

MIGRACION AL SOFTWARE LIBRE

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

UCI

Facultad 9

TÍTULO

PROPUESTA DE MIGRACIÓN A SOFTWARE LIBRE
DE SEÑAL ACN

TRABAJO PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

AUTORES: Dayara Cobas Peláez

Daylin Pérez Grave de Peralta

TUTOR: Ing. Yunior Montaner Hernández.

Ciudad de La Habana, julio de 2008

Año 50 de la Revolución.

El futuro tiene muchos nombres:

Para los débiles es lo inalcanzable,

Para los temerosos, lo desconocido,

Para los valientes es la oportunidad.

Victor Hugo.

DEDICATORIA

A mis padres de donde emana toda mi fuerza, voluntad y alegría.

Daylin Pérez Grave de Peralta.

A mi madre quien es motivo de inspiración en mi vida y a mi sobrinito por darme tanta alegría, aún cuando no ha nacido.

Dayara Cobas Peláez



AGRADECIMIENTOS

A mi Madre por depositar en mi toda su confianza, por siempre estar ahí cuando la necesito y por ser tan especial, quiero que sepas que todos mis logros te los debo a ti y que eres mi razón de ser.

A mi abuela por haber marcado mi vida durante el tiempo que disfruté de su cariño.

A mi hermanita que es motivo de alegría y de impulso para seguir adelante.

A mi tío Alvio por ser como un padre para mí, brindándome su cariño y apoyo incondicional.

A mi tía Maidelín por portarse tan linda conmigo y quererme tanto.

A mi primita Yelena por ser tan especial y darme su cariño.

A Carlos por ser apoyo incondicional y cuidar de mi familia.

A mis tíos Vitico y Yamilka que han sido tan buenos.

A mi prima Liudmila por haberse portado tan bien conmigo brindándome su cariño.

A mis compañeras de cuarto, especialmente a Daylín y Yeter, mis amigas incondicionales que forman parte de mis mejores recuerdos universitarios.

A Yuliet y Grethel por su comprensión y cariño todos estos años.

A Orlando, por estar a mi lado y poder contar con todo su apoyo y cariño.

A toda mi familia, que de una forma u otra me han apoyado tanto toda la vida.

A la Revolución y a Fidel, a quien le debo haber hecho realidad mis sueños.

MUCHAS GRACIAS!!!

Dayara Cobas Peláez



AGRADECIMIENTOS

A las personas más importantes en mi vida: mis padres, que siempre me guiaron y alentaron para llegar a donde hoy me encuentro, que si bien no es la meta, es un gran avance en mi vida profesional y personal. A ellos que apoyaron mis decisiones, que se han sacrificado y siempre han estado a mi lado, decirles que este trabajo, con el hoy me gradúo, no es solamente fruto de mi investigación y estudio sino que en él están plasmados toda la formación que como excelentes padres y profesionales me supieron dar...Mi triunfo es de ustedes, Gracias a los dos por estar, Los amo mucho!!!

A mi hermanito no menos importante, que siempre ha hecho de mi mundo un lugar más feliz, a mi hermana que siempre quiso que yo estudiara mucho, sé que hoy estaría orgullosa de mí; he aquí mi meta para ella.

A mi abuelita "mami Emilia" y a mi abuelo Félix por la preocupación y cariño.

A nuestro Comandante en Jefe por todo, por sus ideas claras, precisas y previsorias, por la Revolución que formó y que entre todos hemos mantenido, y que ha hecho posible la realización de proyectos importantes y de carácter social como lo es la UCI, que hoy hace realidad los sueños de muchos.

A las amistades, las de por mi casa y las que hice en la UCI: las del aula y las del apartamento, en especial a Dayara que es también mi compañera de tesis, a Yeter, May y Yusney que hicieron de mi estancia lejos de casa un poco más grata, por hacerme reír tanto, por acompañarme a lo largo de estos 5 fructíferos años; en fin por ser mis amigos.

A Rudy por estar siempre pendiente de mi, por preocuparse y vivir conmigo cada éxito y fracaso sin juzgarme y ser tan paciente cuando tenía que serlo, en fin, por ser, ante todo y después de todo, mi amigo.

A todos los que me ayudaron de alguna forma en la realización de este trabajo, a los que me preguntaron en alguna ocasión: Cómo va la Tesis?, a todos los que de alguna forma olvido.

MUCHAS GRACIAS!!!

Daylin Pérez Grave de Peralta

DECLARACIÓN DE AUTORÍA

Por este medio declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de julio del año 2008.

Dayara Cobas Peláez
Autor

Daylin Pérez Grave de Peralta
Autor

Ing. Yunior Montaner Hernández.
Tutor

DATOS DE CONTACTO

Tutor: Ing. Yunior Montaner Hernández.

Profesor de la Universidad de las Ciencias Informáticas, Departamento de Técnicas de Programación.
Líder de Desarrollo de Software del Proyecto "Plataforma de Televisión Informativa", Polo Productivo
"Video y Sonido Digital", Facultad 9, UCI.

Dirección: Universidad de las Ciencias Informáticas (UCI), Edificio: 14 Apto: 14101.

Teléfono Laboratorio: +53 – 7 – 8378283 Teléfono Apto: +53 – 7 – 8358466

E-mail: ymontaner@uci.cu

Resumen

Desde el triunfo revolucionario el primero de enero de 1959, el gobierno cubano se ha preocupado por lograr y mantener una total independencia en su gobierno; para el cumplimiento de este principio ha tomado medidas en todos los sectores del país incluyendo el mundo de la informática donde ha adoptado la política de llevar a cabo paulatinamente la migración de todos sus sistemas al software libre (SWL) para cambiar de esta forma la dependencia que tienen sus agencias estatales del software de Microsoft bajo licencia y en vez de ello utilizar SWL desarrollado por una comunidad global de programadores que comparten su código en forma gratuita.

El presente trabajo tiene como objetivo fundamental el de (proponer una estrategia para la migración del canal Señal ACN (Agencia Cubana de Noticias) a SWL), lo que constituye un paso importante en el logro de la independencia tecnológica de nuestro país sobre el monopolio casi absoluto de Microsoft en el campo de los sistemas operativos y software propietario, el alto costo de las licencias de software y los problemas de seguridad.

En este documento se plasman los resultados del estudio realizado sobre el funcionamiento e importancia histórica del canal Señal ACN, toda la investigación acerca de software libre para llevar a cabo el proceso de migración, las implicaciones del costo de la misma y el plan que guiará el proceso.

PALABRAS CLAVES

Migración, Software, Software Libre, Señal, Teletexto, Canal

TABLAS Y FIGURAS

TABLA 1 SISTEMAS Y APLICACIONES DESARROLLADAS UTILIZANDO MONO (15).....	24
TABLA 2 COMPONENTES Y LIBRERÍAS DESARROLLADAS UTILIZANDO MONO (16).....	25
TABLA 3 MATRIZ DAFO PARA LLEVAR A CABO LA MIGRACIÓN.....	62
TABLA 4 MATRIZ DAFO PARA NO LLEVAR A CABO LA MIGRACIÓN.....	64
TABLA 5 REQUISITOS DE HARDWARE.....	65
TABLA 6 PROPUESTA DE MIGRACIÓN.....	66
TABLA 7 CURSOS PARA EL ADIESTRAMIENTO.....	68
TABLA 8 PASOS DE LA ETAPA IV DE MIGRACIÓN.....	70
FIG. 1 ARQUITECTURA DE SEÑAL ACN. (2).....	8
FIG. 2 ARQUITECTURA DE MONODEVELOP (22).....	31
FIG. 3 LOS FRAMEWORKS MÁS POPULARES DE AJAX. (23).....	37
FIG. 4 SERVIDORES WEB MÁS UTILIZADOS. (24).....	38



INDICE

INTRODUCCIÓN	1
CAPÍTULO I: SISTEMA AUTOMATIZADO DE TELETXTO PARA LA PLATAFORMA DE TELEVISIÓN DIGITAL SATELITAL CUBANA.	7
INTRODUCCIÓN	7
CANAL SEÑAL ACN	7
ARQUITECTURA DE SEÑAL ACN.	7
MÓDULOS O SUBSISTEMAS DE SEÑAL ACN.	8
<i>Módulo de Administración</i>	8
<i>Módulo de Actualización de Recursos Multimedia</i>	9
<i>Módulo de Transmisión</i>	9
SEGURIDAD EN EL GESTOR DE CONTENIDOS DE SEÑAL ACN.	9
APORTE SOCIAL QUE TIENE SEÑAL ACN EN LA ACTUALIDAD.	9
DESCRIPCIÓN DE HERRAMIENTAS EMPLEADAS EN EL DESARROLLO DE SEÑAL ACN.	10
HTML	10
Java Script.....	11
Active Server Page (ASP).....	12
Lingo.....	13
C Sharp (C#)	13
SISTEMA DE GESTIÓN DE BASES DE DATOS.	14
SQL Server.....	14
Microsoft SQL Server 2000.....	14
ALTERNATIVAS EXISTENTES PARA LA MIGRACIÓN DE SISTEMAS CON LAS CARACTERÍSTICAS DE SEÑAL ACN A SOFTWARE LIBRE.	15
CONCLUSIONES PARCIALES CAPÍTULO I.	15
CAPÍTULO II: ESTUDIO PARA LA MIGRACIÓN.	16
INTRODUCCIÓN.....	16
POSICIÓN DE CUBA ANTE EL USO Y LA MIGRACIÓN A SOFTWARE LIBRE.....	16
RAZONES Y NECESIDADES PARA HACER LA MIGRACIÓN DE SEÑAL ACN.....	18
MÓDULOS DE ACN. PROPUESTA PARA LA MIGRACIÓN.	21
2.1.1 <i>Módulo de Actualización de Recursos Multimedia</i>	21
2.1.1.1 Plataformas Software Libre. Propuesta para la migración.	21
2.1.1.1.1 Java	21
2.1.1.1.2 Mono.....	22
2.1.1.1.3 Selección de la Plataforma Libre para hacer la migración.....	22
2.1.1.2 IDEs Libres. Propuesta para la migración.	26
2.1.1.2.1 C# Develop.....	27
2.1.1.2.2 Eclipse.....	27
2.1.1.2.3 Selección del IDE Libre para hacer la migración.	28
2.1.1.3 Lenguaje Libre. Propuesta para la migración.	28
2.1.1.3.1 C#	28
2.1.1.3.2 Python	29
2.1.1.3.3 Selección del lenguaje libre para hacer la migración.	29
2.1.2 <i>Módulo de Administración</i>	29
2.1.2.1 IDEs Libres. Propuesta para la migración.	30
2.1.2.1.1 Eclipse.....	30
2.1.2.1.2 MonoDevelop	30

2.1.2.1.3	Selección del IDE Libre para hacer la migración.....	31
2.1.2.2	Lenguajes Libres. Propuesta para la migración.....	32
2.1.2.2.1	PHP.....	32
2.1.2.2.2	Java.....	32
2.1.2.2.3	Selección del lenguaje libre para hacer la migración.....	32
2.1.3	<i>Módulo de Transmisión. Propuesta para realizar la migración.....</i>	34
SERVIDORES WEB LIBRES. PROPUESTA PARA LA MIGRACIÓN.....		37
2.1.4	<i>Apache.....</i>	37
2.1.4.1	Características más importantes del Servidor Web Apache 2.....	40
2.1.5	<i>El Servidor Web que se propone para la migración.....</i>	43
SERVIDORES DE BASE DE DATOS LIBRES. PROPUESTA PARA LA MIGRACIÓN.....		44
2.1.6	<i>Caracterización de los sistemas de gestores de Bases de Datos libres (BDL) más difundidos.....</i>	44
2.1.6.1	PostgreSQL.....	44
2.1.6.1.1	Características de PostgreSQL.....	44
2.1.6.1.2	PostgreSQL 8.2.....	46
2.1.6.2	Firebird.....	49
2.1.6.2.1	El gestor de base de datos que se propone para la migración.....	49
SISTEMA OPERATIVO. PROPUESTA PARA LLEVAR A CABO LA MIGRACIÓN.....		50
CONCLUSIONES PARCIALES CAPÍTULO II.....		51
CAPÍTULO III DESARROLLO DE LA MIGRACIÓN.....		52
3.1	INTRODUCCIÓN.....	52
3.2	ANÁLISIS DE COSTOS Y BENEFICIOS.....	52
3.3	PLAN DE MIGRACIÓN.....	66
3.4	EL MÓDULO PILOTO.....	71
3.5	CONCLUSIONES PARCIALES CAPÍTULO III.....	74
CONCLUSIONES GENERALES.....		75
RECOMENDACIONES.....		76
BIBLIOGRAFÍA CONSULTADA.....		77
REFERENCIA BIBLIOGRÁFICA.....		78
ANEXOS.....		81
GLOSARIO DE TÉRMINOS.....		86

Introducción

La Revolución Cubana triunfó el primero de enero de 1959 y desde entonces ha sido hostigada por un bloqueo que se ha mantenido hasta la actualidad, implantado por el gobierno de Estados Unidos (EE.UU.), que restringe el comercio de cualquier índole con el país.

La rama de la Informática también se ve afectada por este comportamiento que se refleja en su desarrollo porque el paso de computadoras se ve interceptado, al igual que el de cualquier otro tipo de hardware e incluso de software, evidenciándose también en las descargas, ya que cada vez son más bloqueadas, por lo que el pueblo cubano se encuentra privado de ellos y si llegaran a adquirir cualquier software(SW) seguirían los problemas por concepto de actualizaciones y a el lento enlace satelital con el que cuenta la isla.

La Agencia Cubana de Noticias (ACN) unido al trabajo en común de la Universidad de las Ciencias Informáticas (UCI), e importantes entidades cubanas del área de las telecomunicaciones y por iniciativa del Comandante en Jefe Fidel Castro se creó el canal televisual de noticias llamado Señal ACN, que se considera un nuevo soporte informativo creado para trasladar a la pantalla digital materiales cubanos de la prensa escrita y mensajería con una visión dinámica de Cuba que tienen como destinatarios a los miles de internacionalistas en Venezuela, y que hoy llega, no sólo, a esa nación sudamericana, sino también a los cooperantes en Bolivia, Honduras, Guyana, Belice, Haití, Jamaica, Dominica, Santa Lucía, a otras islas del Caribe y a las llamadas zonas de silencio del archipiélago cubano, y que tiene como objetivo el de expandir la realidad cubana y propiciar el intercambio entre los colaboradores y sus familiares.

Este canal está implementado completamente con herramientas y software privativo (SWP) que tienen licencias muy costosas que el gobierno cubano no se puede dar el lujo de pagar, además este tipo de software puede verse como una amenaza a la seguridad porque, como los usuarios no pueden examinar el código, no tienen forma de saber lo que éste hace o qué aberturas en la "parte trasera" del programa hayan podido dejar abiertas los diseñadores para tener acceso en un futuro, en otros casos es imposible encontrar la causa de un resultado erróneo, producido por un componente cuyo funcionamiento se desconoce.

Richard Stallman la figura más relevante del movimiento por el SWL, programador y fundador del proyecto GNU, expresó y no en vano: "Un programa privativo nunca es de confiar"; a continuación se muestran una serie de desventajas del uso de software propietario:

- Es ilegal extender una pieza de software propietario para adaptarla a las necesidades particulares de un problema específico. En caso de que sea vitalmente necesaria tal modificación, es necesario pagar una elevada suma de dinero a la compañía fabricante, para que sea ésta quien lleve a cabo la modificación a su propio ritmo de trabajo y sujeto a su calendario de proyectos.
- La innovación es derecho exclusivo de la compañía fabricante. Si alguien tiene una idea innovadora con respecto a una aplicación propietaria, tiene que elegir entre venderle la idea a la compañía dueña de la aplicación o escribir desde cero su propia versión de una aplicación equivalente, para una vez logrado esto poder aplicar su idea innovadora.
- Es ilegal hacer copias del software propietario sin antes haber contratado las licencias necesarias.
- En la mayoría de los casos el gobierno se hace dependiente de un solo proveedor. (1)

Todos los anteriores inconvenientes y basándose en el hecho de que la utilización del software libre no implica gastos adicionales por concepto de cambio hardware, así como que la adquisición de cualquiera de sus distribuciones puede hacerse de forma gratuita, o a muy bajos precios y no hay que pagar absolutamente nada por su utilización, distribución y/o modificación, unido a las posibilidades de usar, copiar, estudiar, modificar y redistribuir libremente que brinda esta plataforma que cumple con los preceptos enunciados por la sociedad socialista cubana y la importancia del mismo para que el Estado tenga independencia tecnológica, se dispuso, por parte del gobierno cubano, la migración gradual de todos los sistemas del país a Software Libre(SWL), lo que ha propiciado que el canal migre a este tipo de SW, atendiendo a las utilidades que se desprenden de la propia concepción del funcionamiento y diseño de él; pudieran citarse como las más importantes las que se detallan a continuación:

- Bajo costo: se obtiene por la vía de la compra, su precio ronda el costo de producción por lo que es bajo, aunque es muy común que este se obtenga directamente de Internet de forma gratuita. Los servicios de soporte técnico también tienen un precio mucho más bajo que los de otros software. Esto le aporta competitividad.

- Seguridad: La propia superestructura le confiere una gran robustez en cuanto a los mecanismos de seguridad que implementan para su funcionamiento.
- Privacidad: El conocimiento completo del código fuente permite a su vez la privacidad, por cuanto garantiza la detección de cualquier programa, aplicación o segmento de código que pueda ser puesto en el sistema (como es el caso del *spyware*), muy común en Windows, que envía información a sus creadores o a empresas de marketing sobre las características de la plataforma donde está instalado, aplicaciones instaladas, sitios de Internet visitados, etc.
- Calidad y actualización: Están respaldados por una comunidad mundial de desarrolladores que funciona como tribunal de censura a los fallos que puedan presentarse; la cooperación a escala global permite el perfeccionamiento de todo el software con que se cuenta actualmente, así como el surgimiento de nuevas aplicaciones, y lo que es más importante la distribución de estos resultados de modo casi inmediato a través de las redes de alcance global.

Sin duda alguna, como se ha visto anteriormente el uso del SWL es sustentable en Cuba, a partir de las ventajas que tiene con respecto a los del tipo propietario.

Por esto, su aplicación adquiere una relevante significación que puede verse desde tres ámbitos diferentes:

- Político: Desde un primer punto de vista representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación. En el presente Cuba se encuentra a merced de la empresa norteamericana Microsoft y de otras que ofrecen software bajo licencia, que tienen la capacidad legal de reclamar a Cuba que no siga utilizando un sistema operativo (SO) de su propiedad, basada en leyes de propiedad industrial por las cuales también Cuba se rige; esto provocaría una interrupción inmediata del programa de informatización de la sociedad que como parte de la batalla de ideas está desarrollando el país, además pudiera implementarse una campaña de descrédito a la isla, abogando el uso de la piratería informática por parte de las instituciones estatales cubanas.

Desde un segundo, el software libre representa la alternativa para los países pobres, y es por concepción, propiedad social, si se tiene en cuenta que una vez que comienza a circular

rápidamente se encuentra disponible para todos los interesados sin costo alguno o en su defecto a muy bajo costo.

En tercer lugar es desarrollado de forma colectiva y cooperativa, tanto en su creación como en su desarrollo, tanto cuantitativa como cualitativamente mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad.

- Económico: Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

La adquisición de cualquiera de sus distributivas puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajos precios, se garantiza su explotación con un mínimo de recursos, por cuanto no hay que pagar absolutamente nada por su utilización (no requiere de licencia de uso, las cuales son generalmente muy caras), distribución y/o modificación.

- Tecnológico: Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor porcentaje de efectividad, además la corrección de sus errores de programación y obtención de las actualizaciones y nuevas versiones.

