entre dependencias y que les permite continuar su operación cotidiana, tal como crear documentos, responderlos o delegarlos, aún en el caso de que la interrupción del enlace de Internet dure varias horas o hasta días.

Es una arquitectura que mantiene la información distribuida entre diversos servidores locales y que se encarga de mantenerla sincronizada de manera automática y confiable.

SSA

Ver Secretaría de Salud.

# Bibliografía

## Libros

Peter Eeles, Kelli Houston, y Wojtek Kozaczynski. Copyright © 2003 Pearson Education, Inc. 0-201-79166-8. Addison-Wesley Pearson Education. *Building J2EE applications with the Rational Unified Process*.

Deepack Alur, Dan Malks, y John Crupi. Copyright © 2003 Prentice Hall PTR. 0-131-42246-4. Prentice Hall PTR. *Core J2EE Patterns: Best Practices and Design Strategies, 2nd Edition*.

Marty Hall. Copyright © 2000 Prentice Hall PTR. 0-130-89340-4. Prentice Hall PTR. *Core Servlets and JavaServer Pages*.

## Internet

Eric Armstrong, Jennifer Ball, Stephanie Bodoff, Debbie Bode Carson, Ian Evans, Dale Green, Kim Haase, y Eric Jendrock. Copyright © 2004 Sun Microsystems, Inc. Sun Microsystems, Inc. *The J2EE 1.4 Tutorial, 2nd Edition*. URL:  
<http://java.sun.com/docs/books/j2eetutorial/index.html#second>.

Copyright © 2005 Sun Microsystems, Inc. Sun Microsystems, Inc. *The Java Web Services Tutorial v1.6*. URL:  
<http://java.sun.com/webservices/docs/1.6/tutorial/doc/index.html>.

Inderjeet Singh, Beth Stearns, y Mark Johnson. Copyright © 2002 Sun Microsystems, Inc. Sun Microsystems, Inc. *Designing Enterprise Applications with the J2EE Platform, 2nd Edition*. URL:  
[http://java.sun.com/blueprints/guidelines/designing\\_enterprise\\_applications\\_2e/titlepage.html](http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/titlepage.html).

## Colofón

Esta tesis fue desarrollada entre los meses de septiembre a noviembre de 2005 en una máquina con las siguientes características:

- CPU: Intel Pentium IV 2.0 GHz
- Memoria RAM: 768 MB
- Sistema Operativo: Debian GNU/Linux 3.1 (Sarge), kernel 2.6.8-2-386

Fue escrita en formato DocBook versión 4.3 usando XMLmind XML Editor Standard Edition V2.9p1. El archivo origen se encuentra en el CD anexo bajo el directorio DocBook con el nombre Tesis.xml. Fue convertido a formato PDF usando el siguiente comando:

```
$ xmlto -m stylesheet/customization.xsl pdf Tesis.pdf
```