Además con su utilización se fomenta el carácter socialista en todos los aspectos incluyendo la ramas de las Tecnologías de la Información y las Comunicaciones (TIC's) del país, y unimos a la lucha Ideológica en beneficio de toda la humanidad que ayuda a construir una plataforma informática socialista, independiente y soberana, que conjuntamente con los proyectos de hardware libre aporten importantes herramientas para la derrota del capitalismo.

Además se sabe que está en las manos del SWL el desarrollo y soberanía tecnológica de un país, pensando siempre en función del desarrollo endógeno que involucra también al ser humano, al ciudadano y a las comunidades.

Por otra parte no existe una vía establecida para la migración de aplicaciones semejantes a la de Señal ACN a "Software Libre" naciendo el siguiente **problema**:

Falta de una estrategia para la migración de Señal ACN hacia Software Libre.

Llegándose a la conclusión que el **objeto de estudio** es: Estudiar las herramientas y lenguajes del canal Señal ACN y sus homólogas en Software Libre.

Y teniendo como **objetivo general**: Proponer una estrategia para la migración de Señal ACN a Software Libre.

Y como **objetivos específicos** los siguientes:

1. Seleccionar las plataformas y el gestor de base datos dentro de los estudiados para la realizar la migración del sistema.
2. Presentar un plan de migración.
3. Presentar un prototipo funcional para probar la viabilidad del plan.

Enmarcado en el **campo de acción**: El proceso de migración de Señal ACN a Software Libre.

Esperando como posibles resultados: La obtención de un plan para la migración del canal y el desarrollo de un prototipo funcional del módulo de Transmisión para demostrar la viabilidad de la propuesta realizada.

Los autores consideran como **hipótesis**: Si se realiza un profundo estudio y análisis de las tecnologías en que fue implementado Señal ACN y sobre las homólogas en Software Libre, se podrá definir un plan que sirva de guía para la migración del canal a Software Libre.

Destacándose como **variable independiente**: un profundo estudio de las tecnologías en que fue implementado Señal ACN y sobre las homólogas en Software Libre.

Y como **variable dependiente**: definir un plan que sirva de guía para la migración del canal a Software Libre.

Para darle cumplimiento al objetivo trazado y obtener resultados satisfactorios se determinó que las **tareas** a realizar en esta investigación estarían dirigidas a:

1. Realizar un estudio de la actual plataforma de ACN.
2. Realizar estudio sobre las alternativas existentes para migrar Aplicaciones WEB.
3. Realizar un estudio de Gestores de Bases de Datos (BD) libres existentes.



4. Realizar un estudio de los Lenguajes y Entornos Integrado de Desarrollo (IDEs) libres.
5. Realizar un análisis de Costos-Beneficios sobre la migración.
6. Realizar un plan de migración.
7. Presentar prototipo funcional para probar la viabilidad del plan de migración.

Para el desarrollo de la investigación se utilizaron los métodos teóricos siguientes: el histórico-lógico, Analítico-Sintético, Hipotético-Deductivo y el causal y de los empíricos el de la observación. A continuación se especifica el por qué de la selección de los mismos:

- Se aplicará el “Método Histórico” para investigar si existe un proceso planteado en la actualidad, que guíe el proceso de migración de software privativo hacia software libre, y de ser así se indagará en función de conocer a ciencia cierta cuáles son los resultados de la aplicación de dicho proceso en la migración de un sistema determinado.
- El método Analítico-Sintético: para analizar los documentos de importancia en la ACN donde se puede extraer la información relacionada con el canal que sea útil para el proceso de migración.
- El método Hipotético-Deductivo: este método fue utilizado para la verificación de la hipótesis y el surgimiento de un nuevo conocimiento a partir de ésta.
- El método “Causal” para estudiar los factores que provocan la necesidad de un proceso de este tipo.
- El método empírico de la “Observación”.

Capítulo I

Capítulo I: Sistema Automatizado de Teletexto para la Plataforma de televisión Digital Satelital Cubana.

Introducción.

El objetivo fundamental de este capítulo es abordar aspectos que se utilizan como soporte teórico para la migración del Canal ACN a Software Libre, de igual manera se describen sus principales características, arquitectura que utiliza, así como módulos o subsistemas que se utilizaron para su realización, el proceso para llevar a cabo su gestión de seguridad, además aportes sociales en la actualidad y herramientas utilizadas. También se analizan algunas alternativas existentes para la migración de sistemas con las características de Señal ACN a Software Libre. (2)

Canal Señal ACN.

Es una aplicación informática que permite la gestión de las noticias y recursos multimedia del sistema de teletexto, cuenta con una aplicación multimedia capaz de visualizar las noticias y transmite dichas noticias a través de la televisión satelital. Tiene un Sistema de Teletexto para la plataforma de televisión satelital nacional que le facilita a la ACN la trasmisión de las informaciones de la prensa escrita nacional a los colaboradores cubanos en el exterior y a los habitantes de las zonas de silencio.

Arquitectura de Señal ACN.

El sistema se desarrolló usando la arquitectura de capas específicamente la variante de las 3 capas.

- **Capa de Presentación:** en esta capa se encuentra toda la interfaces de usuario y lógica de estas con la responsabilidad de la comunicación con los usuarios del sistema.
- **Capa de Aplicación (Lógica de Negocio):** se encarga de las tareas relacionadas con los procesos del negocio relacionado a la gestión de las noticias y recursos, seguridad y todo el procesamiento que se realiza detrás de la aplicación visible para el usuario.



CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

- **Capa de Datos:** es la responsable de las operaciones de entrada/salida con la base de datos del sistema.

Distribución por capas: Modelo de 3 Capas

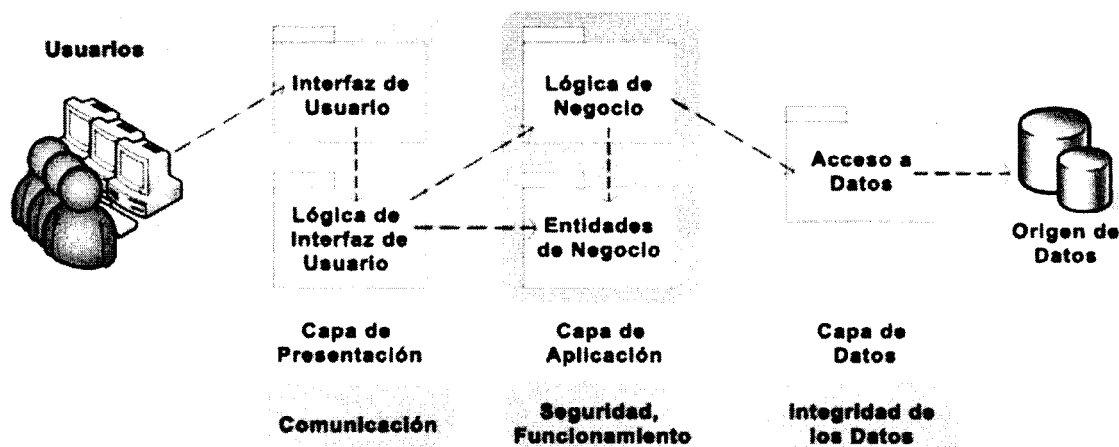


Fig. 1 Arquitectura de Señal ACN. (2)

Módulos o subsistemas de Señal ACN.

Módulo de Administración.

Permite la gestión de las noticias y recursos multimedia según los requerimientos de la Agencia Cubana de Noticias que se mostrarán posteriormente en el canal. Este módulo es la columna vertebral del sistema ya que propone un flujo de trabajo donde a través de él se definen los operadores del canal y su rol dentro del sistema, los cuales pueden realizar diferentes acciones según su nivel de acceso: redactar noticias con toda la información asociada (título, cantidad de pantallas, música de fondo etc.), corregir las noticias que han sido introducidas al sistema pudiendo modificar cada campo de la noticia, también se gestiona la programación del canal publicando las noticias para fechas y horarios específicos, también permite gestionar las secciones que se mostrarán en el canal, así como todo el banco de temas musicales que servirán de fondo en el canal. Este módulo es una aplicación Web desarrollada en ASP nativo y como lenguaje base VBScript. (2)

CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

Módulo de Actualización de Recursos Multimedia.

Módulo capaz de actualizar los recursos externos del sistema entre los servidores que lo soportan, se encarga de copiar de manera automática los archivos multimedia (imágenes, videos, audio) desde el servidor de Administración hasta el de Transmisión y actualizar en la BD en que fueron copiados; este módulo se ejecuta cada 3 minutos y fue enteramente desarrollado en la plataforma .Net para el Framework.Net v 1.1 y lenguaje C#. (2)

Módulo de Transmisión.

Posibilita la visualización automática de las informaciones del teletexto durante las 24 horas en un formato de diseño pre elaborado, además de mostrar otras informaciones como el tiempo que le queda a la noticia al aire, la próxima sección y a la hora que saldrá al aire, muestra también información del fondo musical que se escucha. Esta aplicación de Macromedia Director tiene como lenguaje base el Lingo. (2)

Seguridad en el gestor de contenidos de Señal ACN.

La información manejada por el sistema está protegida de acceso no autorizado, donde se definen usuarios específicos para la administración del sistema y tienen acceso a las opciones según el rol que desempeñe. Solo se accede al subsistema de Administración desde las computadoras localizadas en el Departamento Editorial. A los demás subsistemas se accede solamente desde los servidores donde se encuentren instalados.

La información manejada por el sistema es objeto de cuidadosa protección contra la corrupción y estados inconsistentes. Existen mecanismos de chequeo de integridad y se hacen copias de respaldo para restaurar el sistema en caso de fallo crítico o pérdida total de la información.

Aporte social que tiene Señal ACN en la actualidad.

Señal ACN influye significativamente en la sociedad ya que existen algunos lugares donde se hace muy difícil que las personas se mantengan informadas, dentro de éstas están las llamadas zonas de

CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

silencio, lugares en los que viven alrededor de 70 mil cubanos y donde hasta hace muy poco no llegaba ni tan siquiera la señal de televisión o radio.

En estos lugares la prensa escrita llega una vez por semana y es preciso lanzarla desde una avioneta, lo cual no cumple objetivo si se tiene en cuenta que la prensa escrita nacional se publica a diario y con esa misma frecuencia debe llegar a los lectores; hoy los pobladores de estas zonas tienen acceso de forma dinámica y actualizada a todo el acontecer de la isla gracias a la creación de Señal ACN.

Otro de los aportes que brinda Señal ACN es que favorecer a los colaboradores nacionales en el extranjero para que puedan estar al tanto de la prensa nacional, facilitándoles el acceso a las informaciones de la prensa plana con un ahorro significativo de recursos al país por conceptos de combustible para la transportación de la prensa diaria y recursos de imprenta, además de mantener el intercambio entre ellos y sus familiares.

Descripción de herramientas empleadas en el desarrollo de Señal ACN.

➤ Lenguajes de programación.

El sitio Web Lenguajes de Programación en Internet, dirigido hacia un amplio análisis sobre lenguajes de programación, define a éstos como lenguajes que pueden ser utilizados para controlar el comportamiento de una máquina, particularmente una computadora; compuesto por un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos respectivamente. **(3)**

Seguidamente se realiza una descripción de los lenguajes utilizados para desarrollar el sistema informático propuesto.

HTML.

HTML acrónimo de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, además de permitir aglutinar textos, sonidos e imágenes y combinarlos. Es usado por los navegadores para mostrar las páginas Web al usuario, siendo hoy en día la interfaz más extendida en la red. **(4)**

CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

Se puede decir que es una aplicación de la tecnología de marcado de textos, conocida por los usuarios de computadores. El concepto base, es el de intercalar en la información, marcas o etiquetas, que indiquen a la máquina que el texto circundante debe presentarse en negritas, con colores, o que está en un determinado lenguaje. Estas propiedades adicionales, denominadas marcas, ayudan a los ordenadores a trabajar en forma automática sobre el texto. (2)

El HTML ofrece importantes ventajas para el desarrollo de la aplicación Web que soporta la Administración del sistema de teletexo debido a su gran capacidad multimedia para presentar los contenidos. (2)

Java Script.

Desarrollado por la empresa Netscape Communications, es un lenguaje interpretado, orientado a objetos y con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Con él se pueden crear programas no muy complejos encargados de realizar acciones dentro del ámbito de una página Web, como introducir efectos especiales en las páginas o definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Java Script y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador (5). Es conocido como lenguaje del lado del cliente a causa de esta última característica.

Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el más utilizado de su tipo. Es un lenguaje que permite la programación de scripts y de programas complejos con funciones y estructuras de datos. Además de poner a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente sin necesidad de volver a cargar la página.

Por sus características y facilidades de empleo ofreció formidables utilidades para la programación de las operaciones del lado del cliente de la Administración del sistema de teletexo. (2)



CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

Active Server Page (ASP).

ASP es una tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft. El tipo de servidores que la emplean son aquellos que funcionan con SO de la familia de Windows NT (6). Aunque en sí no es un lenguaje de programación se incluye en esta sección por su importancia para la codificación de páginas activas.

Para escribir páginas ASP se utiliza un lenguaje de scripts, que se colocan en la misma página Web junto con el código HTML. Comúnmente este lenguaje de scripts es Visual Basic Script (VBScript), derivado del conocido Visual Basic, aunque también se pueden escribir los scripts en otro lenguaje como el Java Script (6).

En su ambiente se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para la Web.

Funciona del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante, esto ofrece gran seguridad para el programador sobre su código, porque éste se encuentra únicamente en los archivos del servidor que al ser solicitado, es ejecutado, por lo que los clientes no tienen acceso más que a la página resultante en su navegador.

Las ventajas que reporta el ASP al desarrollo del sistema y razones por las que ha sido utilizado son:

- Es liviano y puede ser ejecutado en PCs normales que tengan Windows y un servidor Web.
- Se puede utilizar desde cualquier computadora que esté conectada a la red que tenga instalado un navegador.
- Las funcionalidades del ASP son bastante amplias, entre las principales están el acceso a base de datos, envío de correo electrónico, creación dinámica de gráficos, entre otras.

Se añadió además la necesidad de la realización del sistema en un corto plazo y los conocimientos avanzados que el equipo de trabajo tuvo de la tecnología. (2)

CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

Lingo.

Lingo es el lenguaje de programación que lleva incorporado Macromedia Director; software de autoría multimedia muy utilizado en la actualidad. Lingo es la herramienta de Director más poderosa para crear una enriquecedora e interactiva experiencia para los usuarios. Un script puede realizar tanto la reproducción de un sonido cuando un usuario hace clic sobre un botón como controlar una serie de acciones (7).

Lingo permite integrar y controlar el comportamiento de textos, imágenes, sonidos y video digital con relativa facilidad, logrando que el desarrollo de la aplicación sea mucho más rápido y flexible. Con la utilización de este lenguaje se puede explotar todas las potencialidades del Director, posibilitando incluso conexiones a servidores de bases de datos. Debido a sus características e importancia en la producción de aplicaciones multimedia se decidió su utilización para la implementación del módulo de Transmisión de noticias. (2)

C Sharp (C#).

Es el lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Aunque es posible escribir código para la plataforma en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usándolo es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET (8).

Es actualmente uno de los lenguajes de programación más populares en informática y comunicaciones. En este lenguaje se toman todas las funcionalidades de lenguajes como Visual Basic y se le añaden las potencialidades de otros como C y C++.

C# brinda una gran integración a los productos de Microsoft a través de la plataforma .NET, lo que facilita la creación de aplicaciones de escritorio y servicios de Windows. Estas son razones por las cuales se decidió su utilización para el desarrollo de un servicio que permita la transferencia automática de los recursos multimedia entre los servidores del sistema. (2)

CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

Sistema de Gestión de Bases de Datos.

SQL Server

Los sistemas que manejan grandes volúmenes de información, como es el caso de los sistemas de teletexto, necesitan de un soporte que les permita almacenar, acceder y modificar estos datos. En este ámbito juega un papel importante los Sistemas Gestores de Bases de Datos (SGBD) utilizados para negociar estas operaciones.

Stanley Su en su libro *"Database Computers: Principles, Architectures and Techniques"* define un Sistema de Gestión de Bases de Datos como un paquete de software que proporciona todas las facilidades para la creación, recuperación, manipulación y mantenimiento de bases de datos, asegurando su integridad, confidencialidad y seguridad (9).

Entre los SGBD más usados se encuentra el SQL Server, el cual fue escogido por el cliente para la gestión de los datos del sistema. El SQL (Structured Query Language) es un lenguaje estándar de comunicación con bases de datos que presenta una notable potencia y versatilidad. Al mismo tiempo exhibe cuantiosas ventajas que ayudarán al trabajo eficiente con el gran volumen de datos que se deberá manejar en el sistema, entre ellas se encuentran:

- La capacidad de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.
- Gran estabilidad y seguridad.
- Soporta procedimientos almacenados, los cuales permiten ejecutar consultas complejas en breve tiempo.
- Posibilita la réplica de datos entre servidores SQL. (2)

Microsoft SQL Server 2000

Microsoft SQL Server 2000 es un sistema cliente/servidor de administración de base de datos relacional, diseñado para soportar un alto volumen de base de datos incluyendo procesamiento de transacciones en línea de alto rendimiento (online transaction processing - OLTP), almacenamiento de datos (data warehousing) y aplicaciones de comercio electrónico (ecommerce).



CAPITULO 1: SISTEMA AUTOMATIZADO DE TELETEXO PARA LA PLATAFORMA DE TELEVISION DIGITAL SATELITAL CUBANA

Alternativas existentes para la migración de sistemas con las características de Señal ACN a Software Libre.

La búsqueda de alternativas para hacer la migración de sistemas con características similares a las de Señal ACN ha sido una de las tareas a seguir por los autores, llegando a la conclusión de que la migración de sistemas a software libre es una problemática que se puede apreciar con frecuencia en grupos de desarrollo y soporte de software. Las ventajas que puede traer consigo desarrollar sistemas sobre software, plataformas y gestores de bases de datos puramente privativos se ven deterioradas con el paso del tiempo. Son varias las instituciones, los países (incluyendo el nuestro) y grupos de desarrollo que, después de un largo período de trabajo y acumulación de experiencias, se lanzan a la migración a software libre de sus sistemas.

En la actualidad la UCI al igual que Cuba da sus primeros pasos encaminados hacia la migración de sus sistemas de gestión interna a alternativas libres, pero la experiencia acumulada es muy escasa y no se cuenta con la documentación de los aislados trabajos que se ha realizado en este sentido.

Desafortunadamente durante la búsqueda no se encontró trabajos que aportaran a la realización de la migración.

Son varias las plataformas o lenguajes libres y los gestores de bases de datos con los que se le pudiese dar solución a la migración de dicho canal, razón por la cual se le ha dedicado un capítulo a este tema.

Conclusiones Parciales Capítulo I.

Hasta este punto se ha presentado una descripción del sistema que será sujeto de la migración, exponiendo un resumen de sus características esenciales, los módulos que lo componen, su arquitectura y la gestión de la seguridad del mismo. También se abordaron las condiciones bajo la cual surgió el proyecto y las tecnologías empleadas para su desarrollo, así como el gestor de base datos y el servidor Web utilizado para el desarrollo y explotación del mismo.

CAPÍTULO II

Capítulo II: Estudio para la migración.

Introducción.

En este capítulo se pretende hacer un estudio de las plataformas software libre con las que se pudiese hacer la migración del canal ACN, así como el servidor Web y el gestor de BD que se pudiera emplear para lograrla. Además de exponer sin ambigüedades las herramientas que se emplearán en la migración, los IDEs y los Sistemas Operativos a utilizar.

Posición de Cuba ante el uso y la migración a Software Libre.

El bloqueo económico sostenido por el gobierno de los Estados Unidos sobre Cuba por más de 45 años, impide el uso y pago de licencias de software producido por compañías estadounidense tan prestigiosas como IBM, Oracle, Adobe, Sun Microsystems, McAfee Computer Virus Software y Microsoft compañía líder mundial en la producción de software privativo y anti-ético como la suelen llamar prestigiosos autores de la rama. Durante todo estos años son más de una las artimañas, mentiras y los obstáculos impuesto por el gobierno de los Estados Unidos para impedir que el desarrollo tecnológico llegue a nuestro país.

Existen muchos medios de divulgación como la prensa y son muchos los artículos y estadísticas que tocan desde muy cerca este asunto, no es para nadie una mentira que cada Byte que llega a nuestro país es pagado por encima de su precio, ya que no se adquiere directamente de las manos de los productores sino por intermediarios que nos hacen pagar grandes sumas. Microsoft en las licencias de sus productos consigna que estos no pueden ser exportados a Cuba, ni a mediadores que comercialicen con Cuba o sencillamente a quienes ellos sospechen que lo hacen. **(10)**

Los asuntos legales expuestos anteriormente no son los únicos motivos para emerger de las turbulentas aguas del Software Privativo, por ello Cuba es uno más de los tantos países en dar los primeros pasos a la migración a Software Libre de sus sistemas de administración pública, educativos y de gestión hospitalaria.

El hacer uso de Software Privativo va más allá de problemas legales, su uso puede verse incluso como un problema de seguridad nacional. En el 2007, durante La Feria Informática de la Habana el Ministro

CAPITULO 2: ESTUDIO PARA LA MIGRACION

de la Informática y las Comunicaciones, Ramiro Valdés Menéndez, llamó a los sistemas informáticos del mundo como un "campo de batalla" donde Cuba está luchando contra el imperialismo.

En la actualidad el estado cubano ya forma parte del Movimiento de Software Libre Mundial, desarrollando sus propias estrategias y métodos, siguiendo las experiencias de países como Alemania, Argentina, Brasil, Chile, China, España, México, República Dominicana y Venezuela, entre otros.

Estos últimos años han sido de gran avance para el desarrollo de la comunidad de software libre en nuestro país, son muchos los esfuerzos y el interés por los organismos centrales del estado y sus dependencias en aras de apoyar este proceso de migración. Los más de 600 Joven Club de Computación también se convertirán paulatinamente en espacios para el desarrollo y consolidación de la comunidad de Software Libre cubana.

En el mes de Junio del año 2001, en Ciudad de la Habana, se firma la llamada Declaración de La Habana sobre el uso de Software Libre y Abierto. Se firmó esta declaración por la celebración del 1er Taller Latinoamericano para la Promoción del uso de Software Libre en la Educación, Ciencia, Cultura y Áreas Sociales; varios representantes de Cuba, Uruguay, Brasil, Ecuador y Colombia acordaron lo siguiente: **(11)**

- Recomendar a los gobiernos de la región el establecimiento de políticas que favorezcan el uso de SWL y abierto en los servicios públicos.
- Recomendar a la comunidad internacional, la celebración, cada año, del día mundial del software libre, preferiblemente el 5 de octubre por ser en esa fecha cuando se dio a conocer la creación del núcleo del SO GNU/Linux, como expresión de un trabajo colectivo acumulado durante varios años para lograr este objetivo.
- Incrementar la divulgación del uso del software libre en nuestra región, mediante estrategias nacionales, regionales y locales adecuadamente concebidas.
- Desarrollar programas de estudios que incrementen la enseñanza de software libre y abierto en las universidades y otros centros docentes de nuestros países.
- Incluir los aspectos de software libre en las políticas que se tracen para favorecer la construcción de un mundo sin exclusión social y con igualdad de acceso a los programas tecnológicos y a la información.

CAPITULO 2: ESTUDIO PARA LA MIGRACION

- Promover iniciativas para hacer llegar al usuario final técnicas, herramientas y metodologías que les permitan utilizar software de código abierto en la solución de sus problemas.
- Promover la presencia de libros y artículos sobre software libre en revistas, sitios Web y otras ediciones de carácter científico.

El uso de software propietario es más que un problema económico, legal y de seguridad, ya que además de hacernos dependientes de la potencia enemiga que más nos ataca, podemos decir que tampoco nos enriquece tecnológicamente, puesto que no nos permite estudiarlo, ni modificarlo, o mejorarlo, ni distribuirlo; por lo cual promueve el consumismo y el individualismo, y no permite desarrollar nuestra propia identidad y cultura. (12)

La posición de Cuba queda bien clara, promover una cultura de libertad e independencia tecnológica es una tarea de todos y que mejor forma de dar nuestro aporte a este proceso que no sea el de revolucionar nuestro entorno y hacerlo a la vez que promovemos el uso del Software Libre.

Razones y necesidades para hacer la migración de Señal ACN.

Cuba se ha trazado una política bien definida para adoptar el uso del software libre por lo que su posición ante el uso del software privativo está bien clara y es esta misma política la principal razón para hacer la migración. Pero surgen varias interrogantes:

- ¿Es necesario hacer la migración de señal ACN?
- ¿Es factible esta migración?

Para dar respuesta a la primera interrogante se deben tener en cuenta los siguientes elementos que involucran a señal ACN:

- SO.
- Servidor Web.
- Gestor de base de datos.

En el Capítulo I se hace alusión a las características de Señal ACN; su plataforma de desarrollo se ejecuta sobre Windows 2003 Server SP2.

Si se habla en términos legales y de licencias sobre el uso de software privativo cabe señalar que este SO costaría, en caso de que se pagase por su licencia, más de 700 dólares por procesador. **(13)**

Sería bueno analizar también el precio que indirectamente se paga por concepto de seguridad y para profundizar más, valdría la pena hacerse la siguiente pregunta:

¿Qué tan seguro es Windows y su servidor Web?

Esta pregunta es muy frecuente en las comunidades de desarrollo y usuarios avanzados de este SO; son varios los artículos y autores que tratan este tema. Expertos en la materia aseguran que este SO no es tan seguro como lo hacen creer sus proveedores, es imposible confiar en su seguridad cuando no se puede ver ni modificar su código fuente para adaptarlo a necesidades propias de cada usuario.

Los datos recogidos por Netcraft¹ muestran que servidores Windows son reiniciados con mucha más frecuencia que máquinas que ejecutan servidores libres, esto es debido entre otras razones a los problemas de seguridad que con frecuencia presenta Windows.

Microsoft pone a disposición de sus usuarios actualizaciones de seguridad con gran frecuencia, estas actualizaciones son parches a fallos del SO que en muchas ocasiones son informados por los propios clientes y es común que estos parches tarden en aparecer hasta 3 meses después de reportados. Aproximadamente la mitad de estos fallos son considerados como críticos y pueden poner en riesgo millones de dólares en dependencia del uso que se le esté dando al sistema.

Si se considera un problema ejecutar a Señal ACN sobre un servidor Windows es también un problema que su gestor de BD sea parte de la gama de productos privativos de Microsoft, SQL Server 2000, con un precio de más de 250 dólares por procesador es el gestor que da soporte actualmente al flujo de información que gestiona el canal. **(13)**

En Señal ACN los productos de Microsoft se ejecutan sobre Windows, esto trae consigo que los riesgos de seguridad a los que está sometido el SO también afecten al gestor de BD y pongan en riesgo toda la información gestionada por el sistema.

¹ Compañía dedicada al análisis y encuestas de servidores Web conectados a Internet.

No se puede negar que Windows 2003 Server y SQL Server 2000 son estables y en los casi dos años que se han empleado para la gestión de noticias no han reportado problemas, ni ocasionado pérdidas de información alguna.

Después de conocer la posición de Cuba ante el uso y migración a software libre se decidió la no utilización de SQL Server 2000 ejecutándose sobre un servidor Windows 2003 Server pues existen prestigiosos gestores y SO libres que brindarían las mismas prestaciones para realizar la migración.

La factibilidad de hacer la migración del sistema se tendrá en cuenta durante el desarrollo de este trabajo y después de un estudio de las posibles Plataformas, Servidores Web y Gestores de Base de Datos se podrá llegar a una conclusión más certera de cuán factible es hacer la migración, que herramientas utilizar, que vía sería la más rápida y por supuesto la menos costosa, pero desde este momento se pueden dejar bien claro algunos beneficios tangibles e intangibles que traerían consigo la migración.

El beneficio tangible más importante asociado a la migración, es la supresión de las caras licencias de software que, aunque hoy por hoy no se pagan, sí hay que tenerlas presentes si se quiere exportar el sistema a otro país.

Son varios los beneficios intangibles asociados a la migración por lo cual es preferible enumerarlos:

1. Alinear el desarrollo tecnológico de un sistema clave, que soporta procesos bases de un centro de alta importancia estratégica, con las políticas tecnológicas de nuestro país.
2. Alcanzar un conocimiento teórico y práctico en materia de esfuerzo, de selección de herramientas y de alternativas tecnológicas que permitan allanar el camino a futuras iniciativas de migración de sistemas similares hacia software libre.
3. Lograr que el canal Señal ACN disfrute de todos los beneficios inherentes al software libre.

Actualmente Señal ACN tiene casi dos años de estar transmitiendo y nunca se había hablado de una posible migración. Es bueno dejar bien claro el por qué ahora es un buen momento para asumir esta tarea.

- El sistema se encuentra en un estado de estabilidad.
- La UCI cuenta con estudiantes preparados para formar un equipo de desarrollo que pueda asumir la nueva versión sobre una plataforma libre.



- Se prevé que la experiencia de Señal ACN se multiplique en otros centros que exigen el cumplimiento de las políticas sobre el empleo de software libre en los sistemas de gestión de noticias, por lo que se necesita probar y desarrollar la versión libre antes que se complete su despliegue.

La migración de Cuba hacia el software libre se encuentra en una etapa investigativa y de determinación de vías y metodologías que hagan del proceso de cambio algo factible y para nada engorroso por lo tanto se necesitan proyectos que sean pioneros en este sentido como lo es éste.

Módulos de ACN. Propuesta para la migración.

2.1.1 Módulo de Actualización de Recursos Multimedia.

En el desarrollo del Capítulo I se realizó una profunda descripción del módulo de Actualización de Recursos Multimedia, donde se expusieron sus características fundamentales y su funcionamiento dentro del sistema.

2.1.1.1 Plataformas Software Libre. Propuesta para la migración.

Para el desarrollo de aplicaciones de escritorio existen varias plataformas en software libre. Entre estas sobresalen dos en particular Java y Mono de las que a continuación se hará una breve descripción.

2.1.1.1.1 Java

La plataforma Java es capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a bytecode² y un conjunto de herramientas de desarrollo.

Esta plataforma se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real.

Las aplicaciones Java pueden usarse de forma variada y es preciso la utilización de un conjunto de herramientas conocidas como JDK (Java Development Kit, o herramientas de desarrollo para Java).

² Bytecode es un código intermedio más abstracto que el código máquina.



2.1.1.1.2 Mono

Mono es un proyecto de código abierto promovido por Novell³, con el objetivo de crear un grupo de herramientas libres, basadas en GNU/Linux, y compatibles con la tecnología .NET⁴, además es multiplataforma con una productividad sin precedentes. Mono provee el software necesario para desarrollar y ejecutar aplicaciones cliente y servidor en Linux, FreeBSD, Solaris, Mac OS X, Windows y Unix, funcionando en varias arquitecturas: x86, PowerPC, S390, SPARC, HPPA, StrongARM. Es compatible con múltiples lenguajes entre los que están C#, Java, Boo, Temerle, Visual Basic.NET y Python (14). Entre los entornos de desarrollo integrado (IDE) que utiliza Mono está Monodevelop en su versión 0.13.1.

2.1.1.1.3 Selección de la Plataforma Libre para hacer la migración.

Llegado el momento de escoger la Plataforma Libre que dará soporte al Módulo de Actualización de Recursos Multimedia se percibe que son varias las características de Mono y Java que son útiles para la migración de éste, así como lo son sus ventajas, desventajas y prestaciones.

A la hora de escoger o determinar cuál podría ser la plataforma más conveniente para migrar a software libre se debe partir del análisis de aquellas que permitan aprovechar al máximo todas las funcionalidades ya desarrolladas, necesitando para ello reprogramar o modificar lo menos posible lo que ya está implementado; lo que provocaría dedicar menos tiempo y recursos para este propósito.

De las plataformas presentadas Java se encuentra muy difundida, existe mucha documentación, son muchos los frameworks y plugins⁵ desarrollados, existe un IDE muy potente y altamente desarrollado que se ajusta prácticamente a todas las necesidades y son muchos los grupos de debate y foros de discusión que le dan soporte. No obstante presenta una serie de inconvenientes como son el hecho de que el proceso de desarrollo es un poco más lento y complicado, no permite la interoperabilidad de múltiples lenguajes.

³ Novell, Inc. es una compañía de origen estadounidense dedicada al software, específicamente en el área de sistemas operativos de redes, como Novell NetWare y Linux, entre otras ramas de la tecnología.

⁴ .NET es el proyecto de Microsoft de una nueva plataforma de desarrollo de software con énfasis en la transparencia de redes, que tenga independencia de la plataforma y que permita un rápido desarrollo de aplicaciones.

⁵ Programa que interactúa con una aplicación central o cliente (un navegador Web, un cliente de correo o un IDE de desarrollo, por ejemplo) para proveer ciertas funcionalidades muy específicas.

CAPITULO 2: ESTUDIO PARA LA MIGRACION

Es posible compilar a bytecode desde múltiples lenguajes, pero no es posible alcanzar el nivel de interacción que se da en .NET que permite desarrollar en diferentes lenguajes y que el código generado es completamente utilizable desde otras aplicaciones en otros lenguajes.

La similitud entre los lenguajes de programación C# y Java es bastante alta pero aún así por cuestiones de arquitectura y formas de implementar y desarrollar no se puede aprovechar en gran medida la versión de Señal ACN en .NET, principalmente se perdería mucho de lo que ya ha sido implementado lo cual sería costoso en tiempo.

Como segunda y última propuesta se encuentra Mono que una de las desventajas que presenta este proyecto es que se encuentra en desarrollo, el IDE de desarrollo se encuentra bastante atrasado y carece de muchas facilidades y ventajas, no existe mucha documentación sobre el mismo y no son muchos los grupos de debate y los foros de discusión que lo soportan, sin embargo es cien por ciento compatible con el lenguaje C# y con muchos otros lenguajes; incorpora el uso de una gran parte de los controles visuales existentes en la plataforma .NET, como son los tan fundamentales e imprescindibles controles de usuario y los datagrids⁶ entre otros; permite el uso total de las funcionalidades brindadas por el .NET framework 1.1, llegando al punto de poder ejecutar ensamblados producidos por el Visual Studio sin necesidad de recompilar, incluye las herramientas de desarrollo y la infraestructura necesarias para crear aplicaciones del cliente y el servidor. A pesar de la juventud del proyecto, existen en la actualidad muchas empresas y grupos de desarrollo que han implementado sistemas y aplicaciones utilizando parcial o totalmente Mono y algunos frameworks desarrollados para el mismo.

⁶ El control de servidor Web Datagrid es una cuadrícula de múltiples columnas, enlazada a datos.

	Sistema o Aplicación	Descripción	Empresa
Aplicaciones comerciales	Plasma	Sistema para la administración de código fuente	Codice Software
	Chrome Compiler	Compilador comercial de Pascal	RemObjects
	Unity	Editor de juegos de 3D	Otee
	GMOVIL	Herramienta para administrar procesos comunes para los puntos de venta de Vodaphone	Advanced Software Production Line
Aplicaciones Software Libre	Last-Exit	Reproductor basado en Gnome.	
	Galaxium	Cliente de mensajería instantánea	
	Gib	Herramienta para generar temas de iconos para Gnome	
	MonoUML	Herramienta CASE ²⁹	
	LAT	Herramienta administrativa de LDAP	

Tabla 1 Sistemas y aplicaciones desarrolladas utilizando Mono (15).

Existen componentes y librerías desarrolladas haciendo un uso parcial o total de Mono.

	Nombre	Descripción
Librerías	GtkSharp	Librería que permite desarrollar aplicaciones graficas nativas para Gnome utilizando Mono.
	Windows.Forms	Parte de la distribución estándar de Mono. Es un trabajo en progreso para implementar las API ³⁰ System.Windows.Form de Microsoft.
	wxNet	Librería para el desarrollo de aplicaciones para wxWindows
Componentes	fyiReporting	Incluye un diseñador de reportes con soporte para las especificaciones RDL. Este software incluye componentes reusables y controles para Windows.Forms
	ReportMan	Motor para la creación de reportes usando xml para su almacenamiento
	FlexCelNet	Es un paquete para la generación de reportes en Visual Studio .NET 2003 y 2005 en archivos Excel y PDF.

Tabla 2 Componentes y librerías desarrolladas utilizando Mono (16).

Después de analizar con profundidad las ventajas y desventajas propias de cada plataforma propuesta, podemos concluir que la plataforma Mono es la más indicada para hacer la migración de este módulo pues no pondría en peligro los servicios que brinda actualmente el Canal, permitiría llevar a cabo la migración en el menor tiempo posible y con el mínimo consumo de recursos y brindaría la posibilidad de reutilizar al máximo todas las clases y componentes.

Estado actual

En la actualidad la plataforma Mono está en un proceso acelerado de desarrollo y mejoras, impulsado por Novell, una empresa muy importante, con mucho prestigio y con grandes perspectivas en el mundo del software que estableció acuerdos de intercambio tecnológico y de cooperación con Microsoft, con muchos de sus clientes más conservadores, que nunca estuvieron dispuestos a cambiar a Linux han comenzado a considerarlo con más seriedad. Mono se encuentra en su versión 1.2.3 y ya se trabaja en el desarrollo de la versión 2.0 que ofrecerá soporte total para ASP.NET 2.0 y ADO.NET 2.0 (17).

Este proyecto ha desencadenado un gran interés en el desarrollo de componentes basados en C#, librerías y frameworks para Mono. Muchos de ellos han sido desarrollados por el mismo equipo de Mono, entre ellos sobresalen Tao framework para el trabajo con extensiones OpenGL, Mono.Directory.LDAP31 y Novell.Directory.LDAP para el acceso a LDAP, Mono.Data para el trabajo con base datos como PostgreSQL, MySQL, Firebird, SQLite, Microsoft SQL Server, Oracle y fuentes de datos ODBC, Mono.Http para el trabajo con manejadores http y Mono.XML para el trabajo con XML.

El Proyecto Mono ha desarrollado un compilado que permitirá a los desarrolladores de software quienes usan Microsoft Visual Basic utilizar sus aplicaciones a través de múltiples plataformas sin ninguna modificación al código. El nuevo compilado Mono Visual Basic permite a los desarrolladores continuar al código en su ambiente preferido de Visual Basic / Visual Studio y agrupar y ejecutar el mismo código basado en una variedad de sistemas operativos y arquitecturas.

Reusabilidad

Al seleccionar la plataforma se tuvo en cuenta varios aspectos fundamentales uno de estos es el grado de reusabilidad que ésta permite.

Para el desarrollo de ACN se aprovecharon las ventajas propias del Framework .NET en su versión 1.1. Las funcionalidades que se utilizan de este framework ya están desarrolladas en Mono, agrupadas en los frameworks Novell.Directory.LDAP, Mono.Data y Mono.XML, entre otros; haciendo que sea casi nulo el código a reprogramar para este fin. Otro de los aspectos que nos brinda Mono es que permite emplear el mismo lenguaje utilizado para programar todas las clases del sistema, es decir, minimiza los esfuerzos en implementación.

2.1.1.2 IDEs Libres. Propuesta para la migración.

IDE (Integrated Development Environment; traducido al español: Entorno Integrado de Desarrollo) puede hacer el trabajo mucho más sencillo, sobretodo si el sistema a desarrollar maneja un buen

número de clases. Además estos entornos permiten mucha más versatilidad para depurar programas. **(18)**.

Después de mencionar algunas características de IDEs, a continuación se presenta una breve descripción de Eclipse y C# Develop ya que existen varios entornos de desarrollos con los que se pueden trabajar, pero que no todos cumplen con las condiciones necesarias para realizar la presente migración.

2.1.1.2.1 C# Develop

C# Develop (CSharpDevelop) es un entorno de desarrollo libre para los lenguajes de programación C#, Visual Basic .NET y Boo. Este IDE es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft. Entre sus características está la escritura de código en C#, ASP.NET, ADO.NET, XML y HTML, con coloreado de sintaxis para los lenguajes C#, HTML, ASP, ASP.NET, VBScript, Visual Basic .NET, y XML. Presenta un conversor bidireccional entre C# y Visual Basic .NET, y unidireccional hacia Boo. Para el completado automático de código, la aplicación incorpora sus propios parsers⁷. SharpDevelop está escrito enteramente en C# y su última versión soltada es la 2.2 **(19)**.

2.1.1.2.2 Eclipse

El IDE de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente le permite a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, también puede trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación.

El Software Development Kit (SDK) o kit de desarrollo de software de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código.

⁷ Programas, procesadores o analizadores.

El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadata en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente. (20)

2.1.1.2.3 Selección del IDE Libre para hacer la migración.

De acuerdo a las características que tiene el sistema no es factible utilizar Eclipse como el entorno de desarrollo hacia el cual migrar, este IDE está desarrollado por completo en el lenguaje Java, donde se presenta el problema habitual de las herramientas Java, y es que se convierten en herramientas demasiado “pesadas”, es decir, necesitan una máquina muy potente para poder ejecutarse de forma satisfactoria con las cuales no cuenta las computadoras destinadas a estas funciones, en gran medida, éstas necesidades vienen determinadas por el uso del API Swing para su interfaz gráfico. Swing es una librería de widgets⁸ portable a cualquier plataforma que disponga de una máquina virtual Java pero a costa de no aprovechar las capacidades nativas del sistema donde se ejecuta, lo cual supone una ejecución sensiblemente más lenta que la de las aplicaciones nativas.

Este es el obstáculo que indica el estudio de otras alternativas, destacándose por sus características y prestaciones el C# Develop.

C# Develop es un IDE libre para C# cuenta con una característica muy fuerte y es que está escrito enteramente en C# favoreciendo la migración de este módulo. Además brinda soporte para el desarrollo de interfaces, clases, namespaces y proyectos en C#. Presenta una gran ventaja y es que permite importar los proyectos creados en Microsoft Visual Studio .Net.

2.1.1.3 Lenguaje Libre. Propuesta para la migración.

2.1.1.3.1 C#

C# nace de Microsoft con la idea de crear un lenguaje con todos los aspectos. Concebido como lenguaje nativo de su plataforma .Net para aplicaciones Web y de Escritorio, se dice que C# es un lenguaje que conjunta un grupo de aspectos de los lenguaje C++, Java y Visual Basic, pero de una forma más versátil y mejorada agregándole cada vez más elementos que faciliten su uso. Presenta un

⁸ *widget* es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*.

estándar libre para la plataforma Mono, la implementación de la plataforma .NET de Mono se basa en los estándares de ECMA C# y la infraestructura del lenguaje común. El compilador de C# (MCS) de mono en estos momentos se considera que cumple la especificación, por lo que es un producto relativamente maduro. MCS está escrito en C# y es capaz de compilarse a si mismo desde enero de 2002, así como a otros muchos programas.

2.1.1.3.2 Python

Python es un lenguaje de programación multiparadigma, que permite varios estilos: programación orientada a objetos, programación estructurada y programación funcional. Es un lenguaje interpretado que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar; el intérprete que presenta se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. También puede utilizarse como un lenguaje de extensión para módulos y aplicaciones que necesitan de una interfaz programable. Python permite mantener de forma sencilla interacción con el SO, y resulta muy adecuado para manipular archivos de texto.

2.1.1.3.3 Selección del lenguaje libre para hacer la migración.

Después de haber mencionado algunas características que presentan los lenguajes C# y Python, se hace bastante difícil migrar el módulo de actualización a este último lenguaje ya que el principal propósito es optimizar el tiempo y una de las desventajas que presenta es que tiene mayor lentitud mientras se ejecuta el programa, pues cada vez que se ejecuta debe ser interpretado por el intérprete, lo que hace más conveniente el uso del estándar libre de C#, ya que el módulo en su versión está desarrollado en C# aunque en la plataforma .Net, lo que hace que la migración sea menos costosa para el equipo de desarrollo en cuanto al tiempo a dedicar a ésta tarea, pues existe una compatibilidad del 100% entre los proyectos desarrollados en .NET y el C# en su estándar libre.

2.1.2 Módulo de Administración.

Durante el desarrollo del Capítulo I se describió el módulo de Administración, donde se plantearon sus características fundamentales y el rol dentro del sistema.

2.1.2.1 IDEs Libres. Propuesta para la migración.

2.1.2.1.1 Eclipse

Eclipse es una herramienta para el programador que facilita al máximo la gestión de proyectos colaborativos, posee una estructura de plug-in que hace sencillo añadir nuevas características y funcionalidades, consta de asistentes (wizards): para la creación, exportación e importación de proyectos, es capaz de generar esqueletos de códigos (templates), su control de versiones es con cvs o con subversión (con subclipse), posee resaltado de sintaxis, autocompletado y tabulador de bloque de código seleccionado junto a muchas otras utilidades de edición que ayudan enormemente al programador.

2.1.2.1.2 MonoDevelop

El **MonoDevelop**, es un IDE de desarrollo libre para GNOME que hace uso de Mono, diseñado inicialmente para el desarrollo con C# y otros lenguajes de .NET y encontrándose en estos momentos en la versión 1.0 (21).

MonoDevelop presenta una arquitectura que se divide en tres capas, la capa principal (Core layer) que provee los servicios para las aplicaciones, la capa de Proyectos (Project layer) que implementa el modelo de objetos del proyecto y la capa Superior que es en general el propio IDE.



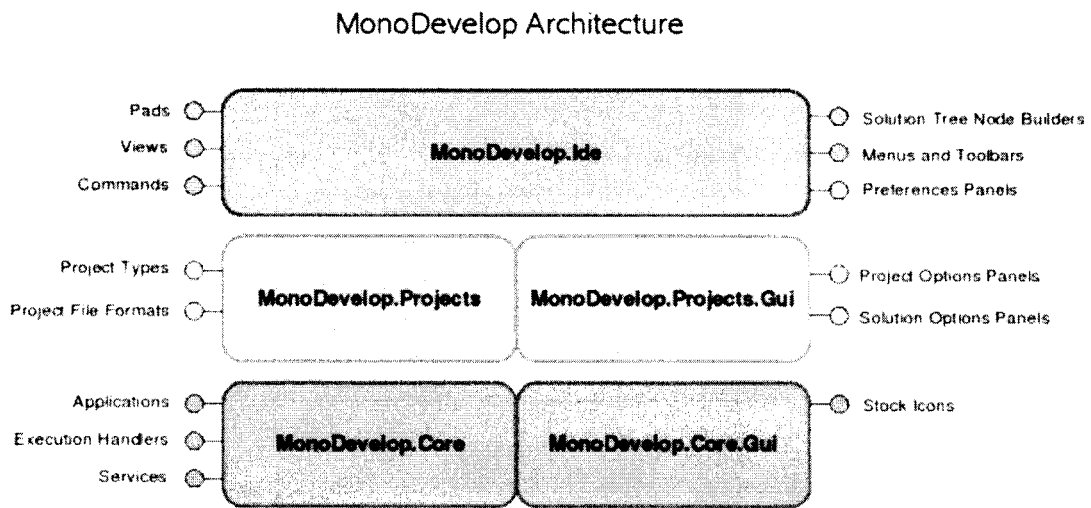


Fig. 2 Arquitectura de MonoDevelop (22).

2.1.2.1.3 Selección del IDE Libre para hacer la migración.

Una de las desventajas que presenta MonoDevelop es que se encuentra bastante atrasado y carece de muchas ventajas, por lo que no existe mucha documentación sobre este IDE y los foros y grupos de debate que lo soportan no son muchos.

Por otra parte se propone realizar la migración con Eclipse como IDE pues se considera que tiene mejor relación calidad-facilidad y su instalación es muy sencilla. En cuanto a las aplicaciones clientes, Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones Web, etc., por ejemplo, GEF (Graphic Editing Framework - Framework para la edición gráfica) es un plugin de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto WYSIWYG⁹ hasta editores de diagramas UML, interfaces gráficas para el usuario (GUI), etc. Dado que los editores realizados con GEF "viven" dentro de Eclipse, además de poder ser usados conjuntamente con otros plugins, hacen uso de su

⁹ Acrónimo de *What You See Is What You Get* (lo que usted ve es lo que usted consigue) le permite a un usuario visualizar un documento según aparecerá en el producto final, y editar directamente el texto, gráficos u otros elementos dentro de esa vista.

interfaz gráfica personalizable y profesional, también se puede decir que mediante plugins se le puede añadir soporte de lenguajes adicionales.

2.1.2.2 Lenguajes Libres. Propuesta para la migración.

En la actualidad son muchos los lenguajes Web libres, los más difundidos son el Python, Perl, Java y PHP, de los cuales estos dos últimos presentan las mejores y más útiles prestaciones que ayudarán a construir el módulo a migrar.

2.1.2.2.1 PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel especialmente diseñado para el desarrollo Web y que puede ser embebido en páginas HTML, ejecutándose en el servidor. Cuenta con algunos entornos de desarrollo Web libres entre los que se destaca Quanta que fue diseñado especialmente para el proyecto KDE¹⁰. Presenta muchas ventajas entre las que se encuentra el Modo de Diseño de Página Visual (VPL) "WYSIWYG", la de visualizar el código fuente o ambas vistas a la vez.

2.1.2.2.2 Java

El lenguaje Java combina la sintaxis para programación genérica, estructurada y orientada a objetos, fue construido desde el principio para ser completamente orientado a objetos, permitiendo la ejecución de un mismo programa en múltiples sistemas operativos, es decir tiene independencia de la plataforma, incluye por defecto soporte para trabajo en red, está diseñado para ejecutar código en sistemas remotos de forma segura, es fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

2.1.2.2.3 Selección del lenguaje libre para hacer la migración.

De los lenguajes de desarrollo en software libre presentados Java no es la opción más indicada atendiendo a las características del módulo objeto de la migración pues para cualquier programa Java necesita dos componentes en el sistema donde se va a ejecutar: una máquina virtual de Java (JVM), y

¹⁰ K Desktop Environment es un entorno de escritorio gráfico e infraestructura de desarrollo para sistemas Unix y, en particular, Linux.

un conjunto de librerías para proporcionar los servicios que pueda necesitar la aplicación además de grandes cantidades de memoria RAM; en cambio PHP es de los más rápidos, más livianos y de los que menos tiempo de desarrollo consume, es sencillo de utilizar, se encuentra ampliamente difundido, existe gran cantidad de documentación, grupos y foros de debate que facilitan todo el proceso de desarrollo.

El módulo a migrar es una aplicación Web compleja, con mucha lógica de negocio por lo que no es recomendable utilizar solo PHP sino que remite al desarrollador a utilizar un framework para simplificar de esta forma el desarrollo de la aplicación mediante la automatización de algunos de los patrones utilizados para resolver tareas comunes, asegurando el mantenimiento y las ampliaciones futuras de la aplicación y de esta forma que el código sea ligero, legible y efectivo.

Dos de los posibles framework a utilizar por sus amplias ventajas son CakePHP y Symfony.

CakePHP es un framework de desarrollo rápido para PHP que provee una estructura extensible para el desarrollo, mantenimiento y utilización de aplicaciones, usando patrones bien conocidos como MVC (Modelo- Vista-Controlador) y ORM (Object-Relational mapping, traducido al español sería mapeo objeto-relacional) que junto a convenciones sobre paradigmas de configuración disminuye los costos de desarrollo y ayuda a los programadores, permitiendo escribir menos código.

No obstante la documentación que hay referente a este framework es muy escasa o casi nula por lo tanto la decisión más acertada es Symfony porque es un framework maduro, bien documentado y con una gran comunidad que lo apoya, fue creado con PHP 5 y está enfocado al desarrollo de aplicaciones Web en el mismo lenguaje de programación, añade una nueva capa por encima de PHP y proporciona las herramientas que simplifican el desarrollo de las aplicaciones Web complejas, separando la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de este módulo.

Symfony automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de la aplicación. Además es compatible con la mayoría de los gestores de bases de datos, como MySQL, Oracle y PostgreSQL, se puede ejecutar en Unix, Linux, etc., además de ser fácil de instalar y configurar en la mayoría de los sistemas operativos antes mencionados, es sencillo y flexible, está basado en la premisa de *"convenir en vez de configurar"* por lo que el desarrollador solo tendría que configurar aquello que no le es convencional; también sigue la mayoría de mejores prácticas y patrones de diseño para la Web y permite un mantenimiento muy sencillo.

Symfony dispone de las herramientas necesarias para crear una completa administración de forma automática. En pocos segundos y sin necesidad de escribir ni una sola línea de código, la aplicación puede disponer de una parte de administración con soporte de las cuatro operaciones básicas (crear, modificar, borrar y obtener registros de la base de datos).

2.1.3 Módulo de Transmisión. Propuesta para realizar la migración

En el Capítulo I se expusieron las características fundamentales del módulo de Transmisión, donde se explicó su función en el sistema.

El mundo del software libre actualmente ha dado pasos agigantados en cuanto a el desarrollo de aplicaciones, muy útiles, potentes y muy competentes incluso con grandes desarrolladores como la Microsoft, por solo citar alguno; sin embargo durante la búsqueda de software con el cual trabajar imágenes, videos y sonidos, es decir herramientas para desarrollar contenidos multimedia, los resultados no fueron tan satisfactorios, pues en esta área el desarrollo es aún muy incipiente, la opción que mas resaltó fue Revolution, pero también se encuentra muy incipiente, con poca documentación y no son tantas sus prestaciones, presenta problemas como los que aparecen a la hora de la reproducción de videos sobre Linux donde utiliza el reproductor XAnim, pero esta es una aplicación muy vieja y no soporta los formatos de compresión modernos como el MPEG o los DIV, y el MPlayer (otro reproductor) presenta incompatibilidad de librerías con Debian (SO que se propone más adelante para instalar en las PC servidores de aplicaciones), además la solución para la reproducción de *.swf que se tendría era a partir del reproductor libre gnash, del cual aún no han aparecido versiones estables. Todo esto además de que posee algunas librerías las cuales no son libres del todo.

Por todo lo anteriormente expuesto los autores han decidido llevar a cabo la migración con AJAX que permite simular a la perfección el comportamiento de aplicaciones de escritorio en la Web por lo que posee todas las prestaciones necesarias para desarrollar el módulo de Transmisión con todas sus opciones y de paso con alta calidad y una buena experiencia para el usuario final.

AJAX(acrónimo de Asynchronous JavaScript And XML) trae consigo muchas ventajas como las inherentes con la recuperación asíncrona de datos, reduciendo el tiempo de espera del usuario, facilidad de manejo del usuario, se reduce el tamaño de la información intercambiada, mayor

portabilidad entre plataformas, no requiere instalación de complementos *applets*¹¹ de Java, ni ningún otro elemento y una de las más importantes es que el código es público, está basado en los estándares abiertos, formado por las siguientes tecnologías:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe¹² en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON (JavaScript Object Notation)¹³ y hasta EBML (Extensible Binary Meta Language)¹⁴.

Y está soportado por los navegadores más utilizados de Internet como son: Mozilla, Internet Explorer, Safari y Opera. Permite a las páginas hacer una pequeña petición de datos al servidor y recibirla sin necesidad de cargarse nuevamente. El incremento de este tipo de actualizaciones elimina el tener que refrescar la página completa, algo bastante apreciado a la hora de operar en una aplicación Web.

Esta tecnología es válida en cualquier plataforma y navegador: Internet Explorer y Mozilla Firefox son los que se llevan las palmas en el mercado de Internet y además son los navegadores en los que es más fácil programar aplicaciones Web AJAX, aunque también es posible construir aplicaciones Web basadas en AJAX para que funcionen en los navegadores más modernos como Amaya, Opera y versiones recientes de Internet Explorer. Es independiente del tipo de tecnología de servidor que se utilice: así como AJAX funciona en cualquier navegador, es perfectamente compatible con cualquier

¹¹ Un *applet* es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador Web.

¹² Un *iframe* es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

¹³ JSON es un formato ligero para el intercambio de datos.

¹⁴ EBML fue diseñado como una extensión binaria simplificada de XML, con el propósito de almacenar y manipular datos de forma jerárquica con campos de longitud variable.



tipo de servidor estándar y lenguaje de programación Web como: PHP, ASP, ASP.Net, Perl, JSP, Cold Fusion.

El ser completamente compatible el desarrollo en éstas tecnologías ha ayudado a AJAX a que vaya cada vez más en ascenso. Mejora la estética de la Web: con AJAX se puede fusionar la imaginación del desarrollador con la usabilidad de una aplicación Web de modo que se puede realizar una aplicación que de no estar dentro de un navegador, podría pasar por una aplicación normal de escritorio.

AJAX se ejecuta en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla, lo que significa aumentar la interactividad, velocidad y usabilidad en la misma ya que el intercambio de información con el servidor se produce en un segundo plano.

Prototype es un framework desarrollado en JavaScript por Sam Stephenson para el desarrollo sencillo y dinámico de páginas Web. Prototype simplifica gran parte del trabajo cuando se pretende desarrollar páginas altamente interactivas, muy útil y es el framework que se propone para su utilización en la migración de este módulo, pues incluye decenas de utilidades, dispone de funciones sencillas para proporcionar ayuda a la hora de escribir los scripts, la librería es compacta y está programada de forma muy eficiente, dispone de una completa documentación de todas las funciones y métodos que componen su Application Programming Interface (API)¹⁵, además de ser muy compatible con el framework Symfony que fue el que se escogió para llevar a cabo la migración del módulo de la Administración del canal.

Ajaxian (un sitio dedicado a AJAX) ha publicado el resultado de una encuesta que ha realizado entre sus lectores sobre sus preferencias a la hora de usar un determinado entorno y aunque el grado de popularidad de un framework no decide su utilización ya que en ocasiones no es proporcional la popularidad con la utilidad, se quiso presentar los resultados ya que en este caso coincidió con el que se utilizará en la migración de este módulo. El siguiente gráfico muestra los resultados:

¹⁵ API es el conjunto de métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.



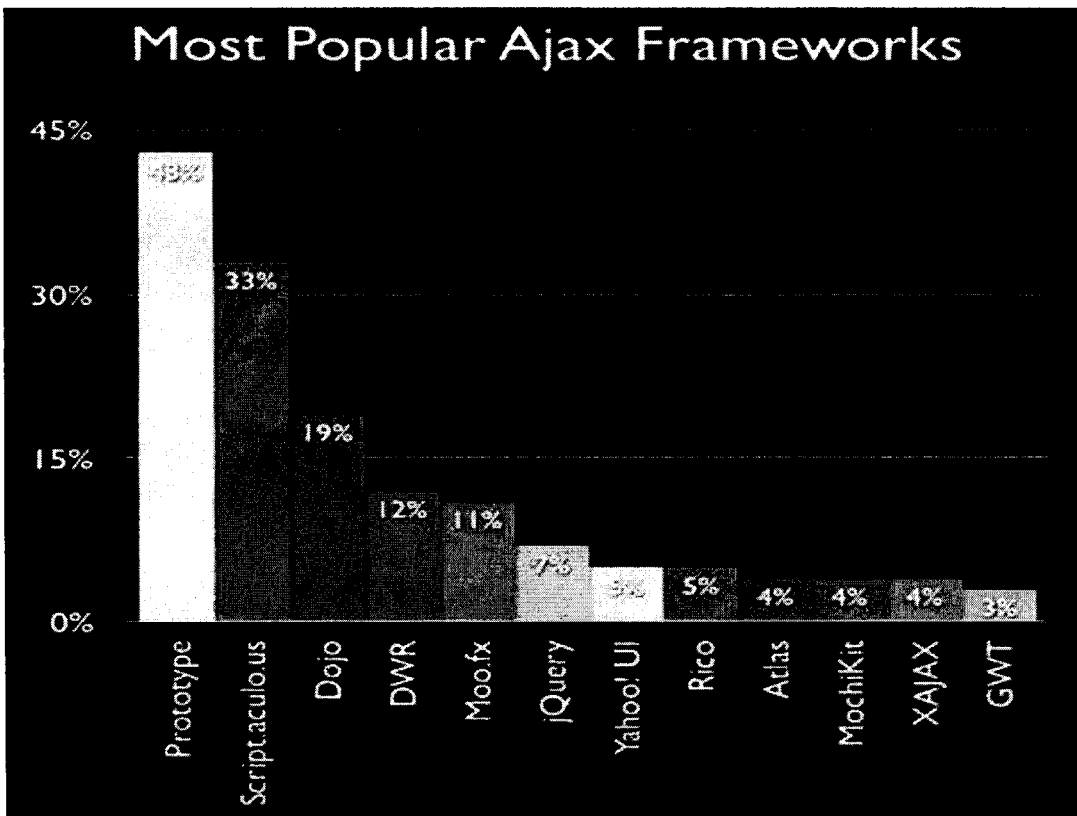


Fig. 3 Los frameworks más populares de AJAX. (23)

Servidores Web Libres. Propuesta para la migración.

2.1.4 Apache

Ya seleccionada las plataformas atendiendo a cada uno de los módulos hacia las cuales se realizará la migración, se debe seleccionar el Servidor Web Software Libre (SWSL) que se encargará de dar respuesta a las peticiones http hechas desde los clientes del sistema. Son más de uno los SWSL con los que se cuenta en la actualidad tales como Apache, Lighttpd, Cherokee, NanoWeb, pero independientemente de la cantidad de servidores, las prestaciones y la disponibilidad de los mismos es difícil dudar ante la utilización del Apache.

El Software Libre más popular y difundido en el mundo es el Apache, fue considerado por mucho tiempo el proyecto punta de lanza del movimiento de Software Libre en el mundo, por su ayuda en el impulso del mismo. Es una suerte que se pueda contar con un SW tan ligero, altamente configurable y a su vez libre como lo es este. Apache es catalogado como el SW número uno, aproximadamente el 65 % de los SWSL utilizados en el mundo son Servidores Apaches, de acuerdo con una encuesta dada a conocer por la compañía Netcraft, ésta compañía muestra periódicamente estadísticas de los SW más utilizados. En la siguiente figura se muestran los resultados de dicha encuesta:

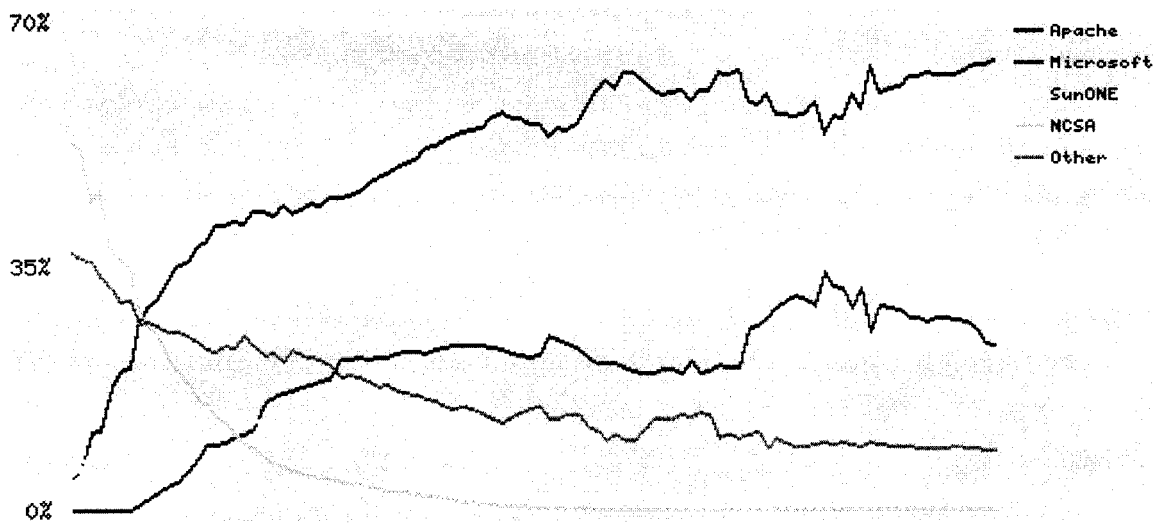


Fig. 4 Servidores Web más utilizados. (24)

Servidor Web

La anterior gráfica solo muestra que es el SWSL más utilizado, ahora se analizarán las ventajas que lo ponen en esta posición y que lo hacen el elegido ineludible para la migración.

Ventajas al utilizar el servidor Web Apache

➤ **Apache es un servidor altamente configurable de diseño modular**

Las prestaciones de Apache son muy sencillas de ampliar en dependencia de las necesidades de quien lo utilice. Sus creadores aseguran que con un nivel aceptable en los lenguajes de programación C o Perl es posible desarrollar un módulo para que realce una función deseada. La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web. Algunos de estos módulos son:

- **mod_ssl** - Comunicaciones Seguras vía TLS.
- **mod_rewrite** - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como PHP en páginas estáticas HTML para así engañar a los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).
- **mod_dav** - Soporte del protocolo WebDaV (RFC 2518)¹⁶.
- **mod_deflate** - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- **mod_auth_ldap** - Permite autenticar usuarios contra un servidor LDAP.
- **mod_proxy_ajp** - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets¹⁷ y JSP¹⁸).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- **mod_perl** - Páginas dinámicas en Perl.
- **mod_php** - Páginas dinámicas en PHP.
- **mod_python** - Páginas dinámicas en Python.

¹⁶ WebDaV es un protocolo que proporciona funcionalidades para crear, cambiar y mover documentos en un servidor remoto (típicamente un servidor Web)

¹⁷ Los servlets son objetos que se ejecutan dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad.

¹⁸ JSP es una tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo.

- **mod_rexx** - Páginas dinámicas en REXX y Object REXX.
- **mod_ruby** - Páginas dinámicas en Ruby¹⁹.
- **mod_mono** - Páginas dinámicas en Mono
- **mod_security** - Filtrado a nivel de aplicación, para seguridad.

- **Apache es una tecnología gratuita de código fuente abierto.**

Apache va mas allá de ser un software gratuito, su código fuente es abierto, este código fuente se puede usar, modificar y distribuir de forma gratuita.

- **Apache funciona en Linux, en otros sistemas de Unix y Windows.**

A pesar de que muchos autores catalogan a Apache como uno de los mayores logros del mundo del Software Libre, este no sólo es ejecutado en sistemas operativos de esta línea. Apache aunque se considera que es más estable y eficiente cuando se ejecuta sobre sistema Linux, también se puede ejecutar sobre Solaris y Mac OS. Muchos sistemas que se ejecutan sobre Windows utilizan como servidor Web a Apache 2.0 en vez de Microsoft Internet Information Server (IIS).

2.1.4.1 Características más importantes del Servidor Web Apache 2

Apache 2 presenta muchas otras características, entre ellas un elaborado índice de directorios, un directorio de alias, negociación de contenidos, informe de errores HTTP configurable, ejecución SetUID²⁰ de programas CGI²¹, gestión de recursos para procesos hijos, integración de imágenes del lado del servidor, reescritura de las URL, comprobación de la ortografía de las URL y manuales online. Otras características esenciales de Apache 2 se muestran a continuación y fueron tomadas de la Biblia de Apache 2.

¹⁹ Ruby es un lenguaje de programación reflexivo y orientado a objetos (lenguaje interpretado).

²⁰ Setuid y setgid son términos de Unix, abreviaturas para "Set User ID" y "Set Group ID", respectivamente. Setuid, también llamado a veces "suid", y setgid son permisos de acceso que pueden asignarse a archivos o directorios en un SO basado en Unix.

²¹ Common Gateway Interface es de las primeras formas de programación Web dinámica. Es utilizado comúnmente para contadores, base de datos, motores de búsqueda, formularios, generadores de e-mail automático y mapas de imágenes, juegos en línea y otros; permite a un cliente solicitar datos de un programa ejecutado en un servidor Web.

➤ Soporte del protocolo HTTP 1.1.

Apache es uno de los primeros servidores Web en integrar el protocolo HTTP 1.1. Es totalmente compatible con el estándar HTTP 1.1 y al mismo tiempo sigue siendo compatible con HTTP 1.0. Está preparado para todas las novedades del nuevo protocolo, por ejemplo, antes de HTTP 1.1, un navegador Web tenía que esperar una respuesta del servidor Web antes de poder emitir otra petición. Con el surgimiento de HTTP 1.1, esto ha dejado de ser así. Un navegador Web puede enviar solicitudes en paralelo, las cuales ahorran ancho de banda dejando de transmitir las cabeceras HTTP en cada solicitud. De algún modo esto ofrece un estímulo del lado del usuario final porque los archivos solicitados en paralelo aparecerán antes en el navegador.

➤ Sencillo, con la configuración basada en un poderoso archivo.

El servidor Apache no posee una interfaz de usuario gráfica para su administración. Se trata de un sencillo archivo de configuración llamado *httpd.config* para algunas distribuciones, en otras como Debian y Ubuntu se llama *apache2.config* y que se puede utilizar para configurar Apache. Únicamente se necesita un editor de texto para su configuración, sin embargo, es lo suficientemente flexible para permitirle repartir la configuración de su host virtual en múltiples archivos para no sobrecargar un único archivo *httpd.config* con toda la gestión de las múltiples configuraciones de servidores virtuales.

➤ Soporte para CGI (Common Gateway Interface).

Apache soporta CGI utilizando los módulos *mod_cgi* y *mod_cgid*. Es compatible con CGI y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web.

➤ Soporte de FastCGI.

Actualmente la mayoría no escribe sus CGI en Perl, cabe preguntarse ¿Cómo pueden hacer sus aplicaciones CGI más rápidas?

Apache también tiene una solución para esto. Utilizando el módulo *mod_fcgi* para implementar un entorno FastCGI dentro de Apache y logrando hacer que las aplicaciones FastCGI arranquen rápidamente.

➤ Soporte de host virtuales.

Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.

➤ Soporte de autenticación http.

Apache soporta autenticación básica basada en la Web. Está también preparado para autenticación por resumen, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autenticación básica utilizando tanto archivos estándar de contraseña como los DBM (gestores de Bases de datos), llamadas a SQL o llamadas a programas externos de autenticación.

➤ Soporte de scripts PHP.

Este lenguaje de script ha comenzado a ser muy utilizado y es con el cual se propone la migración del subsistema de administración del sistema; Apache ofrece un amplio soporte de PHP utilizando el módulo *mod_php*, el cual es de mucha utilidad en este subsistema.

➤ Soporte de servlets de Java.

Los servlets de Java y las JSP se están convirtiendo en algo muy común en los sitios Web dinámicos. Puede ejecutar servlets de Java utilizando el premiado entorno Tomcat con Apache.

➤ Servidor proxy integrado.

Puede convertir Apache en un servidor proxy caché. Sin embargo, la implementación actual del módulo opcional de proxy no soporta HTTP proxy o el último protocolo HTTP 1.1.

➤ Estado del servidor y adaptación de registros.

Apache le da una gran flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador Web. Además, puede adaptar sus archivos de registro a su gusto.

➤ Soporte de Server Side Includes (SSI).

Apache ofrece un conjunto de Server Side Includes que añaden una gran flexibilidad para el desarrollador del sitio Web.

➤ Soporte de Secured Socket Layer (SSL).

Puede crear fácilmente un sitio Web SSL utilizando OpenSSL y el módulo *mod_ssl* de Apache.

2.1.5 El Servidor Web que se propone para la migración.

Se tuvo en cuenta que para la migración del sistema, el servidor Web es un componente muy importante. Se necesita un servidor Web cuyas características y prestaciones no sólo se asemejen a las del Microsoft Internet Information Server IIS, si no que de ser posible tenga un rendimiento mayor, sea escalable y capaz de manejar la concurrencia de múltiples usuarios. Las características y la madurez de Apache 2 asociado a sus prestaciones y su escalabilidad, lo hacen el servidor Web candidato y propuesto para hacer la migración del sistema.

El hecho de poseer el conjunto de características expuestas con anterioridad, que sea software libre y que sea el servidor Web más difundido en el mundo, lo hacen ser el servidor que se propone para la migración del sistema.

Servidores de Base de Datos Libres. Propuesta para la migración.

2.1.6 Caracterización de los sistemas de gestores de Bases de Datos libres (BDL) más difundidos.

2.1.6.1 PostgreSQL.

Es un Sistema de Gestión de Base de Datos Objeto-Relacional (ORDBMS) resultado de varios años de trabajo y evolución del proyecto Ingres en la Universidad de Berkeley. PostgreSQL se distribuye bajo la licencia BSD²² lo cual permite el uso y distribución sin costo, tanto para aplicaciones comerciales como no comerciales; incluye la mayor parte de los tipos de datos especificados en los estándares SQL92²³ y SQL99²⁴, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp; ha sido propulsor y creador de muchos conceptos que existen en el modelo objeto-relacional que se incluyen en sistemas gestores de base de datos puramente comerciales y más difundidos en este ámbito.

Actualmente se encuentra liberada la versión 8.3 de este producto. Características como la orientación a objeto, los tipos de datos, el empleo de funciones, restricciones, triggers²⁵, reglas e integridad transaccional, hacen de PostgreSQL un gestor de base de datos objeto-relacional.

2.1.6.1.1 Características de PostgreSQL

PostgreSQL cumple completamente el ACID, esta es la propiedad que debe cumplir una base de datos para realizar transacciones seguras. ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability, llevándolo al español sería: Durabilidad, Aislamiento, Consistencia e Indivisibilidad. PostgreSQL no sólo se ejecuta en entornos libres, sino que se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc.

²² La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre.

²³ Son estándares del lenguaje SQL.

²⁴ Son estándares del lenguaje SQL.

²⁵ Un trigger o un disparador en una Base de datos es un evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

CAPITULO 2: ESTUDIO PARA LA MIGRACION

En contraste a muchos sistemas de base de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Por su arquitectura de diseño, escala muy bien, al aumentar el número de CPUs y la cantidad de RAM. Soporta transacciones y, desde la versión 7.0, llaves foráneas (con comprobaciones de integridad referencial), además también presenta soporte para triggers y procedimientos almacenados. Soporta los tipos de datos bases y también datos de tipos: fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc.

Además permite la creación de tipos de datos propios. El tamaño máximo de la base de datos es ilimitado, el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB, el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla. El número de índices por tabla es también ilimitado.

PostgreSQL cuenta con una gran comunidad de desarrollo distribuida en todo el mundo. Esto hace posible contar con una amplia documentación pública, organizada, libre y en diferentes idiomas. El código fuente está disponible para todos sin costo alguno. Si se necesita extender o personalizar PostgreSQL de alguna manera, esto se puede hacer con un mínimo esfuerzo sin costos adicionales en dependencia de los conocimientos y experiencia del desarrollador.

Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo. PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC²⁶ esto le proporciona mejor respuesta en ambientes de grandes volúmenes de datos; los principales proveedores de sistemas de base de datos comerciales usan también esta tecnología, por las mismas razones.

PostgreSQL disfruta del soporte de numerosas compañías, las cuales auspician desarrolladores y proveen recursos de hosting. Además existe una numerosa comunidad de compañías que ofrecen soporte de PostgreSQL, desde individuos hasta compañías multinacionales.

Otras características interesantes de PostgreSQL son las siguientes:

- Alta concurrencia, que evita tener que bloquear una tabla cuando se está escribiendo en ella.
- Copias de seguridad en línea.

²⁶ Multi-Version Concurrency Control es una técnica avanzada para mejorar las prestaciones de una base de datos en un entorno multiusuario.

- Replicación asíncrona.
- Transacciones anidadas.
- Optimizador de consultas.

2.1.6.1.2 PostgreSQL 8.2

Las nuevas versiones por lo general son más estables que las versiones anteriores y se basan en la corrección de errores presentes en las precedentes, pero para lograr la estabilidad se necesita de un tiempo después de ser liberada la versión, PostgreSQL, ya cuenta con su versión 8.3 liberada el 4 de febrero de 2008 pero es muy poco el tiempo de prueba que ha tenido por lo que en el presente trabajo se realizará una caracterización de la versión 8.2.

En esta liberación PostgreSQL además de mantener sus funcionalidades básicas incluidas en versiones anteriores, incorpora nuevas características avanzadas, madurez y rendimiento requeridos por los usuarios corporativos, entregando simplicidad de manejo comparable a los sistemas líderes de manejo de bases de datos que lo hacen un gestor más profesional, robusto y estable.

Agrega sintaxis extendida y compatible, e interfaces que han sido solicitadas por su comunidad, haciendo más fácil para nuevos Administradores de BD (DBAs) el uso de todas las características avanzadas de PostgreSQL.

Las nuevas herramientas y características que facilitan la gestión de bases de datos y el desarrollo incluyen:

Mejoras de Rendimiento: la versión 8.2 mejora el rendimiento en alrededor de un 20% en pruebas de sistemas OLTP²⁷ de gran escala. Los usuarios de sistemas *data* warehouse pueden notar mejoras aún mayores. Los cambios incluyen ordenamientos en memoria y en disco más rápido, mejor escalabilidad en sistemas multi-procesador, mejor optimización de consultas sobre datos particionados, cargas masivas más rápidas y outer joins considerablemente acelerados.

²⁷ OnLine Transaction Processing es un tipo de procesamiento de transacciones a través de una red de computadoras.



Bases de Datos Warm Standby: una extensión de «recuperación a un punto en el tiempo», introducida en la versión 8.0, permite a los administradores crear fácilmente una copia para recuperación inmediata (failover) de su clúster de bases de datos.

Construcción en línea de índices: la construcción de índices puede ocurrir mientras las aplicaciones escriben a las tablas de la base de datos, permitiendo el afinamiento de rendimiento sin afectar la disponibilidad.

Características de SQL 2003: PostgreSQL, bien conocido por su conformidad a los estándares, ha agregado sintaxis para varias características introducidas en la especificación ANSI SQL: 2003, incluyendo funciones de agregación estadística, sentencias VALUE con múltiples registros, UPDATE RETURNING y funciones de agregación de múltiples columnas.

Las características avanzadas que se ofrecen con PostgreSQL 8.2 antes que ningún otro sistema de gestor de bases de datos incluyen:

- **Índices Invertidos Generalizados:** una forma más escalable y programable de indexar datos semi-estructurados y texto.
- **DTrace:** PostgreSQL ha sido instrumentado para permitir la trazabilidad a través de un marco genérico de monitoreo, usando DTrace en Solaris y otras herramientas avanzadas de traza.

“Con instancias de data warehouse con tablas de miles de millones de registros y varios terabytes de datos, las capacidades de construcción en línea de índices y hot-standby son cruciales”, dice Theo Schlossnagle, Director en OmniTI Computer Consulting.

Además de las características listadas anteriormente, hay más de 200 nuevas prestaciones y mejoras contribuidas por docenas de desarrolladores para la versión 8.2. Entre ellas se cuentan:

- **Bloqueos consultivos (“advisory locks”):** permiten el control de objetos de bases de datos a nivel de aplicación usando el eficiente motor de bloqueos de PostgreSQL.
- **pg_dump selectivo:** permite extraer vuelcos transaccionalmente consistentes de relaciones, seleccionando la inclusión y exclusión usando expresiones regulares.

CAPITULO 2: ESTUDIO PARA LA MIGRACION

- **Sentencias preparadas:** tiene nuevas interfaces administrativas y mejoras de rendimiento en sentencias preparadas.
- **ISN/ISBN:** este módulo contiene tipos de datos para soportar descriptores de producto ISN para inventarios y sistemas de venta al por menor. Ha sido aumentado para soportar los últimos estándares internacionales.
- **Criptografía:** el módulo pgcrypto, soportando criptografía dentro de la base de datos, fue actualizado con los últimos algoritmos.
- **Mejoras al SQL:** nueva sintaxis, incluyendo UPDATE RETURNING, DROP IF EXISTS, ON COMMIT y nuevos comandos de propiedad ("ownership") y permisos, para hacer más fácil el manejo de objetos de la base de datos en la línea de órdenes.
- **Extracción de registros por lotes en psql:** permite devolver filas a la consola en lotes en lugar de todas a la vez.
- **Reconstrucción de ECPG:** la interfaz embebida C de PostgreSQL ha sido extensivamente reescrita y se agregaron nuevas interfaces.
- **Mejoras importantes en TSearch2:** soporte de UTF-8, tesauros, soporte de reescritura de consultas e indexación GIN.
- **PL/Python:** ahora soporta parámetros con nombre y funciones que devuelven conjuntos.

La arquitectura extensible de PostgreSQL ha conducido a una gran comunidad de extensiones como lenguajes procedurales adicionales, interfaces a sistemas externos, nuevos tipos de índices, replicación y herramientas de alta disponibilidad.

Entre los proyectos comunitarios que han tenido avances significativos en el ciclo del lanzamiento de 8.2 se cuentan:

- **Full Disjunctions:** gracias a Google Summer of Code, PostgreSQL es el primer sistema de base de datos que tiene una implementación de este nuevo y radical algoritmo de join.
- **PGPool II:** PGPool, bajo el auspicio del gobierno Japonés, ha agregado nuevas capacidades de balanceo de carga y particionamiento.



PL/Java: uno de los dos lenguajes procedurales Java ha alcanzado estado de producción completo, permitiendo la utilización completa de Java dentro de la base de datos.

2.1.6.2 Firebird

Firebird es un sistema de administración de base de datos relacional (o RDBMS) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++. El proyecto se desarrolla activamente y el 18 de abril de 2008 fue liberada la versión 2.1. Este SGBD ofrece muchas características de SQL ANSI estándar; funciona en Linux, Windows, MacOSX y en una gran variedad de plataformas UNIX.

Firebird ofrece una concurrencia excelente, alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores. Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows. Su arquitectura es Cliente/Servidor sobre protocolo TCP/IP.

Soporta de transacciones ACID y claves foráneas, es medianamente escalable. Su seguridad está basada en usuarios/roles, sus requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad. Posee pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos. Tiene capacidad de almacenar elementos BLOB (Binary Large Objects). Además de tener soporte de User-Defined Functions (UDFs).

(25)

2.1.6.2.1 El gestor de base de datos que se propone para la migración.

Son varias las características de PostgreSQL 8.2 y Firebird, así como lo son sus ventajas, desventajas y prestaciones, lo cual hace un poco difícil la tarea de seleccionar el servidor de base de datos que se propondrá para la migración del sistema. Es difícil hacer una comparativa entre ambos gestores, hay que basarse en las prestaciones que ambos tienen en común, esto resulta algo verdaderamente engorroso puesto que ambos gestores poseen distintas implementaciones.

Para la migración del sistema se necesita un gestor de base de datos cuyas características y prestaciones se ajusten en gran medida a las siguientes condiciones.

- Gran estabilidad y seguridad.

- Soporte altos niveles de concurrencias.
- Soporte integridad referencial.
- Altos niveles de confiabilidad y escalabilidad.
- Soporte procedimientos almacenados o funciones y vistas.
- Sea estable durante el manejo de grandes volúmenes de información.

Estas seis características son las más críticas y son las que debe presentar el gestor que se emplee para la migración. De los dos gestores analizados con anterioridad el gestor, cuyas características resultaron más semejantes a las necesidades que debe cumplir el gestor que se propondrá para la migración, es PostgreSQL en su versión 8.2.

El hecho de que sea un Sistema de Gestión de Base de Datos Objeto-Relacional, que soporte la integridad referencial y que en ambientes en que se gestionen grandes volúmenes de datos su capacidad operacional se comporte más estable que Firebird, hace que, junto al resto de características anteriormente expuestas en este epígrafe, sea más apropiado para la migración. Por lo tanto, se propone el uso del gestor PostgreSQL 8.2 y su ejecución sobre un SO software libre para hacer un uso más óptimo de sus capacidades en el manejo de grandes volúmenes de información.

La alternativa de utilizar Firebird no resultó del todo tentativa porque posee muy poca documentación y muy pocos clientes gráficos.

Sistema Operativo. Propuesta para llevar a cabo la migración.

El canal Señal ACN está ubicado de manera tal que los servidores Web/BD se encuentran en una máquina, y el servidor de Transmisión de noticias está alojado en otra; por tal razón los autores de este trabajo estiman conveniente, sacándole partida a las prestaciones que proporciona cada una de las distribuciones de sistemas operativos en la actualidad, instalar en dependencia del uso de las PC, el SO que más se adecue, por lo queda organizado para los servidores Web/BD el Debian y para el de Transmisión el Ubuntu.



CAPITULO 2: ESTUDIO PARA LA MIGRACION

Ubuntu es una distribución de Linux de tipo escritorio basada en Debian que está centrada en el usuario final y brinda facilidad de uso. También se puede decir que ahorra mucho tiempo en las instalaciones y permite tener las últimas versiones de las aplicaciones de escritorio en mejor tiempo.

Es muy popular y con mucho soporte en la comunidad. El entorno de escritorio por defecto es GNOME. Posee una gran colección de aplicaciones útiles y sencillas para la configuración de todo el sistema, a través de una interfaz gráfica útil para usuarios que se inician en Linux, el navegador Web oficial es Mozilla Firefox.

El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse, por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema, mejora la accesibilidad y la internacionalización, de modo que el software está disponible para tanta gente como sea posible. Todos los lanzamientos de Ubuntu se proporcionan sin costo.

Debian GNU/Linux que es un SO libre que utiliza el núcleo Linux pero la mayor parte de las herramientas básicas vienen del Proyecto GNU, de ahí el nombre GNU/Linux, contiene programas pre compilados distribuidos en un formato que hace más fácil la instalación en su computadora. Permite SSH para crear conexiones seguras con otras máquinas con SSH instalado.

Este SO es uno de los más recomendados para soportar servidores ya que se compone de 11 arquitecturas diferentes, su principal objetivo es distinguirse por su excelencia técnica. Además cuenta con más de 9 mil paquetes compilados estables facilitándonos múltiples ventajas, presenta una buena interfaz gráfica, además cuenta a su vez con una gran cantidad de soporte y documentación disponible de forma libre y gratuita.

Hasta la fecha existen nueve versiones estables, la última versión es la 4.0, con su última actualización publicada el 17 de febrero de 2008 que es la que utilizaremos para realizar la migración.

Conclusiones Parciales Capítulo II.

En este capítulo se ha propuesto como resultado del profundo estudio realizado durante su elaboración los IDEs, el servidor Web, lenguajes de programación, el gestor de base de datos y el SO a utilizar para llevar a cabo el desarrollo de la migración a software libre del sistema de Señal ACN, teniendo como premisa fundamental que la misma se realice de forma óptima y sin contratiempos.

Capítulo III

Capítulo III Desarrollo de la migración.

3.1 Introducción

En este capítulo se realizará un Plan de Migración el cual tendrá un conjunto de acciones agrupadas en etapas por las que debe transitar el sistema antes de migrar completamente, se analizarán los costos y beneficios de dicha migración para ver la viabilidad del proceso y además se desarrollará un prototipo funcional de uno de los módulos, en este caso, el módulo de Transmisión para demostrar la factibilidad de la solución propuesta.

3.2 Análisis de Costos y Beneficios

El desarrollo de una migración de software está determinado entre otros aspectos por la relación entre los costos y beneficios, es necesario al menos obtener un equilibrio entre estos dos parámetros, de forma tal, que sea factible el seguimiento de la misma, no sería lógico dar curso a una migración cuyos beneficios sean insignificantes en comparación con los costos de la misma.

Generalmente los factores que influyen en el coste de un sistema son los costes iniciales de licencia, los costes de formación y los costes de tiempo de estudio y resolución de problemas.

El modelo de licenciamiento en el Software Libre es radicalmente distinto al de otras soluciones, en este tipo de software no se licencia por cliente/PC o CPU, sino que una única licencia le permite instalar este software en un número ilimitado de máquinas.

Es más sencillo y menos costoso asegurar un sistema libre que uno propietario ya que las herramientas están públicamente disponibles y la documentación para ello también. En los sistemas propietarios, la tarea, aparte de ser muy costosa en cuanto a tiempo requiere de herramientas de terceros que llevan asociado un alto coste, además, la aplicación de los parches del fabricante no está exenta de problemas de disponibilidad, no sólo es que se requiera un reinicio, es que puede introducir nuevos problemas, por contra, en el Software Libre la política de parches de seguridad es mucho más fiable ya que son probados por un gran número de desarrolladores antes de ser liberados, además, la modularidad del diseño de los sistemas operativos libre y la disponibilidad del código fuente hace que

los tiempos de respuesta ante la aparición de un fallo sean muchísimo menores que los de otras soluciones.

Otra vertiente que se une en el costo final de la migración está dada por la determinación del coste total de propiedad (Total Cost of Ownership) abreviado como "TCO" en inglés, deben identificarse todos los costes importantes, el llamado "modelo de costes", y estimar su importe, además de no olvidar los costes "ocultos", como los costes de administración, de actualización, de asistencia técnica, de operación de usuario final, etc. De este modo, el código abierto tiene una serie de ventajas de costes importantes en varias categorías que pueden dar lugar, en muchos casos, a un TCO menor como se muestra a continuación:

El código abierto cuesta mucho menos en su adquisición inicial. En muchos casos, los programas recogidos en las distribuciones abiertas se pueden descargar libremente, linux.org proporciona algunos enlaces a donde encontrar las distribuciones. En caso de que las conexiones de internet no tengan alta velocidad, es preferible pagar una pequeña cantidad a un distribuidor por un paquete agradablemente integrado con CD-ROMs, documentación en papel, y asistencia y aún así el código abierto es menos costoso de adquirir.

Tabla de precios del software propietario (26) tanto privativo (27) como en SWL:



	precios		precios
WINDOWS SERVER 2003 STANDARD R2 SP2+5CLIENTES INGLES OEM	567.21 € 657.96 € con IVA(Impuesto sobre el Valor Añadido)	Debian	\$30.00 por 3 DVD.
Windows XP SP2	379.00 €	Ubuntu	\$ 10.00 por cada DVD. (28)
SQL Server 2000	\$2.100		

Los costes de actualización son típicamente menores: Los costes de actualización a largo plazo son mucho menores para los sistemas abiertos, por ejemplo, actualizar un sistema Microsoft típicamente cuesta la mitad de la compra original, esto es así, porque esencialmente se está en sus manos para los precios a largo plazo, una vez que es el único proveedor; en contraste, los sistemas GNU/Linux se pueden descargar gratis o simplemente comprar de segunda mano generalmente por menos de \$100.00, y una única actualización se puede utilizar en todos los sistemas y si no le gusta su suministrador de GNU/Linux, lo puede cambiar.

El código abierto permite frecuentemente aprovechar hardware más antiguo, con más eficacia que los sistemas cerrados, reduciendo aun más los costes de hardware, y en algunos casos eliminando la necesidad del nuevo hardware.

El código abierto se ejecuta más rápido en hardware más rápido, por supuesto, pero muchos programas abiertos pueden utilizar el hardware antiguo más eficientemente que los programas cerrados, resultando en un coste del hardware menor, e incluso en algunos casos no requiriendo nuevos costes, porque los sistemas "descartados" pueden a menudo utilizarse de nuevo, por ejemplo, los requisitos mínimos para Microsoft Windows 2000 Server de acuerdo a las especificaciones de Microsoft son una CPU compatible con Pentium 133 MHz o superior, 128 MB de RAM mínimo con

256MB como "mínimo recomendado", y 2 GB de disco duro, con al menos 1 GB libre. Según Red Hat, Red Hat Linux 7.1, una distribución común de GNU/Linux, exige al menos un i486 (se recomienda un Pentium), 32MB RAM con 64MB recomendado, y 650MB de espacio en disco, 1,2 GB recomendados.

Conforme se incrementa la cantidad de sistemas y el rendimiento del hardware, esta diferencia en los costes inicial y de actualización se vuelve incluso más sustancial. Proporcional al incremento del número de servidores, las soluciones cerradas se van volviendo cada vez más costosas. En primer lugar, muchos sistemas cerrados, incluyendo a Microsoft, venden licencias por cliente, esto hace que si su hardware puede servir a más clientes, tendrá que pagar más para utilizar ese hardware que ha comprado.

En segundo lugar, puede que usted desee utilizar más ordenadores, en ese caso deberá pagar más licencias de sistemas cerrados, en cambio, para la mayoría de las distribuciones de GNU/Linux, se pueden instalar tantas copias como se desee sin pagos adicionales, y no hay límite de rendimiento incluido en el programa, puede encontrarse un pago adicional por la asistencia técnica añadida, pero aún para esta asistencia se pueden encontrar proveedores compitiendo entre ellos.

De acuerdo con Network World Fusion News, Linux se está utilizando cada vez más en entornos sanitarios, financieros, bancarios y de distribución comercial, a causa de las ventajas de coste cuando se incrementa el número de instalaciones idénticas. De acuerdo a sus cálculos, para un despliegue de 2.000 ordenadores, SCO UnixWare costaría 9 millones de dólares, Windows costaría 8 millones de dólares, y Red Hat Linux 180 dólares.

Hay muchos otros factores en el TCO, pero es difícil categorizar sus efectos en general y es generalmente difícil encontrar números que justifiquen esos otros efectos. En GNU/Linux y UNIX se necesita un número menor de administradores de sistemas, porque la administración es más fácil de automatizar, y los sistemas son más sencillos de poner en marcha.

El GNU/Linux permite instalar múltiples servicios en una única máquina, mientras que las instalaciones de Windows necesitan varias máquinas. La administración del cumplimiento de las licencias puede ser costosa en entornos cerrados, es decir, el tiempo dedicado a comprar las CAL, seguir la pista a los documentos de licencia, y preparar la auditoría siempre implícita, un coste que en código abierto es simplemente irrelevante.

Muchas organizaciones han informado de ahorros importantes utilizando código abierto. Aquí tenemos algunos ejemplos de organizaciones específicas que han ahorrado utilizando código abierto:

El documento *Linux as a Replacement for Windows 2000* es un ejemplo de un análisis comparando RedHat Linux 7.1 y Windows 2000, en el caso de este cliente, utilizando Linux en lugar de Windows 2000 ahorró 10.000\$. El revisor venía de un entorno Windows/DOS, y tras realizar un estudio intensivo sobre el proyecto Linux de varios meses, determinó que "le sorprenderá todo lo que le ofrecen los programas de código abierto".

El Vicepresidente de IT de Intel, Doug Busch, informó de un ahorro de 200 millones de dólares reemplazando costosos servidores UNIX por servidores corriendo GNU/Linux mucho más baratos.

Amazon.com fue capaz de recortar 17 millones de dólares en gastos de tecnología en sólo un trimestre, principalmente porque cambió a Linux. Amazon gastó en tecnología 54 millones de dólares el tercer trimestre (terminado el 30 de septiembre), comparado con los 71 millones de dólares el mismo trimestre del año anterior, y los ejecutivos esperaban que el coste de tecnología como porción de las ventas netas disminuyera en un 20%. **(29)**

El TCO como total se debe considerar, y no sólo ciertas categorías, aún más, dadas estas grandes diferencias, en muchas situaciones el TCO del código abierto es mucho menor que el TCO de sistemas cerrados.

Hubo un tiempo en el que se proclamaba que la instalación de código abierto llevaba más tiempo, pero hoy día, los sistemas abiertos se pueden comprar pre instalado, y los instaladores automáticos suponen un esfuerzo de instalación equivalente.

En resumen, es muy difícil demostrar que una solución cerrada proporciona ventajas que realmente ayuden contra el demostrablemente mayor coste en otras categorías, cuando tienen que competir con productos abiertos maduros para una función dada.

Ahora es tiempo de analizar los **Aspectos no cuantitativos**: En justicia, debemos hacer notar que no todos los aspectos se pueden cuantificar, entre ellos muchos de los más importantes incluyen la libertad, la protección contra los litigios por licencias, y la flexibilidad, además de la innovación que es otro importante y difícil aspecto a medir.

El código abierto protege a los usuarios de los riesgos y desventajas de las soluciones de proveedor único; aunque los defensores de los "programas libres" usan el término "libertad", y algunas empresas ponen el acento en términos diferentes como "diversificación", el concepto es el mismo: los usuarios no quieren quedar prisioneros de un único vendedor.

Las empresas en general prefieren comprar productos para los que tengan gran número de proveedores, porque esto reduce los riesgos: siempre pueden cambiar de proveedor cuando no estén satisfechos o cuando el proveedor original abandona el negocio, lo que se traduce en efectos sobre los productos mismos: si los clientes pueden elegir fácilmente y cambiar entre productos competidores, los precios bajan y la calidad se incrementa, a la inversa, cuando se da una situación próxima al monopolio sobre un producto, con el tiempo el proveedor eleva el precio y limitan sus usos a aquellos que benefician su monopolio.

Con código abierto, los usuarios pueden elegir entre diferentes distribuciones, y si un proveedor cierra, pueden cambiar a otro vendedor, como resultado, los proveedores se ven forzados a proporcionar productos y servicios de buena calidad por precios relativamente bajos, porque los usuarios pueden cambiar si no lo hacen, los usuarios pueden incluso reunirse y mantener el producto por ellos mismos, así fue como nació el proyecto Apache y así es como se hace posible que los grupos de usuarios se protejan ellos mismo del abandono.

El código abierto protege a los usuarios de litigios y de la gestión de licencias. Los vendedores cerrados ganan dinero de la venta de licencias, y están imponiendo nuevos mecanismos de manejar estas licencias cada vez más engorrosos para los consumidores, por ejemplo, Microsoft Windows XP exige la activación del producto, un esquema que provoca que una acumulación de cambios de hardware exija un nuevo código de activación. Los proveedores cerrados litigan contra aquellos que no cumplen con sus complicados requisitos de gestión de licencias, incrementando los riesgos legales para los usuarios, por contraste, los usuarios de código abierto no tienen miedo de litigios por el uso y copia de código abierto, los detalles del licenciamiento aparecen al modificarse y redistribuirse código abierto, pero para ser justos, el código cerrado sencillamente prohíbe esta acción, de forma que se convierte en un nuevo derecho; incluso en estas circunstancias, redistribuir programas abiertos modificados generalmente requiere seguir unas pocas reglas sencillas dependiendo de la licencia, tales como seguir dando crédito a los desarrolladores originales y publicar las modificaciones bajo el mismo esquema de licencia original.

El código abierto tiene mayor flexibilidad: los usuarios de código abierto pueden ajustar el producto tanto como sea necesario para conseguir cubrir sus necesidades en formas que no son posibles sin el código fuente o encontrar quien pueda resolver el problema, que incluso podría ser el desarrollador original del producto, algunos han proclamando que esto crea un "peligro de disgregación", es decir, múltiples versiones incompatibles de un mismo producto, esto sólo es un riesgo para los que creen que



la competencia es demoníaca, en la práctica, el alto coste de mantener el programa por uno mismo hace que los cambios se reviertan a la comunidad.

Hay buenas razones para creer que el código abierto potencia y no limita la innovación. Microsoft proclama públicamente que el código abierto, y particularmente su licencia más común, la GPL, eliminará la innovación, pero los hechos no justifican estas proclamas

En un estudio del prestigioso instituto independiente de investigación Forrester Research en el 2000, 2.500 gestores de TI entrevistados indicaron en un 84% que predecían que el código abierto sería crucial para conseguir innovaciones significativas en la industria, por supuesto, cuando examinamos las más destacadas innovaciones en Informática, se descubre rápidamente que Microsoft no ha realizado ninguna, ni ha sido nunca su primer implementador, por contraste, una buena porción de esas innovaciones clave eran proyectos abiertos, un ejemplo: Tim Berners-Lee, inventor de la World Wide Web, estableció en diciembre de 2001 **(30)** que "un factor muy significativo (en la amplia extensión del uso de la Web en la investigación científica) fue que todos los programas eran lo que ahora llamaríamos "de código abierto", se extendió rápidamente, y se potenció rápidamente, y pudo implantarse en instalaciones gubernamentales y de la gran industria sin tener que atravesar largos procesos de adquisición", nótese que esto no terminó una vez que las ideas originales se desarrollaron: el servidor de Web número uno del año 2001, Apache, es de código abierto, adicionalmente, los casos judiciales recientes han mostrado fuertes evidencias de que la única razón por la que el cerrado Internet Explorer es el primer navegador Web está en los años de uso ilegal del poder monopolístico de Microsoft, esta historia de la innovación no debería ser sorprendente: el código abierto se aproxima al método científico, permitiendo a cualquiera hacer aportaciones o añadir técnicas innovadoras, y ponerlas inmediatamente a disposición del público.

Se ha mostrado por qué la innovación está más próxima a los proyectos abiertos **(31)**. El sitio Web SweetCode informa de programas libres innovadores, aquí tenemos lo que Sweetcode dice de su sitio: "La innovación que muestran el conjunto de los programas sobre el que aquí informamos no es sólo una copia de algo, ni aun menos un añadido a algo, ni una traducción de algo, ni siquiera una nueva implementación de conceptos ya ampliamente conocidos... los programas recogidos en sweetcode deberían sorprenderle de alguna forma interesante".

Si la aproximación cerrada de Microsoft fuera mejor para los investigadores, se debería esperar que estuviese documentado en la comunidad investigadora, sin embargo, lo cierto es lo contrario: el

documento "NT Religious Wars: Why Are DARPA Researchers Afraid of Windows NT?", muestra que, a pesar de la fuerte presión de los clientes de pago, los investigadores informáticos se resistieron tenazmente a basar su investigación en Microsoft.

Otra forma de constatar los costos y beneficios de la presente migración es realizando la matriz DAFO (Debilidades, Amenazas, Fortalezas, Oportunidades). El análisis DAFO consiste en realizar una evaluación de los factores fuertes y débiles que en su conjunto diagnostican la situación interna de una organización, así como su evaluación externa; es decir, las oportunidades y amenazas.

También es una herramienta que puede considerarse sencilla y permite obtener una perspectiva general de la situación estratégica de una organización determinada. Thompson (1998) establece que el análisis DAFO estima el hecho que una estrategia tiene que lograr un equilibrio o ajuste entre la capacidad interna de la organización y su situación de carácter externo; es decir, las oportunidades y amenazas.

Las Fortalezas (F): Es un recurso de tipo interno que posee la institución en mejores condiciones que su competencia y que la hacen tener ventaja sobre los demás, y por lo tanto la hace rigurosa en su accionar. La Institución puede emplear estos elementos para lograr sus objetivos y mejorar su posición competitiva en el mercado. Son aquellos aspectos internos en lo que se es fuerte y que se debe mantener o mejorar para posicionarse adecuadamente en el Mercado; para un mejor entendimiento un ejemplo una fortaleza para llevar a cabo la migración es que se cuenta con el código fuente.

Las Amenazas (A): Son aquellas circunstancias o situaciones del entorno desfavorables para la empresa que pueden afectar negativamente la marcha de la institución, de no tomarse las medidas necesarias en el momento oportuno. Si la gerencia no toma una determinada acción estratégica ante esta tendencia desfavorable, que proviene del entorno, puede llevarla a su estancamiento o incluso su desaparición; para un mejor entendimiento un ejemplo una amenaza para llevar a cabo la migración es la existencia de la posibilidad de privatización de muchas comunidades por trasnacionales o grandes compañías.

Las Oportunidades (O): son aquellas circunstancias o situaciones del entorno (fuerzas que surgen desde fuera de las fronteras de una organización, pero que afectan sus decisiones y acciones internas, así como sus actividades y desarrollo); son potencialmente favorables para la institución u

organización que se quiera estudiar, para un mejor entendimiento un ejemplo una oportunidad para llevar a cabo la migración es la apertura de nuevos mercados basados en tecnologías libres.

Las Debilidades (D): Son limitaciones, defectos o inconsistencias en la institución, que constituyen un obstáculo para la consecución de los objetivos y una merma en la calidad de la gestión. Son recursos de tipo interno que poseen las instituciones y que por el solo hecho de poseerlo la hacen vulnerable en su accionar en relación a su competencia. Aspectos internos en los que se debe mejorar para lograr una posición más competitiva, para un mejor entendimiento un ejemplo una debilidad para llevar a cabo la migración es que se requiere de una inversión en formación.

Se hará una comparación entre dos matrices para analizar si es factible realizar la migración, los aspectos a analizar son las F, A, O, D teniendo en cuenta que:

1. Las fortalezas aprovechan las oportunidades.
2. Las fortalezas debilitan las amenazas.
3. Las debilidades desaprovechan las oportunidades.
4. Las debilidades acentúan las amenazas.

Se hará la comparación entre la cantidad de (X) de cada matriz y la que tenga mayor número de éstas, determinará cuál será la matriz a escoger, si para llevar a cabo la migración o para no llevar a cabo la migración.

A continuación se presentan las F, A, O, D. para conformar la matriz DAFO y analizar si es factible realizar la migración:

Fortalezas:

1. Elevado nivel técnico de los profesionales.
2. Software adaptable.
3. Mayor nivel de seguridad.
4. Se cuenta con el código fuente.
5. Independencia de proveedor.
6. Amplia comunidad de desarrollo.
7. Estabilidad y seguridad en los servidores.

Amenazas:

1. Mercado muy competitivo.
2. Poca documentación sobre software libre equivalente al de Señal ACN.
3. Falta de drivers para la compatibilidad del software libre con el hardware.
4. Ingreso de nuevos competidores.
5. Su desarrollo depende de la motivación del personal desarrollador y de la necesidad.
6. Existe la posibilidad de privatización de muchas comunidades por trasnacionales o grandes compañías.
7. Falta de acabado de los sistemas.

Oportunidades:

1. Mercado de software libre en expansión.
2. Política del país sobre las migraciones a software libre.
3. El software libre es un producto básicamente intelectual.
4. Posibilidad de exportar.
5. Amplia comunidad de desarrolladores.
6. Representa la alternativa para los países pobres que no pueden invertir su dinero en la compra de licencias.
7. Apertura de nuevos mercados basados en tecnologías libres.

Debilidades:

1. Falta de preparación y experiencia del personal.
2. Tecnología atrasada.
3. Requiere un cambio, un periodo de adiestramiento considerable.
4. No hay suficiente responsabilidad legal o soporte.
5. Requiere inversión en formación.
6. En algunos casos hay falta de interfaz amigable.
7. Falta de integración entre componentes y herramientas sobre todo en lo que se refiere a entornos de desarrollo de aplicaciones.

Atendiendo a las D.A.F.O. se creó la siguiente matriz que ratifica el desarrollo de la migración:

CAPITULO 3: DESARROLLO DE LA MIGRACION

		Oportunidades							Amenazas						
		O1	O2	O3	O4	O5	O6	O7	A1	A2	A3	A4	A5	A6	A7
Fortalezas	F1	X	X	X	X	X	-	X	-	X	-	X	X	-	X
	F2	X	X	-	X	X	X	X	X	X	-	-	-	X	X
	F3	-	X	-	X	-	X	-	-	X	-	X	-	-	X
	F4	X	X	X	X	-	X	X	X	X	-	-	-	-	X
	F5	X	X	X	X	X	-	X	X	-	-	X	X	X	X
	F6	X	X	X	X	X	X	X	X	X	X	-	X	X	X
	F7	-	-	X	X	-	-	-	X	-	-	-	-	-	X
Debilidades	D1	X	-	X	X	-	X	X	X	X	-	X	-	-	X
	D2	X	X	X	X	-	X	X	X	-	-	X	X	X	X
	D3	X	-	X	X	-	-	X	X	-	-	X	X	X	-
	D4	X	-	X	X	-	X	X	X	X	-	X	X	X	X
	D5	X	X	X	X	-	X	X	X	-	-	X	-	X	-
	D6	X	X	-	X	-	-	X	X	X	-	X	X	X	X
	D7	X	-	-	X	-	-	X	X	X	-	X	X	-	X

Tabla 3 Matriz DAFO para llevar a cabo la migración.

Para poder comparar y constatar el desarrollo de la migración se realizará otra matriz pero desde el punto de la no realización de la migración; a continuación se exponen sus F, A, O, D.

Fortalezas:

1. Alto nivel técnico de los profesionales.
2. Ampla experiencia en el mercado.

3. Prestigio.
4. Mayor peso de productos en el mercado.
5. Amplia gestión de marketing.
6. Existen muchas alianzas entre desarrolladores de software y hardware.
7. Interfaz amigable.

Amenazas:

1. El software libre, su aparición y desarrollo.
2. Movimiento revolucionario sobre migraciones a SWL.
3. Abundante documentación sobre software libre equivalente al de Señal ACN.
4. Integración entre componentes y herramientas del mundo del software libre.
5. Piratería.
6. Mayor número de ataques informáticos.
7. Dependencia tecnológica del proveedor.

Oportunidades:

1. Mercado de software propietario afianzando su posición en el mercado.
2. Acceso de Cuba a las licencias del Software propietario.
3. El software libre no es bien remunerado.
4. Posibilidades de exportar.
5. Preferencia por parte de muchos usuarios.
6. Interfaz amigable.
7. Grandes empresas proveedoras de software.

Debilidades:

1. Bajo nivel de Seguridad
2. Poca experiencia del personal.
3. Altos costes de licencia.
4. Frena la innovación.
5. No se tiene acceso al código fuente.
6. Mayor responsabilidad legal del usuario.
7. Costes por actualizaciones y nuevas versiones.

CAPITULO 3: DESARROLLO DE LA MIGRACION

Atendiendo a las D.A.F.O. se creó la siguiente matriz que ratifica la no realización de la migración:

		Oportunidades							Amenazas						
		O1	O2	O3	O4	O5	O6	O7	A1	A2	A3	A4	A5	A6	A7
Fortalezas	F1	X	X	-	X	-	X	X	X	X	-	-	X	-	X
	F2	X	-	-	X	-	X	X	X	X	-	X	X	X	-
	F3	X	X	-	X	X	-	X	-	-	-	-	X	X	-
	F4	X	X	-	X	X	X	X	X	X	-	-	-	-	X
	F5	X	X	X	X	X	-	X	X	-	-	X	X	X	X
	F6	X	-	-	X	X	X	X	X	X	-	-	-	-	-
	F7	-	-	-	X	X	X	-	X	X	-	-	-	-	-
Debilidades	D1	-	-	-	X	X	X	-	X	X	-	X	X	X	-
	D2	X	-	-	X	-	X	-	X	X	X	X	X	X	-
	D3	X	X	-	X	X	-	X	X	X	-	-	-	X	-
	D4	X	-	-	X	-	-	-	X	X	-	X	-	-	X
	D5	X	X	-	X	X	-	X	X	X	-	X	-	-	X
	D6	X	-	-	-	X	-	-	X	X	-	X	-	-	-
	D7	X	-	-	-	X	-	-	X	X	-	X	-	-	-

Tabla 4 Matriz DAFO para no llevar a cabo la migración.

Al analizar y comparar los resultados dados por las dos matrices y comparar las cantidades de (X) en cada una, se tiene que para la matriz para llevar a cabo la migración el total de X fue 129 y para la matriz de no llevar a cabo la migración el total de X fue 104, después de este análisis se llegó a la conclusión de que es muy certera la decisión de realizar la migración.

➤ **Atendiendo a los requisitos de hardware:**

La viabilidad de la propuesta debe verse atendiendo a varios aspectos, es por ello que se realizará una comparación entre los requerimientos de hardware del canal Señal ACN en su versión Privativo y en su versión Libre.

Privativo					
Servidor	Procesador	Memoria RAM	Disco Duro	Tarjeta de Red	Tarjeta de Video
Servidor Web/BD	Dual Xeon 2.8 GHz.	4 Gb	210 Gb.	2 Ethernet a 1 Gbps	
Servidor de Trasmisión	Dual Xeon 2.8 GHz.	1 Gb.	160 Gb	Ethernet a 100 Mbps	ATI RADEON X300
Libre					
Servidor Web/BD	Dual-Core Xeon 2.33 GHz	4 Gb	500 Gb	2 puertos Ethernet Gigabit	Capturadora Hauppauge WinTV PVR 350
Servidor de Trasmisión	Dual-Core Xeon 1.60 GHz	1 Gb	80 Gb	Ethernet Gigabit	Exportadora ATI Radeon X300

Tabla 5 Requisitos de Hardware.

Al realizar el enfrentamiento de matrices, la comparación de los requisitos de hardware, además de las razones antes expuestas; se pudo analizar el costo y los beneficios que reporta esta migración y se concluye que resulta factible el desarrollo de la misma.

3.3 Plan de migración

El plan de migración a software libre de Señal ACN se basa en una serie de gestiones que tienen impacto sobre; la plataforma, el sistema de almacenamiento de la información (Servidor de BD), las tecnologías informáticas y los servidores Web.

Este Plan está conformado por una serie de acciones agrupadas estratégicamente en etapas para lograr migrar los SO, la plataforma, los lenguajes, los IDEs, el sistema de BD y el servidor Web que posee Señal ACN, de acuerdo a las características que posee este sistema y de lo que haya sido mejor para su funcionamiento, de acuerdo a estudios realizados.

En el Capítulo II se realizó el estudio de las tecnologías existentes libres que se pudiesen emplear para la migración del sistema, así como los gestores de Bases de Datos, Sistemas Operativos y los Servidores Web que cumplen con las políticas del uso del software libre en Cuba y además atendiendo a las prestaciones que debían brindar para tener las funcionalidades necesarias para el sistema, y de la compatibilidad entre unos y otros para garantizar que el funcionamiento del canal no se viera comprometido.

Como resultado de este estudio, se propuso para cada módulo diferentes herramientas que se muestran a continuación.

Propuesta de Migración		
Módulo de Actualización de Recursos Multimedia.	Módulo de Administración.	Módulo de Transmisión.
Plataforma de desarrollo Mono.	Lenguaje de programación PHP con el framework Symfony	AJAX utilizando el framework Prototype.
Lenguaje de programación C#.	IDE de desarrollo Eclipse.	Sistema Operativo el Ubuntu.
IDE de desarrollo C# Develop.	Servidor Web Apache 2.0.	
	Gestor de BD se propuso PostgreSQL.	
	Sistema Operativo el Debian	

Tabla 6 Propuesta de migración.

Cuando se dispone a realizar la migración es más saludable ir cambiando las tecnologías poco a poco, de manera que no se vea afectado el funcionamiento del sistema y que el cambio no sea tan brusco para los usuarios, por lo que se propone a continuación un tentativo cronograma organizado por cuatro etapas en cada una de las cuales estará conformada por varios pasos para garantizar una migración exitosa.

Etapas I: Adiestramiento

En esta etapa es donde se capacita y adiestra al personal destinado a trabajar con la aplicación, es decir a todos los que interactúan directamente con ella, preparándolos para los cambios inherentes a la migración en su entorno del trabajo, sin que estos afecten el buen funcionamiento del canal.

Dicha capacitación será efectuada mediante cursos que impartirán profesionales debidamente preparado para ello (se recomienda crear vínculos con universidades cercanas en este sentido), los mismos están organizados en horas (h), quedando distribuidos de la siguiente manera:

CAPITULO 3: DESARROLLO DE LA MIGRACION

Cursos	Horas	Módulos	Descripción
GNU/Linux – Entornos de Escritorios (Debian y Ubuntu).	16h.	Administración y Transmisión respectivamente.	En este curso se impartirá todo lo relacionado con GNU/Linux específicamente los Entornos de Escritorios (Debian y Ubuntu).
Administración de servidor de Transmisión.	8h	Transmisión.	En este curso se capacitará al personal encargado de trabajar con el servidor de Transmisión.
Administración de servidor Apache.	8h.	Administración	En este curso se dará adiestramiento al personal en cuanto al trabajo con el servidor Web Apache.
Administración de PostgreSQL.	10h.	Administración	En este curso se dará adiestramiento al personal en cuanto al trabajo con el servidor de BD PostgreSQL.
Acercamiento a la filosofía de aplicaciones AJAX.	2h.	Transmisión.	Se dará una breve panorámica sobre las aplicaciones en AJAX.
Acercamiento a la filosofía de aplicaciones en C#.	2h.	Actualización de Recursos Multimedia	Se dará una breve panorámica sobre las aplicaciones en C#.

Tabla 7 Cursos para el adiestramiento.

Esta Etapa consume 46h, teniendo en cuenta que todo el personal debe recibir con carácter obligatorio los cursos impartidos y que éstos se desarrollaran de forma consecutiva.

Etapa II: Relevamiento

En esta etapa es donde se hará todo el relevamiento que consiste en hacer un levantamiento de todos los software que están instalados en cada PC, es decir dónde se encuentra cada aplicación y cada módulo que posibilita el funcionamiento del canal y luego encontrar todo el software homólogo con el cual darle soporte al canal.

Para este plan esta etapa ya fue realizada; durante el desarrollo del primer capítulo se investigó todo el software que le da soporte al canal con el fin de investigar durante la realización del capítulo II el soporte disponible para el mismo dentro del SWL, encontrándose todos los sustitutos viables para el desarrollo de la migración.

Etapa III: Implementación

Esta etapa está dedicada a la implementación de las aplicaciones que darán el nuevo soporte al canal en su versión libre. Cabe destacar que la implementación de estas aplicaciones no la harán el personal del canal sino un equipo de desarrollo conformado por estudiantes de la UCI, preparados debidamente por ello y que están aptos para la tarea.

Se propone que dicha implementación se realice en un tiempo máximo 45 días, distribuyéndose de la siguiente forma:

- Implementación del módulo de Administración: 45 días.
- Implementación del módulo de Transmisión: 30 días.
- Implementación del módulo de Actualización de Recursos Multimedia: 15 días

La implementación de los tres módulos se realizará de forma paralela, optimizando de esta forma el factor tiempo.

Etapa IV: Migración

En esta etapa se llevara a cabo la migración progresiva de los equipos de las distintas dependencias. Tal proceso será ejecutado de forma progresiva, comenzando por aquellas dependencias que no

utilicen aplicaciones críticas. El proceso de migración deberá ser ejecutado de forma más cuidadosa en la unidad de Administración, en la que se procederá a instalar Debian coexistiendo con Windows, puesto que debe darse un tiempo prudencial para que los usuarios se vayan adaptando al cambio. La secuencia del orden en el que se realizará la migración se muestra a continuación:

Pasos	Tiempo (en días)	Módulos
1. Instalar del SO (Debian) en el servidor Web/BD.	1	Administración
2. Instalar el APACHE (Servidor Web).	1	Administración
3. Instalar PostgreSQL (servidor de Base de Datos).	1	Administración
4. Instalar del SO (Ubuntu) en el servidor de Transmisión.	1	Transmisión
5. Cambiar el sitio de administración por el desarrollado en Symfony.	1	Administración
6. Cambiar la Macromedia destinada a la visualización de las noticias por el sitio construido en AJAX.	1	Transmisión
7. Cambiar la aplicación de actualización de ficheros multimedia por la desarrollada en C#.	1	Actualización de Recursos Multimedia

Tabla 8 Pasos de la etapa IV de migración.

Los pasos del 1 al 4 se realizarán de forma paralela y se le dedicará un día, es decir todas las instalaciones se realizarán en un día; a los siguientes pasos (5, 6, 7) se les dedicará 1 día para cada uno, es importante aclarar que todos estos pasos se llevarán a cabo luego de que culmine la etapa de implementación.

Tiempo estimado dedicado a la aplicación del plan:

Etapa I: Adiestramiento: 46h

Etapa II: Relevamiento: no se le dedica tiempo porque ya fue realizada durante el desarrollo de este trabajo.

Etapa III: Implementación: 45 días.

Etapa IV: Migración: 4 días.

Por lo que se estima que la aplicación del plan se realice en un período de tiempo de 49 días y 46h.

3.4 El módulo piloto

Luego de haber proporcionado todas las herramientas a utilizar para el desarrollo de la migración y basándonos en el plan propuesto anteriormente se desarrolló un prototipo funcional del módulo de Transmisión que fue implementado en AJAX con el framework Prototype, donde se demostró la viabilidad del plan.

Un Caso de uso en ingeniería del software es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

En este prototipo se implementaron los siguientes Casos de Usos (CU) del módulo de Visualización:

1. CU Visualizar Noticia.
2. CU Visualizar Cintillo Informativo.

Descripción del CU Visualizar Noticia:

Caso de uso	
CU-1	Visualizar Noticias.
Propósito	Mostrar de forma amena las noticias publicadas en el canal de teletexto.
Actor: Reloj	
Resumen: El caso de uso se inicializa cuando el actor Reloj detecta que en hay secciones activas. El sistema confecciona una cartelera organizada por secciones donde se muestra el título de las noticias. Luego se visualizan cada una de las noticias publicadas organizadas por secciones temáticas. Además se mostrarán los cintillos informativos activos.	
Precondiciones	Existen secciones activas.
Poscondiciones	Las noticias son visualizadas en el teletexto.
Acción del actor	Respuesta del sistema
1. El caso de uso se inicializa cuando el actor Reloj detecta que hay secciones activas.	2. El sistema muestra el video de presentación.
	3. Crea y muestra la cartelera para cada una de las secciones.
	4. Visualiza las noticias que se encuentran publicadas organizadas por secciones. Al mismo tiempo muestra los cintillos informativos y datos acerca de la sección que se muestra y la siguiente.
Puntos de Extensión	
<i>Acción 3. Se obtienen las noticias de las secciones activas y se crea la cartelera y. Ver CU Buscar Noticia y CU Crear Cartelera. (Anexo IV. Anexo IV.)</i>	
<i>Acción 4. Se muestran los cintillos informativos. Ver CU Visualizar Cintillos. (Anexo IV.)</i>	

Se muestra

el siguiente caso de prueba:

Un Caso de Prueba: En la Ingeniería del software, los casos de prueba son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Caso de Prueba: Visualizar Noticia

Propósito: Corroborar el buen funcionamiento de la visualización del prototipo del módulo de transmisión.

Pre requisitos: Existencia de secciones activas.

- Pasos:**
1. Abrir un navegador Web.
 2. teclear dirección.
 3. hacer en clic en "Ir"
 4. verificar si se visualiza correctamente la noticia activa en ese momento.

Posible resultado Visualización de la noticia.

Resultado obtenido Noticia visualizada.

A continuación se refleja todo el proceso etapa por etapa en la realización del prototipo para demostrar la factibilidad del plan propuesto, menos la etapa # 1, que no puede medirse su realización.

Etapa II: En esta etapa se conoció que el módulo de Visualización se encontraba en una PC del ICRT(Instituto Cubano de Radio y Televisión), que estaba implementado con director y su lenguaje base Lingo, además se buscaron alternativas dentro del SWL para darle soporte, encontrándose el AJAX y su framework Prototype.

Etapa III: Por el hecho de que es un prototipo y no contiene todas las funcionalidades, se dedicó a su implementación 15 días.

Se probó que cumplía con las funcionalidades de los Casos de Usos implementados, y que se desarrolló siguiendo el plan descrito anteriormente, demostrándose la factibilidad del mismo. **Ver**

, **Anexo 1, Anexo 2.**

3.5 Conclusiones Parciales Capítulo III

En este capítulo se ha realizado el desarrollo de la migración a software libre del sistema del canal Señal ACN, utilizando la tecnología, el servidor Web y el gestor de base de datos propuesto para la migración del sistema como resultado del estudio realizado durante la elaboración del Capítulo II. Además para demostrar la factibilidad de la propuesta realizada en el capítulo anterior se ha desarrollado un módulo piloto con el AJAX cumpliendo de esta forma con las funciones del Módulo de Visualización.

Conclusiones Generales

El desarrollo de la investigación correspondiente al presente trabajo de diploma arrojó que:

- El estudio riguroso de todas las tecnologías utilizadas para desarrollar el sistema de teletexto y de las herramientas desarrolladas en software libre con las cuales se propuso realizar la migración permitió el desarrollo y planteamiento de un plan que permite la guía del proceso de migración del canal hacia Software Libre, lo cual constituye el objetivo fundamental que rigió esta investigación, además de la presentación de un prototipo que demuestra la viabilidad de la propuesta de migración a realizar.

Recomendaciones

Luego de realizar un profundo análisis de un significativo número de elementos que contribuyeron a la elaboración del plan que guiará el proceso de migración de Señal ACN a software libre se hace necesario que se listen un grupo de recomendaciones que permitan la buena utilización del mismo:

- Capacitar a los usuarios finales para el trabajo con software libre, a fin de que posean una cultura general sobre el tema y tengan plena conciencia de la necesidad de su aplicación en el sistema.
- Profundizar en el estudio de migraciones a software libre con el objetivo de plantear otras etapas o pasos que no hayan sido comprendidos en la propuesta de solución y que puedan contribuir a la minimización de tiempos de migración y al mayor éxito de la misma.
- Realizar un seguimiento de todo el proceso de migración y velar por los resultados de la aplicación del plan propuesto.
- De resultar exitosa la aplicación del plan propuesto para llevar a cabo la migración de Señal ACN, proponer su extensión a todos los sistemas similares tanto dentro como fuera de la Universidad.

Bibliografía Consultada

1. alzado. *alzado*. [Online] http://www.alzado.org/articulo.php?id_art=457&s=1.
2. Apache. *Apache1*. [En línea] <http://www.apache.org/licenses/>.
3. Apache. *Apache*. [Online] http://www.hospedajeydominios.com/mambo/documentacion-manual_apache-pagina-45.html.
4. Aplicaciones Web. *Aplicaciones Web*. [Online] <http://blog.aplicacionesweb.cl/>.
5. Eclipse. *Eclipse*. [Online] http://campusvirtual.unex.es/cala/epistemowikia/index.php?title=Eclipse_SDK.
6. Guia-Ubuntu. *Guia-Ubuntu*. [Online] <http://www.guia-ubuntu.org/index.php?title=Eclipse>.
7. I'm a Mac. I'm a Blog. . *I'm a Mac. I'm a Blog.* . [Online] <http://jmsala.wordpress.com/2007/11/27/symfony-vs-cake-vs-ruby-on-rails/>.
8. Investic. *Investic*. [En línea] <http://www.investic.net/node/105>.
9. Linalco. *Linalco*. [En línea] <http://www.linalco.com/>.
10. Mono. *Mono*. [En línea] http://www.mono-project.com/Main_Page.
11. News. *News*. [En línea] <http://news.cnet.com>.
12. PostgreSQL. *PostgreSQL*. [En línea] <http://pgfoundry.org/>.
13. Programania. *Programania*. [Online] <http://www.programania.net/>.
14. Prototype. *Prototype*. [Online] <http://prototypejs.org/api> .
15. Revolution. *Revolution*. [Online] <http://www.disca.upv.es/magustim/mmmMultiplataforma/c2361.htm>.
16. Software Libre. *Software Libre*. [En línea] <http://www.softwarelibre.net/>.
17. Symfony. *Symfony1*. [Online] <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>
18. Symfony. *Symfony*. [Online] <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-ii-parte/>.
19. User Linux. *User Linux*. [Online] <http://www.userlinux.net/>.
20. XOOPS. *XOOPS*. [Online] <http://www.esxoops.com/>.

Referencia Bibliográfica


1. Manifiesto del Software Libre. Manifiesto del Software Libre. [Online] [Cited: diciembre 15, 2007.] http://www.cemev.gob.mx/software_libre/html/propietario.html.
2. Hernández, Ruber Hernández Garcia y Yuniór Montaner. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas. Sistema Automatizado para la plataforma de televisión Digital Satelital Cubana. [Online]
3. Lenguajes de Programación. Lenguajes de Programación. [Online] [Cited: febrero 8, 2008.] <http://lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
4. Alvarez, Rubén. DesarrolloWeb.Introducción al HTML. . DesarrolloWeb.Introducción al HTML. . [Online] [Cited: febrero 8, 2008.] <http://www.desarrolloweb.com/articulos/534.php>.
5. Alvarez, Miguel Angel. Introducción al JavaScript. DesarrolloWeb. Introducción al JavaScript. DesarrolloWeb. [Online] [Cited: febrero 7, 2008.] <http://www.desarrolloweb.com/articulos/490.php>.
6. Introducción a la programación en ASP. DesarrolloWeb. Introducción a la programación en ASP. DesarrolloWeb. [Online] [Cited: febrero 5, 2008.] <http://www.desarrolloweb.com/articulos/244.php>.
7. Gross, Phil. Director MX y Lingo. [multimedia] Madrid : ANAYA Multimedia, 2003 pag 784.
8. Sierra, Francisco Javier Ceballos. El lenguaje de programación C#. México D.F : Alfaomega, 2002. pág. 320.
9. Su, Stanley. Database Computers: Principles, Architectures and Techniques. Nueva York : McGraw-Hill, 1988. pág. 511.
10. CUBA VS BLOQUEO.Muros virtuales (I). [Online] [Cited: febrero 19, 2008.] <http://www.cubavsbloqueo.cu/Default.aspx?tabid=1085>.
11. BRASIL, PSL. Declaración de La Habana sobre el uso de Software Libre y Abierto. BRASIL, PSL. Declaración de La Habana sobre el uso de Software Libre y Abierto. [Online] [Cited: febrero 19, 2008.] <http://www.softwarelivre.org/articles/43>.
12. Abel Meneses Abad, Nilet Soto López , Rafael Rodríguez Montero. Informatización Para la Sociedad Cubana desde la perspectiva ideológica del uso de las plataformas Software Libre. 15 p.

13. Rojas, Frank Benavides Dalmendray y Darien Cepero. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas. Estudio para la migración del sistema automatizado para la gestión académica "Akademos" a software libre. [Online].
14. MONO. Mono. MONO. Mono. [Online] [Cited: febrero 19, 2008.] http://www.mono-project.com/Main_Page.
15. MONO. Software . MONO. Software . [Online] [Cited: febrero 20, 2008.] <http://www.mono-project.com/Software>.
16. MONO. Gui Toolkits . MONO. Gui Toolkits . [Online] [Cited: febrero 20, 2008.] http://www.mono-project.com/Gui_Toolkits.
17. MONO. MONO. [Online] [Cited: febrero 20, 2008.] <http://www.mono-project.com>.
18. IDE. Artículos. IDE. Artículos. [Online] [Cited: febrero 20, 2008.] <http://www.desarrolloweb.com/articulos>.
19. IDE. [Online] [Cited: febrero 26, 2008.] Disponible en: <http://www.sharpdevelop.net/>.
20. IDE Eclipse. [Online] [Cited: marzo 20, 2008.] www.eclipse.org/downloads.
21. MONO. Main_Page. MONO. Main_Page. [Online] [Cited: febrero 22, 2008.] http://www.monodevelop.com/Main_Page.
22. MONO. Architecture_Overview. MONO. Architecture_Overview. [Online] [Cited: febrero 20, 2008.] http://www.monodevelop.com/Architecture_Overview.
23. AJAX. AJAX. [Online] [Cited: abril 7, 2008.] <http://ajaxian.com/archives/ajaxiancom-2006-survey-results>.
24. Servidores Web más utilizados. Servidores Web más utilizados. [Online] [Cited: abril 5, 2008.] http://news.netcraft.com/archives/2003/09/01/september_2003_web_server_survey.html.
25. Firebird. Firebird. [Online] [Cited: abril 15, 2008.] <http://www.firebird.com.mx/modules/news/>.
26. Windows Server. Windows Server. [Online] [Cited: mayo 20, 2008.] <https://www.softworld.es/tienda/precios.cgi?SID=MgoC8gOVDxHjU&command=review>.

BIBLIOGRAFIA REFERENCIADA Y WEBGRAFIA


27. Windows XP. Windows XP. [Online] [Cited: mayo 20, 2008.] http://www.areapc.com/_microsoft_windows_xp_home_sp2_10405814_p.html.
28. Ubuntu. Ubuntu. [Online] [Cited: mayo 25, 2008.] <http://www.linux-cd.com.ar>.
29. How Linux saved Amazon millions. How Linux saved Amazon millions. [Online] [Cited: mayo 1, 2008.] <http://news.cnet.com/2100-1001-275155.html&tag=owv>.
30. Charting the Web's next transformation. Charting the Web's next transformation. [Online] [Cited: junio 1, 2008.] <http://news.cnet.com/2008-1082-276939.html>.
31. Linux Today. Linux Today. [Online] [Cited: junio 1, 2008.] http://www.linuxtoday.com/news_story.php3?ltsn=1999-08-01-001-10-NW-CY.

Anexos



Venezuela elevó reservas probadas de crudo

También nuestras empresas estatales de Suramérica están participando en esta certificación; ya tenemos la propuesta del plan de desarrollo y es uno de los ejes fundamentales del Tratado de Seguridad Energética para disponer, en materia de petróleo, de las reservas que se requieran para el desarrollo de nuestra región, indicó Ramírez.



Tiempo restante:
3 seg
Pagina
3 de 4

A
continuación:
Energías:
Estudian
importar gas
licuado de
Venezuela

UNASUR firmo tratado constitutivo



Venezuela aumentó sus reservas de crudo

El también presidente de la estatal Petróleos de Venezuela (Pdvsa), la quinta empresa petrolera del mundo, adelantó que para el 2009 se tiene la meta de certificar 235 mil millones más de barriles de petróleo. En cuanto a la cooperación energética que Venezuela mantiene con algunos países suramericanos, Rafael Ramírez opinó que estas son claras muestras del avance de la integración de América del Sur.



Tiempo
restante:
0 seg
Pagina
3 de 3

A
continuacion:
Hidrocarburos:
Venezuela
elevó reservas
probadas de
crudo

Anexo 1 Prototipo. Sección Actualidad




Venezuela aumentó sus reservas de crudo

El también presidente de la estatal Petróleos de Venezuela (Pdvsa), la quinta empresa petrolera del mundo, adelantó que para el 2009 se tiene la meta de certificar 235 mil millones más de barriles de petróleo. En cuanto a la cooperación energética que Venezuela mantiene con algunos países suramericanos, Rafael Ramírez opinó que estas son claras muestras del avance de la integración de América del Sur.




Tiempo
restante:
0 seg
Pagina
3 de 3

A
continuacion:
Hidrocarburos:
Venezuela
elevó reservas
probadas de
crudo



Estudian importar gas licuado de Venezuela

"El problema que se genera hoy en el mundo ante la escasez de alimentos es por una distribución desigual de la riqueza", subrayó el mandatario. Recomendó volver a las costumbres ancestrales de producción porque somos pueblos de maíz, camote, zapotes entre otras frutas. Pidió al pueblo incentivar la producción a través de los huertos familiares y cultivos hidropónicos y que los gobiernos distribuyan semillas y herramientas agrícolas.



DEL MUNDO

Tiempo restante
1 seg

Página
4 de 5

A
continuacion:
Petroquímica:
Venezuela
propone crear
un banco de
insumos
agrícolas y
créditos

Anexo 2 Prototipo.Sección Del Mundo

Anexo IV. Descripción de casos de usos secundarios.

Buscar Usuario: Es iniciado por los actores Personal de Redacción y Jefe de Redacción al autenticarse en el sistema y gestionar usuarios respectivamente respondiendo al requisito funcional R1 y R2. El caso de uso verifica que un usuario exista en el sistema y devuelve sus datos en caso de que esté en el sistema.

Buscar Noticia: Es iniciado por los actores Corrector, cuando está en el proceso de corrección de noticias, Editor, cuando está publicando o gestionando las noticias y finalmente es ejecutado por el

actor Reloj al visualizar las noticias respondiendo a los requisitos funcionales R6, R7 y R12. El caso de uso recupera las noticias de una sección en particular ofreciendo los datos de ella.

Visualizar Pantalla: Es iniciado por los actores Corrector, cuando está en el proceso de corrección de noticias, Editor, cuando está publicando o gestionando las noticias respondiendo a los requisitos funcionales R6, R7. El caso de uso permite el acceso a la información contenida dentro de una pantalla de una noticia seleccionada, mostrando así todos sus atributos como opciones para modificarlos.

Pre-Visualizar Noticia: Es iniciado por los actores Redactor, al redactar las noticias, Corrector, cuando está en el proceso de corrección de noticias, Editor, cuando está publicando las noticias, así cada actor tendrá una visión de cómo se verá la noticia en el teletexto del canal respondiendo al requisito funcional R4. El caso de uso inserta los atributos título y contenido de las pantallas de la noticia dentro de una interfaz igual a la del teletexto, para así ofrecer una idea de cómo se verá en el canal.

Crear Cartelera: Es iniciado por el actor Reloj a la hora de visualizar las noticias cumpliendo con el requisito R12.1. El caso de uso toma los títulos de las noticias publicadas mostrándolos organizados por las secciones a la que pertenecen y la hora en que se visualizará dicha sección.

Visualizar Cintillos: Es iniciado por el actor Reloj a la hora de visualizar las noticias cumpliendo con el requisito R9. La funcionalidad de este caso de uso es visualizar los cintillos insertados por el Editor, crea además avances noticiosos a partir de los titulares de la próxima sección a mostrar.

Imprimir Reporte: Es inicializado por el Jefe de Redacción cuando requiere imprimir algún reporte. Imprime los datos del reporte según el formato establecido para cada tipo de reporte.



Glosario de Términos

ACN: Agencia Cubana de Noticias, es una división de la AIN (agencia de información nacional) que se centra mayormente en noticias de repercusión nacional y local.

Migración: La migración a Software Libre se refiere a un conjunto de actuaciones cuya finalidad es la sustitución de infraestructuras tecnológicas apoyadas en software propietario por otras con funciones equivalentes basadas en Software Libre.

Software libre: Según la Free Software Foundation (FSF), el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie, donde el requisito indispensable es el acceso al código fuente es un requisito previo.

Software Propietario: Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido.

Hipertexto: En computación, hipertexto es un paradigma en la interfaz del usuario cuyo fin es el de presentar documentos que puedan, según la definición de Ted Nelson, "bifurcarse o ejecutarse cuando sea solicitado" (branch or perform on request). La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos. Si el usuario selecciona un hipervínculo, hace que el programa de computador muestre el documento enlazado en un corto periodo de tiempo.

Lenguaje Interpretado: Lenguaje de programación diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

Plataforma .NET Conjunto de tecnologías para desarrollar y utilizar componentes que permiten crear formularios Web, servicios Web y aplicaciones de Windows.

SDK: Es un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema bastante concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, ordenadores, videoconsolas, sistemas operativos, etcétera.

MVC: Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

ORM: Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos.

Zonas de Silencio: lugares de la geografía nacional en los que viven alrededor de 70 mil cubanos y donde hasta hace muy poco no llegaba ni tan siquiera la señal de televisión o radio.

Framework: es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.